Fall 2019

Neuroengineering of Clustering Algorithms

Leonardo Enzo Brito da Silva

Department: Electrical and Computer Engineering

Recommended Citation

Brito da Silva, Leonardo Enzo, "Neuroengineering of Clustering Algorithms" (2019). *Doctoral Dissertations*. 2828.
https://scholarsmine.mst.edu/doctoral_dissertations/2828

NEUROENGINEERING OF CLUSTERING ALGORITHMS

by

LEONARDO ENZO BRITO DA SILVA

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

2019

Approved by

Dr. Donald C. Wunsch II, Advisor
Dr. Randy H. Moss
Dr. R. Joe Stanley
Dr. Jiangfan Zhang
Dr. Cihan H. Dagli

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following seven articles, which have been published or will be submitted for publication, as follows:

Paper I: Pages 6-135 have been published in the Neural Networks Journal, 2019.

Paper II: Pages 136-156 have been published in the Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2017.

Paper III: Pages 157-178 have been published in the Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), pp. 2351-2358, 2018.

Paper IV: Pages 179-191 have been published in the Neural Networks Journal, vol. 109, pp. 1-5, 2019.

Paper V: Pages 192-252 have been published in the Neural Networks Journal, vol. 121, pp. 208-228, 2020.

Paper VI: Pages 253-302 have been published in the IEEE Transactions on Neural Networks and Learning Systems Journal, vol. 29, no. 6, pp. 2595-2613, 2018.

Paper VII: Pages 303-366 are intended for submission to the IEEE Access Journal.

ABSTRACT

Cluster analysis can be broadly divided into multivariate data visualization, clustering algorithms, and cluster validation. This dissertation contributes neural network-based techniques to perform all three unsupervised learning tasks. Particularly, the first paper provides a comprehensive review on adaptive resonance theory (ART) models for engineering applications and provides context for the four subsequent papers. These papers are devoted to enhancements of ART-based clustering algorithms from (a) a practical perspective by exploiting the visual assessment of cluster tendency (VAT) sorting algorithm as a preprocessor for ART offline training, thus mitigating ordering effects; and (b) an engineering perspective by designing a family of multi-criteria ART models: dual vigilance fuzzy ART and distributed dual vigilance fuzzy ART (both of which are capable of detecting complex cluster structures), merge ART (aggregates partitions and lessens ordering effects in online learning), and cluster validity index vigilance in fuzzy ART (features a robust vigilance parameter selection and alleviates ordering effects in offline learning). The sixth paper consists of enhancements to data visualization using self-organizing maps (SOMs) by depicting in the reduced dimension and topology-preserving SOM grid information-theoretic similarity measures between neighboring neurons. This visualization's parameters are estimated using samples selected via a single-linkage procedure, thereby generating heatmaps that portray more homogeneous within-cluster similarities and crisper between-cluster boundaries. The seventh paper presents incremental cluster validity indices (iCVIs) realized by (a) incorporating existing formulations of online computations for clusters' descriptors, or (b) modifying an existing ART-based model and incrementally updating local density counts between prototypes. Moreover, this last paper provides the first comprehensive comparison of iCVIs in the computational intelligence literature.

ACKNOWLEDGMENTS

First and foremost, I thank God for guiding and blessing me in my life's journey.

Next, I would like to thank Dr. Donald C. Wunsch for not only kindly sharing his immense knowledge, but also for all the academic, career, and personal advice. I am also grateful to Dr. José A. Costa for initiating me into the fascinating world of machine learning. I appreciate the effort and time that my advisory committee members, Dr. Moss, Dr. Stanley, Dr. Zhang, and Dr. Dagli devoted to reviewing this dissertation.

No words can fully express my thankfulness to my beloved parents, Fernando and Lucia, whom despite the immense geographical distance, walked this path by my side and nurtured me with their unconditional love, words of wisdom and encouragement, and invaluable support throughout my entire life. Indeed, family is a man's most precious treasure, his home, his fortress.

Thanks to all ACIL members, particularly my graduate brothers Islam, Seaar, Yongliang, Khalid, Dao, Niklas, Dustin, Sejun, Bryce, Clayton, Tao, Ronit, Raghu, Louis, Austin, Sasha, Blaine, and Sima, for all the insightful discussions and for making the ACIL such a positive and thriving working environment. Additionally, I would also like to extend my appreciation to my research collaborators, Dr. Tauritz, Marketa, Hamed, and Ethar, from whom I learned topics outside my field of study; it was a pleasure to work with you.

Thanks to my dear friends Carlos, Davi, Bruno and Roberta, Samir, Raúl, Aziz, Marty, Ray, Daniel and Margarita, Sujit and Tejaswi, Krishnan, Viraj, Jack, and Kartikeya, who have made the Ph.D. experience much more enjoyable. I also would like to thank the International Affairs Team (Amy, Shawna, Hannah, Jodi, Gene, Robin, and Richard) for all their help and support at Missouri S&T.

Finally, I would like to acknowledge the financial support from CAPES.

Thank you all. Without you, this would have never been possible.

TABLE OF CONTENTS

Page

PUBLICATION DISSERTATION OPTION iii

ABSTRACT iv

ACKNOWLEDGMENTS v

LIST OF ILLUSTRATIONS xvii

LIST OF TABLES xxiv

SECTION

1. INTRODUCTION 1

1.1. ADAPTIVE RESONANCE THEORY 1

1.2. SELF-ORGANIZING MAPS 2

1.3. CLUSTER VALIDATION 2

1.4. RESEARCH CONTRIBUTIONS 3

1.4.1. Adaptive Resonance Theory Neural Network Models Review 3

1.4.2. Clustering Algorithms 3

1.4.2.1. The VAT and fuzzy ART framework (VAT + FA) 3

1.4.2.2. Dual vigilance fuzzy ART (DVFA) 4

1.4.2.3. Distributed dual vigilance fuzzy ART (DDVFA) 4

1.4.2.4. Cluster validity index vigilance test in fuzzy ART (CVIFA) 4

1.4.3. Data Visualization: Information-Theoretic Visualization for SOM (IT-vis) 5

1.4.4. Cluster Validation: Extensions and Comparative Study on Incremental Cluster Validity Indices (iCVIs) 5

PAPER

I. A SURVEY OF ADAPTIVE RESONANCE THEORY NEURAL NETWORK MODELS FOR ENGINEERING APPLICATIONS 6

ABSTRACT 6

1. INTRODUCTION 7

2. ART MODELS FOR UNSUPERVISED LEARNING......................... 9

2.1. ELEMENTARY ARCHITECTURES 9

2.1.1. ART 1 12

2.1.2. ART 2 15

2.1.3. Fuzzy ART.. 17

2.1.4. Fuzzy Min-Max 19

2.1.5. Distributed ART 20

2.1.6. Gaussian ART 22

2.1.7. Hypersphere ART 24

2.1.8. Ellipsoid ART 25

2.1.9. Quadratic neuron ART 27

2.1.10. Bayesian ART 28

2.1.11. Grammatical ART 30

2.1.12. Validity Index-Based Vigilance Fuzzy ART 32

2.1.13. Dual Vigilance Fuzzy ART 33

2.2. TOPOLOGICAL ARCHITECTURES 34

2.2.1. Fuzzy ART with Group Learning 35

2.2.2. TopoART 35

2.3. HIERARCHICAL ARCHITECTURES 37

2.3.1. ARTtree 37

2.3.2. Self-Consistent Modular ART 38

2.3.3. ArboART 39

2.3.4. Joining Hierarchical ART 39

2.3.5. Hierarchical ART with Splitting 40

2.3.6. Distributed Dual Vigilance Fuzzy ART 42

2.4. BICLUSTERING AND DATA FUSION ARCHITECTURES 45

2.4.1. Fusion ART 45

2.4.2. Biclustering ARTMAP 47

2.4.3. Generalized Heterogeneous Fusion ART 49

2.4.4. Hierarchical Biclustering ARTMAP 51

2.5. SUMMARY 51

3. ART MODELS FOR SUPERVISED LEARNING 51

3.1. ARCHITECTURES FOR CLASSIFICATION 51

3.1.1. ARTMAP 54

3.1.2. Fuzzy ARTMAP 58

3.1.3. Fuzzy Min-Max 60

3.1.4. Fusion ARTMAP 60

3.1.5. LAPART 61

3.1.6. ART-EMAP 62

3.1.7. Adaptive Resonance Associative Map 64

3.1.8. Gaussian ARTMAP 65

3.1.9. Probabilistic Fuzzy ARTMAP 66

3.1.10. ARTMAP-IC 68

3.1.11. Distributed ARTMAP 70

3.1.12. Hypersphere ARTMAP 74

3.1.13. Ellipsoid ARTMAP 74

3.1.14. μARTMAP 74

3.1.15. Default ARTMAPs 77

3.1.16. Boosted ARTMAP 79

3.1.17. Fuzzy ARTMAP with Input Relevances 81

3.1.18. Bayesian ARTMAP 83

3.1.19. Generalized ART 85

3.1.20. Self-Supervised ARTMAP 87

3.1.21. Biased ARTMAP 89

3.1.22. TopoART-C 90

3.2. ARCHITECTURES FOR REGRESSION 91

3.2.1. PROBART 91

3.2.2. FasArt and FasBack 93

3.2.3. Fuzzy ARTMAP with Input Relevances 95

3.2.4. Generalized ART 95

3.2.5. TopoART-R 96

3.2.6. Bayesian ARTMAP for Regression 96

3.3. SUMMARY 97

4. ART MODELS FOR REINFORCEMENT LEARNING 97

4.1. REACTIVE FALCON 97

4.2. TEMPORAL DIFFERENCE FALCON 99

4.3. UNIFIED ART 101

4.4. EXTENDED UNIFIED ART 103

5. ADVANTAGES OF ART 105

5.1. SPEED 105

5.2. CONFIGURABILITY 106

5.3. EXPLAINABILITY 106

5.4. PARALLELIZATION AND HARDWARE IMPLEMENTATION .. 107

6. ART CHALLENGES AND OPEN PROBLEMS 107

6.1. INPUT ORDER DEPENDENCY 107

6.2. VIGILANCE PARAMETER ADAPTATION 108

6.3. CATEGORY PROLIFERATION 109

6.4. ONLINE NORMALIZATION 111

6.5. NEW METRICS 111

6.6. DISTRIBUTED REPRESENTATIONS 113

6.7. DICHOTOMY OF MATCH- AND ERROR-BASED LEARNING .. 113

7. CODE REPOSITORIES 114

8. CONCLUSIONS 114

ACKNOWLEDGEMENTS 115

REFERENCES 115

II. VALIDITY INDEX-BASED VIGILANCE TEST IN ADAPTIVE RESONANCE THEORY NEURAL NETWORKS 136

ABSTRACT 136

1. INTRODUCTION 137

2. BACKGROUND AND RELATED WORK 140

2.1. FUZZY ART 140

2.2. CLUSTER VALIDATION 142

2.2.1. Calinski-Harabasz (CH) Index 143

2.2.2. Pakhira-Bandyopadhyay-Maulik (PBM) Index 143

2.2.3. Davies-Bouldin (DB) Index 143

2.2.4. Silhouette (SIL) Index 143

3. METHODOLOGY 144

4. EXPERIMENTAL SETUP 145

5. RESULTS AND DISCUSSION 147

5.1. EXPERIMENT 1: RANDOM PRESENTATION 147

5.2. EXPERIMENT 2: ORDERED PRESENTATION 150

6. CONCLUSION 151

ACKNOWLEDGEMENTS 152

REFERENCES 152

III. A STUDY ON EXPLOITING VAT TO MITIGATE ORDERING EFFECTS IN FUZZY ART 157

ABSTRACT 157

1. INTRODUCTION 157

2. BACKGROUND 161

2.1. FUZZY ART 161

2.2. VISUAL ASSESSMENT OF CLUSTER TENDENCY (VAT) 162

3. METHODOLOGY 163

4. EXPERIMENTAL SETUP 165

5. RESULTS AND DISCUSSION 167

6. CONCLUSION 172

ACKNOWLEDGEMENTS 173

REFERENCES 173

IV. DUAL VIGILANCE FUZZY ADAPTIVE RESONANCE THEORY 179

ABSTRACT 179

1. INTRODUCTION 180

2. BACKGROUND AND RELATED WORK 181

2.1. FUZZY ART 181

2.2. FUZZY TOPOART 182

3. DUAL VIGILANCE FUZZY ART 183

4. EXPERIMENTAL SET-UP 184

5. RESULTS AND DISCUSSION 185

6. CONCLUSIONS 189

ACKNOWLEDGEMENTS 189

REFERENCES 189

V. DISTRIBUTED DUAL VIGILANCE FUZZY ADAPTIVE RESONANCE THEORY LEARNS ONLINE, RETRIEVES ARBITRARILY-SHAPED CLUSTERS, AND MITIGATES ORDER DEPENDENCE 192

ABSTRACT 192

1. INTRODUCTION 193

2. BACKGROUND AND RELATED WORK 195

2.1. ABBREVIATED REVIEW 196

2.2. FUZZY ART 198

2.3. FUZZY TOPOART 200

2.4. DUAL VIGILANCE FUZZY ART 201

3. DISTRIBUTED DUAL VIGILANCE FUZZY ART 203

3.1. DDVFA ARCHITECTURE 203

3.2. MERGE ART MODULE 209

4. EXPERIMENTAL SETUP 214

4.1. DATA SETS 214

4.2. CLUSTERING ALGORITHMS AND PARAMETER TUNING 216

4.2.1. ART-Based Clustering Methods 217

4.2.2. Non-ART-Based Clustering Methods 217

4.3. CLUSTERING PERFORMANCE ASSESSMENT 218

4.4. STATISTICAL ANALYSIS METHODOLOGY 218

4.5. SOFTWARE AND CODE 219

5. RESULTS AND DISCUSSION 219

5.1. DDVFA RESULTS WITH PRE- AND POST-PROCESSING 219

5.1.1. Statistical Analysis of Performance 221

5.1.2. Summary 222

5.2. PERFORMANCE COMPARISON 1: ART-BASED CLUSTERING ALGORITHMS 222

5.2.1. Statistical Analysis of Performance 223

5.2.2. Statistical Analysis of Compactness 225

5.2.3. Summary 227

5.3. PERFORMANCE COMPARISON 2: NON-ART-BASED CLUSTERING ALGORITHMS 228

5.4. COMPUTATIONAL COMPLEXITY ANALYSIS 230

5.5. DDVFA'S HYPER-PARAMETERIZATION 231

5.6. SENSITIVITY TO KERNEL WIDTH PARAMETER 232

6. CONCLUSION 236

APPENDIX 241

SUPPLEMENTARY DATA 242

ACKNOWLEDGEMENTS 242

REFERENCES 242

VI. AN INFORMATION-THEORETIC-CLUSTER VISUALIZATION FOR SELF-ORGANIZING MAPS 253

ABSTRACT 253

1. INTRODUCTION 254

2. BACKGROUND AND RELATED WORK 259

2.1. SELF-ORGANIZING MAPS 259

2.2. SOM-BASED VISUALIZATION METHODS 260

2.2.1. Image-Based Visualizations 260

2.2.2. Graph-Based Visualizations 261

2.2.3. Projection-Based Visualizations 262

2.3. INFORMATION THEORETIC CLUSTERING 265

3. INFORMATION-THEORETIC VISUALIZATION 267

4. EXPERIMENTAL SET-UP ... 269

5. VISUALIZATION EXPERIMENTS ... 271

5.1. REPRESENTATIVE VISUALIZATIONS ... 271

5.2. *K*-NN METHODS AND SENSITIVITY TO THE *MINPTS* PARAMETER ... 275

5.3. VISUALIZATION EXAMPLES AND SENSITIVITY TO SOM SIZE ... 276

6. A CLUSTERING APPLICATION THROUGH IT-VIS ... 282

6.1. RESULTS AND DISCUSSIONS ... 285

7. CONCLUSION ... 293

ACKNOWLEDGEMENTS ... 295

REFERENCES ... 295

VII. INCREMENTAL CLUSTER VALIDITY INDICES FOR HARD PARTITIONS: EXTENSIONS AND COMPARATIVE STUDY ... 303

ABSTRACT ... 303

1. INTRODUCTION ... 304

2. BACKGROUND AND RELATED WORK ... 306

2.1. BATCH CLUSTER VALIDITY INDICES (CVIS) ... 306

2.1.1. Calinski-Harabasz (CH) ... 307

2.1.2. WB-Index (WB) ... 307

2.1.3. Davies-Bouldin (DB) ... 308

2.1.4. Xie-Beni (XB) ... 308

2.1.5. Generalized Dunn's Indices (GDs) ... 309

2.1.6. Pakhira-Bandyopadhyay-Maulik (PBM) ... 309

2.1.7. Silhouette (SIL) ... 310

2.1.8. Partition Separation (PS) ... 310

2.1.9. Negentropy Increment (NI) ... 311

2.1.10. Representative Cross Information Potential (rCIP) 312

2.1.11. Conn_Index 313

2.2. INCREMENTAL CLUSTER VALIDITY INDICES (ICVIS) 314

2.3. ADAPTIVE RESONANCE THEORY (ART) 317

2.3.1. Fuzzy ART 317

2.3.2. Fuzzy ARTMAP 318

3. EXTENSIONS OF ICVIS 318

3.1. INCREMENTAL CALINSKI-HARABASZ INDEX (ICH) 319

3.2. INCREMENTAL WB INDEX (IWB) 320

3.3. INCREMENTAL PAKHIRA-BANDYOPADHYAY-MAULIK INDEX (IPBM) 320

3.4. INCREMENTAL SILHOUETTE INDEX (ISIL) 321

3.5. INCREMENTAL NEGENTROPY INCREMENT (INI) 323

3.6. INCREMENTAL REPRESENTATIVE CROSS INFORMATION POTENTIAL (IRCIP) AND CROSS-ENTROPY (IRH) 323

3.7. INCREMENTAL CONN_INDEX (ICONN_INDEX) 324

4. NUMERICAL EXPERIMENTS DESIGN 326

5. A COMPARATIVE STUDY 328

5.1. CORRECT PARTITIONS 331

5.2. UNDER-PARTITIONS 334

5.3. OVER-PARTITIONS 339

5.4. EXPERIMENTS WITH REAL-WORLD DATA SETS 345

5.5. VISUALIZATION POWER 349

6. A CLOSER LOOK AT ICONN_INDEX 350

7. CONCLUSION 353

ACKNOWLEDGEMENTS 359

REFERENCES 359

SECTION

2. SUMMARY AND CONCLUSIONS 367

REFERENCES 369

VITA 407

LIST OF ILLUSTRATIONS

Figure Page

PAPER I

1. Elementary ART model underlying various WTA designs. The orienting subsystem uses the vigilance threshold to regulate whether ART can go into resonance or if it must reset. 11

2. Elementary ARTMAP model. 53

PAPER II

1. Data sets depicted using principal component analysis projection (a, b, c) and rH*-vis (d, e, f). The 10 × 10 SOMs were trained using the SOM Toolbox (Vesanto et al., 1999). 146

2. Experiment 1: results for the *Iris*, *Seeds* and *Wine* data sets: Rand, adjusted Rand and number of clusters (first, second and third rows, respectively). The CH, PBM, SIL, DB and standard Fuzzy ART (FA) are represented as red triangles, violet squares, black diamonds, green stars and blue circles, respectively. 148

3. Experiment 2: results for the *Iris*, *Seeds* and *Wine* data sets: Rand, adjusted Rand and number of clusters (first, second and third rows, respectively). The CH, PBM, SIL, DB and standard Fuzzy ART (FA) are represented as red triangles, violet squares, black diamonds, green stars and blue circles, respectively. 149

PAPER III

1. Euclidean distance matrix of the *Tetra* data set before and after (VAT) reordering. Dark blocks in the main diagonal indicate cluster tendency. 163

2. Illustration of Fuzzy ART's input order dependency using *Lsun* data set (Ultsch, 2005): the clustering solution on the right side (cluster-by-cluster presentation) is perceptibly better than the one on the left hand side (random input presentation). 164

3. First row {(a), (b)}: selected data sets. Second row {(c), (d)}: best clustering solutions of Fuzzy ART (system 1). Third row {(e), (f)}: best clustering solutions of VAT + Fuzzy ART (system 2). The best solutions were selected using the vigilance parameter (ρ) associated with the peak average *ARI* reported in Table 3 for one of the 50 orders of presentation. 169

4. Parameter sweep analysis of the vigilance for the systems with (blue) and without (red) VAT reordering as a pre-processing. The mean and standard deviations (shaded areas) of the *ARI* are depicted. 170

5. Performance of Fuzzy ART when using the different VAT re-orderings (see Table 3) as pre-processing. Each colored curve represent a specific permutation order presented to the system VAT + Fuzzy ART. 170

6. Parameter sweep analysis of the vigilance for the systems with (blue) and without (red) VAT reordering as a pre-processing. The mean and standard deviations (shaded areas) of the number of clusters are depicted. 171

7. Mean and standard deviations (shaded areas) of the difference between the number of clusters of the systems Fuzzy ART and VAT + Fuzzy ART. 171

PAPER IV

1. CD diagrams (Demšar, 2006) with respect to random (a, c) and VAT (b, d) input presentations using Nemenyi (a, b) and Bonferroni-Dunn tests (c, d). 188

PAPER V

1. Elementary ART architecture, underlying various ART designs. 199

2. DDVFA architecture. Each global ART's F_2 node is a local fuzzy ART (as portrayed in Figure 1) with shared complement-coded input $\boldsymbol{x} \in \mathbb{R}^{2d}$ and vigilance $\rho = \rho_{UB}^{(1)} \geq \rho_{LB}^{(1)}$. 204

3. 3D surfaces, contours, and cross-section cuts representing the normalized match functions (M_{γ}^{n}) using $\gamma^{*} = 1$ and different values of γ. 207

4. Merge ART module. Each $ART^{(2)}$ is a fuzzy ART with $\rho = \rho_{UB}^{(2)}$. 211

5. Data sets used in the experiments. Solely for visualization purposes, the data sets with more than 3 features (i.e., *Dermatology*, *Iris*, *Wine*, *Seeds*, *Wisconsin*, *WDBC*, *Synthetic Control*, *Glass*, and *Ecoli*) are depicted using principal component analysis projection. The data sets' features and projections are scaled to the range [0, 1]. 215

6. Radar charts of the peak average performances (*AR*) of all three different DDVFA systems, which are grouped by the type of activation/match functions (a)-(f). The results are based on 30 runs per data set using $\gamma^{*} = 1$ and $\gamma = 3$. Typically, VAT pre-ordering yielded the best performance, while DDVFA and DDVFA + Merge ART appear to yield a similar performance, with the exception of the single-linkage-based DDVFA, in which using Merge ART makes a noticeable difference when compared to DDVFA by itself. 220

7. Output partitions of the DDVFA system (a)-(d) before, and (e)-(h) after cascading the Merge ART module for the (a,e) *Spiral*, (b,f) *Wave*, (c,g) *Atom*, and (d,h) *Chainlink* data sets. 221

8. CD diagrams for all three DDVFA systems considering all HAC-based distributed activation/match functions. 225

9. CD diagram for selected distributed HAC-based activation/match functions considering all three DDVFA systems. The CD diagram of the single variant is also representative for centroid, complete, median, and weighted. 226

10. CD diagrams comparing the performance of ART-based clustering methods. 227

11. CD diagrams comparing the compactness of the multi-prototype ART-based architectures. 228

12. CD diagrams comparing the performance of two DDVFA systems to SL-HAC, DBSCAN, k-means, and AP clustering algorithms. 229

13. The behavior of the VAT+DDVFA system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (*AR*), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ). 234

14. The behavior of the DDVFA system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (*AR*), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ). 235

15. The behavior of the DDVFA + Merge ART system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (*AR*), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ). 236

16. The behavior of the (a)-(c) VAT + DDVFA, (d)-(f) DDVFA, and (g)-(i) DDVFA + Merge ART systems for different values of parameter γ while holding the dual vigilance parameters constant. Single linkage HAC-based activation and match functions are used. 237

17. The best and most compact output partitions for the *Target* data set using the (a)-(b) VAT + DDVFA and (c)-(d) DDVFA + Merge ART systems. The ordered pairs correspond to (γ, total number of categories). (a) and (c) correspond to fuzzy ART and are subject to category proliferation, whereas (b) and (d) correspond to DDVFA and represent the same data with fewer categories. 238

18. Heat maps corresponding to the average performance (AR) of (a)-(c) VAT + DDVFA, (d)-(f) DDVFA, and (g)-(i) DDVFA + Merge ART, for the *Target* data set when varying parameter γ. More yellow is better, implying a broader range of good parameter values. Sub-figures (a), (d), and (g) correspond to fuzzy ART building blocks, whereas the other portions of the figure correspond to contributions from this paper. 239

PAPER VI

1. Positions of the SOM neurons $\boldsymbol{w}$ in the U-matrix (u). 260

2. Data sets used in the experiments. The data sets *Iris*, *Wine*, *Seeds*, *WDBC*, *Synthetic Control* and *dim032* are depicted using principal component analysis (PCA) projection. 271

3. SOM IT-vis types for the *Wine* data set using $MinPts$ = 30: (a) CIP-vis, (b) H*-vis, (c) rCIP-vis and (d) rH*-vis. 274

4. Elapsed time to compute (a) CIP, (b) rCIP and U-matrix (adapted from (Araújo et al., 2013a,b)), and the necessary parameters (subsets $\mathcal{H}_i$ and statistics) using (c) the modified k-NN (Gokcay & Principe, 2002) and (d) the standard k-NN (Duda et al., 2000). 274

5. The rH*-vis for the *Wine* data set using the modified k-NN (Gokcay & Principe, 2002) (a, b, c) and standard k-NN (Duda et al., 2000) (d, e, f). Different values for the $MinPts$ were used and performance is seen to be robust to the choice of this parameter. 276

6. *Iris* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16. 278

7. *Seeds* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16. 278

8. *Wine* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16. 279

9. *Synthetic Control* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts = 61$, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 8×8, (e)-(h) 16×16 and (i)-(l) 32×32. 279

10. *dim032* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts = 33$, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 10×10, (e)-(h) 20×20 and (i)-(l) 40×40. 280

11. *Wisconsin Diagnostic Breast Cancer* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts = 40$, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 8×8, (e)-(h) 16×16 and (i)-(l) 32×32. 280

12. Clustering the *R15* data set with $MinPts = 25$. First four rows: gray-level image, stable region, markers, watershed ridges and final partitions for rCIP-vis (a)-(e), rH*-vis (f)-(j), U-matrix (k)-(o) and logarithm of U-matrix (p)-(t). The IT-vis used the modified k-NN (Gokcay & Principe, 2002) to generate $\mathcal{H}$. Last four rows: gray-level image, processed image with Gaussian low pass filter, markers, watershed ridges and final partitions for rCIP-vis (u)-(y), rH*-vis (z)-(ad), U-matrix (ae)-(ai) and logarithm of U-matrix (aj)-(an). The IT-vis used the standard k-NN (Duda et al., 2000) to generate $\mathcal{H}$. 287

12. (Cont.) Clustering the *R15* data set with $MinPts = 25$. First four rows: gray-level image, stable region, markers, watershed ridges and final partitions for rCIP-vis (a)-(e), rH*-vis (f)-(j), U-matrix (k)-(o) and logarithm of U-matrix (p)-(t). The IT-vis used the modified k-NN (Gokcay & Principe, 2002) to generate $\mathcal{H}$. Last four rows: gray-level image, processed image with Gaussian low pass filter, markers, watershed ridges and final partitions for rCIP-vis (u)-(y), rH*-vis (z)-(ad), U-matrix (ae)-(ai) and logarithm of U-matrix (aj)-(an). The IT-vis used the standard k-NN (Duda et al., 2000) to generate $\mathcal{H}$. 288

13. AR index for different values of the minimum number of points ($MinPts$) of the subset $\mathcal{H}$ generated using the modified k-NN (Gokcay & Principe, 2002) (first three rows) and the standard k-NN (Duda et al., 2000) (last three rows) for all data sets. In this sweep analysis, the clustering performed using Otsu's method is represented in red and green lines for the rCIP-vis and rH*-vis respectively. The clustering performed using the stable regions approach is represented in black and light blue lines for the rCIP-vis and rH*-vis, respectively.291

PAPER VII

1. (a)-(t) Synthetic data sets. (u)-(v) Real world data sets. High dimensional data sets are shown using a 2-dimensional t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) projection. 330

2. (a)-(m) Behaviors of iCVIs (blue curves) when correctly partitioning the data set *R15*. (n) The number of clusters is depicted by the step-like red curve. Each discrete time instant (x-axis) corresponds to the presentation of one sample. The dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. 333

3. (a) An under-partition of the data set *R15* by fuzzy ART. (b) Presentation order of the clusters. 336

4. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *R15* during the under-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the under-partition (UP) problem starts: the samples delimited by the gray shaded interval are assigned to an existing cluster, instead of forming a new one. (a)-(m) Behaviors of iCVIs (blue curves). (n) Number of clusters (step-like red curve). 338

5. (a) An overpartition of the data set *unbalance* by fuzzy ART. (b) Presentation order of the clusters. 341

6. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *R15* during the over-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the over-partition (OP) problem starts in the cluster delimited by the gray shaded interval. (a)-(m) Behaviors of iCVIs (blue curves). (n) Number of clusters (step-like red curve). 343

7. (a)-(j) Behaviors of iCVIs (blue curves) when correctly partitioning the data set *MNIST*. (k) The number of clusters is depicted by the step-like red curve. Each discrete time instant (x-axis) corresponds to the presentation of one sample. The dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. 346

8. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *MNIST* during the under-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the under-partition (UP) problem starts: the samples delimited by the gray shaded interval are assigned to an existing cluster, instead of forming a new one. (a)-(j) Behaviors of iCVIs (blue curves). (k) Number of clusters (step-like red curve). 347

9. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *MNIST* during the over-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the over-partition (OP) problem starts in the cluster delimited by the gray shaded interval. (a)-(j) Behaviors of iCVIs (blue curves). (k) Number of clusters (step-like red curve). 348

10. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when correctly partitioning the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental ***CONN*** matrices. 355

11. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when under-partitioning of the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental ***CONN*** matrices. 356

12. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when over-partitioning of the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental ***CONN*** matrices. 357

13. Correlation coefficients and MSEs between the batch and incremental versions of Conn_Index as a function of Fuzzy ARTMAP's module A vigilance parameter. The values shown were averaged across 10 runs for (a) *Aggregation* data set under correct partition experiment, (b) *Lsun* data set under under-partition experiment, and (c) *D31* data set under over-partition experiment. 358

LIST OF TABLES

Table Page

PAPER I

1. Unsupervised ART models notation. 10

2. DDVFA’s activation and match functions. 44

3. Merge ART’s activation and match functions. 45

4. Summary of the code representations used by the unsupervised learning ART models. 52

5. Supervised ART models notation. 53

6. Summary of supervised ART models’ key characteristics. 98

PAPER III

1. Summary of some approaches that mitigate order effects. 160

2. Data sets’ summary. 166

3. Results summary. 168

PAPER IV

1. Experimental results. 187

PAPER V

1. Notation for DDVFA. 204

2. DDVFA’s activation and match functions. 207

3. Merge ART’s activation and match functions. 212

4. Summary of the data sets’ characteristics. 216

5. A statistical comparison of the different HAC activation/match functions’ performances per DDVFA system: Friedman-Iman-Davenport p-values. 223

6. A statistical comparison of the different systems’ performances per HAC activation/match function type: Friedman-Iman-Davenport p-values. 223

7. Experimental results summary: performance in terms of AR (mean ± standard deviation). 224

8. Time complexity and hyper-parameterization of all algorithms used in this work. 231

9. A statistical comparison of $\gamma = 1$ versus $\gamma = 3$: Wilcoxon p-values. 233

PAPER VI

1. Data sets' characteristics. 272

2. Peak performance for CONN linkage, *k*-means, U-matrix and logarithm of U-matrix. 289

3. Peak performance for IT-vis methods. 290

4. Clustering results for large SOM sizes. Peak performance for IT-vis methods without the watershed algorithm. 292

PAPER VII

1. Summary of the data sets' characteristics. 329

2. Summary of the iCVIs' behaviors when correctly, under, and over-partitioning the synthetic data sets used in the experiments. 335

3. The iCVIs' that conveyed the most visually informative behavior across the experiments are indicated by a black dot symbol "•" for each data set. The dash symbol "-" indicates that the iCVI was not computed for the corresponding data set. 351

4. Vigilance parameter (ρ) values used in in this study and the average correlation coefficient (R_{avg}) and mean square error (MSE_{avg}) between the incremental and batch Conn_Index curves. 354

SECTION

1. INTRODUCTION

1.1. ADAPTIVE RESONANCE THEORY

Adaptive resonance theory (ART) (Grossberg, 1976a,b, 1980, 2013) is a learning theory that gave rise to numerous neural network models for unsupervised, reinforcement, and supervised learning. It addresses the stability-plasticity problem using a match-based learning approach, thereby avoiding the catastrophic forgetting problem that afflicts error-based models. A detailed discussion on ART models as well as their useful properties and open problems in the field is provided in the chapter corresponding to Paper I. In particular, elementary ART models follow similar design principles but differ with respect to their category abstraction aspect, which limits the types of clusters that they can detect. Moreover, ART belongs to the class of incremental learning methods, and thus the order of input presentation represents an inherent challenge, especially in online learning mode. Therefore, the chapters corresponding to Papers II through V address the arbitrarily-shaped clusters and/or the order of input presentation problems in fuzzy ART-based models. To accomplish this goal, a family of multi-criteria clustering is introduced in the form of dual vigilance ART models, as well as frameworks that exploit visual assessment of cluster tendency (VAT) to sort data samples prior to presentation or use a Merge ART module as a post-processing step.

1.2. SELF-ORGANIZING MAPS

Self-organizing maps (SOMs) (Kohonen, 1982, 2013) are neural network models that are also extensively used across the machine learning modalities. In fact, it is a particularly useful tool for multivariate data visualization, since each SOM neuron is associated with both a weight vector in the data space and a fixed position in a rigid lattice. Therefore, it seeks to realize a topology-preserving dimensionality reduction mapping. The chapter corresponding to Paper VI incorporates information-theoretic similarity measures (Araújo et al., 2013a,b; Gokcay & Principe, 2002) and single-linkage-based k-nearest neighbors (Gokcay & Principe, 2002) to provide an enhanced image-based visualization of a trained SOM.

1.3. CLUSTER VALIDATION

Cluster validation (Xu & Wunsch II, 2009) is an important subtopic of cluster analysis that deals with the assessment of partitions identified by clustering algorithms, whose hyper-parameter settings often consider insights provided by data visualization methods. The quality of the data partitions are measured by cluster validity indices (CVIs), which can be broadly divided into external and internal. While the former computes the degree of agreement to a reference partition (i.e., it takes into account some external information), the latter only uses the data partition itself in its computations. Recently, incremental CVIs (iCVIs) have been developed to evaluate, in online mode, the partitions detected by streaming clustering algorithms (Ibrahim et al., 2018a,b; Moshtaghi et al., 2018; Moshtaghi et al., 2019). The chapter corresponding to Paper VII develops additional iCVIs and provides a comprehensive comparison of their behavior across several synthetic and real-world benchmark data sets.

1.4. RESEARCH CONTRIBUTIONS

This dissertation contributes an extensive review of ART systems and addresses the three fundamental problems of cluster analysis (Bezdek, 2017; Xu & Wunsch II, 2009): data visualization, clustering algorithms, and cluster validation. In particular, it contributes novel neural-network-based clustering and visualization methods using ART and SOM, respectively. It also presents incremental variants of popular cluster validity indices (CVIs) and a thorough comparison study. The main contributions of this dissertation are listed in detail in the following subsections. In observance of the advisory committee suggestions, there are very minor differences (grammar, stylistic, and/or typographical error corrections) between this dissertation and the papers upon which it is based.

1.4.1. Adaptive Resonance Theory Neural Network Models Review. The first paper of this dissertation (Brito da Silva et al., 2019b) consists of a review of ART systems. It encompasses brief descriptions of ART models used for unsupervised, supervised, and reinforcement learning. It also discusses useful ART properties and current challenges. Therefore, this paper serves a dual purpose in that (1) it allows the reader to become familiar with ART and (2) it provides context to the original contributions described in the following ART-based papers of this dissertation.

1.4.2. Clustering Algorithms.

1.4.2.1. The VAT and fuzzy ART framework (VAT + FA). The work in Paper II (Brito da Silva & Wunsch II, 2018a) contributes a framework that uses the visual assessment of cluster tendency (VAT) sorting property to pre-order the inputs presented to fuzzy ART when training is performed in offline mode. The VAT + FA framework mitigates ordering effects and is recommended for practical applications since experimental results showed both superior performance and model compactness when compared to random input presentation, where a statistical difference was observed.

1.4.2.2. Dual vigilance fuzzy ART (DVFA). The work in Paper III (Brito da Silva et al., 2019a) contributes a simple and effective fuzzy ART-based architecture for retrieving arbitrarily-shaped clusters when these are expected in data by using dual vigilance parameters and VAT pre-processing. The performance of DVFA was assessed through experiments with benchmark data sets in random and VAT-ordered presentations, where it was observed that DVFA was statistically comparable to a much more complex fuzzy ART-based topology clustering approach.

1.4.2.3. Distributed dual vigilance fuzzy ART (DDVFA). The work in Paper IV (Brito da Silva et al., 2020) contributes a modular fuzzy ART-based architecture designed for unsupervised learning. It consists of parallel local ART nodes nested in and controlled by a global ART network. It employs dual vigilance parameters, builds multi-prototype cluster representations, and can discover arbitrarily-shaped clusters. Furthermore, DDVFA uses higher-order activation and match functions distributed according to hierarchical agglomerative clustering algorithms that have the potential to generate more compact DDVFA networks and extend the regions of successful vigilance parameter combinations. Coupled with a compatible Merge ART module, DDVFA outperformed other current state-of-the-art fuzzy ART-based methods in experiments carried out with a collection of 30 publicly available benchmark data sets. Moreover, DDVFA was deemed statistically comparable to non-ART clustering algorithms, while still retaining useful properties of the fuzzy ART incremental learning system.

1.4.2.4. Cluster validity index vigilance test in fuzzy ART (CVIFA). The work in Paper V (Brito da Silva & Wunsch II, 2017b) contributes a simple and robust fuzzy ART-based architecture for offline unsupervised learning. The CVIFA is also a member of the dual vigilance fuzzy ART family of architectures, in which the model is augmented by a second vigilance based on cluster validity indices (CVIs) to incrementally guide the data partitioning process: samples are accepted in a category depending on the relative CVI improvement that would result from such action. Experiments with benchmark data sets and random sample presentation showed that CVIFA outperformed standard fuzzy ART, was more robust to variations of the first (standard) vigilance parameter, and yielded more compact models, therefore notably alleviating order dependency.

1.4.3. Data Visualization: Information-Theoretic Visualization for SOM (IT-vis). The work in Paper VI (Brito da Silva & Wunsch II, 2018b) contributes an image-based visualization method for SOMs. The IT-vis combines Renyi's quadratic cross-entropy (to measure similarity between neighboring neurons) and a single-linkage-based selection of data samples for parameter estimation, specifically for visualization purposes in a trained SOM. This visualization is displayed using the unified distance matrix structure. By visual assessment, the method provided a sharper delineation of cluster boundaries. Additionally, it showed robust and efficient performance when performing clustering tasks.

1.4.4. Cluster Validation: Extensions and Comparative Study on Incremental Cluster Validity Indices (iCVIs). The work in Paper VII (Brito da Silva et al., 2019c) extends the family of iCVIs with seven incremental versions of well-known batch (offline) cluster validity indices (CVIs). This was accomplished by incorporating a previously developed incremental computation of compactness to the following sum-of-squares-based CVIs: Calinski-Harabasz, Pakhira-Bandyopadhyay-Maulik, WB, and Silhouette. Incremental versions of the information-theoretic-based CVIs of Negentropy Increment, Representative Cross Information Potential, and Representative Cross Entropy were made viable by using the classic incremental computation of mean, covariance matrix, and probability estimates. Finally, the Conn_Index graph-based CVI was incrementally approximated via a framework consisting of a modified fuzzy ARTMAP system for multi-prototype representation of clusters and dynamic updates of the prototypes' local-density-based similarity matrix. Moreover, the behaviors of thirteen existing iCVIs (including PS and incremental versions of Xie-Beni, Davies-Bouldin, and generalized Dunn's indices 43 and 53) were analyzed in correct, under- and over-partition experiments to compare their explainability power in data stream applications with meaningful temporal information.

PAPER

I. A SURVEY OF ADAPTIVE RESONANCE THEORY NEURAL NETWORK MODELS FOR ENGINEERING APPLICATIONS

Leonardo Enzo Brito da Silva[1,2], Islam Elnabarawy[1] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Laboratory,

Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

This survey samples from the ever-growing family of adaptive resonance theory (ART) neural network models used to perform the three primary machine learning modalities, namely, unsupervised, supervised and reinforcement learning. It comprises a representative list from classic to contemporary ART models, thereby painting a general picture of the architectures developed by researchers over the past 30 years. The learning dynamics of these ART models are briefly described, and their distinctive characteristics such as code representation, long-term memory, and corresponding geometric interpretation are discussed. Useful engineering properties of ART (speed, configurability, explainability, parallelization and hardware implementation) are examined along with current challenges. Finally, a compilation of online software libraries is provided. It is expected that this overview will be helpful to new and seasoned ART researchers.

Keywords: Adaptive Resonance Theory, Clustering, Classification, Regression, Reinforcement Learning, Survey.

1. INTRODUCTION

Adaptive Resonance Theory (ART) (Grossberg, 1976a,b, 1980, 2013) is a biologically plausible theory of how a brain learns to consciously attend, learn and recognize patterns in a constantly changing environment. The theory states that resonance regulates learning in neural networks with feedback (recurrence). Thus, it is more than a neural network architecture, or even a family of architectures. The book *Introduction to Neural and Cognitive Modeling* (Levine, 2019) presents the broad scientific context of neural models, including ART. From its foundations as a cognitive theory, ART has inspired a developing family of system architectures. Even the first ART model (Carpenter & Grossberg, 1987a) was immediately fielded as a key component of the Boeing parts design retrieval system (Caudell et al., 1994, 1991). Network properties that are the basis for ART's selection in engineering applications include fast, stable, incremental learning with relatively small memory requirements and straightforward algorithms (Wunsch II, 2009). In this context, fast learning refers to the ability of the neurons' weight vectors to converge to their asymptotic values directly with each input sample presentation. These, and other properties, make ART networks attractive to many researchers and practitioners, as they have been used successfully in a variety of science and engineering applications.

ART addresses the problem of *stability vs. plasticity* (Carpenter & Grossberg, 1987a; Grossberg, 1980; Mermillod et al., 2013). Plasticity refers to the ability of a learning algorithm to adapt and learn new patterns. In many learning systems plasticity can lead to instability, a situation in which learning new knowledge leads to the loss or corruption of previously learned knowledge, also known as catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990). Informally, stability in learning is referred to as the retention of useful information. A more precise discussion of stability in this sense is in (Carpenter & Grossberg, 1987a; Grossberg, 1976a,b, 1980; Moore, 1989). Note that this concept is distinct from the treatment of stability in the control theory literature. ART addresses this stability-plasticity dilemma by introducing the ability to learn arbitrary input patterns in a fast and stable self-organizing fashion without suffering from catastrophic forgetting.

Previous studies with similar objectives of surveying the ART neural network literature can be found in (Amorim et al., 2011; Du, 2010; Jain et al., 2000, 2014; Lerner & Guterman, 2008; RamaKrishna et al., 2014). This survey expands on those works, compiling a broad and informative sampling of ART neural network architectures from the ever-growing machine learning literature. Over the past three decades, a myriad of ART systems have been presented and studied, and it is impossible to be completely comprehensive. Thus, this survey captures a representative set of examples of various ART architectures in the unsupervised, supervised and reinforcement learning domains, as well as some models that cross these boundaries and/or combine multiple learning modalities. The overarching goal of this survey is to provide researchers with an accessible coverage of these models, focusing on their motivations, dynamics and interpretations for engineering applications; and a discussion of open problems for consideration. It is not meant as a comparative assessment of these models but rather as a roadmap to assess options.

The remainder of this paper is organized as follows. Section 2 presents a sampling of unsupervised learning (UL) ART models divided into elementary, topological, hierarchical, biclustering and data fusion architectures. Section 3 discusses supervised learning (SL) ART models for both classification and regression. Reinforcement learning (RL) ART models are discussed in Section 4. Sections 5 and 6 discuss some of the useful properties of ART architectures and open problems in this field, respectively. Section 7 provides links to some repositories of ART neural network code, and Section 8 concludes the paper.

2. ART MODELS FOR UNSUPERVISED LEARNING

2.1. ELEMENTARY ARCHITECTURES

At their core, the elementary ART models are predominantly used for unsupervised learning applications. However, they also lay the foundation to build complex ART-based systems capable of performing all three machine learning modalities (Secs. 2, 3, and 4). This section describes the main characteristics of ART family members in terms of their code representation, long-term memory unit, system dynamics (which encompasses activation, match, resonance and learning) and user-defined parameters. For clarity, Table 1 summarizes the common notation used in the following subsections.

An elementary ART neural network model (Figure 1) usually consists of two fully connected layers as well as a system responsible for its decision-making capabilities:

- Feature representation field F_1: this is the input layer. In feedforward mode, the output $\boldsymbol{y}^{(F_1)}$ of this layer, or short-term memory (STM), simply propagates the input samples $\boldsymbol{x} \in \mathbb{R}^d$ to the F_2 layer via the bottom-up long-term memory units (LTMs) $\boldsymbol{\theta}^{bu}$. In feedback mode, the F_1 layer works as a comparator, in which $\boldsymbol{x}$ and the F_2's expectation (in the form of a top-down LTM $\boldsymbol{\theta}^{td}$) are compared and the outcome $\boldsymbol{y}^{(F_1)}$ is sent to the orienting subsystem. Hence, F_1 is also known as the comparison layer.

- Category representation field F_2: this layer yields the network output $\boldsymbol{y}^{(F_2)}$ (STM). It is also known as the recognition or competitive layer. Neurons, prototypes, categories and templates are used interchangeably when referring to the F_2 nodes. The LTM associated with a category j is $\boldsymbol{\theta}_j = \{\boldsymbol{\theta}_j^{bu}, \boldsymbol{\theta}_j^{td}\}$, $j = 1, ..., N$. Note that not all elementary ART models discussed in this survey have independent bottom-up and top-down LTM parts; however, $\boldsymbol{\theta}$ is always used to indicate the LTM (or set of adaptive parameters) of a given category.

Table 1. Unsupervised ART models notation.

Notation	Description
$\boldsymbol{x}$	input sample ($\boldsymbol{x} \in \boldsymbol{X}$)
d	original data dimensionality ($\boldsymbol{x} \in \mathbb{R}^d$)
F_1	feature representation field
F_2	category representation field
N	number of categories
$\boldsymbol{y}^{(F_1)}$	F_1 activity/output (STM)
$\boldsymbol{y}^{(F_2)}$	F_2 activity/output (STM)
c	a category
$\boldsymbol{\theta}$	category parameters (LTM unit)
T	activation function
M	match function
J	chosen category index (via WTA)
ρ	vigilance parameter
VR	vigilance region

- Orienting subsystem: this is a system that regulates both the search and learning mechanisms by inhibiting or allowing categories to resonate.

Note that some ART models represent pre-processing procedures of the input samples by another layer preceding F_1, namely the input field F_0. In this survey, it is assumed that the inputs to an ART network have already gone through the required transformations, and thus this layer is omitted from the discussion.

ART models are competitive, self-organizing, dynamic and modular networks. When a sample $\boldsymbol{x}$ is presented, a winner-takes-all (WTA) competition takes place over its categories at the output layer F_2. Then, the neuron J that optimizes that model's *activation (or choice) function* T across the nodes is chosen, e.g., the neuron that maximizes some similarity measure to the presented sample

$$J = \arg\max_j (T_j). \tag{1}$$

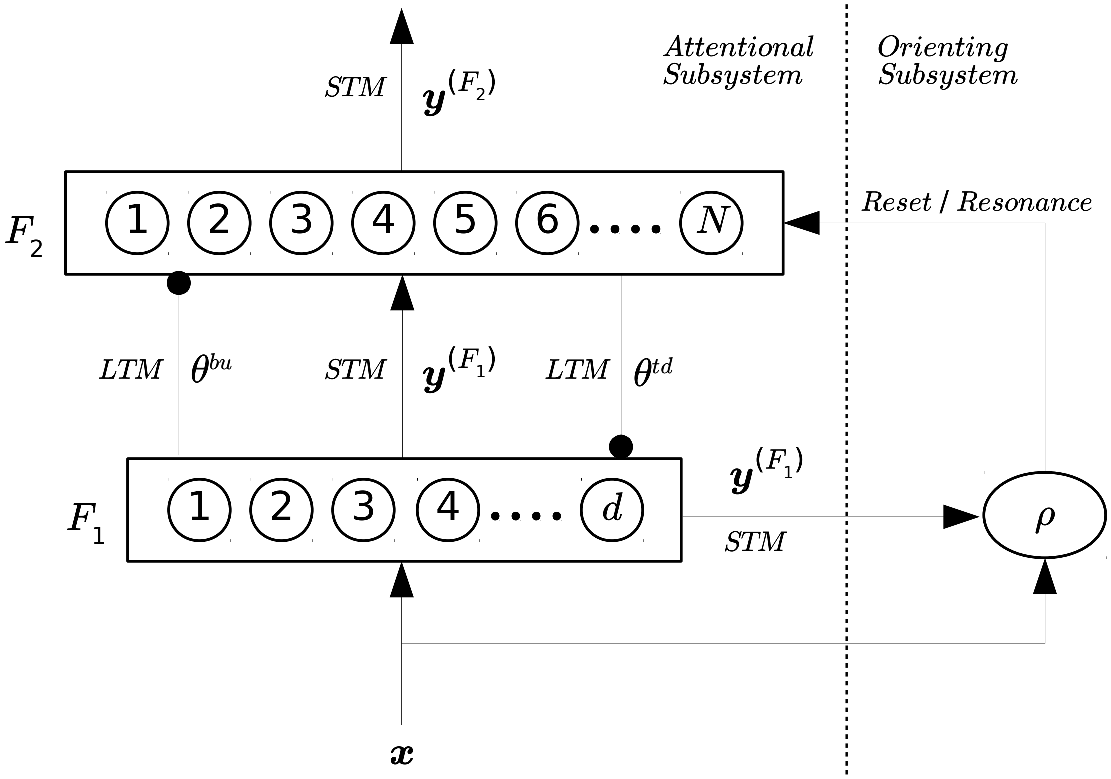


Figure 1. Elementary ART model underlying various WTA designs. The orienting subsystem uses the vigilance threshold to regulate whether ART can go into resonance or if it must reset.

A category represents a hypothesis. Therefore, a hypothesis test cycle, commonly referred to as a vigilance test, is performed by the orienting subsystem to determine the adequacy of the selected category, i.e., the winner category must satisfy a match criterion (or several match criteria). If the confidence on such a hypothesis is larger than the minimum threshold (namely, the vigilance parameter ρ), the neural network enters in a resonance state, and learning (i.e., adaptation of the long-term memory (LTM) units) is allowed. Otherwise, category J is inhibited, the next highest ranked category is selected, and the search resumes. If no category satisfies the required resonance condition(s), then a new one is created to encode the presented input sample. This ability to reject a hypothesis/category via a two-way similarity measure, i.e. *permissive clustering* (Seiffertt & Wunsch II, 2010), makes ART stand out from other methods, such as k-means (MacQueen, 1967).

A vigilance region (VR) for a given network category j can be defined in the data space as

$$VR_j = \{\boldsymbol{x} : M_j(\boldsymbol{x}) \text{ satisfies the resonance constraint}\}, \tag{2}$$

where M_j is the *match function*, which yields the confidence on hypothesis j. In other words, it is the region in the input space containing the set of all points such that the resonance criteria is met. Therefore satisfying (or not) the vigilance test for sample $\boldsymbol{x}$ can be modeled using

$$\mathbb{1}_{VR_j}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in VR_j \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $\mathbb{1}_{\{\cdot\}}(\cdot)$ is the indicator function.

The resonance constraint in Eq. (2) depends on the vigilance parameter ρ, which regulates the granularity of the network as ART maps samples to categories. Particularly, lower vigilance encourages generalization (Vigdor & Lerner, 2007). Selecting the vigilance parameter is a difficult task in clustering problems. Concretely, the problem of choosing the number of clusters is traded for the problem of choosing the vigilance value.

Distinct ART models feature specific LTM units; activation and match functions; vigilance criteria; and learning laws. Nonetheless, Algorithm 1 summarizes the general dynamics of an elementary ART model.

2.1.1. ART 1. The ART 1 neural network (Carpenter & Grossberg, 1987a) and its engineering applications rely on set theoretic operators to cluster binary input samples using a similarity measure based on Hamming distance (Serrano-Gotarredona et al., 1998).

LTM. ART 1 categories are parameterized with bottom-up and top-down adaptive weight vectors $\boldsymbol{\theta} = \{\boldsymbol{w}^{bu}, \boldsymbol{w}^{td}\}$.

Algorithm 1: Elementary ART algorithm.

```
Input  : $\boldsymbol{x}$, $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}\}$ (parameters).
Output : $\boldsymbol{y}^{(F_2)}$.

/* Notation                                                        */
$C$:  set of ART nodes.
$\Lambda$:  subset of highly active nodes ($\Lambda \subseteq C$).
$\boldsymbol{\theta}$:  LTM unit.
$\boldsymbol{\alpha}$:  activation function parameter(s).
$\boldsymbol{\beta}$:  learning function parameter(s).
$\boldsymbol{\gamma}$:  match function parameter(s).
$\boldsymbol{\rho}$:  vigilance parameter(s).
$\boldsymbol{\lambda}$:  initialization parameter(s).
$f_T(\cdot)$:  activation function.
$f_M(\cdot)$:  match function.
$f_L(\cdot)$:  learning function.
$f_V(\cdot)$:  vigilance function (e.g., $f_V = \bigwedge_k \mathbb{1}^k_{VR_J}(\boldsymbol{x})$).
$f_N(\cdot)$:  initialization function.
$k$:  number of resonance constraints.
/* Training                                                        */
1  Present input sample: $\boldsymbol{x} \in \boldsymbol{X}$.
2  Compute activation function(s): $T_j = f_T(\boldsymbol{x}, \boldsymbol{\theta}_j, \boldsymbol{\alpha}),\ \forall j \in C$.
3  Perform WTA competition:
   $J = \underset{j \in \Lambda}{\arg\max}\,(T_j)$.
4  Compute match function(s): $M_J^k = f_M^k(\boldsymbol{x}, \boldsymbol{\theta}_J, \boldsymbol{\gamma}),\ \forall k,\ k \geq 1$.
5  Perform vigilance test(s): $V_J = f_V(\mathbb{1}^1_{VR_J}(\boldsymbol{x}), ..., \mathbb{1}^k_{VR_J}(\boldsymbol{x}))$.
6  if $V_J$ is TRUE then
7      Update category $J$: $\theta_J^{new} = f_L(\boldsymbol{x}, \boldsymbol{\theta}_J^{old}, \boldsymbol{\beta})$.
8  else
9      Deactivate category $J$: $\Lambda \leftarrow \Lambda - \{J\}$.
10     if $\Lambda \neq \{\emptyset\}$ then
11         Go to step 3.
12     else
13         Set $J = |C| + 1$.
14         Create new category: $C \leftarrow C \cup \{J\}$.
15         Initialize new category: $\theta_J^{new} = f_N(\boldsymbol{x}, \boldsymbol{\lambda})$.
       end
   end
16 Set output: $y_j^{(F_2)} = \begin{cases} 1, & \text{if } j = J \\ 0, & \text{otherwise} \end{cases}$.
17 Go to step 1.
```

Activation. When a sample $\boldsymbol{x}$ is presented to ART 1, the activation function of each category j is computed as

$$T_j = \|\boldsymbol{x} \cap \boldsymbol{w}_j^{bu}\|_1 \doteq \langle \boldsymbol{w}_j^{bu}, \boldsymbol{x} \rangle = \sum_{i=1}^{d} x_i w_{ji}^{bu}, \tag{4}$$

where $\boldsymbol{x}$ is a binary input, $\cap$ is a component-wise binary logic AND, $\boldsymbol{w}^{bu}$ is the bottom-up weight vector, $\| \cdot \|_1$ is the L_1 norm, and $\langle \cdot, \cdot \rangle$ is an inner product.

When a node J is selected via the WTA competition, the F$_2$ activity (short-term memory - STM) becomes

$$y_j^{(F_2)} = \begin{cases} 1, & \text{if } j = J \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Moreover, the F$_1$ activity (short-term memory - STM) is defined as

$$\boldsymbol{y}^{(F_1)} = \begin{cases} \boldsymbol{x}, & \text{if } F_2 \text{ is inactive} \\ \boldsymbol{x} \cap \boldsymbol{w}_J^{td}, & \text{otherwise} \end{cases}. \tag{6}$$

Note that the WTA competition always includes one uncommitted node (i.e., a node that has not undergone adaptation, as opposed to committed nodes), which is guaranteed to satisfy the vigilance criterion following Eq. (7).

Match and resonance. The highest activated node J is tested for resonance using

$$M_J = \frac{\|\boldsymbol{y}^{(F_1)}\|_1}{\|\boldsymbol{x}\|_1} = \frac{\|\boldsymbol{x} \cap \boldsymbol{w}_J^{td}\|_1}{\|\boldsymbol{x}\|_1}, \tag{7}$$

where $VR_J = \{\boldsymbol{x} : M_J(\boldsymbol{x}) \geq \rho\}$ and $\rho \in [0, 1]$. The vigilance criterion checks if $\mathbb{1}_{VR_J}(\boldsymbol{x})$ is true, and, in the affirmative case, the category is allowed to learn.

Learning. When the system enters a resonant state, learning ensues as

$$\boldsymbol{w}_J^{td}(new) = \boldsymbol{x} \cap \boldsymbol{w}_J^{td}(old), \tag{8}$$

$$\boldsymbol{w}_J^{bu}(new) = \frac{L}{L-1+\|\boldsymbol{w}_J^{td}(new)\|_1}\boldsymbol{w}_J^{td}(new), \tag{9}$$

where $L > 1$ is a user-defined parameter (larger values of L bias the selection of uncommitted nodes over committed ones). Note that the bottom-up weight vectors are normalized versions of their top-down counterparts. If an uncommitted node is selected to learn sample $\boldsymbol{x}$, then another one is created and initialized as

$$\boldsymbol{w}^{td} = \vec{\mathbf{1}}, \tag{10}$$

$$\boldsymbol{w}^{bu} = \frac{L}{L-1+d}\boldsymbol{w}^{td}. \tag{11}$$

ART 1 features the following useful properties discussed in (Serrano-Gotarredona et al., 1998): "*vigilance or variable coarseness, self-scaling, self-stabilization in a small number of iterations, online learning, capturing rare events, direct access to familiar input patterns, direct access to subset and superset patterns, biasing the network to form new categories.*" ART 1 properties are also discussed in (Georgiopoulos et al., 1990, 1991, 1992; Heileman et al., 1994).

2.1.2. ART 2. ART 2 (Carpenter & Grossberg, 1987b) and 2-A (Carpenter et al., 1991b) represent the initial effort toward extending ART 1 (Sec. 2.1.1) applications to real-valued data. ART 2, which is based on the Euclidean (L_2) metric, was largely supplanted by fuzzy ART (Sec. 2.2), which is based on the city-block (L_1) metric. The L_1 norm renders fuzzy ART more computationally tractable and biologically plausible. This system has become one of the most widely used and referenced foundational building blocks for ART networks. This was followed by other architectures such as the ART 3 (Carpenter & Grossberg, 1990) hierarchical architecture, exact ART (Raijmakers & Molenaar, 1997) (which is a complete ART network based on ART 2) and correlation-based ART (Yavaş & Alpaslan, 2009) along with its hierarchical variant (Yavaş & Alpaslan, 2012) which use correlation analysis methods for category matching. Particularly, the ART 2-A (Carpenter et al., 1991b) architecture was developed following ART 2 with the same properties and a much faster speed.

LTM. The internal category representation in ART 2-A consists of an adaptive scaled weight vector $\boldsymbol{\theta} = \{\boldsymbol{w}\}$.

Activation. The activation function of each category j in response to a normalized input sample $\boldsymbol{x}$ is computed as

$$T_j = \begin{cases} \alpha \sum_i x_i, & \text{if } j \text{ is uncommitted} \\ \boldsymbol{x}^T \boldsymbol{w}_j, & \text{if } j \text{ is committed} \end{cases}, \tag{12}$$

where $\alpha \leq \frac{1}{\sqrt{d}}$ is the choice parameter.

Match and resonance. The category with the highest activation value is chosen via WTA selection. Its match function is computed as

$$M_J = T_J, \tag{13}$$

and the vigilance test is performed to determine whether resonance occurs using the following: $M_J \geq \rho$, where $0 \leq \rho \leq 1$ is the vigilance threshold.

If the winning category passes the vigilance test, resonance occurs, and the category is allowed to learn this input pattern. If the category fails the vigilance test, a reset signal is triggered for this category, and the category with the next highest activation is selected for the same process.

Learning. When resonance occurs, the weights of the winning category are updated as

$$\boldsymbol{w}_J(new) = \begin{cases} \boldsymbol{x}, & \text{if } J \text{ is uncommitted} \\ \beta\boldsymbol{x} + (1-\beta)\boldsymbol{w}_J(old), & \text{if } J \text{ is committed} \end{cases}, \tag{14}$$

where $0 < \beta \leq 1$ is the learning rate.

2.1.3. Fuzzy ART. Fuzzy ART (Carpenter et al., 1991c) is arguably the most widely used ART model. It extends the capabilities of ART 1 (Sec. 2.1.1) to process real-valued data by incorporating fuzzy set theoretic operators (Zadeh, 1965). Typically, samples are pre-processed by applying complement coding (Carpenter et al., 1992, 1991a). This transformation doubles the original input dimension while imposing a constant norm ($\boldsymbol{x} \leftarrow [\boldsymbol{x}, \vec{\boldsymbol{1}} - \boldsymbol{x}]$):

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{2d} x_i = \sum_{i=1}^{d} x_i + \sum_{i=1}^{d}(1 - x_i) = d. \tag{15}$$

This process encodes the degree of presence and absence of each data feature. The augmented input vector prevents a category proliferation type due to weight erosion (Carpenter, 1997; Moore, 1989).

LTM. Each category LTM unit is a weight vector $\boldsymbol{\theta} = \{\boldsymbol{w}\}$. If complement coding is employed, then $\boldsymbol{w} = [\boldsymbol{u}, \boldsymbol{v}^c]$, and the geometric interpretation of a category is a hyperrectangle (or hyperbox), in the data space, with the lower left corner $\boldsymbol{u}$ and upper right corner $\boldsymbol{v}^c$ representing feature ranges (minimum and maximum data statistics).

Activation. The activation function of a category j is defined as (Weber law)

$$T_j = \frac{\|\boldsymbol{x} \wedge \boldsymbol{w}_j\|_1}{\alpha + \|\boldsymbol{w}_j\|_1}, \tag{16}$$

where $\wedge$ is a component-wise fuzzy AND/intersection (minimum operation), and $\alpha > 0$ is the choice parameter. The latter is related to the system's complexity (it can be seen as a regularization parameter that penalizes large weights), and its role has been investigated in (Georgiopoulos et al., 1996). The activation function measures the fuzzy subsethood degree (Kosko, 1986) of $\boldsymbol{w}_j$ in $\boldsymbol{x}$ (Carpenter et al., 1991c) and is biased towards smaller categories.

Note that the F$_1$ activity is defined as

$$\boldsymbol{y}^{(F_1)} = \begin{cases} \boldsymbol{x}, & \text{if } F_2 \text{ is inactive} \\ \boldsymbol{x} \wedge \boldsymbol{w}_J, & \text{otherwise} \end{cases}, \tag{17}$$

and when the winner node J is selected, the F$_2$ activity follows Eq. (5).

Match and resonance. The hypothesis testing cycle is conducted using the following match function

$$M_J = \frac{\|\boldsymbol{y}^{(F_1)}\|_1}{\|\boldsymbol{x}\|_1} = \frac{\|\boldsymbol{x} \wedge \boldsymbol{w}_J\|_1}{\|\boldsymbol{x}\|_1}, \tag{18}$$

where $VR_J = \{\boldsymbol{x} : M_J(\boldsymbol{x}) \geq \rho\}$ and $\rho \in [0, 1]$ is the vigilance parameter. Fuzzy ART VRs are hyperoctagons as discussed in (Anagnostopoulos & Georgiopoulos, 2002; Meng et al., 2016; Verzi et al., 2006). The vigilance criterion checks if $\mathbb{1}_{VR_J}(\boldsymbol{x})$ is true, and, in the affirmative case, the category is allowed to learn. An uncommitted category will always satisfy the match criterion. Like the activation function (Eq. (16)), the match function is also a measure of fuzzy subsethood degree (Kosko, 1986); particularly of $\boldsymbol{x}$ in $\boldsymbol{w}_J$ (Carpenter et al., 1991c). The hypothesis testing cycle ensures that if learning takes place, the updated category will not exceed the maximum allowed size. Specifically, category j's size is measured as

$$R_j = \|\boldsymbol{v}_j - \boldsymbol{u}_j\|_1 = \sum_{i=1}^{d} \left[(1 - w_{j,d+i}) - w_{j,i}\right] = d - \|\boldsymbol{w}_j\|_1, \tag{19}$$

where, considering the complement coded inputs, $-d \leq R_j \leq d$ (for an uncommitted category: $R_j = -d$). Particularly, the match function measures the size of the category if it is allowed to learn the presented sample. Thus, the vigilance criterion imposes an upper bound to the category size defined by the vigilance parameter (ρ)

$$R_J \oplus \boldsymbol{x} = d - \|\boldsymbol{x} \wedge \boldsymbol{w}_j\|_1 \leq d(1 - \rho), \tag{20}$$

where $R_J \oplus \boldsymbol{x}$ represents the smallest hyperrectangle capable of enclosing both R_J and the presented sample $\boldsymbol{x}$.

Learning. If the vigilance test fails, then the winner category is inhibited, and the search continues until another one is found or created. When the vigilance criterion is met by category J, it adapts using

$$\boldsymbol{w}_J(new) = (1 - \beta)\boldsymbol{w}_J(old) + \beta(\boldsymbol{x} \wedge \boldsymbol{w}_J(old)), \tag{21}$$

where $\beta \in (0, 1]$ is the learning parameter (setting $\beta = 1$ is known as *fast learning* mode). If an uncommitted node is recruited to learn sample $\boldsymbol{x}$, then another one is created and initialized as $\boldsymbol{w} = \vec{\mathbf{1}}$. According to Eq. (21), the norm of a weight vector is monotonically non-increasing during learning since a category's hyperrectangle can only expand (Vigdor & Lerner, 2007). Fuzzy ART learning properties are discussed in (Huang et al., 1994, 1995).

2.1.4. Fuzzy Min-Max. The fuzzy min-max neural network (Simpson, 1993) is an unsupervised learning network that uses fuzzy set theory to build clusters using a hyperbox representation discovered via the fuzzy min-max learning algorithm. Each category in fuzzy min-max is represented explicitly as a hyperbox, with the minimum and maximum points of the hyperbox as well as a value for the membership function that measures the degree to which each input pattern falls within this category. The category hyperboxes are adjusted to fit each input sample using a contraction and expansion algorithm that expands the hyperbox of the winning category to fit the input sample and then contracts any other hyperboxes that are found to overlap with the new hyperbox boundaries.

2.1.5. Distributed ART. The distributed ART (Carpenter, 1996a,b, 1997) features distributed code representation for activation, match and learning processes to improve noise robustness and memory compression in a system that features fast and stable learning. Particularly, in WTA mode, distributed ART reduces in functionality to fuzzy ART (Sec. 2.2).

LTM. The distributed ART LTM units consist of bottom-up ($\boldsymbol{\tau}^{bu}$) and top-down ($\boldsymbol{\tau}^{td}$) adaptive thresholds ($\boldsymbol{\theta} = \{\boldsymbol{\tau}^{bu}, \boldsymbol{\tau}^{td}\}$), which are initialized as small random values and $\vec{\mathbf{0}}$, respectively. When employing complement coding, the geometric interpretation of a category j is a family of hyperrectangles nested by the activation levels $y_j^{(F_2)} \in [0, 1]$. The edges of hyperrectangle $R_j(y_j^{(F_2)})$ are defined, for each input dimension i, as the bounded interval $\left[[y_j^{(F_2)} - \tau_{j,i}^{bu}]^+, 1 - [y_j^{(F_2)} - \tau_{j,d+i}^{bu}]^+\right]$ — where $[\xi]^+ = \max(0, \xi)$ is a rectifier operator. Note that the R_j size decreases as $y_j^{(F_2)}$ increases. Particularly, setting $y_j^{(F_2)} = 1$ yields the smallest hyperrectangle $R(1)$, and the substitution $\boldsymbol{w}_j = (\vec{\mathbf{1}} - \boldsymbol{\tau}^{bu})$ corresponds to fuzzy ART's LTM.

Activation. The activation function can be defined as a choice-by-difference (Carpenter & Gjaja, 1994) ($T_j \in [0, d]$) variant

$$T_j = \|[\boldsymbol{x} \wedge (\vec{\mathbf{1}} - \boldsymbol{\tau}_j^{bu}) - \boldsymbol{\Delta}_j]^+\|_1 + (1 - \alpha)\|[\boldsymbol{\tau}_j^{bu} - \boldsymbol{\delta}_j]^+\|_1 \,, \; 0 < \alpha < 1, \tag{22}$$

or a Weber law (Carpenter & Grossberg, 1987a) ($T_j \in [0, 1]$) variant

$$T_j = \frac{\|[\boldsymbol{x} \wedge (\vec{\mathbf{1}} - \boldsymbol{\tau}_j^{bu}) - \boldsymbol{\Delta}_j]^+\|_1}{\alpha + d - \|[\boldsymbol{\tau}_j^{bu} - \boldsymbol{\delta}_j]^+\|_1} \,, \; \alpha > 0, \tag{23}$$

where $[\boldsymbol{\xi}]^+$ is a component-wise rectifier operator (i.e., $[\xi_k]^+ = \max(0, \xi_k)$ for each component k of vector $\boldsymbol{\xi}$), and $\boldsymbol{\Delta}$ and $\boldsymbol{\delta}$ are the medium-term memory (MTM) depletion parameters. After the nodes' activations are computed, the F_2 activity can be obtained by employing the increased-gradient content-addressable-memory (IG CAM) rule:

$$y_j^{(F_2)} = \begin{cases} \dfrac{(T_j)^p}{\sum\limits_{\lambda \in \Lambda} (T_\lambda)^p}, & \text{if } j \in \Lambda \\ 0, & \text{otherwise} \end{cases}, \tag{24}$$

such that $\|\boldsymbol{y}^{(F_2)}\|_1 = 1$ and $p > 0$. The subset Λ consists of the nodes such that $T_J \geq T_j$ for $J \in \Lambda$ and $j \notin \Lambda$. Some examples are the Q-max rule (see Sec. 3.1.10) or greater than average activations (i.e., $\Lambda = \{j : T_j \geq T_{avg}\}$, $T_{avg} = 1/N \sum_{j=1}^{N} T_j$). Note that the power law $f(\zeta) = \zeta^p$ converges to WTA when $p \longrightarrow +\infty$.

Match and resonance. The distributed ART's match function is defined as

$$M = \frac{\|\boldsymbol{y}^{(F_1)}\|_1}{\|\boldsymbol{x}\|_1}, \tag{25}$$

where the F_1 activity is given by

$$\boldsymbol{y}^{(F_1)} = \boldsymbol{x} \wedge \boldsymbol{\sigma}, \tag{26}$$

and

$$\sigma_i = \sum_{j=1}^{N} [y_j^{(F_2)} - \tau_{ji}^{td}]^+ \, , \; \sigma_i \in [0, 1]. \tag{27}$$

Resonance occurs if $\mathbb{1}_{VR}(\boldsymbol{x}) = 1$, where $VR = \{\boldsymbol{x} : M(\boldsymbol{x}) \geq \rho\}$ and $\rho \in [0, 1]$. Otherwise, the MTM depletion parameters are updated as

$$\Delta_{ji}(new) = \Delta_{ji}(old) \vee (x_i \wedge [y_j - \tau_{ji}^{bu}]^+), \tag{28}$$

$$\delta_{ji}(new) = \delta_{ji}(old) \vee (y_j \wedge \tau_{ji}^{bu}), \tag{29}$$

and the distributed dynamics continue by recomputing Eqs. (24) through (25). Note that the depletion parameters $\boldsymbol{\Delta}$ and $\boldsymbol{\delta}$ are (re)set to $\vec{\mathbf{0}}$ at the beginning of every input sample presentation.

Learning. When the system enters a resonant state, distributed learning takes place according to the nodes' activation levels. Specifically, the top-down adaptive thresholds are updated using the distributed outstar learning law (Carpenter, 1994):

$$\tau^{td}_{ji}(new) = \tau^{td}_{ji}(old) + \beta \frac{[\sigma_i - x_i]^+}{\sigma_i} \left[y_j^{(F_2)} - \tau^{td}_{ji}(old) \right]^+, \tag{30}$$

whereas the bottom-up adaptive thresholds are updated using the distributed instar learning law (Carpenter, 1997):

$$\tau^{bu}_{ji}(new) = \tau^{bu}_{ji}(old) + \beta \left[y_j^{(F_2)} - \tau^{bu}_{ji}(old) - x_i \right]^+, \tag{31}$$

where $\beta \in [0, 1]$ is the learning rate. The adaptive thresholds' components, $\in [0, 1]$, start near zero and monotonically increase during the learning process. After learning takes place, the depletion parameters $\boldsymbol{\Delta}$ and $\boldsymbol{\delta}$ are both reset to their initial values ($\vec{\mathbf{0}}$). In WTA mode, the distributed instar and outstar learning laws become the instar (Grossberg, 1972) and outstar (Grossberg, 1968, 1969) laws, respectively, and thus distributed ART reduces to fuzzy ART (Sec. 2.2).

2.1.6. Gaussian ART. Gaussian ART (Williamson, 1996) was developed to reduce category proliferation in noisy environments and to provide a more efficient category LTM unit.

LTM. Each category j is a Gaussian distribution composed by mean $\boldsymbol{\mu}_j \in \mathbb{R}^d$, standard deviation $\boldsymbol{\sigma}_j \in \mathbb{R}^d$ and instance counting n_j (i.e., the number of samples encoded by category j used to compute its a priori probability). Therefore, a category is geometrically interpreted as a hyperellipse in the data space.

Activation. Gaussian ART is rooted in Bayes' decision theory, and as such its activation function is defined as:

$$T_j = \hat{p}(c_j|\boldsymbol{x}) = \frac{\hat{p}(\boldsymbol{x}|c_j)\hat{p}(c_j)}{\hat{p}(\boldsymbol{x})}, \tag{32}$$

where the likelihood is estimated as

$$\hat{p}(\boldsymbol{x}|c_j) = \frac{exp\left[-\frac{1}{2}\left(\boldsymbol{\mu}_j - \boldsymbol{x}\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{\mu}_j - \boldsymbol{x}\right)\right]}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_j)}}, \tag{33}$$

and the prior as

$$\hat{p}(c_j) = \frac{n_j}{\sum_{i=1}^{N} n_i}. \tag{34}$$

Note that the evidence $\hat{p}(\boldsymbol{x})$ is neglected in the computations (since it is equal for all categories c_j), and feature independence is assumed, i.e., $\boldsymbol{\Sigma}_j$ is a diagonal matrix ($\boldsymbol{\Sigma}_j = diag(\sigma^2_{j,1}, ..., \sigma^2_{j,d})$). Therefore, since it assumes uncorrelated features, it cannot capture covarying data. A category J is then chosen following the maximum a posteriori (MAP) criterion:

$$J = \arg\max_j(T_j) = \arg\max_j\left[\hat{p}(c_j|\boldsymbol{x})\right]. \tag{35}$$

Match and resonance. The match function is defined as a normalized version of $\hat{p}(\boldsymbol{x}|c_j)$:

$$M_J = exp\left[-\frac{1}{2}\left(\boldsymbol{\mu}_J - \boldsymbol{x}\right)^T \boldsymbol{\Sigma}_J^{-1}\left(\boldsymbol{\mu}_J - \boldsymbol{x}\right)\right], \tag{36}$$

which is then compared to the vigilance parameter threshold $\rho \in (0, 1]$. Note that in the original Gaussian ART paper (Williamson, 1996), a log discriminant is used to reduce the computational burden in both the activation (Eq. (32)) and match (Eq. (36)) functions.

Learning. When the vigilance criterion is met, learning ensues for the resonant category J as

$$n_J(new) = n_J(old) + 1, \tag{37}$$

$$\hat{\boldsymbol{\mu}}_J(new) = \left(1 - \frac{1}{n_J(new)}\right)\hat{\boldsymbol{\mu}}_J(old) + \frac{1}{n_J(new)}\boldsymbol{x}, \tag{38}$$

$$\sigma^2_{J,i}(new) = \left(1 - \frac{1}{n_J(new)}\right)\sigma^2_{J,i}(old) + \frac{1}{n_J(new)}\left(\mu_{J,i}(new) - x_i\right)^2. \tag{39}$$

If a new category is created, then it is initialized with $n_{N+1} = 1$, $\boldsymbol{\mu}_{N+1} = \boldsymbol{x}$, and $\boldsymbol{\Sigma}_{N+1} = \sigma_{init}^2 \boldsymbol{I}$ (isotropic). The initial standard deviation σ_{init} in Gaussian ART directly affects the number of categories created.

2.1.7. Hypersphere ART. The hypersphere ART (Anagnostopoulos & Georgiopoulos, 2000) architecture was designed as a successor for fuzzy ART (Section 2.2) that inherits its advantageous qualities while utilizing fewer categories and having a more efficient internal knowledge representation.

LTM. Each category is represented as $\boldsymbol{\theta} = \{R, \boldsymbol{m}\}$, where $\boldsymbol{m}_j \in \mathbb{R}^d$ and $R_j \in \mathbb{R}$ are the centroid and radius, respectively. Since it does not require complement coding of input samples, it uses $d + 1$ memory per category, which is a smaller memory requirement than fuzzy ART, which uses $2d$ memory to represent the hyperrectangular categories. Naturally, categories are hyperspheres in the data space.

Activation. The activation function T_j for each F_2 category j is calculated as:

$$T_j = \begin{cases} \dfrac{\bar{R} - \max(R_j, ||\boldsymbol{x} - \boldsymbol{m}_j||_2)}{\bar{R} - R_j + \alpha}, & \text{if } j \text{ is committed} \\ \dfrac{\bar{R}}{2\bar{R} + \alpha}, & \text{if } j \text{ is uncommitted} \end{cases}, \tag{40}$$

where $|| \cdot ||_2$ is the L_2 norm, $\alpha \in (0, +\infty)$ is the choice parameter and $\bar{R} \in [R_{\max}, +\infty)$ is the radial extend parameter which controls the maximum possible category size achieved during training. The lower-bound $R_{\max}$ is defined as:

$$R_{\max} = \frac{1}{2} \max_{p,q} ||\boldsymbol{x}_p - \boldsymbol{x}_q||_2. \tag{41}$$

Match and resonance. The winning category J is selected using WTA competition, and the match function is computed as

$$M_J = \begin{cases} 1 - \dfrac{\max(R_J, ||\boldsymbol{x} - \boldsymbol{m}_J||_2)}{\bar{R}}, & \text{if } J \text{ is committed} \\ 1, & \text{if } J \text{ is uncommitted} \end{cases}, \tag{42}$$

where the vigilance criterion is $M_J \geq \rho$ ($\rho \in [0, 1]$ is the vigilance parameter).

Learning. If the winning category satisfies the vigilance test, then resonance occurs, and the radius R_J and centroid $\boldsymbol{m}_J$ of the winning node are updated as follows:

$$R_J^{\text{new}} = R_J^{\text{old}} + \frac{\beta}{2}\left[\max\left(R_J^{\text{old}}, ||\boldsymbol{x} - \boldsymbol{m}_J^{\text{old}}||_2\right) - R_J^{\text{old}}\right], \tag{43}$$

$$\boldsymbol{m}_J^{\text{new}} = \boldsymbol{m}_J^{\text{old}} + \frac{\beta}{2}\left(\boldsymbol{x} - \boldsymbol{m}_J^{\text{old}}\right)\left[1 - \frac{\min\left(R_J^{\text{old}}, ||\boldsymbol{x} - \boldsymbol{m}_J^{\text{old}}||_2\right)}{||\boldsymbol{x} - \boldsymbol{m}_J^{\text{old}}||_2}\right], \tag{44}$$

where $\beta \in (0, 1]$ is the learning rate parameter.

If the winning category fails the vigilance test, it is reset, and the process is repeated. Eventually, either a category succeeds or a new one is created with its radius and centroid initialized as $R_{N+1} = 0$ and $\boldsymbol{m}_{N+1} = \boldsymbol{x}$, respectively.

2.1.8. Ellipsoid ART. Ellipsoid ART (Anagnostopoulos & Georgiopoulos, 2001a,b) is a generalization of hypersphere ART that uses hyperellipses instead of hyperspheres to represent the categories. These require $2d + 1$ memory and are subjected to two distinct constraints during training: (1) maintain a constant ratio between the lengths of their major and minor axes, and (2) maintain a fixed direction of their major axis once it is set. These restrictions, however, can pose some limitations to the categories discovered by ellipsoid ART depending on the order in which the input samples are presented.

LTM. A category j in ellipsoid ART is described by its parameters $\boldsymbol{\theta}_j = \{\boldsymbol{m}_j, \boldsymbol{d}_j, R_j\}$, where $\boldsymbol{m}_j$ is the centroid of the category's hyperellipse, $\boldsymbol{d}_j$ is the direction of the category's major axis and R_j is the category's radius (or half the length of its major axis).

Activation. The distance between an input sample and a category j is calculated using Eq. (45), where $||\cdot||_2$ is the L_2 norm, and $\mu \in (0, 1]$ is a user-specified parameter that defines the ratio between a category's major and minor axes. The activation function T_j for each category j is then calculated using Eq. (46), where $\alpha \in (0, +\infty)$ is the choice parameter, $\bar{R} \geq \frac{1}{\mu} \max\limits_{p,q} ||\boldsymbol{x}_p - \boldsymbol{x}_q||_2$ and $\omega \geq 1$ are user-specified parameters.

Match and resonance. The match function of the winning category J selected via WTA is given by Eq. (47). Resonance occurs if $M_J \geq \rho$, where $\rho \in [0, 1]$ is the vigilance parameter.

$$dis(\boldsymbol{x}, \boldsymbol{m}_j) = \begin{cases} \dfrac{1}{\mu}\sqrt{||\boldsymbol{x} - \boldsymbol{m}_j||_2^2 - \left(1 - \mu^2\right)\left[\boldsymbol{d}_j^T\left(\boldsymbol{x} - \boldsymbol{m}_j\right)\right]^2} & \text{if } \boldsymbol{d}_j \neq \vec{\boldsymbol{0}} \\ ||\boldsymbol{x} - \boldsymbol{m}_j||_2 & \text{if } \boldsymbol{d}_j = \vec{\boldsymbol{0}} \end{cases}, \tag{45}$$

$$T_j = \begin{cases} \dfrac{\bar{R} - R_j - \max\left\{R_j, dis(\boldsymbol{x}, \boldsymbol{m}_j)\right\}}{\bar{R} - 2R_j + \alpha}, & \text{if } j \text{ is committed} \\ \dfrac{\bar{R}}{2\bar{R}\omega + \alpha}, & \text{if } j \text{ is uncommitted} \end{cases}, \tag{46}$$

$$M_J = \begin{cases} 1 - \dfrac{R_J + \max\left\{R_J, dis(\boldsymbol{x}, \boldsymbol{m}_J)\right\}}{\bar{R}}, & \text{if } J \text{ is committed} \\ 1, & \text{if } J \text{ is uncommitted} \end{cases}. \tag{47}$$

Learning. If the winning category J satisfies the vigilance test, then it is updated as follows:

$$R_J^{\text{new}} = R_J^{\text{old}} + \frac{\beta}{2}\left[\max\left\{R_J^{\text{old}}, dis(\boldsymbol{x}, \boldsymbol{m}_J^{\text{old}})\right\} - R_J^{\text{old}}\right], \tag{48}$$

$$\boldsymbol{m}_J^{\text{new}} = \boldsymbol{m}_J^{\text{old}} + \frac{\beta}{2}\left(\boldsymbol{x} - \boldsymbol{m}_J^{\text{old}}\right)\left[1 - \frac{\min\left\{R_J^{\text{old}}, dis(\boldsymbol{x}, \boldsymbol{m}_J^{\text{old}})\right\}}{dis(\boldsymbol{x}, \boldsymbol{m}_J^{\text{old}})}\right], \tag{49}$$

$$\boldsymbol{d}_j = \frac{\boldsymbol{x}_{(2)} - \boldsymbol{m}_J}{||\boldsymbol{x}_{(2)} - \boldsymbol{m}_J||_2}, \tag{50}$$

where $\beta \in (0, 1]$ is the learning rate, and $\boldsymbol{x}_{(2)}$ represents the second input sample to be encoded by this category. When a new category is created, its major axis direction $\boldsymbol{d}_J$ is initially set to the zero vector $\vec{\boldsymbol{0}}$, and then Eq. (50) is used to update it when the second pattern is committed to the category. The hyperellipse's major axis direction stays fixed after that.

If the winning category fails the vigilance check, then it is inhibited, and the entire process is repeated until a winner category satisfies the resonance criterion. If no existing category succeeds, then a new category is created with its LTM unit initialized with $R_{N+1} = 0$, $\boldsymbol{m}_{N+1} = \boldsymbol{x}$, and $\boldsymbol{d}_{N+1} = \vec{\boldsymbol{0}}$.

2.1.9. Quadratic neuron ART. The quadratic neuron ART model (Su & Liu, 2002, 2005) was developed in the context of a multi-prototype-based clustering framework that integrates dynamic prototype generation and hierarchical agglomerative clustering to retrieve arbitrarily shaped data structures.

LTM. A category j is a quadratic neuron (DeClaris & Su, 1991, 1992; Su et al., 1997; Su & Liu, 2001) parameterized by $\boldsymbol{\theta}_j = \{s_j, \boldsymbol{W}_j, \boldsymbol{b}_j\}$, where s_j, $\boldsymbol{W}_j = [w_{k,i}^{(j)}]_{d\times d}$, and $\boldsymbol{b}_j$ are the adaptable LTMs. Particularly, these neurons are hyperellipsoid structures in the multidimensional data space.

Activation. The activation of a quadratic neuron j is given by

$$T_j = exp\left(-s_j^2 \|\boldsymbol{z}_j - \boldsymbol{b}_j\|_2^2\right), \tag{51}$$

where $\boldsymbol{z}_j$ is a linear transformation of the input $\boldsymbol{x}$

$$\boldsymbol{z}_j = \boldsymbol{W}_j \boldsymbol{x}. \tag{52}$$

Match and resonance. After the winning node J is selected using WTA competition, the system will enter a resonant state if node J's response is larger than or equal to the vigilance parameter ρ, i.e., if $M_J \geq \rho$, where the match function is equal to the activation function (Eq. (51)).

Learning. If the vigilance criterion is satisfied for node J, then its parameters $\boldsymbol{p} \in \{s_j, \boldsymbol{W}_j, \boldsymbol{b}_j\}$ are adapted using gradient ascent

$$\boldsymbol{p}(new) = \boldsymbol{p}(old) + \eta \frac{\partial T_J}{\partial \boldsymbol{p}(old)}, \tag{53}$$

where η is the learning rate. Specifically,

$$b_{J,i}(new) = b_{J,i}(old) + \eta_b \left[2 s_J^2 T_J \left(z_{J,i} - b_{J,i}\right)\right], \tag{54}$$

$$w_{k,i}^{(J)}(new) = w_{k,i}^{(J)}(old) + \eta_w \left[-2 s_J^2 T_J \left(z_{J,k} - b_{J,k}\right) x_i\right], \tag{55}$$

$$s_J(new) = s_J(old) + \eta_s \left(-2 s_J T_J \|\boldsymbol{z}_J - \boldsymbol{b}_J\|_2^2\right), \tag{56}$$

where η_b, η_w and η_s are the learning rates. Otherwise, a new category is created and initialized with $\boldsymbol{b}_{N+1} = \boldsymbol{x}$, $\boldsymbol{W}_{N+1} = \boldsymbol{I}_{d \times d}$, and $s_{N+1} = s_{init}$, where $s_{init} \in \mathbb{R}$ is a user-defined parameter.

2.1.10. Bayesian ART. *LTM.* Bayesian ART (Vigdor & Lerner, 2007) is another architecture using multidimensional Gaussian distributions to parameterize the categories: $\boldsymbol{\theta} = \{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), p\}$, where $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and p are the mean, covariance matrix and prior probability, respectively. The latter parameter is computed using the number of samples n learned by a category.

Activation. Like Gaussian ART (Sec. 2.1.6), Bayesian ART also integrates Bayes decision theory in its framework. Thus, its activation function is given by the posterior probability of category j:

$$T_j = \hat{p}(c_j|\boldsymbol{x}) = \frac{\hat{p}(\boldsymbol{x}|c_j)\hat{p}(c_j)}{\sum\limits_{l=1}^{N} \hat{p}(\boldsymbol{x}|c_l)\hat{p}(c_l)}, \tag{57}$$

where $\hat{p}(\boldsymbol{x}|c_j)$ is the same as Eq. (33) but uses a full covariance matrix (instead of diagonal), and $\hat{p}(c_j)$ is the estimated prior probability of category j as in Eq. (34).

Match and resonance. After the WTA competition is performed and the winner category J is selected using the maximum a posteriori probability (MAP) criterion (Eq. (35)), the match function is computed as

$$M_J = \det(\boldsymbol{\Sigma}_J), \tag{58}$$

such that the vigilance criterion is designed to limit category J's hypervolume. The vigilance test is defined as $M_J \leq \rho$, where ρ represents the maximum allowed hypervolume.

Learning. If the selected category resonates (i.e., the match criterion is satisfied), then learning occurs. The sample count and means are updated using Eq. (37) and Eq. (38), respectively. The covariance matrix is updated as:

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_J(new) = &\left(\frac{n_J(old)}{n_J(new)}\right)\hat{\boldsymbol{\Sigma}}_J(old)\\ &+ \frac{1}{n_J(new)}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_J(new))(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_J(new))^T \odot \boldsymbol{I},\end{aligned} \tag{59}$$

which corresponds to the sequential maximum-likelihood estimation of parameters for a multidimensional Gaussian distribution (Vigdor & Lerner, 2007). The Hadamard product $\odot$ is used when a diagonal covariance matrix is desired. Otherwise, a new category is created with $n_{N+1} = 1$, $\boldsymbol{\mu}_{N+1} = \boldsymbol{x}$, and $\boldsymbol{\Sigma}_{N+1} = \boldsymbol{\Sigma}_{init}$. Naturally, the initial covariance matrix should satisfy the vigilance constraint (i.e., $\boldsymbol{\Sigma}_{init} = \sigma^2_{init}\boldsymbol{I}$, where $\sigma^2_{init} \ll \rho^{1/d}$). In this ART model, categories can grow and shrink.

Many studies further developed Bayesian ART. For instance, to reduce the original model's computational cost and noise sensitivity, as well as to circumvent the instability issues associated with estimating covariance matrices with small sample sizes in high dimensional spaces, kernel Bayesian ART (Masuyama et al., 2018a) uses the kernel Bayes' rule (Fukumizu et al., 2013) and the correntropy-induced metric (Liu et al., 2007; Santamaria et al., 2006) for the activation and match functions, respectively. Topological kernel Bayesian ART (Masuyama et al., 2019) further extends the latter ART model by incorporating topological learning based on growing neural gas (Fritzke, 1995) to reduce input order sensitivity (see Sec. 6.1). As another example, to realize an associative memory, the Bayesian ART variant in (Chin et al., 2016) employs a multiple channel version of the adaptive resonance associative map model (Tan, 1995) (see Sec. 3.1.7).

2.1.11. Grammatical ART. The Grammatical ART (GramART) architecture (Meuth, 2009) represents a specialized version of ART designed to work with variable-length input patterns which are used to encode grammatical structure. It builds templates while adhering to a Backus-Naur form of grammatical structure (Knuth, 1964).

LTM. To allow for comparisons between variable-length input patterns, GramART uses a generalized tree representation to encode its internal categories. Each node in a category's tree contains an array representing the distribution of the different possible grammatical symbols at that node.

Activation. The activation function for a category j is defined as a parallel to fuzzy ART's activation function (Sec. 2.2), but GramART defines its own operator for calculating the intersection between a category and an input pattern. A tree in GramART is defined as an ordered pair (V, R) where V is a set of nodes (vertices) and R is a set of binary relations that describe the structure of the tree. For nodes v_1 and v_2:

$$R(v_1, v_2) = \begin{cases} 0, & \text{if } v_2 \text{ is not a successor of } v_1 \\ > 0, & \text{if } v_2 \text{ is a successor of } v_1 \end{cases}, \tag{60}$$

The activation of a category j in GramART is given by

$$T_j = \frac{|\boldsymbol{x} \cap \boldsymbol{w}_j|}{\|\boldsymbol{w}_j\|}, \tag{61}$$

where the intersection operator $|\boldsymbol{x} \cap \boldsymbol{w}_j|$, referred to in GramART as the trace of $\boldsymbol{x}$ in $\boldsymbol{w}_j$, is defined as:

$$|\boldsymbol{x} \cap \boldsymbol{w}_j| = \sum_{i=0}^{r} w_j(i, x_i), \tag{62}$$

and where $w_j(i, x_i)$ represents the value of node i in the template $\boldsymbol{w}_j$ corresponding to the same symbol in the input pattern $\boldsymbol{x}$. The tree norm operator $\|\boldsymbol{w}_j\|$ is defined as the number of nodes in the tree.

Match and resonance. The category with the highest activation value is chosen using WTA selection, and the following vigilance criterion is checked to determine whether the input pattern resonates with this category:

$$M_J = \frac{|\boldsymbol{x} \cap \boldsymbol{w}_J|}{\|\boldsymbol{x}\|} > \rho. \tag{63}$$

If this vigilance criterion is satisfied, resonance occurs and the category is allowed to learn this input pattern. Otherwise, it is reset, and the category with the next best activation is checked.

Learning. When resonance occurs, the weight of the winning category J is updated using the following learning rule for each node i in the template:

$$w_{J,i}(new) = \frac{w_{J,i}(old)U_{J,i} + \delta_J}{U_{J,i} + 1}, \tag{64}$$

where

$$\delta_j = \begin{cases} 1, & \text{if } x_i = j \\ 0, & \text{otherwise} \end{cases}, \tag{65}$$

and $U_{J,i}$ is the number of prior updates to node i in this category.

The weights are updated recursively down the grammar tree, and they reflect the probability of a tree symbol occurring in the node representing this particular category.

2.1.12. Validity Index-Based Vigilance Fuzzy ART. The validity index-based vigilance fuzzy ART (CVIFA) (Brito da Silva & Wunsch II, 2017) endows fuzzy ART with a second vigilance criterion based on cluster validity indices (Xu & Wunsch II, 2009). The usage of this immediate reinforcement signal alleviates input order dependency and allows for a more a robust hyperparameterization.

LTM. This is a fuzzy ART-based architecture. Therefore, categories are hyperrectangles as described in Sec. 2.2.

Activation. The CVIFA activation function is equal to fuzzy ART's and thus, is computed using Eq. (16) in Sec. 2.2.

Match and resonance. After a winner J is selected, the first match function (M_J^1) is identical to fuzzy ART's (Eq. (18) in Sec. 2.2), whereas the second (M_J^2) is defined as

$$M_J^2 = \Delta f = f(\hat{\Omega}) - f(\Omega), \tag{66}$$

which represents the penalty (or reward) incurred by assigning sample $\boldsymbol{x}$ to category J and thereby changing the current clustering state of the data set from Ω to $\hat{\Omega}$ (if there is no change in assignment, then $M_J^2 = 0$). The function $f(\Omega)$ corresponds to a cluster validity index value given a partition $\Omega = \{\omega_1, ..., \omega_k\}$ of disjointed clusters ω_i (defined by categories i), where $\bigcup_{i=1}^{k} \omega_i = \boldsymbol{X}$. The second vigilance region is then $VR_J^2 = \{\boldsymbol{x} : M_J^2(\boldsymbol{x}) \geq \rho_2\}$, and $\rho_2 \in \mathbb{R}$. The second vigilance criterion checks if $\mathbb{1}_{VR_J^2}(\boldsymbol{x}) = 1$. In the affirmative case, the category is allowed to learn. Note that the discussion so far implies the maximization of a cluster validity index; naturally, when minimization is sought, the inequality in the definition of VR_J^2 should be reversed. This is a greedy algorithm that selects the best clustering assignment based on immediate feedback. Of course, performance is biased toward the data structures favored by the selected cluster validity index.

Learning. If both vigilance tests are satisfied, then learning ensues using fuzzy ART's learning rule (Eq. (21) in Sec. 2.2). Otherwise, the search resumes or a new category is created. Note that the CVIFA model learns in offline mode, given that the entire data is used to compute Eq. (66).

2.1.13. Dual Vigilance Fuzzy ART. The dual vigilance fuzzy ART (DVFA) (Brito da Silva et al., 2019) seeks to retrieve arbitrarily shaped clusters with low parameterization requirements via a single fuzzy ART module. This is accomplished by augmenting fuzzy ART with two vigilance parameters, namely, the upper bound ($\rho_{\text{UB}} \in [0, 1]$) and lower bound ($0 \leq \rho_{\text{LB}} \leq \rho_{\text{UB}} \leq 1$), representing quantization and cluster similarity, respectively.

LTM. The dual vigilance fuzzy ART categories are hyperrectangles (see Sec. 2.2).

Activation. The activation function of the dual vigilance fuzzy ART is the same as fuzzy ART's (Eq. (16) in Sec. 2.2).

Match and resonance. When a category J is chosen by the WTA competition, it is subjected to a dual vigilance mechanism. The first match function (M_J^1) uses ρ_{UB} in Eq. (18), whereas the second (M_J^2) employs a more relaxed constraint; i.e., it uses ρ_{LB} in Eq. (18).

Learning. If the first vigilance criterion is satisfied, then learning proceeds as in fuzzy ART (Eq. (21)). Otherwise, the second test is performed, and, if satisfied, a new category is created and mapped to the same cluster as the category undergoing the dual vigilance tests via a mapping matrix $\boldsymbol{W}^{map} = \left[w_{n,k}^{map}\right]_{N \times K}$ (where N is the number of categories and K is the number of clusters):

$$w_{n,k}^{map}(new) = \begin{cases} 1, & \text{if } n = N+1 \text{ and } k = K \\ 0, & \text{if } n = N+1 \text{ and } k \neq K \\ w_{n,k}^{map}(old), & \text{if } n \neq N+1 \text{ and } k = K \\ w_{n,k}^{map}(old), & \text{if } n \neq N+1 \text{ and } k \neq K \end{cases}. \tag{67}$$

Alternately, if both tests fail, then the search continues with the next highest ranked category; if there are none left, then a new node is created and the matrix $\boldsymbol{W}^{map}$ expands:

$$w_{n,k}^{map}(new) = \begin{cases} 1, & \text{if } n = N+1 \text{ and } k = K+1 \\ 0, & \text{if } n = N+1 \text{ and } k \neq K+1 \\ 0, & \text{if } n \neq N+1 \text{ and } k = K+1 \\ w_{n,k}^{map}(old), & \text{if } n \neq N+1 \text{ and } k \neq K+1 \end{cases}. \tag{68}$$

The associations between categories and clusters are permanent in this incremental many-to-one mapping (multi-prototype representation of clusters), and they enable the data structures of arbitrary geometries to be detected by dual vigilance fuzzy ART's simple design.

2.2. TOPOLOGICAL ARCHITECTURES

The ART models discussed in this section are designed to enable multi-category representation of clusters, thus capturing the data topology more faithfully. Generally, they are used to cluster data in which arbitrarily shaped structures are expected (multi-prototype clustering methods).

2.2.1. Fuzzy ART with Group Learning. Fuzzy ART with group learning model (Isawa et al., 2007) augments fuzzy ART (Sec. 2.2) with topology learning (inspired by neural-gas (Martinetz & Schulten, 1994; Martinetz & Shulten, 1991)) to retrieve clusters with arbitrary shapes. The code representation, LTMs and dynamics of fuzzy ART remain the same. However, when a sample is presented, a connection between the first and second resonant categories (if both exist) is created by setting the corresponding entry of an adjacency matrix to one. This model also possesses an age matrix, which tracks the duration of such connections and whose dynamics are as follows: the entry related to the first and second current resonant categories is refreshed (i.e., set to zero) following a sample presentation, whereas all other entries related to the first resonant category are incremented by one. Connections with an age value above a certain threshold expire, i.e., they are pruned (note that the threshold varies deterministically over time). This procedure allows this model to dynamically create and remove connections between categories during learning (co-occurrence of resonant categories, thus following a Hebbian approach). Clusters are defined by groups of connected categories.

The fuzzy ART combining overlapped category in consideration of connections variant (Isawa et al., 2008a) was developed to mitigate category proliferation, which is accomplished by merging the first resonant category with another connecting and overlapping category. Another variant introduced in (Isawa et al., 2008b, 2009) augments the latter model with individual and adaptive vigilance parameters to further reduce category proliferation.

2.2.2. TopoART. Fuzzy topoART (Tscherepanow, 2010) is a model that combines fuzzy ART (Sec. 2.2) and topology learning (inspired by self-organizing incremental neural networks (Furao & Hasegawa, 2006)). Specifically, it features the same representation, activation/match functions, vigilance test and search/learning mechanisms as fuzzy ART, while integrating noise robustness and topology-based learning.

Briefly, the topoART model consists of two fuzzy ART-based modules (topoARTs A and B) that cluster, in parallel, the data in two hierarchical levels, while sharing the same complement coded inputs. Each category is endowed with an instance counting feature n (i.e., sample count), such that every τ learning cycles (i.e., iterations) categories that encoded less than a minimum number of samples ϕ are dynamically removed. Once this threshold is reached, "candidate" categories become "permanent" categories which can no longer be deleted. In this setup, module A serves as a noise filtering mechanism for module B. The propagation of a sample to module B depends on which type of module A's category was activated. Specifically, a sample is fed to module B if and only if the corresponding module A's resonant category is "permanent"; therefore, module B will only focus on certain regions of the data space. Note that no additional information is passed from module A to B, and both can form clusters independently.

Regarding the hierarchical structure, the vigilance parameters of modules A and B are related by

$$\rho_b = \frac{1}{2}\left(\rho_a + 1\right), \tag{69}$$

such that module B's maximum category size is 50% smaller than module A's (ρ_a and ρ_b are module A's and B's vigilance parameters, respectively), which implies that module B has a higher granularity ($\rho_b \geq \rho_a$) and thus yields a finer partition of the data set.

TopoART employs competitive and cooperative learning: not only the winner category J_1 but also the second winner J_2 is allowed to learn (naturally, both need to satisfy the vigilance criteria). The learning rates are set as $\beta_{J_2} < \beta_{J_1} = 1$, such that the second winner partially learns to encode the presented sample. If the first and second winner both exist, then they are linked to establish a topological structure. These lateral connections are permanent, unless categories are removed via the noise thresholding procedure. Clusters are formed by the connected categories, thus better reflecting the data distribution and enabling the discovery of arbitrarily shaped data structures (topoART is a graph-based multi-prototype clustering method).

Finally, in prediction mode, the following activation function, which is independent of category size, is used:

$$T_j = 1 - \frac{\left\| \left(\boldsymbol{x} \wedge \boldsymbol{w}_j \right) - \boldsymbol{w}_j \right\|_1}{\| \boldsymbol{x} \|_1}, \tag{70}$$

the vigilance test is neglected, and only “permanent” nodes can be activated.

A number of topoART variants have been developed in the literature, e.g., the hypersphere topoART (Tscherepanow, 2012), which replaces fuzzy ART modules with hypersphere ARTs (Sec. 2.1.7); the episodic topoART (Tscherepanow et al., 2012), which incorporates temporal information (i.e., time variable and thus the order of input presentation) to build a spatio-temporal mapping throughout the learning process and generate “episode-like” clusters; and the topoART-AM (Tscherepanow et al., 2011), which builds hierarchical hetero-associative memories via a recall mechanism.

2.3. HIERARCHICAL ARCHITECTURES

Elementary ART modules have been used as building blocks to construct both bottom-up (agglomerative) and top-down (divisive) hierarchical architectures. Typically, these follow one of two designs (Massey, 2009): (i) cascade (series connection) of ART modules in which the output of a preceding ART layer is used as the input for the succeeding one, or (ii) parallel ART modules enforcing different vigilance criteria while having a common input layer.

2.3.1. ARTtree. The ARTtree (Wunsch II et al., 1993) is a way of building a hierarchy of ART neural modules in which an input sample is sent simultaneously to every module in every level of the tree. Each node in the ART tree hierarchy is connected to one of its parent’s F_2 categories, and each of the F_2 categories in this node is connected to one of its children. The nodes in each layer of the tree hierarchy share a common vigilance value, and the vigilance typically increases further down the tree such that tiers of the tree that have more nodes are associated with higher vigilance values.

When an input sample is presented to the ARTtree hierarchy, all the ART nodes can be allowed to perform their match and activation functions, but only the node connected to its parent's winning F_2 category is allowed to resonate with and learn this pattern. Therefore, resonance only cascades down a single path in the ARTtree, and no other nodes outside that path are allowed to learn this sample. This can effectively allow ART to perform a type of varying-k-means clustering (Wunsch II et al., 1993).

The highly parallel nature of ARTtree lends itself well to hardware-based implementations, such as optoelectronic implementations (Wunsch II et al., 1993) and massively parallel implementations via general purpose Graphics Processing Unit (GPU) acceleration (Kim & Wunsch II, 2011). The study presented in (Kim & Wunsch II, 2011) performed this task using NVIDIA CUDA GPU hardware and an implementation of ARTtree that uses fuzzy ART units in the tree nodes. The results reported in the study show a massive speed boost for deep trees when compared to the CPU in terms of computing time, while smaller trees performed worse on the GPU due to the high data transfer penalties between the CPU and GPU memory.

2.3.2. Self-Consistent Modular ART. The self-consistent modular ART (SMART) (Bartfai, 1994) is a modular architecture designed to perform hierarchical divisive clustering (i.e., to represent different levels of data granularity in a top-down approach). It builds a self-consistent hierarchical structure via self-organization and uses ART 1 (Sec. 2.1.1) as elementary units. In this architecture, a number of ART modules operate in parallel with different vigilance parameter values, while receiving the same input samples and connecting in a manner that makes the hierarchical cluster representation self-consistent. These connections are such that many-to-one mapping of specific to general categories is learned across such modules. Specifically, the hierarchy is explicitly represented via associative links between modules.

Concretely, a two-level SMART architecture can be implemented using an ARTMAP (Sec. 3.1.1) in auto-associative mode; i.e., ARTMAP is used in an unsupervised manner by presenting the same input sample to both modules A and B with different vigilance parameters and forcing a hierarchical structure by making $\rho_A > \rho_B$, such that module B enforces its categorization (an internal supervision) on module A.

2.3.3. ArboART. ArboART (Ishihara et al., 1995) is an agglomerative hierarchical clustering method based on ART. More specifically, it uses ART 1.5-SSS (small sample size) (Ishihara et al., 1993) (variant of ART 1.5 (Levine & Penz, 1990), which in turn is a variation of ART 2 (Carpenter & Grossberg, 1987b)), as a building block. Briefly, prototypes of one ART are the inputs for another ART with looser vigilance (similarity constraint). Therefore, prototypes obtained from a lower level (bottom part of the dendrogram) are fed to the next ART layer. ART modules on higher layers have decreasingly lower vigilance values, i.e., the similarity constraint is less strict. This enables the construction of a tree (hierarchical graph structure). One of the advantages over traditional hierarchical methods is that it does not require a full recomputation when a new sample is added, only partial recomputations are needed in ART (inside the specific clusters). ArboART uses several layers of ART as well as one-pass learning. Concretely, it makes super-clusters of previous clusters in a hierarchical way, thereby making a generalization of categories in the process.

2.3.4. Joining Hierarchical ART. The joining hierarchical ART (HART-J) (Bartfai, 1996; Bartfai & White, 2000) is a hierarchical agglomerative clustering method (bottom-up approach) that uses ART 1 modules (Sec. 2.1.1) as building blocks and follows a cascade design. Specifically, each layer of this multi-layer model corresponds to an ART 1 network that clusters the prototypes generated by the preceding layer. The input of layer l is given by:

$$\boldsymbol{x}_l = \boldsymbol{x}_{l-1} \cap \boldsymbol{w}_{l-1,J}, \; l = \{2, ..., L\}, \tag{71}$$

where L is the number of layers, $\boldsymbol{w}_{l-1,J}$ is the resonant neuron J of layer $l-1$ and $\boldsymbol{x}_1$ is equal to the input sample $\boldsymbol{x}$. Interestingly, it is not imperative to reduce the vigilance values at higher layers to generate the hierarchy: the "effective" vigilance level of layer l is given by:

$$\hat{\rho}_l = \prod_{j=1}^{l} \rho_j, \tag{72}$$

which decreases even if the vigilance increases with l given that $\rho_l \in [0, 1] \; \forall l$. This fact is used to derive an upper bound for the maximum number of layers L_{max}. If all vigilance values are equal to ρ, then $L_{max} = \lfloor n + 1 \rceil$, where n is the minimum integer that satisfies

$$n > -\frac{\log d}{\log \rho}, \tag{73}$$

assuming that the input samples are complement coded (see Sec. 2.2).

Naturally, succeeding networks can learn (at most) the number of prototypes from the previous layer. Learning can occur in sequential (learning is paused until the previous layer is stabilized) or parallel (learning occurs in each layer in each presentation of inputs) modes. The former generates fewer categories, but the training time, measured in number of epochs, is much smaller using the parallel approach.

HART-J is compared to SMART in (Bartfai, 1995). Contrary to SMART, HART-J has no associative connection or feedback between hierarchical layers as a mechanism to enforce self-consistency. The constraint that causes the lower layers to have greater vigilance values than the higher layers guarantees consistency. In HART-J, the hierarchies "emerge" since there are no explicit links. It is reported that SMART builds a less compact model (larger number of categories) due to categorization forced by its internal feedback mechanism, whereas HART-J builds a simpler and more compact network.

2.3.5. Hierarchical ART with Splitting. The hierarchical ART with splitting (HART-S) (Bartfai & White, 1997b, 2000) consists of a cascade of ART 1 (Sec. 2.1.1) modules that performs incremental hierarchical divisive clustering (successive splitting in a top-down approach). A fuzzy HART-S (Bartfai & White, 1997a) variant uses a cascade of fuzzy ARTs, where each module clusters the difference between the input and the weight vector of the resonant category belonging to the preceding layer. Specifically, the input to layer l ($l = \{1, ..., L\}$, where L is the maximum number of layers) is given by:

$$\boldsymbol{x}_l = \boldsymbol{x}_{l-1} \wedge \boldsymbol{w}^c_{l-1,J}, \tag{74}$$

which recursively corresponds to

$$\boldsymbol{x}_l = \boldsymbol{x}_1 \wedge \left(\bigwedge_{i=1}^{l-1} \boldsymbol{w}^c_{i,J} \right), \tag{75}$$

where $\boldsymbol{x}_1 = \boldsymbol{x}$ is the data sample and $\boldsymbol{w}^c_{i,J}$ is the complement of the weight vector associated with the resonant neuron J of layer i.

The hierarchy is explicitly represented by links between parent and children categories in a tree-like structure. These adaptive associative connections between consecutive modules ensure that only children of the preceding parent module can be activated. In its most general case, the fuzzy ART modules in each layer have their own set of parameters. Particularly, Fuzzy HART-S uses two global parameters: a resolution parameter $\epsilon \in [0, 1]$ to control the depth of the hierarchical tree (i.e., if $\|x_l\|_1 < \epsilon S$, then there is no more splitting, where $S = \|\boldsymbol{x}\|_1$) and a feature threshold parameter to control the propagation of features throughout the layers.

Strategies to prune and rebuild prototypes to improve HART-S in terms of network complexity (measured by the number of categories) are presented in (Bartfai & White, 1998). During learning, the former strategy removes small clusters (and all their children if applicable) based on a cluster size threshold (percentage of the total number of samples), and the latter changes the components of a prototype weight vector to better reflect the samples associated with them.

2.3.6. Distributed Dual Vigilance Fuzzy ART. The distributed dual vigilance fuzzy ART (DDVFA) (Brito da Silva et al., 2020) is a dual vigilance-based ART model designed to improve memory compression and perform several ART-based hierarchical agglomerative clustering (HAC) methods online. It consists of a global ART module whose F$_2$ nodes are local fuzzy ARTs: the global module is used for decision making while the local module builds multi-prototype representations of clusters (many-to-one mappings).

The activation of a global ART F$_2$ node i (T_i^g) is a function $f(\cdot)$ of the activations of the k F$_2$ nodes of its corresponding local fuzzy ART module:

$$T_i^g = f\left(T_1^i,\ \dots,\ T_j^i,\ \dots,\ T_k^i\right), \tag{76}$$

where T_j^i is the activation function of the F$_2$ node j of the local fuzzy ART module i, which uses a higher order activation function defined as

$$T_j^i = \left(\frac{\|\boldsymbol{x} \wedge \boldsymbol{w}_j^i\|_1}{\alpha + \|\boldsymbol{w}_j^i\|_1}\right)^{\gamma}, \tag{77}$$

and $\gamma \geq 1$ is a power parameter whose role is akin to a kernel width. Similarly, the match function of a global ART F$_2$ node i (M_i^g) is defined as

$$M_i^g = g\left(M_1^i,\ \dots,\ M_j^i,\ \dots,\ M_k^i\right), \tag{78}$$

where M_j^i is the match function of the F$_2$ node j of the local fuzzy ART module i, which uses the following normalized higher order match function

$$M_j^i = \left(\frac{\|\boldsymbol{w}_j^i\|_1}{\|\boldsymbol{x}\|_1}\right)^{\gamma^*} T_j^i, \tag{79}$$

where $0 \leq \gamma^* \leq \gamma$ is the reference kernel width with respect to which the match function is normalized. Both functions $f(\cdot)$ and $g(\cdot)$ are based on HAC methods, as listed in Table 2.

The DDVFA features a dual vigilance mechanism: when a sample $\boldsymbol{x}$ is presented, and the F$_2$ node I of the global ART is the winner, then $VR_I^g = \{\boldsymbol{x} : M_I^g(\boldsymbol{x}) \geq \rho_{LB}\}$ and $\rho_{LB} \in [0, 1]$. The vigilance criterion checks if $\mathbb{1}_{VR_I^g}(\boldsymbol{x})$ is true. If not, the search continues, or a new local fuzzy ART module is created. If so, the corresponding local fuzzy ART module is allowed to learn. The local fuzzy ART module imposes a stricter constraint for its winner node J: $VR_J^l = \{\boldsymbol{x} : M_J^l(\boldsymbol{x}) \geq \rho_{UB}\}$ and $0 \leq \rho_{LB} \leq \rho_{UB} \leq 1$. Again, the vigilance criterion checks if $\mathbb{1}_{VR_J^l}(\boldsymbol{x})$ is true, and, if so, category J is allowed to learn. Otherwise, the search resumes or a new node is created following the standard ART dynamics.

When input order cannot be addressed via an offline pre-processing strategy (Sec. 6.1), then DDVFA should be used in conjunction with a Merge ART module to mitigate input order dependency in online learning applications. This module is connected to DDVFA in series, i.e., in a cascade design. The inputs to Merge ART are fuzzy ART modules with all their corresponding categories. Like DDVFA, Merge ART's F$_2$ nodes are also fuzzy ART modules. When a DDVFA's fuzzy ART node l is fed to Merge ART, an activation matrix $\boldsymbol{T}_{k,l} = [t_{i,j}]_{R \times C}$ (where R and C are the number of categories in Merge ART node k and DDVFA node l, respectively) is computed as

$$t_{i,j} = \left(\frac{\|\boldsymbol{w}_j^l \wedge \boldsymbol{w}_i^k\|_1}{\alpha + \|\boldsymbol{w}_i^k\|_1} \right)^{\gamma}, \tag{80}$$

where $\boldsymbol{w}_j^l$ is the weight vector of category j of DDVFA local fuzzy ART module l, and $\boldsymbol{w}_i^k$ is the weight vector of category i of Merge ART module k. The actual activation of Merge ART node k uses matrix $\boldsymbol{T}_{k,l}$ and follows one of the HAC forms as listed in Table 3. Assuming Merge ART's F$_2$ node K is the winner, its match matrix $\boldsymbol{M}_{K,l} = [m_{i,j}]_{R \times C}$ is computed as

$$m_{i,j} = \left(\frac{\|\boldsymbol{w}_i^K\|_1}{\|\boldsymbol{w}_j^l\|_1} \right)^{\gamma^*} t_{i,j}, \tag{81}$$

Table 2. DDVFA's activation and match functions.

HAC method	$T_i^g = f(\cdot)$	$M_i^g = g(\cdot)$
single	$\max_j \left(T_j^i\right)$	$\max_j \left(M_j^i\right)$
complete	$\min_j \left(T_j^i\right)$	$\min_j \left(M_j^i\right)$
median	$\operatorname{median}_j \left(T_j^i\right)$	$\operatorname{median}_j \left(M_j^i\right)$
average[a]	$\frac{1}{k_i} \sum_{j=1}^{k_i} T_j^i$	$\frac{1}{k_i} \sum_{j=1}^{k_i} M_j^i$
weighted[b]	$\sum_{j=1}^{k_i} p_j T_j^i$	$\sum_{j=1}^{k_i} p_j M_j^i$
centroid[c]	$\left(\frac{\lVert \boldsymbol{x} \wedge \boldsymbol{w}_c^i \rVert_1}{\alpha + \lVert \boldsymbol{w}_c^i \rVert_1}\right)^{\gamma}$	$\left(\frac{\lVert \boldsymbol{w}_c^i \rVert_1}{\lVert \boldsymbol{x} \rVert_1}\right)^{\gamma^*} T_i^g$

[a,b] k_i is the number of F_2 nodes in local fuzzy ART module i.

[b] $p_j = \frac{n_j^i}{n_i^g}$, where n_j^i is the number of samples encoded by category j of local fuzzy ART module i, and $n_i^g = \sum_j n_j^i$.

[c] $\boldsymbol{w}_c^i$ is the "centroid" representing all categories of local fuzzy ART module i, whose l component is computed as $w_{c,l}^i = \min_j \left(w_{j,l}^i\right)$ for $l = \{1, \ldots, 2d\}$.

where the actual match of Merge ART node K uses matrix $\boldsymbol{M}_{K,l}$ and one of the HAC formulations listed in Table 3. If the vigilance constraint is satisfied (i.e., $M_K \geq \rho_{LB}$), then $ART_K(new) \leftarrow ART_K(old) \cup ART_l$, i.e., ART_K and ART_l become a single module. To further reduce model complexity, the final step of Merge ART consists of feeding the weight vectors of each ART module to an independent fuzzy ART parameterized with $\rho = \rho_{UB}$, γ and γ^*. Note that the Merge ART module can be run once or until convergence, where the latter is defined as no change in the Merge ART nodes between two consecutive iterations.

Table 3. Merge ART's activation and match functions.

HAC method	$T_k = f(\cdot)$	$M_k = g(\cdot)$
single	$\max\limits_{i,j} \left([t_{ij}]\right)$	$\max\limits_{i,j} \left([m_{ij}]\right)$
complete	$\min\limits_{i,j} \left([t_{ij}]\right)$	$\min\limits_{i,j} \left([m_{ij}]\right)$
median	$\operatorname*{median}\limits_{i,j} \left([t_{ij}]\right)$	$\operatorname*{median}\limits_{i,j} \left([m_{ij}]\right)$
average	$\frac{1}{RC} \sum\limits_{i=1}^{R} \sum\limits_{j=1}^{C} t_{ij}$	$\frac{1}{RC} \sum\limits_{i=1}^{R} \sum\limits_{j=1}^{C} m_{ij}$
weighted[a]	$\sum\limits_{i=1}^{R} \sum\limits_{j=1}^{C} p_i p_j t_{ij}$	$\sum\limits_{i=1}^{R} \sum\limits_{j=1}^{C} p_i p_j m_{ij}$
centroid[b]	$\left(\frac{\|\boldsymbol{w}_c^k \wedge \boldsymbol{w}_c^l\|_1}{\alpha + \|\boldsymbol{w}_c^k\|_1}\right)^{\gamma}$	$\left(\frac{\|\boldsymbol{w}_c^k\|_1}{\|\boldsymbol{w}_c^l\|_1}\right)^{\gamma^*} T_k$

[a] $p_i = \frac{n_i^k}{n_k}$ and $p_j = \frac{n_j^l}{n_l}$, where n_i^k is the number of samples encoded by category i of Merge ART node k, and $n_k = \sum\limits_i n_i^k$. The variables n_j^l and n_l refer to DDVFA node l and are defined similarly.

[b] $\boldsymbol{w}_c^k$ and $\boldsymbol{w}_c^l$ are the "centroids" representing all categories of $ART_k^{(2)}$ and $ART_l^{(1)}$, respectively. Their components are given by $w_{c,n}^k = \min\limits_j \left(w_{j,n}^k\right)$ and $w_{c,n}^l = \min\limits_j \left(w_{j,n}^l\right)$, where $n = \{1, \ldots, 2d\}$.

2.4. BICLUSTERING AND DATA FUSION ARCHITECTURES

2.4.1. Fusion ART.

Fusion ART (Tan et al., 2007) extends ART capabilities by augmenting it with multiple and independent F_1 layers (input channels or fields), all of which are connected to a shared F_2 layer. This model is then capable of learning mappings across multiple channels simultaneously.

Activation. The activation function of a category j is a weighted sum of the activation functions of each input field

$$T_j = \sum_{k=1}^{K} \gamma^k \frac{\|\boldsymbol{x}^k \wedge \boldsymbol{w}_j^k\|_1}{\alpha^k + \|\boldsymbol{w}_j^k\|_1}, \tag{82}$$

where $\boldsymbol{x}^k$ is the complement-coded input to the k^{th} F_1 layer (F_1^k or channel k), and $\gamma^k \in [0,1]$ and $\alpha^k \in (0,\infty)$ are the contribution and choice parameters of F_1^k, respectively. The variable K is the total number of input channels such that $\boldsymbol{x} = [\boldsymbol{x}^1, ..., \boldsymbol{x}^k, ..., \boldsymbol{x}^K]$ and category j's LTM is $\boldsymbol{w}_j = [\boldsymbol{w}_j^1, ..., \boldsymbol{w}_j^k, ..., \boldsymbol{w}_j^K]$.

Match and resonance. When category J is selected by the WTA competition, one match function is computed for each channel

$$M_J^k = \frac{\|\boldsymbol{y}^{(F_1^k)}\|_1}{\|\boldsymbol{x}^k\|_1} = \frac{\|\boldsymbol{x}^k \wedge \boldsymbol{w}_J^k\|_1}{\|\boldsymbol{x}^k\|_1}, \tag{83}$$

where $VR_J^k = \{\boldsymbol{x}^k : M_J^k(\boldsymbol{x}^k) \geq \rho^k\}$, and $\rho^k \in [0,1]$ is F_1^k's vigilance parameter. The global vigilance test is satisfied if all channels meet their individual vigilance criteria simultaneously, i.e., if $\bigwedge_{k=1}^{K} \mathbb{1}_{VR_J^k}(\boldsymbol{x}^k) = 1$. A mismatch (i.e., the latter condition is not satisfied) triggers a category reset and the match tracking mechanism, which simultaneously raises all input fields' vigilance parameters. The search then continues until a resonant category is found or created. Then, learning takes place as

$$\boldsymbol{w}_J^k(new) = (1-\beta^k)\boldsymbol{w}_J^k(old) + \beta^k(\boldsymbol{x}^k \wedge \boldsymbol{w}_J^k(old)), \ \forall k, \tag{84}$$

where $\beta^k \in (0,1]$ is the learning parameter of layer F_1^k. When a new input is presented, $\rho^k = \bar{\rho}_k$, where $\bar{\rho}_k$ is the baseline vigilance of layer F_1^k. Additionally, if an input to a channel is not present, then it is set to $\vec{\mathbf{1}}$ to enable the prediction/recovery of missing values.

Fusion ART generalizes some other ART models, i.e., by appropriately designing fusion ART, it can reduce to different ART models and perform distinct machine learning modalities: (i) 1 channel (samples) fusion ART reduces to ART (Carpenter et al., 1991c) (Sec. 2.2) and performs match-based unsupervised learning, (ii) 2 channels (samples and class labels) fusion ART reduces to adaptive resonance associative map - ARAM (Tan, 1995) (Sec. 3.1.7) and performs association-based supervised learning and (iii) 3 channels (states, actions and rewards) fusion ART reduces to fusion architecture for learning, cognition, and navigation - FALCON (Tan, 2004) (Secs. 4.1 and 4.2) and performs reinforcement learning. Additionally, fusion ART can perform instruction-based learning by rule-based knowledge integration (generation of IF-THEN rules mapping antecedents and consequents from one channel to another and rule insertion capability).

Fusion ART has been used in the realization of a hierarchical planner (Subagdja & Tan, 2012), as well as of different types of long term memory models: episodic (Leconte et al., 2014, 2016; Nasir et al., 2019; Nasir et al., 2018; Park & Kim, 2016; Park et al., 2018; Park et al., 2015; Subagdja & Tan, 2015; Subagdja et al., 2012; Wang et al., 2010, 2012a,b, 2017), semantic (Nasir et al., 2019; Nasir et al., 2018; Subagdja et al., 2012; Wang et al., 2012b, 2017) and procedural (Wang et al., 2012b, 2017) among a number of other applications. See also (Tan et al., 2019) in this issue.

2.4.2. Biclustering ARTMAP. Biclustering ARTMAP (BARTMAP) (Xu & Wunsch II, 2011; Xu et al., 2012) is based on fuzzy ARTMAP (Carpenter et al., 1992) (Sec. 3.1.2) and was designed to find correlation-based subspace clustering. It uses two fuzzy ART modules (ART$_a$ and ART$_b$) connected through a regulatory inter-ART module to achieve a biclustering of the data matrix on both the input space (rows) and the feature space (columns). The ART$_b$ module is used to cluster the feature vectors and create a set of feature clusters. Then, the samples are presented to the ART$_a$ module while using the inter-ART module to integrate the clustering results on both the feature and input spaces and create biclusters that capture the local relations between the inputs and features. Note that BARTMAP learns in offline mode. This architecture was shown to perform fast and stable biclustering of gene expression data (Xu & Wunsch II, 2011) and later modified to build a collaborative filtering recommendation system (Elnabarawy et al., 2016).

The BARTMAP algorithm begins by presenting all the feature vectors to ART$_b$ (which is a standard fuzzy ART module), using it to build clusters of the feature vectors. Next, it begins presenting the input vectors to ART$_a$ and allows it to build clusters in the input space. If ART$_a$ places an input in a previously committed category, the inter-ART module then computes the similarity between the new sample and the samples in the existing cluster, but only within each feature cluster from ART$_\mathrm{b}$, thereby testing the correlation between the new sample and each of the existing biclusters. If any of the biclusters passes a user-defined correlation threshold η, the cluster is updated with the new sample. However, if none of the current biclusters passes, the ART$_\mathrm{a}$ vigilance threshold is temporarily increased (match tracking mechanism, see Sec. 3.1.1), and the sample is presented again to find a new cluster. If no suitable cluster is found that also satisfies the correlation threshold, the ART$_\mathrm{a}$ vigilance will eventually be increased enough to force the creation of a new cluster.

Consider the data matrix $\boldsymbol{X} = [x_{i,j}]_{N\times d}$, encompassing N samples in a d-dimensional feature space. After ART$_\mathrm{b}$ detects N_b clusters of features, the k^{th} input to ART$_\mathrm{a}$ becomes $\boldsymbol{x}_k = [\boldsymbol{x}_k^{c_1^b}, ..., \boldsymbol{x}_k^{c_i^b}, ..., \boldsymbol{x}_k^{c_{N_b}^b}] \in \mathbb{R}^d$, where $\boldsymbol{x}_k^{c_i^b}$ comprises the subset of components of $\boldsymbol{x}_k$ associated with the i^{th} feature cluster identified by ART$_\mathrm{b}$ (c_i^b). The similarity between the input sample $\boldsymbol{x}_k$ and an ART$_\mathrm{a}$ cluster c_j^a with n_j^a samples, across an ART$_\mathrm{b}$ feature cluster c_i^b with n_i^b features, is defined using the average Pearson correlation coefficient (Bain & Engelhardt, 1992) as follows:

$$\bar{r}_{c_j^a,c_i^b}(\boldsymbol{x}_k) = \frac{1}{n_j^a}\sum_{l=1,\boldsymbol{x}_l\in c_j^a}^{n_j^a} r_{c_j^a,c_i^b}(\boldsymbol{x}_k^{c_i^b},\boldsymbol{x}_l^{c_i^b}), \tag{85}$$

where

$$r_{c_j^a,c_i^b}(\boldsymbol{x}_k^{c_i^b},\boldsymbol{x}_l^{c_i^b}) = \frac{\sum_{t=1}^{n_i^b}(x_{k,t}^{c_i^b} - \bar{x}_k^{c_i^b})(x_{l,t}^{c_i^b} - \bar{x}_l^{c_i^b})}{\sqrt{\sum_{t=1}^{n_i^b}(x_{k,t}^{c_i^b} - \bar{x}_k^{c_i^b})^2}\sqrt{\sum_{t=1}^{n_i^b}(x_{l,t}^{c_i^b} - \bar{x}_l^{c_i^b})^2}}. \tag{86}$$

Here, $x_{m,t}^{c_i^b}$ refers to the value for sample $\boldsymbol{x}_m$ at feature t within the ART$_\mathrm{b}$ cluster c_i^b ($m = k, l$). Similarly, $\bar{x}_m^{c_i^b}$ denotes the average value of $\boldsymbol{x}_m$ across all the features in ART$_\mathrm{b}$'s cluster c_i^b:

$$\bar{x}_m^{c_i^b} = \frac{1}{n_i^b}\sum_{t=1}^{n_i^b} x_{m,t}^{c_i^b}. \tag{87}$$

2.4.3. Generalized Heterogeneous Fusion ART. The generalized heterogeneous fusion ART (Meng et al., 2014) is a model designed to perform co-clustering of heterogeneous data (i.e., mixed data types). It extends the heterogeneous fusion ART (Meng & Tan, 2012), which is a two-channels fusion ART-based model, to a multiple channel architecture. The distinctive characteristic of the generalized heterogeneous fusion ART is that its learning functions vary according to each data type, i.e., when a winner node J satisfies the vigilance criterion, different channels are adapted following different learning functions $f_L^k(\cdot)$. For instance, if the input $\boldsymbol{x}^k$ corresponds to a visual feature from image data or a text feature from a document, then the corresponding weight vector is updated following Eq. (84). Alternately, if $\boldsymbol{x}^k$ is a feature from data meta-information, then the weight vector of the corresponding channel k is adapted using the recursive mean formula

$$\boldsymbol{w}_J^k(new) = \left(1 - \frac{1}{n_J(new)}\right)\boldsymbol{w}_J^k(old) + \frac{1}{n_J(new)}\boldsymbol{x}^k, \tag{88}$$

$$n_J(new) = n_J(old) + 1, \tag{89}$$

where n_J corresponds to the number of samples encoded by node J.

Another key characteristic of the generalized heterogeneous fusion ART is the adaptive channel weighting: the contribution parameters are initially uniformly initialized, and then, during learning, undergo self-adaptation using

$$\gamma^k(new) = \frac{R^k}{\sum\limits_{k=1}^{K} R^k}, \ \forall k, \tag{90}$$

where

$$R^k = exp\left(-\frac{1}{N}\sum_{j=1}^{N} D_j^k\right), \tag{91}$$

$$D_j^k = \frac{\frac{1}{n_j}\sum_{l=1}^{n_j} \|\boldsymbol{w}_j^k - \boldsymbol{x}_l^k\|_1}{\|\boldsymbol{w}_j^k\|_1}. \tag{92}$$

The variable R is a robustness measure used to estimate the discriminative power of each channel given the intra-cluster scatter. In practice, performing the offline computations in Eq. (92) can be expensive. Therefore, since only D_J^k needs to be updated after the presentation of each sample, then $\gamma^k(new)$ can be estimated incrementally. Particularly, when there is a resonant committed node J, if $\boldsymbol{x}^k$ is a meta-information feature, then

$$\begin{aligned} D_J^k(new) = &\frac{n_J(old)}{n_J(new)\|\boldsymbol{w}_J^k(new)\|_1}\left(\|\boldsymbol{w}_J^k(old)\|_1 D_J^k(old) - \|\boldsymbol{w}_J^k(new) - \frac{n_J(old)}{n_J(new)}\boldsymbol{w}_J^k(old)\|_1\right. \\ &\left. + \frac{1}{n_J(old)}\|\boldsymbol{w}_J^k(new) - \boldsymbol{x}^k\|_1\right), \end{aligned} \tag{93}$$

otherwise,

$$\begin{aligned} D_J^k(new) = &\frac{n_J(old)}{n_J(new)\|\boldsymbol{w}_J^k(new)\|_1}\left(\|\boldsymbol{w}_J^k(old)\|_1 D_J^k(old) - \|\boldsymbol{w}_J^k(old) - \boldsymbol{w}_J^k(new)\|_1\right. \\ &\left. + \frac{1}{n_J(old)}\|\boldsymbol{w}_J^k(new) - \boldsymbol{x}^k\|_1\right). \end{aligned} \tag{94}$$

If a new category is created, regardless of $\boldsymbol{x}^k$ type, the contribution parameters are updated via a proportionality change

$$\gamma^k(new) = \frac{\left(R^k\right)^{\frac{N}{N+1}}}{\sum_{k=1}^{K}\left(R^k\right)^{\frac{N}{N+1}}}, \ \forall k, \tag{95}$$

where N is the number of categories.

Note that the generalized heterogeneous fusion ART can also include prior knowledge by appropriate initialization of the network.

2.4.4. Hierarchical Biclustering ARTMAP. Hierarchical biclustering ARTMAP (Kim, 2016) uses BARTMAP (2.4.2) iteratively to obtain a hierarchy of biclusters. The algorithm begins by running BARTMAP on the complement-coded data with low vigilance values, which produces a relatively small number of larger-sized biclusters. In the following step, hierarchical BARTMAP uses a bicluster matching threshold and a correlation fitness function to build and evaluate the biclusters at the current level. After that, the BARTMAP algorithm is used again on each of the resulting clusters with increased vigilance and correlation thresholds. These are adjusted by small values that are a function of the number of samples as well as the number of features and average correlation in each bicluster. The hierarchical BARTMAP algorithm repeats those two steps recursively for a specified number of times. Then, the best layer in the recursive tree that optimizes the desired cluster validity index (Xu & Wunsch II, 2009), or any other user-specified criteria, is chosen.

2.5. SUMMARY

Table 4 summarizes the nature of the category representations of the ART elementary models described in the previous subsections during activation, match and learning stages. Particularly, it lists if winner-takes-all (WTA) or distributed (D) coding is employed by these networks.

3. ART MODELS FOR SUPERVISED LEARNING

3.1. ARCHITECTURES FOR CLASSIFICATION

ART models used for supervised learning applications typically follow an ARTMAP architecture (Figure 2), which consists of two elementary ART units (ART$_a$ and ART$_b$) interconnected by an associative learning network, namely the map field, that performs multidimensional mappings between categories of both such units, as well as allowing for

Table 4. Summary of the code representations used by the unsupervised learning ART models.

ART model	Activation	Match	Learning	Reference(s)
ART 1	WTA	WTA	WTA	(Carpenter & Grossberg, 1987a)
ART 2-A	WTA	WTA	WTA	(Carpenter et al., 1991b)
Fuzzy ART	WTA	WTA	WTA	(Carpenter et al., 1991c)
Fuzzy Min-Max	WTA	WTA	WTA	(Simpson, 1993)
ARTtree	WTA	WTA	WTA	(Wunsch II et al., 1993)
SMART	WTA	WTA	WTA	(Bartfai, 1994)
ArboART	WTA	WTA	WTA	(Ishihara et al., 1995)
Distributed ART	D	D	D	(Carpenter, 1996a,b, 1997)
Gaussian ART	WTA	WTA	WTA	(Williamson, 1996)
HART-J/S	WTA	WTA	WTA	(Bartfai, 1996; Bartfai & White, 1997b)
Hypersphere ART	WTA	WTA	WTA	(Anagnostopoulos & Georgiopoulos, 2000)
Ellipsoid ART	WTA	WTA	WTA	(Anagnostopoulos & Georgiopoulos, 2001a,b)
Quadratic Neuron ART	WTA	WTA	WTA	(Su & Liu, 2002, 2005)
Bayesian ART	WTA	WTA	WTA	(Vigdor & Lerner, 2007)
Fusion ART	WTA	WTA	WTA	(Tan et al., 2007)
Fuzzy ART with Group Learning	WTA	WTA	WTA	(Isawa et al., 2007)
Grammatical ART	WTA	WTA	WTA	(Meuth, 2009)
TopoART	WTA	WTA	D	(Tscherepanow, 2010)
BARTMAP	WTA	WTA	WTA	(Xu & Wunsch II, 2011)
Generalized Heterogeneous Fusion ART	WTA	WTA	WTA	(Meng et al., 2014)
Hierarchical BARTMAP	WTA	WTA	WTA	(Kim, 2016)
CVIFA	WTA	WTA	WTA	(Brito da Silva & Wunsch II, 2017)
DVFA	WTA	WTA	WTA	(Brito da Silva et al., 2019)
DDVFA	D	D	WTA	(Brito da Silva et al., 2020)

WTA: winner-takes-all code.
D: distributed code.

associative recalls when the input to one of the ART modules is missing. Notably, ARTMAP models usually inherit the properties of their elementary ART building blocks. This section describes the main characteristics of members of the supervised ART family in terms of their map field LTM units, dynamics (which encompasses activation, match, resonance criterion and learning) and user-defined parameters. For clarity, Table 5 summarizes the notation used in the following subsections.

When an ARTMAP architecture is used for pattern recognition or classification tasks, typically ART$_a$ clusters data samples while ART$_b$ clusters class labels in parallel. Therefore, while ART maps samples to categories, an ARTMAP architecture goes one step further and maps categories to classes. During training, ART$_a$ is subjected to a certain level of agreement with ART$_b$'s activity, given that the latter encodes the target labels.

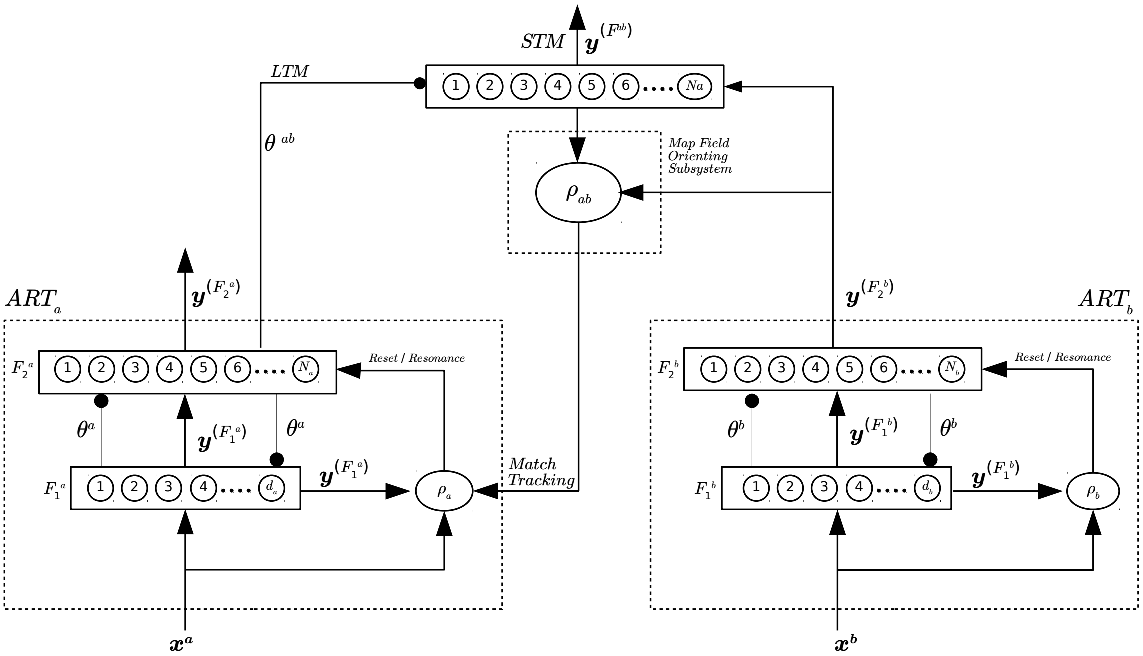


Figure 2. Elementary ARTMAP model.

Table 5. Supervised ART models notation.

Notation	Description
$\boldsymbol{x}^l$	input sample to ART$_l$ ($\boldsymbol{x}^l \in \boldsymbol{X}^l$)
d_l	input data dimensionality ($\boldsymbol{x}^l \in \mathbb{R}^{d_l}$)
F_1^l	feature representation field of ART$_l$
F_2^l	category representation field of ART$_l$
F^{ab}	map field
c^l	a category in ART$_l$
N_l	number of categories in ART$_l$
$\boldsymbol{y}^{(F_1^l)}$	F_1^l activity (STM)
$\boldsymbol{y}^{(F_2^l)}$	F_2^l activity (STM)
$\boldsymbol{y}^{(F^{ab})}$	F^{ab} activity (STM)
$\boldsymbol{\theta}^{ab}$	map field parameters (LTM unit)
M^{ab}	map field match function
J	ART$_\mathrm{a}$ chosen category index (via WTA)
K	ART$_\mathrm{b}$ chosen category index (via WTA)
ρ_l	vigilance parameter of ART$_l$
$\bar{\rho}$	ART$_\mathrm{a}$ baseline vigilance parameter

Variable l indexes the elementary ART modules: $l = a, b$.

This is performed by a second vigilance test that uses ART$_\mathrm{b}$'s supervisory signal (i.e., response) to trigger a mismatch or allow learning given an incorrect or correct prediction, respectively. Specifically, when ART$_\mathrm{a}$'s prediction is disproven by ART$_\mathrm{b}$'s, the map field triggers a match tracking mechanism in which ART$_\mathrm{a}$'s resonant category is inhibited, the baseline vigilance is temporarily changed and the search process restarts, causing ART$_\mathrm{a}$ to select another category. Therefore, the map field is a critic, i.e., its purpose is to assess the quality of the mapping between both ART modules and the necessity of adding a new node based on a supervised signal. By engaging the match tracking mechanism, ARTMAP trades generalization for specificity to decrease training error.

Often, ART$_\mathrm{b}$ is omitted and an N_b-dimensional vector of labels is used in its place (since ART$_\mathrm{b}$'s vigilance parameter would typically be set to 1, which would correspond to the number of categories being equal to the number of classes). Moreover, ART$_\mathrm{a}$'s baseline vigilance parameter, which controls the granularity of the input space, is usually set to a small value since this correlates with improved generalization capabilities and a higher level of compression, i.e., network complexity. During inference (or testing), supervised ART models usually operate in feedforward mode, in which resonance and learning are disabled. Algorithm 2 summarizes the general dynamics of an elementary ARTMAP model.

3.1.1. ARTMAP. The first adaptive resonance theory supervised predictive mapping (predictive ART or ARTMAP) model (Carpenter et al., 1991a) consists of two binary ART 1 modules (Sec. 2.1.1), ART$_\mathrm{a}$ and ART$_\mathrm{b}$, connected via an inter-ART associative memory, namely the map field F$^\mathrm{ab}$. The latter performs multidimensional mappings between the binary input samples clustered by modules A and B. Moreover, when the input of a module is missing, it can be recalled by such associative memory. The map field LTM $\boldsymbol{\theta}^{ab}$ is represented by a matrix $\boldsymbol{W}^{ab} = [w_{ij}^{ab}]_{N_a \times N_b}$ such that $w_{ij}^{ab} = 1$ if there is an association between category i of ART$_\mathrm{a}$ and category j of ART$_\mathrm{b}$ and zero otherwise. N_a and N_b are the

Algorithm 2: Elementary ARTMAP algorithm.

```
Input  : $\{\boldsymbol{x}^a, \boldsymbol{x}^b\}$, {ART$_\mathrm{a}$ and ART$_\mathrm{b}$ parameters}, $\{\boldsymbol{\beta}_{ab}, \boldsymbol{\gamma}_{ab}, \boldsymbol{\rho}_{ab}, \boldsymbol{\lambda}_{ab}\}$ (map field
         parameters).
Output : $\boldsymbol{y}^{(F^{ab})}$ (map field activity).

/* Notation                                                        */
$C_l$:  set of ART$_l$ nodes ($l = a, b$).
$\boldsymbol{\theta}^{ab}$:  map field LTM unit.
$\boldsymbol{\beta}_{ab}$:  map field learning function parameter(s).
$\boldsymbol{\gamma}_{ab}$:  map field match function parameter(s).
$\boldsymbol{\rho}_{ab}$:  map field vigilance parameter(s).
$\boldsymbol{\lambda}_{ab}$:  map field initialization parameter(s).
$\bar{\boldsymbol{\rho}}_a$:  ART$_\mathrm{a}$'s baseline vigilance parameter(s).
$f_M^{ab}(\cdot)$:  map field match function.
$f_L^{ab}(\cdot)$:  map field learning function.
$f_V^{ab}(\cdot)$:  map field vigilance function.
$f_N^{ab}(\cdot)$:  map field initialization function.
$f_I^{ab}(\cdot)$:  map field inference function.
$f_{MT}(\cdot)$:  match tracking function.
/* Training                                                        */
1  Present input $\boldsymbol{x}^b \in \boldsymbol{X}^b$ to ART$_\mathrm{b}$.
2  Perform the dynamics of ART$_\mathrm{b}$ (Alg. 1).
3  Present input $\boldsymbol{x}^a \in \boldsymbol{X}^a$ to ART$_\mathrm{a}$.
4  Perform the dynamics of ART$_\mathrm{a}$ (Alg. 1).
5  Compute the map field's match function:
   $M_J^{ab} = f_M^{ab}(\boldsymbol{y}^{(F_2^b)}, \boldsymbol{\theta}_J^{ab}, \boldsymbol{\gamma}_{ab})$.
6  Perform the map field vigilance test: $V_J = f_V^{ab} = \mathbb{1}_{VR_J^{ab}}(\boldsymbol{x}^a)$.
7  if $V_J$ is TRUE then
8      Update ART$_\mathrm{a}$'s and ART$_\mathrm{b}$'s resonant categories $J$ and $K$, respectively (Alg. 1).
9      if ART$_a$ OR ART$_b$ created a new node then
10         $\theta_{\cdot,\cdot}^{ab} = f_N^{ab}(J, K, \boldsymbol{\lambda}_{ab})$.
11     else
12         Update the map field: $\theta_J^{ab}(new) = f_L^{ab}(\boldsymbol{y}^{(F_2^b)}, \boldsymbol{\theta}_J^{ab}(old), \boldsymbol{\beta}_{ab})$.
13 else
14     Inhibit ART$_\mathrm{a}$'s resonant category $J$.
15     Trigger ART$_\mathrm{a}$'s match tracking mechanism: $\boldsymbol{\rho}_a(new) = f_{MT}(\boldsymbol{\rho}_a(old))$
16     Go to step 4.
17 Reset ART$_\mathrm{a}$'s vigilance parameter(s) to baseline value(s):
   $\boldsymbol{\rho}_a(new) = \bar{\boldsymbol{\rho}}_a$.
18 Go to step 1.
```

Algorithm 2 (cont.): Elementary ARTMAP algorithm.

```
   /* Inference                                                    */
1  Present input $\boldsymbol{x}^a \in \boldsymbol{X}^a$ to ART$_\text{a}$. ;
2  Perform the dynamics of ART$_\text{a}$ (Alg. 1). ;
3  Compute the degree of association to each ART$_\text{b}$ node $k$ according to ART$_\text{a}$'s
     activity(s): ;
   $\sigma_k = f_I^{ab}(\boldsymbol{y}^{F_2^a}, \boldsymbol{\theta}^{ab})$. ;
4  Set output: ;
   $y_j^{(F^{ab})} = \begin{cases} 1, & \text{if } j = \arg\max_k(\sigma_k) \\ 0, & \text{otherwise} \end{cases}$.;
5  Go to step 1.;
```

number of nodes in ART$_\text{a}$ and ART$_\text{b}$, respectively. The matrix $\boldsymbol{W}^{ab}$ is initialized as $\vec{\mathbf{1}}$ (i.e., the row vector $\boldsymbol{w}_1^{ab} = \vec{\mathbf{1}}$). The bottom-up and top-down weight vectors of both ART 1's are initialized as described in Sec. 2.1.1.

Training. The map field F$^\text{ab}$ activity is defined as

$$\boldsymbol{y}^{(F^{ab})} = \begin{cases} \boldsymbol{y}^{(F_2^b)} \cap \boldsymbol{w}_J^{ab}, & \text{if both ARTs are active (training)} \\ \boldsymbol{w}_J^{ab}, & \text{if only ART}_\text{a} \text{ is active (prediction)} \\ \boldsymbol{y}^{(F_2^b)}, & \text{if only ART}_\text{b} \text{ is active} \\ \vec{\mathbf{0}}, & \text{otherwise} \end{cases}, \tag{96}$$

where $\boldsymbol{w}_J^{ab} = (w_{J1}, ..., w_{JN_b})$ is the J^{th} row of $\boldsymbol{W}^{ab}$, which is associated with ART$_\text{a}$'s resonant category J.

After resonant nodes for both ART modules have been selected following the presentation of a sample pair $(\boldsymbol{x}^a, \boldsymbol{x}^b)$, the map field match function is computed as

$$M_J^{ab} = \frac{\|\boldsymbol{y}^{(F^{ab})}\|_1}{\|\boldsymbol{y}^{(F_2^b)}\|_1} = \frac{\|\boldsymbol{y}^{(F_2^b)} \cap \boldsymbol{w}_J^{ab}\|_1}{\|\boldsymbol{y}^{(F_2^b)}\|_1}, \tag{97}$$

where the vigilance test is satisfied if $M_J^{ab} \geq \rho_{ab}$. During training, if ART$_\mathrm{a}$'s prediction is correct (i.e., confirmed by ART$_\mathrm{b}$'s supervised signal feedback), all three modules learn. Otherwise, a match tracking mechanism (MT+) is engaged, such that ART$_\mathrm{a}$'s vigilance parameter is temporarily raised by an amount small enough to inhibit the resonant category

$$\rho_a = M_J^a + \epsilon, \ 0 < \epsilon \ll 1, \tag{98}$$

and the search process restarts. Either another resonant category is found or a new one is created, and the vigilance returns to its baseline value ($\rho_a = \bar{\rho}_a$) upon the presentation of a new input pair. Complement coding is usually employed to avoid cases in which ART$_\mathrm{a}$'s vigilance is raised to a value greater than one.

Now consider that the resonant categories of ART$_\mathrm{a}$ and ART$_\mathrm{b}$ are J and K, respectively. When the map field vigilance test is satisfied ($M_J^{ab} \geq \rho_{ab}$), then ART$_\mathrm{a}$ and ART$_\mathrm{b}$ are updated as described in Sec. 2.1.1, and the map field weight vector associated with category J is updated as

$$w_{Jk}^{ab}(new) = \boldsymbol{y}^{(F_2^b)} \cap \boldsymbol{w}_J^{ab}(old) = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{otherwise} \end{cases}, \tag{99}$$

such that it becomes permanently associated with ART$_\mathrm{b}$'s category K. The F_1^a, F_2^a and F^ab layers may be viewed as input, hidden and output layers, respectively.

Inference. In prediction mode, it is sufficient to track the map field's weight vector $\boldsymbol{w}_J^{ab}$ and set it as the systems' output, i.e., when an ART$_\mathrm{a}$'s resonant category J is found, the predicted class K is obtained as

$$K = \arg\max_k \left(\sigma_k\right), \tag{100}$$

where

$$\sigma_k = \sum_{j=1}^{N_a} w_{jk}^{ab} y_j^{(F_2^a)}, \tag{101}$$

such that $\boldsymbol{\sigma} = [\sigma_1, ..., \sigma_{N_b}]^T = \boldsymbol{W}^{ab\,T} \boldsymbol{y}^{(F_2^a)}$.

Properties pertaining to ARTMAP learning are investigated in (Georgiopoulos et al., 1994). A simplified ARTMAP version, namely the simple ARTMAP (Serrano-Gotarredona et al., 1998), replaces ART$_\mathrm{b}$ (and thus its F$_2^\mathrm{b}$ activity $\boldsymbol{y}^{(F_2^b)}$) with a binary vector $\boldsymbol{y}^b$ indicating the class membership of the input sample $\boldsymbol{x}^a$ (i.e., $y_k^b = 1$ if $\boldsymbol{x}^a$ belongs to class k, and $y_i^b = 0 \; \forall i \neq k$).

3.1.2. Fuzzy ARTMAP. Fuzzy ARTMAP (Carpenter et al., 1992) is to ARTMAP what fuzzy ART is to ART 1: it extends the capabilities of ARTMAP to enable the processing of real-valued data by replacing logical with fuzzy AND intersection. Thus, fuzzy ARTMAP also consists of two fuzzy ART modules, ART$_\mathrm{a}$ and ART$_\mathrm{b}$, connected by a map field F$^\mathrm{ab}$ that maps the categories of one ART to another via a matrix of weights $\boldsymbol{\theta}^{ab} = \{\boldsymbol{W}^{ab}\}$, as described in Sec. 3.1.1.

Training. The map field F$^\mathrm{ab}$ activity is defined as

$$\boldsymbol{y}^{(F^{ab})} = \begin{cases} \boldsymbol{y}^{(F_2^b)} \wedge \boldsymbol{w}_J^{ab}, & \text{if both ARTs are active (training)} \\ \boldsymbol{w}_J^{ab}, & \text{if only ART}_\mathrm{a} \text{ is active (prediction)} \\ \boldsymbol{y}^{(F_2^b)}, & \text{if only ART}_\mathrm{b} \text{ is active} \\ \vec{\boldsymbol{0}}, & \text{otherwise} \end{cases} . \tag{102}$$

During training, ART$_\mathrm{a}$ and ART$_\mathrm{b}$ perform their dynamics (Sec. 2.2) simultaneously and independently, with their respective inputs, until both establish resonant nodes J and K, respectively. Then, the map field computes its activity vector using these two pieces of information, as defined in Eq. (102). Next, a second (map field) vigilance test is performed to assess the mapping correctness using

$$M_J^{ab} = \frac{\|\boldsymbol{y}^{(F^{ab})}\|_1}{\|\boldsymbol{y}^{(F_2^b)}\|_1} = \frac{\|\boldsymbol{y}^{(F_2^b)} \wedge \boldsymbol{w}_J^{ab}\|_1}{\|\boldsymbol{y}^{(F_2^b)}\|_1}, \tag{103}$$

and, if it satisfies $M_J^{ab} \geq \rho_{ab}$ ($\rho_{ab} \in [0, 1]$), then learning takes place. Otherwise, in response to a mismatch, the match tracking mechanism (M+) is triggered: the current resonant category J is inhibited (lateral reset), ART$_\text{a}$'s vigilance parameter is raised by a small constant (Eq. (98)), and the search continues with the remaining nodes until a resonant category that satisfies both ρ_a and ρ_{ab} is either found or created. Finally, ρ_a is reset to its baseline value $\rho_a = \bar{\rho}_a$ for the presentation of the following sample. The study in (Anagnostopoulos & Georgiopoulos, 2003), however, indicates that not using match tracking (MT+) reduces the computational burden and model complexity while improving generalization capabilities (Andonie & Sasu, 2006).

In both fuzzy ART modules learning ensues as described in Sec. 2.2, whereas the map field's parameters are updated such that a permanent association is made between the active nodes of ART$_\text{a}$ and ART$_\text{b}$

$$w_{Jk}^{ab}(new) = \boldsymbol{y}^{(F_2^b)} \wedge \boldsymbol{w}_J^{ab}(old) = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{otherwise} \end{cases}. \tag{104}$$

Note that uncommitted nodes participate in the WTA competition. They are initialized as $\vec{\mathbf{1}}$, and the ones belonging to ART$_\text{a}$ are mapped to all ART$_\text{b}$ nodes. A slow-learning mode was introduced in (Carpenter et al., 1995):

$$\boldsymbol{w}_J^{ab}(new) = (1 - \beta_{ab})\, \boldsymbol{w}_J^{ab}(old) + \beta_{ab} \left[\boldsymbol{y}^{(F_2^b)} \wedge \boldsymbol{w}_J^{ab}(old)\right], \tag{105}$$

where β_{ab} is the map field's learning rate, and the conditional probability $p(c_K^b|c_J^a)$ can be estimated nonparametrically as

$$\hat{p}(c_K^b|c_J^a) = \frac{w_{JK}^{ab}}{\sum\limits_{i=1}^{N_b} w_{Ji}^{ab}}. \tag{106}$$

Inference. In testing mode only ART$_\mathrm{a}$ is active. Its output is used to make a prediction and concretely retrieve the labels from ART$_\mathrm{b}$ via the F$^\mathrm{ab}$'s weight matrix (Eqs. (100) and (101)). Note that training, prediction/inference and learning are all WTA (based on a single category).

The simplified fuzzy ARTMAP (Kasuba, 1993) is a simplification of the original fuzzy ARTMAP specifically devised for classification tasks, in which, like simple ARTMAP (Sec. 3.1.1), ART$_\mathrm{b}$ is replaced by vectors indicating the class labels. Another simplified design is discussed in (Vakil-Baghmisheh & Pavešić, 2003).

3.1.3. Fuzzy Min-Max. Fuzzy Min-Max (Simpson, 1992) is a supervised learning neural network classifier that uses fuzzy sets for its internal categories, like its clustering counterpart (Sec. 2.1.4). It is composed of three layers of neurons: an input layer F$_\mathrm{A}$, a layer of hyperbox nodes F$_\mathrm{B}$ and a layer of class nodes F$_\mathrm{C}$. The hyperbox fuzzy sets are adjusted using an expansion-and-contraction-based fuzzy min-max classification learning algorithm that adjusts the fuzzy associations between the inputs and classes. It accomplishes that by identifying which hyperbox to expand for each input and expanding it accordingly. Then, it identifies any resulting overlap between hyperboxes of different classes and minimally adjusts these hyperboxes to eliminate the overlap.

3.1.4. Fusion ARTMAP. Fusion ARTMAP (Asfour et al., 1993) is a modular neural network model designed to classify data originating from multiple sources (i.e., to perform sensor fusion). It generalizes fuzzy ARTMAP (Sec. 3.1.2) by incorporating multiple ART modules, one for each sensor. The outputs of these local ART modules are fed to a fuzzy ARTMAP, specifically, to the latter's ART$_\mathrm{a}$ module, since ART$_\mathrm{b}$ receives the class labels. Another key feature of fusion ARTMAP is the parallel match tracking. Following an incorrect prediction, the vigilance parameter of each ART module is raised (individual ARTs and fuzzy ARTMAP's ART$_a$)

$$\rho_k = \bar{\rho}_k + \Delta\rho, \ \forall k, \tag{107}$$

$$\Delta\rho = \left(M_J^n - \bar{\rho}_n\right) + \epsilon, \tag{108}$$

$$n = \arg\min_k \left(M_J^k\right), \tag{109}$$

where ρ_k and $\bar{\rho}_k$ are the vigilance and baseline vigilance of ART module k, respectively. Each ART module can have its own baseline vigilance parameter, or the entire fusion ARTMAP system can have a single common baseline vigilance. The variable M_J^k is the match function value of ART module k's category J. Note that ART module n yielded the smallest match value and is therefore deemed the least predictive.

The vigilance values of the local ART modules and fuzzy ARTMAP's ART$_a$ are increased by the same value, which is enough to promote a mismatch in ART module n. Therefore, the latter is forced to promote a new search, while the other modules maintain their output. This procedure enables credit assignment to specific modules instead of uniformly blaming all modules regardless of their predictive power. Fusion ARTMAP improves memory compression (compared to single-ART module systems that concatenate all sensor data into a single large vector) given the sharing of the local ART's weight vectors across fuzzy ARTMAP.

The generalized symmetric fusion ARTMAP (Asfour et al., 1993) replaces fuzzy ARTMAP with a global ART module that receives the outputs of all local ART modules and is responsible for the decision-making process. This model can handle multiple input sensors and multiple supervised inputs. In cases consisting of only one supervised input, the functionality is reduced to fusion ARTMAP.

3.1.5. LAPART. The LAPART 1 (Healy et al., 1993) and LAPART 2 (Healy & Caudell, 1998) neural networks are two ART-based logic inference and supervised learning architectures. The LAPART 1 architecture uses two ART 1 networks, A and B, to learn logic inference and association, wherein if network A assigns its input sample to a category, that results in network B assigning its input to the corresponding category. It then uses the learned inference associations between the two networks to test hypotheses and classification decisions. The LAPART 2 algorithm uses the same architecture but introduces a lateral reset procedure and builds a rule extraction network that was shown to converge in two passes through the training data.

3.1.6. ART-EMAP. Adaptive resonance theory with spatial and temporal evidence integration (ART-EMAP) (Carpenter & Ross, 1995) augments fuzzy ARTMAP with a number of features to manage noisy or ambiguous data: distributed representation during inference, integration of spatial-time information, extension of the map field into a multiple field EMAP module and a fine-tuning unsupervised learning stage.

Training. ART-EMAP training is identical to fuzzy ARTMAP's (Sec. 3.1.2).

Inference. ART-EMAP introduces two contrast enhancement procedures for distributed activation: the normalized power rule defined as

$$y_j^{(F_2^a)} = \frac{(T_j^a)^p}{\sum\limits_{i=1}^{N_a} (T_i^a)^p}, \quad p > 1, \tag{110}$$

and the threshold rule

$$y_j^{(F_2^a)} = \frac{[T_j^a - T]^+}{\sum\limits_{i=1}^{N_a} [T_i^a - T]^+}, \tag{111}$$

where T is a threshold parameter, and $[\xi]^+ = \max\{0, \xi\}$ is a rectifier operation. The activity of the first map field F_1^{ab} is then defined as

$$\boldsymbol{y}^{(F_1^{ab})} = \boldsymbol{S}^{ab}, \tag{112}$$

where

$$S_k^{ab} = \sum_{j=1}^{N_a} w_{jk}^{ab} y_j^{(F_2^a)}. \tag{113}$$

A class is predicted using such distributed representation via the second map field activity F_2^{ab}

$$y_k^{(F_2^{ab})} = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{otherwise} \end{cases}, \tag{114}$$

where

$$K = \arg\max_k \left[y_k^{(F_1^{ab})} \right]. \tag{115}$$

To address ambiguity (i.e., categories with similar activation values), the F_2^{ab} activity can be redefined as:

$$y_k^{(F_2^{ab})} = \begin{cases} 1, & \text{if } y_k^{(F_1^{ab})} > (DC) y_j^{(F_1^{ab})} \ \ \forall j \neq k \\ 0, & \text{otherwise} \end{cases} \tag{116}$$

where $DC \geq 1$ is a decision criterion. While $\boldsymbol{y}_k^{(F_2^{ab})} = \vec{\boldsymbol{0}}$, the system waits for another input (i.e., data samples from the same, yet unknown, class) until the inequality in Eq. (116) is satisfied. Moreover, the power rule can also be applied to the F_1^{ab} activity

$$y_k^{(F_1^{ab})} = \frac{(S_k^{ab})^q}{\sum_{i=1}^{N_b} (S_i^{ab})^q}, \ q > 1, \tag{117}$$

where the q is the power parameter.

To handle noisy environments, ART-EMAP uses a map evidence accumulation field F_E^{ab} that combines information from multiple F_1^{ab} activities over time:

$$T_k^{ab}(new) = T_k^{ab}(old) + y_k^{(F_1^{ab})}, \tag{118}$$

where T_k^{ab} is the evidence accumulating MTM. It is initialized as zero ($\boldsymbol{T}^{ab} = \vec{\mathbf{0}}$) and reset once the DC is satisfied. The F_2^{ab} activity can then be redefined as

$$y_k^{(F_2^{ab})} = \begin{cases} 1, & \text{if } T_k > (DC)T_j \;\; \forall j \neq k \\ 0, & \text{otherwise} \end{cases}, \tag{119}$$

where improved accuracy correlates with larger DC values and a greater number of samples (Carpenter & Ross, 1995).

Finally, to learn from the samples used to disambiguate prediction, an unsupervised learning stage ("rehearsal") takes place. In this fine-tuning stage, the LTMs of $\mathrm{ART_a}$, $\mathrm{ART_b}$ and the map field maintain their values, whereas another set of weights from $\mathrm{F_2^a}$ to $\mathrm{F_E^{ab}}$ is adapted when such samples are re-presented to the system.

3.1.7. Adaptive Resonance Associative Map. The fuzzy adaptive resonance associative map (ARAM) (Tan, 1995) extends ART autoassociative to heteroassociative mappings by connecting two ARTs (A and B) via a common category representation field $\mathrm{F_2}$.

LTM. Fuzzy ARAM has two $\mathrm{F_1}$ layers connected to a single $\mathrm{F_2}$ layer whose LTM unit is $\boldsymbol{\theta} = \{\boldsymbol{w} = [\boldsymbol{w}^a, \boldsymbol{w}^b]\}$.

Activation. When normalized and complement coded inputs ($\boldsymbol{x} = [\boldsymbol{x}^a, \boldsymbol{x}^b]$) are presented, the activation function is computed as

$$T_j = \gamma \frac{|\boldsymbol{x}^a \wedge \boldsymbol{w}_j^a|}{\alpha_a + |\boldsymbol{w}_j^a|} + (1-\gamma)\frac{|\boldsymbol{x}^b \wedge \boldsymbol{w}_j^b|}{\alpha_b + |\boldsymbol{w}_j^b|}, \tag{120}$$

where $\gamma \in [0,1]$ is the contribution parameter. Note that there is an independent set of parameters for each module: choice parameters $\alpha_m > 0$, learning parameters $\beta_m \in [0,1]$ and vigilance parameters $\rho_m \in [0,1]$, where $m \in \{a, b\}$.

<u>Match and resonance.</u> Consider that node J has been selected via a WTA competition. F$_1$ and F$_2$ activities are defined as:

$$y_j^{(F_1^m)} = \begin{cases} \boldsymbol{x}^m, & \text{if } F_2^m \text{ is inactive} \\ \boldsymbol{x}^m \wedge \boldsymbol{w}_J^m, & \text{otherwise} \end{cases}, \tag{121}$$

where $m \in \{a, b\}$, and

$$y_j^{(F_2)} = \begin{cases} 1, & \text{if } j = J \\ 0, & \text{otherwise} \end{cases}. \tag{122}$$

The match functions are computed for node J as

$$M_J^m = \frac{\|\boldsymbol{y}^{(F_1^m)}\|_1}{\|\boldsymbol{x}^m\|_1} = \frac{\|\boldsymbol{x}^m \wedge \boldsymbol{w}_J^m\|_1}{\|\boldsymbol{x}^m\|_1}, \tag{123}$$

and resonance occurs if $M_J^m \geq \rho_m$ for both $m \in \{a, b\}$ simultaneously. Thus, $VR_J = \{[\boldsymbol{x}^a, \boldsymbol{x}^b] : M_J^a(\boldsymbol{x}^a) \geq \rho_a \text{ and } M_J^b(\boldsymbol{x}^b) \geq \rho_b\}$. In this case, learning ensues such that the weights $\boldsymbol{w}_J^m$ are updated using fuzzy ART's learning rule (Eq. (21) in Sec. 2.2). Otherwise, a match tracking mechanism temporarily raises the baseline $\bar{\rho}_a$ (which is reset at the start of each sample presentation) as in fuzzy ARTMAP (Sec. 3.1.2), and the search for another resonant category continues. If an uncommitted category is recruited, then another one is initialized as $\boldsymbol{w}^m = \vec{\mathbf{1}}$. Specifically, when such dynamics take place and $\gamma = 1$, fuzzy ARAM is functionally equivalent to fuzzy ARTMAP (Tan, 1995).

3.1.8. Gaussian ARTMAP. The Gaussian ARTMAP (Williamson, 1996) is a discriminative model (Vigdor & Lerner, 2007) that uses Gaussian ART elementary units (Sec. 2.1.6) as building blocks.

Training. Training follows the standard ARTMAP dynamics (Sec. 3.1.1), where the match tracking mechanism is triggered following a predictive error.

Inference. During testing mode, predictions are made considering the total probability of each class, i.e., by using Eqs. (100) and (101) with $y_j^{(F_2^a)} = T_j^a$ (Eq. (32)).

3.1.9. Probabilistic Fuzzy ARTMAP. The probabilistic fuzzy ARTMAP (Lim & Harrison, 1997a, 2000a) combines fuzzy ARTMAP's code compression ability (Sec. 3.1.2) with the probability density function estimation of probabilistic neural networks (PNN) (Specht, 1990) in a hybrid system: during training, a fuzzy ARTMAP variant is used to generate prototypes in a supervised manner, whereas during inference, the PNN uses Bayes decision theory to make predictions.

Training. Training is similar to fuzzy ARTMAP, except for the following:

1. Map field dynamics: the activity of $\mathrm{F^{ab}}$ used to compute the match function (Eq. (103) in Sec. 3.1.2) is defined as

$$\boldsymbol{y}^{(F^{ab})} = \boldsymbol{y}^{(F_2^b)} \wedge \frac{\boldsymbol{w}_J^{ab}}{\|\boldsymbol{w}_J^{ab}\|_1}, \tag{124}$$

and when learning ensues, $\boldsymbol{W}^{ab}$ is updated using

$$\boldsymbol{w}_J^{ab}(new) = \boldsymbol{w}_J^{ab}(old) + \boldsymbol{y}^{(F^{ab})}; \tag{125}$$

2. If the match tracking mechanism is engaged, then the condition

$$0 \leq \rho_a \leq \min\left(1, M_J^a + \epsilon\right), \; 0 < \epsilon \ll 1, \tag{126}$$

is enforced to enable identical categories to be associated with different classes (Lim & Harrison, 1997b);

3. Centroids $\boldsymbol{\mu}_j^a$ are embedded in ART$_\mathrm{a}$ (i.e., the LTM unit is $\boldsymbol{\theta}^a = \{\boldsymbol{w}^a, \boldsymbol{\mu}^a\}$). These are initialized as $\boldsymbol{\mu}_j^a = \vec{\boldsymbol{0}}$ and recursively estimated using

$$\boldsymbol{\mu}_j^a(new) = \boldsymbol{\mu}_j^a(old) + \frac{1}{\|\boldsymbol{w}_J^{ab}\|_1}\left(\boldsymbol{x}^a - \boldsymbol{\mu}_j^a(old)\right), \tag{127}$$

where $\boldsymbol{x}^a$ is complement-coded for fuzzy ARTMAP categories $\boldsymbol{w}$ but not for the centroids $\boldsymbol{\mu}$.

Inference. Prediction is accomplished using the maximum a posteriori (MAP) or minimum-risk estimate:

$$\hat{p}(c_k^b|\boldsymbol{x}^a) = \hat{p}(\boldsymbol{x}^a|c_k^b)\hat{p}(c_k^b)l(c_{jk}), \tag{128}$$

where $l(c_{jk})$ represents the cost of selecting c_k^b when the true class is c_j^b. The prior probability estimate of a given class k is given by the ratio of the number of samples encoded by ART$_\mathrm{a}$'s prototypes that are mapped to class k to the total number of samples presented to the probabilistic fuzzy ARTMAP:

$$\hat{p}(c_k^b) = \frac{\sum\limits_{j=1}^{N_a} w_{jk}^{ab}}{\sum\limits_{k=1}^{N_b}\sum\limits_{j=1}^{N_a} w_{jk}^{ab}}, \tag{129}$$

and $p(\boldsymbol{x}^a|c_k^b)$ is estimated using the Parzen-window method (Cacoullos, 1966; Parzen, 1962) with isotopic Gaussians kernels ($\boldsymbol{\Sigma}_j = \sigma_j^2\boldsymbol{I}$)

$$\hat{p}(\boldsymbol{x}^a|c_k^b) = \sum_{j=1}^{N_a} \frac{\mathbb{1}_{c_k^b}(\boldsymbol{\mu}_j^a)}{\sum\limits_{i=1}^{N_a} \mathbb{1}_{c_k^b}(\boldsymbol{\mu}_i^a)} \frac{e^{\left(-\frac{\|\boldsymbol{x}^a-\boldsymbol{\mu}_j^a\|_2^2}{2\sigma_j^2}\right)}}{(2\pi)^{\frac{d}{2}}\,\sigma_j^d}, \tag{130}$$

where

$$\mathbb{1}_{c_k^b}(\boldsymbol{\mu}_j^a) = \begin{cases} 1, & \text{if } \boldsymbol{\mu}_j^a \in c_k^b \\ 0, & \text{otherwise} \end{cases}. \tag{131}$$

The kernels used for the realization of the Parzen-window density estimation have heteroscedastic components, which are computed as

$$\sigma_j = \frac{1}{r} \min_i \|\boldsymbol{\mu}_j^a - \boldsymbol{\mu}_i^a\|_2, \tag{132}$$

or determined using the k-nearest neighbors method (Duda et al., 2000)

$$\sigma_j = \frac{1}{k} \sum_{i=1}^{k} \|\boldsymbol{\mu}_j^a - \boldsymbol{\mu}_i^a\|, \; 1 \leq k \leq N_a - 1, \tag{133}$$

where r is a user-defined overlapping parameter, and $\boldsymbol{\mu}_j^a$ and $\boldsymbol{\mu}_i^a$ belong to different classes in Eqs. (132) and (133).

3.1.10. ARTMAP-IC. The ARTMAP-IC model (Carpenter & Markuzon, 1998) is a fuzzy ARTMAP variant whose key characteristics are (i) a new match tracking mechanism (MT-) to reduce model complexity and handle "inconsistent cases" (IC) and (ii) the inclusion of instance counting (via a new counting field F$_3$) for probabilistic distributed prediction. "Inconsistent cases" are scenarios in which identical samples pair with different classes.

ARTMAP-IC replaces ART$_\text{b}$ with a vector $\boldsymbol{y}^b$ encoding the classes of the classification problem, such that, for a given input $\boldsymbol{x}^a$ presented to ART$_\text{a}$,

$$y_i^b = \begin{cases} 1, & \text{if } \boldsymbol{x}^a \in \text{ class } i \\ 0, & \text{otherwise} \end{cases}. \tag{134}$$

The activity of the counting field F_3 (located in-between $\mathrm{ART_a}$ and F^{ab}) is defined as

$$y_j^{(F_3)} = \begin{cases} y_j^{(F_2^a)}, & \text{training} \\ \dfrac{c_j y_j^{(F_2^a)}}{\sum_{i=1}^{N_a} c_i y_i^{(F_2^a)}}, & \text{prediction} \end{cases}, \tag{135}$$

where the instance counting weight c_j records the number of samples that are encoded by category j, i.e., the number of times it is activated. The map field F^{ab} activity can then be defined as

$$\boldsymbol{y}^{(F^{ab})} = \begin{cases} \boldsymbol{y}^b \wedge \boldsymbol{U}, & \text{training} \\ \boldsymbol{U}, & \text{prediction} \end{cases} \tag{136}$$

where the kth component of the map field's input is

$$U_k = \sum_{j=1}^{Na} w_{jk}^{ab} y_j^{(F_3)}, k = 1, ..., N_b, \tag{137}$$

and here N_b represents the number of classes.

Training. During training, the match function is defined as

$$M_J^{ab} = \frac{\|\boldsymbol{y}^b \wedge \boldsymbol{U}\|_1}{\|\boldsymbol{y}^b\|_1} = \|\boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab}\|_1, \tag{138}$$

since $\boldsymbol{U} = \boldsymbol{w}_J^{ab}$ (because $\boldsymbol{y}^{(F_2^a)} = \boldsymbol{y}^{(F_3)}$) and $\|\boldsymbol{y}^b\|_1 = 1$. If the vigilance criterion is not satisfied ($M_J^{ab} < \rho_{ab}$), then the new match tracking mechanism (MT-) is engaged such that $\mathrm{ART_a}$'s vigilance is set to

$$\rho_a(new) = M_J^a + \epsilon,\ \epsilon \leq 0 \text{ and } \|\epsilon\| \text{ small}, \tag{139}$$

and the search proceeds as with fuzzy ARTMAP. Otherwise, if learning ensues, then fuzzy ART$_a$ and the map field weight vectors learn as described in Secs. 2.2 and 3.1.2, respectively. The instance counting is updated as

$$c_j(new) = c_j(old) + y_j^{(F_2^a)}, \tag{140}$$

where c_j's are initialized as 0.

Inference. During testing, no search occurs, and ARTMAP-IC uses the Q-max rule to distribute F_2^a activity via the following contrast enhancement procedure:

$$y_j^{(F_2^a)} = \begin{cases} \dfrac{T_j}{\sum\limits_{\lambda \in \Lambda} T_\lambda}, & \text{if } j \in \Lambda \\ 0, & \text{otherwise} \end{cases}, \tag{141}$$

where Λ is the set formed by the Q categories with the largest activation values (Q is a user-defined parameter). This is similar to k-nearest neighbors (Duda et al., 2000) where Q assumes the role of k (Carpenter & Markuzon, 1998). Setting $Q = 1$ leads to WTA mode.

Finally, the probability of class k is then computed as

$$\sigma_k = \frac{U_k}{\sum\limits_{l=1}^{N_b} U_l} = \frac{\sum\limits_{j \in \Lambda} w_{jk}^{ab} c_j T_j}{\sum\limits_{l=1}^{N_b} \sum\limits_{j \in \Lambda} w_{jl}^{ab} c_j T_j}. \tag{142}$$

3.1.11. Distributed ARTMAP. Distributed ARTMAP (dARTMAP) (Carpenter et al., 1998) was developed to improve supervised ART models regarding model compactness and noise robustness (i.e., reduce category proliferation) while performing fast and stable learning via distributed representation. It features distributed activation, match and learning functions. Notably, distributed ARTMAP generalizes the following supervised ART models (Carpenter, 2003): "dARTMAP $\supset$ ARTMAP-IC $\supset$ default ARTMAP $\supset$ fuzzy ARTMAP", where $\supset$ is used to indicate containment considering this ARTMAP's ecosystem.

In case of classification problems, distributed ARTMAP uses distributed ART (Sec. 2.1.5) as a building block for ART_a, while replacing ART_b with a binary vector indicating the input's class membership (Eq. (134) in Sec. 3.1.10). The distributed ARTMAP uses an increased-gradient content-addressable memory (IG CAM) rule for contrast enhancement. A CAM rule defines a function that yields the steady state values of the network's STM when an input sample is presented. Particularly, distributed ARTMAP's CAM rule defines a power function that is controlled by a parameter p. The latter has a role akin to the variance in Gaussian kernels, and, as it tends to infinity, the network converges to WTA.

Training. During training, the distributed ARTMAP alternates between distributed and WTA modes. Like ARTMAP-IC (Sec. 3.1.10), distributed ARTMAP features a counting field F_3^a (for instance counting purposes) which is cascaded to F_2^a and employs the MT- match tracking search algorithm. Briefly, the distributed representation undergoes the unsupervised (Eqs. (25) to (27)) and supervised vigilance (i.e., prediction assessment) tests, and if one of them fails the system switches to WTA mode and its corresponding dynamics are carried out (in which nodes can be added incrementally). Otherwise, distributed mode dynamics take place.

Particularly, the distributed ARTMAP uses the distributed choice-by-difference activation function (Eq. (22) in Sec. 2.1.5 disregarding the depletion parameters)

$$T_j = \sum_{i=1}^{2d} \left[x_i^a \wedge (1 - \tau_i^{bu}) \right] + (1 - \alpha) \sum_{i=1}^{2d} \tau_i^{bu}, \ \alpha \in (0, 1), \tag{143}$$

and, after these are computed, the following subsets of highly active nodes are considered:

1. $\Lambda = \{j : T_j \geq T^u\}$
2. $\Lambda' = \{j : T_j = (2 - \alpha)d\}$

where T^u is the activation function of an uncommitted node ($\boldsymbol{\tau}^{bu} = \boldsymbol{\tau}^{td} = \vec{\mathbf{0}}$). The IG CAM rule specifies the following functions for the steady-state activities of distributed ARTMAP's modes:

- Distributed mode
  - If $\Lambda' \neq \{\emptyset\}$, then

$$y_j^{(F_2^a)} = \begin{cases} \dfrac{1}{|\Lambda'|}, & \forall j \in \Lambda' \\ 0, & \text{otherwise} \end{cases}, \tag{144}$$

where $|\cdot|$ represents the cardinality of a set.

  - If $\Lambda' = \{\emptyset\}$ and $\Lambda \neq \{\emptyset\}$, then

$$y_j^{(F_2^a)} = \begin{cases} \dfrac{1}{1 + \sum\limits_{\lambda \in \Lambda, \lambda \neq j} \left[\dfrac{(2-\alpha)d - T_j}{(2-\alpha)d - T_\lambda} \right]^p}, & \forall j \in \Lambda \\ 0, & \text{otherwise} \end{cases}, \tag{145}$$

where $p \in (0, \infty)$ is the power parameter. The ART$_\mathrm{a}$'s counting field F$_3$ activity is then defined as

$$y_j^{(F_3^a)} = \frac{c_j y_j^{(F_2^a)}}{\sum\limits_{\lambda=1}^{C} c_\lambda y_\lambda^{(F_2^a)}}, \tag{146}$$

where C is the number of ART$_\mathrm{a}$'s committed nodes, and c_j is the instance counting of node j (if uncommitted, then $c_j = 0$). The signal used in the ART$_\mathrm{a}$'s match function is then

$$\sigma_i = \sum_{j=1}^{C} \left[y_j^{(F_3^a)} - \tau_{j,i}^{td} \right]^+, \; i = 1, ..., 2d. \tag{147}$$

- WTA mode
  - If $\Lambda \neq \{\emptyset\}$, then the winner node is $J = \underset{j \in \Lambda}{\arg\max}\,(T_j)$.

– If $\Lambda = \{\emptyset\}$, then the uncommitted node is recruited to learn the presented input sample.

The ART$_\text{a}$'s counting field F$_3$ activity is then

$$y_j^{(F_3^a)} = y_j^{(F_2^a)} = \begin{cases} 1, & \text{if } j = J \\ 0, & \text{otherwise} \end{cases}, \tag{148}$$

and the signal used in the ART$_\text{a}$'s match function is

$$\sigma_i = \left(1 - \tau_{J,i}^{td}\right), \; i = 1, ..., 2d. \tag{149}$$

If the vigilance test of ART$_\text{a}$ is not satisfied (Eqs. (25) to (27)) in Sec. 2.1.5), then distributed ARTMAP reverts to WTA mode, and the search continues until a resonant node is either found or created. Finally, the output class is then estimated using Eqs. (100) and (101) with $y_j^{(F_3^a)}$ in place of $y_j^{(F_2)}$. If the prediction is incorrect, then match tracking is engaged using the MT- algorithm (Sec. 3.1.10). Otherwise, ART$_\text{a}$ adapts using the distributed ART learning laws described in Sec. 2.1.5 (the top-down thresholds' components are updated using $y_j^{(F_3^a)}$ in place of $y_j^{(F_2)}$ in Eq. (30)), and the instance countings are updated using Eq. (140) in Sec. 3.1.10.

Note that if the distributed ARTMAP system enters a resonant state while in distributed mode, then, prior to learning, a credit assignment stage takes place in which the nodes permanently associated with the wrong class are inhibited, the F$_2^\text{a}$ activity is renormalized (i.e., $\|\boldsymbol{y}^{(F_2^a)}\|_1 = 1$) and the F$_3^\text{a}$ activity and the signal $\boldsymbol{\sigma}$ are recomputed using Eqs. (146) and (147), respectively.

Inference. To make a prediction for a new sample $\boldsymbol{x}$, distributed ARTMAP operates similarly to the training phase but always in distributed mode and with search and learning disabled (i.e., in feedforward mode).

3.1.12. Hypersphere ARTMAP. Hypersphere ARTMAP (Anagnostopoulos & Georgiopoulos, 2000) closely follows the operation of fuzzy ARTMAP (Sec. 3.1.2) but instead uses hypersphere ART (Sec. 2.1.7) modules for ART_a and ART_b. ART_b is responsible for clustering the classes ($\boldsymbol{x}^b$), ART_a does the data samples ($\boldsymbol{x}^a$) and the inter-ART maps the ART_a categories to the ART_b categories regulated by the match tracking procedure.

3.1.13. Ellipsoid ARTMAP. Similar to hypersphere ARTMAP (Sec.3.1.12), ellipsoid ARTMAP (Anagnostopoulos & Georgiopoulos, 2001a,b) uses ellipsoid ART (Sec. 2.1.8) for both its ART_a and ART_b modules while closely following fuzzy ARTMAP's operation (Sec. 3.1.2). The boosted ellipsoid ARTMAP variant was presented in (Anagnostopoulos et al., 2002a,b) to enhance the generalization capability and reduce category proliferation in the original model, both of which are achieved by allowing for non-zero training error. This variant also incorporates other improvements such as allowing many-to-many mappings and the ability to handle inconsistent cases (see Sec. 3.1.10); the latter would render the original model unstable (Anagnostopoulos et al., 2002a,b). Comparative studies with these and other ARTMAP models can be found in (Anagnostopoulos et al., 2003; Le et al., 2005).

3.1.14. μARTMAP. The μARTMAP model (Gomez-Sanchez et al., 2002; Sanchez et al., 2000) is a fuzzy ARTMAP variant developed to reduce category proliferation due to overlapping classes, consequently improving generalization capability. This is accomplished by regulating the conditional entropy between the input (ART_a) and output (ART_b) spaces

$$H(ART_b|ART_a) = \sum_{j=1}^{N_a} h_j, \tag{150}$$

where h_j is the contribution of ART_a's node j to the total entropy:

$$h_j = -\hat{p}(c_j^a) \sum_{k=1}^{N_b} \hat{p}(c_k^b|c_j^a) \log_2 \hat{p}(c_k^b|c_j^a), \tag{151}$$

and the probabilities are estimated using the map field's LTM unit, whose dynamics are similar to PROBART's (Marriott & Harrison, 1995) (Sec. 3.2.1). This process indirectly controls the training error, which is relaxed to address overfitting.

Training. Training is divided into two phases, and the first one is performed online. Assuming the resonant categories of ART$_\mathrm{a}$ and ART$_\mathrm{b}$ are J and K, respectively, the map field vigilance test is defined using Eq. (151):

$$M_J^{ab} = h_J, \tag{152}$$

where

$$\hat{p}(c_k^b|c_j^a) = \begin{cases} \dfrac{y_k^{(F^{ab})}}{\|\boldsymbol{y}^{(F^{ab})}\|_1}, & \text{if } j = J \\ \dfrac{w_{jk}^{ab}}{\|\boldsymbol{w}_j^{ab}\|_1}, & \text{otherwise} \end{cases}, \tag{153}$$

$$\hat{p}(c_j^a) = \begin{cases} \dfrac{\|\boldsymbol{y}^{(F^{ab})}\|_1}{\|\boldsymbol{y}^{(F^{ab})}\|_1 + \sum\limits_{i=1, i \neq J}^{N_a} \|\boldsymbol{w}_i^{ab}\|_1}, & \text{if } j = J \\ \dfrac{\|\boldsymbol{w}_j^{ab}\|_1}{\|\boldsymbol{y}^{(F^{ab})}\|_1 + \sum\limits_{i=1, i \neq J}^{N_a} \|\boldsymbol{w}_i^{ab}\|_1}, & \text{otherwise} \end{cases}. \tag{154}$$

Note, however, that if J is an uncommitted node, then

$$\hat{p}(c_k^b|c_J^a) = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{otherwise} \end{cases}, \tag{155}$$

which implies $h_J = 0$. The value of h_J measures the homogeneity of ART$_\mathrm{b}$ nodes (i.e., classes) associated with ART$_\mathrm{a}$'s category J. If $M_J^{ab} \leq h_{max}$, where h_{max} is a user-defined parameter, then the map field vigilance is satisfied, and learning ensues as in PROBART

(Eq. (209)). Otherwise, ART$_\text{a}$'s node J is inhibited, and the search continues without changing ART$_\text{a}$'s vigilance parameter. Note that $h_{max} = 0$ implies mapping to a single class, whereas $h_{max} > 0$ allows mapping to different classes (i.e., non-zero training error).

Next, an offline training phase is performed to measure the overlap between categories. In this second training phase no learning is permitted within the ART modules. Probabilities are re-estimated using

$$\hat{p}(c_k^b|c_j^a) = \frac{v_{jk}^{ab}}{\|\boldsymbol{v}_j^{ab}\|_1}, \tag{156}$$

$$\hat{p}(c_j^a) = \frac{\|\boldsymbol{v}_j^{ab}\|_1}{\sum\limits_{i=1}^{N^a}\|\boldsymbol{v}_i^{ab}\|_1}, \tag{157}$$

where a temporary map field co-occurrence matrix $\boldsymbol{V}^{ab}$ is updated in an unsupervised manner, i.e., without match tracking (initialization: $\boldsymbol{V}^{ab} = \boldsymbol{0}$). The total entropy H is computed using Eq. (150), and if $H > H_{max}$, where H_{max} is a user-defined parameter, then the mapping is considered too entropic. ART$_\text{a}$'s category M with the largest contribution h_M is removed, and the baseline vigilance $\bar{\rho}_a$ is increased for all new uncommitted categories as

$$\bar{\rho}_a = \frac{\|\boldsymbol{w}_M^a\|_1}{\|\boldsymbol{x}^a\|_1} + \epsilon, \tag{158}$$

thus adaptively tuning individual vigilance parameters of ART$_\text{a}$'s categories. The samples that were associated with node M are re-presented, and the learning process resumes. This entire process is repeated until $H \leq H_{max}$. Notably, if $h_{max}, H_{max} \geq \log_2 N_b$ then μARTMAP behaves similarly to PROBART, whereas if $h_{max} = 0$ and $H_{max} \geq \log_2 N_b$, then μARTMAP behaves similarly to fuzzy ARTMAP.

Inference. Predictions are made using Eqs. (100) and (101), i.e., the class output K is estimated as the one that has the largest frequency of association with ART$_\text{a}$'s resonant category J.

Under certain conditions, μARTMAP creates large categories that lead to considerable overlaps and decrease the system's performance. The safe-μARTMAP (Gomez-Sanchez et al., 2001) variant is a generalization of μARTMAP that adds another vigilance criterion to mediate learning. Specifically, to avoid the formation of large hyperrectangles that enclose far apart samples belonging to the same class, besides passing both the ART_a and the map field vigilance tests, an ART_a category also needs to undergo a distance criterion defined as

$$M_J^{\Delta w} = \frac{\|\boldsymbol{w}_J^a\|_1 - \|\boldsymbol{w}_J^a \wedge \boldsymbol{x}^a\|_1}{\|\boldsymbol{x}^a\|_1}. \tag{159}$$

Learning only occurs if this third vigilance test is also satisfied ($M_J^{\Delta w} \leq \delta, 0 < \delta < 1 - \rho$). This test imposes a restriction on the instantaneous change of a category size, which is upper bounded by $\|\boldsymbol{x}^a\|_1\delta$. Particularly, safe-$\mu$ARTMAP reduces to μARTMAP when $\delta = 1$ (which effectively implies the absence of a constraint).

3.1.15. Default ARTMAPs. The default ARTMAP 1 model (Carpenter, 2003) is characterized by the usage of a distributed representation to perform continuously valued predictions, as opposed to binary and fuzzy ARTMAP models (Secs. 3.1.1 and 3.1.2), which use WTA code representation.

Training. Default ARTMAP 1's training is akin to fuzzy ARTMAP's, except that (i) ART_b is absent (default ARTMAP 1 is a simplified architecture), (ii) its ART_a module employs the choice-by-difference activation function defined as (Carpenter & Gjaja, 1994)

$$T_j = \|\boldsymbol{x} \wedge \boldsymbol{w}_j^a\|_1 + (1 - \alpha)(d - \|\boldsymbol{w}_j^a\|_1), \alpha \in (0, 1), \tag{160}$$

and (iii) its match tracking algorithm is MT- search (Carpenter & Markuzon, 1998).

Inference. As opposed to fuzzy ARTMAP, default ARTMAP 1 uses a distributed representation for inference, where two subsets of highly active neurons are selected as:

1. $\Lambda = \{\lambda = 1, ..., N_a : T_\lambda > \alpha d\}$

2. $\Lambda' = \{\lambda = 1, ..., N_a : T_\lambda = d\ (i.e.,\ \boldsymbol{w}_\lambda = \boldsymbol{x}^a)\}$

Next, the IG CAM rule is applied:

- If $\Lambda' \neq \{\emptyset\}$, then

$$y_j = \begin{cases} \dfrac{1}{|\Lambda'|}, & \forall j \in \Lambda' \\ 0, & \text{otherwise} \end{cases}, \tag{161}$$

where $|\cdot|$ represents the cardinality of a set.

- If $\Lambda' = \{\emptyset\}$, then

$$y_j = \begin{cases} \dfrac{\left[\dfrac{1}{d - T_j}\right]^p}{\sum\limits_{\lambda\in\Lambda}\left[\dfrac{1}{d - T_\lambda}\right]^p}, & \forall j \in \Lambda \\ 0, & \text{otherwise} \end{cases} . \tag{162}$$

Finally, the predictions for each class are obtained using Eqs. (100) and (101) in Sec. 3.1.1.

In a WTA system, such as fuzzy ARTMAP, after learning a sample, an immediate re-presentation is guaranteed to yield a correct prediction, i.e., it passes the “next-input-test”. However, the default ARTMAP 1 WTA prediction during training might not be the same as the distributed one. To overcome this problem, the default ARTMAP 2 model (Amis & Carpenter, 2007) introduces the “distributed-next-input-test” during training to ensure that a correct prediction would also be performed under a distributed representation. Briefly, in order to anticipate an error, after learning from a sample in a WTA mode, the prediction is verified again using a distributed representation. If the distributed prediction is correct, then learning resumes by returning to WTA mode and presenting the next sample. Otherwise, the match tracking mechanism is engaged, the system reverts to WTA mode, the resonant category is inhibited and the network restarts the search to learn more from that sample.

3.1.16. Boosted ARTMAP. Boosted ARTMAP (Verzi et al., 1998) is a variant of fuzzy ARTMAP (Sec. 3.1.2) closely related to PROBART (Sec. 3.2.1). It is inspired by boosting theory (Schapire, 1990) and was developed to improve the fuzzy ARTMAP's generalization capability (since it is prone to overfitting the training data) and to create less complex networks (i.e., to reduce the type of category proliferation caused by overlapping classes). These are addressed by regulating the training error, which is allowed to be non-zero. Particularly, boosted ARTMAP's ART$_a$ and ART$_b$ modules are boosted ART models (which are identical to fuzzy ART, except that the categories are endowed with individual vigilance parameters), and its map field dynamics are equal to PROBARTs'.

Training. Boosted ARTMAP learning is offline. After a first pass through the data, the error of ART$_a$'s category j is estimated as

$$\varepsilon_j = p_j e_j = \frac{\|\boldsymbol{w}_j^{ab}\|_1 - \max\limits_k \left(w_{jk}^{ab}\right)}{\sum\limits_{m=1}^{N_a} \sum\limits_{n=1}^{N_b} w_{mn}^{ab}}, \tag{163}$$

where

$$p_j = \hat{p}\left(\boldsymbol{x} \text{ selects } c_j^a\right) = \hat{p}(c_j^a) = \frac{\|\boldsymbol{w}_j^{ab}\|_1}{\sum\limits_{m=1}^{N_a} \sum\limits_{n=1}^{N_b} w_{mn}^{ab}}, \tag{164}$$

$$e_j = \hat{p}\left(c^* \text{ not predicted by } c_j^a\right) = 1 - \frac{\max\limits_k \left(w_{jk}^{ab}\right)}{\|\boldsymbol{w}_j^{ab}\|_1}, \tag{165}$$

and the total error is given by

$$\varepsilon_T = \sum_{j=1}^{N_a} \varepsilon_j = \frac{\sum\limits_{j=1}^{N_a} \left[\|\boldsymbol{w}_j^{ab}\|_1 - \max\limits_k \left(w_{jk}^{ab}\right)\right]}{\sum\limits_{m=1}^{N_a} \sum\limits_{n=1}^{N_b} w_{mn}^{ab}}, \tag{166}$$

where c^* is the true class. Then, the vigilance parameters of ART$_\mathrm{a}$'s nodes are raised by a user-defined parameter δ:

$$\rho_\lambda(new) = \rho_\lambda(old) + \delta,\ \lambda \in \Lambda, \tag{167}$$

where $\Lambda = \{\lambda : \varepsilon_\lambda > \varepsilon_{max}\}$, i.e., Λ is the subset of nodes λ with contributions ε_λ to the total error ε_T larger than the desired error ε_{max}. If $\Lambda = \{\emptyset\}$ but the total error ε_T is above the desired error ε_{max} (i.e., if $\varepsilon_T > \varepsilon_{max}$), then the vigilances of all nodes j with the largest contribution ε_j are increased following Eq. (167). Note that when new nodes are added to the system, their initial vigilance parameter is set to a relaxed baseline value $\bar{\rho}$.

Inference. In prediction mode, when a sample is presented, the corresponding class label is obtained using the map field weight vector associated with ART$_\mathrm{a}$'s resonant category J

$$K = \arg\max_k \left[w_{Jk}^{ab}\right]. \tag{168}$$

As discussed in (Gomez-Sanchez et al., 2002), due to the lack of a match tracking mechanism, this version of boosted ARTMAP cannot handle "populated exceptions", i.e., when samples from one class surround another and it is necessary to create a category inside another category. The second version of boosted ARTMAP (Verzi et al., 2006) augments its predecessor with a match tracking mechanism to regulate the training error, whose map field dynamics are discussed next.

Training. During learning, when a sample pair is presented and ART$_\mathrm{a}$'s and ART$_\mathrm{b}$'s resonant nodes are J and K, respectively, the map field match function is given by

$$M_J^{ab} = (1 - e'_J)\frac{\|\boldsymbol{y}^{(F_2^b)} \wedge \boldsymbol{w}_J^{ab'}\|_1}{\|\boldsymbol{y}^{(F_2^b)}\|_1}, \tag{169}$$

and resonance occurs if the winning category satisfies $M_J > (1-\epsilon)\rho_{ab}$, where $\epsilon \in [0,1]$ is the error tolerance parameter that binds the training error. The map field then learns as in PROBART (Eq. (209)). Otherwise, the match tracking mechanism is engaged. The temporary variables e'_J and $\boldsymbol{w}_J^{ab'}$ in Eq. (169) are computed as if category J was allowed to learn:

$$w_{Jl}^{ab'} = \begin{cases} 1, & \text{if } l = \arg\max_k \left(w_{Jk}^{ab}\right) \\ \lceil 0+\epsilon \rceil, & \text{otherwise} \end{cases}, \tag{170}$$

$$e'_J = 1 - \frac{\max_k \left(w_{Jk}^{ab''}\right)}{\|\boldsymbol{w}_J^{ab''}\|_1}, \tag{171}$$

$$w_{Jl}^{ab''} = \begin{cases} w_{Jl}^{ab} + 1, & \text{if } l = K \\ w_{Jl}^{ab}, & \text{otherwise} \end{cases}, \tag{172}$$

where $\lceil \cdot \rceil$ is the ceiling function. If node J is uncommitted, then $\boldsymbol{w}_J^{ab'} = \vec{\mathbf{1}}$ and $e'_J = 0$ (no mismatch will take place).

Inference. Predictions are made using Eq. (168).

Note that boosted ART generalizes fuzzy ART, and boosted ARTMAP reduces in functionality to fuzzy ARTMAP by setting $\varepsilon_d = 0$ and $\rho^{ab} > 0.5$ and to PROBART by setting $\varepsilon_d = 1$. Boosted ARTMAP performs empirical risk minimization, however, variants of boosted ARTMAP, such as (Verzi et al., 2006; Verzi et al., 2002, 2001), perform structural risk minimization and use Rademacher penalization (Koltchinskii, 2001).

3.1.17. Fuzzy ARTMAP with Input Relevances. The fuzzy ARTMAP with input relevances (FAMR) model (Andonie & Sasu, 2003; Andonie & Sasu, 2006; Andonie et al., 2003b) is a fuzzy ARTMAP variant that modifies the map field dynamics, while maintaining the remaining dynamics of fuzzy ARTMAP. Thus, the incremental and non-parametric estimation of posterior probabilities based on the map field is augmented to reflect the degree of importance of incoming samples, especially when these are arriving from multiple heterogeneous sources corrupted by different noise levels.

Training. Particularly, a sample arriving at time $t > 0$ has a relevance factor $q_t \in (0, \infty)$. It is a user-defined or computed parameter, e.g., samples may be ranked based on their source noise level or have their relevance factors made proportional to its importance. Assuming the resonant categories of ART$_\mathrm{a}$ and ART$_\mathrm{b}$ are J and K, respectively, then the map field recursive update equations are based on the stochastic approximation procedure (Andonie, 1990):

$$w_{jk}^{ab}(new) = \begin{cases} w_{jk}^{ab}(old), & j \neq J \\ (1 - A_t)\, w_{jk}^{ab}(old) + A_t, & j = J, k = K \\ (1 - A_t)\, w_{jk}^{ab}(old), & j = J, k \neq K \end{cases}, \tag{173}$$

where

$$A_t = \frac{q_t}{Q_J(new)}, \tag{174}$$

$$Q_J(new) = Q_J(old) + q_t, \tag{175}$$

and $\boldsymbol{Q} = [Q_1 ... Q_{N_a}]$. Thus, an entry $w_{i,j}^{ab}$ of the map field matrix $\boldsymbol{W}^{ab}$ is an estimate of $p(c_k^b | c_k^a)$. If a new category K is created in ART$_\mathrm{b}$, then the map field weights w_{jk}^{ab} are adapted as:

$$w_{jk}^{ab}(new) = \begin{cases} \dfrac{q_0}{N_b(new) Q_j}, & \forall j, k = K \\ w_{jk}^{ab}(old) - \dfrac{w_{jK}^{ab}(new)}{N_b(new) - 1}, & \forall j, k \neq K \end{cases}, \tag{176}$$

where $N_b(new) = N_b(old) + 1$ is the new number of nodes in ART$_\mathrm{b}$. If a new category is created in ART$_\mathrm{a}$ ($J = N_a + 1$), then Q_J is set as $q_0 \geq 0$ (initial relevance parameter) and $w_{Jk}^{ab} = 1/N_b, \forall k$. Finally, the map field's vigilance test is redefined as

$$M_J^{ab} = N_b w_{JK}^{ab}, \tag{177}$$

such that $M_J^{ab} \geq \rho_{ab}$ must be satisfied for resonance to occur.

Inference. Predictions are made similarly to fuzzy ARTMAP (Sec. 3.1.2).

3.1.18. Bayesian ARTMAP. Bayesian ARTMAP (Vigdor & Lerner, 2007) is a generative model based on Bayes' decision theory (Vigdor & Lerner, 2007) that uses Bayesian ART modules (Sec. 2.1.10) as building blocks and represents class density by Gaussian mixtures. Moreover, the posterior probabilities in Bayes' theorem are estimated within and between ART modules.

Training. During training, the map field LTM unit is a matrix of association frequency (sample count) $\boldsymbol{\theta}^{ab} = \{\boldsymbol{N} = [n_{kj}]_{N_b \times N_a}\}$ that is used to estimate the ART$_\mathrm{a}$ and ART$_\mathrm{b}$ joint probability distribution

$$\hat{p}(c_k^b, c_j^a) = \frac{n_{kj}}{\sum\limits_{i=1}^{N_b} \sum\limits_{l=1}^{N_a} n_{il}}, \tag{178}$$

such that soft and hard mappings between ART modules are possible, i.e., a deterministic many-to-one mapping or a probabilistic many-to-many mapping based on $\hat{p}(c_k^b, c_j^a)$. The match tracking mechanism is triggered by the system if the match function value for ART$_\mathrm{a}$'s resonant category J

$$M_J^{ab} = \hat{p}(c_k^b | c_J^a) = \frac{n_{k,J}}{\sum\limits_{i=1}^{N_b} n_{i,J}}, \tag{179}$$

does not satisfy $M_J^{ab} \geq \rho_{ab}$, where ρ_{ab} represents the minimum class posterior probability threshold. Note that setting $\rho_{ab} = 1$ enforces a hard many-to-one mapping, and Bayesian ARTMAP reduces to Gaussian ARTMAP (Sec. 3.1.8) during inference. In case of a mismatch, ART$_a$'s vigilance is temporarily changed to

$$\rho_a = M_J^a - \delta,\ 0 \le \delta \ll M_J^a, \tag{180}$$

where M_J^a is computed using Eq. (58). The search continues until another resonant node is found or a new node is created. When learning finally ensues, the matrix $\boldsymbol{N}$ entry n_{KJ} (class K and ART$_a$'s resonant node J association) is updated as

$$n_{KJ}(new) = n_{KJ}(old) + 1. \tag{181}$$

Inference. During testing, the class of an unseen sample is predicted using

$$K = \arg\max_k \left(\hat{p}(c_k^b|\boldsymbol{x}^a)\right), \tag{182}$$

where

$$\hat{p}(c_k^b|\boldsymbol{x}^a) = \frac{\sum_{j=1}^{N_a} \hat{p}(c_k^b|c_j^a)\hat{p}(\boldsymbol{x}^a|c_j^a)\hat{p}(c_j^a)}{\sum_{i=1}^{N_b}\sum_{l=1}^{N_a} \hat{p}(c_i^b|c_l^a)\hat{p}(\boldsymbol{x}^a|c_l^a)\hat{p}(c_l^a)}, \tag{183}$$

$$\hat{p}(c_j^a) = \frac{\sum_{k=1}^{N_b} n_{kj}}{\sum_{l=1}^{N_a}\sum_{k=1}^{N_b} n_{kl}}, \tag{184}$$

$$\hat{p}(c_k^b|c_j^a) = \frac{n_{kj}}{\sum_{i=1}^{N_b} n_{ij}}. \tag{185}$$

Bayesian ARTMAP has been combined with topology learning in (Masuyama et al., 2018b; Nooralishahi et al., 2018), kernel frameworks in (Masuyama et al., 2018a,b) and expectation maximization in (Tang & Han, 2010).

3.1.19. Generalized ART. The generalized ART (Yap et al., 2008) is a hybrid model that combines a Gaussian ARTMAP (Sec. 2.1.6) variant to cluster samples in the input space and a generalized regression neural network (Specht, 1991) to perform prediction. In this model, the mapping is one-to-one (bijective) and thus $N_a = N_b = N$.

<u>*Training*</u>. Like Gaussian and Bayesian ARTs (Secs. 2.1.6 and 2.1.10, respectively), the two modified Gaussian ART modules A and B use Bayes' theorem to compute their activation functions (posterior probability as in Eq. (32)), where the prior $\hat{p}(c_j^a)$ is estimated using Eq. (34). Again, the evidence $\hat{p}(\boldsymbol{x}^a)$ is the same for all categories and thus does not influence the WTA competition. The conditional probability estimate $\hat{p}(\boldsymbol{x}^a|\theta_j^a)$ is given by

$$\hat{p}(\boldsymbol{x}^a|\theta_j^a) \propto exp\left[-\frac{1}{2}\lambda(\delta_j^a(\boldsymbol{x}^a))\right], \tag{186}$$

where $\lambda(\delta_j^a)$ is defined an ε-insensitive loss function to handle outliers and noisy data

$$\lambda(\delta_j^a) = \begin{cases} 0, & \text{if } \delta_j^a \leq \varepsilon_a \\ \delta_j^a - \varepsilon_a, & \text{otherwise} \end{cases}, \tag{187}$$

$\varepsilon_a \geq 0$ is a user-defined parameter (if $\varepsilon = 0$, then Eq. (187) reduces to the Laplacian loss function), and

$$\delta_j^a(\boldsymbol{x}^a) = \sum_{i=1}^{d} \left| \frac{\mu_{ji}^a - x_i}{\sigma_{ji}^a} \right|, \tag{188}$$

the parameters $\boldsymbol{\mu}_j^a$, $\boldsymbol{\sigma}_j^a$ and n_j^a correspond to the centroid, standard deviation and sample count of ART$_\text{a}$'s category j.

When ART$_\text{a}$'s BMU is selected via WTA, the following match functions are computed

$$M_J^a = \hat{p}(\boldsymbol{x}^a|c_j^a), \tag{189}$$

$$M_J^b = \hat{p}(x^b|c_j^b), \tag{190}$$

where the system enters a resonant state if $M_J^m \geq \rho_m$, $\rho_m \in [0, 1]$, $m \in \{a, b\}$, i.e., if both vigilance tests are simultaneously satisfied. If learning ensues, then

$$n_J^a(new) = n_J^a(old) + 1, \tag{191}$$

$$\boldsymbol{\mu}_J^a(new) = \left[1 - \frac{1}{n_J^a(new)}\right] \boldsymbol{\mu}_J^a(old) + \frac{1}{n_J^a(new)}\boldsymbol{x}^a, \tag{192}$$

$$\boldsymbol{\sigma}_J^a(new) = \left[1 - \frac{1}{n_J^a(new)}\right] \boldsymbol{\sigma}_J^a(old) + \frac{1}{n_J^a(new)}\left|\boldsymbol{\mu}_J^a(new) - \boldsymbol{x}^a\right|. \tag{193}$$

where the standard deviation update is based on the Laplacian distribution.

For a newly created category, $n_{N+1}^a = 1$, $\boldsymbol{\mu}_{N+1}^a = \boldsymbol{x}^a$, $\boldsymbol{\sigma}_{N+1}^a = \gamma_a$, $\boldsymbol{\sigma}_{N+1}^a = \sigma_{init}^2\vec{\mathbf{1}}$ (user-defined initial standard deviation). Similar dynamics hold for ART$_b$, and for both modules $N = N + 1$.

Inference. A prediction for an unseen sample $\boldsymbol{x}$ is made using

$$f(\boldsymbol{x}^a) = \frac{\sum\limits_{j=1}^{N} \frac{\hat{p}(c_j^a|\boldsymbol{x}^a)}{\sigma_j^b}\mu_j^b}{\sum\limits_{j=1}^{N} \frac{\hat{p}(c_j^a|\boldsymbol{x}^a)}{\sigma_j^b}}, f(\boldsymbol{x}^a) \in \mathbb{R}^1. \tag{194}$$

The enhanced generalized ART (Yap et al., 2010) adds network pruning and rule extraction strategies to the original model. Moreover, $\hat{p}(\boldsymbol{x}^a|c_j^a)$ is formally defined as the Laplacian likelihood function

$$\hat{p}(\boldsymbol{x}^a|c_j^a) = \frac{1}{2^d \prod\limits_{i=1}^{d} \sigma_{ji}^a} exp\left[-\sum_{i=1}^{d} \frac{1}{\sigma_{ij}^a}\left|\mu_{ij}^a - x_i^a\right|\right], \tag{195}$$

and, like Gaussian ART, ART$_a$'s match function is a normalized version of Eq. (195)

$$M_J^a = \hat{p}(\boldsymbol{x}^a|c_j^a) = exp\left[-\sum_{i=1}^{d} \frac{1}{\sigma_{ij}^a}\left|\mu_{ij}^a - x_i^a\right|\right], \tag{196}$$

where for resonance to occur in ART$_\mathrm{a}$, $M_J^a \geq \rho_a$ must be satisfied. The match tracking mechanism compares M_J^b to ρ_b

$$M_J^b = \hat{p}(\boldsymbol{x}^a|c_j^a) = exp\left[-\sum_{i=1}^{d}\frac{1}{\sigma_{ij}^a}\left|\mu_{ij}^a - x_i^a\right|\Big|\right], \tag{197}$$

and if it is not satisfied, then the match tracking mechanism temporarily raises ρ_a, inhibits the current winner category J and resumes the search. The learning and prediction mechanisms are the same as Generalized ART.

The improved generalized ART (Yap et al., 2011) builds upon the enhanced generalized ART by (i) incorporating an ordering algorithm (Dagher et al., 1999) to determine the order of input presentation and (ii) providing multivariate prediction $f(\boldsymbol{x}^a) \in \mathbb{R}^L$ when in inference mode:

$$f_l(\boldsymbol{x}^a) = \frac{\sum\limits_{j=1}^{N} \frac{\hat{p}(c_j^a|\boldsymbol{x}^a)}{\sigma_{jl}^b}\mu_{jl}^b}{\sum\limits_{j=1}^{N} \frac{\hat{p}(c_j^a|\boldsymbol{x}^a)}{\sigma_{jl}^b}}, \; l \in \{1, ..., L\}. \tag{198}$$

3.1.20. Self-Supervised ARTMAP. The self-supervised ARTMAP (Amis & Carpenter, 2010; Carpenter, 2019) is a model designed for self-supervised learning applications. This machine learning modality consists of a supervised learning phase, in which only certain data features are specified, followed by an unsupervised phase, in which all the data features are specified. Similar to fuzzy ARTMAP (Sec. 3.1.2), this model's LTM is defined by $\boldsymbol{\theta} = \{\boldsymbol{w} = [\boldsymbol{u}, \boldsymbol{v}^c]\}$, whose geometric interpretation is a hyperrectangle in the data space. An artifact of this learning modality is the "undercommitted" categories, defined by the presence of "undercommitted" features (i.e., $\exists i : u_i > v_i$).

Training. During the first phase, where supervised learning takes place for a predefined number of epochs, only $\bar{d}$ features are presented to the network. That is, a sample $\boldsymbol{x}$ carries information only with respect to a subset of features. The latter are complement coded, whereas the unspecified features are set to 1's:

$$x_i = \begin{cases} x_i, & \text{if } i = 1, \ldots, \bar{d} \\ 1 - x_i, & \text{if } i = d + 1, \ldots, d + \bar{d} \\ 1, & \text{otherwise} \end{cases}, \tag{199}$$

such that $\|\boldsymbol{x}\|_1 = 2d - \bar{d}$ and $\bar{d} \leq d$. Then, an activation function based on choice-by-difference (Carpenter & Gjaja, 1994) is computed for each category j:

$$T_j = \frac{(2d - \|\boldsymbol{x}\|_1) - \left(\|\boldsymbol{w}_j\|_1 - \|\boldsymbol{x} \wedge \boldsymbol{w}_j^a\|_1\right)}{1 - \gamma\phi_j} - \alpha\left(d - \|\boldsymbol{w}_j\|_1\right), \tag{200}$$

where $0 < \alpha < 1$ is the choice parameter, $0 < \gamma < 1 - \alpha$ is the undercommitment factor and $0 \leq \phi_j \leq 1$ is the degree of undercommitment of category j, defined as

$$\phi_j = \frac{1}{d}\sum_{i=1}^{d}\left[u_{j,i} - v_{j,i}\right]^+ = \frac{1}{d}\sum_{i=1}^{d}\left[w_{j,i} - (1 - w_{j,d+i})\right]^+, \tag{201}$$

where $[\cdot]^+$ is a rectifier operator. After the activation functions are computed, a subset of highly active categories is formed: $\Lambda = \{j : T_j \geq T^u = \alpha d\}$, where T^u is the activation function of a uncommitted category (initialized as $\boldsymbol{w} = \vec{\mathbf{1}}$). If $\Lambda = \{\emptyset\}$, then an uncommitted category is recruited and permanently mapped to the class label-paired with the current input sample. Otherwise, the mapping of the resonant committed category J is assessed. If it is correct, then learning ensues as

$$\boldsymbol{w}_J(new) = \boldsymbol{w}_J(old) - \beta_1\left[\boldsymbol{w}_J(old) - \boldsymbol{x}\right]^+, \tag{202}$$

where $[\cdot]^+$ is a component-wise rectifier operator, and $\beta_1 \in (0, 1]$ is the learning parameter of this first training phase. If the prediction is incorrect, then the match tracking mechanism (user-defined MT+ or MT-, see Sec. 3.1.10) inhibits the resonant neuron, slightly changes the baseline vigilance parameter $\bar{\rho}$ and restarts the search.

During the second phase, unsupervised learning takes place for another pre-defined number of epochs. As opposed to the previous phase, all the data features are presented (i.e., $\boldsymbol{x} = [\boldsymbol{x}, \vec{\mathbf{1}} - \boldsymbol{x}]$), and distributed representation is employed. Additionally, the network runs in slow learning mode, and no mismatches occur (the vigilance parameter is set to zero). Particularly, if $\Lambda = \{\emptyset\}$, then no learning takes place. Next, the activation functions are computed using Eq. (200). The distributed activity $\boldsymbol{y}^{(F_2)}$ of layer F_2 is established using the IG CAM rule described in Sec. 3.1.15 (Eqs. (161) and (162)). All weight vectors are thus updated using the distributed instar learning law

$$\boldsymbol{w}_j(new) = \boldsymbol{w}_j(old) - \beta_2 \left[y_j \vec{\mathbf{1}} - \left(\vec{\mathbf{1}} - \boldsymbol{w}_j(old) \right) - \boldsymbol{x} \right]^+, \tag{203}$$

where $j \in \Lambda$, and $\beta_2 \in [0, 1]$ is the learning parameter of the second training phase.

Inference. In inference mode, the self-supervised ARTMAP dynamics are identical to the unsupervised training stage, except that no learning takes place. Predictions are made using Eqs. (100) and (101) in Sec. 3.1.1.

3.1.21. Biased ARTMAP. Biased ARTMAP (Carpenter & Gaddam, 2010) augments fuzzy ARTMAP with a featural biasing mechanism to handle ordering effects that arise in fast online learning mode. Said mechanism temporarily alters the network's focus among the input sample features following a predictive error.

Training. During training, the choice-by-difference activation function (Eq. (160)) is used to find the winner category J, whose match function is computed as

$$M_J = \frac{\|\tilde{\boldsymbol{y}}^{(F_1)}\|_1}{\|\tilde{\boldsymbol{x}}\|_1}, \tag{204}$$

$$\tilde{\boldsymbol{x}} = [\boldsymbol{x} - \boldsymbol{e}]^{+}, \tag{205}$$

$$\tilde{\boldsymbol{y}}^{(F_1)} = \left[\boldsymbol{y}^{(F_1)} - \boldsymbol{e}\right]^{+}, \tag{206}$$

where $[\cdot]^{+}$ is a component-wise rectifier operator, $\tilde{\boldsymbol{x}}$ is the biased complement coded input vector, $\tilde{\boldsymbol{y}}^{(F_1)}$ is the biased F_1 activity and $\boldsymbol{e} \in \mathbb{R}^{2d}$ is the bias vector, which is set to $\vec{\boldsymbol{0}}$ at the beginning of each input presentation (such that $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ and $\tilde{\boldsymbol{y}}^{(F_1)} = \boldsymbol{y}^{(F_1)}$). If the category J successfully passes the vigilance test (i.e., if it satisfies $M_J \geq \rho$) and is mapped to the correct class, then the learning dynamics are identical to fuzzy ART's (Eq. (21) in Sec. 2.2). Alternately, if the prediction based on the resonant category is incorrect, then the bias vector is updated using Eq. (207), the match tracking algorithm alters the vigilance parameter value (MT-, Sec. 3.1.10) and the search resumes. The bias strength parameter λ in Eq. (207) can be selected by cross-validation procedures (note that setting $\lambda = 0$ implies an unbiased model, i.e., fuzzy ARTMAP).

Inference. In prediction mode, biased ARTMAP behaves identically to fuzzy ARTMAP (Sec. 3.1.2).

$$e_i(new) = \begin{cases} e_i(old), & \text{if } \lambda\left[\left[y_i^{(F_1)} - e_i(old)\right]^{+} - \dfrac{\|\boldsymbol{y}^{(F_1)}\|_1}{2d}\right] \leq 0 \\ e_i(old), & \text{if } e_i(old) \geq \lambda\left[\left[y_i^{(F_1)} - e_i(old)\right]^{+} - \dfrac{\|\boldsymbol{y}^{(F_1)}\|_1}{2d}\right] > 0 \\ \dfrac{\left[y_i^{(F_1)} - \dfrac{\|\boldsymbol{y}^{(F_1)}\|_1}{2d}\right]}{1 + \lambda^{-1}}, & \text{if } y_i^{(F_1)} > e_i(old) \text{ and } \lambda\left[\left[y_i^{(F_1)} - e_i(old)\right] - \dfrac{\|\boldsymbol{y}^{(F_1)}\|_1}{2d}\right] > e_i(old) \end{cases}, \quad \lambda \geq 0 \tag{207}$$

3.1.22. TopoART-C. TopoART-C (Tscherepanow & Riechers, 2012) is an incremental classifier based on fuzzy topoART (Sec. 2.2.2). In this architecture, each topoART module (A and B) is augmented with a classification layer F_3 that is connected to the category layer F_2. Additionally, module B is endowed with a mask layer F_0 preceding its feature layer F_1 to handle incomplete data.

Training. During training, the vigilance tests are layered: the first is unsupervised and equal to fuzzy ART's (Sec. 2.2), while the second is supervised and determines whether a correct class prediction was made. These must be simultaneously satisfied for the system to enter a resonant state and learn.

Inference. Prediction is made using topoART B, since topoART A is only used to filter noise and is therefore disregarded. Specifically, such a prediction depends on whether an unknown sample is completely enclosed by at least one category (which implies alternative activation function (Eq. (70)) equal to 1). In the affirmative case, the system predicts the class associated with the smallest node (measured using Eq. (19)). In the negative case, the system makes a prediction based on a subset of highly active categories. Note that if the sample has missing values, then only non-missing attributes are used in the computations.

3.2. ARCHITECTURES FOR REGRESSION

The supervised ART models described so far have been primarily used for classification purposes. Although, in theory, all ARTMAP variants may be used to perform regression tasks (Sasu & Andonie, 2013). For instance, fuzzy ARTMAP was shown to be a universal function approximator in (Verzi et al., 2003). This section reviews architectures developed specifically for incremental function approximation/interpolation. An experimental comparative study on some of these ART-based regression models can be found in (Sasu & Andonie, 2012).

3.2.1. PROBART. The PROBART model (Marriott & Harrison, 1995) is a fuzzy ARTMAP variant designed to approximate noisy continuous mappings. It has a distinct map field dynamic, whose activity is given by

$$\boldsymbol{y}^{(F^{ab})} = \begin{cases} \boldsymbol{w}_J^{ab} + \boldsymbol{y}^{(F_2^b)}, & \text{if both ARTs are active} \\ \boldsymbol{w}_J^{ab}, & \text{if only ART}_\text{a} \text{ is active} \\ \boldsymbol{y}^{(F_2^b)}, & \text{if only ART}_\text{b} \text{ is active} \\ \vec{\boldsymbol{0}}, & \text{otherwise} \end{cases} . \tag{208}$$

This change turns the map field's weight matrix $\boldsymbol{W}^{ab}$ into a frequency counter for the co-occurrence of resonant categories in both ART modules (i.e., it records the number of associations between nodes of ART_a and ART_b), thereby storing probabilistic information. Note that in this model it is initialized as $\boldsymbol{W}^{ab} = \vec{\boldsymbol{0}}$.

Training. PROBART does not possess a match tracking mechanism, since it is adequate for classification tasks (Marriott & Harrison, 1995) and rule extraction (Carpenter & Tan, 1995) but not for regression (Srinivasa, 1997). Moreover, it directly affects the probability estimation process. Therefore, ART_a's vigilance remains fixed. When learning ensues, F^{ab} weights are updated as

$$\boldsymbol{w}_J^{ab}(new) = \boldsymbol{w}_J^{ab}(old) + \boldsymbol{y}^{(F^{ab})}, \tag{209}$$

considering that ART_a's and ART_b's resonant nodes are J and K, respectively.

Inference. The l^{th} component of the prediction $\hat{f}(\boldsymbol{x}^a)$, when ART_a's resonant category is J, is computed as

$$\hat{f}_l(\boldsymbol{x}^a) = \frac{1}{\|\boldsymbol{w}_J^{ab}\|_1} \sum_{k=1}^{N_b} w_{Jk}^{ab} w_{kl}^b = \sum_{k=1}^{N_b} p_{Jk} w_{kl}^b, \tag{210}$$

where $p_{Jk} = \hat{p}(c^b_k|c^a_J) = \frac{w^{ab}_{Jk}}{\|\boldsymbol{w}^{ab}_J\|_1}$, $\boldsymbol{w}^{ab}_J$ is the J^{th} row of $\boldsymbol{W}^{ab}$, $\|\boldsymbol{w}^{ab}_J\|_1$ is the total number of samples associated with ART$_\mathrm{a}$'s node J across all ART$_\mathrm{b}$ nodes, w^{ab}_{Jn} is the number of co-activations of ART$_\mathrm{a}$'s node J and ART$_\mathrm{b}$'s node n, $l \in \{1, ..., d_b\}$ and d_b is the original non-complement coded dimension (number of features) of ART$_\mathrm{b}$'s input samples. The prediction is thus an average weighted by the conditional probabilities. Note that to perform accurate mappings, PROBART requires large ART$_\mathrm{a}$ vigilance parameter values, consequently generating a large number of categories (Gomez-Sanchez et al., 2002).

PROBART's generalization capability is limited by its WTA prediction, which is addressed by the modified PROBART (Srinivasa, 1997) via distributed prediction. The training process is identical for both models; the difference lies in the inference mode. Each feature l of the prediction $\hat{f}'(\boldsymbol{x}^a)$ is computed as

$$\hat{f}'_l(\boldsymbol{x}^a) = \frac{\sum\limits_{m\in\mathcal{S}} M_m \gamma_m \hat{f}_{m,l}(\boldsymbol{x}^a)}{\sum\limits_{m\in\mathcal{S}} M_m \gamma_m}, \tag{211}$$

where $\mathcal{S}$ is the set of ART$_\mathrm{a}$'s resonant nodes for input $\boldsymbol{x}^a$ (i.e., $M_m \geq \rho_a$, M_m is the match function value of ART$_\mathrm{a}$'s neuron m), $\hat{f}_{m,l}(\boldsymbol{x}^a)$ is ART$_\mathrm{a}$'s neuron m prediction for feature l computed from Eq (210) and γ_m is ART$_\mathrm{a}$'s neuron m's frequency of winning. Concretely, the prediction is an average weighted by ART$_\mathrm{a}$'s nodes' match function values and instance countings. The size of the set $\mathcal{S}$ considered for distributed prediction is defined for each component l using a heuristic that minimizes the root mean squared error over the entire training set.

3.2.2. FasArt and FasBack. FasArt (Izquierdo et al., 1996, 2001) is a neuro-fuzzy system that reinterprets fuzzy ARTMAP (Sec. 3.1.2) as a fuzzy logic system by defining categories as decomposable fuzzy sets in their data spaces (universes).

Training. The training dynamics are identical to fuzzy ARTMAP's (ART$_\text{a}$, ART$_\text{b}$, and the map field), with the exception that the activation function, now also regarded as a fuzzy membership function, is defined as

$$T_j = \prod_{i=1}^{d} T_{j,i}, \tag{212}$$

where $T_{j,i}$ is a triangular fuzzy membership function

$$T_{j,i} = \begin{cases} \left[\dfrac{\gamma(x_i - w_{j,i}) + 1}{\gamma(m_{j,i} - w_{j,i}) + 1} \right]^+, & \text{if } x_i \leq m_{j,i} \\ \left[\dfrac{\gamma(1 - x_i - w_{j,d+i}) + 1}{\gamma(1 - m_{j,i} - w_{j,d+i}) + 1} \right]^+, & \text{if } x_i > m_{j,i} \end{cases}, \tag{213}$$

the parameter γ is the fuzzification rate that controls the width of the fuzzy set support (and consequently the generalization capabilities) and $\boldsymbol{m}_j$ is the centroid associated with category j. The fuzzy support associated with category j is thus defined by $\boldsymbol{w}_j$, $\boldsymbol{m}_j$ and γ. The weight vector $\boldsymbol{w}_J$ of a resonant category J is updated using fuzzy ART's learning dynamics (Eq. (21) in Sec. 2.2), whereas the centroid is updated using

$$\boldsymbol{m}_J(new) = (1 - \beta_c)\boldsymbol{m}_J(old) + \beta_c \boldsymbol{x}, \tag{214}$$

where $\beta_c \in (0, 1]$ is the centroid's learning parameter. This learning dynamic is the same for both ART modules. However, note that the LTMs of ART$_\text{a}$ are also subjected to the constraint of making a correct prediction.

Inference. The prediction of each feature l is obtained using the following defuzzification procedure (average of fuzzy set centroids):

$$\hat{f}_l(\boldsymbol{x}^a) = \frac{\sum\limits_{k=1}^{N_b} \sum\limits_{j=1}^{N_b} m^b_{k,l} w^{ab}_{j,k} T^a_j}{\sum\limits_{k=1}^{N_b} \sum\limits_{j=1}^{N_b} w^{ab}_{j,k} T^a_j}, \tag{215}$$

where T_j^a is the activation of ART$_\mathrm{a}$'s category j, $m_{k,l}^b$ is the l^{th} component of ART$_\mathrm{b}$'s centroid $\boldsymbol{m}_k^b$ associated with category k and $w_{j,k}^{ab}$ is the $\{j, k\}$ entry of the map field matrix $\boldsymbol{W}^{ab}$. Note that FasArt is a universal function approximator (Izquierdo et al., 2001).

For fine-tuning purposes, particularly to improve performance and network compactness (i.e., to reduce category proliferation), FasBack (Izquierdo et al., 1997; Izquierdo et al., 2001) enhances FasArt with error-based learning by using the gradient descent optimization method to adapt some of its parameters

$$\boldsymbol{p}(new) = \boldsymbol{p}(old) - \eta \frac{\partial \mathcal{E}}{\partial \boldsymbol{p}(old)}, \tag{216}$$

where $\boldsymbol{p} \in \{\boldsymbol{m}_j^a, \boldsymbol{m}_k^b, w_{i,j}^{ab}\}$, η is the learning rate, $\mathcal{E}$ is error to be minimized

$$\mathcal{E} = \frac{1}{2} \| \hat{f}(\boldsymbol{x}^a) - \boldsymbol{y}^* \|_2^2, \tag{217}$$

and $\hat{f}(\boldsymbol{x}^a)$ and $\boldsymbol{y}^*$ are the system's prediction and desired response, respectively. Note that two learning cycles are performed: a match-based one followed by an error-based one.

FasArt has spawned many variants including recurrent (Palmero et al., 2000), distributed (Parrado-Hernández et al., 2003, 1999) and dynamic (Izquierdo et al., 2009) models.

3.2.3. Fuzzy ARTMAP with Input Relevances. The FAMR (Andonie & Sasu, 2006; Andonie et al., 2003a) (Sec. 3.1.17), when used for regression applications, makes predictions similarly to PROBART (Eq. (210) in Sec. 3.2.1). Particularly, PROBART is said to be a special case of FAMR with its parameters set to $q_0 = 0$, $q_t = q \in (0, \infty)$ (constant) and $\rho_{ab} = 0$.

3.2.4. Generalized ART. The generalized ART and its variants (Sec. 3.1.19) can be used for both classification and regression problems, for instance, by setting $\rho_b = 1$ for the former and $\rho_b = \rho_a$ for the latter (Yap et al., 2008).

3.2.5. TopoART-R. TopoART-R (Tscherepanow, 2011) is a variant of fuzzy topoART (Sec. 2.2.2) designed for regression purposes. In this model, topoART module B is endowed with an input control layer F_0 preceding its feature layer F_1 to process samples with missing attributes (i.e., to make predictions, since in this case dependent and independent variables are treated as missing and non-missing, respectively).

Training. TopoART-R training is similar to topoART (Sec. 2.2.2); however, it does not perform topological learning. Particularly, the complement coded independent and dependent variables are concatenated as a single input vector to be presented to the network. During the vigilance test stage, two match functions are independently computed for the dependent and independent variables.

Inference. Similar to topoART-C (Sec. 3.1.22), during testing, module A is disregarded, the activation function used is given by Eq. (70) in Sec. 2.2.2 and the prediction strategy depends on whether the input sample is fully enclosed by at least one "partial" category (i.e., a hyperrectangle in the multidimensional subspace formed by the non-missing attributes of the presented sample, from which a prediction is sought). In the affirmative case, a "temporary" category is created from the intersection of these "partial" categories. Then, the prediction for a given missing attribute is the center of the interval defined by the corresponding upper and lower bound components of the "temporary" category. In the negative case, the "temporary" category is created as a weighted average of a subset of highly active nodes, and then the prediction is carried out as previously described.

3.2.6. Bayesian ARTMAP for Regression. The Bayesian ARTMAP for regression (Sasu & Andonie, 2013) uses two Bayesian ART modules to perform clustering on both the input and the output spaces. All the dynamics of Bayesian ARTMAP discussed in Sec. 3.1.18 hold, except for the prediction (i.e., the function approximation) which is given by:

$$\hat{f}(\boldsymbol{x}^a) = \sum_{k=1}^{N_b} \hat{p}(c_k^b|\boldsymbol{x}^a)\boldsymbol{\mu}_k^b, \qquad (218)$$

where $\hat{p}(c_k^b|\boldsymbol{x}^a)$ is computed as described in Sec. 3.1.18. The Bayesian ARTMAP for regression was shown to be a universal function approximator (Sasu & Andonie, 2013).

3.3. SUMMARY

Table 6 summarizes the architectures discussed in terms of their training, inference/testing and the map field's mapping characteristics. Particularly, it lists if winner-takes-all (WTA) or distributed (D) coding is employed by these networks and whether the learned mapping is many-to-one ($ART_a \mapsto ART_b$, surjective) or many-to-many (many-to-one and one-to-many).

4. ART MODELS FOR REINFORCEMENT LEARNING

The ART models described in the following subsections are used to perform reinforcement learning in which agents learn online by interacting with the environment. ART-based reinforcement learning systems have found growing applications, for instance, in the computer games (da Silva & Goes, 2018; Wang et al., 2009; Wang & Tan, 2015) and situational awareness (Brannon et al., 2006, 2009) domains.

4.1. REACTIVE FALCON

The reactive fusion architecture for learning, cognition and navigation (R-FALCON) (Tan, 2004) is a fusion ART-based model (Sec. 2.4.1) that possesses three channels (or F_1 layers): the sensory field ($\mathrm{F}_1^{\mathrm{s}}$), the motor field ($\mathrm{F}_1^{\mathrm{a}}$) and the feedback field ($\mathrm{F}_1^{\mathrm{r}}$), which are used to learn mappings across states ($\boldsymbol{s} = [s_1, ..., s_n]$, where $s_j \in [0, 1], \forall j$), actions ($\boldsymbol{a} = [a_1, ..., a_m]$, $a_i \in [0, 1], \forall i$), and rewards ($r \in [0, 1]$), respectively. The general sense-act-learn dynamics of R-FALCON are described next.

Table 6. Summary of supervised ART models' key characteristics.

ART model	Training	Inference	Mapping	Reference(s)
Classification				
ARTMAP	WTA	WTA	many-to-one	(Carpenter et al., 1991a)
Fuzzy ARTMAP	WTA	WTA	many-to-one	(Carpenter et al., 1992)
Fuzzy Min-Max	WTA	WTA	many-to-one	(Simpson, 1992)
Fusion ARTMAP	WTA	WTA	many-to-many	(Asfour et al., 1993)
LAPART 1	WTA	WTA	many-to-one	(Healy et al., 1993)
ART-EMAP	WTA	D	many-to-one	(Carpenter & Ross, 1995)
ARAM	WTA	WTA	many-to-many	(Tan, 1995)
Gaussian ARTMAP	WTA	D	many-to-one	(Williamson, 1996)
Probabilistic fuzzy ARTMAP	WTA	D	many-to-many	(Lim & Harrison, 1997a)
ARTMAP IC	WTA	D	many-to-many[a]	(Carpenter & Markuzon, 1998)
distributed ARTMAP	WTA/D	D	many-to-many[a]	(Carpenter et al., 1998)
Hypersphere ARTMAP	WTA	WTA	many-to-one	(Anagnostopoulos & Georgiopoulos, 2000)
Ellipsoid ARTMAP	WTA	WTA	many-to-one	(Anagnostopoulos & Georgiopoulos, 2001a,b)
μ-ARTMAP	WTA	WTA	many-to-many	(Gomez-Sanchez et al., 2002)
Default ARTMAP 1	WTA	D	many-to-many[a]	(Carpenter, 2003)
Boosted ARTMAP	WTA	WTA	many-to-many	(Verzi et al., 2006)
FAMR	WTA	WTA	many-to-many	(Andonie & Sasu, 2006)
Default ARTMAP 2	WTA/D	D	many-to-many[a]	(Amis & Carpenter, 2007)
Bayesian ARTMAP	WTA	D	many-to-many	(Vigdor & Lerner, 2007)
Generalized ART	WTA	D	one-to-one	(Yap et al., 2008)
Self-supervised ARTMAP	WTA/D	D	many-to-many[a]	(Amis & Carpenter, 2010)
Biased ARTMAP	WTA	WTA	many-to-many[a]	(Carpenter & Gaddam, 2010)
TopoART-C	WTA	D	many-to-one	(Tscherepanow & Riechers, 2012)
Regression				
PROBART	WTA	WTA	many-to-many	(Marriott & Harrison, 1995)
Modified PROBART	WTA	D	many-to-many	(Srinivasa, 1997)
FasART/FasBack	WTA	D	many-to-one	(Izquierdo et al., 2001)
FAMR	WTA	WTA	many-to-many	(Andonie & Sasu, 2006)
Generalized ART	WTA	D	one-to-one	(Yap et al., 2008)
TopoART-R	WTA	D	many-to-many	(Tscherepanow, 2011)
Bayesian ARTMAP	WTA	D	many-to-many	(Sasu & Andonie, 2013)

[a] One-to-many mapping of inconsistent cases via match tracking MT-.

Prediction. Consider an agent currently at a state $\boldsymbol{s}$. The inputs to R-FALCON's $\mathrm{F}_1^{\mathrm{s}}$, $\mathrm{F}_1^{\mathrm{a}}$ and $\mathrm{F}_1^{\mathrm{r}}$ layers are set to $\boldsymbol{x}^s = \boldsymbol{s}$, $\boldsymbol{x}^a = \vec{\mathbf{1}}$ and $\boldsymbol{x}^r = [1, 0]$, respectively. Note that the feedback field is modeled using $\boldsymbol{x}^r = [r, 1 - r]$. A node J is then selected via a WTA competition (node J maximizes Eq. (82) in Sec. 2.4.1). This setting of $\boldsymbol{x}^r$ is used to bias selection toward maximal rewards during prediction.

Action selection policy. The activity of layer $\mathrm{F}_1^{\mathrm{a}}$, given by

$$\boldsymbol{y}^{(F_1^a)} = \boldsymbol{x}^a \wedge \boldsymbol{w}_J^a = \boldsymbol{w}_J^a, \qquad (219)$$

is used to select the action I as

$$I = \operatorname*{arg\,max}_{1 \leq i \leq m} \left(y_i^{(F_1^a)} \right). \tag{220}$$

The agent performs the selected action I and then enters a new state $\boldsymbol{s}'$.

Learning. Learning ensues similarly to fusion ART (Sec. 2.4.1) using the appropriate F_1 layers' inputs, which depend on the feedback received from performing the selected action I:

- Positive feedback (reward): F_1 layers' inputs are set to $\boldsymbol{x}^s = \boldsymbol{s}$, $\boldsymbol{x}^a = \boldsymbol{a}$, and $\boldsymbol{x}^r = \boldsymbol{r}$.
- Negative feedback (penalty): F_1 layers' inputs are set to $\boldsymbol{x}^s = \boldsymbol{s}$, $\boldsymbol{x}^a = \bar{\boldsymbol{a}} = \vec{\boldsymbol{1}} - \boldsymbol{a}$, and $\boldsymbol{x}^r = \bar{\boldsymbol{r}} = \vec{\boldsymbol{1}} - \boldsymbol{r}$.

R-FALCON suffers from category proliferation, so it must undergo pruning heuristics to enhance interpretability and scalability. Moreover, it can only effectively handle problems with immediate rewards.

4.2. TEMPORAL DIFFERENCE FALCON

The temporal difference fusion architecture for learning, cognition and navigation (TD-FALCON) (Tan, 2006; Tan et al., 2008; Tan & Xiao, 2005) is a fusion ART-based model developed to effectively handle problems with both immediate and delayed rewards. This is accomplished by integrating the temporal difference methods (Sutton & Barto, 2018) of Q-learning (Watkins & Dayan, 1992) and state-action-reward-state-action (SARSA) (Rummery & Niranjan, 1994) in the learning framework. Therefore, TD-FALCON is a value iteration method that learns action policies and value functions for state-action pairs via temporal difference learning. Briefly, the TD-FALCON dynamics are as follows.

Prediction. For a given state $\boldsymbol{s}$, the value function of all actions in the set of actions is predicted by setting the inputs to TD-FALCON's $\mathrm{F}_1^{\mathrm{s}}$, $\mathrm{F}_1^{\mathrm{a}}$, and $\mathrm{F}_1^{\mathrm{r}}$ to $\boldsymbol{x}^s = \boldsymbol{s}$, $\boldsymbol{x}^a = \boldsymbol{a}$ and $\boldsymbol{x}^r = \vec{\mathbf{1}}$, respectively. The action vector $\boldsymbol{a}$ is such that $a_I = 1$ and $a_i = 0$ for $i \neq I$, when taking action I. A node J is then selected via a WTA competition (node J maximizes Eq. (82) in Sec. 2.4.1) for each action.

Action selection policy. The $\mathrm{F}_1^{\mathrm{r}}$ layer activities, given by

$$\boldsymbol{y}^{(F_1^r)} = \boldsymbol{x}^r \wedge \boldsymbol{w}_J^r = \boldsymbol{w}_J^r, \tag{221}$$

are then used to compute the Q-values

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \frac{y_1^{(F_1^r)}}{\sum_{i=1}^{m} y_i^{(F_1^r)}}. \tag{222}$$

An action is then chosen using either a decay ϵ-greedy or softmax policy in order to address the exploration-exploitation trade-off.

Learning. Finally, the system acts. The agent is now in a new state $\boldsymbol{s}'$; receives feedback from the environment; and learns using the state ($\boldsymbol{x}^s = \boldsymbol{s}$), action ($\boldsymbol{x}^a = \boldsymbol{a}$), and reward ($\boldsymbol{x}^r = [Q(\boldsymbol{s}, \boldsymbol{a}), 1 - Q(\boldsymbol{s}, \boldsymbol{a})]$) triad. The value function used in $\boldsymbol{x}^r$ is estimated using

$$Q(\boldsymbol{s}, \boldsymbol{a}) = Q(\boldsymbol{s}, \boldsymbol{a}) + \Delta Q(\boldsymbol{s}, \boldsymbol{a}), \tag{223}$$

where

$$\Delta Q(\boldsymbol{s}, \boldsymbol{a}) = \alpha e_{TD}, \tag{224}$$

e_{TD} is the temporal difference error, and α is the learning rate. Particularly, the TD error for Q-learning (off-policy) is

$$e_{TD} = r + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}', \boldsymbol{a}') - Q(\boldsymbol{s}, \boldsymbol{a}), \tag{225}$$

while the TD error for SARSA (on-policy) is

$$e_{TD} = r + \gamma Q(\boldsymbol{s}', \boldsymbol{a}') - Q(\boldsymbol{s}, \boldsymbol{a}), \tag{226}$$

where r is the immediate feedback, and $\gamma \in [0, 1]$ is the discount factor. Additionally, TD-FALCON incorporates self-scaling (Q-values $\in [0, 1]$) by using

$$\Delta Q(\boldsymbol{s}, \boldsymbol{a}) = \alpha e_{TD} \left(1 - Q(\boldsymbol{s}, \boldsymbol{a})\right). \tag{227}$$

TD-FALCON trades faster learning for a less compact network (category proliferation), compared to gradient-based reinforcement learning approaches, which have a smaller network complexity or memory footprint (i.e., less neurons) but their training process is considerably slower. One of the limitations of this ART model is the bounded Q-values in the range [0, 1], which restricts the classes of problems it can tackle. TD-FALCON has been employed in a multi-agent RL system in (Xiao & Tan, 2007) as well as combined with belief-desire-intention systems in (Tan et al., 2010, 2011).

4.3. UNIFIED ART

The unified ART (Seiffertt & Wunsch II, 2010) is an ART model designed for mixed-modality learning so that it seamlessly switches among the canonical machine learning modalities (UL, SL and RL). An important characteristic of this integration is the weight sharing between modalities. It uses a Markov Decision Process and Q-learning framework, and it has found application, for instance, in the field of situational awareness (Brannon et al., 2006, 2009).

Briefly, the unified ART consists of a fuzzy ART module (Sec. 2.2) and a controller. The latter is represented by a matrix $\boldsymbol{V} = [v_{ij}]_{N \times m}$ whose entries v_{ij} estimate value functions (N and m are the number of categories and available actions, respectively).

Prediction. Upon presentation of an input $\boldsymbol{s}$, the fuzzy ART dynamics are performed. If an uncommitted category is selected, then the controller's matrix $\boldsymbol{V}$ needs to be expanded accordingly.

Action selection policy. After the output activity $\boldsymbol{y}^{(F_2)}$ of layer F$_2$ is established, it is used to select an action I such that

$$I = \underset{1 \leq i \leq m}{\arg\max} \left(a_i \right), \tag{228}$$

where

$$\boldsymbol{a} = \boldsymbol{V}^T \boldsymbol{y}^{(F_2)} = [a_1 ... a_m]^T. \tag{229}$$

The output activity is binary and defined by Eq. (5) in Sec. 2.1.1 when in WTA mode. Alternately, to reduce category proliferation, the output activity can be defined in the distributed mode by setting $y_j^{(F_2)} = T_j$, where the activation functions are computed using Eq. (16).

Learning. After undertaking the selected action, the environment transitions to the next state $\boldsymbol{s}'$, and the system learns according to the type of signal received from the environment. Assuming WTA mode with resonant node J, one of the following takes place:

- Supervised signal (I^*): this signal has the highest priority. If the correct action was selected (i.e., $I = I^*$), then the controller learns as

$$v_{J,i} = \begin{cases} v_{max}, & \text{if } i = I \\ 0, & \text{otherwise} \end{cases}, \tag{230}$$

  where v_{max} is the maximum allowable value. Otherwise, a mismatch triggers a search for a new resonant neuron, within the fuzzy ART module, that maps to I^*. If none is found, then a new neuron $N + 1$ is created and mapped to such action (Eq. (230)).

- Reinforcement signal (r): In case of a reward, the controller learns as

  $$v_{J,I} = v_{J,I} + \alpha r, \tag{231}$$

  where α is a learning rate. Conversely, a penalty causes a mismatch in the fuzzy ART module, which then initiates a search for a new resonant node. The controller still learns using Eq. (231).

- Unsupervised signal ($\{\emptyset\}$): this scenario corresponds to the absence of a signal. No learning takes place in the controller.

Note that, for all signal types, when a resonant neuron is found within the fuzzy ART module, it is adapted according to the fast learning mode described in Sec. 2.2.

4.4. EXTENDED UNIFIED ART

The extended unified ART (Seiffertt & Wunsch II, 2010) is another fuzzy ART-based model designed to perform mixed-modality learning, which is accomplished via layered, modality-dependent, vigilance tests. These multiple vigilance criteria must be simultaneously satisfied for the ART system to enter a resonant state and ensue learning. Particularly, this model encodes the states in fuzzy ART's weight matrix $\boldsymbol{W} = [w_{i,j}]_{N \times n}$, and the value functions of the state-action pairs in both the critic's matrix $\boldsymbol{V} = [v_{i,j}]_{N \times m}$ and the actor's matrix $\boldsymbol{U} = [u_{i,j}]_{N \times m}$ (whose role is akin to ARTMAP's map field matrix $\boldsymbol{W}^{ab}$ (Sec. 3.1.1)), where N is the number of categories, n is the dimension of the state space and m is the number of available actions. Uncommitted nodes are initialized by augmenting $\boldsymbol{W}$ with a row equal to $\vec{\mathbf{1}}$, while $\boldsymbol{U}$ and $\boldsymbol{V}$ are expanded with row vectors containing small random values.

Prediction. Upon arriving at a state $\boldsymbol{s}$, the highest active node J is found following fuzzy ART's dynamics (Sec. 2.2) using the choice-by-difference activation function (Eq. (160) in Sec. 3.1.15).

Action selection policy. An action is selected using

$$I = \underset{1 \leq i \leq m}{\arg\max} \left(u_{J,i} \right), \tag{232}$$

where $\boldsymbol{u}_J$ is the J^{th} row of $\boldsymbol{U}$.

Learning. After performing the chosen action, the environment evolves to the next state $\boldsymbol{s}'$ following its dynamics; vigilance tests and learning ensue in consonance with the type of signal feedback from the environment. Particularly, in the unsupervised learning mode, the extended unified ART learning dynamics are akin to fuzzy ART's, where there exists only a single match function M_J^{UL} (Eq. (18)) and a corresponding unsupervised vigilance test and parameter ρ_{UL}. In this learning mode, neither the actor nor the critic are updated. In the reinforcement learning mode, a reinforcement vigilance test is performed in addition to the unsupervised vigilance test, where the match function M_J^{RL} is equal to the temporal difference error (Sec. 4.2) computed using the corresponding entry of $\boldsymbol{V}$ as the Q-values; if satisfied ($M_J^{RL} > \rho_{RL}$, where $\rho_{RL} \geq 0$ is the reinforcement learning vigilance parameter), the actor is updated as

$$u_{J,I} = \min \left(u_{J,I} + \alpha r, u_{max} \right), \tag{233}$$

where u_{max} is the upper bound for any entry of $\boldsymbol{U}$, and the critic is updated using Eq. (231). If the RL test is not satisfied, a mismatch occurs, and a new search is triggered for the next highest ranking category. This process is repeated until a category satisfies the UL vigilance test while also being associated with an action (Eq. (232)) that is different from the one taken at $\boldsymbol{s}$ (i.e., $i \neq I$), or a new category is created. Finally, the supervised learning mode adds a second match function M_J^{SL} on top of the unsupervised one. The former is akin to default ARTMAP's (Sec. 3.1.15) and assesses if the action taken was the correct one (i.e., $I = I^*$). In the affirmative case, only the actor is updated,

$$u_{J,i} = \begin{cases} u_{max}, & \text{if } i = I \\ 0, & \text{otherwise} \end{cases}, \tag{234}$$

whereas in the negative case, a match tracking procedure (MT-) (Carpenter & Markuzon, 1998) slightly decreases fuzzy ART's baseline vigilance parameter during this input presentation cycle, and the search restarts. Note that in all learning modes, when a category is allowed to learn, it does so by following fuzzy ART's learning dynamics (Sec. 2.2).

5. ADVANTAGES OF ART

5.1. SPEED

One of the main advantages of ART neural network architectures is the speed with which they can process data and the relatively small number of epochs they typically require to converge. This is combined with the fact that they can be operated entirely in an online mode, which makes them very effective when working with streaming data or datasets that are too large to fit entirely in memory.

Particularly, the ART 1 (Sec. 2.1.1) and fuzzy ART (Sec. 2.2) neural networks only require an amount of work linear in the number N of samples in the dataset per epoch, and the amount of work performed for each input sample presentation is similarly linear in the number of features d in the dataset, and the number of category templates k, that this sample is compared against. This leads to a running time complexity of $O(Ndk)$, which means that the running time will grow linearly with the growth of any of these variables when the remaining variables are constant. In the absolute worst case, when each sample is put in its own category, this running time degrades to $O(N^2d)$ since $k = N$ in this case; although this situation is uncommon. The same running time complexity analysis applies to other ART neural architectures that faithfully follow the same learning algorithm. A discussion of fuzzy ART computational complexity analysis was presented in (Granger et al., 1998) and summarized in other studies such as (Majeed et al., 2018; Meng et al., 2016, 2014).

5.2. CONFIGURABILITY

Another one of ART's main advantages is its ease of configurability (Wunsch II, 2009). For many unsupervised learning ART neural architectures, the most influential parameter is the vigilance value ρ, which controls when resonance occurs between an input sample and a category and subsequently, whether this category would be allowed to learn the sample or not. In this way, the ART architectures do not require the choice of the number of clusters, unlike many other clustering algorithms. Meanwhile, the choice of which ART architecture to use and the choice of a reasonable vigilance value can allow the discovery of useful clusters without needing to tweak many sensitive parameter values.

5.3. EXPLAINABILITY

The way that ART builds well-behaved templates representing the categories it learns from the data is another one of its core strengths (Wunsch II, 2009). After sufficient learning has taken place, these templates can provide the ability to interpret the results of the neural network learning (Carpenter & Tan, 1995; Healy & Caudell, 2006, 2019; Healy et al., 2009; Tan, 1997) and to visualize the boundaries of each discovered category or cluster. This transparency is a valuable property, since many other types of neural networks can only be used as a black-box component that cannot be readily explained or interpreted.

5.4. PARALLELIZATION AND HARDWARE IMPLEMENTATION

Another major strength of ART neural networks is their potential for massive parallelism and hardware implementation (Wunsch II, 2009). Notably, early contributions include optoelectronics (Blume & Esener, 1995; Caudell, 1992; Wunsch II, 1991; Wunsch II et al., 1993), analog (Ho et al., 1994) and VLSI (Serrano-Gotarredona & Linares-Barranco, 1996; Serrano-Gotarredona et al., 1998; Tsay & Newcomb, 1991) systems and, more recently, an implementation in memristive hardware (Versace et al., 2012). Although ART networks are incremental learners, and thus suffer from ordering effects (see Sec. 6.1), the calculation of the match and activation function for each category can easily be done in parallel. Thus, ART models lend themselves well to GPU implementations, e.g., fuzzy ART in (Martínez-Zarzuela et al., 2007, 2009), fuzzy ARTMAP in (Martínez-Zarzuela et al., 2011) and ARTtree in (Kim & Wunsch II, 2011). This offers the opportunity for a lower cost, energy consumption and memory footprint than other neural networks' hardware while maintaining online learning capabilities.

6. ART CHALLENGES AND OPEN PROBLEMS

6.1. INPUT ORDER DEPENDENCY

An important problem faced by all agglomerative clustering or incremental learning algorithms, including ART, is order-dependence of data presentation. This is especially true in the fast online learning mode. Many approaches have been developed to mitigate such ordering effects, and they mostly consist of suitable pre- and post-processing strategies (cf. (Brito da Silva & Wunsch II, 2018) and the references cited within). Particularly, for supervised ART models, these strategies include Max-Min clustering (Tou & Gonzalez, 1974) in (Dagher et al., 1998, 1999); class-by-class presentation in (Sit et al., 2009), genetic algorithms (Eiben & Smith, 2015) in (Baek et al., 2014; Palaniappan & Eswaran, 2009);

uncorrelated feature-based ordering in (Oong & Isa, 2014); featural biasing in (Carpenter & Gaddam, 2010); and voting strategies in (Amis & Carpenter, 2007, 2010; Carpenter, 2003; Carpenter et al., 1992; Carpenter & Markuzon, 1998; Lim & Harrison, 2000a,b; Williamson, 1996). In regard to unsupervised ART models, examples of strategies are split, merge and delete operations in (Lughofer, 2008); merging methods in (Brito da Silva et al., 2020; Isawa et al., 2008a,b, 2009); cluster validity index-based vigilance tests in (Brito da Silva & Wunsch II, 2017); learning topologies in (Masuyama et al., 2019; Tscherepanow, 2010, 2012); and exploiting the ordering properties of visual assessment of cluster tendency (VAT) (Bezdek, 2017; Bezdek & Hathaway, 2002) in (Brito da Silva & Wunsch II, 2018). The presentation order of inputs still remains an open problem (even if there is meaningful temporal information embedded in the order of sample presentation (e.g., a time series), and it is much more pronounced when presentation is done in a random order), thus requiring further investigation.

6.2. VIGILANCE PARAMETER ADAPTATION

The vigilance is the single most important parameter in any ART model. Selecting suitable values is critical to the network performance and complexity, especially in clustering applications. However, it is often set empirically in an ad hoc manner. In the unsupervised learning mode, vigilance adaptation has been addressed in fuzzy ART through the usage of game theory (Fudenberg & Tirole, 1991) in (Fung & Liu, 1999); the activation maximization, confliction minimization and hybrid integration rules in (Meng et al., 2013, 2016, 2019); the combination with particle swarm optimization (Kennedy & Eberhart, 1995) and cluster validity indices (Xu & Wunsch II, 2009) in (Smith & Wunsch II, 2015); defining the vigilance as a function of the category size in (Isawa et al., 2008b, 2009); or modeling it as a fuzzy membership function in (Majeed et al., 2018). Despite these contributions, setting the vigilance parameter still remains a challenging task worthy of further exploration, particularly in the online learning mode.

6.3. CATEGORY PROLIFERATION

Category proliferation leads to ART systems with a large number of templates, thus increasing the system complexity, computational burden and memory footprint (Sit et al., 2009; Zhang et al., 2014). Moreover, it also reduces the generalization ability of supervised ART models (Georgiopoulos et al., 2001; Koufakou et al., 2001). Category proliferation is typically caused by:

(i) Weight vector erosion during learning (Carpenter et al., 1991c; Moore, 1989). This phenomenon predominantly affects ART 1- and fuzzy ART-based models (see Secs. 2.1.1, 3.1.1, 2.2, 3.1.2).

(ii) Data with noise (Marriott & Harrison, 1995; Parrado-Hernández et al., 2003, 1999) and/or outliers. ART models are typically sensitive to noisy data, especially in fast learning and WTA modes (Parrado-Hernández et al., 1999). Moreover, noise can cause classes to overlap (Blume & Van Blerkom, 2000).

(iii) Data with overlapping classes (Georgiopoulos et al., 2001; Koufakou et al., 2001; Sit et al., 2009). Data sets of such nature are particularly challenging and usually lead to the overfitting problem in supervised ART models (Anagnostopoulos et al., 2002a,b; Georgiopoulos et al., 2001; Henniges et al., 2005; Koufakou et al., 2001) given the operation of ARTMAP's match tracking mechanism (Blume & Van Blerkom, 2000; Marriott & Harrison, 1995; Sit et al., 2009). Such a case is a major source of category proliferation (Sit et al., 2009).

(iv) The category geometry of the ART model chosen, which might be inadequate to represent the data at hand (Williamson, 1996).

(v) Input order presentation, but to a lesser extent.

The previously mentioned causes have been respectively addressed by:

(i) Using the complement code transformation (Carpenter et al., 1992, 1991a) (see Sec. 2.2).

(ii) Slow learning (Carpenter et al., 1992; Carpenter et al., 1995; Carpenter et al., 1991c), distributed learning (Carpenter, 1996a,b, 1997; Carpenter et al., 1998; Parrado-Hernández et al., 2003, 1999), alternative learning rules (Lee et al., 1995; Lee et al., 1998) and pruning strategies (Carpenter & Tan, 1995; Tan et al., 2009; Tscherepanow, 2010).

(iii) Controlling the training error (Anagnostopoulos et al., 2002a,b; Gomez-Sanchez et al., 2001, 2002; Sanchez et al., 2000; Verzi et al., 2006; Verzi et al., 1998), through cross-validation techniques (Georgiopoulos et al., 2001; Koufakou et al., 2001) and pruning strategies (Blume & Van Blerkom, 2000; Lin & Soo, 1997; Pourpanah et al., 2016) as well as using genetic algorithms (Eiben & Smith, 2015) to evolve ARTMAP models (Al-Daraiseh et al., 2006; Kaylani et al., 2009), detecting and removing samples from overlapping regions (Matias & Neto, 2018; Matias et al., 2017), using a threshold filtering procedure (Zhang et al., 2014) and augmenting existing supervised ART models with a series of modifications (Blume & Van Blerkom, 2000; Sit et al., 2009).

(iv) Selecting ART models with category geometry suitable to represent the data manifold. Examples of geometries include hyperrectangles (Carpenter et al., 1991c), hyperspheres (Anagnostopoulos & Georgiopoulos, 2000), hyperellipsoids (Anagnostopoulos & Georgiopoulos, 2001a,b; Vigdor & Lerner, 2007; Williamson, 1996) and irregular polytopes (Amorim et al., 2007).

(v) Ordering effects are discussed in detail in Sec. 6.1, see also (Brito da Silva & Wunsch II, 2018).

Although several studies in the literature have investigated the category proliferation problem, mitigating it is still an ongoing challenge worthy of further research effort, particularly considering causes (ii) and (iii).

6.4. ONLINE NORMALIZATION

It is a well-known good practice among machine learning and artificial intelligence practitioners to pre-process data through some transformation before presenting it to a neural network. Common transformations are linear (or min-max) normalization, which confines the data samples to the hypercube $[0, 1]^d$ and standardization, in which all data attributes are transformed to have zero mean and unit variance. Both of the aforementioned pre-processing techniques require prior knowledge of the data statistics, e.g., the minimum and maximum for linear normalization, and the mean and variance for standardization. Nonetheless, this information might not be available in some online learning scenarios, thereby adding another challenge to ART systems. Contributions in this direction include the online normalization approaches discussed in (Meng et al., 2015, 2019; Swope, 2012), which address this problem for specific ART models by keeping track of the data ranges and scaling the ART parameters (LTMs) and inputs based on the current minimum and maximum estimates and the ART model developed in (Park et al., 2019), which also tracks the data ranges but whose dynamics are such that learning ensues without normalization, thus allowing its hyperrectangular categories to spawn arbitrary ranges. Despite these contributions, online data transformation, LTM scaling procedures and the ability to process raw data still require further development for many other ART models.

6.5. NEW METRICS

Another challenging area in the development of ART neural networks is the use of new metrics and representations that would allow ART to more robustly solve some domain-specific problems (Wunsch II, 2009), such as grammar inference and natural language processing (Meuth, 2009). Some cases require customized neural network designs, such as when the data structure is neither binary nor continuous-valued vectors or when the data has many categorical attributes with large sets of possible values for each attribute (mixed-type data is addressed in (Lam et al., 2015) in the context of unsupervised feature extraction). In such general cases, it would be highly desirable to have ART models that can deal with this data in its native form without requiring transformations while still maintaining the desirable properties that hold for many existing ART models.

Application oriented activation functions can endow ART-based systems with novel and useful capabilities such as the ability to both discriminate and generalize in a single network (thus performing many-to-many mappings) (Lavoie et al., 1997, 1999) or to track moving patterns (Lavoie, 1999). To maintain ART network's stability property when designing activation functions, it is vital that the value of the custom activation increases when the resonant category undergoes learning (Lavoie, 1999; Lavoie et al., 1997, 1999). Tailored activations are discussed in (Lavoie, 1999; Lavoie et al., 1997, 1999) and modifications include:

(i) Making the activation a function of additional parameters (e.g., vigilance and time).

(ii) Defining functionally different activations for distinct category types.

(iii) Varying the activation function parameters without resetting the network's LTM.

Note that all these modifications do not change the dynamics of the elementary ART model; however, changing the activation function implies changing the search order among the ART categories. Other alternative activation functions have been presented in (Blume & Van Blerkom, 2000; Carpenter & Gjaja, 1994). Additionally, there have been some attempts at combining ART with evolutionary computing approaches in (Elnabarawy et al., 2017) and nested monte carlo search in (Illetskova et al., 2019) as well as other hyper-heuristics (see references cited in (Elnabarawy et al., 2017)), but there are still many challenges and opportunities that need to be addressed in this area.

6.6. DISTRIBUTED REPRESENTATIONS

The WTA category selection process used in most ART architectures can sometimes lead to category proliferation (see Sec. 6.3) and is one of the limiting factors of ART's capacity for mapping complex relations (Parrado-Hernández et al., 2003; Wunsch II, 2009). Extending the capabilities of many ART architectures toward distributed representations would lead to greater representational power for these architectures and allow them to encode more complex templates. However, the challenging aspect of this process is to maintain the desirable speed and stability of those ART systems in the presence of this distributed representation. There are examples of architectures that use distributed representations (see Tables 4 and 6), especially in supervised learning, however there are still many issues to be investigated.

6.7. DICHOTOMY OF MATCH- AND ERROR-BASED LEARNING

In (Wunsch II, 2009) the conjecture is made that the dichotomy of match-based learning (i.e., Hebbian learning and ART) and error-based learning (i.e., using backpropagation (Rumelhart et al., 1986; Werbos, 1974, 1990) in feed-forward neural networks (Haykin, 2009) such as deep learning architectures (Goodfellow et al., 2016)) is likely a false one. This still lacks a definitive resolution. Some contributions combined the use of match-based and error-based learning such as in (Izquierdo et al., 2001; Su & Liu, 2002, 2005) by using gradient methods to optimize some of the ART parameters. However, the problem of building a system that can do both match- and error-based learning like animals appear to be capable of remains a more complex and interesting challenge that holds great promise for much more stable and effective machine learning. In biology, there are clear examples of learning that can happen quickly under the right circumstances, implying match-based learning, as well as incrementally improving through supervised or reinforcement learning in a way that implies error-based learning. The ability to master both types of learning and resolve this conjecture is believed to be a gateway to building machine learning systems that are fast and stable, possessing the ability for life-long learning and being resilient in the face of unpredictable changes in the environment.

7. CODE REPOSITORIES

A list of publicly available online source code/repositories is provided below:

- `github.com/ACIL-Group`
- `techlab.bu.edu/main/article/software`
- `ntu.edu.sg/home/asahtan/downloads.htm`
- `http://www2.imse-cnm.csic.es/~bernabe`
- `ee.bgu.ac.il/~boaz/software.html`
- `libtopoart.eu`

8. CONCLUSIONS

This survey presents an overview of ART models used to perform unsupervised learning (a.k.a. clustering), classification, regression and reinforcement learning tasks. It provides a description for each model focusing on the motivation behind their designs, their dynamics and key characteristics such as their code representation and long-term memory unit. Advantages of ART are discussed as well as open problems. Although mature, the field has room to grow and is still full of opportunities.

ACKNOWLEDGEMENTS

This research was sponsored by the Missouri University of Science and Technology Mary K. Finley Endowment and Intelligent Systems Center; the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance code BEX 13494/13-9; the Army Research Laboratory (ARL) and the Lifelong Learning Machines program from DARPA/MTO, and it was accomplished under Cooperative Agreement Number W911NF-18-2-0260. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Al-Daraiseh, A., Georgiopoulos, M., Anagnostopoulos, G., Wu, A. S., & Mollaghasemi, M. (2006). GFAM: A Genetic Algorithm Optimization of Fuzzy ARTMAP. In *IEEE International Conference on Fuzzy Systems* (pp. 315–322).

Amis, G. P., & Carpenter, G. A. (2007). Default ARTMAP 2. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 777–782).

Amis, G. P., & Carpenter, G. A. (2010). Self-supervised ARTMAP. *Neural Networks*, *23*, 265 – 282.

Amorim, D. G., Delgado, M. F., & Ameneiro, S. B. (2007). Polytope ARTMAP: Pattern Classification Without Vigilance Based on General Geometry Categories. *IEEE Transactions on Neural Networks*, *18*, 1306–1325.

Amorim, D. G., Delgado, M. F., Ameneiro, S. B., & Amorim, R. R. (2011). Evolução das Redes ART e suas Funcionalidades. *Revista OPARA*, *1*, 40 – 59.

Anagnostopoulos, G. C., Bharadwaj, M., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2003). Exemplar-based pattern recognition via semi-supervised learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2782–2787). volume 4.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 59–64). volume 6.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001a). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1221–1226). volume 2.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001b). Ellipsoid ART and ARTMAP for incremental unsupervised and supervised learning. In *Aerospace/Defense Sensing, Simulation, and Controls* (pp. 293–304). International Society for Optics and Photonics.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2002). Category regions as new geometrical concepts in Fuzzy-ART and Fuzzy-ARTMAP. *Neural Networks*, *15*, 1205 – 1221.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2003). Putting the Utility of Match Tracking in Fuzzy ARTMAP Training to the Test. In V. Palade, R. J. Howlett, & L. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 1–6). Berlin, Heidelberg: Springer Berlin Heidelberg.

Anagnostopoulos, G. C., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2002a). Boosted ellipsoid ARTMAP. In *Proc. SPIE*. volume 4739.

Anagnostopoulos, G. C., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2002b). Reducing generalization error and category proliferation in ellipsoid ARTMAP via tunable misclassification error tolerance: boosted ellipsoid ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2650–2655). volume 3.

Andonie, R. (1990). A Converse H-theorem for Inductive Processes. *Comput. Artif. Intell.*, *9*, 161–167.

Andonie, R., & Sasu, L. (2003). A Fuzzy ARTMAP Probability Estimator with Relevance Factor. In *Proc. of the 11th European Symposium on Artificial Neural Networks (ESANN)* (pp. 367–372).

Andonie, R., & Sasu, L. (2006). Fuzzy ARTMAP with input relevances. *IEEE Transactions on Neural Networks*, *17*, 929–941.

Andonie, R., Sasu, L., & Beiu, V. (2003a). A Modified Fuzzy ARTMAP Architecture for Incremental Learning Function Approximation. In *Proc. IASTED Int. Conf. Neural Networks and Computational Intelligence (NCI)* (pp. 124–129).

Andonie, R., Sasu, L., & Beiu, V. (2003b). Fuzzy ARTMAP with relevance factor. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1975–1980). volume 3.

Asfour, Y. R., Carpenter, G. A., Grossberg, S., & Lesher, G. W. (1993). Fusion ARTMAP: an adaptive fuzzy network for multi-channel classification. In *Proc. Third International Conference on Industrial Fuzzy Control and Intelligent Systems* (pp. 155–160).

Baek, J., Lee, H., Lee, B., Lee, H., & Kim, E. (2014). An efficient genetic selection of the presentation order in simplified fuzzy ARTMAP patterns. *Applied Soft Computing*, *22*, 101–107.

Bain, L. J., & Engelhardt, M. (1992). *Introduction to Probability and Mathematical Statistics*. (2nd ed.). Brooks/Cole, Cengage Learning.

Bartfai, G. (1994). Hierarchical clustering with ART neural networks. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 940–944). volume 2.

Bartfai, G. (1995). A comparison of two ART-based neural networks for hierarchical clustering. In *Proc. Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems* (pp. 83–86).

Bartfai, G. (1996). An ART-based modular architecture for learning hierarchical clusterings. *Neurocomputing*, *13*, 31 – 45.

Bartfai, G., & White, R. (1997a). A fuzzy ART-based modular neuro-fuzzy architecture for learning hierarchical clusterings. In *Proc. 6th International Fuzzy Systems Conference* (pp. 1713–1718). volume 3.

Bartfai, G., & White, R. (1997b). Adaptive Resonance Theory-based Modular Networks for Incremental Learning of Hierarchical Clusterings. *Connection Science*, *9*, 87–112.

Bartfai, G., & White, R. (1998). Learning and optimisation of hierarchical clusterings with ART-based modular networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2352–2356). volume 3.

Bartfai, G., & White, R. (2000). Incremental Learning and Optimization of Hierarchical Clusterings with ART-Based Modular Networks. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 87–131). Heidelberg: Physica-Verlag HD.

Bezdek, J. C. (2017). *A Primer on Cluster Analysis: 4 Basic Methods that (usually) Work*. First Edition Design Publishing.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Blume, M., & Esener, S. (1995). Optoelectronic Fuzzy ARTMAP processor. *Optical Computing*, *10*, 213–215.

Blume, M., & Van Blerkom, D. A. (2000). Fuzzy ARTMAP Modifications for Intersecting Class Distributions. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 27–54). Heidelberg: Physica-Verlag HD.

Brannon, N., Conrad, G., Draelos, T., Seiffertt, J., & Wunsch II, D. C. (2006). Information Fusion and Situation Awareness using ARTMAP and Partially Observable Markov Decision Processes. In *Proc. IEEE International Joint Conference on Neural Network (IJCNN)* (pp. 2023–2030).

Brannon, N., Seiffertt, J., Draelos, T., & Wunsch II, D. C. (2009). Coordinated machine learning and decision support for situation awareness. *Neural Networks*, *22*, 316 – 325. Goal-Directed Neural Systems.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019). Dual vigilance fuzzy adaptive resonance theory. *Neural Networks*, *109*, 1–5.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2020). Distributed dual vigilance fuzzy adaptive resonance theory learns online, retrieves arbitrarily-shaped clusters, and mitigates order dependence. *Neural Networks*, *121*, 208 – 228.

Brito da Silva, L. E., & Wunsch II, D. C. (2017). Validity Index-based Vigilance Test in Adaptive Resonance Theory Neural Networks. In *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Brito da Silva, L. E., & Wunsch II, D. C. (2018). A study on exploiting VAT to mitigate ordering effects in Fuzzy ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2351–2358).

Cacoullos, T. (1966). Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, *18*, 179–189.

Carpenter, G. A. (1994). A distributed outstar network for spatial pattern learning. *Neural Networks*, *7*, 159 – 168.

Carpenter, G. A. (1996a). Distributed activation, search, and learning by ART and ARTMAP neural networks. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 244–249).

Carpenter, G. A. (1996b). Distributed ART networks for learning, recognition, and prediction. In *Proc. World Congress on Neural Networks (WCNN)* (pp. 333 – 344).

Carpenter, G. A. (1997). Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks. *Neural Networks*, *10*, 1473 – 1494.

Carpenter, G. A. (2003). Default ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1396–1401). volume 2.

Carpenter, G. A. (2019). Looking to the future: Learning from experience, averting catastrophe. *Neural Networks*, *118*, 204 – 207.

Carpenter, G. A., & Gaddam, S. C. (2010). Biased ART: A neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. *Neural Networks*, *23*, 435 – 451.

Carpenter, G. A., & Gjaja, M. N. (1994). Fuzzy ART Choice Functions. *Proc. World Congress on Neural Networks (WCNN)*, (pp. 713–722).

Carpenter, G. A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., & Grossberg, S. (1987b). ART 2: self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.*, *26*, 4919–4930.

Carpenter, G. A., & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, *3*, 129–152.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*, 565 – 588.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1995). A fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems. *IEEE Transactions on Neural Networks*, *6*, 1330–1336.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991b). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, *4*, 493 – 504.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991c). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Carpenter, G. A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, *11*, 323 – 336.

Carpenter, G. A., Milenova, B. L., & Noeske, B. W. (1998). Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks*, *11*, 793 – 813.

Carpenter, G. A., & Ross, W. D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, *6*, 805–818.

Carpenter, G. A., & Tan, A.-H. (1995). Rule Extraction: From Neural Architecture to Symbolic Representation. *Connection Science*, *7*, 3–27.

Caudell, T. P. (1992). Hybrid optoelectronic adaptive resonance theory neural processor, ART1. *Appl. Opt.*, *31*, 6220–6229.

Caudell, T. P., Smith, S. D., Escobedo, R., & Anderson, M. (1994). NIRS: Large scale ART-1 neural architectures for engineering design retrieval. *Neural Networks*, *7*, 1339 – 1350.

Caudell, T. P., Smith, S. D., Johnson, G. C., & Wunsch II, D. C. (1991). Application of neural networks to group technology. In *Proceedings of SPIE - The International Society for Optical Engineering* (pp. 612–621). volume 1469.

Chin, W. H., Loo, C. K., Seera, M., Kubota, N., & Toda, Y. (2016). Multi-channel Bayesian Adaptive Resonance Associate Memory for on-line topological map building. *Applied Soft Computing*, *38*, 269 – 280.

da Silva, A. R., & Goes, L. F. W. (2018). HearthBot: An Autonomous Agent Based on Fuzzy ART Adaptive Neural Networks for the Digital Collectible Card Game HearthStone. *IEEE Transactions on Games*, *10*, 170–181.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1998). Ordered fuzzy ARTMAP: a fuzzy ARTMAP algorithm with a fixed order of pattern presentation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1717–1722). volume 3.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1999). An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance. *IEEE Transactions on Neural Networks*, *10*, 768–778.

DeClaris, N., & Su, M.-C. (1991). A novel class of neural networks with quadratic junctions. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1557–1562). volume 3.

DeClaris, N., & Su, M.-C. (1992). Introduction to the theory and applications of neural networks with quadratic junctions. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1320–1325). volume 2.

Du, K.-L. (2010). Clustering: A neural network approach. *Neural Networks*, *23*, 89 – 107.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. (2nd ed.). John Wiley & Sons.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing*. (2nd ed.). Springer Publishing Company, Incorporated.

Elnabarawy, I., Tauritz, D. R., & Wunsch II, D. C. (2017). Evolutionary Computation for the Automated Design of Category Functions for Fuzzy ART: An Initial Exploration. In *Proc. Genetic and Evolutionary Computation Conference Companion* GECCO'17 (pp. 1133–1140). New York, NY, USA: ACM.

Elnabarawy, I., Wunsch II, D. C., & Abdelbar, A. M. (2016). Biclustering ARTMAP Collaborative Filtering Recommender System. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2986–2991).

Fritzke, B. (1995). A Growing Neural Gas Network Learns Topologies. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7* (pp. 625–632). MIT Press.

Fudenberg, D., & Tirole, J. (1991). *Game Theory*. Cambridge, MA: MIT Press.

Fukumizu, K., Song, L., & Gretton, A. (2013). Kernel Bayes' Rule: Bayesian Inference with Positive Definite Kernels. *Journal of Machine Learning Research*, *14*, 3753–3783. URL: `http://jmlr.org/papers/v14/fukumizu13a.html`.

Fung, W. K., & Liu, Y. H. (1999). A game-theoretic formulation on adaptive categorization in ART networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1081–1086). volume 2.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *19*, 90 – 106.

Georgiopoulos, M., Fernlund, H., Bebis, G., & Heileman, G. L. (1996). Order of Search in Fuzzy ART and Fuzzy ARTMAP: Effect of the Choice Parameter. *Neural Networks*, *9*, 1541 – 1559.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1990). Convergence Properties of Learning in ART1. *Neural Computation*, *2*, 502–509.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1991). Properties of learning related to pattern diversity in ART1. *Neural Networks*, *4*, 751 – 757.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1992). The N-N-N conjecture in ART1. *Neural Networks*, *5*, 745 – 753.

Georgiopoulos, M., Huang, J., & Heileman, G. L. (1994). Properties of learning in ARTMAP. *Neural Networks*, *7*, 495 – 506.

Georgiopoulos, M., Koufakou, A., Anagnostopoulos, G. C., & Kasparis, T. (2001). Overtraining in fuzzy ARTMAP: Myth or reality? In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1186–1190). volume 2.

Gomez-Sanchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Lopez-Coronado, J. (2001). Safe-μARTMAP: a new solution for reducing category proliferation in fuzzy ARTMAP. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 1197–1202). volume 2.

Gomez-Sanchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Lopez-Coronado, J. (2002). μARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP. *IEEE Transactions on Neural Networks*, *13*, 58–69.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Granger, E., Savaria, Y., Lavoie, P., & Cantin, M.-A. (1998). A comparison of self-organizing neural networks for fast clustering of radar pulses. *Signal Processing*, *64*, 249 – 269.

Grossberg, S. (1968). A prediction theory for some nonlinear functional-differential equations i. learning of lists. *Journal of Mathematical Analysis and Applications*, *21*, 643 – 694.

Grossberg, S. (1969). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, i. *Journal of Mathematics and Mechanics*, *19*, 53–91.

Grossberg, S. (1972). Neural expectation: cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik*, *10*, 49–57.

Grossberg, S. (1976a). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, *23*, 187–202.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, *87*, 1–51.

Grossberg, S. (2013). Adaptive Resonance Theory: how a brain learns to consciously attend, learn, and recognize a changing world. *Neural networks*, *37*, 1–47.

Haykin, S. S. (2009). *Neural networks and Learning Machines*. (3rd ed.). Upper Saddle River, NJ: Pearson Education.

Healy, M. J., & Caudell, T. P. (1998). Guaranteed two-pass convergence for supervised and inferential learning. *IEEE Transactions on Neural Networks*, *9*, 195–204.

Healy, M. J., & Caudell, T. P. (2006). Ontologies and Worlds in Category Theory: Implications for Neural Systems. *Axiomathes*, *16*, 165–214.

Healy, M. J., & Caudell, T. P. (2019). Episodic Memory: A Hierarchy of Spatiotemporal Concepts. *Neural Networks*, *this issue*.

Healy, M. J., Caudell, T. P., & Smith, S. D. G. (1993). A neural architecture for pattern sequence verification through inferencing. *IEEE Transactions on Neural Networks*, *4*, 9–20.

Healy, M. J., Olinger, R. D., Young, R. J., Taylor, S. E., Caudell, T., & Larson, K. W. (2009). Applying category theory to improve the performance of a neural architecture. *Neurocomputing*, *72*, 3158 – 3173.

Heileman, G. L., Georgiopoulos, M., & Juxin Hwang (1994). A survey of learning results for ART1 networks. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 1222–1225). volume 2.

Henniges, P., Granger, E., & Sabourin, R. (2005). Factors of overtraining with fuzzy ARTMAP neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1075–1080). volume 2.

Ho, C. S., Liou, J. J., Georgiopoulos, M., Heileman, G. L., & Christodoulou, C. (1994). Analogue circuit design and implementation of an adaptive resonance theory (ART) neural network architecture. *International Journal of Electronics*, *76*, 271–291.

Huang, J., Georgiopoulos, M., & Heileman, G. L. (1994). Properties of learning in fuzzy ART. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 756–761). volume 2.

Huang, J., Georgiopoulos, M., & Heileman, G. L. (1995). Fuzzy ART properties. *Neural Networks*, *8*, 203 – 213.

Illetskova, M., Elnabarawy, I., Brito da Silva, L. E., Tauritz, D. R., & Wunsch II, D. C. (2019). Nested Monte Carlo Search Expression Discovery for the Automated Design of Fuzzy ART Category Choice Functions. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* GECCO'19 (pp. 171–172). New York, NY, USA: ACM.

Isawa, H., Matsushita, H., & Nishio, Y. (2008a). Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3595–3600).

Isawa, H., Matsushita, H., & Nishio, Y. (2008b). Improved Fuzzy Adaptive Resonance Theory Combining Overlapped Category in Consideration of Connections. In *IEEE Workshop on Nonlinear Circuit Networks (NCN)* (pp. 8–11).

Isawa, H., Matsushita, H., & Nishio, Y. (2009). Fuzzy ART Combining Overlapped Categories Using Variable Vigilance Parameters. In *Proc. International Workshop on Nonlinear Circuits and Signal Processing (NCSP)* (pp. 661–664).

Isawa, H., Tomita, M., Matsushita, H., & Nishio, Y. (2007). Fuzzy Adaptive Resonance Theory with Group Learning and its Applications. In *Proc. International Symposium on Nonlinear Theory and its Applications (NOLTA)* (pp. 292–295).

Ishihara, S., Hatamoto, K., Nagamachi, M., & Matsubara, Y. (1993). ART1.5SSS for Kansei engineering expert system. In *Proc. International Conference on Neural Networks (IJCNN)* (pp. 2512–2515). volume 3.

Ishihara, S., Ishihara, K., Nagamachi, M., & Matsubara, Y. (1995). arboART: ART based hierarchical clustering and its application to questionnaire data analysis. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 532–537). volume 1.

Izquierdo, J. M. C., Almonacid, M., Pinzolas, M., & Ibarrola, J. (2009). dFasArt: Dynamic neural processing in FasArt model. *Neural Networks*, *22*, 479 – 487.

Izquierdo, J. M. C., Dimitriadis, Y. A., Araúzo, M., & Coronado, J. L. (1996). FasArt: A New Neuro-Fuzzy Architecture for Incremental Learning in System Identification. In *IFAC Proceedings Volumes* (pp. 2532 – 2537). volume 29.

Izquierdo, J. M. C., Dimitriadis, Y. A., & Coronado, J. L. (1997). FasBack: matching-error based learning for automatic generation of fuzzy logic systems. In *Proc. International Fuzzy Systems Conference* (pp. 1561–1566). volume 3.

Izquierdo, J. M. C., Dimitriadis, Y. A., Sánchez, E. G., & Coronado, J. L. (2001). Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, *14*, 407 – 425.

Jain, L. C., Lazzerini, B., & Halici, U. (Eds.) (2000). *Innovations in ART Neural Networks* volume 43. (1st ed.). Dordrecht: Physica-Verlag Heidelberg.

Jain, L. C., Seera, M., Lim, C. P., & Balasubramaniam, P. (2014). A review of online learning in supervised neural networks. *Neural Computing and Applications*, *25*, 491–509.

Kasuba, T. (1993). Simplified Fuzzy ARTMAP. *AI Expert*, *8*, 18–25.

Kaylani, A., Georgiopoulos, M., Mollaghasemi, M., & Anagnostopoulos, G. C. (2009). AG-ART: An adaptive approach to evolving ART architectures. *Neurocomputing*, *72*, 2079 – 2092.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1942–1948). volume 4.

Kim, S. (2016). *Novel approaches to clustering , biclustering algorithms based on adaptive resonance theory and intelligent control*. Ph.D. thesis Missouri University of Science and Technology.

Kim, S., & Wunsch II, D. C. (2011). A GPU based Parallel Hierarchical Fuzzy ART clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2778–2782).

Knuth, D. E. (1964). Backus Normal Form vs. Backus Naur Form. *Communications of the ACM*, *7*, 735–736.

Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, *47*, 1902–1914.

Kosko, B. (1986). Fuzzy entropy and conditioning. *Information Sciences*, *40*, 165 – 174.

Koufakou, A., Georgiopoulos, M., Anagnostopoulos, G., & Kasparis, T. (2001). Cross-validation in Fuzzy ARTMAP for large databases. *Neural Networks*, *14*, 1279 – 1291.

Lam, D., Wei, M., & Wunsch II, D. C. (2015). Clustering Data of Mixed Categorical and Numerical Type With Unsupervised Feature Learning. *IEEE Access*, *3*, 1605–1613.

Lavoie, P. (1999). Choosing a choice function: granting new capabilities to ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1988–1993). volume 3.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1997). Multiple categorization using fuzzy ART. In *Proceedings of International Conference on Neural Networks (ICNN'97)* (pp. 1983–1988). volume 3.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1999). Generalization, discrimination, and multiple categorization using adaptive resonance theory. *IEEE Transactions on Neural Networks*, *10*, 757–767.

Le, Q., Anagnostopoulos, G. C., Georgiopoulos, M., & Ports, K. (2005). An experimental comparison of semi-supervised ARTMAP architectures, GCS and GNG classifiers. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3121–3126). volume 5.

Leconte, F., Ferland, F., & Michaud, F. (2014). Fusion Adaptive Resonance Theory Networks Used as Episodic Memory for an Autonomous Robot. In B. Goertzel, L. Orseau, & J. Snaider (Eds.), *Artificial General Intelligence* (pp. 63–72). Cham: Springer International Publishing.

Leconte, F., Ferland, F., & Michaud, F. (2016). Design and integration of a spatio-temporal memory with emotional influences to categorize and recall the experiences of an autonomous mobile robot. *Autonomous Robots*, *40*, 831–848.

Lee, C. J., Yoon, C. G., & Lee, C. W. (1995). A new learning method to improve the category proliferation problem in fuzzy ART. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1393–1396). volume 3.

Lee, J. S., Yoon, C. G., & Lee, C. W. (1998). Learning method for fuzzy ARTMAP in a noisy environment. *Electronics Letters*, *34*, 95–97.

Lerner, B., & Guterman, H. (2008). Advanced Developments and Applications of the Fuzzy ARTMAP Neural Network in Pattern Classification. In L. C. Jain, M. Sato-Ilic, M. Virvou, G. A. Tsihrintzis, V. E. Balas, & C. Abeynayake (Eds.), *Computational Intelligence Paradigms: Innovative Applications* (pp. 77–107). Berlin, Heidelberg: Springer Berlin Heidelberg.

Levine, D. S. (2019). *Introduction to Neural and Cognitive Modeling*. (3rd ed.). New York: Routledge.

Levine, D. S., & Penz, P. A. (1990). ART 1.5–A simplified adaptive resonance network for classifying low-dimensional analog data. In *Proc. of International Conference on Neural Networks (IJCNN)* (pp. 639–642). volume 2.

Lim, C. P., & Harrison, R. F. (1997a). An Incremental Adaptive Network for On-line Supervised Learning and Probability Estimation. *Neural Networks*, *10*, 925 – 939.

Lim, C. P., & Harrison, R. F. (1997b). Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration. *Neural Networks*, *10*, 755 – 774.

Lim, C. P., & Harrison, R. F. (2000a). ART-Based Autonomous Learning Systems: Part I — Architectures and Algorithms. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 133–166). Heidelberg: Physica-Verlag HD.

Lim, C. P., & Harrison, R. F. (2000b). ART-Based Autonomous Learning Systems: Part II — Applications. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 167–188). Heidelberg: Physica-Verlag HD.

Lin, T.-H., & Soo, V.-W. (1997). Pruning fuzzy ARTMAP using the minimum description length principle in learning from clinical databases. In *Proc. Ninth IEEE International Conference on Tools with Artificial Intelligence* (pp. 396–403).

Liu, W., Pokharel, P. P., & Príncipe, J. C. (2007). Correntropy: Properties and Applications in Non-Gaussian Signal Processing. *IEEE Transactions on Signal Processing*, *55*, 5286–5298.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). University of California Press volume 1.

Majeed, S., Gupta, A., Raj, D., & Rhee, F. C.-H. (2018). Uncertain fuzzy self-organization based clustering: interval type-2 fuzzy approach to adaptive resonance theory. *Information Sciences*, *424*, 69 – 90.

Marriott, S., & Harrison, R. F. (1995). A modified fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural Networks*, *8*, 619 – 641.

Martinetz, T., & Schulten, K. (1994). Topology representing networks. *Neural Networks*, *7*, 507 – 522.

Martinetz, T. M., & Shulten, K. J. (1991). A “Neural-Gas” Network Learns Topologies. In T. Kohonen, K. Mäkisara, O. Simula, & J. Kangas (Eds.), *Artificial Neural Networks* (pp. 397–402).

Martínez-Zarzuela, M., Díaz Pernas, F. J., Díez Higuera, J. F., & Rodríguez, M. A. (2007). Fuzzy ART Neural Network Parallel Computing on the GPU. In F. Sandoval, A. Prieto, J. Cabestany, & M. Graña (Eds.), *Computational and Ambient Intelligence* (pp. 463–470). Berlin, Heidelberg: Springer Berlin Heidelberg.

Martínez-Zarzuela, M., Díaz-Pernas, F. J., de Pablos, A. T., Perozo-Rondón, F., Antón-Rodríguez, M., & González-Ortega, D. (2011). Fuzzy ARTMAP Based Neural Networks on the GPU for High-Performance Pattern Recognition. In J. M. Ferrández, J. R. Álvarez Sánchez, F. de la Paz, & F. J. Toledo (Eds.), *New Challenges on Bioinspired Applications* (pp. 343–352). Berlin, Heidelberg: Springer Berlin Heidelberg.

Martínez-Zarzuela, M., Pernas, F. J. D., de Pablos, A. T., Rodríguez, M. A., Higuera, J. F. D., Giralda, D. B., & Ortega, D. G. (2009). Adaptative Resonance Theory Fuzzy Networks Parallel Computation Using CUDA. In J. Cabestany, F. Sandoval, A. Prieto, & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence* (pp. 149–156). Berlin, Heidelberg: Springer Berlin Heidelberg.

Massey, L. (2009). Discovery of hierarchical thematic structure in text collections with adaptive resonance theory. *Neural Computing and Applications*, *18*, 261–273.

Masuyama, N., Loo, C. K., & Dawood, F. (2018a). Kernel Bayesian ART and ARTMAP. *Neural Networks*, *98*, 76 – 86.

Masuyama, N., Loo, C. K., Ishibuchi, H., Nojima, Y., & Lin, Y. (2018b). Topological Kernel Bayesian ARTMAP. In *2018 World Automation Congress (WAC)* (pp. 1–5).

Masuyama, N., Loo, C. K., & Wermter, S. (2019). A Kernel Bayesian Adaptive Resonance Theory with A Topological Structure. *International journal of neural systems*, *29*, 1850052–1 – 1850052–19.

Matias, A. L. S., & Neto, A. R. R. (2018). OnARTMAP: A Fuzzy ARTMAP-based Architecture. *Neural Networks*, *98*, 236 – 250.

Matias, A. L. S., Neto, A. R. R., & Rocha, A. (2017). Opposite-to-Noise ARTMAP Neural Network. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in Computational Intelligence* (pp. 507–519). Cham: Springer International Publishing.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In G. H. Bower (Ed.), *Psychology of Learning and Motivation* (pp. 109 – 165). Academic Press volume 24.

Meng, L., & Tan, A. H. (2012). *Heterogeneous Learning of Visual and Textual Features for Social Web Image Co-Clustering*. Technical Report School of Computer Engineering, Nanyang Technological University.

Meng, L., Tan, A.-H., Leung, C., Nie, L., Chua, T.-S., & Miao, C. (2015). Online Multimodal Co-indexing and Retrieval of Weakly Labeled Web Image Collections. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* ICMR '15 (pp. 219–226). New York, NY, USA: ACM.

Meng, L., Tan, A.-H., & Wunsch II, D. (2013). Vigilance adaptation in adaptive resonance theory. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7).

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2016). Adaptive scaling of cluster boundaries for large-scale social media data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 2656–2669.

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2019). *Adaptive Resonance Theory in Social Media Data Clustering: Roles, Methodologies, and Applications*. Cham: Springer International Publishing.

Meng, L., Tan, A. H., & Xu, D. (2014). Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 2293–2306.

Mermillod, M., Bugaiska, A., & Bonin, P. (2013). The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, *4*, 1–3.

Meuth, R. J. (2009). *Meta-Learning Computational Intelligence Architectures*. Ph.D. thesis Missouri University of Science and Technology.

Moore, B. (1989). ART 1 and pattern clustering. In *Proceedings of the 1988 connectionist models summer school* (pp. 174–185). Morgan Kaufmann Publishers San Mateo, CA.

Nasir, J., Kim, D.-H., & Kim, J.-H. (2019). ART neural network-based integration of episodic memory and semantic memory for task planning for robots. *Autonomous Robots*, .

Nasir, J., Yoo, Y., Kim, D., & Kim, J. (2018). User Preference-Based Dual-Memory Neural Model With Memory Consolidation Approach. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2294–2308.

Nooralishahi, P., Loo, C. K., & Seera, M. (2018). Semi-supervised topo-Bayesian ARTMAP for noisy data. *Applied Soft Computing*, *62*, 134 – 147.

Oong, T. H., & Isa, N. A. M. (2014). Feature-Based Ordering Algorithm for Data Presentation of Fuzzy ARTMAP Ensembles. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 812–819.

Palaniappan, R., & Eswaran, C. (2009). Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. *Applied Soft Computing*, *9*, 100–106.

Palmero, G. I. S., Dimitriadis, Y. A., Izquierdo, J. M. C., Sánchez, E. G., & Hernández, E. P. (2000). ART-Based Model Set for Pattern Recognition: FasArt Family. In H. Bunke, & A. Kandel (Eds.), *Neuro-Fuzzy Pattern Recognition* (pp. 145–175). World Scientific.

Park, G., Choi, J., & Kim, J. (2019). Developmental Resonance Network. *IEEE Transactions on Neural Networks and Learning Systems*, *30*, 1278–1284.

Park, G., & Kim, J. (2016). Deep Adaptive Resonance Theory for learning biologically inspired episodic memory. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 5174–5180).

Park, G., Yoo, Y., Kim, D., & Kim, J. (2018). Deep ART Neural Model for Biologically Inspired Episodic Memory and Its Application to Task Performance of Robots. *IEEE Transactions on Cybernetics*, *48*, 1786–1799.

Park, G.-M., Yoo, Y.-H., & Kim, J.-H. (2015). REM-ART: Reward-based electromagnetic adaptive resonance theory. In *Proc. Int. Conf. Artif. Intell. (ICAI)* (pp. 805–811). volume 1.

Parrado-Hernández, E., Gómez-Sánchez, E., & Dimitriadis, Y. A. (2003). Study of distributed learning as a solution to category proliferation in Fuzzy ARTMAP based neural systems. *Neural Networks*, *16*, 1039 – 1057.

Parrado-Hernández, E., Gómez-Sánchez, E., Dimitriadis, Y. A., & Coronado, J. L. (1999). A neuro-fuzzy system that uses distributed learning for compact rule set generation. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 441–446). volume 3.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, *33*, 1065–1076.

Pourpanah, F., Lim, C. P., & Saleh, J. M. (2016). A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction. *Expert Systems with Applications*, *49*, 74 – 85.

Raijmakers, M. E., & Molenaar, P. C. (1997). Exact ART: A Complete Implementation of an ART Network. *Neural Networks*, *10*, 649 – 669.

RamaKrishna, K., Ramam, V. A., & Rao, R. S. (2014). Mathematical Neural Network (MaNN) Models Part III: ART and ARTMAP in OMNI_METRICS. *Journal of Applicable Chemistry*, *3*, 919 – 989.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, *97*, 285 – 308.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1* (pp. 318–362). Cambridge, MA, USA: MIT Press.

Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. Technical Report CUED/F-INFENG/TR 166 Engineering Department, Cambridge University.

Sanchez, E. G., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Coronado, J. L. (2000). MicroARTMAP: use of mutual information for category reduction in fuzzy ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 47–52). volume 6.

Santamaria, I., Pokharel, P. P., & Principe, J. C. (2006). Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Transactions on Signal Processing*, *54*, 2187–2197.

Sasu, L. M., & Andonie, R. (2012). Function Approximation with ARTMAP Architectures. *International Journal of Computers, Communications & Control*, *7*, 957–967.

Sasu, L. M., & Andonie, R. (2013). Bayesian ARTMAP for regression. *Neural Networks*, *46*, 23 – 31.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Seiffertt, J., & Wunsch II, D. C. (2010). *Unified Computational Intelligence for Complex Systems* volume 6 of *Evolutionary Learning and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Serrano-Gotarredona, T., & Linares-Barranco, B. (1996). A Modified ART 1 Algorithm more Suitable for VLSI Implementations. *Neural Networks*, *9*, 1025 – 1043.

Serrano-Gotarredona, T., Linares-Barranco, B., & Andreou, A. G. (1998). *Adaptive Resonance Theory Microchips: Circuit Design Techniques*. Norwell, MA, USA: Kluwer Academic Publishers.

Simpson, P. K. (1992). Fuzzy min-max neural networks. i. classification. *IEEE Transactions on Neural Networks*, *3*, 776–786.

Simpson, P. K. (1993). Fuzzy min-max neural networks - part 2: Clustering. *IEEE Transactions on Fuzzy Systems*, *1*, 32–.

Sit, W. Y., Mak, L. O., & Ng, G. W. (2009). Managing Category Proliferation in Fuzzy ARTMAP Caused by Overlapping Classes. *IEEE Transactions on Neural Networks*, *20*, 1244–1253.

Smith, C., & Wunsch II, D. C. (2015). Particle Swarm Optimization in an adaptive resonance framework. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–4).

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109 – 118.

Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, *2*, 568–576.

Srinivasa, N. (1997). Learning and generalization of noisy mappings using a modified probart neural network. *IEEE Transactions on Signal Processing*, *45*, 2533–2550.

Su, M.-C., DeClaris, N., & Liu, T.-K. (1997). Application of neural networks in cluster analysis. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1–6). volume 1.

Su, M.-C., & Liu, T.-K. (2001). Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, *37*, 165 – 175.

Su, M.-C., & Liu, Y.-C. (2002). A hierarchical approach to ART-like clustering algorithm. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 788–793). volume 1.

Su, M.-C., & Liu, Y.-C. (2005). A new approach to clustering data with arbitrary shapes. *Pattern Recognition*, *38*, 1887 – 1901.

Subagdja, B., & Tan, A.-H. (2012). iFALCON: A neural architecture for hierarchical planning. *Neurocomputing*, *86*, 124 – 139.

Subagdja, B., & Tan, A.-H. (2015). Neural modeling of sequential inferences and learning over episodic memory. *Neurocomputing*, *161*, 229 – 242.

Subagdja, B., Wang, W., Tan, A.-H., Tan, Y.-S., & Teow, L.-N. (2012). Memory Formation, Consolidation, and Forgetting in Learning Agents. In *Proc. 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 1007–1014). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems volume 2 of *AAMAS'12*.

Sutton, R. S., & Barto, A. G. (2018). *Introduction to Reinforcement Learning*. (2nd ed.). Cambridge, MA, USA: MIT Press.

Swope, J. A. (2012). ARTdECOS, adaptive evolving connectionist model and application to heart rate variability. *Evolving Systems*, *3*, 95–109.

Tan, A.-H. (1995). Adaptive Resonance Associative Map. *Neural Networks*, *8*, 437 – 446.

Tan, A.-H. (1997). Cascade ARTMAP: integrating neural computation and symbolic knowledge processing. *IEEE Transactions on Neural Networks*, *8*, 237–250.

Tan, A.-H. (2004). FALCON: a fusion architecture for learning, cognition, and navigation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3297–3302). volume 4.

Tan, A.-H. (2006). Self-organizing Neural Architecture for Reinforcement Learning. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 470–475). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence Through Interaction: Towards a Unified Theory for Learning. In D. Liu, S. Fei, Z.-G. Hou, H. Zhang, & C. Sun (Eds.), *Advances in Neural Networks - ISNN 2007* (pp. 1094–1103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Feng, Y.-H., & Ong, Y.-S. (2010). A self-organizing neural architecture integrating desire, intention and reinforcement learning. *Neurocomputing*, *73*, 1465 – 1477. Advances in Computational Intelligence and Learning.

Tan, A.-H., Lu, N., & Xiao, D. (2008). Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning With Delayed Evaluative Feedback. *IEEE Transactions on Neural Networks*, *19*, 230–244.

Tan, A.-H., Ong, Y.-S., & Tapanuj, A. (2011). A hybrid agent architecture integrating desire, intention and reinforcement learning. *Expert Systems with Applications*, *38*, 8477 – 8487.

Tan, A.-H., Subagdja, B., Wang, D., & Meng, L. (2019). Self-organizing neural networks for universal learning and multimodal memory encoding. *Neural Networks*, *this issue*.

Tan, A.-H., & Xiao, D. (2005). Self-organizing cognitive agents and reinforcement learning in multi-agent environment. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 351–357).

Tan, S. C., Rao, M. V. C., & Lim, C. P. (2009). An online pruning strategy for supervised ARTMAP-based neural networks. *Neural Computing and Applications*, *18*, 387–395.

Tang, X.-l., & Han, M. (2010). Semi-supervised Bayesian ARTMAP. *Applied Intelligence*, *33*, 302–317.

Tou, J. T., & Gonzalez, R. C. (1974). *Pattern recognition principles*. Addison-Wesley,.

Tsay, S. W., & Newcomb, R. W. (1991). VLSI implementation of ART1 memories. *IEEE Transactions on Neural Networks*, *2*, 214–221.

Tscherepanow, M. (2010). TopoART: A Topology Learning Hierarchical ART Network. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 157–167). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2011). An Extended TopoART Network for the Stable On-line Learning of Regression Functions. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *International Conference on Neural Information Processing (ICONIP)* (pp. 562–571). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In P. Perner (Ed.), *Proc. Industrial Conference on Data Mining (ICDM)* (pp. 22–34). ibai-publishing.

Tscherepanow, M., Kortkamp, M., & Kammer, M. (2011). A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*, *24*, 906 – 916.

Tscherepanow, M., Kühnel, S., & Riechers, S. (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 24–29).

Tscherepanow, M., & Riechers, S. (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 18–23).

Vakil-Baghmisheh, M.-T., & Pavešić, N. (2003). A Fast Simplified Fuzzy ARTMAP Network. *Neural Processing Letters*, *17*, 273–316.

Versace, M., Kozma, R. T., & Wunsch, D. C. (2012). Adaptive Resonance Theory Design in Mixed Memristive-Fuzzy Hardware. In R. Kozma, R. E. Pino, & G. E. Pazienza (Eds.), *Advances in Neuromorphic Memristor Science and Applications* (pp. 133–153). Dordrecht: Springer Netherlands.

Verzi, S. J., Heileman, G. L., & Georgiopoulos, M. (2006). Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory. *Neural Networks*, *19*, 446 – 468.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Anagnostopoulos, G. (2002). Off-line structural risk minimization and BARTMAP-S. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2533–2538). volume 3.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Anagnostopoulos, G. C. (2003). Universal approximation with Fuzzy ART and Fuzzy ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1987–1992). volume 3.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Healy, M. J. (1998). Boosted ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 396–401). volume 1.

Verzi, S. J., Heileman, G. L., Georgiopoulus, M., & Healy, M. J. (2001). Rademacher penalization applied to fuzzy ARTMAP and boosted ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1191–1196). volume 2.

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*, 1628–1644.

Wang, D., Subagdja, B., Tan, A.-H., & Ng, G.-W. (2009). Creating human-like autonomous players in real-time first person shooter computer games. In *Proc. Twenty-First Innovative Applications of Artificial Intelligence Conference* (pp. 173 – 178).

Wang, D., & Tan, A. (2015). Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game. *IEEE Transactions on Computational Intelligence and AI in Games*, *7*, 123–138.

Wang, W., Subagdja, B., Tan, A.-H., & Starzyk, J. A. (2010). A self-organizing approach to episodic memory modeling. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Wang, W., Subagdja, B., Tan, A.-H., & Starzyk, J. A. (2012a). Neural Modeling of Episodic Memory: Encoding, Retrieval, and Forgetting. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1574–1586.

Wang, W., Subagdja, B., Tan, A.-H., & Tan, Y.-S. (2012b). A self-organizing multi-memory system for autonomous agents. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Wang, W., Tan, A., & Teow, L. (2017). Semantic Memory Modeling and Memory Interaction in Learning Agents. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*, 2882–2895.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.

Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis Harvard University.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*, 1550–1560.

Williamson, J. R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, *9*, 881 – 897.

Wunsch II, D. C. (1991). *An optoelectronic learning machine: invention, experimentation, analysis of first hardware implementation of the ART 1 neural network*. Ph.D. thesis University of Washington.

Wunsch II, D. C. (2009). ART properties of interest in engineering applications. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 3380–3383).

Wunsch II, D. C., Caudell, T. P., Capps, C. D., Marks, R. J., & Falk, R. A. (1993). An optoelectronic implementation of the adaptive resonance neural network. *IEEE Transactions on Neural Networks*, *4*, 673–684.

Xiao, D., & Tan, A. (2007). Self-Organizing Neural Architectures and Cooperative Learning in a Multiagent Environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *37*, 1567–1580.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, *24*, 709–716.

Xu, R., Wunsch II, D. C., & Kim, S. (2012). Methods and systems for biclustering algorithm. U.S. Patent 9,043,326 Filed January 28, 2012, claiming priority to Provisional U.S. Patent Application, January 28, 2011, issued May 26, 2015.

Yap, K. S., Lim, C. P., & Abidin, I. Z. (2008). A Hybrid ART-GRNN Online Learning Neural Network With a ε-Insensitive Loss Function. *IEEE Transactions on Neural Networks*, *19*, 1641–1646.

Yap, K. S., Lim, C. P., & Au, M. T. (2011). Improved GART Neural Network Model for Pattern Classification and Rule Extraction With Application to Power Systems. *IEEE Transactions on Neural Networks*, *22*, 2310–2323.

Yap, K. S., Lim, C. P., & Mohamad-Saleh, J. (2010). An enhanced generalized adaptive resonance theory neural network and its application to medical pattern classification. *Journal of Intelligent & Fuzzy Systems*, *21*, 65–78.

Yavaş, M., & Alpaslan, F. N. (2009). Behavior categorization using Correlation Based Adaptive Resonance Theory. In *Proc. 17th Mediterranean Conference on Control and Automation* (pp. 724–729).

Yavaş, M., & Alpaslan, F. N. (2012). Hierarchical behavior categorization using correlation based adaptive resonance theory. *Neurocomputing*, *77*, 71 – 81.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338 – 353.

Zhang, Y., Ji, H., & Zhang, W. (2014). TPPFAM: Use of threshold and posterior probability for category reduction in fuzzy ARTMAP. *Neurocomputing*, *124*, 63 – 71.

II. VALIDITY INDEX-BASED VIGILANCE TEST IN ADAPTIVE RESONANCE THEORY NEURAL NETWORKS

Leonardo Enzo Brito da Silva[1,2] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering

Missouri University of Science and Technology, Rolla, MO, USA

[2]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF, Brazil

Email: {leonardoenzo, wunsch}@ieee.org

ABSTRACT

One of the distinguishing features of Adaptive Resonance Theory (ART) is that it relies on a second similarity check, called a vigilance test, to accept or reject a sample into a given category. Generic unsupervised versions of ART rely on a single layer vigilance test, whereas their supervised counterparts possess a second layer test based on classification errors that trigger a match tracking procedure regulated by an inter-ART block. This work uses a second layer vigilance test based on validity indices. A new sample is accepted into a category if its match function surpasses the vigilance test of both layers: the standard first check is based on minimum similarity, and the second check analyses whether setting that sample as belonging to the winner category results in an improvement of the current data partition according to the chosen validity index used as a cost function. Namely, if the new clustering state is superior to the previous one, then learning is allowed for the winning category. Otherwise, the algorithm proceeds as usual in ART implementations. Thus, this local greedy heuristic uses the validity index as a reinforcement signal, looking at the immediate reward to guide the learning of the ART categories without an additional external optimizer algorithm. A sweep analysis of the first layer vigilance parameter was performed and experiments indicate that the presented approach outperforms the standard

Fuzzy ART neural network when samples are randomly presented. When samples are presented in a predefined order, Fuzzy ART obtains the best peak performance, however the modified approach was less sensitive to parameter variations.

1. INTRODUCTION

Cluster analysis consists of finding the data's natural structure, where within-cluster samples are highly similar or homogeneous and between-cluster samples are highly dissimilar or heterogeneous. The literature is very fruitful in terms of clustering methods, and comprehensive reviews can be found in (Xu & Wunsch II, 2005; Xu & Wunsch II, 2009; Xu & Wunsch II, 2010). Moreover, clustering can be used as a preprocessing stage for classification applications (Chou et al., 2017; Liu et al., 2006). A particularly interesting neural network-based clustering approach is the one that uses Adaptive Resonance Theory (ART) (Carpenter & Grossberg, 1987a, 1988; Grossberg, 1976a,b). The latter is a learning theory developed as a solution to the stability-plasticity dilemma, i.e., newly learned rules must be stored without overwriting current memory content (catastrophic forgetting) in the knowledge base.

ART belongs to the class of hard competitive learning clustering methods, and is related to the leader-follower algorithm (Xu & Wunsch II, 2009). Contrary to many clustering algorithms, the number of clusters do not need to be set a priori, as the ART categories are created as needed according to a minimum degree of similarity defined by the vigilance parameter. It controls the granularity of the clustering outcome: the larger the vigilance the stricter the similarity constraint and the more clusters are formed. Conversely, the lower the vigilance value the lower the number of clusters formed. Briefly, when an input sample is presented, an activation function is computed for each node to answer "from the set of current categories, which one best matches the input sample?". Next, a second similarity calculation is performed, known as the vigilance check, to answer "is the winning category similar enough?". If the answer is no to the latter question, then another winner-takes-all (WTA) competition follows without that winning category. The entire process incrementally builds a mapping from samples to categories.

ART has spawned many neural network architecture implementations for both supervised (ARTMAP) and unsupervised (ART) learning such as Fuzzy (Carpenter et al., 1992, 1991), Gaussian (Williamson, 1996), Bayesian (Vigdor & Lerner, 2007), Hypersphere (Anagnostopoulos & Georgiopoulos, 2000) and Ellipsoid (Anagnostopoulos & Georgiopoulos, 2001), just to name a few. Each of which, has its own particular category representation, measures to compute similarity and ways to define the vigilance criterion. A common characteristic of the ART family members is that performance is sensitive to the setting of the global vigilance parameter. This is true, not only in a global sense/system level, but also in a local sense/category level. Many approaches to address this challenge have been presented. For instance, (Smith & Wunsch II, 2015) uses particle swarm optimization to find independent (local) vigilance thresholds for each Fuzzy ART category to better suit the clusters they represent, where validity indices are used as cost functions; whereas (Meng et al., 2013, 2016) developed the activation maximization, the confliction minimization and the hybrid integration rules to adapt local vigilance parameters.

Moreover, ART was used to build a system that integrates unsupervised (UL), supervised (SL) and reinforcement (RL) learning: the Unified ART architecture (Seiffertt & Wunsch II, 2010), which seamlessly switches among these three machine learning modalities. It uses a Markov Decision Process framework. One of its important characteristics is the weight sharing among modalities. An application in the field of situation awareness is discussed in (Brannon et al., 2006, 2009). This paper uses a second layer vigilance test based on validity indices, similar to the reinforcement learning scenario discussed in (Seiffertt & Wunsch II, 2010): a greedy heuristic looks at the immediate reward of adding a sample to a category and, it allows learning only if the next clustering state is superior to the previous according to the validity index used as the cost function.

Another ART-based system that integrates UL, SL, RL and rule-based knowledge integration is the Fusion ART (Tan et al., 2007). It learns associative mappings through multiple feature representation fields and a single category representation field architecture that unifies ART, Adaptive Resonance Associative Map (ARAM) (Tan, 1995) and Fusion Architecture for Learning, COgnition, and Navigation (FALCON) (Tan, 2004, 2006; Tan et al., 2008). A one- (samples), two- (samples, class labels) and three- (sensory/state, motor/actions, feedback/rewards) channel Fusion ART reduces to ART (UL paradigm), ARAM (SL paradigm) and FALCON systems (RL paradigm), respectively. Additionally it has accessible interpretation since category field nodes can generate IF-THEN rules that maps antecedents and consequents from one channel to another and it possesses the ability to insert rules.

The clusters formed throughout the learning process in competitive and online/incremental learning methods is an artifact of the sample presentation order and the parameter setting. By making the training samples equally weighted when the learning process is done, (Wang, 1997) develped learning rate rules that make, when certain conditions are satisfied, competitive learning networks yield the same outcome regardless of the order of presentation. Alternatively, (Lughofer, 2008) uses split-and-merge (guided by a validity index to select the best partition at each incremental stage) and removal of satellite clusters strategies to improve clustering solutions. Since ART belongs to this class of algorithms, its cluster formation is dependent on the order of sample presentation (Xu & Wunsch II, 2009; Xu & Wunsch II, 2011). Therefore, in this work, the behavior of the presented approach is investigated by performing a series of experiments using random and predefined cluster-by-cluster presentations. Then, the performance is analyzed in terms of the quality of the partitions compared to standard Fuzzy ART, which is used as the baseline.

This paper is organized into six sections: Section 2 presents an overview of Fuzzy ART and validity indices, Section 3 introduces the approach methodology, Section 4 describes the experimental set up, Section 5 displays and discusses the results obtained and Section 6 concludes this paper.

2. BACKGROUND AND RELATED WORK

2.1. FUZZY ART

Fuzzy ART (Carpenter et al., 1991) is a neural network implementation of ART that can process binary and real valued data by incorporating fuzzy set theory operations in the fitness calculations (activation and match functions). It is an adaptable (plastic), stable, fast, self-organizing incremental learning method. It is composed by the feature representation field F_1 (input layer), category representation field F_2 (output layer) and the orienting subsystem, which is responsible for determining if the input and category are sufficiently similar according to the pre-defined global threshold (vigilance parameter). It operates by shutting down a category using a reset mechanism or allowing it to update its weights. Used and unused categories are referred to as committed and uncommitted, respectively. This type of ART produces easy to interpret hyper-rectangles as a category representation.

Let $\boldsymbol{x} \in \mathbb{R}^d$ be a data sample presented to Fuzzy ART. Briefly, the algorithm performs the following steps:

1. Compute the *activation function* T of each category j:

$$T_j = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|}, \tag{1}$$

where $\boldsymbol{w}_j$ is the weight vector representing category j, $\alpha > 0$ is the choice parameter whose purpose consists of breaking ties, $\wedge$ is a component-wise fuzzy AND operation ($\min(x_l, w_{j,l})$, $l = 1, ..., 2d$) and $|\cdot|$ is the L_1 norm. T represents the percentage of $\boldsymbol{w}_j$ covered by $\boldsymbol{x}$ (degree of overlap).

2. Sort T in descending order and perform a winner-takes-all (WTA) competition.
3. Compute the *match function* M of the best ranked category i:

$$M_i = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_i|}{|\boldsymbol{x}|}, \tag{2}$$

M represents the percentage of $\boldsymbol{x}$ covered by $\boldsymbol{w}_i$.

4. Evaluate the *match criterion* v_1 (vigilance test):

$$v_1 : M_i \geq \rho, \tag{3}$$

where $\rho \in [0, 1]$ is the vigilance parameter.

5. If v_1 is satisfied (resonance condition), then *learning* takes place:

$$\boldsymbol{w}_i^{new} = (1 - \beta)\boldsymbol{w}_i^{old} + \beta(\boldsymbol{x} \wedge \boldsymbol{w}_i^{old}), \tag{4}$$

where $0 < \beta \leq 1$ is the learning rate.

6. Otherwise, select the second ranked category and repeat the process. If no category satisfies v_1, then a new category is created.

Layered vigilance tests produce more complex systems where several criteria v_i need to be simultaneously satisfied (Seiffertt & Wunsch II, 2010). The Unified ART architecture (Seiffertt & Wunsch II, 2010) uses Fuzzy ART as a building block, in which the second vigilance test v_2 changes for each learning modality: in the unsupervised scenario it can be thought of as always true, in the supervised learning setting it is based on the error between the estimated output (system output) and the true output (target output) and, finally, to perform reinforcement learning it is equal to the temporal difference error. A second layer test based on Euclidean distance is introduced in (Huang et al., 2014) (along with heuristics to automate parameter tuning) to improve performance of standard ART 2 (Carpenter & Grossberg, 1987b). Another reinforcement signal-based system is the Performance-guided Adaptive Resonance Theory (P-ART) (Lee et al., 2004, 2003; Palmer-Brown & Lee, 2005) which toggles between ART1 (Carpenter & Grossberg, 1987a) fast learning and Learning Vector Quantization (Kohonen, 1990) according to external feedback received in consecutive times, i.e., this is the snap-drift algorithm that changes learning following a performance increase or decrease.

2.2. CLUSTER VALIDATION

A ubiquitous challenge in cluster analysis consists of assessing the quality of partitions obtained by different methods, since the data structure groundtruth is not available - in fact, modifications in parameter set-up can make the same algorithm return a different solution. Validity indices are quantitative measures developed for such a purpose. They evaluate partitions and provide a systematic way to select an optimal solution according to the best value of a given index. Usually, they exhibit some type of compromise between measures of compactness (within-cluster scatter) and isolation (between-cluster separation). Numerous criteria have been presented in the literature; for comprehensive reviews and experimental studies refer to (Milligan & Cooper, 1985; Vendramin et al., 2010; Xu & Wunsch II, 2005; Xu & Wunsch II, 2009). In this work, the following relative validity indices were used (in the following, k is the number of clusters, N is the cardinality of the data set):

2.2.1. Calinski-Harabasz (CH) Index. the CH index (variance ratio criterion) (Caliński & Harabasz, 1974; Xu & Wunsch II, 2009) is defined as:

$$CH = \frac{tr(S_B)}{tr(S_W)} \times \frac{N-k}{k-1}, \tag{5}$$

where $tr(\cdot)$ is the trace operator, S_B and S_W are the between and within-cluster scatter matrices, respectively. Higher values of CH indicate better clustering solutions.

2.2.2. Pakhira-Bandyopadhyay-Maulik (PBM) Index. the PBM index (Pakhira et al., 2004) comprises a trade-off among three components, it is given by:

$$PBM = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^2, \tag{6}$$

where D_k is the maximum between-cluster separation, E_k is the sum of the total within-cluster scatter among the samples and their cluster centroids and E_1 considers only one cluster comprising all of the data samples. Higher values of PBM indicate better clustering solutions.

2.2.3. Davies-Bouldin (DB) Index. the DB index (Davies & Bouldin, 1979; Xu & Wunsch II, 2009) considers the average ratio of compactness to isolation among all clusters. The DB index is given by:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j}\left(\frac{e_i + e_j}{d_{i,j}}\right), \tag{7}$$

where e_i and e_j are the average Euclidean distances of all samples of clusters i and j to their respective centroid, and $d_{i,j}$ is the distance between centroids i and j. Smaller values of DB indicate better clustering solutions.

2.2.4. Silhouette (SIL) Index. the SIL index (silhouette width criterion) (Rousseeuw, 1987) is defined as:

$$SIL = \frac{1}{N}\sum_{i=1}^{N} \frac{b(x_i) - a(x_i)}{\max[a(x_i), b(x_i)]}, \tag{8}$$

where $a(x_i)$ is the average Euclidean distance of sample i to the remaining samples in the same cluster, and $b(x_i)$ is the minimum average Euclidean distance between sample i and the samples in the remaining clusters. Higher values of SIL indicate better clustering solutions.

3. METHODOLOGY

The validity index-based vigilance test in ART networks consists of using a second match criterion, v_2, based on validity indices in a generic ART framework. This enables the system to accept or reject a sample as part of a given category based on the improvement of the current clustering state, which is measured by said validity index. This is fundamentally different then running a standard ART network and choosing the best partition according to a relative validity index; here the validity index is embedded at the ART learning level. Let a clustering state of a data set $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ be given by the partition $\Omega = \{\omega_1, ..., \omega_k\}$ of disjointed clusters ω_i, where $\bigcup_{i=1}^{k} \omega_i = \boldsymbol{X}$. The second match criterion v_2 is given by:

$$v_2 : J_i \geq \delta, \tag{9}$$

$$J_i = f(\hat{\Omega}) - f(\Omega), \tag{10}$$

where J_i represents the improvement of going from the previous clustering state Ω to a clustering state $\hat{\Omega}$ that includes sample x in category i. The $f(\cdot)$ is the fitness function used; here it is a relative validity index (naturally, if the validity index should be minimized then the inequality sign should be reversed). This approach relates to (Seiffertt & Wunsch II, 2010) in the reinforcement learning scenario. Here, the reinforcement signal is obtained via the validity index. This greedy heuristic selects the next best clustering state according to the immediate reward.

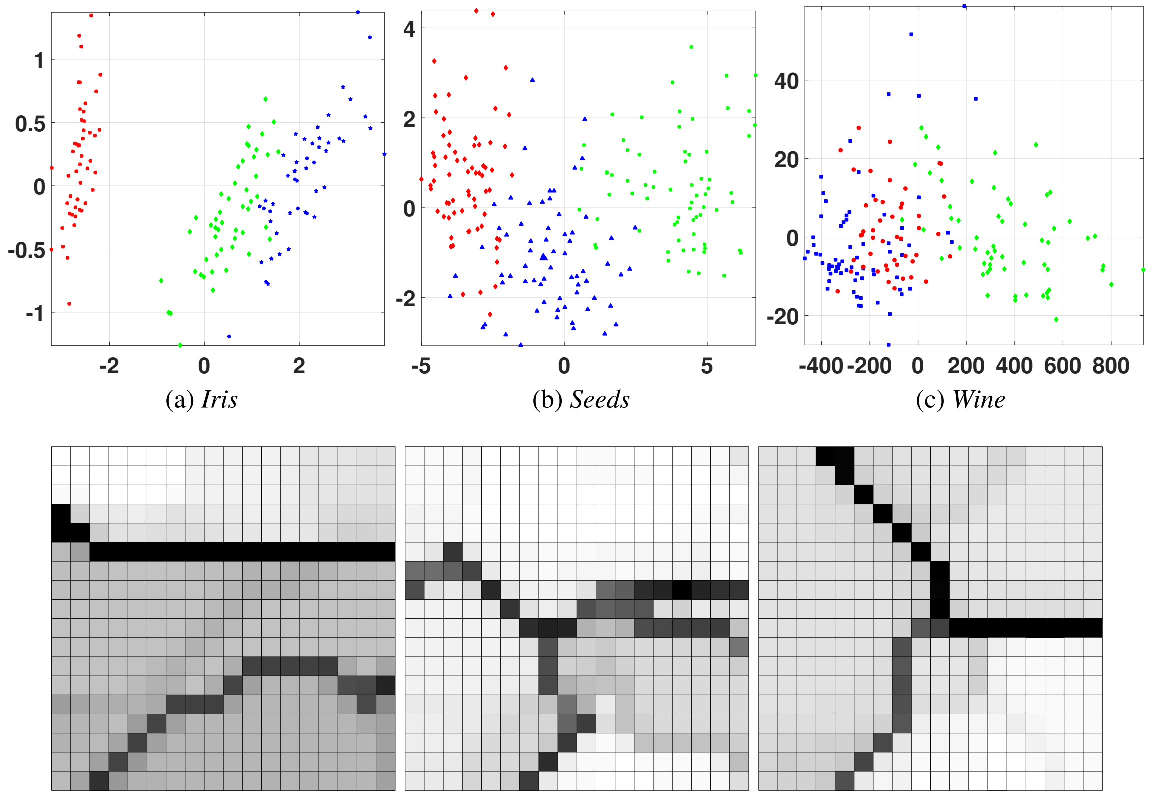
The labels of all samples are set to zero at the very beginning (one single cluster) of the first epoch (pass through the samples). When the samples are mapped to categories, then the validity index is computed considering the unlabeled samples as a part of the same cluster. From the second epoch onward, all samples have been assigned to categories, thus the validity index is computed considering the clustering partition rendered by the ART categories. In this work, the Fuzzy ART flavor of ART is used in the experiments; nonetheless, the application of this methodology to other ART family members is straightforward.

4. EXPERIMENTAL SETUP

In this work, the experiments were carried out using the MATLAB software environment (Statistics and Machine Learning Toolbox) and the Cluster Validity Analysis Platform Toolbox (Wang et al., 2009). Three benchmark data sets from the UC Irvine Machine Learning Repository (Bache & Lichman, 2013) were used for proof of concept: *Iris* (4 features, 150 samples, 3 clusters with 50:50:50 ratio), *Seeds* (7 features, 210 samples, 3 clusters with 70:70:70 ratio) and *Wine* (13 features, 178 samples, 3 clusters with 59:71:48 ratio). Principal component analysis projection (Xu & Wunsch II, 2009) and Self-Organizing Map (SOM) (Kohonen, 1982) rH*-vis heatmap (Brito da Silva & Wunsch II, 2017a, 2018) are used in Figure 1 to visualize these data sets (their relative sizes, approximate shapes and separability). Linear normalization was applied to all data sets in order to scale their features to the range $[0, 1]$. Additionally, complement coding was applied to the inputs of Fuzzy ART.

To investigate the order of the sample presentation's dependency on the presented method and the standard Fuzzy ART, two experiments were conducted: the first emulates a real clustering problem. Thus, the samples of each data set were shuffled and then presented to Fuzzy ART (henceforth regarded as a random presentation). In the second, the samples were presented in a predefined cluster-by-cluster fashion (henceforth regarded as an ordered presentation).

Twenty equally spaced values of v_1 vigilance parameter ρ in the range $[0, 0.9]$ were scanned, and 30 runs were performed for each one of such values for the random presentation and one run for the ordered presentation; the value for v_2 vigilance parameter



(a) *Iris*

(b) *Seeds*

(c) *Wine*

(d) *Iris*

(e) *Seeds*

(f) *Wine*

Figure 1. Data sets depicted using principal component analysis projection (a, b, c) and rH*-vis (d, e, f). The 10 × 10 SOMs were trained using the SOM Toolbox (Vesanto et al., 1999).

was set to $\delta = 0$ (thus the inclusion of a sample in a category is accepted only if it yields a clustering state with a better validity index value). The maximum number of epochs was set to 20, and two additional stopping criteria were used: no change in the Fuzzy ART network weights or no change in the classification of the samples, both between two consecutive epochs. The choice parameter (α) was set to 10^{-3}, and the learning rate (β) was set to 1 (fast learning). Moreover, in all Fuzzy ART implementations, we allow a full search among the committed categories and do not permit the uncommitted category to participate in the WTA competitive process; a new category is added only if none of the current committed categories pass the vigilance check(s). In the latter case, fast commit is applied, in which the weights of the new category become equal to the current sample.

In order to assess the quality of the final partitions extracted by the presented clustering approach, the external validity indices Rand (R) (Rand, 1971; Xu & Wunsch II, 2009) and adjusted Rand (AR) (Hubert & Arabie, 1985; Xu & Wunsch II, 2009) were used:

$$R = \frac{tp + tn}{tp + fp + fn + tn}, \tag{11}$$

$$AR = \frac{\binom{N}{2}(tp + tn) - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}{\binom{N}{2}^2 - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}, \tag{12}$$

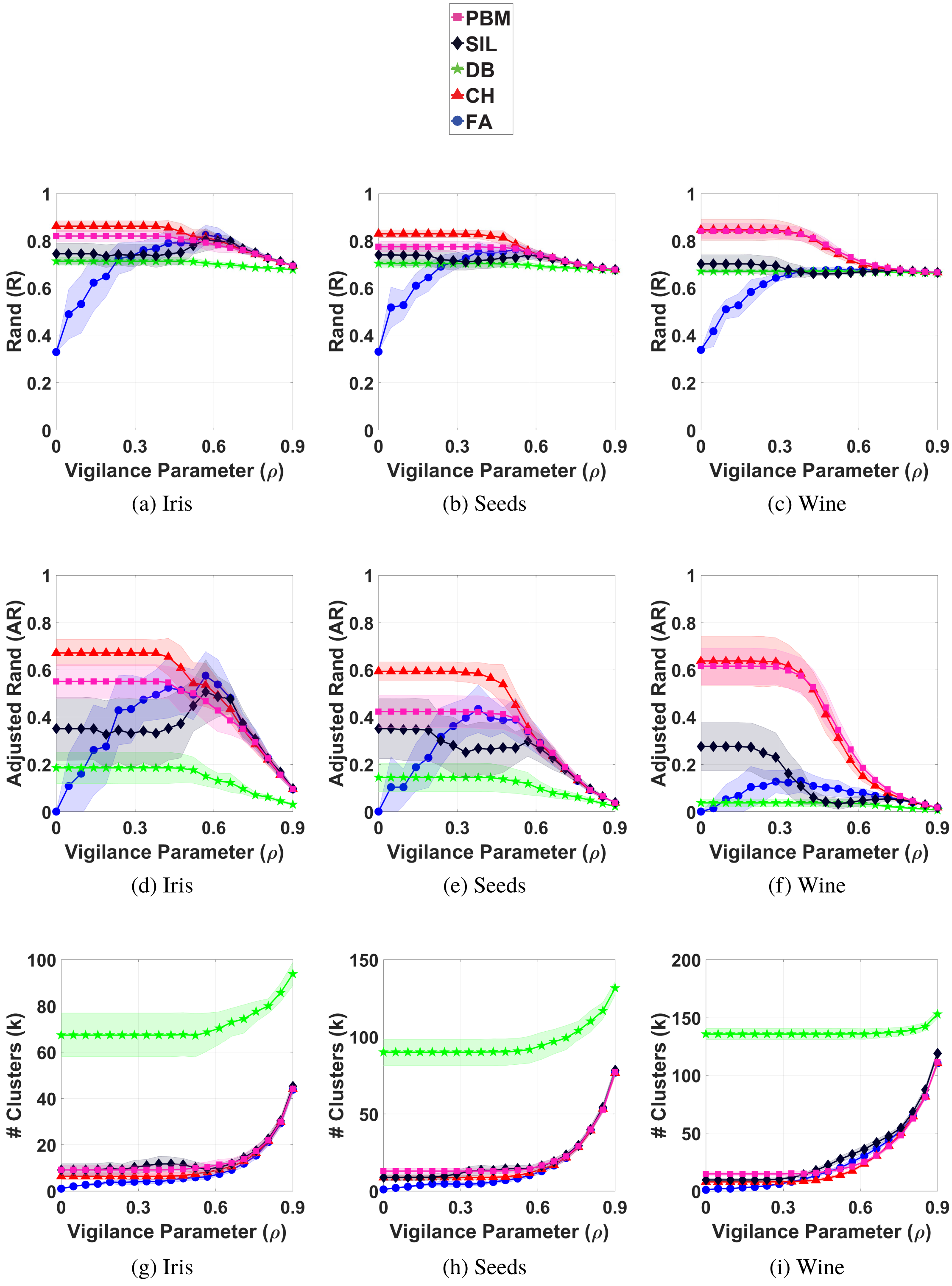
where tp, tn, fp and fn stand for true positive, true negative, false positive and false negative, respectively, according to a reference partition and the clustering outcome. Note that the R and AR are not used as fitness functions of the presented method. The source code of the validity index-based vigilance test in Fuzzy ART is provided at the Applied Computational Intelligence Laboratory public GitLab repository (Brito da Silva & Wunsch II, 2017b).

5. RESULTS AND DISCUSSION

The results obtained by using the validity index-based vigilance test in the two different training scenarios described in Section 4 are depicted in Figs. 2 and 3; the standard Fuzzy ART results are depicted as the baseline performance level. Specifically, Figure 2 depicts the mean R, AR and k along with their respective standard deviations for the validity index-based vigilance check using CH, PBM, SIL and DB for v_2. Conversely, Figure 3 illustrates the same quantities for the ordered presentation experiment.

5.1. EXPERIMENT 1: RANDOM PRESENTATION

Figure 2 shows that for all three data sets, the presented approach outperforms the standard Fuzzy ART considering both average and peak average performance; precisely, the best results were obtained using CH, followed by PBM, SIL and DB, in that order. The



(a) Iris

(b) Seeds

(c) Wine

(d) Iris

(e) Seeds

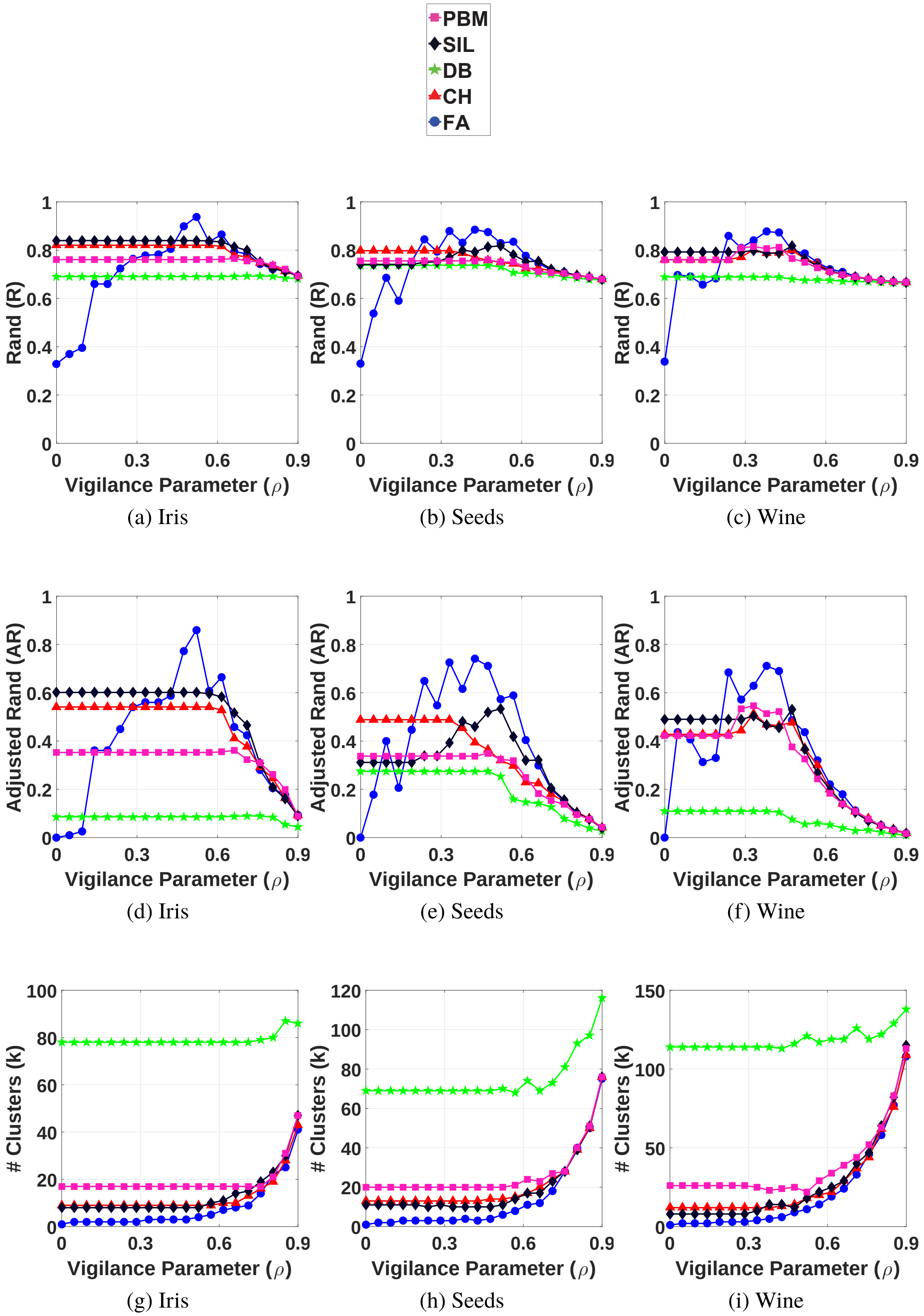
(f) Wine

(g) Iris

(h) Seeds

(i) Wine

Figure 2. Experiment 1: results for the *Iris*, *Seeds* and *Wine* data sets: Rand, adjusted Rand and number of clusters (first, second and third rows, respectively). The CH, PBM, SIL, DB and standard Fuzzy ART (FA) are represented as red triangles, violet squares, black diamonds, green stars and blue circles, respectively.



(a) Iris (b) Seeds (c) Wine

(d) Iris (e) Seeds (f) Wine

(g) Iris (h) Seeds (i) Wine

Figure 3. Experiment 2: results for the *Iris*, *Seeds* and *Wine* data sets: Rand, adjusted Rand and number of clusters (first, second and third rows, respectively). The CH, PBM, SIL, DB and standard Fuzzy ART (FA) are represented as red triangles, violet squares, black diamonds, green stars and blue circles, respectively.

worst results obtained by the DB are due to a considerably large number of mismatches (the current categories fail the second vigilance test for many samples), thus a considerable amount of categories were created, which ended up representing singletons in many cases. This significantly increases the computational cost of training, as compared to the other validity indices. Of course, the fastest method is the standard Fuzzy ART, since no additional calculations are necessary; it is followed by the approach using CH, PBM, SIL and DB, which is not surprising, since the computational complexity of these methods are $O(nN)$, $O(n(k^2 + N))$, $O(nN^2)$ and $O(n(k^2 + N))$, respectively (Vendramin et al., 2010).

For a large interval of low v_1 vigilance values, the performance of the presented approach is virtually constant, thus it seems robust to its respective parameter selection process; the performance degrades for large values. Therefore, we recommend using $\rho = 0$ as a rule of thumb. Basically, this setting neglects v_1 and only considers v_2, which is based on the validity index performance. Regarding the dynamics of v_1 and v_2, increasing the value of ρ makes the first vigilance check more strict regarding the similarity constraint between the sample and the winning category, while the setting of δ used for v_2 made the output stable up to the point where the presented approach behaved similarly to standard Fuzzy ART. In general, for such large ρ values, the likelihood of passing v_2 after satisfying v_1 tends to increase.

5.2. EXPERIMENT 2: ORDERED PRESENTATION

Regarding the second scenario of the predefined ordered presentation, the presented method no longer has superior peak performance: the standard Fuzzy ART has the advantage. Nonetheless, the same general behavior is observed: for small ρ vigilance values, the presented method yields better results than standard Fuzzy ART with large plateaus of practically constant performance (again, using $\rho = 0$ seems to be a reasonable typical parameter setting). This presentations order, however, does not seem suitable for the CH, PBM and DB indices, since their performance has decreased. Interestingly, a significant performance boost of the SIL can be observed, and in fact, it was the best performing index among the ones tested: SIL, CH, PBM and DB are ranked in this order of performance.

For the above experiments, we do not claim these results are comprehensive. Various approaches such as mentioned in (Halkidi & Vazirgiannis, 2008; Lago-Fernández & Corbacho, 2010; Taşdemir & Merényi, 2011; Xu & Wunsch II, 2005; Xu & Wunsch II, 2009) were not considered. In general, the standard Fuzzy ART creates a smaller number of categories (clusters) in both experiments among the methods and data sets; yet, this does not lead to superior performance in Experiment 1, since it does not necessarily imply agreement with the groundtruth partitions.

6. CONCLUSION

This work presents a validity index-based vigilance test in the ART neural network framework, i.e., a second stage of vigilance checking based on validity index is integrated into the learning procedure. We show that, for the data in these experiments, the presented method outperformed standard Fuzzy ART for random sample presentation, thus alleviating Fuzzy ART's dependency on the presentation order of the samples. In ordered cluster-by-cluster presentations, standard Fuzzy ART yielded a better peak performance. Nonetheless, this is not a considerable drawback, since in real clustering applications samples are usually shuffled among classes.

The best results were obtained with the CH and SIL indices for both input presentation scenarios. This corroborates the findings of other studies in which CH and SIL were deemed the best performing validity indices (Milligan & Cooper, 1985; Vendramin et al., 2010; Xu et al., 2012). On the other hand, DB yielded the worst results. Additionally, the presented approach extends the range of effective first vigilance check thresholds of standard Fuzzy ART, since it allows for a consistent performance at a wider vigilance interval for both input presentation scenarios; all things being equal and without any prior knowledge, setting the first vigilance parameter to zero is therefore a reasonable starting point for the clustering process.

It is straightforward to expand the presented approach to other types of ART architectures. Certainly, the performance is bounded by the geometric representation's limitations of the category of that ART family member. It is also constrained by the ability of a given validity index to identify good partitions in a specific data set with particular characteristics: different validity indices exhibit biases towards different structures.

ACKNOWLEDGEMENTS

Support from the National Science Foundation, the Missouri University of Science and Technology Intelligent Systems Center, and the Mary K. Finley Endowment is gratefully acknowledged. Leonardo Enzo Brito da Silva was supported by the CAPES Foundation, Ministry of Education of Brazil, under grant number BEX 13494/13-9.

REFERENCES

Anagnostopoulos, G. C., & Georgiopoulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 59–64). volume 6.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1221–1226). volume 2.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Brannon, N., Conrad, G., Draelos, T., Seiffertt, J., & Wunsch II, D. C. (2006). Information Fusion and Situation Awareness using ARTMAP and Partially Observable Markov Decision Processes. In *Proc. IEEE International Joint Conference on Neural Network (IJCNN)* (pp. 2023–2030).

Brannon, N., Seiffertt, J., Draelos, T., & Wunsch II, D. C. (2009). Coordinated machine learning and decision support for situation awareness. *Neural Networks*, *22*, 316 – 325. Goal-Directed Neural Systems.

Brito da Silva, L. E., & Wunsch II, D. C. (2017a). SOM IT-vis. URL: `https://git.mst.edu/acil-group/som-it-vis`.

Brito da Silva, L. E., & Wunsch II, D. C. (2017b). VI-based Fuzzy ART. URL: `https://git.mst.edu/acil-group/CVI-Fuzzy-ART`.

Brito da Silva, L. E., & Wunsch II, D. C. (2018). An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2595–2613.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*, 1–27.

Carpenter, G. A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., & Grossberg, S. (1987b). ART 2: self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.*, *26*, 4919–4930.

Carpenter, G. A., & Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, *21*, 77–88.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Chou, C.-H., Hsieh, S.-C., & Qiu, C.-J. (2017). Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction. *Applied Soft Computing*, *56*, 298 – 316.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*, 224–227.

Grossberg, S. (1976a). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, *23*, 187–202.

Halkidi, M., & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, *29*, 773 – 786.

Huang, Y.-T., Cheng, F.-T., Shih, Y.-H., & Chen, Y.-L. (2014). Advanced ART2 scheme for enhancing metrology-data-quality evaluation. *Journal of the Chinese Institute of Engineers*, *37*, 1064–1079.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kohonen, T. (1990). Improved versions of learning vector quantization. In *1990 IJCNN International Joint Conference on Neural Networks* (pp. 545–550). volume 1.

Lago-Fernández, L. F., & Corbacho, F. (2010). Normality-based validation for crisp clustering. *Pattern Recognition*, *43*, 782 – 795.

Lee, S. W., Palmer-Brown, D., & Roadknight, C. M. (2004). Performance-guided neural network for rapidly self-organising active network management. *Neurocomputing*, *61*, 5 – 20. Hybrid Neurocomputing: Selected Papers from the 2nd International Conference on Hybrid Intelligent Systems.

Lee, S. W., Palmer-Brown, D., Tepper, J. A., & Roadknight, C. M. (2003). Snap-drift: real-time, performance-guided learning. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* (pp. 1412–1416 vol.2). volume 2.

Liu, B., Wan, C., & Wang, L. (2006). An efficient semi-unsupervised gene selection method via spectral biclustering. *IEEE Transactions on NanoBioscience*, *5*, 110–114.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

Meng, L., Tan, A.-H., & Wunsch II, D. (2013). Vigilance adaptation in adaptive resonance theory. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7).

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2016). Adaptive scaling of cluster boundaries for large-scale social media data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 2656–2669.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*, 159–179.

Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern Recognition*, *37*, 487 – 501.

Palmer-Brown, D., & Lee, S. W. (2005). Continuous reinforced snap-drift learning in a neural architecture for proxylet selection in active computer networks. *International Journal of Simulation: Systems, Science and Technology*, *6*, 11–21.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, *66*, 846–850.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53 – 65.

Seiffertt, J., & Wunsch II, D. C. (2010). *Unified Computational Intelligence for Complex Systems* volume 6 of *Evolutionary Learning and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Smith, C., & Wunsch II, D. C. (2015). Particle Swarm Optimization in an adaptive resonance framework. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–4).

Taşdemir, K., & Merényi, E. (2011). A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*, 1039–1053.

Tan, A.-H. (1995). Adaptive Resonance Associative Map. *Neural Networks*, *8*, 437 – 446.

Tan, A.-H. (2004). FALCON: a fusion architecture for learning, cognition, and navigation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3297–3302). volume 4.

Tan, A.-H. (2006). Self-organizing Neural Architecture for Reinforcement Learning. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 470–475). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence Through Interaction: Towards a Unified Theory for Learning. In D. Liu, S. Fei, Z.-G. Hou, H. Zhang, & C. Sun (Eds.), *Advances in Neural Networks - ISNN 2007* (pp. 1094–1103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Lu, N., & Xiao, D. (2008). Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning With Delayed Evaluative Feedback. *IEEE Transactions on Neural Networks*, *19*, 230–244.

Vendramin, L., Campello, R. J. G. B., & Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, *3*, 209–235.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (1999). Self-Organizing Map in Matlab: the SOM Toolbox. In *Proceedings of the Matlab DSP Conference* (pp. 35–40).

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*, 1628–1644.

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

Wang, L. (1997). On competitive learning. *IEEE Transactions on Neural Networks*, *8*, 1214–1217.

Williamson, J. R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, *9*, 881 – 897.

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2010). Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, *3*, 120–154.

Xu, R., & Wunsch II, D. C. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, *24*, 709–716.

Xu, R., Xu, J., & Wunsch, D. C. (2012). A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*, 1243–1256.

III. A STUDY ON EXPLOITING VAT TO MITIGATE ORDERING EFFECTS IN FUZZY ART

Leonardo Enzo Brito da Silva[1,2] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering,
Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

The clustering structures formed by Adaptive Resonance Theory (ART) and many other algorithms are dependent on input presentation/permutation order. In this work, we exploit Visual Assessment of cluster Tendency (VAT) as a pre-processor for Fuzzy ART in order to mitigate this problem. This approach is a global strategy that uses similarity-based ordering before clustering. Experimental results show that this framework improved peak and average performance, reduced the number of categories, and incurred less variability in the clustering outcome. By enhancing performance and reducing sensitivity to input order presentation, this approach is recommended when it is suitable to perform off-line incremental learning.

1. INTRODUCTION

Unsupervised learning (or clustering) methods employ search control strategies that seek optimal solutions in light of a suitable cost function. By performing this task, one is able to organize or summarize the data. Many clustering approaches have been devised (Xu & Wunsch II, 2005; Xu & Wunsch II, 2009; Xu & Wunsch II, 2010), and, among these, there are incremental learning methods. Some characteristics of such systems are (Fisher, 1993; Giraud-Carrier, 2000; Langley, 1995): the ability to handle data streams (i.e., learn one sample at a time), fast processing and reasonable memory demands (due to time and space constraints), no reprocessing of previously seen samples and being capable of promptly employing current knowledge. Additionally, learning should be performed indefinitely, in principle, ad infinitum (life-long learning) (Wenzel & Förstner, 2009; Wenzel & Hotz, 2010).

Due to the nature of incremental learning, these systems are susceptible to ordering effects (Béjar et al., 1993; Cornuéjols, 1993; Fisher, 1993, 1996; Fisher et al., 1992; Giraud-Carrier, 2000; Langley, 1995; MacGregor, 1988; Mauro et al., 2004, 2005; Roure & Talavera, 1998; Talavera & Roure, 1998; Wang, 1997; Wenzel & Förstner, 2009; Wenzel & Hotz, 2010), i.e., the clusters may differ according to the order of input presentation. This order dependency can reveal itself in three levels, namely in the scope of concepts (categories), instances (samples) and attributes (features) (Langley, 1995). Since incremental learning is an inherent part of human learning, this order dependency has been studied in the computational intelligence, education and cognitive psychology fields (Langley, 1995).

An ideal incremental learning system would be order insensitive (Mauro et al., 2004, 2005; Wenzel & Förstner, 2009; Wenzel & Hotz, 2010); realistically, if it is sensitive, there should exist an input sequence that yields the optimal performance and thus learning should be facilitated by presenting examples in a meaningful order (Wenzel & Förstner, 2009; Wenzel & Hotz, 2010). In this context, there are many different orderings for presenting samples such as presenting batches of classes (same class samples presented consecutively) or alternating them, common or uncommon examples, according to increasing/decreasing levels of complexity as well as specific and general examples (Langley, 1995).

Therefore, finding permutations of samples that optimize performance in order dependent systems is of great interest and, as mentioned previously, many strategies have been presented in the literature to mitigate ordering effects for both supervised and unsupervised incremental learning. The taxonomy disussed in (Roure & Talavera, 1998) classifies these according to their scope (global and local) and application phase while performing the clustering task (before, during, and after). Global methods have access to the entire data in advance. Particularly related to the approach presented in our work is the seed selection and (dis)similarity-based ordering methods, which consists of drawing a random seed and subsequently selecting maximally similar or dissimilar samples iteratively (Fisher, 1993). As opposed to global methods, which are suitable to off-line learning applications, local methods have limited foresight (myopic) and are applied on-line during clustering.

Adaptive Resonance Theory (ART) (Carpenter & Grossberg, 1987) is an example of an incremental learning system and thus it is affected by input sequence dependency, especially when using fast learning. It is a neural network-based clustering method that possesses many useful properties (Wunsch II, 2009), in particular, solving the stability-plasticity dilemma. In order to mitigate this problem in ART, input ordering algorithms have also been presented. Some strategies presented in the literature regarding ART and other incremental learning systems are listed in Table 1.

In this work, we investigate the use of Visual Assessment of cluster Tendency (VAT) (Bezdek & Hathaway, 2002) as a pre-processing stage for Fuzzy ART (Carpenter et al., 1991) in the unsupervised and off-line learning scenario, thereby improving performance and reducing the variance of the results. The remainder of this paper is divided as follows: Section 2 provides a review of Fuzzy ART and VAT; Sections 3, 4, 5 and 7 correspond to methodology, experimental setup, results and discussion and conclusion sections, respectively.

Table 1. Summary of some approaches that mitigate order effects.

Strategy/Approach	Experimental framework	Learning paradigm	Order of effect scope[a]	Classification[b]	Reference(s)
split, merge, (dis)similarity ordering[c], measures to assess category quality, agglomerative method for tree initialization	COBWEB (Fisher, 1987)	Unsupervised	instances	global (pre-processing) & local (during learning)	(Fisher, 1993; Fisher et al., 1992)
buffering ("Not Yet" strategy)	COBWEB-like system	Unsupervised	instances	local (during learning)	(Roure & Talavera, 1998; Talavera & Roure, 1998)
backtraking	INTHELEX (Esposito et al., 2003)	Supervised	instances	local (during learning)	(Mauro et al., 2004, 2005)
class-by-class sequences generated based on estimation of Bayes error bounds	iLDAaPCA (Uray et al., 2007) and Logistic Regression	Supervised	concepts	global (pre-processing)	(Wenzel & Förstner, 2009; Wenzel & Hotz, 2010)
learning rates that provide equal weighting of samples	competitive neural networks	Unsupervised	instances	local (during learning)	(Wang, 1997)
ordering based on the number of 1's	ART1 (Carpenter & Grossberg, 1987)	Unsupervised	instances	global (pre-processing)	(Dagli & Huggahalli, 1993; Rao & Gu, 1995)
Max-Min clustering (Tou & Gonzalez, 1974)	Fuzzy ARTMAP (Carpenter et al., 1992)	Supervised	instances	global (pre-processing)	(Dagher et al., 1998, 1999)
genetic algorithms	Fuzzy ARTMAP (Carpenter et al., 1992)	Supervised	instances	global (pre-processing)	(Baek et al., 2014; Palaniappan & Eswaran, 2009)
uncorrelated feature based ordering	Fuzzy ARTMAP (Carpenter et al., 1992)	Supervised	instances	global (pre-processing)	(Oong & Isa, 2014)
voting strategies	Fuzzy ARTMAP (Carpenter et al., 1992), Default ARTMAP 1 (Carpenter, 2003) and 2 (Amis & Carpenter, 2007)	Supervised	instances	global (post-processing)	(Amis & Carpenter, 2007; Carpenter, 2003; Carpenter et al., 1992; Vigdor & Lerner, 2006)
split, merge, delete	ART-like system	Unsupervised	instances	local (during learning)	(Lughofer, 2008)
validity index based vigilance test	Fuzzy ART (Carpenter et al., 1991)	Unsupervised	instances	global & during learning	(Brito da Silva & Wunsch II, 2017)

[a] Scope based on (Langley, 1995).

[b] Classification based on (Roure & Talavera, 1998).

[c] Dissimilarity was deemed best in this study due to the algorithm biases and the fact that this ordering uniformly samples the data space.

2. BACKGROUND

2.1. FUZZY ART

Fuzzy ART (Carpenter et al., 1991) can handle binary and real-valued data. When a new sample $\boldsymbol{x}$ is presented at the input layer F_1, a winner-takes-all competition takes place over all categories $\boldsymbol{w}_j$ at the output layer F_2, and the best matching category is selected as the one that maximizes the *activation function* T_j:

$$T_j = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|}, \tag{1}$$

where $\alpha > 0$ is called the choice parameter. The category choice function defines the order of search and is biased towards smaller categories. Next, a hypothesis testing cycle with respect to the best matching category i is conducted using the *match function* M_i:

$$M_i = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_i|}{|\boldsymbol{x}|}. \tag{2}$$

The match function tests if a category is able to enclose the sample without surpassing the maximum category size defined by the vigilance parameter ($0 \leq \rho \leq 1$). If the category satisfies this constraint (if $M_i \geq \rho$), then learning is allowed:

$$\boldsymbol{w}_i^{new} = (1 - \beta)\boldsymbol{w}_i^{old} + \beta(\boldsymbol{x} \wedge \boldsymbol{w}_i^{old}), \tag{3}$$

where $0 < \beta \leq 1$ is the learning rate.

Otherwise, this category is reset, the subsequent ranked category is selected, and the process is repeated. If no category satisfies this constraint then a new one is created to represent this input pattern. Fuzzy ART features fast, stable, plastic, incremental on-line and off-line learning. One of the fast learning consequences is precisely the dependency on the order of presentation of samples.

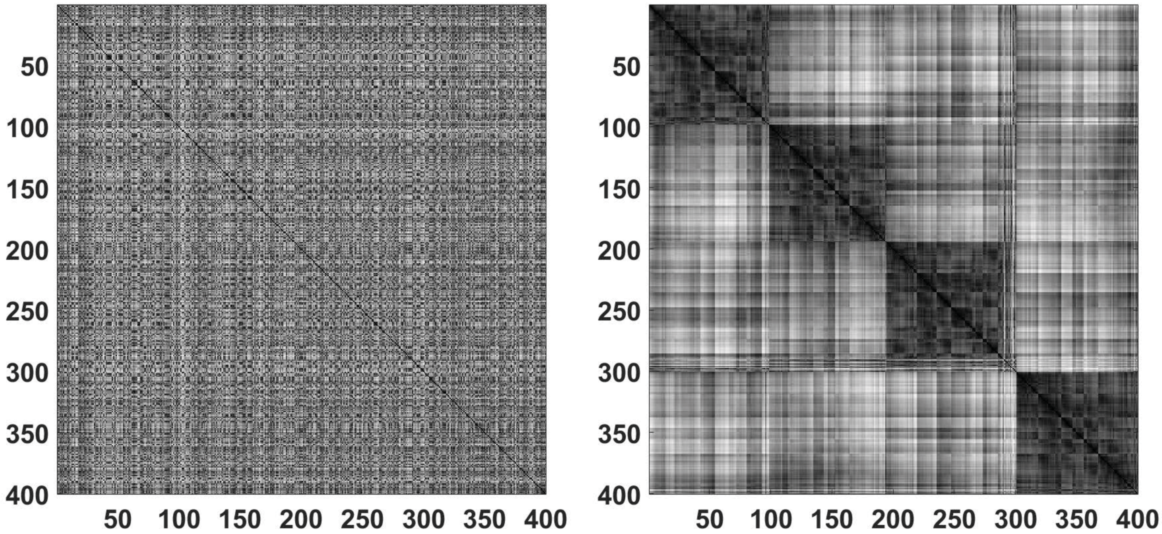
2.2. VISUAL ASSESSMENT OF CLUSTER TENDENCY (VAT)

The Visual Assessment of cluster Tendency (VAT) (Bezdek & Hathaway, 2002) is a visualization technique that consists of a rearranged dissimilarity matrix that displays, as a heat map, the pairwise distance between samples, where the distinctive feature is that dark blocks represent cluster tendency. Briefly, after computing a dissimilarity measure between all samples (e.g., the Euclidean distance), then one of the farthest samples is selected as a starting point (or seed) (Bezdek & Hathaway, 2002). Next, remaining samples are iteratively added based on the minimum distance to any sample in the current growing subset in a single linkage hierarchical clustering algorithm fashion (VAT is related to Prim's minimum spanning tree (Prim, 1957) and is $O(n^2)$) (Bezdek, 2017; Bezdek & Hathaway, 2002; Havens & Bezdek, 2012; Havens et al., 2009b). The indices of this sequential inclusion of samples are saved and used to reorder the data's pairwise distance matrix, which is then depicted as a gray-level image. For convenience, Algorithm 3 reproduces the VAT method, while Figure 1 illustrates an example of VAT of the *Tetra* data set (Ultsch, 2005) shown in Figure 3b.

Algorithm 3: VAT (Bezdek, 2017; Bezdek & Hathaway, 2002; Hathaway et al., 2006; Havens & Bezdek, 2012; Havens et al., 2009b, 2013)

```
Input  : Data (dis)similarity matrix $D_{N \times N}$.
Output : VAT reordered (dis)similarity matrix $\tilde{D}_{N \times N}$.

Initialization: $I \leftarrow \emptyset$, $J \leftarrow \{1, ..., N\}$, $P \leftarrow \emptyset$.
Seeding: $(i, j) \leftarrow \underset{r \in J\ ,\ s \in J}{\arg\max} \{D_{r,s}\}$, $P(1) \leftarrow \{i\}$, $I \leftarrow I \cup \{i\}$, $J \leftarrow J - \{i\}$.
for $t \in \{2, ..., N\}$ do
    $(i, j) \leftarrow \underset{r \in I\ ,\ s \in J}{\arg\min} \{D_{r,s}\}$, $P(t) \leftarrow \{j\}$, $I \leftarrow I \cup \{j\}$, $J \leftarrow J - \{j\}$.
Reorder: use indices $P$ to reorder $D$ and generate $\tilde{D}$.
```



(a) random order image

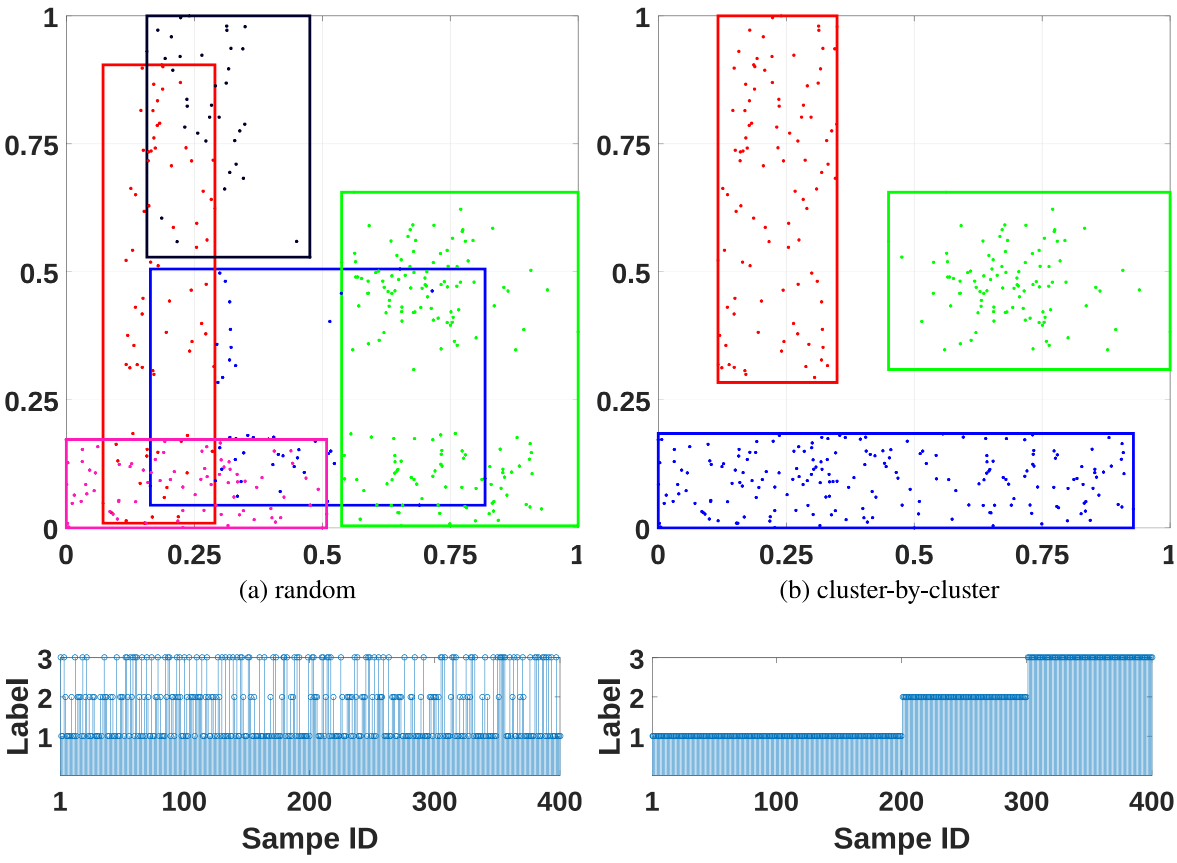
(b) VAT image

Figure 1. Euclidean distance matrix of the *Tetra* data set before and after (VAT) reordering. Dark blocks in the main diagonal indicate cluster tendency.

3. METHODOLOGY

Consider the data set *Lsun* (Ultsch, 2005). Assuming the clusters are known, via an oracle, and samples are presented to Fuzzy ART in a *cluster-by-cluster* fashion (Figs. 2b and 2d), the clustering task becomes fairly easy. This is in accordance with a two class ordering study in (Clapper & Bower, 1994) as discussed in (Langley, 1995). On the other hand, in a realistic scenario, in which samples are randomly presented, then Figs. 2a and 2c depict the best performance obtained after a vigilance parameter grid search (given this order of presentation).

In clustering applications, prior to the clustering algorithm selection, usually a visualization method is used to provide the data analyst an insight on the number and sizes of clusters, thus approximating the data distribution. This aids the practitioner to select the clustering algorithm parameters such as the number of clusters, which is required for many of such algorithms, and also biases one's expectation regarding the clustering outcome. One of the visualization techniques that has been extensively used is VAT.



(a) random

(b) cluster-by-cluster

(c) random

(d) cluster-by-cluster

Figure 2. Illustration of Fuzzy ART's input order dependency using *Lsun* data set (Ultsch, 2005): the clustering solution on the right side (cluster-by-cluster presentation) is perceptibly better than the one on the left hand side (random input presentation).

Examples of ART-like frameworks that make use of VAT for clustering are (Dakshayani et al., 2016; Lam et al., 2015; Sledge & Keller, 2008; Sledge et al., 2008; Srinivasulu & Dakshayani, 2016); however, VAT ordering ability is not exploited in these settings so as to improve the clustering algorithm performance but only for visualization and validation purposes, as originally intended. On the other hand, since VAT aligns single linkage partitions (Havens et al., 2009b), then this ordering property may be exploited beyond data visualization, e.g., for clustering purposes such as in (Havens et al., 2009a, 2013).

Therefore, the framework presented here consists of pre-processing the data by using the VAT ordering prior to feeding it to the Fuzzy ART network, thus this pre-processing stage is completely parameter free. This is an off-line learning approach, since VAT needs access to the entire data set. Additionally, it falls under similarity-based ordering as discussed in (Fisher, 1993). Naturally, one could use the ordering approach introduced in (Dagher et al., 1998, 1999), where the required number of clusters parameter could be estimated using VAT; however, this would require performing clustering twice: once to obtain the ordering using the Max-Min algorithm (Tou & Gonzalez, 1974) and then again with Fuzzy ART. Additionally, similarly to VAT, this ordering method selects an outermost sample as the initial seed.

To use this framework, we assume that it is possible to at least sample the data set, store a subset and perform off-line learning, prior to deploying Fuzzy ART. Here we use the standard VAT with Euclidean distance metric (with normalized data and prior to applying complement coding (Carpenter et al., 1991)). Different orderings may be obtained given the (dis)similarity measure chosen, such as iVAT (Havens & Bezdek, 2012; Wang et al., 2010) that uses a path-based similarity measure, or other VAT variants.

4. EXPERIMENTAL SETUP

In this study, MATLAB and the CVAP toolbox (Wang et al., 2009) were used. A mix of artificial and real world data sets (listed in Table 2) were used in the experiments. The latter consisted of 50 runs with a vigilance parameter sweep analysis using a step size of 10^{-2} for each data set. The data sets' samples were shuffled and directly presented to a Fuzzy ART (network 1: the control group used to assess baseline performance) while another Fuzzy ART (network 2: the experimental group) was fed the reordered samples generated by the VAT pre-processing stage (the input to VAT is the same randomized samples fed to the network 1). In other words, in each run, the data was shuffled and presented to two systems: Fuzzy ART (System 1) and VAT + Fuzzy ART (System 2).

Table 2. Data sets' summary.

Data set	N[a]	D[a]	K[a]	Reference(s)
Chainlink	1000	3	2	(Ultsch, 2005)
Lsun	400	2	3	(Ultsch, 2005)
Tetra	400	3	4	(Ultsch, 2005)
Dermatology	358	34	6	(Bache & Lichman, 2013)
Ecoli	336	7	8	(Bache & Lichman, 2013)
Seeds[b]	210	7	3	(Bache & Lichman, 2013; Charytanowicz et al., 2010)
Face	320	2	4	(Ilc, 2013; Ilc & Dobnikar, 2011)
Flame	240	2	2	(Fränti, Pasi et al., 2015; Fu & Medico, 2007)

[a] N: number of samples, D: dimensionality, K: number of clusters. For detailed descriptions of these data sets refer to the reference(s) column.
[b] Contributors gratefully acknowledge support of their work by the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

The data sets' features were scaled to the range [0, 1] and, subsequently, complement coding (Carpenter et al., 1991) was applied. Fuzzy ART was trained with fast learning ($\beta = 1$), choice parameter (α) equal to 10^{-3} and a single epoch (one pass through the data). The Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) measured the agreement between the output and reference partitions:

$$ARI = \frac{\binom{N}{2}(tp + tn) - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}{\binom{N}{2}^2 - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}, \tag{4}$$

where tp, tn, fp and fn stand for true positive, true negative, false positive and false negative, respectively. MATLAB code for Fuzzy ART and VAT is available at the Applied Computational Intelligence Laboratory public GitLab repository (Brito da Silva & Wunsch II, 2018).

5. RESULTS AND DISCUSSION

The performances of both systems in terms of ARI were recorded over 50 runs and then averaged for each vigilance parameter value (average performance). In practical applications, we assume that a user is able to make, to some extent, an informed guess regarding the value of the vigilance parameter (ρ), given the availability of expert domain knowledge, their experience, or by optimizing a suitable cost function such as validity indices (Xu & Wunsch II, 2009, Sec. 10). Thus, we analyze the peak average performance (best performing vigilance parameter). Alternatively, if no additional information is available, we also evaluate the average average performance (performance averaged across all vigilance values experimented with), so that we aim to know, on average and regardless of the vigilance chosen, how the systems compare. Both of these performances are reported in Table 3 as $mean \pm standard\ deviation$.

Table 3 also lists the number of different effective permutations (i.e., permutations that cause change in the performance behavior of the ARI versus ρ curve) to which both systems were subjected to during the 50 runs of the experiments. The system with standard Fuzzy ART was fed with 50 different permutations, whereas the system with VAT preprocessing was presented with a considerably smaller number of input sequences: the source for the variance relates to the fact that there may be pairs of data samples with the same distance over the dissimilarity matrix (Bezdek & Hathaway, 2002). We did not employ a tie breaking strategy; instead these points were picked given the initial data randomization: both the starting points, which are the ones that are pairwise farthest apart (there are at least two initial points), and the points to be selected throughout the VAT ordering procedure.

Figure 3 illustrates the clustering results for selected $3D$ data sets. It is visually noticeable that the procedure is able to significantly improve performance. The full results obtained after performing the experiments outlined in Section 4 are depicted in Figs. 4 and 5 regarding the ARI performance, and, in Figs. 6 and 7, regarding the number of clusters. Order sensitivity may be measured according to the variance of a method's performance

Table 3. Results summary.

Data set	VAT + Fuzzy ART					Fuzzy ART					$\hat{\mathbb{E}}[diff]$
	$\mu_{avg} \pm \sigma_{avg}$	$\mu_{max} \pm \sigma_{max}$	#categories	ρ	#permutations	$\mu_{avg} \pm \sigma_{avg}$	$\mu_{max} \pm \sigma_{max}$	#categories	ρ	#permutations	
Chainlink	0.35 ± 0.27	0.94 ± 0.03	2.00 ± 0.00	0.40	2	0.15 ± 0.08	0.36 ± 0.24	5.34 ± 1.76	0.40	50	6.61
Dermatology	0.25 ± 0.18	0.60 ± 0.07	10.00 ± 0.00	0.48	50	0.11 ± 0.06	0.20 ± 0.07	20.00 ± 2.89	0.45	50	8.90
Ecoli	0.15 ± 0.14	0.51 ± 0.00	21.92 ± 2.02	0.73	2	0.08 ± 0.07	0.21 ± 0.05	22.38 ± 2.16	0.71	50	1.85
Face	0.18 ± 0.11	0.40 ± 0.37	4.00 ± 0.00	0.29	2	0.15 ± 0.14	0.54 ± 0.21	2.74 ± 0.78	0.23	50	1.14
Flame	0.20 ± 0.13	0.46 ± 0.02	4.48 ± 0.50	0.42	2	0.12 ± 0.06	0.20 ± 0.07	7.04 ± 1.19	0.49	50	1.13
Lsun	0.36 ± 0.29	0.94 ± 0.07	3.00 ± 0.00	0.44	2	0.23 ± 0.12	0.40 ± 0.22	5.18 ± 0.94	0.44	50	2.70
Seeds	0.25 ± 0.20	0.64 ± 0.02	6.88 ± 1.00	0.55	2	0.18 ± 0.12	0.38 ± 0.08	5.12 ± 1.04	0.41	50	2.32
Tetra	0.35 ± 0.27	0.99 ± 0.00	4.00 ± 0.00	0.52	4	0.19 ± 0.11	0.42 ± 0.07	14.70 ± 1.81	0.60	50	5.52

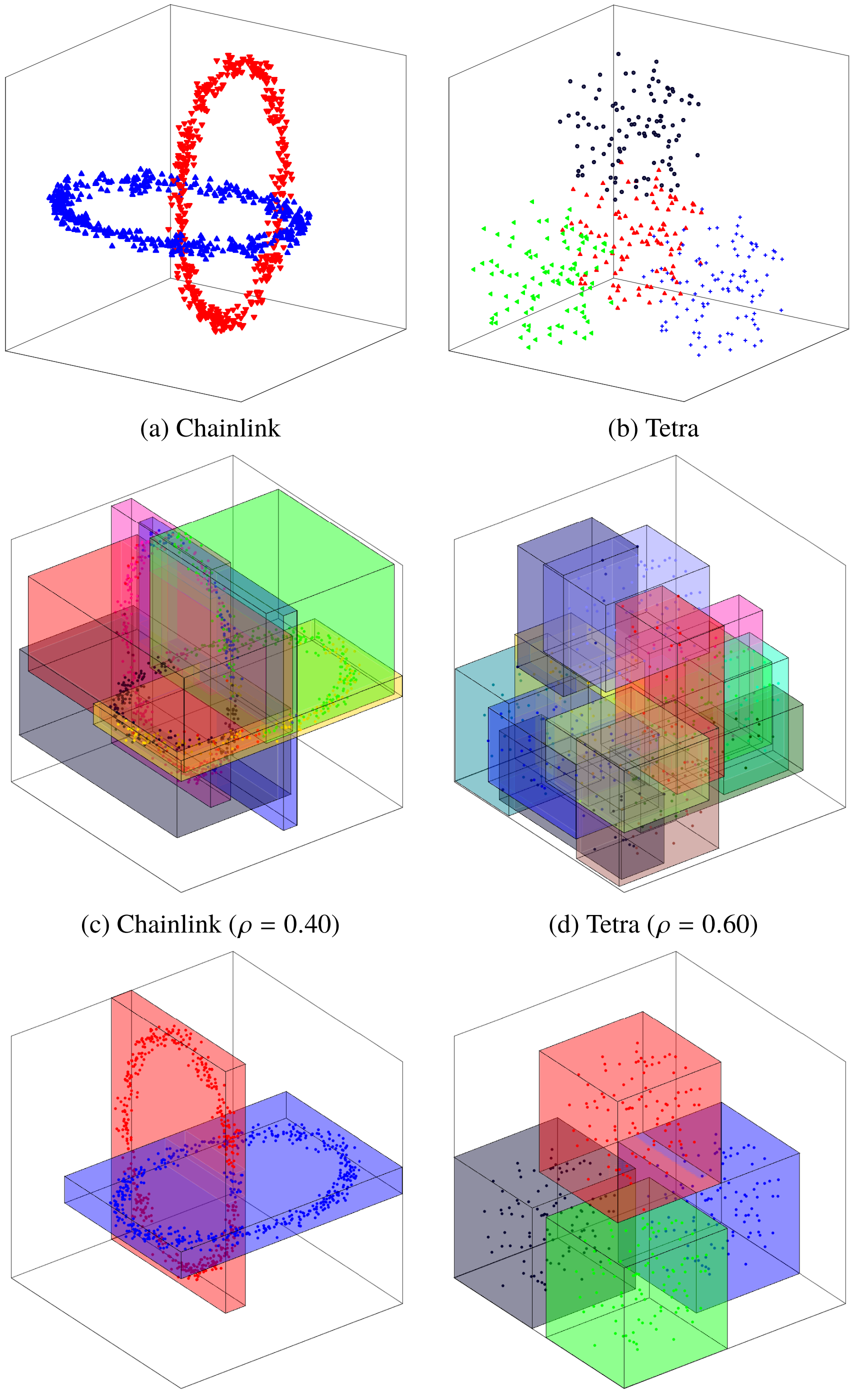
The subscripts 'avg' and 'max' refer to average average and peak average performances (*ARI*), respectively. The values of the vigilance parameters (ρ) are listed with respect to the peak average performance. The number of effective permutations and expected difference of the number of clusters created is also reported.

given the different input sequences (Langley, 1995; Wenzel & Förstner, 2009; Wenzel & Hotz, 2010). For most data sets, we observe that the variability of the clustering outcome is considerably mitigated by using the VAT ordering procedure (Figs. 4 and 6). The sensitivity analysis of the different VAT orderings are depicted in Figure 5. Most of these orderings resulted in a similar performance behavior over the vigilance parameter space; except for the *Face* and *Flame* data sets, despite the fact that they were only presented with 2 different orderings. Interestingly, the *Dermatology* data set was presented with 50 different orderings (due to the reasons discussed above), and yet the performance behaviors with respect to the vigilance parameter are very consistent.

Additionally, Table 3 lists the expected difference of the number of categories between both frameworks:

$$\hat{\mathbb{E}}[diff] = \frac{1}{N_\rho N_{runs}} \sum_{\forall \rho} \sum_{\forall runs} (N_{FA} - N_{VAT+FA}), \tag{5}$$

where N_ρ, N_{runs}, N_{FA}, N_{VAT+FA} are the number parameterizations (101), the number of runs (50) and number of categories in both systems, respectively. The computations were performed with respect to the same value of the vigilance parameter (ρ). According to Table 3, since the expected value is positive, we may infer that ordering with VAT leads to a smaller number of categories for the data sets experimented with. Figure 7 illustrates $\hat{\mathbb{E}}[diff|\rho]$ along with the estimated conditional standard deviations for each data set.



(a) Chainlink

(b) Tetra

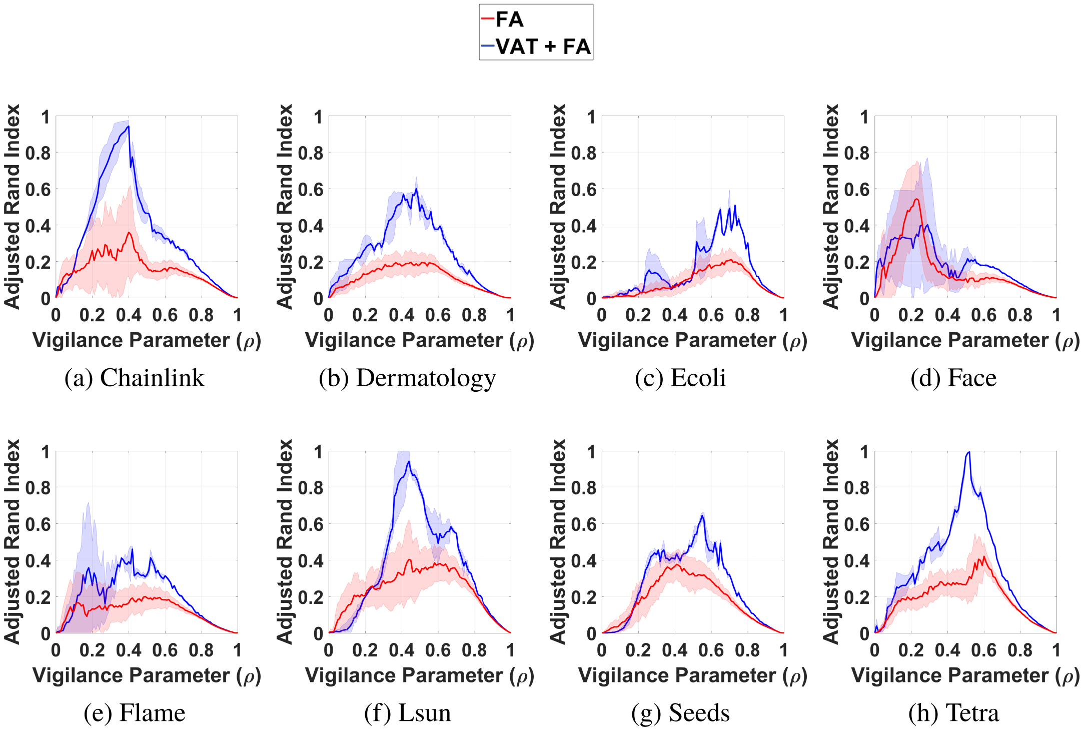
(c) Chainlink (ρ = 0.40)

(d) Tetra (ρ = 0.60)

(e) Chainlink (ρ = 0.40)

(f) Tetra (ρ = 0.52)

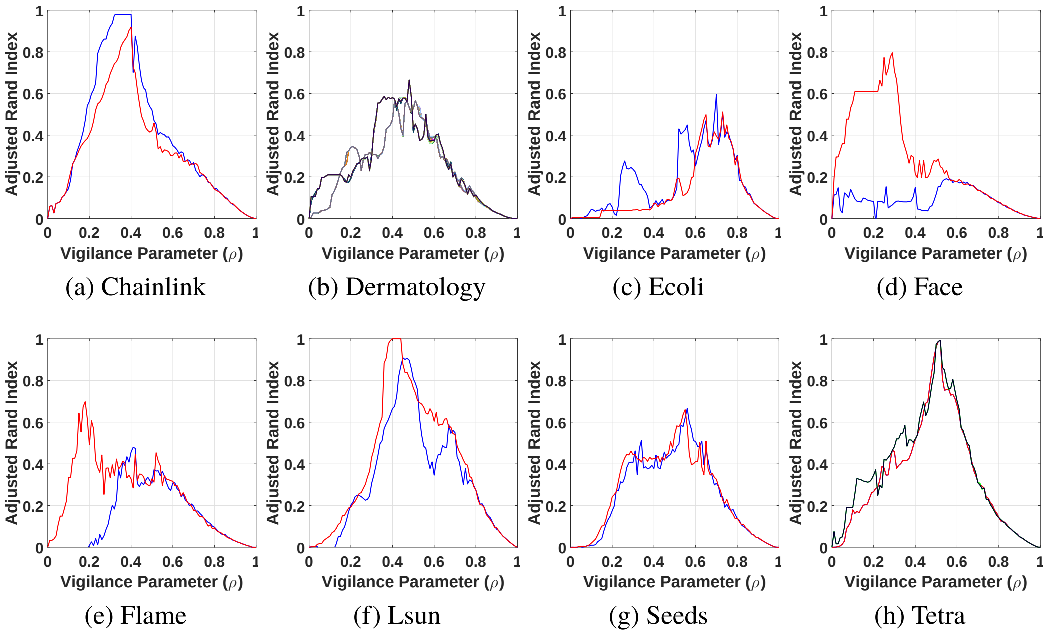
Figure 3. First row {(a), (b)}: selected data sets. Second row {(c), (d)}: best clustering solutions of Fuzzy ART (system 1). Third row {(e), (f)}: best clustering solutions of VAT + Fuzzy ART (system 2). The best solutions were selected using the vigilance parameter (ρ) associated with the peak average *ARI* reported in Table 3 for one of the 50 orders of presentation.



(a) Chainlink (b) Dermatology (c) Ecoli (d) Face

(e) Flame (f) Lsun (g) Seeds (h) Tetra

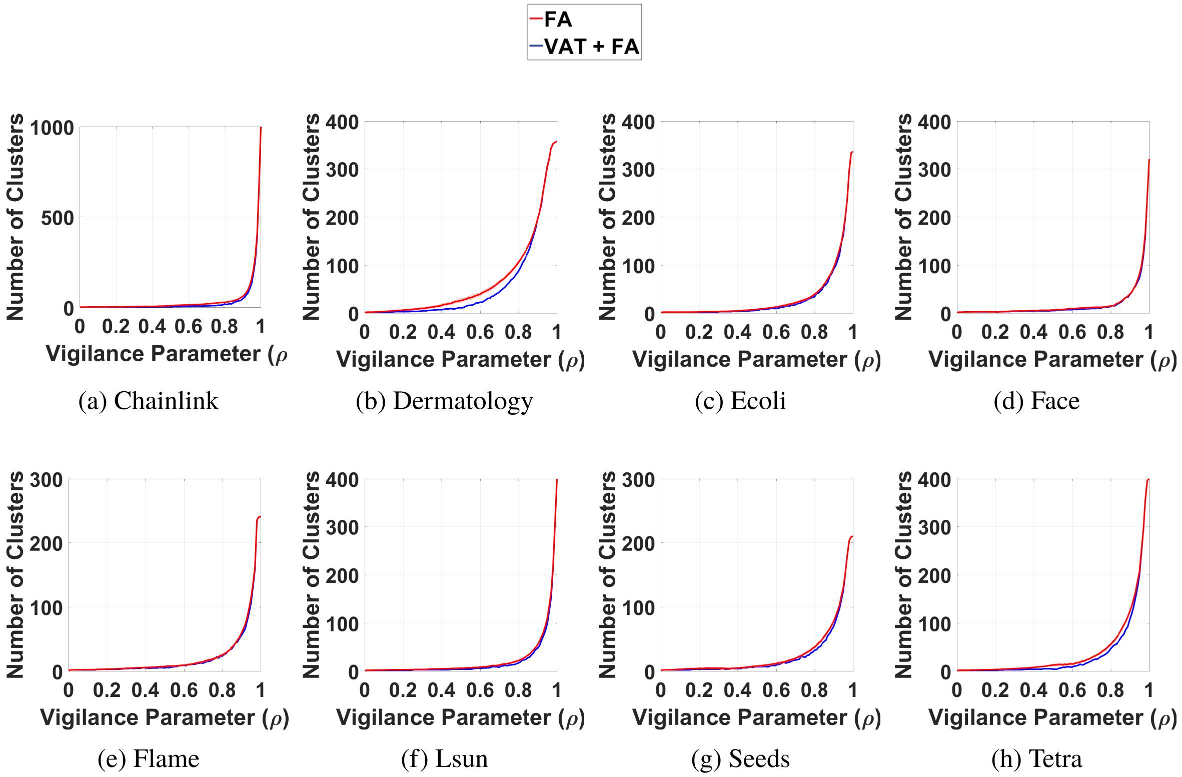
Figure 4. Parameter sweep analysis of the vigilance for the systems with (blue) and without (red) VAT reordering as a pre-processing. The mean and standard deviations (shaded areas) of the *ARI* are depicted.



(a) Chainlink (b) Dermatology (c) Ecoli (d) Face

(e) Flame (f) Lsun (g) Seeds (h) Tetra

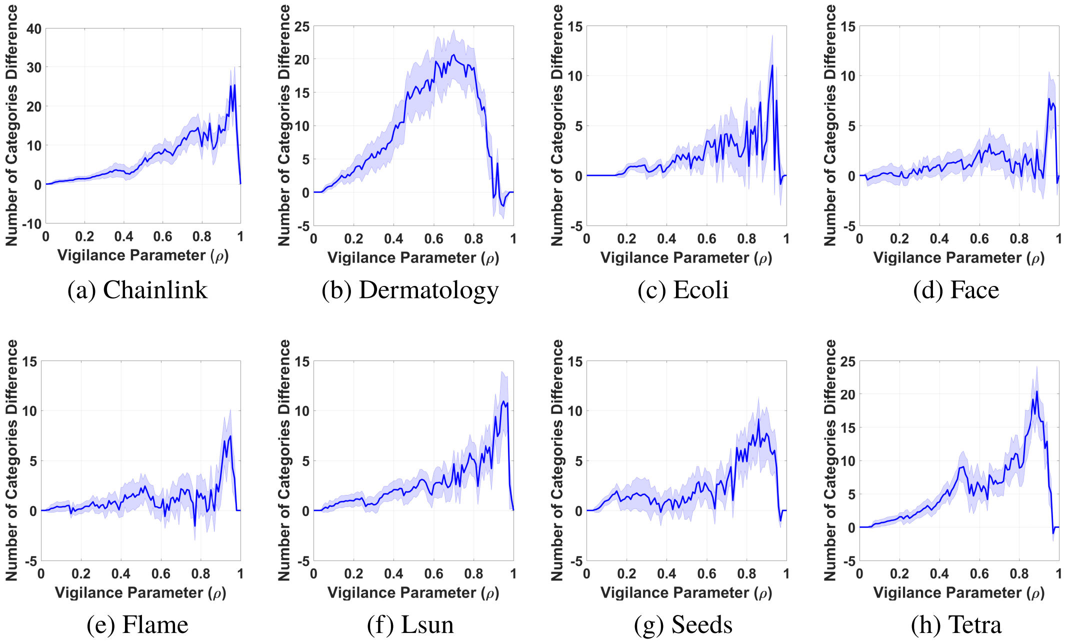
Figure 5. Performance of Fuzzy ART when using the different VAT re-orderings (see Table 3) as pre-processing. Each colored curve represent a specific permutation order presented to the system VAT + Fuzzy ART.



(a) Chainlink (b) Dermatology (c) Ecoli (d) Face

(e) Flame (f) Lsun (g) Seeds (h) Tetra

Figure 6. Parameter sweep analysis of the vigilance for the systems with (blue) and without (red) VAT reordering as a pre-processing. The mean and standard deviations (shaded areas) of the number of clusters are depicted.



(a) Chainlink (b) Dermatology (c) Ecoli (d) Face

(e) Flame (f) Lsun (g) Seeds (h) Tetra

Figure 7. Mean and standard deviations (shaded areas) of the difference between the number of clusters of the systems Fuzzy ART and VAT + Fuzzy ART.

Since this ordering procedure works consistently for most, but not all, data sets (e.g., *Face* and *Flame* data sets), we employed Wilcoxon signed-ranks test (Wilcoxon, 1945) to statistically compare both frameworks, as recommended in (Demšar, 2006). We observe that there is a statistical difference between these under a 0.1 significance level considering (a) peak average performance (p-value: 0.0209), (b) average average performance (p-value: 0.0143), and (c) the number of clusters created under the peak average performance parameterization (p-value: 0.0801). Thus, for the instances of the classes of problems this framework is applicable to, the performance seems to be significantly enhanced.

Direct application of VAT to large data sets is not practical due to (a) the computational costs associated with the ordering method by itself and the calculations of pairwise (dis)similarities for all samples, and (b) the storage requirements of the (dis)similarity matrix (Bezdek, 2017; Hathaway et al., 2006; Havens et al., 2013). In order to overcome this limitation, one could resort to sampling strategies such as (Vitter, 1985) and Maximin (Hathaway et al., 2006) (which is closely related to the ordering procedure in (Dagher et al., 1998, 1999)), since sampling methods have been successfully employed for both visualization (Hathaway et al., 2006) and clustering (Guha et al., 1998; Havens et al., 2013; Wang et al., 2011) purposes. We also note that our study does not address noisy data sets.

6. CONCLUSION

This work presented a study on the VAT + Fuzzy ART framework to alleviate input order dependency. Results show that it yields statistically significant improved peak and average performances as well as smaller number of categories generated; besides it incurred less variability in the clustering outcome on many data sets. Therefore, if at all possible, reordering the samples using VAT is recommended, as this pre-processing is able to significantly affect the quality of the clustering structures created by Fuzzy ART. We expect that this framework will improve performance on other incremental clustering approaches also.

ACKNOWLEDGEMENTS

Support from the National Science Foundation, the Missouri University of Science and Technology Intelligent Systems Center, and the Mary K. Finley Endowment is gratefully acknowledged. Leonardo Enzo Brito da Silva was supported by the CAPES Foundation, Ministry of Education of Brazil, under grant number BEX 13494/13-9.

REFERENCES

Amis, G. P., & Carpenter, G. A. (2007). Default ARTMAP 2. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 777–782).

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Baek, J., Lee, H., Lee, B., Lee, H., & Kim, E. (2014). An efficient genetic selection of the presentation order in simplified fuzzy ARTMAP patterns. *Applied Soft Computing*, *22*, 101–107.

Béjar, J., Cortés, U., & Poch, M. (1993). *LINNEO+: A classification methodology for ill-structured domains*. Technical Report LSI-93-22-R Facultat d'InformÃătica de Barcelona.

Bezdek, J. C. (2017). *A Primer on Cluster Analysis: 4 Basic Methods that (usually) Work*. First Edition Design Publishing.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Brito da Silva, L. E., & Wunsch II, D. C. (2017). Validity Index-based Vigilance Test in Adaptive Resonance Theory Neural Networks. In *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Brito da Silva, L. E., & Wunsch II, D. C. (2018). VAT + Fuzzy ART Framework. URL: `https://git.mst.edu/acil-group/VAT-Fuzzy-ART`.

Carpenter, G. A. (2003). Default ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1396–1401). volume 2.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. (2010). Complete Gradient Clustering Algorithm for Features Analysis of X-Ray Images. In E. Pietka, & J. Kawa (Eds.), *Information Technologies in Biomedicine* (pp. 15–24). Berlin, Heidelberg: Springer Berlin Heidelberg.

Clapper, J. P., & Bower, G. H. (1994). Category invention in unsupervised learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 443 – 460.

Cornuéjols, A. (1993). Getting order independence in incremental learning. In P. B. Brazdil (Ed.), *Machine Learning: ECML-93* (pp. 196–212). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1998). Ordered fuzzy ARTMAP: a fuzzy ARTMAP algorithm with a fixed order of pattern presentation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1717–1722). volume 3.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1999). An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance. *IEEE Transactions on Neural Networks*, *10*, 768–778.

Dagli, C. H., & Huggahalli, R. (1993). A neural network approach to group technology. In J. Wang, & Y. Takefuji (Eds.), *Neural Networks in Design and Manufacturing* (pp. 1–55). Singapore: World Scientific.

Dakshayani, G., Srinivasulu, A., & Haneesh, K. (2016). Study on Clustering Large Data Using Fuzzy Adaptive Resonance Theory. *International Research Journal Of Engineering And Technology (IRJET)*, *3*, 1673–1677.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Esposito, F., Ferilli, S., Fanizzi, N., Basile, T. M. A., & Mauro, N. D. (2003). Incremental multistrategy learning for document processing. *Applied Artificial Intelligence*, *17*, 859–883.

Fisher, D. (1993). Ordering Effects in Incremental Learning. In *Proc. of the 1993 AAAI Spring Symposium on Training Issues in Incremental Learning* SS-93-06 (pp. 35–42). Menlo Park, California: The AAAI Press.

Fisher, D. (1996). Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*, *4*, 147–179.

Fisher, D., Xu, L., & Zard, N. (1992). Ordering Effects in Clustering. In D. Sleeman, & P. Edwards (Eds.), *Machine Learning Proceedings 1992* (pp. 163 – 168). San Francisco (CA): Morgan Kaufmann.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*, 139–172.

Fränti, Pasi et al. (2015). Clustering datasets. URL: `http://cs.uef.fi/sipu/datasets/` accessed on May 4, 2017.

Fu, L., & Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, *8*, 1–15.

Giraud-Carrier, C. (2000). A Note on the Utility of Incremental Learning. *AI Communications*, *13*, 215–223.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An Efficient Clustering Algorithm for Large Databases. In *Proc. ACM SIGMOD Int. Conf. Manag. Data* (pp. 73–84). New York, NY, USA: ACM.

Hathaway, R. J., Bezdek, J. C., & Huband, J. M. (2006). Scalable visual assessment of cluster tendency for large data sets. *Pattern Recognition*, *39*, 1315 – 1324.

Havens, T. C., & Bezdek, J. C. (2012). An Efficient Formulation of the Improved Visual Assessment of Cluster Tendency (iVAT) Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *24*, 813–822.

Havens, T. C., Bezdek, J. C., Keller, J. M., & Popescu, M. (2009a). Clustering in ordered dissimilarity data. *International Journal of Intelligent Systems*, *24*, 504–528.

Havens, T. C., Bezdek, J. C., Keller, J. M., Popescu, M., & Huband, J. M. (2009b). Is vat really single linkage in disguise? *Annals of Mathematics and Artificial Intelligence*, *55*, 237–251.

Havens, T. C., Bezdek, J. C., & Palaniswami, M. (2013). Scalable single linkage hierarchical clustering for big data. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (pp. 396–401).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Ilc, N. (2013). Datasets package. URL: `https://www.researchgate.net/publication/239525861_Datasets_package` accessed on Sep 11, 2017.

Ilc, N., & Dobnikar, A. (2011). Gravitational Clustering of the Self-Organizing Map. In A. Dobnikar, U. Lotrič, & B. Šter (Eds.), *Adaptive and Natural Computing Algorithms* (pp. 11–20). Berlin, Heidelberg: Springer volume 6594.

Lam, D., Wei, M., & Wunsch II, D. C. (2015). Clustering Data of Mixed Categorical and Numerical Type With Unsupervised Feature Learning. *IEEE Access*, *3*, 1605–1613.

Langley, P. (1995). Order Effects in Incremental Learning. In P. Reimann, & H. Spada (Eds.), *Learning in Humans and Machines: Towards an Interdisciplinary Learning Science* (pp. 154–165). Emerald Group Publishing Limited.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

MacGregor, J. N. (1988). The Effects of Order on Learning Classifications by Example: Heuristics for Finding the Optimal Order. *Artificial Intelligence*, *34*, 361 – 370.

Mauro, N. D., Esposito, F., Ferilli, S., & Basile, T. M. A. (2004). A Backtracking Strategy for Order-independent Incremental Learning. In *Proceedings of the 16th European Conference on Artificial Intelligence* ECAI'04 (pp. 460–464). Amsterdam, The Netherlands, The Netherlands: IOS Press.

Mauro, N. D., Esposito, F., Ferilli, S., & Basile, T. M. A. (2005). Avoiding Order Effects in Incremental Learning. In S. Bandini, & S. Manzoni (Eds.), *AI*IA 2005: Advances in Artificial Intelligence* (pp. 110–121). Berlin, Heidelberg: Springer Berlin Heidelberg.

Oong, T. H., & Isa, N. A. M. (2014). Feature-Based Ordering Algorithm for Data Presentation of Fuzzy ARTMAP Ensembles. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 812–819.

Palaniappan, R., & Eswaran, C. (2009). Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. *Applied Soft Computing*, *9*, 100–106.

Prim, R. C. (1957). Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal*, *36*, 1389–1401.

Rao, H. A., & Gu, P. (1995). A multi-constraint neural network for the pragmatic design of cellular manufacturing systems. *International Journal of Production Research*, *33*, 1049–1070.

Roure, J., & Talavera, L. (1998). Robust Incremental Clustering with Bad Instance Orderings: A New Strategy. In H. Coelho (Ed.), *Progress in Artificial Intelligence — IBERAMIA 98* (pp. 136–147). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sledge, I. J., & Keller, J. M. (2008). Growing neural gas for temporal clustering. In *2008 19th International Conference on Pattern Recognition* (pp. 1–4).

Sledge, I. J., Keller, J. M., & Alexander, G. L. (2008). Emergent trend detection in diurnal activity. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 3815–3818).

Srinivasulu, A., & Dakshayani, G. (2016). Clustering Large Data with Mixed Values Using Extended Fuzzy Adaptive Resonance Theory. *Indonesian Journal of Electrical Engineering and Computer Science*, *4*, 617–628.

Talavera, L., & Roure, J. (1998). A buffering strategy to avoid ordering effects in clustering. In C. Nédellec, & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (pp. 316–321). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tou, J. T., & Gonzalez, R. C. (1974). *Pattern recognition principles*. Addison-Wesley,.

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Uray, M., Skočaj, D., Roth, P. M., & Bischof, H. (2007). Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches. In *Proceedings of the British Machine Vision Conference 2007, University of Warwick, UK* (pp. 1–10).

Vigdor, B., & Lerner, B. (2006). Accurate and Fast Off and Online Fuzzy ARTMAP-Based Image Classification With Application to Genetic Abnormality Diagnosis. *IEEE Transactions on Neural Networks*, *17*, 1288–1300.

Vitter, J. S. (1985). Random Sampling with a Reservoir. *ACM Trans. Math. Softw.*, *11*, 37–57.

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

Wang, L. (1997). On competitive learning. *IEEE Transactions on Neural Networks*, *8*, 1214–1217.

Wang, L., Leckie, C., Kotagiri, R., & Bezdek, J. (2011). Approximate pairwise clustering for large data sets via sampling plus extension. *Pattern Recognition*, *44*, 222 – 235.

Wang, L., Nguyen, U. T. V., Bezdek, J. C., Leckie, C. A., & Ramamohanarao, K. (2010). iVAT and aVAT: Enhanced Visual Analysis for Cluster Tendency Assessment. In M. J. Zaki, J. X. Yu, B. Ravindran, & V. Pudi (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 16–27). Berlin, Heidelberg: Springer Berlin Heidelberg.

Wenzel, S., & Förstner, W. (2009). *The Role of Sequences for Incremental Learning*. Technical Report TR-IGG-P-2009-04 Department of Photogrammetry, University of Bonn.

Wenzel, S., & Hotz, L. (2010). The Role of Sequences for Incremental Learning. In J. Filipe, A. L. N. Fred, & B. Sharp (Eds.), *ICAART 2010 - Proceedings of the International Conference on Agents and Artificial Intelligence* (pp. 434–439). Valencia, Spain: INSTICC Press volume 1.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*, 80–83.

Wunsch II, D. C. (2009). ART properties of interest in engineering applications. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 3380–3383).

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2010). Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, *3*, 120–154.

IV. DUAL VIGILANCE FUZZY ADAPTIVE RESONANCE THEORY

Leonardo Enzo Brito da Silva[1,3], Islam Elnabarawy[2] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering,
Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]Applied Computational Intelligence Lab., Dept. of Computer Science,
Missouri University of Science and Technology, Rolla, MO 65409 USA.

[3]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

Clusters retrieved by generic Adaptive Resonance Theory (ART) networks are limited to their internal categorical representation. This study extends the capabilities of ART by incorporating multiple vigilance thresholds in a single network: stricter (data compression) and looser (cluster similarity) vigilance values are used to obtain a many-to-one mapping of categories-to-clusters. It demonstrates this idea in the context of Fuzzy ART, presented as Dual Vigilance Fuzzy ART (DVFA), to improve the ability to capture clusters with arbitrary geometry. DVFA outperformed Fuzzy ART for the datasets in our experiments while yielding a statistically-comparable performance to another more complex, multi-prototype Fuzzy ART-based architecture.

Keywords: Clustering, Adaptive Resonance Theory, ART, Visual Assessment of Cluster Tendency, Topology, Unsupervised.

1. INTRODUCTION

Adaptive Resonance Theory (ART) (Carpenter & Grossberg, 1987) is a learning theory introduced to address the stability-plasticity dilemma. It has inspired many neural network architectures. These share mechanisms where resonance and resets are triggered by a vigilance test under the control of an orienting subsystem. The various architectures differ in their internal categorical representations, which restrict the shapes of the clusters they can retrieve. This paper contributes a simple and effective method for retrieving arbitrary clusters using dual vigilance parameters.

The literature contains many clustering approaches that harness the multi-prototype representation power to capture arbitrarily-shaped clusters. For ART-based systems, hierarchical frameworks were presented in (Su & Liu, 2005) using quadratic neurons (Su & Liu, 2001) and in (Brito da Silva & Wunsch II, 2015) using Fuzzy ART (Carpenter et al., 1991). Notably, TopoART (Tscherepanow, 2010) and its variants (Tscherepanow et al., 2011, 2012) have an architecture that makes use of multiple ART building blocks coupled with topology-based learning (Furao & Hasegawa, 2006). Other approaches that augment ART with topology learning include (Isawa et al., 2008, 2007).

Layered vigilance parameters can generate complex ART-based systems (e.g., for mixed-modality learning) (Seiffertt & Wunsch II, 2010). This paper augments Fuzzy ART by adding a second vigilance test to enable multi-prototype representation in a single ART module. Since its building block is Fuzzy ART, it inherits properties such as fast stable incremental learning and sensitivity to input order.

2. BACKGROUND AND RELATED WORK

2.1. FUZZY ART

Fuzzy ART (Carpenter et al., 1991) is a two-layered neural network connected by a set of adaptable weights $\boldsymbol{w}$: the normalized and complement-coded input ($\boldsymbol{x} \leftarrow [\boldsymbol{x}, \boldsymbol{1} - \boldsymbol{x}]$, $0 \leq x_i \leq 1 \; \forall i$) is presented to the F_1 layer, and the discovered categories are represented via the F_2 layer neurons. Fuzzy ART is controlled by the choice parameter ($\alpha > 0$), learning rate ($\beta \in (0, 1]$) and vigilance parameter ($\rho \in [0, 1]$). The algorithm proceeds as follows:

1. Present input $\boldsymbol{x}$ to the F_1 layer and calculate the choice function T_j for each F_2 category j:

$$T_j = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|}, \tag{1}$$

$$\boldsymbol{p} \wedge \boldsymbol{q} \equiv \boldsymbol{p} \text{ AND } \boldsymbol{q} : (\boldsymbol{p} \wedge \boldsymbol{q})_i \equiv \min(p_i, q_i), \tag{2}$$

$$|\boldsymbol{p}| \equiv \sum_i |p_i|. \tag{3}$$

Then, select the winning category using a winner-take-all competition:

$$J = \arg\max_j \{T_j\}. \tag{4}$$

2. Perform a vigilance check using the match criterion ν_1:

$$\nu_1 : M_J = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_J|}{|\boldsymbol{x}|} \geq \rho. \tag{5}$$

If the winning category satisfies ν_1, then update its weights:

$$\boldsymbol{w}_J^{new} = (1 - \beta)\boldsymbol{w}_J^{old} + \beta(\boldsymbol{x} \wedge \boldsymbol{w}_J^{old}). \tag{6}$$

3. If the winning category fails ν_1, then reset it and repeat 2 until a winner passes. If no existing category succeeds, then a new category is created.

2.2. FUZZY TOPOART

The primary comparison architecture used in this paper is Fuzzy TopoART (Tscherepanow, 2010), which combines Fuzzy ART and topology-based learning (Furao & Hasegawa, 2006). Standard TopoART uses two identical Fuzzy ART modules (A and B) that process data in parallel. It is controlled by module A's vigilance parameter (ρ_a), the learning rate of the second winner (β_{sbm}), the minimum number of samples learned (ϕ) and the number of cycles between noise removal procedures (τ). The algorithm is similar to Fuzzy ART (in the following $i = \{a, b\}$ represents the modules):

1. If the current iteration equals $k\tau$ ($k \in \mathbb{N}^*$), then remove all categories j such that $n^i_j < \phi$ (n^i_j represents the number of samples encoded by category j of module i).

2. Present input $\boldsymbol{x}$ and select the winning category J (Eqs. (1)-(4)).

3. Perform vigilance check ν_1 using ρ_i in place of ρ (Eq. (5)). If satisfied, then update the winner's weights using $\beta_{bm} = 1$ in (Eq. (6)) and increment n^i_J.

   (a) Search for a second winning category that also satisfies ν_1. If such a category exists, update its weights using $\beta_{sbm} < \beta_{bm}$ in (Eq. (6)), and create an edge with category J.

4. If category J fails ν_1, then reset it and repeat 3 until a winner passes. If no existing category passes, then create a new one.

Modules A and B's algorithms are identical. However, a sample is propagated to module B if it has resonated with a category of module A such that $n^a_J \geq \phi$, which serves as a filtering mechanism. Additionally, module B has a higher vigilance parameter that reduces the maximum category size by 50% (Tscherepanow, 2010; Tscherepanow et al., 2011, 2012):

$$\rho_b = \frac{1}{2}\left(\rho_a + 1\right). \tag{7}$$

By enforcing $\rho_b \geq \rho_a$, modules A and B yield coarser and finer partitions of the dataset, respectively. This multi-prototype approach allows both modules to learn topological structures, thus discovering arbitrarily-shaped clusters. Finally, the following activation function is used for prediction (Tscherepanow, 2010; Tscherepanow et al., 2011, 2012):

$$T_j = 1 - \frac{\left|\left(\boldsymbol{x} \wedge \boldsymbol{w}_j\right) - \boldsymbol{w}_j\right|}{|\boldsymbol{x}|}. \tag{8}$$

3. DUAL VIGILANCE FUZZY ART

This paper introduces the idea of using two vigilance thresholds, demonstrated here with Dual Vigilance Fuzzy ART (DVFA)[1] consisting of two layered vigilance parameters that regulate data compression/quantization and cluster similarity, i.e., a tighter and a looser constraint, respectively. DVFA is controlled by upper bound ($\rho^{UB} \in [0, 1]$) and lower bound ($0 \leq \rho^{LB} \leq \rho^{UB} \leq 1$) vigilance parameters and makes use of a binary matrix $\boldsymbol{M}_{map} = \left[m_{r,c}\right]$ to map categories (rows) to clusters (columns), like Fuzzy ARTMAP (Carpenter et al., 1992). The algorithm proceeds as follows:

1. Present input $\boldsymbol{x}$ and select the winning category J (Eq. (1)-(4)).

2. Perform vigilance check v_1 using ρ^{UB} in place of ρ (Eq. (5)). If satisfied, then update the winner's weights using Eq. (6).

3. If v_1 fails, then perform a second test, v_2, using ρ^{LB} in place of ρ (Eq. (5)).

[1] DVFA MATLAB code is available at https://github.com/ACIL-Group/DVFA.

(a) If v_2 is satisfied, then a new category I is created and assigned to the same cluster as category J. This is accomplished by adding a new row I to $\boldsymbol{M}_{map}$ equal to row J. This process can be viewed as "splitting" the parent category.

(b) If v_2 fails, then create a new category I and expand both the rows and columns of $\boldsymbol{M}_{map}$ to encode the new cluster K:

$$m_{r,c} = \begin{cases} 1, & \text{if } r = I \text{ and } c = K \\ 0, & \text{if } r = I \text{ and } c \neq K \\ 0, & \text{if } r \neq I \text{ and } c = K \\ m_{r,c}, & \text{if } r \neq I \text{ and } c \neq K \end{cases} \tag{9}$$

This framework is a multi-prototype approach that builds a many-to-one mapping of categories to clusters using a single ART module. This allows the data distribution to be captured more faithfully so DVFA can retrieve clusters of arbitrary geometries.

4. EXPERIMENTAL SET-UP

The experiments were performed using MATLAB, Orange (Demšar et al., 2013) and LibTopoART[2] (Tscherepanow, 2010). The CVAP toolbox (Wang et al., 2009) was used to compute the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) of the partitions found by the clustering algorithms. Linear normalization and complement coding (Carpenter et al., 1991) were applied to all datasets, which comprise a miscellaneous set of characteristics. Since Fuzzy ART is sensitive to the order of input presentation, the Visual Assessment of cluster Tendency (VAT) (Bezdek & Hathaway, 2002) was used as a pre-processor in part of this study, as it significantly improves the performance of Fuzzy ART and other agglomerative clustering algorithms (Brito da Silva & Wunsch II, 2018).

[2] LibTopoART (version 0.74), available at https://www.libtopoart.eu.

In these experiments, a grid search was used for parameter tuning of the clustering methods. The $[0, 1]$ interval with a step size of 10^{-2} was used to search for the Fuzzy ART vigilance parameter. The lower and upper bound vigilances (ρ^{LB}, ρ^{UB}) of DVFA were scanned in the parameter space $[0, 1] \times [0, 1]$ with a step size of 10^{-2}, while subjected to the constraint $\rho^{UB} \geq \rho^{LB}$. Finally, TopoART parameters were searched in the following intervals: $\rho_a \in [0, 1]$, $\beta_{sbm} \in [0, 0.75]$, $\phi \in [1, 4]$, and $\tau \in [10\%, 30\%]$ with respect to the data cardinality. To ensure a fair comparison, these step sizes were 10^{-2}, 0.25, 1 and 10%, respectively (which is roughly the same number of parameter combinations as DVFA). Module B's output was selected as the final clustering solution.

For each dataset, 30 runs were performed in two different scenarios in which randomized data was: (1) directly presented to the Fuzzy ART-based systems, and (2) pre-ordered using VAT (per (Brito da Silva & Wunsch II, 2018)). The maximum number of epochs, choice parameter (α) and learning rate (β) were set to 1, 10^{-3} and 1, respectively. Moreover, in Fuzzy ART and DVFA implementations, the uncommitted category did not take part in the winner-take-all process. Hence, if none of the current categories satisfy the vigilance constraint, then a new category is created.

5. RESULTS AND DISCUSSION

Table 1 lists the best average performances ($mean \pm standard\ deviation$) achieved by DVFA, TopoART and Fuzzy ART. First, each parameter combination's performances are averaged across all their runs, and then, the maximum average is reported. As expected, pre-ordering noticeably alters the performance of these incremental learners (Brito da Silva & Wunsch II, 2018). The average ranks of Fuzzy ART, DVFA and TopoART observed in the experiments are (a) 2.97, 1.72 and 1.31, in the random presentation case, and (b) 2.72, 1.41 and 1.88 when using VAT, in that order. Therefore, considering the means, the DVFA outperformed Fuzzy ART in most of the datasets in both random and VAT-based pre-processing scenarios. In the latter scenario, DVFA outperformed TopoART more frequently, with the additional advantage of not setting as many parameters since VAT is parameter-free. Moreover, DVFA does not use explicit topological information or rely on multiple networks processing samples in parallel. TopoART achieved the best performance in more instances during random sample presentation.

For most datasets in these experiments, the best DVFA results were achieved when both layers' vigilance parameters were set above 0.6 and relatively close to each other. Specifically, the parameter subspace defined by the cosine distance:

$$1 - cos\left(\begin{bmatrix}\rho^{LB}\\ \rho^{UB}\end{bmatrix}, \begin{bmatrix}1\\ 1\end{bmatrix}\right) \leq c \tag{10}$$

subjected to the constraint $0.6 \leq \rho^{LB} \leq \rho^{UB} \leq 1$ roughly models the region that yields acceptable performance (particularly for random input presentation), where c reflects the size of such a region. Thus, after performing these experiments, this guideline is recommended for parameter setting.

Following Occam's razor principle, Table 1 also lists the average number of categories created by each Fuzzy ART-based method and the average number of clusters retrieved, with respect to the most compressed model, i.e., the one with the smallest number of categories. Naturally, for Fuzzy ART the number of clusters also corresponds to the number of categories. As expected, VAT pre-processing led to a decrease in the number of categories created (Brito da Silva & Wunsch II, 2018). The compression levels of the best average DVFA performances are data- and input order-dependent. For instance, no compression was achieved (i.e, the number of categories is equal to the number of samples) for the *Moon*, *Wine*, *Spiral* and *Synthetic Control* datasets. Conversely, the *Tetra* and *Face* datasets had the most succinct representations. Generally, TopoART provided the most compact clusters, which is expected since it has a more intricate architecture that uses data topology information during the learning process.

Table 1. Experimental results.

Dataset	(N, d, K)[a]	Fuzzy ART		DVFA			TopoART B		
		ARI	#categories	ARI	#categories	#clusters	ARI	#categories	#clusters
			Random input presentation						
Atom (Ultsch, 2005)	(800, 3, 2)	0.51 ± 0.00	55.2 ± 2.1	0.57 ± 0.07	193.8 ± 10.4	8.1 ± 1.4	**0.90 ± 0.06**	81.7 ± 2.3	11.3 ± 2.9
Chainlink (Ultsch, 2005)	(1000, 3, 2)	0.37 ± 0.24	5.8 ± 1.9	0.40 ± 0.16	572.8 ± 8.7	3.9 ± 1.1	**1.00 ± 0.00**	65.1 ± 2.3	2.4 ± 0.7
Face (Ilc, 2013)	(320, 2, 4)	0.53 ± 0.18	2.7 ± 0.8	0.53 ± 0.18	2.7 ± 0.8	2.7 ± 0.8	**0.99 ± 0.01**	24.1 ± 1.6	4.9 ± 1.0
Iris (Bache & Lichman, 2013)	(150, 4, 3)	0.49 ± 0.07	8.5 ± 1.1	**0.65 ± 0.11**	58.1 ± 4.2	3.7 ± 0.8	0.60 ± 0.08	16.3 ± 1.9	4.3 ± 1.5
Jain (Fränti, Pasi et al., 2015; Jain & Law, 2005)	(373, 2, 2)	0.56 ± 0.22	3.1 ± 0.7	0.59 ± 0.27	6.4 ± 2.8	2.3 ± 0.8	**0.75 ± 0.15**	49.0 ± 2.6	8.4 ± 2.0
Lsun (Ultsch, 2005)	(400, 2, 3)	0.44 ± 0.19	5.6 ± 0.9	0.64 ± 0.15	116.6 ± 8.6	4.5 ± 0.9	**0.79 ± 0.13**	61.7 ± 2.4	9.3 ± 2.6
Moon (Ilc, 2013)	(514, 2, 4)	0.28 ± 0.07	5.8 ± 1.1	0.37 ± 0.10	86.5 ± 10.6	8.7 ± 1.5	**0.65 ± 0.16**	71.8 ± 2.2	13.3 ± 3.7
Path based (Chang & Yeung, 2008; Fränti, Pasi et al., 2015)	(300, 2, 3)	0.27 ± 0.10	11.6 ± 1.4	0.48 ± 0.09	122.5 ± 5.3	9.1 ± 1.3	**0.51 ± 0.07**	49.0 ± 2.9	16.4 ± 2.9
Ring (Ilc, 2013)	(800, 2, 2)	0.09 ± 0.01	29.2 ± 2.8	0.20 ± 0.04	501.2 ± 4.7	10.6 ± 1.7	**0.99 ± 0.05**	50.4 ± 1.9	2.5 ± 0.6
Seeds[b](Bache & Lichman, 2013)	(210, 7, 3)	0.34 ± 0.07	5.2 ± 0.9	**0.54 ± 0.13**	128.2 ± 6.6	3.5 ± 1.0	0.45 ± 0.13	15.0 ± 2.8	4.9 ± 1.3
Spiral (Chang & Yeung, 2008; Fränti, Pasi et al., 2015)	(312, 2, 3)	0.09 ± 0.01	56.0 ± 2.4	0.17 ± 0.01	312.0 ± 0.0	29.1 ± 1.9	**0.29 ± 0.08**	64.9 ± 1.9	27.3 ± 4.2
Synthetic Control (Bache & Lichman, 2013)	(600, 60, 6)	0.09 ± 0.03	34.0 ± 3.5	**0.58 ± 0.05**	598.0 ± 0.0	15.0 ± 1.5	0.33 ± 0.08	22.6 ± 5.6	8.4 ± 2.2
Target (Ultsch, 2005)	(770, 2, 6)	0.57 ± 0.04	19.8 ± 1.6	0.65 ± 0.02	204.2 ± 12.4	10.3 ± 0.8	**1.00 ± 0.00**	41.7 ± 2.0	6.4 ± 0.7
Tetra (Ultsch, 2005)	(400, 3, 4)	0.38 ± 0.09	16.0 ± 2.9	**0.69 ± 0.15**	199.6 ± 7.9	5.3 ± 1.3	0.62 ± 0.13	32.4 ± 4.1	7.6 ± 2.3
Wave (Ilc, 2013)	(287, 2, 2)	0.14 ± 0.09	4.3 ± 0.8	0.19 ± 0.04	189.0 ± 1.4	9.6 ± 1.1	**0.44 ± 0.19**	36.6 ± 2.0	7.2 ± 2.1
Wine (Bache & Lichman, 2013)	(178, 13, 3)	0.09 ± 0.03	14.3 ± 1.4	**0.64 ± 0.14**	178.0 ± 0.0	7.6 ± 1.5	0.42 ± 0.14	9.8 ± 2.8	5.1 ± 1.7
			VAT ordering						
Atom (Ultsch, 2005)	(800, 3, 2)	0.87 ± 0.05	2.0 ± 0.0	**1.00 ± 0.00**	148.5 ± 0.5	2.0 ± 0.0	0.87 ± 0.02	29.1 ± 1.0	5.0 ± 0.0
Chainlink (Ultsch, 2005)	(1000, 3, 2)	0.95 ± 0.03	2.0 ± 0.0	**1.00 ± 0.00**	117.2 ± 3.0	2.0 ± 0.0	**1.00 ± 0.00**	8.0 ± 0.0	2.0 ± 0.0
Face (Ilc, 2013)	(320, 2, 4)	0.38 ± 0.27	3.0 ± 0.0	**1.00 ± 0.00**	62.0 ± 0.0	4.0 ± 0.0	**1.00 ± 0.00**	9.0 ± 0.0	4.0 ± 0.0
Iris (Bache & Lichman, 2013)	(150, 4, 3)	0.72 ± 0.17	3.5 ± 0.5	0.72 ± 0.17	6.1 ± 2.0	2.5 ± 0.5	**0.82 ± 0.07**	13.2 ± 1.7	4.5 ± 0.5
Jain (Fränti, Pasi et al., 2015; Jain & Law, 2005)	(373, 2, 2)	0.71 ± 0.17	2.5 ± 0.5	**1.00 ± 0.00**	112.0 ± 0.0	2.0 ± 0.0	0.74 ± 0.08	27.5 ± 0.5	12.5 ± 0.5
Lsun (Ultsch, 2005)	(400, 2, 3)	0.93 ± 0.07	3.0 ± 0.0	**1.00 ± 0.00**	218.6 ± 0.5	3.0 ± 0.0	0.93 ± 0.06	17.4 ± 0.5	5.3 ± 2.0
Moon (Ilc, 2013)	(514, 2, 4)	0.54 ± 0.03	5.0 ± 0.0	**0.97 ± 0.03**	514.0 ± 0.0	4.5 ± 0.5	0.63 ± 0.06	17.5 ± 0.5	11.0 ± 0.0
Path based (Chang & Yeung, 2008; Fränti, Pasi et al., 2015)	(300, 2, 3)	0.49 ± 0.01	9.6 ± 0.5	**0.85 ± 0.12**	78.0 ± 0.0	5.4 ± 0.5	0.60 ± 0.04	24.6 ± 0.5	11.4 ± 0.5
Ring (Ilc, 2013)	(800, 2, 2)	0.23 ± 0.04	7.5 ± 0.5	**1.00 ± 0.00**	20.5 ± 0.5	2.0 ± 0.0	0.87 ± 0.13	16.5 ± 0.5	3.0 ± 1.0
Seeds[b](Bache & Lichman, 2013)	(210, 7, 3)	**0.64 ± 0.02**	6.9 ± 1.0	**0.64 ± 0.02**	6.9 ± 1.0	6.9 ± 1.0	0.60 ± 0.01	32.1 ± 1.0	8.0 ± 0.0
Spiral (Chang & Yeung, 2008; Fränti, Pasi et al., 2015)	(312, 2, 3)	0.24 ± 0.03	9.0 ± 0.0	**1.00 ± 0.00**	42.3 ± 0.5	3.0 ± 0.0	0.23 ± 0.00	15.7 ± 0.5	12.0 ± 1.4
Synthetic Control (Bache & Lichman, 2013)	(600, 60, 6)	0.61 ± 0.03	12.0 ± 0.0	0.61 ± 0.03	12.0 ± 0.0	12.0 ± 0.0	**0.66 ± 0.04**	35.7 ± 1.0	12.4 ± 0.5
Target (Ultsch, 2005)	(770, 2, 6)	0.64 ± 0.00	10.0 ± 0.0	**1.00 ± 0.00**	69.5 ± 2.5	6.0 ± 0.0	0.85 ± 0.15	11.0 ± 0.0	7.0 ± 1.0
Tetra (Ultsch, 2005)	(400, 3, 4)	0.99 ± 0.00	4.0 ± 0.0	0.99 ± 0.00	4.0 ± 0.0	4.0 ± 0.0	**1.00 ± 0.00**	4.0 ± 0.0	4.0 ± 0.0
Wave (Ilc, 2013)	(287, 2, 2)	0.38 ± 0.10	2.0 ± 0.0	**1.00 ± 0.00**	14.5 ± 0.5	2.0 ± 0.0	0.38 ± 0.03	8.5 ± 0.5	5.5 ± 0.5
Wine (Bache & Lichman, 2013)	(178, 13, 3)	0.58 ± 0.07	7.4 ± 0.5	0.58 ± 0.16	78.6 ± 0.5	13.2 ± 2.5	**0.64 ± 0.00**	15.6 ± 0.5	5.0 ± 0.0

Best peak average results are reported in bold.

[a] The (N, d, K) tuple represents the number of samples, dimensions and clusters of each dataset, in this order.

[b] Contributors gratefully acknowledge support of their work by the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

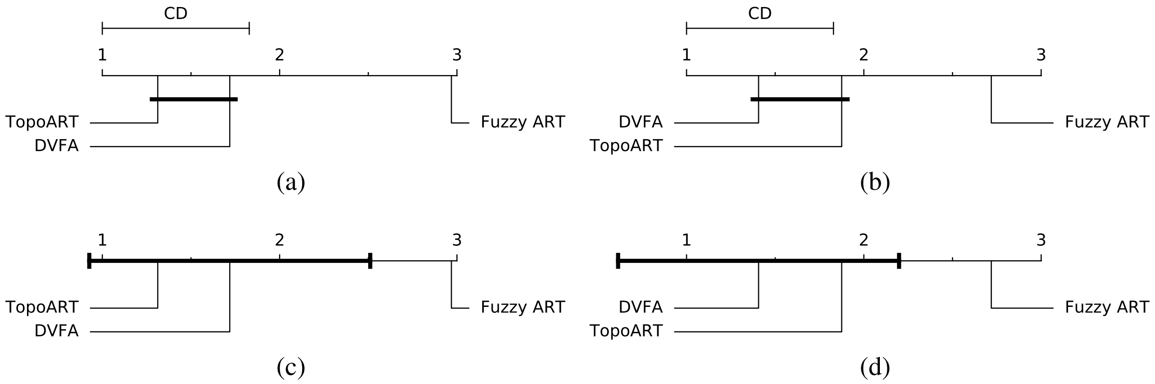


Figure 1. CD diagrams (Demšar, 2006) with respect to random (a, c) and VAT (b, d) input presentations using Nemenyi (a, b) and Bonferroni-Dunn tests (c, d).

Following the statistical algorithm comparison procedure described in (Demšar, 2006), there is sufficient evidence to conclude that these Fuzzy ART-based systems do not perform equally at 0.05 significance level using the Iman statistic (based on the Friedman statistic) (Demšar, 2006). Moreover, Figure 1 depicts the critical difference (CD) diagrams (Demšar, 2006) using the Nemenyi and Bonferroni-Dunn tests at 0.05 significance level. In the CD diagram of the former (Figs. 1a and 1b), DVFA and TopoART are linked, hence not significantly different but nonetheless surpassing Fuzzy ART. The CD diagram of the latter (Figs. 1c and 1d) uses DVFA as the control algorithm and supports the same conclusion. Therefore, TopoART and DVFA yield comparable performances, both significantly better than Fuzzy ART. Specifically, TopoART and DVFA have superior average ranks when data is presented randomly and VAT ordering is used, respectively.

Finally, the compactness of the multi-prototype networks were assessed in a principled manner by employing the Wilcoxon signed-ranks test (Demšar, 2006). As expected, at a 0.05 significance level, TopoART creates more compact networks than DVFA (*p*-values: 0.0015 (random) and 0.0072 (VAT)). Hence, DVFA trades simple design for network compactness.

6. CONCLUSIONS

This work presented the idea of using multiple vigilance levels in one ART node, showcased in Dual Vigilance Fuzzy ART (DVFA), which can capture arbitrarily-shaped datasets by dynamically creating a many-to-one mapping between categories and clusters. This is accomplished by presenting the data samples in a suitable order and making use of two layered vigilance parameters in a single Fuzzy ART unit. This allows for a multi-prototype representation of clusters. On average, when performance was assessed off-line (where it is possible to pre-process the dataset using VAT), DVFA, TopoART and Fuzzy ART ranked first, second and third, respectively. However, in on-line learning with randomly presented samples, TopoART and DVFA swapped the relative positions of their average rankings. Nonetheless, no statistically-significant difference was observed between them. Considering the simplicity of DVFA, these results are encouraging for its use when arbitrarily-shaped clusters are needed.

ACKNOWLEDGEMENTS

Research was sponsored in part by the Missouri University of Science and Technology Mary K. Finley Endowment and Intelligent Systems Center; the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance code BEX 13494/13-9; and the Army Research Laboratory (ARL), and was accomplished under Cooperative Agreement Number W911NF-18-2-0260. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Brito da Silva, L. E., & Wunsch II, D. C. (2015). Multi-prototype local density-based hierarchical clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–9).

Brito da Silva, L. E., & Wunsch II, D. C. (2018). A study on exploiting VAT to mitigate ordering effects in Fuzzy ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2351–2358).

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Chang, H., & Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*, 191–203.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., & Zupan, B. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, *14*, 2349–2353.

Fränti, Pasi et al. (2015). Clustering datasets. URL: `http://cs.uef.fi/sipu/datasets/` accessed on May 4, 2017.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *19*, 90 – 106.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Ilc, N. (2013). Datasets package. URL: `https://www.researchgate.net/publication/239525861_Datasets_package` accessed on Sep 11, 2017.

Isawa, H., Matsushita, H., & Nishio, Y. (2008). Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3595–3600).

Isawa, H., Tomita, M., Matsushita, H., & Nishio, Y. (2007). Fuzzy Adaptive Resonance Theory with Group Learning and its Applications. In *Proc. International Symposium on Nonlinear Theory and its Applications (NOLTA)* (pp. 292–295).

Jain, A. K., & Law, M. H. C. (2005). Data Clustering: A User's Dilemma. In S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds.), *Pattern Recognition and Machine Intelligence* (pp. 1–10). Berlin, Heidelberg: Springer Berlin Heidelberg volume 3776 of *Lecture Notes in Computer Science*.

Seiffertt, J., & Wunsch II, D. C. (2010). *Unified Computational Intelligence for Complex Systems* volume 6 of *Evolutionary Learning and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Su, M.-C., & Liu, T.-K. (2001). Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, *37*, 165 – 175.

Su, M.-C., & Liu, Y.-C. (2005). A new approach to clustering data with arbitrary shapes. *Pattern Recognition*, *38*, 1887 – 1901.

Tscherepanow, M. (2010). TopoART: A Topology Learning Hierarchical ART Network. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 157–167). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M., Kortkamp, M., & Kammer, M. (2011). A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*, *24*, 906 – 916.

Tscherepanow, M., Kühnel, S., & Riechers, S. (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 24–29).

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

V. DISTRIBUTED DUAL VIGILANCE FUZZY ADAPTIVE RESONANCE THEORY LEARNS ONLINE, RETRIEVES ARBITRARILY-SHAPED CLUSTERS, AND MITIGATES ORDER DEPENDENCE

Leonardo Enzo Brito da Silva[1,3], Islam Elnabarawy[2] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering,
Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]Applied Computational Intelligence Lab., Dept. of Computer Science,
Missouri University of Science and Technology, Rolla, MO 65409 USA.

[3]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

This paper presents a novel adaptive resonance theory (ART)-based modular architecture for unsupervised learning, namely the distributed dual vigilance fuzzy ART (DDVFA). DDVFA consists of a global ART system whose nodes are local fuzzy ART modules. It is equipped with distributed higher-order activation and match functions and a dual vigilance mechanism. Together, these allow DDVFA to perform unsupervised modularization, create multi-prototype cluster representations, retrieve arbitrarily-shaped clusters, and reduce category proliferation. Another important contribution is the reduction of order-dependence, an issue that affects any agglomerative clustering method. This paper demonstrates two approaches for mitigating order-dependence: pre-processing using visual assessment of cluster tendency (VAT) or post-processing using a novel Merge ART module. The former is suitable for batch processing, whereas the latter also works for online learning. Experimental results in online mode carried out on 30 benchmark data sets show that DDVFA cascaded with Merge ART statistically outperformed the best other ART-based systems when samples were randomly presented. Conversely, they were found to be statistically equivalent in offline mode when samples were pre-processed using VAT. Remarkably, performance comparisons to non-ART-based clustering algorithms show that DDVFA (which learns incrementally) was also statistically equivalent to the non-incremental (offline) methods of density-based spatial clustering of applications with noise (DBSCAN), single linkage hierarchical agglomerative clustering (SL-HAC), and k-means, while retaining the appealing properties of ART. Links to the source code and data are provided. Considering the algorithm's simplicity, online learning capability, and performance, it is an ideal choice for many agglomerative clustering applications.

Keywords: Fuzzy, Adaptive Resonance Theory, Clustering, Distributed Representation, Topology, Visual Assessment of Cluster Tendency.

1. INTRODUCTION

There is a rich literature of clustering methods (Xu & Wunsch II, 2005; Xu & Wunsch II, 2009; Xu & Wunsch II, 2010), and among the neural network-based ones, adaptive resonance theory (ART) (Carpenter & Grossberg, 1987) is of great interest due to its many useful properties (Brito da Silva et al., 2019b; Wunsch II, 2009), particularly the fact that it addresses the *stability-plasticity dilemma*. ART networks learn, after sufficient exposure to the environment, prototypical representations or archetypes that reflect groups of samples (Bartfai, 1994, 1996); i.e., a succinct or compressed representation of the data.

The distributed dual vigilance fuzzy ART (DDVFA) introduced here extends the dual vigilance mechanism of dual vigilance fuzzy ART (DVFA) (Brito da Silva et al., 2019a) to perform several ART-based online hierarchical agglomerative clustering (HAC) methods, reduce category proliferation, and alleviate input ordering effects in online learning mode (when used in a framework that includes another ART-based module also introduced in this work). DDVFA belongs to the class of modular neural networks (Auda & Kamel, 1998, 1999; Auda et al., 1996). Specifically, it is designed for the unsupervised learning task of clustering. This class of network architectures employs a divide-and-conquer approach and shares the following main features (Auda & Kamel, 1998, 1999; Auda et al., 1996): task decomposition (breaking down a complex problem) and multi-module decision making (combining local decisions in a single global consensus). Commonly, unsupervised learning methods are used as a pre-processing stage to partition the data to be handled by supervised modules. ART-based systems have been used for such purposes in supervised modular networks (Auda & Kamel, 1998, 1999; Auda et al., 1996).

A current challenge for incremental learners, such as ART-based systems, is the order of sample presentation. Thus, suitable pre-processing, post-processing, and/or during learning strategies are usually employed when applicable (see references in (Brito da Silva & Wunsch II, 2018a)). For instance, merging strategies are commonly used in conjunction with incremental learners (e.g., (Benites & Sapozhnikova, 2017; Isawa et al., 2008a,b, 2009; Lughofer, 2008; Swope, 2012; Zhang et al., 2006)); here, a novel ART-based network provides such functionality. Additionally, visualization and assessment are valuable assets when performing cluster analysis (Bezdek, 2017; Brito da Silva & Wunsch II, 2018b; Xu & Wunsch II, 2009); here, the visual assessment of cluster tendency (VAT) technique (Bezdek, 2017; Bezdek & Hathaway, 2002) is used for its sample ordering properties to emulate scenarios in which such data pre-processing is practical, as per (Brito da Silva & Wunsch II, 2018a).

This paper presents the following main contributions:

1. A novel modular fuzzy ART-based architecture (DDVFA). Unsupervised dynamic modularization (creation of new local modules as needed) and multi-prototype representation are accomplished by employing dual vigilance parameters associated with global and local fuzzy ART modules.

2. Novel higher order distributed activation and normalized match functions based on HAC methods embedded in the incremental learning process. Suitably setting the HAC-based activation/match functions allows DDVFA to retrieve arbitrarily shaped

clusters, and higher order match functions have the potential to generate more compact DDVFA networks (as per (Carpenter, 1997; Carpenter et al., 1998)) and extend the regions of successful dual vigilance parameter combinations.

3. A novel Merge ART module compatible with DDVFA for post-processing purposes in online learning applications. This procedure compensates for the errors caused by the random order of input presentations thus enabling improved performance.

4. An analysis of the behavior of the DDVFA with and without pre-processing (VAT) and post-processing (Merge ART) strategies, as well as a discussion on its hyper-parameters and computational complexity.

The results show that together, these features enable DDVFA to yield an improved performance compared to other current state-of-the-art fuzzy ART-based technologies.

The remainder of this paper is divided as follows: Section 2 provides a brief overview of related works, focusing on ART, fuzzy ART, fuzzy topoART, and dual vigilance fuzzy ART; Section 3 introduces the distributed dual vigilance fuzzy ART; Section 4 describes the experimental set-up; Section 5 reports and discusses the results; and Section 7 is the conclusion.

2. BACKGROUND AND RELATED WORK

Adaptive resonance theory (ART) (Grossberg, 1976) is the theory that learning is often mediated by resonant feedback in neural circuits. It inspired the development of many neural network architectures, each with its own internal categorical representation, while sharing the same design principles. The ART *matching rule* (Carpenter & Grossberg, 1987) is a key property of these ART systems (Amis & Carpenter, 2007; Carpenter, 2003); it regulates the interaction between top-down expectations (represented by the internal categories or templates) and the bottom-up inputs. This process is guided by an orienting subsystem, which performs a hypothesis test, called the *vigilance check*, that either shuts down or enables an ART category to learn. A set of ART categories discretizes the data, thus summarizing it via clusters. The vigilance parameter (see Eq. (4)) controls category size and thus the granularity of this discretization. ART templates have specific properties and governing equations based on their internal representation, e.g., hyperboxes (Carpenter et al., 1991b); Gaussians (Vigdor & Lerner, 2007; Williamson, 1996); hyperspheres (Anagnostopoulos & Georgiopoulos, 2000); hyperellipses (Anagnostopoulos & Georgiopoulos, 2001); and others.

Numerous ART-based architectures have been conceived, such as predictive ART (ARTMAP) for supervised mappings (Carpenter et al., 1992, 1991a); fusion ART (Tan et al., 2007), whose variants have been effectively used for semi-supervised (Meng et al., 2014), supervised (Tan, 1995), and reinforcement learning applications (Tan, 2004, 2006; Tan et al., 2008); Biclustering ARTMAP (BARTMAP) (Xu & Wunsch II, 2011) for biclustering applications, such as gene expression analysis (Xu & Wunsch II, 2011) and collaborative filtering (Elnabarawy et al., 2016); and ART networks endowed with multiple vigilance tests (Brito da Silva & Wunsch II, 2017; Gomez-Sanchez et al., 2001; Huang et al., 2014; Seiffertt & Wunsch II, 2010). A brief review of ART networks related to the contributions of this work is provided next, where emphasis is given to the architectures used for comparison purposes, thereby making this paper self-contained. For a detailed discussion of ART models developed over the past three decades the reader is referred to (Brito da Silva et al., 2019b).

2.1. ABBREVIATED REVIEW

ART has been used as the basis for several hierarchical clustering methods, which can be classified into bottom-up (agglomerative or merging methods) and top-down (divisive or splitting methods) (Xu & Wunsch II, 2009). Hierarchical ART architectures generally follow two main designs (Massey, 2009): (a) a series/cascade of ART modules where the output of one ART (i.e., a prototype) is the input of the next (Bartfai, 1996; Bartfai &

White, 1997a,b; Benites & Sapozhnikova, 2017; Carpenter & Grossberg, 1990; Chen & Lin, 2001; Chen et al., 1999; Hung et al., 1996a,b; Ishihara et al., 1995; Yavaş & Alpaslan, 2012) or (b) parallel ART modules sharing the same inputs and using different vigilance values (Bartfai, 1994; Kim & Wunsch II, 2011; Tscherepanow, 2010, 2012; Tscherepanow et al., 2011, 2012; Švaco et al., 2014; Wunsch II, 1991; Wunsch II et al., 1993). Generally, the hierarchical relationships between ART modules are defined implicitly by the input signal flow, explicitly by enforcing constraints or connections, and/or by the setting of multiple vigilance parameters to define hierarchies. Alternately, hierarchies within the same ART can be created by designing custom ART activation functions (Lavoie et al., 1997, 1999) or by analyzing its distributed activation patterns (Davenport & Titus, 2004). ART-based hierarchical approaches have been successfully applied, for instance, in the text mining (Bouchachia & Mittermeir, 2003; Massey, 2009) and robotics (Švaco et al., 2014; Yavaş & Alpaslan, 2012) domains.

Another branch of clustering includes multi-prototype-based methods. These allow multiple prototypes to represent a single cluster and more accurately capture the data topology, thereby typically handling clusters with arbitrary shapes. Multi-prototype representations have been successfully used for clustering (Araújo et al., 2013a,b; Guha et al., 1998; Taşdemir & Merényi, 2009; Tyree & Long, 1999), multivariate data visualization (Brito da Silva & Wunsch II, 2018b; Taşdemir & Merényi, 2009; Ultsch & Siemon, 1990), and cluster validation purposes (Halkidi & Vazirgiannis, 2008; Taşdemir & Merényi, 2011). In the context of ART, examples include the combination of an ART-like system using quadratic neurons (Su & Liu, 2001) and hierarchical clustering (Su & Liu, 2002, 2005) and the related approach (Brito da Silva & Wunsch II, 2015) using fuzzy ART (Carpenter et al., 1991b). Other methods have augmented ART-based systems by employing dual vigilance parameters (Brito da Silva et al., 2019a), connecting the first and second resonating categories (Isawa et al., 2008a,b, 2009, 2007; Tscherepanow, 2010, 2012; Tscherepanow et al., 2011, 2012), or replacing fuzzy ART's nodes with growing cell structures (Fritzke, 1994) in a hybrid architecture (Kim et al., 2011).

Although based on multi-prototype representation, many of the previously mentioned approaches do not adopt distributed activation, match or learning, which improves a network's noise robustness and compactness (Carpenter, 1997; Carpenter et al., 1998). The distributed ART model (Carpenter, 1997) is endowed with all of these distributed features, however it does not possess a mechanism to build, in an unsupervised manner, a permanent and binary many-to-one mapping of categories to clusters (i.e., a multi-prototype cluster representation). Thus, it is still limited by its nested hyperbox cluster abstractions. Distributed learning is also featured in the ART variants introduced in (Kondadadi & Kozma, 2002; Yousuf & Murphey, 2010). In the ART literature, the power of distributed representation has been harnessed to perform, for instance, (a) unsupervised feature extraction (Lam et al., 2015); (b) hierarchical clustering (Carpenter & Grossberg, 1990; Chen & Lin, 2001) – although featuring distributed representation, the latter approaches are cascade architectures not designed to model arbitrarily-shaped clusters; and (c) supervised learning systems such as the distributed ARTMAP (Carpenter et al., 1998) (which is a generalization of a variety of other ART models (Carpenter, 2003) such as (Amis & Carpenter, 2007; Carpenter, 2003; Carpenter et al., 1992, 1991b; Carpenter & Markuzon, 1998)), some topoART variants (Tscherepanow, 2011; Tscherepanow & Riechers, 2012), default ARTMAPs (Amis & Carpenter, 2007; Carpenter, 2003), and the adaptive resonance associative map (Tan, 1995) variants introduced in (Benites & Sapozhnikova, 2017; Sapozhnikova, 2009).

2.2. FUZZY ART

Fuzzy ART (Carpenter et al., 1991b) is an ART architecture designed to work with real-valued data and follows the ART design depicted in Figure 1. Concisely, when a sample $\boldsymbol{x} \in \mathbb{R}^d$ is presented at the feature representation field F_1, it activates the category j at the

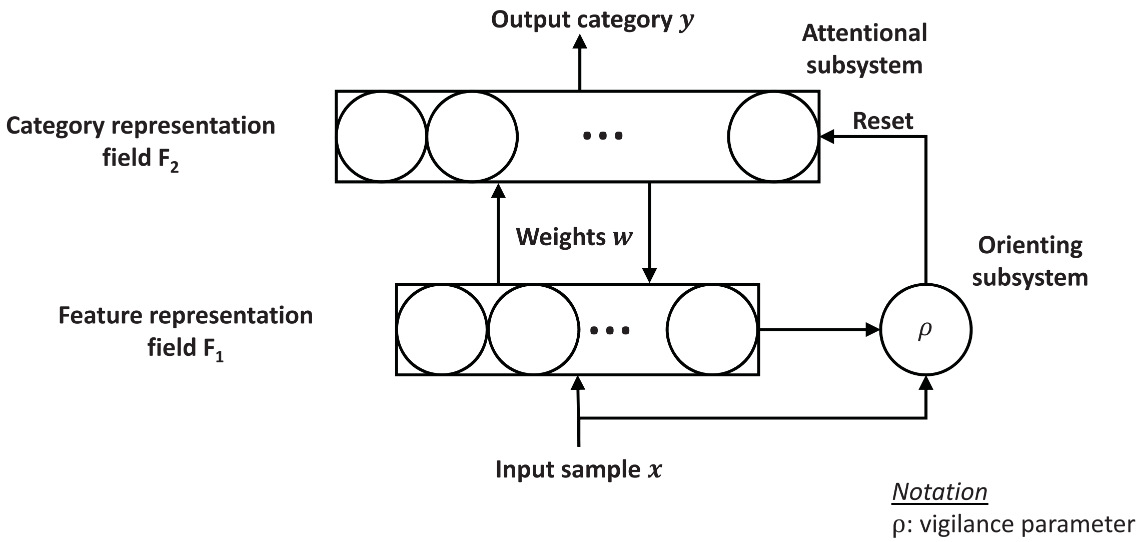


Figure 1. Elementary ART architecture, underlying various ART designs.

category representation field F_2 whose weight vector $\boldsymbol{w}_j$ maximizes the following *activation function*:

$$T_j = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|}, \tag{1}$$

where $\wedge$ represents the fuzzy intersection, defined as the element-wise minimum operation

$$(\boldsymbol{x} \wedge \boldsymbol{w}_j)_i \equiv \min(x_i, w_{j,i}), \tag{2}$$

$|\cdot|$ is the L_1 norm, and $\alpha > 0$ is the choice parameter, which is usually set to a small value. A comprehensive study on its behavior can be found in (Georgiopoulos et al., 1996).

Next, a *match function* evaluates the best matching category as:

$$M_j = \frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j|}{|\boldsymbol{x}|}, \tag{3}$$

and a *vigilance check* v is performed using the computed match value:

$$v : M_j \geq \rho, \tag{4}$$

where $0 \leq \rho \leq 1$ is the vigilance parameter. If ν is satisfied, then the winning category's weight vector is updated as:

$$\boldsymbol{w}_j^{new} = (1 - \beta)\boldsymbol{w}_j^{old} + \beta(\boldsymbol{x} \wedge \boldsymbol{w}_j^{old}), \tag{5}$$

where $0 < \beta \leq 1$ is the learning rate parameter. Otherwise, this category is deactivated, and the search continues by activating the next highest ranked category. If none of them satisfies this constraint, then a new category is created to encode sample $\boldsymbol{x}$. Thus, the problem of selecting the number of clusters is traded for the one of selecting the vigilance value ρ.

Fuzzy ART features many appealing properties such as scalability, speed, stability, plasticity, online and offline incremental learning modes, as well as simple implementation, transparency, and novelty detection (Amis & Carpenter, 2007; Carpenter, 2003; Mulder & Wunsch II, 2003; Wunsch II, 2009; Xu & Wunsch II, 2009).

2.3. FUZZY TOPOART

Fuzzy topoART (Tscherepanow, 2010) incorporates topology-based learning (Furao & Hasegawa, 2006) into ART. Briefly, it consists of multiple independent fuzzy ART modules where the preceding modules filter the shared inputs to subsequent ones. Standard topoART consists of two identical modules: A and B. During training, which is processed in parallel for all modules, an "instance counting" feature accounts for the number of samples N learned by a given category. Every τ learning cycles/iterations (number of sample presentations), a noise thresholding procedure is performed to remove categories with less than ϕ samples. Once the threshold is surpassed, "candidate" categories become "permanent" categories. A sample is propagated to module B if it has resonated with a "permanent" category of module A.

The granularity of the solutions is defined by the modules' different vigilance parameter values. Module B's vigilance parameter is (Tscherepanow, 2010; Tscherepanow et al., 2011, 2012):

$$\rho_b = \frac{1}{2}\left(\rho_a + 1\right), \tag{6}$$

where ρ_a is module A's vigilance parameter. Since $\rho_b \geq \rho_a$, modules A and B yield increasingly finer partitions of a given data set. Categories are laterally connected by edges between the first and second resonating categories (i.e., the two highest ranked categories that simultaneously satisfy the vigilance test (Eq. (4))) to mirror the input distribution. This multi-prototype method enables topoART modules to learn topologies and capture clusters with arbitrary geometries. Besides competitive learning, it also uses cooperative learning by allowing the second winner (sbm) to learn with a smaller learning rate than the first (bm): $\beta_{sbm} < \beta_{bm} = 1$. Finally, to compensate for fuzzy ART's bias toward small categories, topoART uses a particular activation function for prediction, which is independent of category size (Tscherepanow, 2010; Tscherepanow et al., 2011, 2012):

$$T_j = 1 - \frac{\left|\left(\boldsymbol{x} \wedge \boldsymbol{w}_j\right) - \boldsymbol{w}_j\right|}{|\boldsymbol{x}|}. \tag{7}$$

TopoART has spawned several variants for unsupervised (Tscherepanow, 2012; Tscherepanow et al., 2011, 2012), supervised (Tscherepanow, 2011; Tscherepanow & Riechers, 2012), and semi-supervised (Nooralishahi et al., 2018) learning paradigms.

2.4. DUAL VIGILANCE FUZZY ART

Dual vigilance fuzzy ART (DVFA) (Brito da Silva et al., 2019a) consists of a single ART module equipped with two layered vigilance parameters. The larger vigilance value is referred to as the "upper bound" (ρ_{UB}) and is responsible for the data compression/quantization, whereas the lower vigilance value is referred to as the "lower bound" (ρ_{LB}) and is responsible for the cluster similarity. Briefly, when a category J is activated after a winner-take-all competition, then a vigilance check with a large value is performed (using ρ_{UB} in Eq. (4)); if it is satisfied, then it behaves identically to fuzzy ART. However, if this test fails, then a second test is performed with a slightly smaller vigilance value (using ρ_{LB} in Eq. (4)). If category J satisfies this looser constraint, then a new category I is created and permanently assigned to the same cluster C as the tested category via a binary mapping matrix $\boldsymbol{W}^{map} = \left[w_{r,c}^{map}\right]_{P\times K}$, which expands as

$$w_{r,c}^{map} = \begin{cases} 1, & r = I, c = C \\ 0, & r = I, c \neq C \\ w_{r,c}^{map}, & r \neq I, \forall c \end{cases}, \tag{8}$$

where P and K are the current number of categories and clusters, respectively. On the other hand, if both vigilance tests fail for all categories, then a new one (I) is created and $\boldsymbol{W}^{map}$ expands as

$$w_{r,c}^{map} = \begin{cases} 1, & r = I, c = K+1 \\ 0, & r = I, c \neq K+1 \\ 0, & r \neq I, c = K+1 \\ w_{r,c}^{map}, & r \neq I, c \neq K+1 \end{cases}, \tag{9}$$

note that in all scenarios, the index of the new category is $I = P + 1$.

Therefore, each output cluster maintains a list of categories it is represented by, which is a similar principle employed by fuzzy ARTMAP's map field matrix (Carpenter et al., 1992) to perform supervised mappings of categories to classes. However, DVFA's matrix $\boldsymbol{W}^{map}$ performs an unsupervised many-to-one (surjective) mapping of categories to clusters (this is a multi-prototype approach). In this manner, the data distribution can be more faithfully mirrored, and clusters of arbitrary geometries may be retrieved.

3. DISTRIBUTED DUAL VIGILANCE FUZZY ART

The distributed dual vigilance fuzzy ART (DDVFA) neural network architecture described in Section 3.1 can be viewed as an "*ART of ARTs*", in which each node in the category representation field F_2 of a global ART is itself a local ART, where the latter represents a given data cluster. Equivalently, it can be seen as an unsupervised modular neural network consisting of local ARTs whose multi-module decision making system is a global ART. Since ART-based systems are sensitive to the order of input presentation, Section 3.2 presents an approach to compensate for this dependency: the output of a DDVFA module (layer 1) is cascaded into a compatible Merge ART module (layer 2).

3.1. DDVFA ARCHITECTURE

Table 1 lists the notation used in this section, and Figure 2 depicts a generic DDVFA. It is a modular structure in which a global ART controls local parallel ARTs via a vigilance feedback between these modules – cf. ART tree (Wunsch II, 1991; Wunsch II et al., 1993), in which F_2 nodes are also ART modules, but these are not controlled by a global ART module. The global ART acts as a mapping mechanism analogous to the inter-ART module in fuzzy ARTMAP architectures (Asfour et al., 1993; Carpenter et al., 1992), thus maintaining hierarchical consistency. This relates to self-consistent modular ART (Bartfai, 1994); however, DDVFA uses a bottom-up agglomerative approach, whereas the former uses a top-down divisive approach limited to hyperrectangular cluster representations. Concretely, DDVFA is a multi-prototype hierarchical agglomerative clustering (HAC) method that builds a self-consistent two-level hierarchy of categories.

Similar to DVFA, the vigilance parameters of the global and local ARTs are denoted as ρ_{LB} and ρ_{UB}, respectively, where the constraint $\rho_{LB} \leq \rho_{UB}$ is enforced. Setting $\rho_{UB} = \rho_{LB}$ reduces the DDVFA to a generic fuzzy ART framework, which ensures that each global ART's F_2 node (i.e., each local ART) encodes one category. Alternately, setting

Table 1. Notation for DDVFA.

Notation	Description
$\boldsymbol{X}$	a data set $\boldsymbol{X} = \{\boldsymbol{x}_l\}_{l=1}^{N} \in \mathbb{R}^d$.
$ART_j^{(i)}$	global ART's F_2 node j (layer i).
$T^{ART_j^{(i)}}$, $M^{ART_j^{(i)}}$	activation and match functions of local $ART_j^{(i)}$, respectively.
$\boldsymbol{w}_k^{ART_j^{(i)}}$	k^{th} category weight vector of local $ART_j^{(i)}$.
$T_k^{ART_j^{(i)}}$, $M_k^{ART_j^{(i)}}$	activation and match functions of $\boldsymbol{w}_k^{ART_j^{(i)}}$, respectively.
$\gamma \geq 1$	kernel width.
$0 \leq \gamma^* \leq \gamma$	reference kernel width.
$\rho_{UB}^{(i)} \geq \rho_{LB}^{(i)}$	lower and upper bound vigilance parameters (layer i).
$\boldsymbol{T}_{p,q}$, $\boldsymbol{M}_{p,q}$	activation and match matrices between local $ART_p^{(1)}$ and local $ART_q^{(2)}$.
$n_k^{ART_j^{(i)}}$	number of samples encoded by category k of local $ART_j^{(i)}$ (instance counting).
$n^{ART_j^{(i)}}$	total number of samples encoded by local $ART_j^{(i)}$ (instance counting).

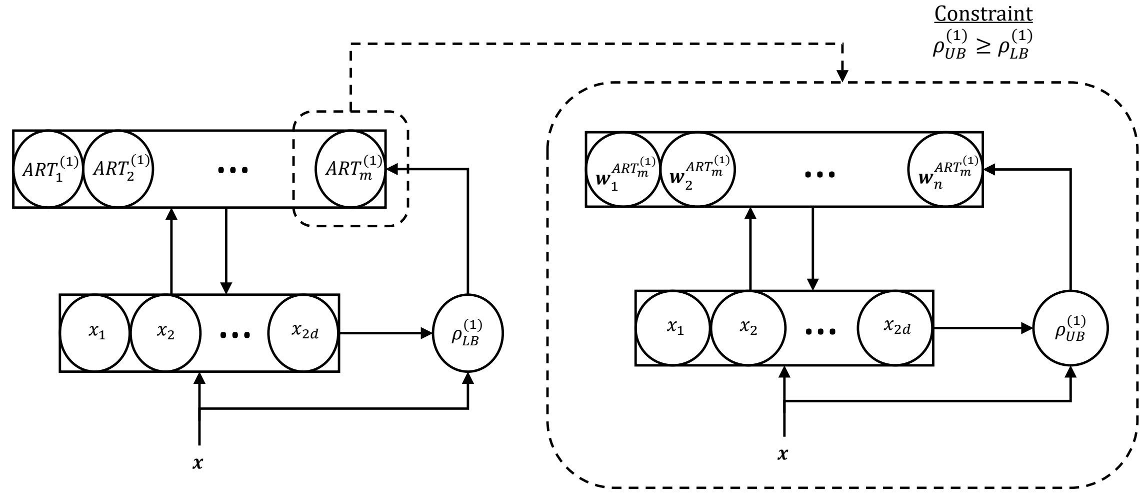


Figure 2. DDVFA architecture. Each global ART's F_2 node is a local fuzzy ART (as portrayed in Figure 1) with shared complement-coded input $\boldsymbol{x} \in \mathbb{R}^{2d}$ and vigilance $\rho = \rho_{UB}^{(1)} \geq \rho_{LB}^{(1)}$.

ρ_{UB} strictly greater than ρ_{LB} builds a multiple category representation for each cluster, thus enabling an approximation of that cluster's geometry over the data space according to the underlying assumption of the activation and match functions, which are to be set a priori. The vigilance parameters ρ_{LB} and ρ_{UB} reflect the minimum similarity of a cluster and the granularity level of the data quantization (i.e., the categories' sizes), respectively. In other words, the rationale is to restrict the maximum internal category size of each local ART while maintaining a smaller similarity constraint for the cluster represented by each global ART F_2 node. Thus, local ART modules (or clusters) can be added as needed.

The inner workings of DDVFA are the same as a generic ART architecture, as reviewed in Section 2. However, the activation $T^{ART_i}(\cdot)$ and match $M^{ART_i}(\cdot)$ functions of the global ART's F_2 node i are a distributed version of the local ART_i categories' activation $T_j^{ART_i}$ and match $M_j^{ART_i}$ functions based on HAC, where $j = \{1, ..., k\}$ represents the categories. Specifically, the activation and match functions of global ART's F_2 node i in layer (1) are given by a function of local $ART_i^{(1)}$'s k nodes:

$$T^{ART_i^{(1)}} = f\left(T_1^{ART_i^{(1)}}, T_2^{ART_i^{(1)}}, \dots, T_k^{ART_i^{(1)}}\right), \tag{10}$$

where

$$T_j^{ART_i^{(1)}} = \left(\frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j^{ART_i^{(1)}}|}{\alpha + |\boldsymbol{w}_j^{ART_i^{(1)}}|}\right)^{\gamma}, \; j \in \{1, \dots, k\}, \tag{11}$$

and

$$M^{ART_i^{(1)}} = g\left(M_1^{ART_i^{(1)}}, M_2^{ART_i^{(1)}}, \dots, M_k^{ART_i^{(1)}}\right), \tag{12}$$

where

$$M_j^{ART_i^{(1)}} = \left(\frac{|\boldsymbol{x} \wedge \boldsymbol{w}_j^{ART_i^{(1)}}|}{|\boldsymbol{x}|}\right)^{\gamma}, \; j \in \{1, \dots, k\}. \tag{13}$$

In this study, for simplicity, $f(\cdot) = g(\cdot)$ in (10) and (12), i.e., the same functional relationship is used for the activation and match functions. These are listed in Table 2 and are based on HAC methods (Xu & Wunsch II, 2009).

A power parameter $\gamma \geq 1$ is employed here in both the activation and match functions. Like the power parameter used in (Carpenter, 1997; Carpenter et al., 1998), γ assumes the role of a kernel width, facilitates the dual vigilance parameters selection, and reduces category proliferation (Section 5.6). Setting $\gamma = 1$ corresponds to a standard fuzzy ART module, in which a moderately far sample would still have a reasonably large value for the match function. This extension of successful dual vigilance parameters occurs because the match and activation functions (when $\gamma = 1$) decay linearly and slowly for samples outside a category's hyperrectangular boundaries and thus, by increasing γ, steeper decays are created (Figure 3). FasArt (Izquierdo et al., 2001) and the fuzzy min-max neural network (Simpson, 1992, 1993) variant (Gabrys & Bargiela, 2000) devise custom fuzzy membership functions, using an analogous parameter to control the membership values.

The property exploited here is the fact that the activation and match functions become more "selective" (as expected from a power rule as a contrast-enhancement procedure (Carpenter, 1997; Carpenter et al., 1998)); e.g., in Figure 3 their trapezoidal form approaches a rectangular membership function. Therefore, regarding the match function, increasing γ makes far samples less similar and a category's vigilance region (Meng et al., 2016) smaller (Figure 3). Naturally, when applying a power rule to a scalar in the range $[0, 1]$, such as the case of the match and activation functions, its value decreases with γ. Therefore, to account for the scaling effect, instead of using (13), the match function is normalized as:

$$M_j^{ART_i^{(1)}} = \left(\frac{|\boldsymbol{w}_j^{ART_i^{(1)}}|}{|\boldsymbol{x}|} \right)^{\gamma^*} T_j^{ART_i^{(1)}}, j \in \{1, \ldots, k\}, \tag{14}$$

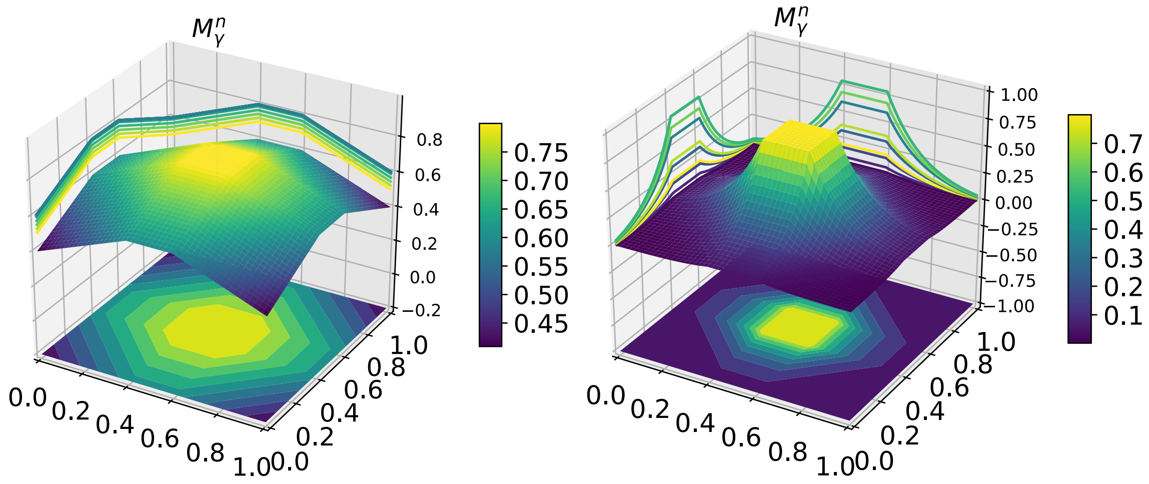
Table 2. DDVFA's activation and match functions.

Method	$T^{ART_i^{(1)}} = f(\cdot)$	$M^{ART_i^{(1)}} = g(\cdot)$
single	$\max\limits_j \left(T_j^{ART_i^{(1)}}\right)$	$\max\limits_j \left(M_j^{ART_i^{(1)}}\right)$
complete	$\min\limits_j \left(T_j^{ART_i^{(1)}}\right)$	$\min\limits_j \left(M_j^{ART_i^{(1)}}\right)$
median	$\operatorname*{median}\limits_j \left(T_j^{ART_i^{(1)}}\right)$	$\operatorname*{median}\limits_j \left(M_j^{ART_i^{(1)}}\right)$
average[a]	$\frac{1}{k_i} \sum\limits_{j=1}^{k_i} T_j^{ART_i^{(1)}}$	$\frac{1}{k_i} \sum\limits_{j=1}^{k_i} M_j^{ART_i^{(1)}}$
weighted[b]	$\sum\limits_{j=1}^{k_i} p_j T_j^{ART_i^{(1)}}$	$\sum\limits_{j=1}^{k_i} p_j M_j^{ART_i^{(1)}}$
centroid[c]	$\left(\frac{\lvert \boldsymbol{x} \wedge \boldsymbol{w}_c \rvert}{\alpha + \lvert \boldsymbol{w}_c \rvert}\right)^{\gamma}$	$\left(\frac{\lvert \boldsymbol{x} \wedge \boldsymbol{w}_c \rvert}{\lvert \boldsymbol{x} \rvert}\right)^{\gamma}$

[a,b] k_i represents the number of categories in $ART_i^{(1)}$.

[b] $p_j = \frac{n_j^{ART_i^{(1)}}}{n^{ART_i^{(1)}}}$ and $n^{ART_i^{(1)}} = \sum\limits_j n_j^{ART_i^{(1)}}$. This represents an a priori probability of $ART_i^{(1)}$'s category j analogous to (Vigdor & Lerner, 2007; Williamson, 1996).

[c] $\boldsymbol{w}_c$ is the centroid representing all categories of $ART_i^{(1)}$, where its l component is given by $w_{c,l} = \min\limits_j \left(w_{j,l}\right)$ for $l = \{1, \ldots, 2d\}$.



(a) $\gamma = 1$

(b) $\gamma = 10$

Figure 3. 3D surfaces, contours, and cross-section cuts representing the normalized match functions (M_γ^n) using $\gamma^* = 1$ and different values of γ.

where $0 \leq \gamma^* \leq \gamma$ is the reference kernel width with respect to which the match function is normalized (see Appendix A). In this paper's experiments, such normalization was performed with respect to the match function values of a standard fuzzy ART (i.e., $\gamma^* = 1$). Note that the higher order HAC-based activation functions in Eq. (11) do not change the search order for global ART when varying γ for single, complete, and centroid methods; but it may for weighted and average. Additionally, it also does not affect the search order within the local fuzzy ART module using the higher order activation and match functions.

Remark 1. A power law was introduced in distributed ART/ARTMAP (Carpenter, 1997) for the increased gradient content-addressable memory rule as a contrast enhancement procedure, and it has been used in other ART variants such as distributed ARTMAP (Carpenter et al., 1998) and default ARTMAPs (Amis & Carpenter, 2007; Carpenter, 2003). As opposed to the latter ART systems, where the activation functions are normalized to 1 with respect to a subset of highly active nodes, DDVFA's activation functions are not normalized, but rather its match functions. Specifically, the latter are normalized using a reference parameter γ^* and with respect to an individual category; additionally, DDVFA's match-reset-search mechanism itself is distinct and uses winner-takes-all learning, as opposed to distributed ART's distributed learning.

Remark 2. There are subtle, yet fundamental, differences between DVFA and DDVFA besides the architecture itself and the distributed HAC-based higher order nature of the activation and normalized match functions. The first one relates to the search mechanism. In DVFA, it is possible for categories mapped to the same cluster to be brought up during the search process. Conversely, in DDVFA, if a global ART node does not satisfy the vigilance test, then its local ART and the cluster it represents (which includes all its categories) is shut down and will not appear again during global ART's search. Another difference is that, according to Eq. (12) and Table 2, the match functions are distributed, and, in the case of single and complete variants, the category selected by winner-takes-all competition and the category subjected to the vigilance test are not required to be the same.

Naturally, DDVFA integrates a winner-take-all mechanism to select among global ART's F_2 nodes (i.e., local Fuzzy ARTs) with a variety of distributed HAC-based activation/match functions, which are computed using local fuzzy ART's weight vectors. According to their definitions (Table 2), they range from winner-take-all (single) and loser-take-all (complete) to completely distributed (average, centroid, and weighted). DDVFA can be viewed as an ART-based online incremental approximate (prototype-based) HAC method. If $\rho_{UB}^{(1)} = 1$, then the approach reduces to an ART-based HAC, since each local fuzzy ART's category encodes a single sample, and the dendrogram cut-level is defined by the global ART module's vigilance parameter $\rho_{LB}^{(1)}$. Algorithm 4 summarizes the DDVFA's pseudocode.

3.2. MERGE ART MODULE

The order of input presentation is a challenge for incremental learners as it plays a significant role in such systems' performance (see references in (Brito da Silva & Wunsch II, 2018a)). For this reason, a Merge ART module (Figure 4) is introduced here to be placed at layer 2, i.e., on top of the DDVFA in a cascade design. It acts as another ART module with dual vigilance parameters in which the inputs are ART nodes from DDVFA. It has its own set of parameters that are independent of DDVFA. However, for simplicity, DDVFA's activation and match functions functional forms were kept to maintain the same underlying cluster assumptions, and $(\rho_{LB}^{(2)}, \rho_{UB}^{(2)})$ were set to $(\rho_{LB}^{(1)}, \rho_{UB}^{(1)})$.

The merging process consists of unions or concatenation of local fuzzy ARTs (merging step) followed by compressions within each set of local fuzzy ARTs (compression step). Let $\boldsymbol{T}_{k,l} = [t_{ij}]_{R\times C}$ and $\boldsymbol{M}_{k,l} = [m_{ij}]_{R\times C}$ be the activation and match matrices of Merge ART's F_2 node $ART_k^{(2)}$ when the input $ART_l^{(1)}$ (from DDVFA) is presented, where R and C are the number of categories of Merge ART's $ART_k^{(2)}$ and DDVFA's $ART_l^{(1)}$, respectively.

Algorithm 4: DDVFA

```
Input  : $\boldsymbol{x}$, $\alpha$, $\beta$, $\rho_{UB}^{(1)}$, $\rho_{LB}^{(1)}$, $\gamma$, $\gamma^*$, method.
Output : DDVFA clusters.

1  Present input sample $\boldsymbol{x} \in \boldsymbol{X}$.
2  Compute $T_j^{ART_i^{(1)}}$, $\forall\, i, j$ (Eq. (11)).
3  Compute $T^{ART_i^{(1)}}$, $\forall\, i$ (Eq. (10), Table 2's method).
4  Find the winning node $I \leftarrow \arg\max_i \{T^{ART_i^{(1)}}\}$.
5  Compute $M_j^{ART_I^{(1)}}$, $\forall\, j$ (Eq. (14)).
6  Compute $M^{ART_I^{(1)}}$ (Eq. (12), Table 2's method).
7  Evaluate vigilance test $v_1$ : $M^{ART_I^{(1)}} \geq \rho_{LB}^{(1)}$.
8  if $v_1$ is satisfied (resonance) then
9      Find winning category $J \leftarrow \arg\max_j \{T_j^{ART_I^{(1)}}\}$.
10     Evaluate vigilance test $v_2$ : $M_J^{ART_I^{(1)}} \geq \rho_{UB}^{(1)}$.
11     if $v_2$ is satisfied (resonance) then
12         Update category $J$ weight vector $\boldsymbol{w}_J^{ART_I^{(1)}}$ (Eq. 5).
13     else
14         Inhibit category $J$. If there are still active categories in $ART_I^{(1)}$ then go to
             step 9; otherwise create a new category using fast commit ($\boldsymbol{w}_{new}^{ART_I^{(1)}} \leftarrow \boldsymbol{x}$).
15 else
16     Inhibit $ART_I^{(1)}$. If there are still active $F_2$ nodes in global ART then go to step 4;
         otherwise create a new ART node and apply fast commit ($\boldsymbol{w}_{new}^{ART_{new}^{(1)}} \leftarrow \boldsymbol{x}$).
```

The entries of matrices $\boldsymbol{T}_{k,l}$ and $\boldsymbol{M}_{k,l}$ are computed as:

$$t_{i,j} = \left(\frac{|\boldsymbol{w}_j^{ART_l^{(1)}} \wedge \boldsymbol{w}_i^{ART_k^{(2)}}|}{\alpha + |\boldsymbol{w}_i^{ART_k^{(2)}}|} \right)^{\gamma}, \tag{15}$$

$$m_{i,j} = \left(\frac{|\boldsymbol{w}_i^{ART_k^{(2)}}|}{|\boldsymbol{w}_j^{ART_l^{(1)}}|} \right)^{\gamma^*} t_{i,j}. \tag{16}$$

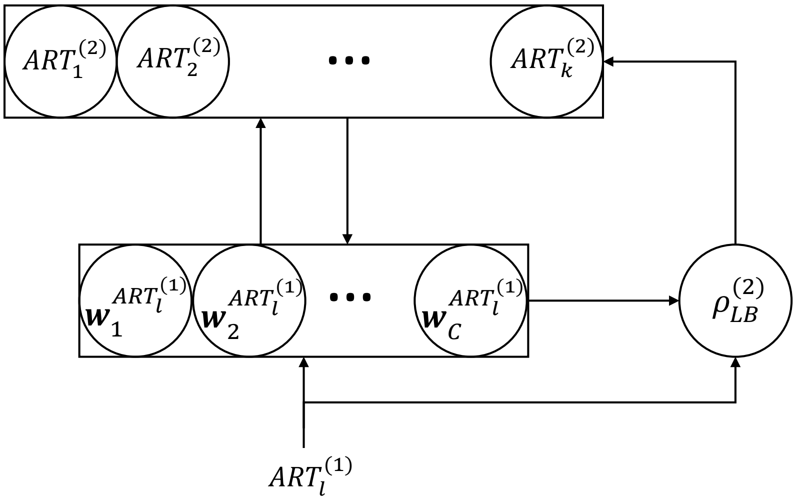

$ART_1^{(2)}$
$ART_2^{(2)}$
$ART_k^{(2)}$
$\boldsymbol{w}_1^{ART_l^{(1)}}$
$\boldsymbol{w}_2^{ART_l^{(1)}}$
$\boldsymbol{w}_C^{ART_l^{(1)}}$
$\rho_{LB}^{(2)}$
$ART_l^{(1)}$

Figure 4. Merge ART module. Each $ART^{(2)}$ is a fuzzy ART with $\rho = \rho_{UB}^{(2)}$.

The activation and match functions of the Merge ART module are listed in Table 3. When resonance is triggered, i.e., when the condition $M^{ART_K^{(2)}} \geq \rho_{LB}^{(2)}$ is satisfied, then $ART_K^{(2)}(new) \leftarrow ART_K^{(2)}(old) \cup ART_l^{(1)}$. Finally, to compress the representation, i.e., to reduce the number of categories, in the last step of the Merge ART procedure, the category weight vectors $\boldsymbol{w}^{ART_k^{(2)}}$ and instance countings $n^{ART_k^{(2)}}$ of each local ART module are fed to a fuzzy ART with higher order activation and match functions, using the parameters $\rho = \rho_{UB}^{(2)}$, $\gamma^* = 1$, and γ; in this case, when a category learns using Eq. (5) then its instance counting is updated as $n^{new} = n^{old} + n^w$, where n^w is the instance counting of the category w presented as an input.

The Merge ART module can be triggered at any stage during incremental learning. For convenience, in this study it is activated by the end of one epoch (a full pass through the data, similar to (Swope, 2012)), i.e., after N samples are presented to the learning system, where N is made equal to the data cardinality. Therefore, this framework may perform online incremental approximate HAC without computing a distance matrix with the entire data or requiring full recomputations when new samples are presented. Again, as the vigilance parameter ρ_{UB} approaches 1, there is little to no data compression. Merge

Table 3. Merge ART's activation and match functions.

Method	$T^{ART_k^{(2)}} = f(\cdot)$	$M^{ART_k^{(2)}} = g(\cdot)$
single	$\max_{i,j}\left([t_{ij}]\right)$	$\max_{i,j}\left([m_{ij}]\right)$
complete	$\min_{i,j}\left([t_{ij}]\right)$	$\min_{i,j}\left([m_{ij}]\right)$
median	$\operatorname{median}_{i,j}\left([t_{ij}]\right)$	$\operatorname{median}_{i,j}\left([m_{ij}]\right)$
average	$\frac{1}{RC}\sum_{i=1}^{R}\sum_{j=1}^{C} t_{ij}$	$\frac{1}{RC}\sum_{i=1}^{R}\sum_{j=1}^{C} m_{ij}$
weighted[a]	$\sum_{i=1}^{R}\sum_{j=1}^{C} p_i p_j t_{ij}$	$\sum_{i=1}^{R}\sum_{j=1}^{C} p_i p_j m_{ij}$
centroid[b]	$\left(\frac{\lvert\boldsymbol{w}_c^{ART_k^{(2)}} \wedge \boldsymbol{w}_c^{ART_l^{(1)}}\rvert}{\alpha+\lvert\boldsymbol{w}_c^{ART_k^{(2)}}\rvert}\right)^{\gamma}$	$\left(\frac{\lvert\boldsymbol{w}_c^{ART_k^{(2)}} \wedge \boldsymbol{w}_c^{ART_l^{(1)}}\rvert}{\lvert\boldsymbol{w}_c^{ART_l^{(1)}}\rvert}\right)^{\gamma}$

[a] $p_i = \frac{n_i^{ART_k^{(2)}}}{n^{ART_k^{(2)}}}$ and $p_j = \frac{n_j^{ART_l^{(1)}}}{n^{ART_l^{(1)}}}$. This represents an a priori probability of categories i and j analogous to (Vigdor & Lerner, 2007; Williamson, 1996). Statistical independence is assumed.

[b] $\boldsymbol{w}_c^{ART_k^{(2)}}$ and $\boldsymbol{w}_c^{ART_l^{(1)}}$ are the centroids representing all categories of $ART_k^{(2)}$ and $ART_l^{(1)}$, respectively. Each of their n components is given by $w_{c,n}^{ART_k^{(2)}} = \min_j\left(w_{j,n}^{ART_k^{(2)}}\right)$ and $w_{c,n}^{ART_l^{(1)}} = \min_j\left(w_{j,n}^{ART_l^{(1)}}\right)$, where $n = \{1, ..., 2d\}$.

ART relates to traditional HAC approaches using ART's activation function as the similarity measure and the match function as the dendrogram threshold level, i.e., the activation and match functions of the Merge ART module perform an ART-based HAC using the weight vectors created by DDVFA. Algorithm 5 summarizes the Merge ART module's pseudocode.

Remark 3. Merging strategies are commonly employed in ART-based systems. The Merge ART module presented here is closely related to the ART category merging methods discussed in (Benites & Sapozhnikova, 2017; Isawa et al., 2008a,b, 2009; Swope, 2012; Zhang et al., 2006) and especially the frameworks in (Benites & Sapozhnikova, 2017; Swope, 2012). In the latter, fuzzy ART weights are merged via a fuzzy ART module with its own set of parameters. Although both the DDVFA + Merge ART and the

Algorithm 5: Merge ART module

```
Input  : DDVFA, $\left\{\alpha, \beta, \rho_{UB}^{(2)}, \rho_{LB}^{(2)}, \gamma, \gamma^*, method\right\}$ inherited from DDVFA, number of
         iterations $t$ (optional).
Output : Merge ART clusters.

/* Merging step.                                                    */
1  repeat
2      for $l = \{1, \ldots, No.\ global\ ART\ F_2\ nodes\}$ do
3          Present input node $ART_l^{(1)} \in$ DDVFA.
4          Compute $T^{ART_k^{(2)}}$, $\forall\ k$ (Table 3's method).
5          Find the winning node $K \leftarrow \arg\max_k \{T^{ART_k^{(1)}}\}$.
6          Compute $M^{ART_K^{(2)}}$ (Table 3's method).
7          Evaluate vigilance test $v_1 : M^{ART_K^{(2)}} \geq \rho_{LB}^{(2)}$.
8          if $v_1$ is satisfied (resonance) then
9              $ART_K^{(2)} \leftarrow ART_K^{(2)} \cup ART_l^{(1)}$.
10         else
11             Inhibit node $K$. If there are still active nodes in Merge ART then go to
                 step 5; otherwise create a new ART node and apply fast commit
                 ($ART_{new}^{(2)} \leftarrow ART_l^{(1)}$).
12     DDVFA ← Merge ART.
   until stopping criteria: reaching a predefined number of iterations t or there is no
     change in Merge ART nodes (convergence)
/* Compression step.                                                */
13 for each $ART_k^{(2)} \in$ Merge ART do
14     $ART_k^{(2)} \leftarrow FA\left(\{\mathbf{w}, n\} \in ART_k^{(2)}, \rho_{UB}^{(2)}, \gamma, \gamma^*, \alpha, \beta\right)$.
       /* FA: Fuzzy ART algorithm ran for 1 epoch.                      */
```

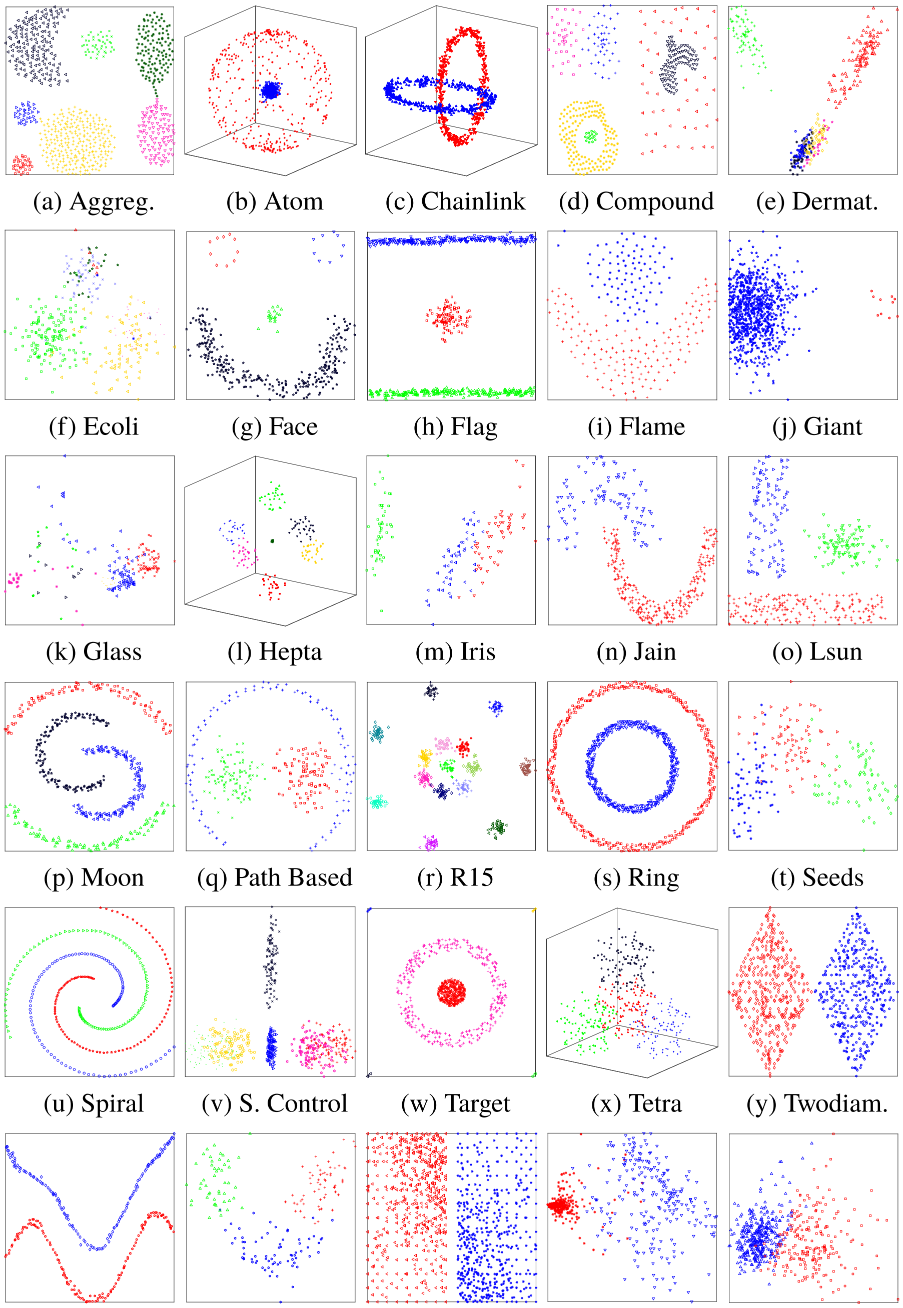
strategy in (Benites & Sapozhnikova, 2017; Swope, 2012) use a fuzzy ART framework for merging, they have the following fundamental differences: (a) Merge ART's inputs are local fuzzy ART modules from DDVFA (i.e., subsets of categories) to be merged using a fuzzy ART framework augmented with HAC-based distributed higher order activation and match functions; (b) the output of the merging procedure includes not only categories but also ART modules; (c) Merge ART's compression step does not use an activation threshold (as in (Swope, 2012)), but instead it uses higher order activation/match functions (in contrast to (Benites & Sapozhnikova, 2017; Swope, 2012)); (d) the weight update is not based on an overlap/gap between weights (as in (Swope, 2012)), but instead it follows standard fuzzy ART rules (Eq. (5)) which correspond to the weight merging in (Benites & Sapozhnikova, 2017) (and (Zhang et al., 2006) in fast learning mode); and (e) the vigilance parameter used to cluster samples is also used to merge weights during the compression step (in contrast to (Swope, 2012)).

The Merge ART module was designed such that its output can be used to replace DDVFA when the merging procedure is done. The fact that $\rho_{LB}^{(2)}$ used to concatenate DDVFA's local Fuzzy ARTs is smaller than $\rho_{UB}^{(1)}$ used to cluster the samples, ($\rho_{LB}^{(2)} = \rho_{LB}^{(1)} \leq \rho_{UB}^{(1)} = \rho_{UB}^{(2)}$), conforms with the findings reported in (Swope, 2012) that this setting yields a good performance for merging fuzzy ART weights. This is expected, since the overall architecture (DDVFA + Merge ART) is multi-layered and related to ART-based serial structures (e.g., (Bartfai, 1996; Ishihara et al., 1995)), which in turn typically follow similar parameterization.

4. EXPERIMENTAL SETUP

4.1. DATA SETS

A mix of 30 real-world and artificial benchmark data sets comprising diverse characteristics were used in the experiments. They are available at the UCI Machine Learning Repository (Bache & Lichman, 2013), Fundamental Clustering Problem Suite (Ultsch, 2005), Clustering data sets (Fränti, Pasi et al., 2015), and Data package (Ilc, 2013). Figure 5 illustrates these data sets, and Table 4 summarizes their characteristics. Linear normalization was applied to all data sets to scale their features to the range $[0, 1]$, as well as complement coding, which is a useful data representation technique to mitigate a type of category proliferation in fuzzy ART caused by weight erosion.



(a) Aggreg. (b) Atom (c) Chainlink (d) Compound (e) Dermat.

(f) Ecoli (g) Face (h) Flag (i) Flame (j) Giant

(k) Glass (l) Hepta (m) Iris (n) Jain (o) Lsun

(p) Moon (q) Path Based (r) R15 (s) Ring (t) Seeds

(u) Spiral (v) S. Control (w) Target (x) Tetra (y) Twodiam.

(z) Wave (aa) Wine (ab) Wingnut (ac) Wisconsin (ad) WDBC

Figure 5. Data sets used in the experiments. Solely for visualization purposes, the data sets with more than 3 features (i.e., *Dermatology*, *Iris*, *Wine*, *Seeds*, *Wisconsin*, *WDBC*, *Synthetic Control*, *Glass*, and *Ecoli*) are depicted using principal component analysis projection. The data sets' features and projections are scaled to the range [0, 1].

Table 4. Summary of the data sets' characteristics.

Data set	(N, d, k)	type	reference(s)
Aggregation	(788,2,7)	Artificial	(Fränti, Pasi et al., 2015; Gionis et al., 2007)
Atom	(800,3,2)	Artificial	(Ultsch, 2005)
Chainlink	(1000,3,2)	Artificial	(Ultsch, 2005)
Compound	(399,2,6)	Artificial	(Fränti, Pasi et al., 2015; Zahn, 1971)
Dermatology	(358,34,6)	Real World	(Bache & Lichman, 2013)
Ecoli	(336,7,8)	Real World	(Bache & Lichman, 2013)
Face	(320,2,4)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011)
Flag	(640,2,3)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011, 2012)
Flame	(240,2,2)	Artificial	(Fränti, Pasi et al., 2015; Fu & Medico, 2007)
Giant	(862,2,2)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011, 2012)
Glass	(214,10,6)	Real World	(Bache & Lichman, 2013)
Hepta	(212,3,7)	Artificial	(Ultsch, 2005)
Iris	(150,4,3)	Real World	(Bache & Lichman, 2013; Fisher, 1936)
Jain	(373,2,2)	Artificial	(Fränti, Pasi et al., 2015; Jain & Law, 2005)
Lsun	(400,2,3)	Artificial	(Ultsch, 2005)
Moon	(514,2,4)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011, 2012)
Path based	(300,2,3)	Artificial	(Chang & Yeung, 2008; Fränti, Pasi et al., 2015)
R15	(600,2,15)	Artificial	(Fränti, Pasi et al., 2015; Veenman et al., 2002)
Ring	(800,2,2)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011, 2012)
Seeds[a]	(210,7,3)	Real World	(Bache & Lichman, 2013; Charytanowicz et al., 2010)
Spiral	(312,2,3)	Artificial	(Chang & Yeung, 2008; Fränti, Pasi et al., 2015)
Synthetic Control[b]	(600,60,6)	Real World	(Bache & Lichman, 2013)
Target	(770,2,6)	Artificial	(Ultsch, 2005)
Tetra	(400,3,4)	Artificial	(Ultsch, 2005)
Two Diamonds	(800,2,2)	Artificial	(Ultsch, 2005)
Wave	(287,2,2)	Artificial	(Ilc, 2013; Ilc & Dobnikar, 2011, 2012)
Wine	(178,13,3)	Real World	(Bache & Lichman, 2013)
Wingnut	(1016,2,2)	Artificial	(Ultsch, 2005)
Wisconsin	(683,9,2)	Real World	(Bache & Lichman, 2013)
WDBC[c]	(569,30,2)	Real World	(Bache & Lichman, 2013)

The (N, d, k) triad represents the number of samples, features and clusters of a data set.
[a] The contributors gratefully acknowledge support of their work by the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.
[b] Image courtesy of Eamonn Keogh.
[c] Wisconsin Diagnostic Breast Cancer.

4.2. CLUSTERING ALGORITHMS AND PARAMETER TUNING

To set the parameters of the clustering algorithms employed in the experiments, grid searches were performed through their parameter spaces. For all algorithms, the best solution was selected according to the parameter combination that yielded the peak average performance.

4.2.1. ART-Based Clustering Methods. Fuzzy ART, fuzzy topoART, and DVFA were compared to DDVFA. In the experiments performed, fuzzy ART's, DVFA's and DDVFA's vigilance parameters were scanned in the range $[0, 1]$ with identical step sizes equal to 0.01 (DVFA's and DDVFA's vigilances were also subjected to the constraint $\rho_{UB} \geq \rho_{LB}$). For all fuzzy ART modules, the maximum number of epochs was set to 1 (online mode), the choice parameter (α) was set to 0.001, and the learning rate (β) was set to 1 (fast learning). DDVFA's parameters γ^* and γ were set to 1 and 3, respectively; and, for simplicity, $\rho_{UB}^{(1)} = \rho_{UB}^{(2)}$ and $\rho_{LB}^{(1)} = \rho_{LB}^{(2)}$. Moreover, in all the fuzzy ART implementations, no uncommitted category participated in the winner-take-all competitive process. If none of the current committed categories satisfy the vigilance criteria, then a new one is created and set to the current sample (fast commit). Regarding topoART, the parameters ρ_a, β_{sbm}, ϕ and τ were scanned in the ranges $[0, 1]$ with a step size of 0.008, $[0, 0.75]$ with a step size of 0.25, $[1, 4]$ with a step size of 1, and $[10\%, 30\%]$ of the data cardinality with a step size of 10%, respectively. These ranges and step sizes generated approximately the same number of parameter combinations for topoART, DVFA, and DDVFA. Module B's clusters were taken as topoART's output. Finally, for all these methods, 30 runs were performed for each data set in both random and VAT ordered presentation scenarios.

4.2.2. Non-ART-Based Clustering Methods. Density-based spatial clustering of applications with noise (DBSCAN) (Ester et al., 1996), affinity propagation (AP) (Frey & Dueck, 2007), k-means (MacQueen, 1967), and single linkage hierarchical agglomerative clustering (SL-HAC) (Xu & Wunsch II, 2009) were compared to DDVFA. In the experiments performed, DBSCAN's $MinPts$ parameter was varied in the range $[1, 4]$ with a step size of 1, while eps was scanned in the range $[0, \sqrt{d}]$ with equally spaced 1300 values, where d is the dimensionality of the data (thus encompassing the full range of possible distance values in the d-dimensional unit cube). The number of clusters k in k-means was varied in the range $[1, N]$, where N is the cardinality of the data set (thus encompassing the full range of possible values for the number of clusters). Additionally, k-means was repeated 10 times, and the best solution, according to the cost function being minimized, was selected for each value of k. The AP's damping factor λ was varied in the range [0.5, 1] with equally spaced 5200 values, and the preference parameter was set as the median of the data samples' similarities. The maximum iteration limit was set to 200 for AP and 300 for k-means. SL-HAC used Euclidean distance, and its dendrogram was cut at all merging levels. Finally, for all these methods, a single run was performed for each randomized data set, since they are global approaches that are either not (or almost not) order dependent.

4.3. CLUSTERING PERFORMANCE ASSESSMENT

The adjusted rand index (AR) (Hubert & Arabie, 1985) is an external cluster validity index commonly used in the unsupervised learning literature to measure the level of agreement between a data sets' reference partition (i.e., ground truth structure) and a discovered partition (Xu & Wunsch II, 2009). It was used in this work to evaluate the quality of the solutions returned by all clustering algorithms. The (AR) is defined as:

$$AR = \frac{\binom{N}{2}(tp + tn) - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}{\binom{N}{2}^2 - [(tp + fp)(tp + fn) + (fn + tn)(fp + tn)]}, \tag{17}$$

where tp, tn, fp and fn stand for true positive, true negative, false positive, and false negative, respectively.

4.4. STATISTICAL ANALYSIS METHODOLOGY

The clustering algorithms were compared following the procedures discussed in (Demšar, 2006):

1. The quantities of interest (i.e., performance in terms of AR and network compactness) were tested for equality using Iman-Davenport's correction (Iman & Davenport, 1980) of Friedman's non-parametric rank sum test (Friedman, 1937, 1940).

2. If there was sufficient evidence to reject the null hypothesis, then a critical difference (CD) diagram (Demšar, 2006) was generated using Nemenyi's post-hoc test (Nemenyi, 1963).

4.5. SOFTWARE AND CODE

The experiments were conducted using MATLAB, scikit-learn (Pedregosa et al., 2011), Orange (Demšar et al., 2013), and Cluster Validity Analysis Platform (Wang et al., 2009). The MATLAB code for fuzzy ART, DVFA, and DDVFA is available at the Applied Computational Intelligence Laboratory group GitHub repositories[1,2]. The topoART experiments were carried out using LibTopoART[3] (Tscherepanow, 2010), whereas the other clustering algorithms' implementations were from scikit-learn[4].

5. RESULTS AND DISCUSSION

5.1. DDVFA RESULTS WITH PRE- AND POST-PROCESSING

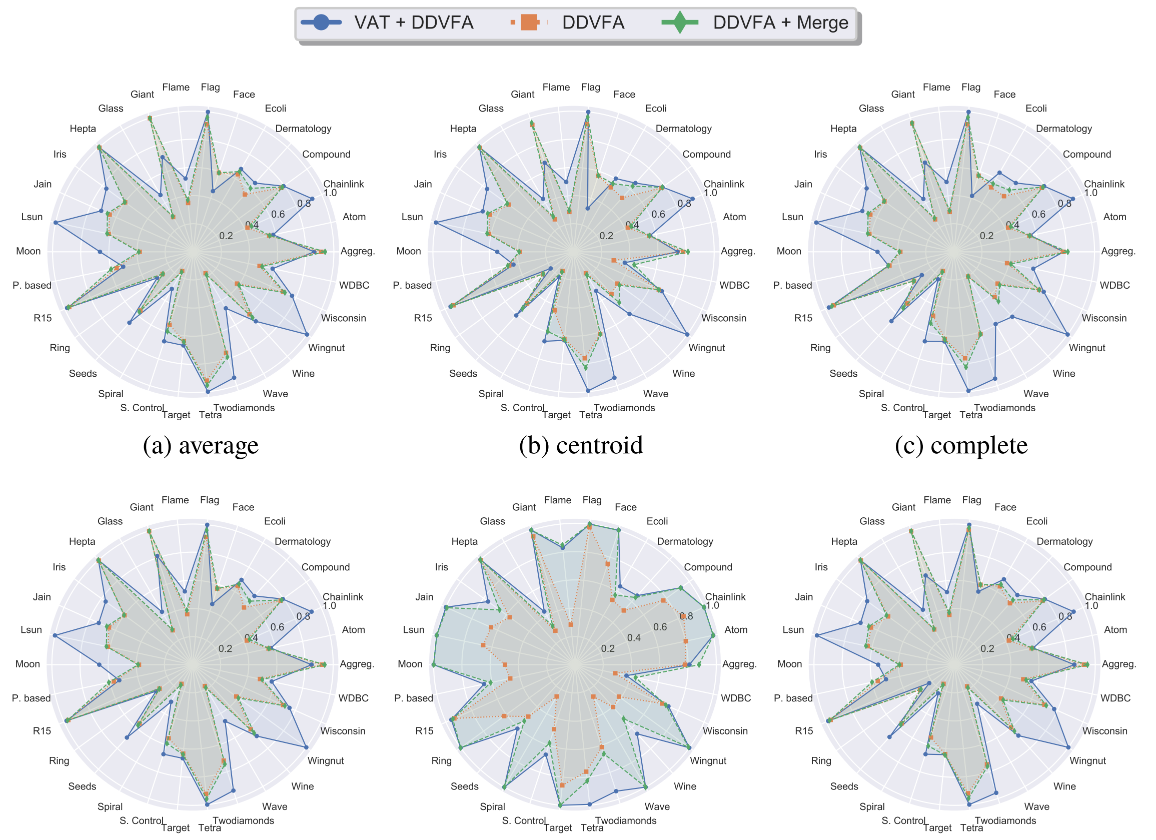
This study investigates DDVFA's order of presentation dependency by analyzing two frameworks: an offline approach that consists of pre-ordering the shuffled samples using VAT (Bezdek & Hathaway, 2002), as per (Brito da Silva & Wunsch II, 2018a), and an online approach in which the samples are solely randomized prior to presentation. The latter is a more realistic scenario when an online incremental learner is required, i.e., a learning system is confronted with a data stream. That is why all the experiments were conducted with one epoch (single pass), so each data sample is only presented once.

[1] https://github.com/ACIL-Group/DVFA.

[2] https://github.com/ACIL-Group/DDVFA.

[3] LibTopoART v0.74, available at https://www.libtopoart.eu.

[4] http://scikit-learn.org



(a) average

(b) centroid

(c) complete

(d) median

(e) single

(f) weighted

Figure 6. Radar charts of the peak average performances (*AR*) of all three different DDVFA systems, which are grouped by the type of activation/match functions (a)-(f). The results are based on 30 runs per data set using $\gamma^* = 1$ and $\gamma = 3$. Typically, VAT pre-ordering yielded the best performance, while DDVFA and DDVFA + Merge ART appear to yield a similar performance, with the exception of the single-linkage-based DDVFA, in which using Merge ART makes a noticeable difference when compared to DDVFA by itself.

Employing the methodology described in subsection 4.2, the experiments were performed with the following three systems: (1) DDVFA, (2) VAT + DDVFA, and (3) DDVFA + Merge ART. The results are summarized in Figure 6, which depicts radar charts of the peak average performance of all the mentioned systems grouped by the type of HAC-based activation/match functions (i.e., per Tables 2 and 3's method): (6a) average, (6b) centroid, (6c) complete, (6d) median, (6e) single, and (6f) weighted. It shows that, in general, VAT pre-ordering yields a better performance than pure DDVFA or post-processing with Merge ART. The latter approaches yielded a similar performance across all types of activation/match functions, except for the single-linkage based DDVFA, in which using Merge ART makes a significant difference compared to DDVFA by itself. For instance, Figure 7 illustrates the outputs of DDVFA before and after cascading it with Merge ART for the *Spiral*, *Wave*, *Atom* and *Chainlink* data sets.

5.1.1. Statistical Analysis of Performance. Using the Iman-Davenport test, a statistical analysis was conducted to quantitatively assess if the performances of the different types of HAC-based activation/match functions (average vs. centroid vs. complete vs. median vs. single vs. weighted) were equivalent when fixing the type of DDVFA system. All these performance equivalency hypotheses were rejected at a 0.05 significance level (Table 5). Therefore, Nemenyi's test was performed, and Figure 8 depicts the resulting CD diagrams. They indicate that the best performing groups seem to be: (Figure 8a)

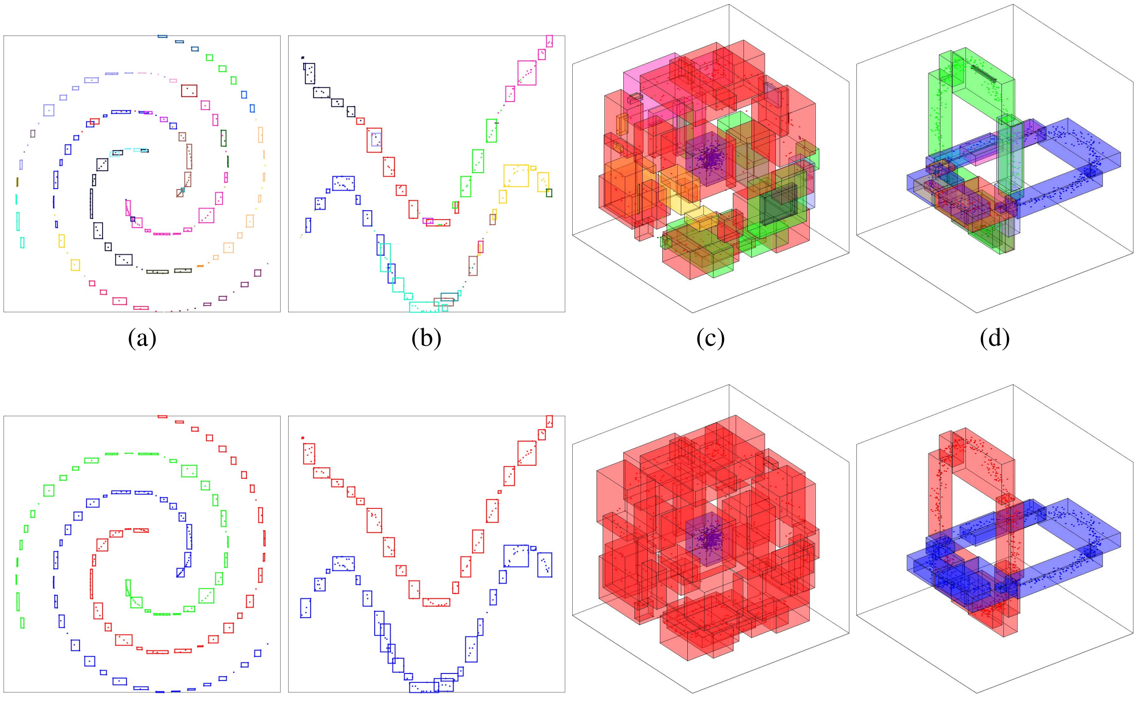


Figure 7. Output partitions of the DDVFA system (a)-(d) before, and (e)-(h) after cascading the Merge ART module for the (a,e) *Spiral*, (b,f) *Wave*, (c,g) *Atom*, and (d,h) *Chainlink* data sets.

{average, single, weighted, median}, (Figure 8b) {weighted, median}, and (Figure 8c) {single, weighted}; and the worst performing groups seem to be: (Figure 8a) {centroid}, (Figure 8b) {centroid, complete}, and (Figure 8c) {centroid, complete}, respectively. The fact that the best average rank for DDVFA is achieved by the weighted variant is expected since it considers additional information in the form of local prior probabilities.

A similar statistical analysis was conducted to determine if the performances of the systems (DDVFA vs. VAT + DDVFA vs. DDVFA + Merge ART) were equivalent when fixing the type of activation/match functions. All these null hypotheses were rejected at a 0.05 significance level (Table 6). Therefore, Nemenyi's test was performed, and, for clarity, Figure 9 solely depicts the resulting CD diagrams of selected HAC-based activation/match functions. Typically, pre-processing with VAT or post-processing with the Merge ART module are statistically equivalent, and, as expected, they are statistically better than just feeding the shuffled data directly to DDVFA.

5.1.2. Summary. The statistical analysis suggests that pre-processing with VAT or post-processing with Merge ART yields better results than just DDVFA. Furthermore, in general, single, median, average and weighted HAC-based activation/match functions appear to be statistically equivalent. Thus, the recommended systems are DDFVA + Merge ART for online learning mode and random presentation, and VAT + DDVFA for offline learning mode and applications where pre-ordering is feasible; for both of these systems the single-linkage variant is recommended since it appeared in the top 2 average rank for both learning modes.

5.2. PERFORMANCE COMPARISON 1: ART-BASED CLUSTERING ALGORITHMS

Table 7 lists the *AR* peak average performance of fuzzy ART, DVFA, topoART B, and DDVFA for both random and VAT ordered presentation scenarios. Given the results of Section 5.1's statistical analyses, the VAT + DDVFA and DDVFA + Merge ART systems

Table 5. A statistical comparison of the different HAC activation/match functions' performances per DDVFA system: Friedman-Iman-Davenport p-values.

System	DDVFA	VAT + DDVFA	DDVFA + Merge ART
p-value[a]	1.1056e-09	4.2657e-08	6.8745e-13

[a] Considering a given system, all HAC activation/match function types are statistically compared.

Table 6. A statistical comparison of the different systems' performances per HAC activation/match function type: Friedman-Iman-Davenport p-values.

Method	average	centroid	complete	median	single	weighted
p-value[a]	3.1048e-11	3.7364e-10	2.4092e-14	3.8147e-13	1.1102e-16	9.8684e-10

[a] Considering a given activation/match function type, all three DDVFA systems are statistically compared.

were selected, and the performance was recorded with respect to single linkage-based activation and match functions variant.

5.2.1. Statistical Analysis of Performance. The hypothesis that these algorithms perform equally was tested using the Iman-Davenport statistic and rejected at a 0.05 significance level for both random (p-value=1.1102E-16) and VAT orderings (p-value=3.2012E-07). Therefore, the CD diagrams were further computed, as shown in Figure 10, using Nemenyi's test. As shown, VAT pre-processing (offline incremental mode) equalizes performance, such that all multi-prototype ART-based algorithms become statistically similar, while also outperforming fuzzy ART. Alternately, when data is presented randomly in an online incremental mode DDVFA + Merge ART yields a statistically better performance than all the other ART-based algorithms at a 0.05 significance level. DVFA and topoART B were observed to be statistically equivalent while also surpassing standard fuzzy ART (as expected per (Brito da Silva et al., 2019a)).

Table 7. Experimental results summary: performance in terms of *AR* (mean ± standard deviation).

Data set	ART-based Clustering Algorithms[a]								Non-ART-based Clustering Algorithms[b]			
	Random Order				VAT pre-order[d]							
	Fuzzy ART	DVFA	TopoART B	DDVFA[c]	Fuzzy ART	DVFA	TopoART B	DDVFA	SL-HAC	DBSCAN	K-means	AP
Aggregation	0.5085 ± 0.1014	0.7428 ± 0.0892	**0.8828 ± 0.0610**	0.8780 ± 0.0500	0.7502 ± 0.0093	0.8089 ± 0.0000	**0.9810 ± 0.0005**	0.8095 ± 0.0001	0.8186	0.9039	0.7906	0.6783
Atom	0.5128 ± 0.0028	0.5739 ± 0.0746	0.9033 ± 0.1777	**1.0000 ± 0.0000**	0.8740 ± 0.0452	**1.0000 ± 0.0000**	0.9278 ± 0.0075	**1.0000 ± 0.0000**	1.0000	1.0000	0.5877	0.5232
Chainlink	0.3714 ± 0.2440	0.4002 ± 0.1616	0.9993 ± 0.0018	**1.0000 ± 0.0000**	0.9510 ± 0.0317	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	1.0000	1.0000	0.2528	0.2567
Compound	0.6026 ± 0.1338	0.6674 ± 0.1053	0.7964 ± 0.0658	**0.9242 ± 0.0000**	0.7860 ± 0.0040	0.9182 ± 0.0000	0.8885 ± 0.0601	**0.9258 ± 0.0000**	0.9270	0.9531	0.7510	0.5607
Dermatology	0.2073 ± 0.0664	0.6302 ± 0.1143	0.4422 ± 0.1311	**0.6399 ± 0.0603**	0.5994 ± 0.0658	0.5994 ± 0.0658	**0.7224 ± 0.0277**	0.6576 ± 0.0877	0.3740	0.4273	0.8703	0.7358
Ecoli	0.2192 ± 0.0740	**0.6074 ± 0.1252**	0.5170 ± 0.0956	0.5687 ± 0.0887	0.5081 ± 0.0033	0.6102 ± 0.0041	**0.6580 ± 0.0093**	0.6398 ± 0.0007	0.5101	0.5075	0.6966	0.2516
Face	0.5319 ± 0.1847	0.5319 ± 0.1847	0.9941 ± 0.0106	**1.0000 ± 0.0000**	0.3848 ± 0.2697	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	1.0000	1.0000	0.2815	0.3014
Flag	0.8107 ± 0.1649	0.8107 ± 0.1649	0.9998 ± 0.0009	**1.0000 ± 0.0000**	0.7374 ± 0.0000	**1.0000 ± 0.0000**	0.9321 ± 0.0799	**1.0000 ± 0.0000**	1.0000	1.0000	0.7184	0.7147
Flame	0.2041 ± 0.0706	0.4075 ± 0.2072	0.4440 ± 0.1429	**0.8508 ± 0.0000**	0.4563 ± 0.0151	0.5921 ± 0.2363	**0.9766 ± 0.0083**	0.8310 ± 0.0000	0.9172	0.9659	0.4616	0.4871
Giant	0.8416 ± 0.3613	0.8416 ± 0.3613	0.9956 ± 0.0244	**1.0000 ± 0.0000**	0.6635 ± 0.4840	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	1.0000	1.0000	0.0444	0.0210
Glass	0.1707 ± 0.0488	**0.3818 ± 0.1170**	0.3238 ± 0.1199	0.3162 ± 0.0000	0.3507 ± 0.0299	0.3507 ± 0.0299	0.4191 ± 0.0388	**0.4340 ± 0.0681**	0.3210	0.3210	0.5456	0.3405
Hepta	0.8923 ± 0.0399	0.9865 ± 0.0248	0.9628 ± 0.0240	**1.0000 ± 0.0000**	0.9345 ± 0.0573	**1.0000 ± 0.0000**	0.9433 ± 0.0225	**1.0000 ± 0.0000**	1.0000	1.0000	1.0000	1.0000
Iris	0.4863 ± 0.0748	0.6526 ± 0.1075	0.6048 ± 0.0808	**0.6596 ± 0.0000**	0.7236 ± 0.1714	0.7247 ± 0.1703	**0.8227 ± 0.0707**	0.7600 ± 0.0061	0.6141	0.6141	0.7163	0.7565
Jain	0.5629 ± 0.2221	0.5950 ± 0.2672	0.7578 ± 0.1537	**0.9914 ± 0.0000**	0.7124 ± 0.1672	**1.0000 ± 0.0000**	0.6958 ± 0.0003	**1.0000 ± 0.0000**	0.9758	0.9758	0.5767	0.2977
Lsun	0.4368 ± 0.1897	0.6415 ± 0.1498	0.7890 ± 0.1489	**1.0000 ± 0.0000**	0.9263 ± 0.0656	**1.0000 ± 0.0000**	0.9613 ± 0.0137	**1.0000 ± 0.0000**	1.0000	1.0000	0.6619	0.6660
Moon	0.2774 ± 0.0668	0.3721 ± 0.1020	0.6829 ± 0.1741	**1.0000 ± 0.0000**	0.5398 ± 0.0330	0.9669 ± 0.0315	0.6651 ± 0.1230	**1.0000 ± 0.0000**	1.0000	1.0000	0.3508	0.3162
Path based	0.2671 ± 0.0976	0.4780 ± 0.0900	0.5278 ± 0.0593	**0.6097 ± 0.0145**	0.4947 ± 0.0123	**0.8513 ± 0.1218**	0.6236 ± 0.0355	0.6573 ± 0.0008	0.6122	0.6087	0.5667	0.4525
R15	0.7922 ± 0.0459	0.9347 ± 0.0360	0.9205 ± 0.0280	**0.9465 ± 0.0227**	0.9634 ± 0.0023	0.9634 ± 0.0023	**0.9857 ± 0.0001**	0.9575 ± 0.0116	0.9460	0.9347	0.9928	0.9928
Ring	0.0924 ± 0.0124	0.2022 ± 0.0429	0.9768 ± 0.0688	**1.0000 ± 0.0000**	0.2333 ± 0.0446	**1.0000 ± 0.0000**	0.8418 ± 0.1609	**1.0000 ± 0.0000**	1.0000	1.0000	0.1696	0.1366
Seeds	0.3414 ± 0.0675	**0.5373 ± 0.1289**	0.4579 ± 0.1604	0.5360 ± 0.0850	**0.6432 ± 0.0197**	**0.6432 ± 0.0197**	0.5813 ± 0.0482	0.6087 ± 0.0210	0.4259	0.4215	0.7049	0.3134
Spiral	0.0870 ± 0.0058	0.1740 ± 0.0145	0.3004 ± 0.0784	**1.0000 ± 0.0000**	0.2443 ± 0.0301	**1.0000 ± 0.0000**	0.2019 ± 0.0023	**1.0000 ± 0.0000**	1.0000	1.0000	0.1389	0.1593
Synthetic Control	0.0894 ± 0.0283	**0.5841 ± 0.0524**	0.3320 ± 0.0805	0.5831 ± 0.0358	0.6081 ± 0.0266	0.6081 ± 0.0266	0.6640 ± 0.0197	**0.6690 ± 0.0182**	0.5530	0.5525	0.6217	0.5285
Target	0.5679 ± 0.0394	0.6515 ± 0.0178	0.9989 ± 0.0020	**1.0000 ± 0.0000**	0.6407 ± 0.0023	**1.0000 ± 0.0000**	0.8950 ± 0.1068	**1.0000 ± 0.0000**	1.0000	1.0000	0.6838	0.5831
Tetra	0.3793 ± 0.0912	0.6928 ± 0.1472	0.6143 ± 0.1392	**0.8269 ± 0.1306**	0.9933 ± 0.0000	0.9933 ± 0.0000	**1.0000 ± 0.0000**	0.9933 ± 0.0000	0.9462	0.9462	1.0000	0.9178
Twodiamonds	0.2917 ± 0.0934	0.5879 ± 0.2602	0.6245 ± 0.3250	**0.6628 ± 0.4411**	**0.9570 ± 0.0508**	**0.9570 ± 0.0508**	0.9460 ± 0.0482	0.9410 ± 0.0775	0.8980	0.9266	1.0000	0.6418
Wave	0.1407 ± 0.0912	0.1929 ± 0.0437	0.4466 ± 0.1656	**1.0000 ± 0.0000**	0.3770 ± 0.1039	**1.0000 ± 0.0000**	0.3315 ± 0.0827	**1.0000 ± 0.0000**	1.0000	1.0000	0.2016	0.2075
Wine	0.0893 ± 0.0320	**0.6359 ± 0.1353**	0.4317 ± 0.1472	0.5138 ± 0.0863	0.5807 ± 0.0729	0.5846 ± 0.1580	0.5851 ± 0.0338	**0.6578 ± 0.0140**	0.4071	0.4363	0.8537	0.4464
Wingnut	0.3736 ± 0.3298	0.3736 ± 0.3298	0.3422 ± 0.2140	**0.9921 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	1.0000	1.0000	0.5047	0.4624
Wisconsin	0.4267 ± 0.1263	0.6909 ± 0.0084	0.6689 ± 0.0194	**0.7044 ± 0.0011**	0.6909 ± 0.0079	**0.7291 ± 0.0011**	0.7171 ± 0.0635	0.7240 ± 0.0060	0.7053	0.8257	0.8465	0.3997
WDBC	0.0722 ± 0.0318	**0.4772 ± 0.1568**	0.3278 ± 0.1150	0.4381 ± 0.1596	0.3257 ± 0.0026	0.4209 ± 0.0130	**0.4472 ± 0.0468**	0.3724 ± 0.0041	0.3021	0.3305	0.7302	0.4821

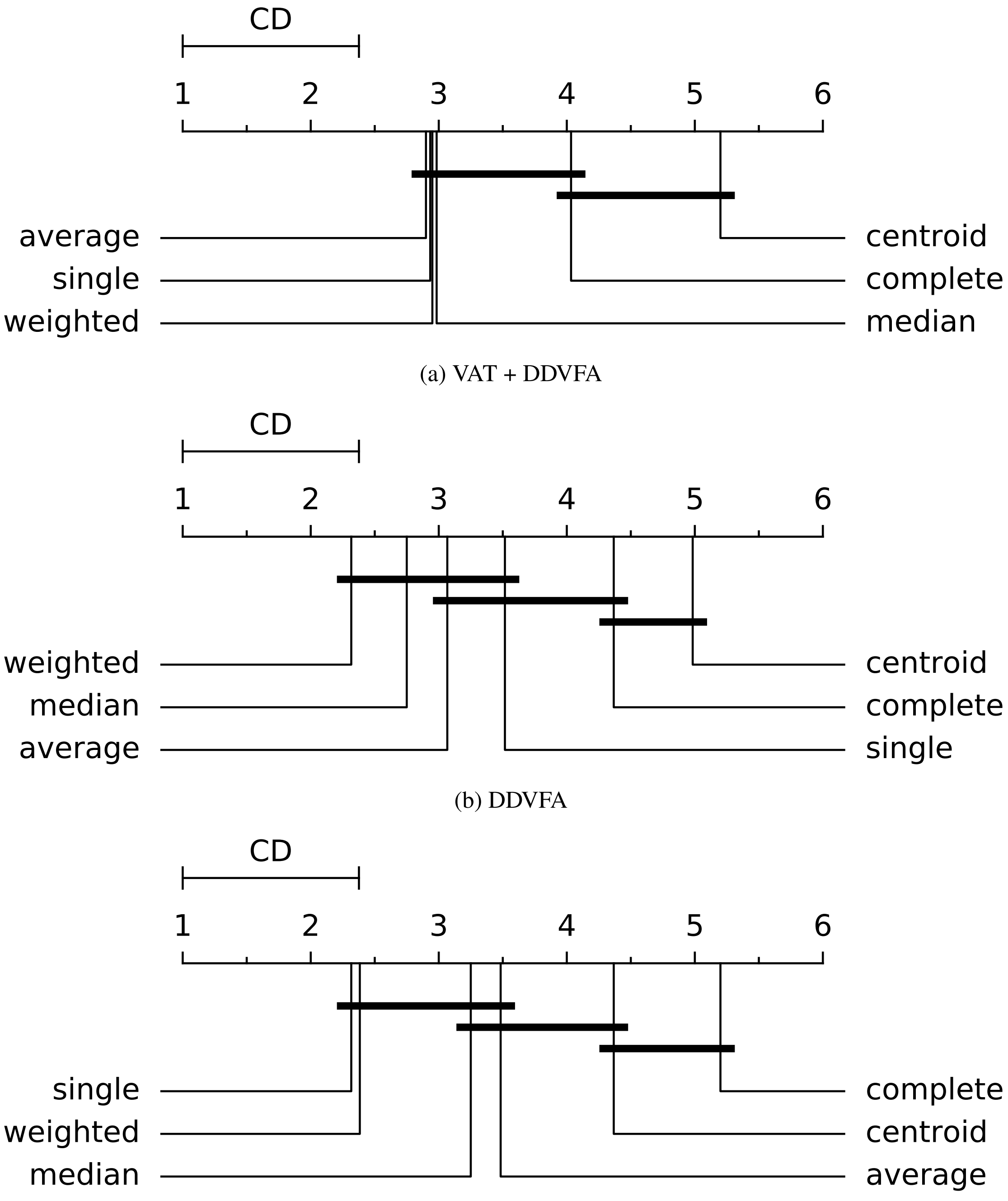
Bold values indicate best average performance across shuffled or VAT pre-ordered data (non-ART based methods are not included).

[a] Mean and standard deviations of *AR* over 30 runs with respect to the best parameter combination are reported.

[b] Since these clustering algorithms are either completely (or almost) insensitive to order presentation, data samples were randomized, a single run was performed, and the peak *AR* corresponding to the best parameters are reported (except for k-means where 10 repetitions were performed in such run).

[c] DDVFA + Merge ART system.

[d] All four ART-based algorithms were fed the VAT pre-ordered data.

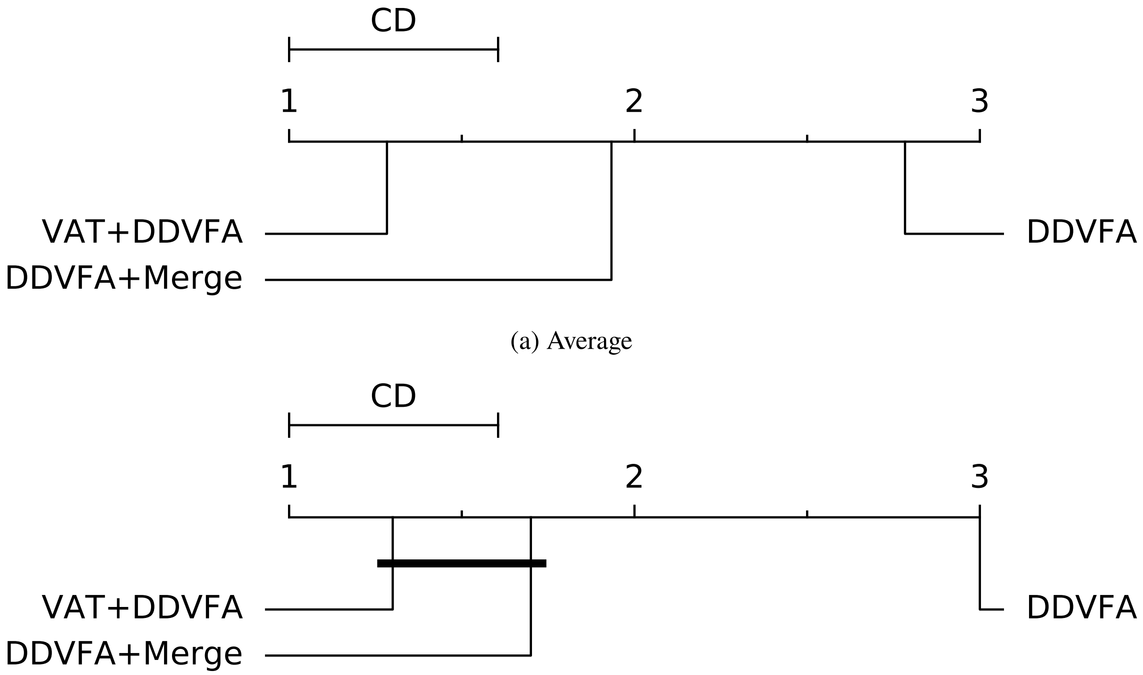


(a) VAT + DDVFA

(b) DDVFA

(c) DDVFA + Merge ART

Figure 8. CD diagrams for all three DDVFA systems considering all HAC-based distributed activation/match functions.

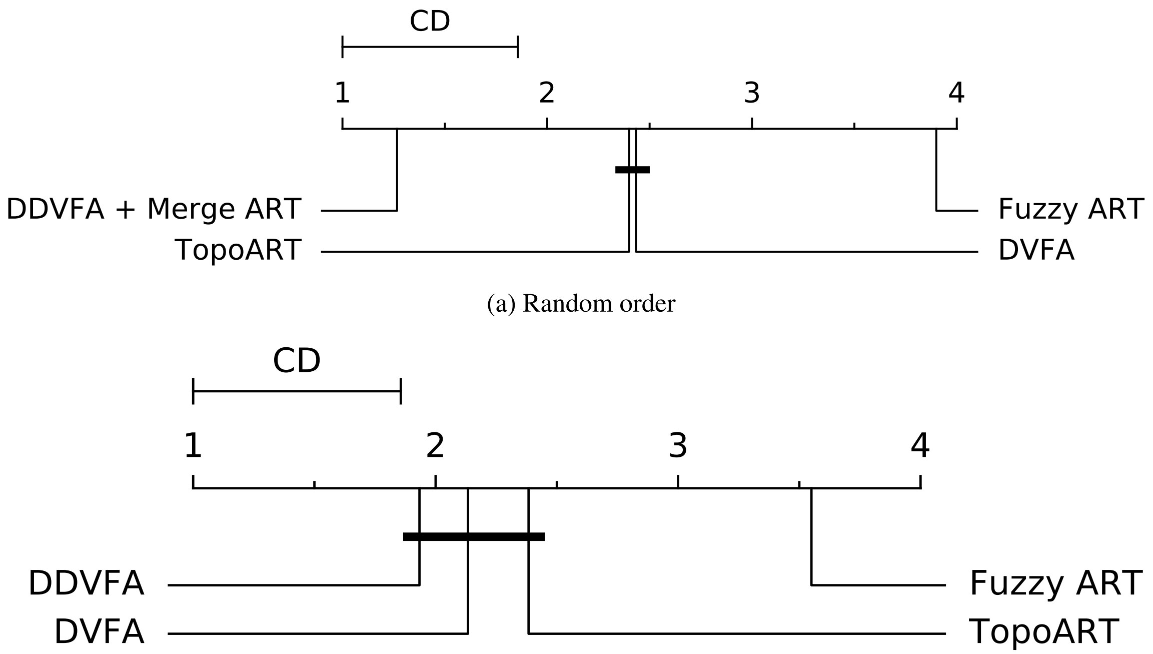


(a) Average

(b) Single

Figure 9. CD diagram for selected distributed HAC-based activation/match functions considering all three DDVFA systems. The CD diagram of the single variant is also representative for centroid, complete, median, and weighted.

5.2.2. Statistical Analysis of Compactness. The compactness of the multi-prototype ART-based networks was also compared, i.e., the number of categories that were created to represent the data sets' clusters. The hypothesis of equivalence (using Iman-Davenport's test) was rejected at a 0.05 significance level, with p-values equal to (a) 5.2039E-03 for VAT pre-ordering and (b) 1.7622E-02 for random presentation. Given this outcome, the corresponding CD diagrams were generated as shown in Figure 11 using Nemenyi's test. In online learning mode (Figure 11a), in which samples are presented randomly, topoART has the best average ranking for compactness. Yet, in offline learning mode (Figure 11b), in which order-dependence can be managed via VAT pre-processing, DDVFA has a better average compactness ranking than topoART. However, their observed compactness were



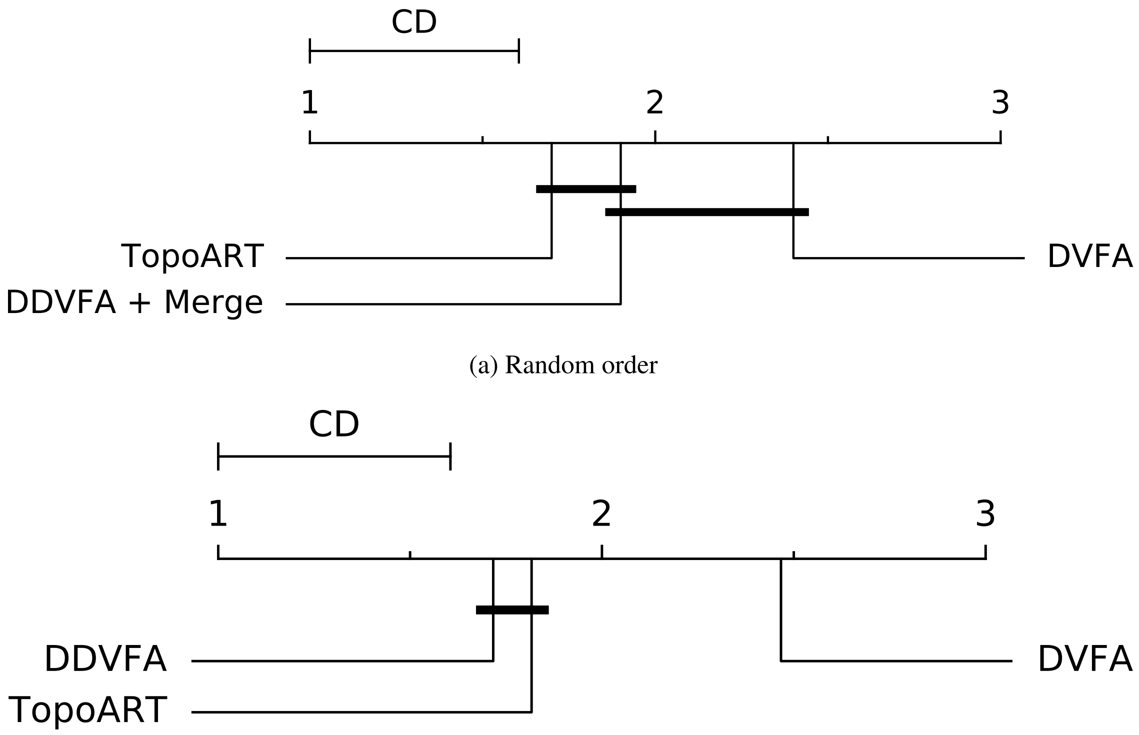
(a) Random order

(b) VAT pre-order

Figure 10. CD diagrams comparing the performance of ART-based clustering methods.

similar and with no statistically significant difference. As expected, topoART creates more compact networks than DVFA in all scenarios (Brito da Silva et al., 2019a). Note that improved compactness may be obtained by carefully tuning DDVFA's parameter γ.

5.2.3. Summary. The statistical analysis suggests that if pre-processing with VAT, then topoART, DVFA, and DDVFA seem to perform equally; whereas for random presentation DDVFA + Merge ART's performance was observed to be statistically better than the remaining ART-based systems. Moreover, no statistical differences were found between the compactness of topoART and DDVFA systems using single linkage functions for randomly or VAT ordered presentations, and both achieved a better average rank than DVFA.



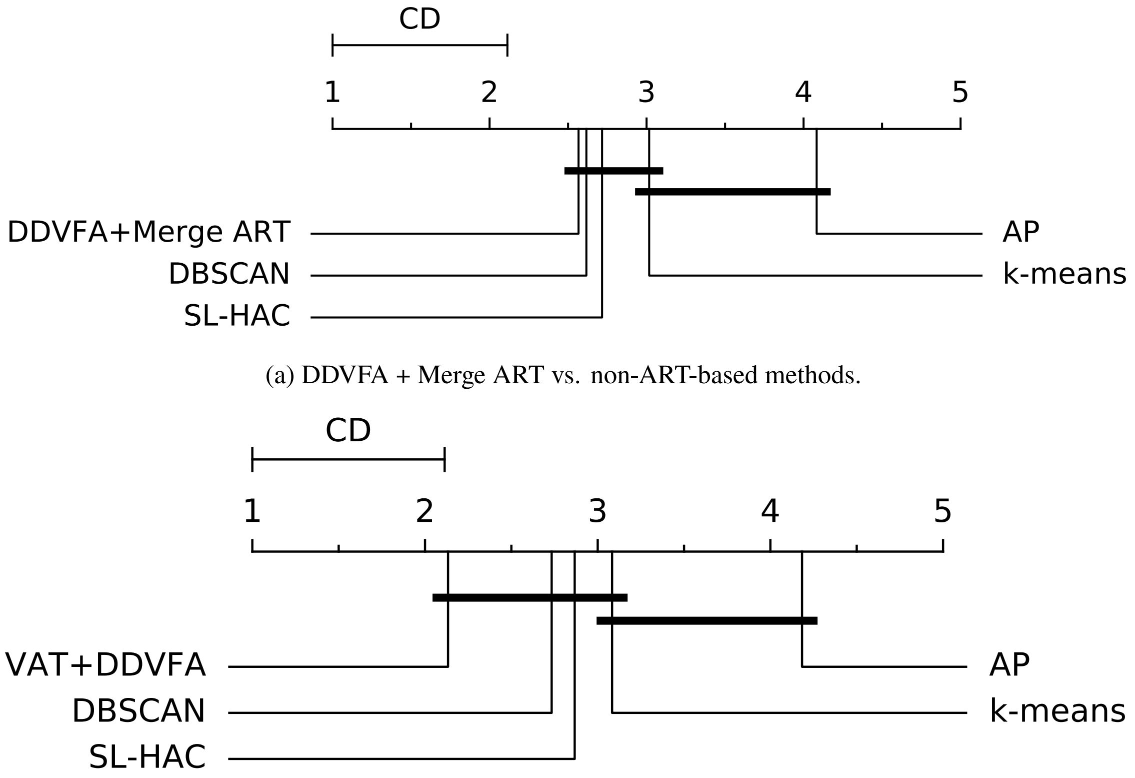
(a) Random order

(b) VAT pre-order

Figure 11. CD diagrams comparing the compactness of the multi-prototype ART-based architectures.

5.3. PERFORMANCE COMPARISON 2: NON-ART-BASED CLUSTERING ALGORITHMS

Table 7 also reports the performance of k-means, DBSCAN, affinity propagation (AP), and single linkage (SL-HAC). Again, the Iman-Davenport test was used to compare these algorithms to (a) VAT + DDVFA, and (b) DDVFA + Merge ART. These null hypotheses were rejected at a 0.05 significance level with p-values equal to (a) 5.4413E-06, and (b) 4.4746E-04. Next, the CD diagrams were generated using Nemenyi's test, as shown in Figure 12. For these data sets and at a 0.05 significance level, no statistical difference was observed between the k-means and AP, while the two DDVFA systems (VAT + DDVFA and DDVFA + Merge ART) seem to be statistically equivalent to DBSCAN, SL-HAC, and k-means. Nevertheless, both DDVFA systems have a smaller average rank value



(a) DDVFA + Merge ART vs. non-ART-based methods.

(b) VAT + DDVFA vs. non-ART-based methods.

Figure 12. CD diagrams comparing the performance of two DDVFA systems to SL-HAC, DBSCAN, k-means, and AP clustering algorithms.

(particularly when using the VAT pre-processor). This on par performance is remarkable, especially regarding the comparison with the DDVFA + Merge ART system, since in this case clustering is performed both incrementally and online, as opposed to the other global clustering methods. Re-performing the computations using the entire data set is not required if a new sample is presented (cf., SL-HAC). Therefore, it is possible to extend the current knowledge base. Moreover, the weight vectors do not cycle, and previously acquired knowledge is not forgotten (cf., k-means). These important advantages of the DDVFA systems are inherited from fuzzy ART.

5.4. COMPUTATIONAL COMPLEXITY ANALYSIS

Table 8 shows a summary of the running time complexity for each of the algorithms used in this work. The complexity shown is for a data set of N input samples, with k categories. Since each of the algorithms' running time grows linearly with the dimensionality of the input d, it was omitted from the table for the sake of brevity. For fuzzy ART, topoART, DVFA, and DDVFA algorithms, the table indicates the time complexity for a single pass through the data set (i.e., when these algorithms operate in online mode), whereas entries showing a t parameter indicate that this time grows linearly with each iteration of the respective algorithm. The indicated running time complexity is a measure of how the time needed to run each of these algorithms for a single combination of user-specified parameters grows with the size of the data set and the number of categories. Specifically, following the experimental setup described in Section 4.2, the number of user-specified parameter combinations used to tune each algorithm is reported in Table 8.

The computational complexity of DDVFA and Merge ART can be observed directly from Algorithms 4 and 5, respectively, where k refers to the total number of categories across all of the ART nodes inside DDVFA. For each input sample presentation in DDVFA, the activation and match functions are calculated once for each of those k categories, and each of the operations in Table 2 only performs an additional k computations, leading to an overall computational complexity of $O(nk)$. For the Merge ART procedure, in each of the t iterations that the algorithm performs until convergence, subsets of the k nodes are compared pair-wise with other nodes, which leads to k^2 work in the worst case. This results in overall $O(k^2t)$ computational complexity for Merge ART, although it is worth noting that this is a pessimistic upper-bound, since most of the time only small subsets of k are compared, leading to the actual work being only a fraction of k^2. Moreover, since ART-based algorithms do not require the user to directly set the value of k, the total number of categories in each of these algorithms, and therefore the value of k, will vary depending on the way the algorithm builds its internal representation.

Table 8. Time complexity and hyper-parameterization of all algorithms used in this work.

Algorithm	Time complexity	User-specified parameter(s)	No. of parameter combinations
Fuzzy ART	$O(Nk)$ (Meng et al., 2016)	ρ	101
TopoART	$O(Nk)$	$(\rho_a, \beta_{sbm}, \phi, \tau)$	5040
DVFA	$O(Nk)$	(ρ_{LB}, ρ_{UB})	5151
DDVFA	$O(Nk)$	$(\rho^{(1)}_{LB}, \rho^{(1)}_{UB})$	5151
VAT[a]	$O(N^2)$ (Havens & Bezdek, 2012)	-	-
Merge ART[b]	$O(k^2 t)$	$(\rho^{(2)}_{LB}, \rho^{(2)}_{UB})$	5151[c]
DBSCAN	$O(N^2)$ (Schubert et al., 2017)	$(MinPts, eps)$	5200
AP	$O(N^2 t)$ (Shang et al., 2012)	λ	5200
SL-HAC[d]	$O(N^2)$ (Sibson, 1973)	dendrogram cut-off level	data dependent
k-means[e]	$O(Nkt)$ (Xu & Wunsch II, 2009)	k	N (data dependent)

[a] Pre-processing.

[b] Post-processing.

[c] Number of parameter combinations for the system DDVFA + Merge ART. Note that all three DDVFA systems (viz., DDVFA, VAT + DDVFA, and DDVFA + Merge ART) undergo the same number of parameter combinations (5151), since VAT does not require parameterization and Merge ART uses $\rho^{(2)}_{LB} = \rho^{(1)}_{LB}$ and $\rho^{(2)}_{UB} = \rho^{(1)}_{UB}$.

[d] The number of possible parameter values is equal to the the total number of merging levels in the data sets' dendrogram.

[e] The number of possible parameter values is equal to the data sets' number of samples N.

5.5. DDVFA'S HYPER-PARAMETERIZATION

As discussed in Section 4.2.1, besides the choice (α) and learning (β) parameters that all fuzzy ART networks require to be set, DDVFA requires the selection of the following additional six parameters: γ^*, γ, $\rho^{(1)}_{UB}$, $\rho^{(2)}_{UB}$ $\rho^{(1)}_{LB}$, and $\rho^{(2)}_{LB}$. Note however that, in the experiments carried out, $\gamma^* = 1$, $\rho^{(1)}_{UB} = \rho^{(2)}_{UB}$, and $\rho^{(1)}_{LB} = \rho^{(2)}_{LB}$. Thus, there are effectively three additional parameter to be set: $\rho^{(1)}_{UB}$, $\rho^{(1)}_{LB}$ and γ.

Particularly, in the experiments carried out, for all three systems (DDVFA, VAT + DDVFA, and DDVFA + Merge ART), the centroid-based method consistently yielded the best results (in terms of performance and model compactness) when setting $\rho^{(1)}_{UB} = \rho^{(1)}_{LB}$ for all data sets. Similarly, for the complete method, the majority of the best results achieved by the DDVFA and DDVFA + Merge ART systems also used $\rho^{(1)}_{UB} = \rho^{(1)}_{LB}$. Under such parameter setting, DDVFA reduces in functionality to a fuzzy ART network augmented with higher order activation and match functions. Regarding the remaining HAC methods (i.e., single, median, average, and weighted) the setting of the upper and lower bound vigilance parameters is data dependent, thus requiring careful tuning as with most clustering algorithms. In practical unsupervised learning applications, one may resort to expert domain knowledge or to relative incremental cluster validity indices (Brito da Silva et al., 2019c; Ibrahim et al., 2018a,b; Moshtaghi et al., 2019) to drive the parameter tuning process.

Finally, a detailed discussion regarding the kernel width parameter γ, and its implications to model compactness (i.e., category proliferation) and performance, is provided in the following subsection. In this work, fixing γ to 2 or 3 achieved good results in the experiments carried out.

5.6. SENSITIVITY TO KERNEL WIDTH PARAMETER

To examine the behavior of the DDVFA systems with respect to parameter γ, $\gamma = 1$ and $\gamma = 3$ were arbitrarily set, and Wilcoxon's signed-ranks tests (Wilcoxon, 1945) were conducted to compare the performance and compactness of the best dual vigilance parameter combination (peak average performances over 30 runs). The results are reported in Table 9.

Regarding the HAC-based activation/match functions, a significant statistical difference for both performance and compactness was observed for (a) all DDVFA systems using the single HAC method, (b) the majority of DDVFA systems using centroid, median, and complete HAC methods. Average and weighted variants do not appear to be very much affected by changing parameter γ between these two values. With respect to the three DDVFA systems, performance and compactness are affected by parameter γ, except for the compactness of the VAT + DDVFA system which remains mostly unaffected.

Due to these statistical analysis results, the DDVFA systems' behavior was further investigated using single-linkage HAC activation/match functions with respect to parameter γ. The study is performed by varying γ in the interval $[0, 5]$ with a step size of 0.5 and observing the following aspects: peak average performance (AR), number of clusters, and number of categories created. The last two quantities were examined since DDVFA belongs to the class of multi-prototype-based clustering methods, i.e., each cluster may be

Table 9. A statistical comparison of $\gamma = 1$ versus $\gamma = 3$: Wilcoxon p-values.

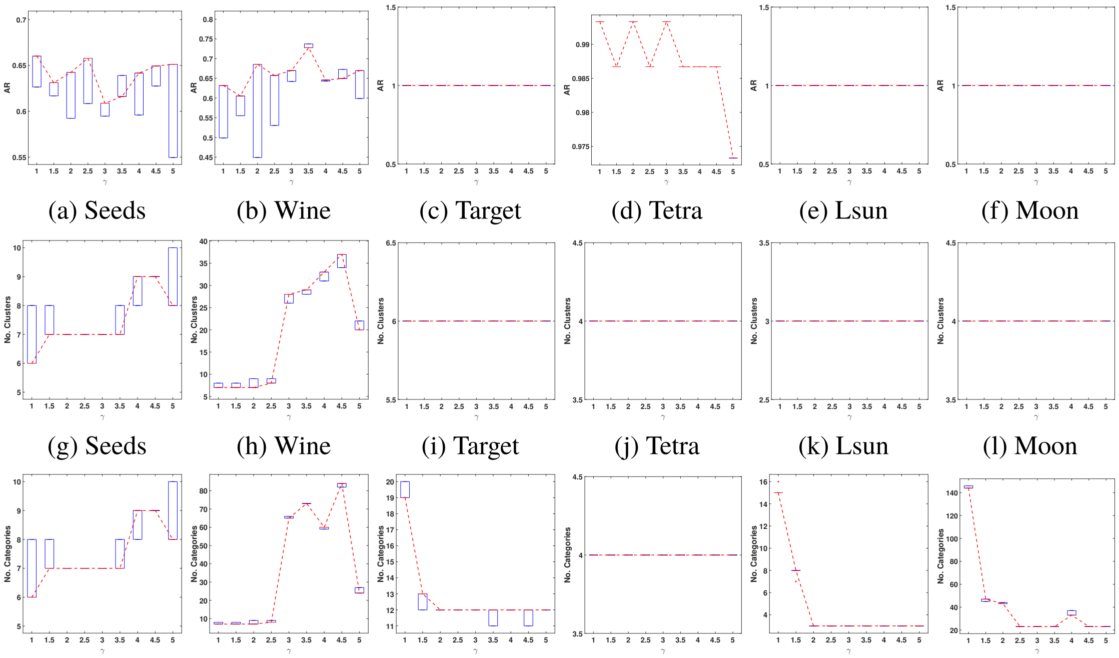
Systems	Methods					
	Average	Centroid	Complete	Median	Single	Weighted
Performance						
VAT + DDVFA	**2.1228E-02**	**4.8300E-03**	1.0200E-01	**3.9650E-02**	**3.1506E-02**	1.6480E-01
DDVFA	1.3591E-01	**1.8254E-06**	**3.7323E-04**	**5.2872E-04**	**1.9209E-06**	3.4935E-01
DDVFA + Merge ART	2.2101E-01	**3.3445E-06**	**4.0355E-04**	**1.7515E-02**	**2.6539E-03**	1.5884E-01
Compactness						
VAT + DDVFA	7.1864E-01	1.8663E-01	5.6445E-01	3.2279E-01	**1.7982E-02**	7.9707E-01
DDVFA	**6.8344E-03**	**1.7697E-03**	**7.1966E-05**	**7.9639E-03**	**6.6540E-06**	**3.0581E-03**
DDVFA + Merge ART	1.0000E+00	**4.5022E-06**	**1.5649E-05**	**3.1513E-02**	**8.0045E-04**	2.0223E-01

Bold values indicate statistically significant results.

represented by multiple categories. Such behaviors are illustrated in Figs. 13 through 15. For clarity, and according to the recommendations outlined in Section 5.1, only the behavior with respect to the data sets *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* is reported.

For each value of γ, the vigilance parameter combination corresponding to the best average performance over 10 different input permutation orders is selected. Following Occam's razor and the principle of parsimony (Duda et al., 2000), among all models that yield the best performance, the one with the simplest clustering structure is selected, i.e., the one that requires the smaller number of categories to encode its clustering partition. Thus, the depicted box-plots relate to the simplest model that achieved the peak average performance for each value of γ.

Remark 4. Note that the vigilance parameter combinations that yield each box-plot in Figs. 13 through 15 are not held constant across the different values of γ; therefore, they may not be necessarily the same. For instance, Figure 13 shows that, for the VAT + DDVFA system, given a value of γ, there is a dual vigilance parameter combination that can find the correct partitions ($AR = 1$) with similar compactness levels (number of categories) across γ values for the *Target*, *Tetra*, *Lsun*, and *Moon* data sets. Analogously, given a value γ, there is a dual vigilance parameter combination for the DDVFA + Merge ART system that yields maximum AR for the *Target*, *Lsun*, and *Moon* data sets; however,



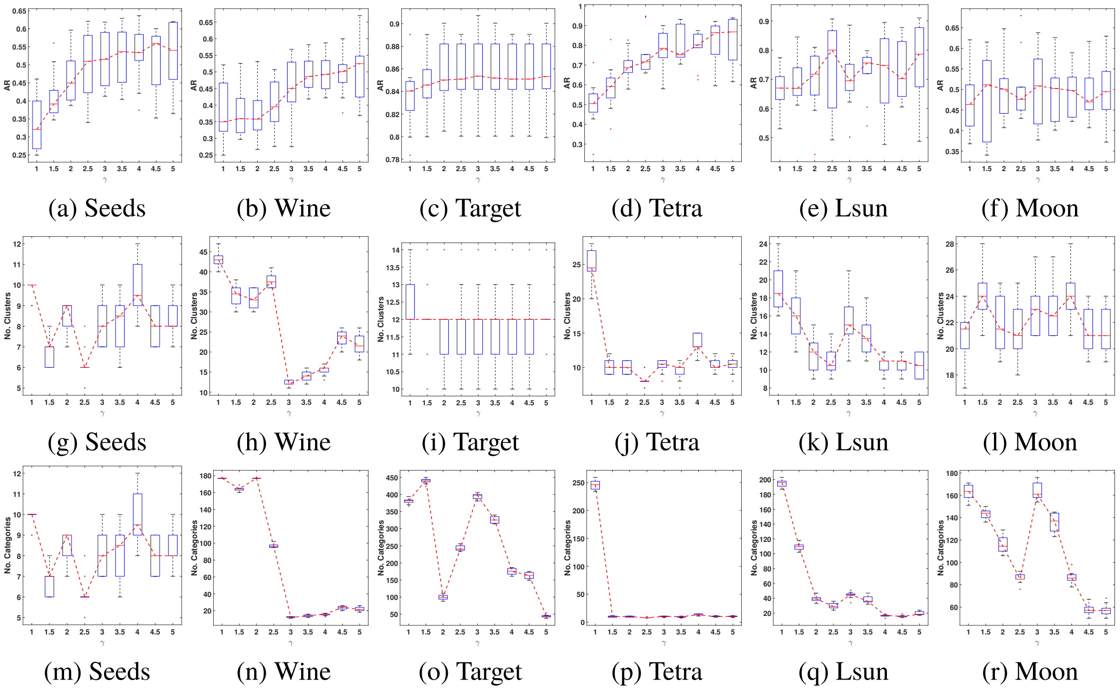
(a) Seeds (b) Wine (c) Target (d) Tetra (e) Lsun (f) Moon

(g) Seeds (h) Wine (i) Target (j) Tetra (k) Lsun (l) Moon

(m) Seeds (n) Wine (o) Target (p) Tetra (q) Lsun (r) Moon

Figure 13. The behavior of the VAT+DDVFA system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (AR), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ).

the number of categories fluctuates when the samples are randomly presented. If the dual vigilance parameter combination is held constant, e.g., by setting it to the best combination associated with $\gamma = 1$, then, for other γ values, the behaviors with respect to performance, number of clusters and categories may change for both systems, as shown in Figure 16 for the *Target* data set. Note the increase in the number of categories due to the increase of γ: the smallest dual vigilance parameter values required to achieve the best performance for $\gamma = 1$ are somewhat large, and the same values coupled with a more selective kernel (larger γ) result in more categories being created.



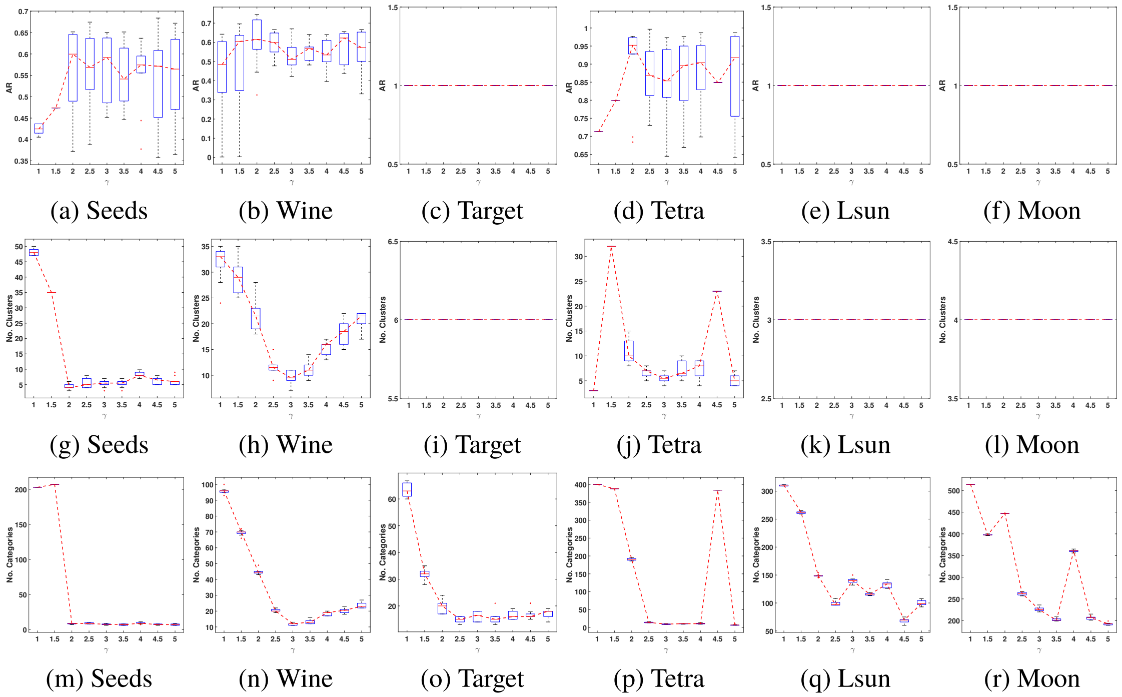
(a) Seeds (b) Wine (c) Target (d) Tetra (e) Lsun (f) Moon

(g) Seeds (h) Wine (i) Target (j) Tetra (k) Lsun (l) Moon

(m) Seeds (n) Wine (o) Target (p) Tetra (q) Lsun (r) Moon

Figure 14. The behavior of the DDVFA system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (AR), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ).

Naturally, the behavior of the DDVFA systems with respect to γ is data- and system-dependent. Although some AR performance fluctuation exists across the values of γ for some data sets, it generally seems to be fairly robust to this parameter. The number of categories, i.e., the compression level, often drastically changes with γ. For example, setting $\gamma = 1$ (i.e., using standard fuzzy ART building blocks) versus $\gamma = 2$ already yields noticeable changes in many data sets as shown in Figs. 13 through 15, especially for the DDVFA + Merge ART system. Furthermore, the number of categories appears to decrease by increasing γ as this tendency was observed in many of the data sets in Figs. 13 through 15. Specifically, Figure 17 illustrates this effect in the *Target* data set. These experimental results are consistent with previous findings in related work, in which improved memory



(a) Seeds (b) Wine (c) Target (d) Tetra (e) Lsun (f) Moon

(g) Seeds (h) Wine (i) Target (j) Tetra (k) Lsun (l) Moon

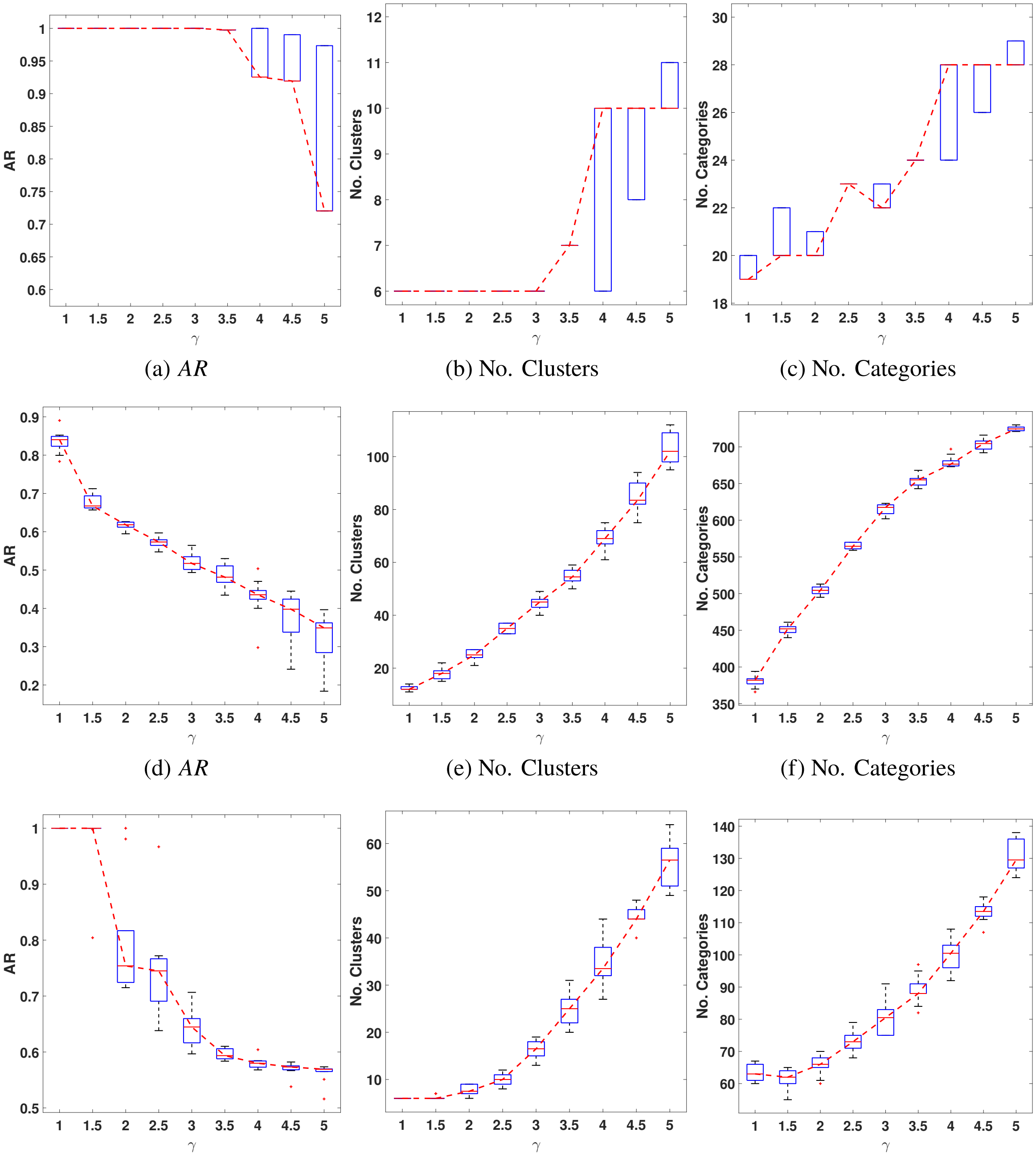
(m) Seeds (n) Wine (o) Target (p) Tetra (q) Lsun (r) Moon

Figure 15. The behavior of the DDVFA + Merge ART system with respect to parameter γ using the *Seeds*, *Wine*, *Target*, *Tetra*, *Lsun*, and *Moon* data sets: (a)-(f) peak average performance (AR), (g)-(l) number of clusters, and (m)-(r) total number of categories created. Both the number of clusters and categories are taken with respect to the most compact model that yields the depicted peak average performance (i.e., dual vigilance parameterization is *not* held constant while varying parameter γ).

compression is achieved when using power rules coupled with distributed learning in ART-systems (Carpenter, 1997; Carpenter et al., 1998). Another important aspect refers to the region of the dual vigilance parameter space which correlates with better performance; such a region seems to increase with the value of γ for some data sets (e.g., the *Target* data set in Figure 18), usually at the expense of the network's compactness.

6. CONCLUSION

This paper presents distributed dual vigilance fuzzy ART (DDVFA), a novel, modular, hierarchically self-consistent ART-based architecture for incremental, unsupervised learning. DDVFA features a number of innovations that differ from other ART-based sys-



(a) *AR* (b) No. Clusters (c) No. Categories

(d) *AR* (e) No. Clusters (f) No. Categories

(g) *AR* (h) No. Clusters (i) No. Categories

Figure 16. The behavior of the (a)-(c) VAT + DDVFA, (d)-(f) DDVFA, and (g)-(i) DDVFA + Merge ART systems for different values of parameter γ while holding the dual vigilance parameters constant. Single linkage HAC-based activation and match functions are used.

tems. It relies on dual vigilance parameters to handle data quantization (local scale) and cluster similarity (global scale), features multi-prototype representations, and higher-order

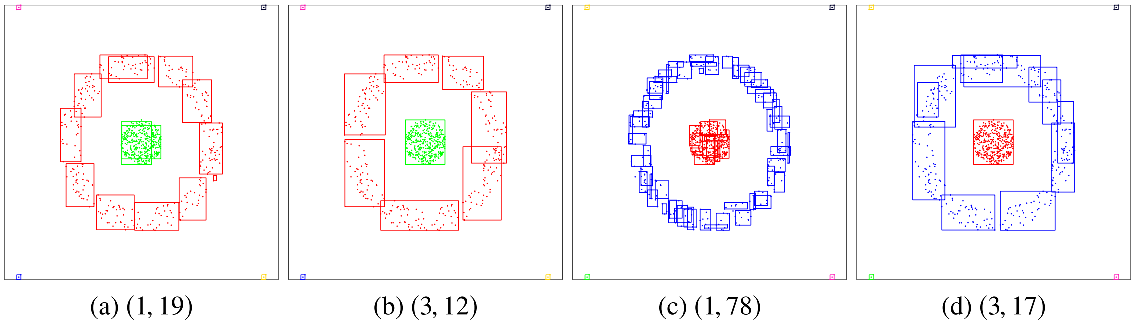
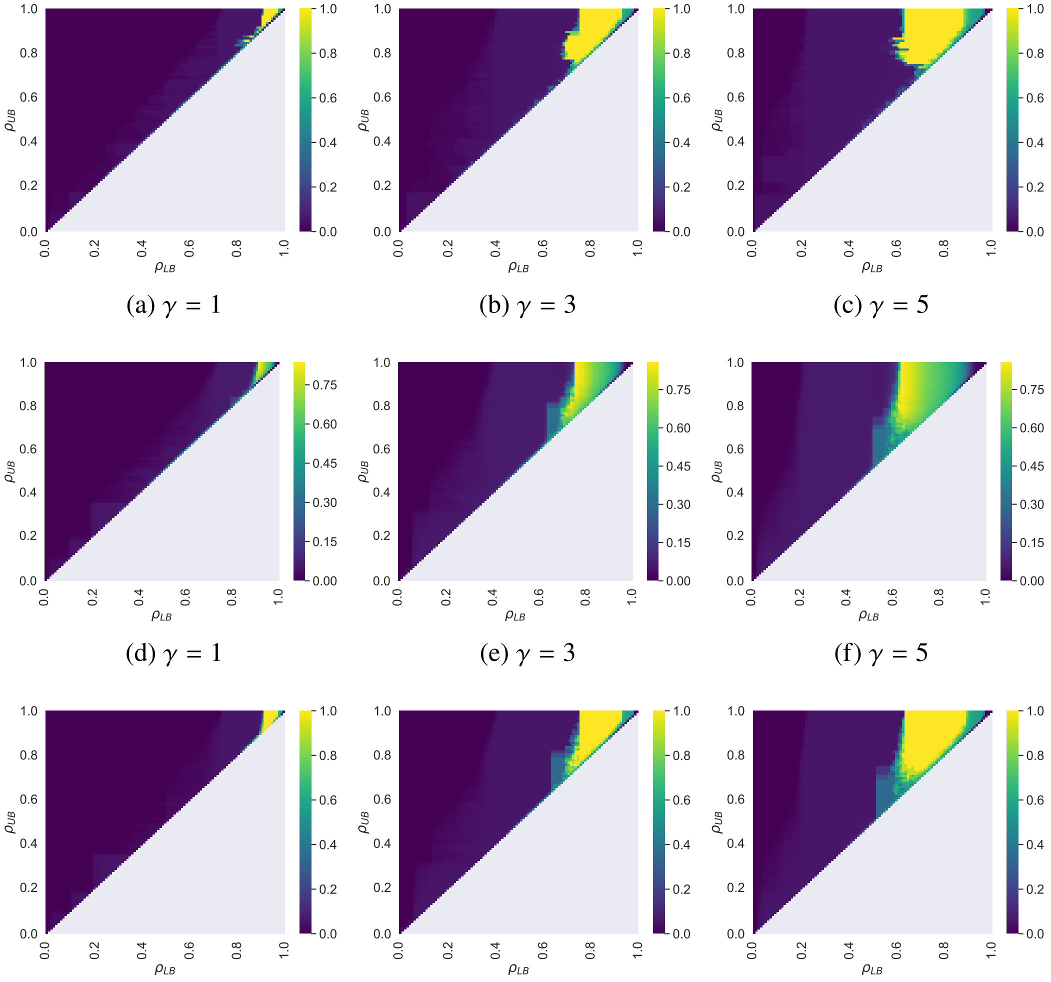


Figure 17. The best and most compact output partitions for the *Target* data set using the (a)-(b) VAT + DDVFA and (c)-(d) DDVFA + Merge ART systems. The ordered pairs correspond to (γ, total number of categories). (a) and (c) correspond to fuzzy ART and are subject to category proliferation, whereas (b) and (d) correspond to DDVFA and represent the same data with fewer categories.

distributed activation and match functions. DDVFA consists of a global ART network whose nodes are local ART modules. The learning mechanism of the former is triggered by the feedback from the latter, thus enabling the system to capture arbitrary data distributions when using appropriate activation/match functions. DDVFA enables both one- and multi-category representations of clusters (i.e., one-to-one and one-to-many mappings of categories to clusters) according to the setting of the upper and lower vigilance parameter values.

Like all agglomerative clustering algorithms, notably fuzzy ART and DVFA, DDVFA is sensitive to input order presentation. This work therefore introduces a compatible Merge ART module that yields improved performance in the online mode where samples arrive in a random order and pre-processing cannot be employed. Experiments were conducted with random and VAT ordered samples. As expected, the latter approach yields better average performance ranks, and thus it is recommended in applications where the offline learning mode is available. Otherwise, for online incremental learning, the usage of a Merge ART module cascaded with DDVFA is recommended, given that the latter showed superior performance and less sensitivity to input presentation order. The VAT + DDVFA and DDVFA + Merge ART systems were found to be statistically equivalent in this paper's



(a) γ = 1 (b) γ = 3 (c) γ = 5

(d) γ = 1 (e) γ = 3 (f) γ = 5

(g) γ = 1 (h) γ = 3 (i) γ = 5

Figure 18. Heat maps corresponding to the average performance (AR) of (a)-(c) VAT + DDVFA, (d)-(f) DDVFA, and (g)-(i) DDVFA + Merge ART, for the *Target* data set when varying parameter γ. More yellow is better, implying a broader range of good parameter values. Sub-figures (a), (d), and (g) correspond to fuzzy ART building blocks, whereas the other portions of the figure correspond to contributions from this paper.

experiments. Naturally, the type of distributed activation/match functions used for the similarity definition is data-dependent; the single-linkage-based ones typically yielded the best and second best average performance rank when cascading Merge ART and pre-processing with VAT, respectively. Conversely, weighted-based activation/match functions yielded the best average performance rank when solely using DDVFA. Naturally, as with other ART algorithms, the dual vigilance parameters must be carefully tuned.

The combination of DDVFA + Merge ART significantly outperformed fuzzy ART, DVFA, and topoART in most of the data sets with randomly presented samples, where a statistical difference was observed. Conversely, when pre-processing with VAT, no statistical difference was observed among the ART networks, except for fuzzy ART. The compactness (i.e., number of categories created) of the networks generated by the multi-prototype ART-based architectures was also compared, and again, no statistical difference was observed between these DDVFA systems and topoART. Furthermore, the clustering performance of these best performing DDVFA systems were compared with single-linkage hierarchical agglomerative clustering (HAC), DBSCAN, k-means and affinity propagation. The results indicated that these DDVFA systems are statistically equivalent to the first three clustering algorithms mentioned, and performed statistically better than affinity propagation. This is noteworthy since DDVFA-based systems are based on incremental learning, whereas all the other non-ART-based algorithms used batch learning. Incremental learning is an important and often essential capability in application. For problems needing incremental learning, DDVFA + Merge ART is demonstrably superior.

Finally, this work discussed DDVFA's computational complexity and hyper parameterization. Particularly, it investigated the effect of the kernel width parameter γ in the behavior of DDVFA. The performance was robust toward this parameter, and with appropriate selection it can potentially increase the compactness (or equivalently, reduce the model complexity) of the DDVFA systems. This memory compression characteristic is consistent with findings from previous related work (distributed ART and ARTMAP systems), which combines power rules and distributed learning. Moreover, it was observed that γ can extend the subspace of dual vigilance parameter combinations that yield effective performance.

APPENDIX

DERIVATION OF THE MATCH FUNCTION IN DDVFA

This section contains the derivation of Eq. (14). Let $M_\gamma = M_j^{ART_i^{(1)}}$ be the activation function of category j of $ART_i^{(1)}$ using γ and M_{γ^*} the activation function of the same category using γ^*. Then, the normalized version of M_γ with respect to M_{γ^*} (M_γ^n) is defined as

$$M_\gamma^n = \left(\max(M_{\gamma^*}) - \min(M_{\gamma^*})\right)\left(\frac{M_\gamma - \min(M_\gamma)}{\max(M_\gamma) - \min(M_\gamma)}\right) + \min(M_{\gamma^*}). \tag{18}$$

The values of $\max(M_{\gamma^*})$ and $\max(M_\gamma)$ are easily obtainable, since any point inside the hyperrectangular category representation would have this value, particularly the weight $\boldsymbol{w} = \boldsymbol{w}_j^{ART_i^{(1)}}$ of category j itself. Furthermore, when using complement coding, $|\boldsymbol{x}| = d$ is a constant. The values $\min(M_{\gamma^*})$ and $\min(M_\gamma)$ must be located at some corner of the d-dimensional unit hyperbox data space $[0, 1]^d$. These values can also be easily calculated for data sets with small dimensionalities. However, as the dimension increases, searching 2^d points quickly becomes impractical. Therefore, since a match function M satisfies $0 \leq M \leq 1$ by definition, a design decision was made to set $\min(M_{\gamma^*}) = \min(M_\gamma) = 0$ in the normalization procedure. Hence,

$$M_\gamma^n = \max(M_{\gamma^*}))\left(\frac{M_\gamma}{\max(M_\gamma)}\right) = \left(\frac{|\boldsymbol{w} \wedge \boldsymbol{w}|}{|\boldsymbol{x}|}\right)^{\gamma^*}\left(\frac{\frac{|\boldsymbol{x}\wedge\boldsymbol{w}|}{|\boldsymbol{x}|}}{\frac{|\boldsymbol{w}\wedge\boldsymbol{w}|}{|\boldsymbol{x}|}}\right)^{\gamma} = \left(\frac{|\boldsymbol{w}|}{|\boldsymbol{x}|}\right)^{\gamma^*}\left(\frac{|\boldsymbol{x} \wedge \boldsymbol{w}|}{c + |\boldsymbol{w}|}\right)^{\gamma}, \tag{19}$$

where the constant c is inserted to safeguard against divisions by zero (since $0 \leq \rho d \leq |\boldsymbol{w}| \leq d$). This parameter implies that $\boldsymbol{w} = \boldsymbol{x}$ no longer yields a match function value equal to 1. By making c equal to the choice parameter α, then Eq. (19) becomes

$$M_\gamma^n = \left(\frac{|\boldsymbol{w}|}{|\boldsymbol{x}|}\right)^{\gamma^*} T_\gamma, \tag{20}$$

where $T_\gamma = T_j^{ART_i^{(1)}}$ is the activation function of category j of $ART_i^{(1)}$ using γ (Eq. (11)). Naturally, if $\gamma^* = 0$ then $M_\gamma^n = T_\gamma$, and for $\alpha \ll |\boldsymbol{w}|$, if $\gamma = \gamma^*$ then $M_\gamma^n \approx M_{\gamma^*}$ (Eq. (13)).

SUPPLEMENTARY DATA

Supplementary material related to this article can be found online at `https://doi.org/10.1016/j.neunet.2019.08.033`.

ACKNOWLEDGMENT

This research was sponsored by the Missouri University of Science and Technology Mary K. Finley Endowment and Intelligent Systems Center; the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance code BEX 13494/13-9; the Army Research Laboratory (ARL) and the Lifelong Learning Machines program from the DARPA / Microsystems Technology Office, and it was accomplished under Cooperative Agreement Number W911NF-18-2-0260. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Amis, G. P., & Carpenter, G. A. (2007). Default ARTMAP 2. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 777–782).

Anagnostopoulos, G. C., & Georgiopoulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 59–64). volume 6.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1221–1226). volume 2.

Araújo, D., Neto, A. D., & Martins, A. (2013a). Information-theoretic clustering: A representative and evolutionary approach. *Expert Systems with Applications*, *40*, 4190–4205.

Araújo, D., Neto, A. D., & Martins, A. (2013b). Representative cross information potential clustering. *Pattern Recognition Letters*, *34*, 2181 – 2191.

Asfour, Y. R., Carpenter, G. A., Grossberg, S., & Lesher, G. W. (1993). Fusion ARTMAP: an adaptive fuzzy network for multi-channel classification. In *Proc. Third International Conference on Industrial Fuzzy Control and Intelligent Systems* (pp. 155–160).

Auda, G., & Kamel, M. (1998). Modular Neural Network Classifiers: A Comparative Study. *J. Intell. Robot. Syst.*, *21*, 117–129.

Auda, G., & Kamel, M. (1999). Modular Neural Networks: A Survey. *International Journal of Neural Systems*, *09*, 129–151.

Auda, G., Kamel, M., & Raafat, H. (1996). Modular Neural Network Architectures for Classification. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1279–1284). volume 2.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Bartfai, G. (1994). Hierarchical clustering with ART neural networks. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 940–944). volume 2.

Bartfai, G. (1996). An ART-based modular architecture for learning hierarchical clusterings. *Neurocomputing*, *13*, 31 – 45.

Bartfai, G., & White, R. (1997a). A fuzzy ART-based modular neuro-fuzzy architecture for learning hierarchical clusterings. In *Proc. 6th International Fuzzy Systems Conference* (pp. 1713–1718). volume 3.

Bartfai, G., & White, R. (1997b). Adaptive Resonance Theory-based Modular Networks for Incremental Learning of Hierarchical Clusterings. *Connection Science*, *9*, 87–112.

Benites, F., & Sapozhnikova, E. (2017). Improving scalability of ART neural networks. *Neurocomputing*, *230*, 219–229.

Bezdek, J. C. (2017). *A Primer on Cluster Analysis: 4 Basic Methods that (usually) Work*. First Edition Design Publishing.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Bouchachia, A., & Mittermeir, R. (2003). A neural cascade architecture for document retrieval. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1915–1920). volume 3.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019a). Dual vigilance fuzzy adaptive resonance theory. *Neural Networks*, *109*, 1–5.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019b). A survey of adaptive resonance theory neural network models for engineering applications. *Neural Networks*, .

Brito da Silva, L. E., M. Melton, N., & Wunsch II, D. C. (2019c). Incremental Cluster Validity Indices for Hard Partitions: Extensions and Comparative Study. *CoRR*, *abs/1902.06711*. arXiv:1902.06711v1 [cs.LG].

Brito da Silva, L. E., & Wunsch II, D. C. (2015). Multi-prototype local density-based hierarchical clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–9).

Brito da Silva, L. E., & Wunsch II, D. C. (2017). Validity Index-based Vigilance Test in Adaptive Resonance Theory Neural Networks. In *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Brito da Silva, L. E., & Wunsch II, D. C. (2018a). A study on exploiting VAT to mitigate ordering effects in Fuzzy ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2351–2358).

Brito da Silva, L. E., & Wunsch II, D. C. (2018b). An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2595–2613.

Carpenter, G. A. (1997). Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks. *Neural Networks*, *10*, 1473 – 1494.

Carpenter, G. A. (2003). Default ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1396–1401). volume 2.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, *3*, 129–152.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*, 565 – 588.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Carpenter, G. A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, *11*, 323 – 336.

Carpenter, G. A., Milenova, B. L., & Noeske, B. W. (1998). Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks*, *11*, 793 – 813.

Chang, H., & Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*, 191–203.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. (2010). Complete Gradient Clustering Algorithm for Features Analysis of X-Ray Images. In E. Pietka, & J. Kawa (Eds.), *Information Technologies in Biomedicine* (pp. 15–24). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, P. P., & Lin, W.-C. (2001). Multi-resolution distributed ART neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. A19–A24). volume Addendum.

Chen, P. P., Lin, W.-C., & Hung, H.-L. (1999). Multi-resolution fuzzy ART neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1973–1978). volume 3.

Davenport, M. P., & Titus, A. H. (2004). Multilevel category structure in the ART-2 network. *IEEE Transactions on Neural Networks*, *15*, 145–158.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., & Zupan, B. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, *14*, 2349–2353.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. (2nd ed.). John Wiley & Sons.

Elnabarawy, I., Wunsch II, D. C., & Abdelbar, A. M. (2016). Biclustering ARTMAP Collaborative Filtering Recommender System. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2986–2991).

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 226–231). AAAI Press.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Fränti, Pasi et al. (2015). Clustering datasets. URL: `http://cs.uef.fi/sipu/datasets/` accessed on May 4, 2017.

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*, 972–976.

Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, *32*, 675–701.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, *11*, 86–92.

Fritzke, B. (1994). Growing cell structures–A self-organizing network for unsupervised and supervised learning. *Neural Networks*, *7*, 1441 – 1460.

Fu, L., & Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, *8*, 1–15.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *19*, 90 – 106.

Gabrys, B., & Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. *IEEE Transactions on Neural Networks*, *11*, 769–783.

Georgiopoulos, M., Fernlund, H., Bebis, G., & Heileman, G. L. (1996). Order of Search in Fuzzy ART and Fuzzy ARTMAP: Effect of the Choice Parameter. *Neural Networks*, *9*, 1541 – 1559.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*, 1–30.

Gomez-Sanchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Lopez-Coronado, J. (2001). Safe-μARTMAP: a new solution for reducing category proliferation in fuzzy ARTMAP. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 1197–1202). volume 2.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An Efficient Clustering Algorithm for Large Databases. In *Proc. ACM SIGMOD Int. Conf. Manag. Data* (pp. 73–84). New York, NY, USA: ACM.

Halkidi, M., & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, *29*, 773 – 786.

Havens, T. C., & Bezdek, J. C. (2012). An Efficient Formulation of the Improved Visual Assessment of Cluster Tendency (iVAT) Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *24*, 813–822.

Huang, Y.-T., Cheng, F.-T., Shih, Y.-H., & Chen, Y.-L. (2014). Advanced ART2 scheme for enhancing metrology-data-quality evaluation. *Journal of the Chinese Institute of Engineers*, *37*, 1064–1079.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Hung, H.-L., Liao, H.-Y. M., Lin, S.-J., Lin, W.-C., & Fan, K.-C. (1996a). Cascade fuzzy ART: a new extensible database for model-based object recognition. *Proc. SPIE*, *2727*, 187–198.

Hung, H.-L., Liao, H.-Y. M., Sze, C.-J., Lin, S.-J., Lin, W.-C., & Fan, K.-C. (1996b). CFART: A Multi-Resolutional Adaptive Resonance System. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 1312–1317). volume 2.

Ibrahim, O. A., Keller, J. M., & Bezdek, J. C. (2018a). Analysis of streaming clustering using an incremental validity index. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8).

Ibrahim, O. A., Wang, Y., & Keller, J. M. (2018b). Analysis of incremental cluster validity for big data applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *26*, 47–62.

Ilc, N. (2013). Datasets package. URL: `https://www.researchgate.net/publication/239525861_Datasets_package` accessed on Sep 11, 2017.

Ilc, N., & Dobnikar, A. (2011). Gravitational Clustering of the Self-Organizing Map. In A. Dobnikar, U. Lotrič, & B. Šter (Eds.), *Adaptive and Natural Computing Algorithms* (pp. 11–20). Berlin, Heidelberg: Springer volume 6594.

Ilc, N., & Dobnikar, A. (2012). Generation of a clustering ensemble based on a gravitational self-organising map. *Neurocomputing*, *96*, 47–56.

Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics - Theory and Methods*, *9*, 571–595.

Isawa, H., Matsushita, H., & Nishio, Y. (2008a). Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3595–3600).

Isawa, H., Matsushita, H., & Nishio, Y. (2008b). Improved Fuzzy Adaptive Resonance Theory Combining Overlapped Category in Consideration of Connections. In *IEEE Workshop on Nonlinear Circuit Networks (NCN)* (pp. 8–11).

Isawa, H., Matsushita, H., & Nishio, Y. (2009). Fuzzy ART Combining Overlapped Categories Using Variable Vigilance Parameters. In *Proc. International Workshop on Nonlinear Circuits and Signal Processing (NCSP)* (pp. 661–664).

Isawa, H., Tomita, M., Matsushita, H., & Nishio, Y. (2007). Fuzzy Adaptive Resonance Theory with Group Learning and its Applications. In *Proc. International Symposium on Nonlinear Theory and its Applications (NOLTA)* (pp. 292–295).

Ishihara, S., Ishihara, K., Nagamachi, M., & Matsubara, Y. (1995). arboART: ART based hierarchical clustering and its application to questionnaire data analysis. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 532–537). volume 1.

Izquierdo, J. M. C., Dimitriadis, Y. A., Sánchez, E. G., & Coronado, J. L. (2001). Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, *14*, 407 – 425.

Jain, A. K., & Law, M. H. C. (2005). Data Clustering: A User's Dilemma. In S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds.), *Pattern Recognition and Machine Intelligence* (pp. 1–10). Berlin, Heidelberg: Springer Berlin Heidelberg volume 3776 of *Lecture Notes in Computer Science*.

Kim, B., Ban, S.-W., & Lee, M. (2011). Growing fuzzy topology adaptive resonance theory models with a push-pull learning algorithm. *Neurocomputing*, *74*, 646 – 655.

Kim, S., & Wunsch II, D. C. (2011). A GPU based Parallel Hierarchical Fuzzy ART clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2778–2782).

Kondadadi, R., & Kozma, R. (2002). A modified fuzzy ART for soft document clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2545–2549). volume 3.

Lam, D., Wei, M., & Wunsch II, D. C. (2015). Clustering Data of Mixed Categorical and Numerical Type With Unsupervised Feature Learning. *IEEE Access*, *3*, 1605–1613.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1997). Multiple categorization using fuzzy ART. In *Proceedings of International Conference on Neural Networks (ICNN'97)* (pp. 1983–1988). volume 3.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1999). Generalization, discrimination, and multiple categorization using adaptive resonance theory. *IEEE Transactions on Neural Networks*, *10*, 757–767.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). University of California Press volume 1.

Massey, L. (2009). Discovery of hierarchical thematic structure in text collections with adaptive resonance theory. *Neural Computing and Applications*, *18*, 261–273.

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2016). Adaptive scaling of cluster boundaries for large-scale social media data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 2656–2669.

Meng, L., Tan, A. H., & Xu, D. (2014). Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 2293–2306.

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., & Bailey, J. (2019). Online cluster validity indices for performance monitoring of streaming data clustering. *International Journal of Intelligent Systems*, *34*, 541–563.

Mulder, S. A., & Wunsch II, D. C. (2003). Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, *16*, 827–832.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Ph.D. thesis Princeton University.

Nooralishahi, P., Loo, C. K., & Seera, M. (2018). Semi-supervised topo-Bayesian ARTMAP for noisy data. *Applied Soft Computing*, *62*, 134 – 147.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Sapozhnikova, E. P. (2009). ART-Based Neural Networks for Multi-label Classification. In N. M. Adams, C. Robardet, A. Siebes, & J.-F. Boulicaut (Eds.), *Advances in Intelligent Data Analysis VIII* (pp. 167–177). Berlin, Heidelberg: Springer volume 5772.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.*, *42*, 19:1–19:21.

Seiffertt, J., & Wunsch II, D. C. (2010). *Unified Computational Intelligence for Complex Systems* volume 6 of *Evolutionary Learning and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Shang, F., Jiao, L., Shi, J., Wang, F., & Gong, M. (2012). Fast affinity propagation clustering: A multilevel approach. *Pattern Recognition*, *45*, 474 – 486.

Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, *16*, 30–34.

Simpson, P. K. (1992). Fuzzy min-max neural networks. i. classification. *IEEE Transactions on Neural Networks*, *3*, 776–786.

Simpson, P. K. (1993). Fuzzy min-max neural networks - part 2: Clustering. *IEEE Transactions on Fuzzy Systems*, *1*, 32–.

Su, M.-C., & Liu, T.-K. (2001). Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, *37*, 165 – 175.

Su, M.-C., & Liu, Y.-C. (2002). A hierarchical approach to ART-like clustering algorithm. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 788–793). volume 1.

Su, M.-C., & Liu, Y.-C. (2005). A new approach to clustering data with arbitrary shapes. *Pattern Recognition*, *38*, 1887 – 1901.

Swope, J. A. (2012). ARTdECOS, adaptive evolving connectionist model and application to heart rate variability. *Evolving Systems*, *3*, 95–109.

Taşdemir, K., & Merényi, E. (2009). Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *20*, 549–562.

Taşdemir, K., & Merényi, E. (2011). A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*, 1039–1053.

Tan, A.-H. (1995). Adaptive Resonance Associative Map. *Neural Networks*, *8*, 437 – 446.

Tan, A.-H. (2004). FALCON: a fusion architecture for learning, cognition, and navigation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3297–3302). volume 4.

Tan, A.-H. (2006). Self-organizing Neural Architecture for Reinforcement Learning. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 470–475). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence Through Interaction: Towards a Unified Theory for Learning. In D. Liu, S. Fei, Z.-G. Hou, H. Zhang, & C. Sun (Eds.), *Advances in Neural Networks - ISNN 2007* (pp. 1094–1103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Lu, N., & Xiao, D. (2008). Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning With Delayed Evaluative Feedback. *IEEE Transactions on Neural Networks*, *19*, 230–244.

Tscherepanow, M. (2010). TopoART: A Topology Learning Hierarchical ART Network. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 157–167). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2011). An Extended TopoART Network for the Stable On-line Learning of Regression Functions. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *International Conference on Neural Information Processing (ICONIP)* (pp. 562–571). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In P. Perner (Ed.), *Proc. Industrial Conference on Data Mining (ICDM)* (pp. 22–34). ibai-publishing.

Tscherepanow, M., Kortkamp, M., & Kammer, M. (2011). A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*, *24*, 906 – 916.

Tscherepanow, M., Kühnel, S., & Riechers, S. (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 24–29).

Tscherepanow, M., & Riechers, S. (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 18–23).

Tyree, E. W., & Long, J. A. (1999). The use of linked line segments for cluster representation and data reduction. *Pattern Recognition Letters*, *20*, 21 – 29.

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Ultsch, A., & Siemon, H. P. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. International Neural Networks Conference (INNC)* (pp. 305–308).

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1273–1280.

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*, 1628–1644.

Švaco, M., Jerbić, B., & Šuligoj, F. (2014). ARTgrid: A Two-level Learning Architecture Based on Adaptive Resonance Theory. *Advances in Artificial Neural Systems*, *2014*, 1–9.

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*, 80–83.

Williamson, J. R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, *9*, 881 – 897.

Wunsch II, D. C. (1991). *An optoelectronic learning machine: invention, experimentation, analysis of first hardware implementation of the ART 1 neural network*. Ph.D. thesis University of Washington.

Wunsch II, D. C. (2009). ART properties of interest in engineering applications. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 3380–3383).

Wunsch II, D. C., Caudell, T. P., Capps, C. D., Marks, R. J., & Falk, R. A. (1993). An optoelectronic implementation of the adaptive resonance neural network. *IEEE Transactions on Neural Networks*, *4*, 673–684.

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2010). Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, *3*, 120–154.

Xu, R., & Wunsch II, D. C. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, *24*, 709–716.

Yavaş, M., & Alpaslan, F. N. (2012). Hierarchical behavior categorization using correlation based adaptive resonance theory. *Neurocomputing*, *77*, 71 – 81.

Yousuf, A., & Murphey, Y. L. (2010). A Supervised Fuzzy Adaptive Resonance Theory with Distributed Weight Update. In L. Zhang, B.-L. Lu, & J. Kwok (Eds.), *Advances in Neural Networks - ISNN 2010* (pp. 430–435). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zahn, C. T. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, *C-20*, 68–86.

Zhang, L., Wang, G., & Wang, W. (2006). A New Fuzzy ART Neural Network Based on Dual Competition and Resonance Technique. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 792–797). Berlin, Heidelberg: Springer volume 3971.

VI. AN INFORMATION-THEORETIC-CLUSTER VISUALIZATION FOR SELF-ORGANIZING MAPS

Leonardo Enzo Brito da Silva[1,2] and Donald C. Wunsch II[2]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering,

Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

Improved data visualization will be a significant tool to enhance cluster analysis. In this work, an information-theoretic-based method for cluster visualization using self-organizing maps (SOM) is presented. The information theoretic visualization (IT-vis) has the same structure as the unified distance matrix, but instead of depicting Euclidean distances between adjacent neurons, it displays the similarity between the distributions associated with adjacent neurons. Each SOM neuron has an associated subset of the data set whose cardinality controls the granularity of the IT-vis and with which the first and second order statistics are computed and used to estimate their probability density functions. These are used to calculate the similarity measure, based on Renyi's quadratic cross entropy and cross information potential (CIP). The introduced visualizations combine the low computational cost and kernel estimation properties of the representative CIP and the data structure representation of a single-linkage based grouping algorithm to generate an enhanced SOM-based visualization. The visual quality of the IT-vis is assessed by comparing it to other visualization methods for several real world and synthetic benchmark data sets. (Thus, this paper also contains a significant literature survey). The experiments demonstrate the IT-vis cluster revealing capabilities, in which cluster boundaries are sharply captured.

Additionally, the information theoretic visualizations are used to perform clustering of the SOM. Compared to other methods, IT-vis of large SOMs yielded the best results in this study, for which the quality of the final partitions was evaluated using external validity indices.

Keywords: Self-organizing feature maps, Data visualization, Information theory, Entropy, Review, Survey, Clustering.

1. INTRODUCTION

Data visualization methods are useful tools that provide additional information to support exploratory data analysis. A widely used visualization method is the self-organizing map (SOM) (Kohonen, 1982). Each position in the SOM lattice is associated with a weight in the data space; therefore, a non-linear dimensionality reduction is achieved when mapping from the input space (data space) to the output space (SOM lattice). This property is exploited by visualization techniques which aim to infer data characteristics from the SOM neurons. In image-based visualization methods, gray-level images display some feature of the data captured by the neurons, such as Euclidean distance and density distribution of data samples; these characteristics are measured in the data space and imprinted in the output space; this category of visualizations includes, for instance, the unified distance matrix (U-matrix) (Ultsch, 1993; Ultsch & Siemon, 1990), gradient of components matrix (Costa & Yin, 2010), boundary-matrix (Manukyan et al., 2012), gravitational algorithm enhanced U-matrix (Brito da Silva & Costa, 2013a), data histograms (Vesanto, 1999; Zhang & Li, 1993), smoothed data histograms (Pampalk et al., 2002) and P-matrix (Ultsch, 2003). In graph-based visualization methods, the weights of the edges (connections) between vertices (neurons) in the SOM lattice depict information such as local distances or local-density distributions; examples of this category of visualization include the Cluster Connections (Merkl & Rauber, 1997), CONNvis (Taşdemir & Merényi, 2009) (which also displays topology information), DISTvis and CONNDISTvis (Taşdemir, 2010). Projection-based visualization methods are mainly characterized by their distance-preserving property in addition to the topology preservation of standard SOM; examples of this category of visualization include Visualization-induced SOM (Yin, 2001, 2002a,b, 2008), AC-ViSOM (Tapan & Siong, 2008), Probabilistic Regularized SOM (Wu & Chow, 2005), Polar SOM (Xu et al., 2010) and Probabilistic Polar SOM (Xu et al., 2011).

Some other data visualization approaches include linear and non-linear dimensionality reduction methods. The classical projection method examples are linear principal component analysis (Xu & Wunsch II, 2009, Sec. 9.2.1) and multidimensional scaling (Xu & Wunsch II, 2009, Sec. 9.3.2) (e.g., Sammon's mapping (Sammon, 1969)). The first projects data onto the subspace that is spanned by a selection of the top eigenvectors of the data covariance matrix; the latter aim to preserve inter-point relations in the original high dimensional space and the lower dimensional projected subspace. An alternative visualization method is the Visual Assessment of (cluster) Tendency (Bezdek & Hathaway, 2002) and its variants, such as (Bezdek et al., 2007; Huband et al., 2004; Wang et al., 2008, 2010), which re-order the dissimilarity matrix of the data such that similar samples are placed close to each other. This rearranged matrix is displayed as a gray level image, where dark blocks reveal cluster tendency.

Many data visualization techniques are eventually used to perform clustering or clustering related tasks. Clustering is usually performed by maximizing the similarity within groups and minimizing the similarity between groups, for which many algorithms have been presented (Xu & Wunsch II, 2005; Xu & Wunsch II, 2009; Xu & Wunsch II, 2010). In the case of SOM, one approach is the two-step prototype-based clustering framework (Vesanto & Alhoniemi, 2000), which consists of a vector quantization step followed by clustering of the prototypes. The data samples are labeled according to the structure found in the prototypes. For instance, digital image processing techniques were applied to the U-matrix visualization to enable clustering of the SOM using mathematical morphology operators and the watershed transform (Costa & Netto, 1999); this approach was extended to recursively generate a hierarchical tree of SOM networks representing different levels of data granularity (Costa & Netto, 2001). Alternatively, visualization techniques may be used to extract features and perform clustering (Brito da Silva & Costa, 2013b,d). CONNvis visualization is based on a local-density-based similarity matrix CONN, which was used to devise a relative validity index (Conn_Index) (Taşdemir & Merényi, 2007, 2011) and for prototype-based clustering algorithms: CONN linkage hierarchical clustering (Taşdemir et al., 2011) and vector quantization approximate spectral clustering (Taşdemir, 2012). Polar SOM has also been used to perform grid clustering (Xu et al., 2015). The appeal of these approaches for both visualization and prototype-based clustering is the ability to capture the clusters' boundaries in high-dimensional complex data structures with arbitrary geometries, density distributions, sizes and levels of overlap. This versatility of the SOM makes it an effective tool for data exploration and exploitation, thereby being extensively used for clustering tasks.

In cluster analysis, one of the ways to measure the separation between clusters is to use divergences and non-linear distances. For instance, in multi-representative clustering approaches, such as the ones reviewed in (Martins et al., 2004b), a competitive neural network is the vector quantization algorithm used to generate the clusters' representatives, which are then connected according to their Mahalanobis distance (Martins et al., 2003) or Kullback-Leibler divergence (Martins et al., 2004a) with regards to a user defined threshold value. Additionally, stability regions (plateaus) can be identified.

Recently, information theoretic learning (ITL) (Principe, 2010) has emerged. In the scope of ITL, information theoretic clustering (ITC) was presented (Gokcay & Principe, 2000, 2002) based on the insight that data samples interact with each other. This interaction is similar to potential fields in physics. ITC is based on Renyi's quadratic entropy estimator, and it assesses the quality of partitions via a Clustering Evaluation Function (CEF). The

CEF is a non-linear weighted function to measure the distance between distributions. In this non-parametric approach to clustering, the function evaluates the similarity between probability density functions (pdfs) that have been calculated from the data samples.

The CEF is based on Renyi's cross information potential (CIP), which in turn originated from Renyi's entropy that was computed using the Parzen-window method (Duda et al., 2000) with a Gaussian kernel as the pdf estimation method. The objective consists of minimizing the CEF, or conversely, maximizing the entropy between distributions using this cost function. The CEF performs calculations between each pair of samples belonging to different clusters. The clustering task is interpreted as a permutation problem in which the goal is to minimize the CEF. Different strategies have been used, such as exhaustive enumeration, a modified k-change algorithm, and simulated annealing. The performance was judged comparable to simple supervised classifiers, such as a single perceptron, but still lower than a multilayer perceptron (Gokcay & Principe, 2002).

ITC does not impose a structure to the data. Nonetheless, one challenge depends on setting appropriate values to the kernel parameter for the pdf estimation, i.e. the covariance of the multidimensional Gaussian kernel. This user-defined parameter controls the interaction between samples. Its value is problem-specific and must be carefully chosen. Although the kernel can be adapted to the distribution of a set of nearest samples (Gokcay & Principe, 2002), this is a much more general challenge belonging to the realm of kernel methods. Furthermore, it is not practical to perform pairwise calculations when dealing with large data sets, as this makes the approach very computationally intensive.

Therefore, the representative cross information potential (rCIP) was introduced in (Araújo et al., 2013b). In this modified version of CIP, prototypes generated by a vector quantization method (e.g., *k*-means (MacQueen, 1967)) create Voronoi cells whose data points are used to infer higher order statistical information, thus reducing the computational cost (handling prototypes instead of the entire data set) and removing the requirement of defining the covariance matrix for the pdf estimation. The performance of this approach is sensitive to the number of prototypes.

The CIP/rCIP is a descriptor with good discriminant power; thus, several CEFs were based on it. These approaches aim to minimize the CEF through optimization methods such as simulated annealing (Gokcay & Principe, 2000), a variant of the *k*-change algorithm (Gokcay & Principe, 2002), and genetic algorithms (Araújo et al., 2013a). Other approaches include hierarchical clustering and linear programming (Araújo et al., 2013b).

ITL was successfully used for digital image processing as a contrast enhancement method and also for segmentation (Araújo et al., 2013a,b; Gokcay & Principe, 2002; Rao et al., 2009). Additionally, ITL was embedded in the SOM learning algorithm (Chalasani & Principe, 2010, 2015), in which Correntropy Induced Metric (Liu et al., 2007) was used to improve the magnification factor in SOMs and generate an enhanced U-matrix. Another information theoretic approach to SOM is based on minimizing the free energy quantity (related to mutual information). The inclusion of the similarity interaction feature in the neighborhood kernel of this type of SOM has been used to improve the visual assessment of clusters through the U-matrix as well as quantization and topology errors (Kamimura, 2013).

Defining suitable visualization methods are of great interest as they help one to formulate a hypothesis for the data distribution (Vesanto, 1999). Therefore, this work presents an image-based SOM visualization which is displayed similarly to the U-matrix. The shade is imprinted as a result of combining Renyi's cross-entropy and cross-information potential (both CIP and rCIP) (Araújo et al., 2013a,b; Gokcay & Principe, 2002; Rao et al., 2009) and a single-linkage-based grouping algorithm (Gokcay & Principe, 2002) for parameter estimation. The presented visualizations are compared to other visualization methods, both for the visual assessment of clusters and clustering of the SOM.

The remainder of this paper is organized as follows: Section 2 reviews the main concepts related to SOM networks and visualization methods as well as ITC; Section 3 describes the presented approach; Section 4 details the experimental set-up; Section 5 showcases visualizations for some data sets; Section 6 presents and discusses the results obtained when clustering the SOM using the presented visualization; finally, Section 7 draws relevant conclusions.

2. BACKGROUND AND RELATED WORK

2.1. SELF-ORGANIZING MAPS

The self-organizing map (SOM) (Kohonen, 1982) is widely used for data visualization and clustering. Its neurons are arranged in a lattice (output space), in an organized manner, according to a given topology (e.g., hexagonal or rectangular). During the training process, the closest neuron $\boldsymbol{w}$ to each data sample $\boldsymbol{x}$ in the input space, is determined and updated according to:

$$\boldsymbol{w}_j(t+1) = \frac{\sum_{i=1}^{N} h_{j,bmu}(t)\boldsymbol{x}_i}{\sum_{i=1}^{N} h_{j,bmu}(t)}, \tag{1}$$

which corresponds to the batch training algorithm. The $\boldsymbol{w}_j$ is the weight of neuron j, $\boldsymbol{x}_i$ is the i^{th} sample presented, N is the cardinality of the data set, and $h_{j,bmu}(t)$ is the monotonically decreasing neighborhood function centered on the best matching unit (BMU), such that a limited number of neighboring neurons to the BMU also participate in the learning process. The neighborhood kernel is usually defined as a Gaussian function of the distance between the neurons in the output space,

$$h_{j,bmu}(t) = exp\left(-\frac{||\boldsymbol{r}_{bmu} - \boldsymbol{r}_j||^2}{2\sigma^2(t)}\right), \tag{2}$$

where $||\cdot||$ is the Euclidean distance; $\boldsymbol{r}_j$ and $\boldsymbol{r}_{bmu}$ are the locations of neuron j and the BMU in the lattice, and σ is the monotonically decreasing neighborhood radius.

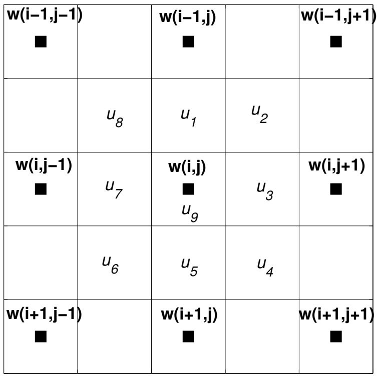
2.2. SOM-BASED VISUALIZATION METHODS

2.2.1. Image-Based Visualizations. the unified distance matrix (U-matrix) (Ultsch, 1993; Ultsch & Siemon, 1990) is a heat map of the Euclidean distances between neighboring neurons in the SOM lattice. It is usually displayed as a gray-level image in which bright and dark pixels correspond to evidence of clusters (valleys) and boundaries (hills), respectively. Figure 1 illustrates the positions (i, j) of the neurons $(\boldsymbol{w})$ in the lattice and their positions in the U-matrix array $U(u)$.

The values of the U-matrix at positions $\{u_1, u_3, u_5, u_7\}$ and $\{u_2, u_4, u_6, u_8\}$ are analogous to (3) and (4), respectively:

$$U(u_1) = ||\boldsymbol{w}(i, j) - \boldsymbol{w}(i - 1, j)||, \tag{3}$$

$$U(u_2) = \frac{||\boldsymbol{w}(i, j) - \boldsymbol{w}(i - 1, j + 1)|| + ||\boldsymbol{w}(i - 1, j) - \boldsymbol{w}(i, j + 1)||}{2\sqrt{2}}. \tag{4}$$



w(i−1,j−1) ■		w(i−1,j) ■		w(i−1,j+1) ■
	u_8	u_1	u_2	
w(i,j−1) ■	u_7	w(i,j) ■ u_9	u_3	w(i,j+1) ■
	u_6	u_5	u_4	
w(i+1,j−1) ■		w(i+1,j) ■		w(i+1,j+1) ■

Figure 1. Positions of the SOM neurons $\boldsymbol{w}$ in the U-matrix (u).

The values for the positions in the U-matrix that are directly associated with the neurons are based on the distances to their neighbors. For instance, for the neuron $\boldsymbol{w}(i, j)$ at position u_9,

$$U(u_9) = g\left(U(u_1), U(u_3), U(u_5), U(u_7)\right), \tag{5}$$

where $g(\cdot)$ can be the mean or median functions. Some drawbacks of the U-matrix include the existence of several local minima, noise, and boundaries that are not well-defined (Costa & Netto, 1999, 2001).

The boundary-matrix (Manukyan et al., 2012) is a visualization scheme that computes inter-neuron distances similarly to the U-matrix and was devised to display sharpened cluster boundaries in sparsely-matched SOMs (low sample to neuron ratios). The boundary-matrix is generated after a cluster reinforcement phase (post-processing strategy) performed over the SOM neurons. The Euclidean distances are displayed using grid-lines whose thicknesses are proportional to the boundary-matrix and overlaid on the SOM component planes (Vesanto, 1999).

Regarding density features of the data, the smoothed data histograms (Pampalk et al., 2002) is a visualization method that aims to estimate the data pdf by allowing more than one BMU for each data sample, which is the case in data histograms (Vesanto, 1999; Zhang & Li, 1993) defined by the number of samples inside the Voronoi cells pertaining to each SOM neuron. The number of BMUs considered for each data sample is controlled by a user-defined smoothing parameter: the lower bound corresponds to the traditional data histogram, whereas the upper bound depicts only one big cluster.

2.2.2. Graph-Based Visualizations. the CONNvis (Taşdemir & Merényi, 2009) is a visualization technique for SOMs whose main feature is the depiction of data topology information. It is a weighted version of the induced Delaunay triangulation graph (Martinetz & Schulten, 1994), in which the weights of the edges encode the local data density distribution between adjacent neurons. The weights of the graph are stored in a connectivity/similarity matrix (CONN) (Taşdemir & Merényi, 2005), where each element

(i, j) consists of the number of samples $\boldsymbol{x}$ in a data set X for which neurons i and j are the first and the second BMUs and vice-versa, thus capturing portions of their receptive fields (RF) (Taşdemir et al., 2011):

$$CONN(i, j) = |RF|_{ij} + |RF|_{ji}, \tag{6}$$

$$RF_{ij} = \left\{\boldsymbol{x}_k \in RF_i \mid \|\boldsymbol{x}_k - \boldsymbol{w}_j\| \leq \|\boldsymbol{x}_k - \boldsymbol{w}_l\| \ \forall l \neq i\right\}, \tag{7}$$

$$RF_i = \left\{\boldsymbol{x}_k \in X \mid \|\boldsymbol{x}_k - \boldsymbol{w}_i\| \leq \|\boldsymbol{x}_k - \boldsymbol{w}_j\| \ \forall j\right\}. \tag{8}$$

The CONNvis is rendered on the SOM lattice (or in the data space for low-dimensional data sets), where neurons that have non-null entries in the CONN matrix are connected according to specific color and line width encoding. By analyzing the CONNvis, one is able to evaluate topology violations regarding a trained SOM. For both clustering and visualization, CONNvis requires that the number of neurons must be much smaller than the number of samples (Brito da Silva & Costa, 2013c; Taşdemir, 2012; Taşdemir et al., 2011).

Cluster Connections (Merkl & Rauber, 1997) and DISTvis (Taşdemir, 2010) are graph-based SOM visualizations that depict local distances. Cluster Connections displays the connections of neighboring neurons in the output grid proportional to their weights' similarity, whereas DISTvis is a rendering of the graph DIST, whose edges' weights encode Euclidean distances on the SOM grid, allowing connections between any neurons. Thresholds and gray-level scales for intensity coloring are used to enhance the visual representation of clusters in both methods. Merging local distance and local density information in a single graph-based visualization is accomplished by hybridizing CONNvis with DISTvis: CONNDISTvis (Taşdemir, 2010). Analyzing these graphs facilitates the understanding of the correlation between density and distance information.

2.2.3. Projection-Based Visualizations. the standard SOM allows the distortion of the clusters' shapes as it does not preserve inter-neuron distances, thus the data distribution is not faithfully represented. Visualization-induced SOM (ViSOM) (Yin, 2001, 2002a,b,

2008) was introduced to address this challenge and also to provide a low computational cost alternative to data projection methods such as multidimensional scaling. As opposed to the latter, ViSOM provides a mapping function to allow the projection of new data samples on the trained manifold without re-performing all the calculations using the whole data set. ViSOM has a distance-preserving property in addition to the topology preservation present in standard SOM. ViSOM is a uniform quantizer, whereas SOM is a density-based quantizer: its neurons are uniformly distributed over the data manifold. This is accomplished by regularizing the lateral forces between neurons through a resolution parameter, which allows for contractions and expansions of the net so as to preserve the inter-neuron distances in the input and output spaces. This parameter is set according to the maximum variance or scope of the data.

Probabilistic Regularized SOM (PRSOM) (Wu & Chow, 2005) introduces a cost function for the manifold learning process (in contrast to ViSOM), which consists of a soft vector quantization error term and a regularized multidimensional scaling term. The latter constrains the inter-neuron distances in order to make the input resemble the output as faithfully as possible. PRSOM uses a soft assignment (as opposed to the hard assignment of SOM and ViSOM), in which each neuron has a weighted probabilistic assignment that take into account neighboring neurons. Hence, in order to improve data visualization, the PRSOM takes advantage of SOM's low computational cost as well as the preservation of inter-neuron distances through multidimensional scaling. In the context of manifold learning, SOM, ViSOM and PRSOM are discrete approximations to principal curves/surfaces (Hastie & Stuetzle, 1989; LeBlanc & Tibshirani, 1994). Additionally, ViSOM is considered to be a special case of PRSOM. PRSOM has an associated image-based visualization: a coloring scheme is used to render the accumulated probability matrix, which displays clusters and empty regions.

The AC-ViSOM (Tapan & Siong, 2008) is a hybridization of ViSOM and the modified Adaptive Coordinates (modified-AC) (Tapan & Teh, 2007; Teh & Sarwar, 2008), which aims to automate the selection of the regularization parameter, improve the ViSOM resource utilization (quantity of dead neurons) and reduce the mean squared error. The modified-AC is a variation of the Adaptive Coordinates (Merkl & Rauber, 1997), which is a method that mimics, in the output space, the displacement of the SOM weights in the input space. In this manner, information obtained during the training process is used for visualization. The drawback is that the organization of the neurons in the fixed grid is lost. The modified-AC uses as the adaptation factor the difference between the normalized distance in the input and output spaces. The selection of the regularization parameter in AC-ViSOM is based on the ratio between the maximum distances in the input and output spaces of the SOM (i.e., the ratio between normalization factors).

In the Polar SOM (PolSOM) (Xu et al., 2010), the output space is defined in a polar coordinate system. The neurons and data positions in the output space are encoded using radii and angles, to express the importance of each feature and the features themselves, respectively. The neurons are distributed in the polar plane in the intersections of rings and radial axes. The data samples have their associated positions in the output space adapted throughout the learning process in order to be close to their respective BMU. When projecting data, this representation emphasizes the differences among the clusters by displaying a correlation between features (angles) and feature values (radii). The PolSOM preserves topology and inter-neuron distance, and it has the same ViSOM advantage: an explicit mapping function, thus no re-computation is needed for new data samples. The Probabilistic Polar SOM (Xu et al., 2011) is a variant of Polar SOM that uses a weighted probabilistic assignment (soft assignment) similar to PRSOM, thus aiming to minimize a soft vector quantization error.

2.3. INFORMATION THEORETIC CLUSTERING

Let a data set be $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, where each sample $\boldsymbol{x}_i \in \mathbb{R}^d$. In order to make use of information theoretic clustering (ITC) (Gokcay & Principe, 2002), first it is necessary to estimate the data pdf via the Parzen-window method, for which the Gaussian kernel is given by,

$$f(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} G(\boldsymbol{x};\boldsymbol{x}_i,\Sigma), \tag{9}$$

where $G(\boldsymbol{x};\boldsymbol{\mu},\Sigma)$ is the multidimensional Gaussian function parametrized by the mean ($\boldsymbol{\mu}$) and the covariance matrix (Σ):

$$G(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \tag{10}$$

Second, it is necessary to work with the data entropy. The concept of entropy is related to the measure of randomness of a random variable, and ultimately, the amount of information present. Thus, in order to measure the uncertainty of X, Renyi's Entropy (Rényi, 1961) can be computed using (9) as,

$$H(X;\alpha) = \frac{1}{1-\alpha}\log\left(\int_{-\infty}^{\infty} f^{\alpha}(\boldsymbol{x})d\boldsymbol{x}\right), \tag{11}$$

where α is an order parameter, and $\alpha > 0$ and $\alpha \neq 1$.

Using the Parzen-window estimator, Renyi's quadratic entropy ($\alpha = 2$) can be calculated from the data samples as (Gokcay & Principe, 2000, 2002; Rao et al., 2009),

$$H(X;2) = -\log\left(\int_{-\infty}^{\infty} f^{2}(\boldsymbol{x})d\boldsymbol{x}\right) = -\log\left(V(X)\right), \tag{12}$$

where $V(\boldsymbol{x}) \geq 0$ is the information potential and is given by:

$$V(X) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G(\boldsymbol{x}_i; \boldsymbol{x}_j, \Sigma_{i,j}). \tag{13}$$

When using a constant spherically symmetric Gaussian kernel in (9) for all samples ($\Sigma = \sigma^2 I$), then $\Sigma_{i,j} = 2\sigma^2 I, \forall(i, j)$, due to the fact that the integral of the product of two Gaussians is equal to another Gaussian whose variance is the sum of the first two.

Analogously, the concepts embedded in (11) through (13) can be extended to consider two random variables X and Y with distributions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$, thus yielding Renyi's cross-entropy (Rao et al., 2009),

$$H(X, Y; 2) = -\log\left(\int_{-\infty}^{\infty} f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x}\right) = -\log\left(V(X, Y)\right), \tag{14}$$

$$V(X, Y) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} G(\boldsymbol{x}_i; \boldsymbol{x}_j, \Sigma_{i,j}), \tag{15}$$

where $V(X, Y)$ is the cross information potential (CIP), $\boldsymbol{x}_i \in f(\boldsymbol{x})$ and $\boldsymbol{x}_j \in g(\boldsymbol{x})$. Again, $\Sigma_{i,j} = \Sigma_i + \Sigma_j$ (Araújo et al., 2013a,b).

The problem of minimizing the CIP between clusters (minimizing the mutual information) is the same as maximizing the entropy. Of course, when optimizing the CEF, minimizing $V(X, Y)$ is equivalent to maximizing $H(X, Y)$. The minimum value of the CEF is not zero when comparing two equal distributions, as it is in the case of other divergence measurements such as the Kullback-Leibler, Bhattacharya and so on. However, it does have the same general behavior (Gokcay & Principe, 2002).

The main challenge of CIP is the need for pairwise calculations for all of the samples in the data set as well as for estimating the covariance matrix Σ. In order to overcome these, (Araújo et al., 2013a,b) presented the representative cross information potential (rCIP). The calculations of the rCIP are made only with prototypes of the data, and the covariance matrix is computed using the data falling into the Voronoi cells associated with them. In this work, both approaches are followed to compute the cross information potential; notwithstanding, the covariance matrix Σ is always estimated using $MinPts$ samples, which are points that represent the k-nearest samples to a given neuron.

A detailed discussion regarding the influence of the translation (mean) and rotation (covariance) of distributions with respect to the behavior of CIP and rCIP, while also varying the value of the kernel width, can be found in (Araújo et al., 2013a). Briefly, the largest value is obtained when distributions completely overlap, whereas translations of the mean diminishes it. The best case to observe the separation of distributions corresponds to parallel clusters and the worst, to aligned clusters; isotropic clusters lie in-between.

3. INFORMATION-THEORETIC VISUALIZATION

This section presents an improvement to the previously mentioned image-based visualization techniques in order to enhance data analysis. This is achieved by imprinting, in a heat map, SOM neurons' similarities based information-theoretic measures: cluster tendency is revealed by highlighting the clusters' boundaries. Consider a SOM neural network trained with a given data set. Each neuron i is associated with a subset $\mathcal{H}_i$ of the data set with at least $MinPts$ data points. First, these subsets $\mathcal{H}_i$ are generated based on Voronoi cells, i.e., the BMUs for each data point are determined. If the number of points for a given neuron i is less than $MinPts$, then the subset $\mathcal{H}_i$ is reset to include all of the $MinPts$ closest to this neuron. This step is accomplished using standard k-Nearest Neighbors (k-NN) (Duda et al., 2000), in which all the neighbors are inside a hypersphere whose center is neuron i. Alternatively, the subset $\mathcal{H}_i$ may be generated using the grouping algorithm discussed in (Gokcay & Principe, 2002), which is hereafter referred to as a modified k-NN. Starting at neuron i, it iteratively generates the subset $\mathcal{H}_i$ in a single-linkage manner, i.e., by adding the closest data point to the current subset $\mathcal{H}_i$. The modified k-NN enables a better estimate of the data structure (Gokcay & Principe, 2002) since the geometry of the region where the samples of $\mathcal{H}_i$ scatter is not a hypersphere.

Subsequently, the sample mean $\boldsymbol{\mu}_i$ and sample covariance matrix Σ_i are computed (including neuron i in the calculations of Σ_i) using the subsets $\mathcal{H}_i$ with $MinPts$ ($MinPts \geq d + 1$, where d is the data dimensionality (Duda et al., 2000, p. 112)):

$$\boldsymbol{\mu}_i = E\left[\mathcal{H}_i\right] \approx \frac{1}{N_i} \sum_{\boldsymbol{x} \in \mathcal{H}_i} \boldsymbol{x}, \tag{16}$$

$$\Sigma_i = E\left[\left(\mathcal{H}_i - \boldsymbol{\mu}_i\right)\left(\mathcal{H}_i - \boldsymbol{\mu}_i\right)^T\right] \approx \frac{1}{N_i - 1} X_i^T X_i, \tag{17}$$

where $E[\cdot]$ is the expectation operator, N_i is the cardinality of the subset $\mathcal{H}_i$, and X_i is the subset $\mathcal{H}_i$ with zero mean.

Now, using this information, it is possible to compute the CIP or rCIP between each pair of subsets $\mathcal{H}$ or each pair of adjacent neurons, respectively. In the first case, the CIP is computed using all the data points in the subsets $\mathcal{H}_i$ and $\mathcal{H}_j$ associated with neurons i and j, respectively, to evaluate the distance between their underlying distributions. For the second case, when computing the rCIP, only the means $\boldsymbol{\mu}$ and the covariance matrices Σ of the neurons i and j are used to compute the rCIP between the neurons.

The information theoretic visualizations (IT-vis), which includes CIP-vis and rCIP-vis, are then generated in a fashion similar to the U-matrix (see Section 2.2). The resulting image follows the same arrangement as the U-matrix: for a SOM with rectangular grid size $a \times b$, the CIP-vis and rCIP-vis have a size of $(2a-1) \times (2b-1)$. The pixel position related to the diagonal connections (primary and secondary), is the mean of these connection values. The positions of the neurons themselves may be computed using functions such as mean, median, maximum, minimum, weighted average and so on. Here, the simple median and the rectangular topology was used.

In the output space, the resulting gray-level image depicts the similarity between the adjacent neurons located in the input space. The range of values obtained when computing the CIP or rCIP can be very wide, and in fact, extremely similar neurons may dominate the visualization with very high CIP or rCIP values, in the sense that similarity details expressed by the dynamic range of lower similarity values are compressed; thus, hiding smaller differences in other data regions. In order to overcome this issue and observe the finer details, a perceptual scaling is performed using a logarithmic transformation (Gonzalez & Woods, 2006, Sec. 3.2.2). This scaling has the same rationale generally employed to display Fourier Transforms of digital images. This transformation performs a mapping that expands and compresses the dynamic range of low and high similarity values, respectively. Concretely, taking the logarithm of the CIP is equivalent to computing the negative of the entropy H. Therefore, in most cases, the entropy is better suited for visualization. In order to keep the context of a similarity visualization, the negative of the entropy H^* and representative entropy rH^* were used throughout, instead of changing from similarity (CIP and rCIP) to dissimilarity (original H and rH formulas) representations,

$$H^* = \log\left(CIP + 1\right), \tag{18}$$

$$rH^* = \log\left(rCIP + 1\right). \tag{19}$$

Additionally, in order to improve the contrast in most visualizations, we added a constant with a value of 1. Hereafter, we refer to the visualizations CIP-vis, rCIP-vis, H*-vis or rH*-vis collectively as IT-vis. Algorithm 6 summarizes the steps to generate the IT-vis.

4. EXPERIMENTAL SET-UP

MATLAB, the SOM Toolbox (Vesanto et al., 1999) and the Cluster Validity Analysis Platform (Wang et al., 2009) were used. The presented visualization method was applied to real world and synthetic data sets from the UCI Machine Learning Repository (Bache

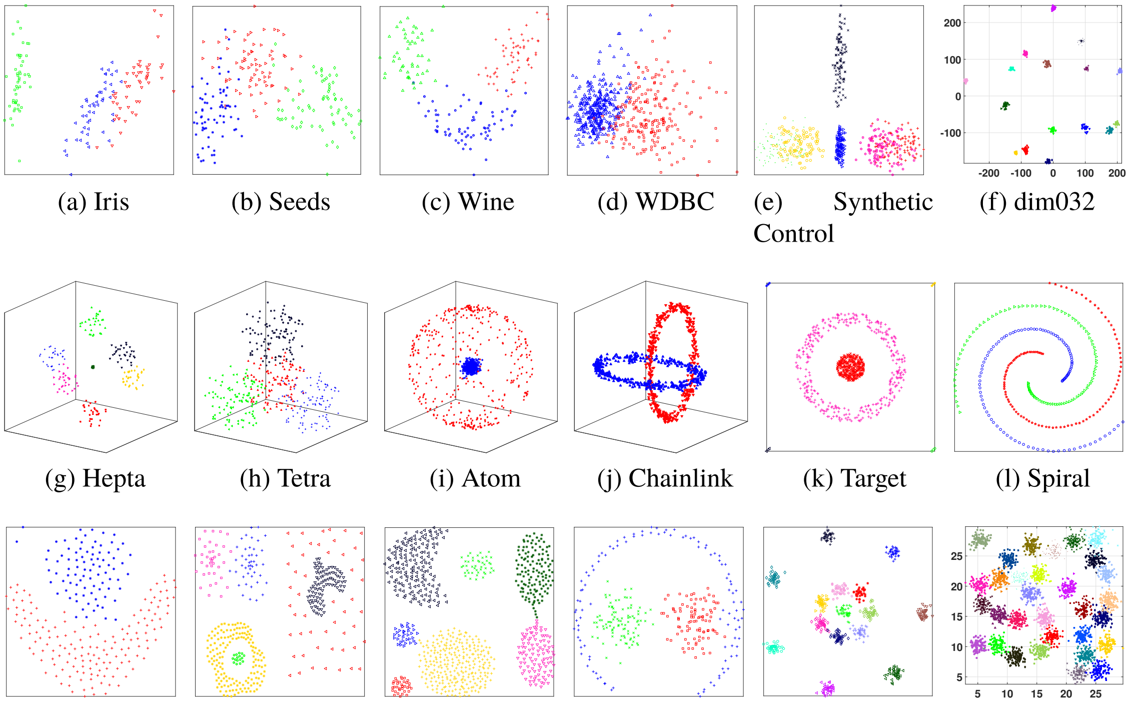
Algorithm 6: IT-vis Generation

```
Train the SOM.
Generate the subsets $\mathcal{H}_i$: if the number of points falling into the Voronoi region of
  neuron $i$ is less than $MinPts$, then recreate $\mathcal{H}_i$ using standard $k$-NN (Duda et al., 2000)
  or modified $k$-NN (Gokcay & Principe, 2002).
Compute the sample covariance matrices $\Sigma_i$ (17).
Compute the CIP (15) or rCIP; the later requires the additional calculation of the
  sample means $\boldsymbol{\mu}_i$ (16).
Generate the desired IT-vis visualization (CIP-vis, rCIP-vis, $\mathrm{H}^*$-vis or $\mathrm{rH}^*$-vis) follow-
  ing the arrangement of the U-matrix depicted in Figure 1.
```

& Lichman, 2013), the Fundamental Clustering Problem Suite (FCPS) (Ultsch, 2005) and also from (Fränti, Pasi et al., 2015) (which has a collection of data sets from (Chang & Yeung, 2008; Fränti et al., 2006; Fu & Medico, 2007; Gionis et al., 2007; Veenman et al., 2002; Zahn, 1971)). Figure 2 is an illustration of these.

Table 1 summarizes the characteristics of the data sets used in the experiments. Linear normalization was used in order to keep the data sets' attributes in the range [0, 1].

The SOM neurons were initialized in the subspace spanned by the eigenvectors of the two largest eigenvalues of the data covariance matrix (linear initialization); the maps were trained using the batch mode as it requires less parameters when tuning and converges faster (Kohonen, 2013). The Gaussian kernel function was used with a final radius (σ_f) equal to 1, and the total number of epochs was set to 3×10^3. The MATLAB code used in these experiments is being made available at the Applied Computational Intelligence Laboratory public GitLab repository (Brito da Silva & Wunsch II, 2017).



(a) Iris (b) Seeds (c) Wine (d) WDBC (e) Synthetic Control (f) dim032

(g) Hepta (h) Tetra (i) Atom (j) Chainlink (k) Target (l) Spiral

(m) Flame (n) Compound (o) Aggregation (p) Path based (q) R15 (r) D31

Figure 2. Data sets used in the experiments. The data sets *Iris*, *Wine*, *Seeds*, *WDBC*, *Synthetic Control* and *dim032* are depicted using principal component analysis (PCA) projection.

5. VISUALIZATION EXPERIMENTS

5.1. REPRESENTATIVE VISUALIZATIONS

The *Wine* data set was chosen as a case study to illustrate the visualizations presented in this subsection. The CIP-vis, H^*-vis, rCIP-vis and rH^*-vis computed from a 16 × 16 trained SOM are depicted in Figure 3 using 30 data samples per neuron. The first and second order statistics were computed using the modified k-NN (Gokcay & Principe, 2002). The standard CIP (Gokcay & Principe, 2002) was computed using all $MinPts$ data samples per neuron (Figure 3a and 3b), and the representative CIP (Araújo et al., 2013b) only used the neurons' statistics (Figure 3c and 3d).

Table 1. Data sets' characteristics.

Data set	# attributes	# samples	# clusters	type	source
Iris (Fisher, 1936)	4	150	3	Real World	UCI
Wine	13	178	3	Real World	UCI
Seeds[a] (Charytanowicz et al., 2010)	7	210	3	Real World	UCI
WDBC[b]	30	569	2	Real World	UCI
Synthetic Control[c]	60	600	6	Synthetic	UCI
Atom	3	800	2	Synthetic	FCPS
Chainlink	3	1000	2	Synthetic	FCPS
Hepta	3	212	7	Synthetic	FCPS
Target	2	770	2+outliers	Synthetic	FCPS
Tetra	3	400	4	Synthetic	FCPS
dim032 (Fränti et al., 2006)	32	1024	16	Synthetic	DIM-sets (high)
Flame (Fu & Medico, 2007)	2	240	2	Synthetic	Shape Sets
Compound (Zahn, 1971)	2	399	6	Synthetic	Shape Sets
Aggregation (Gionis et al., 2007)	2	788	7	Synthetic	Shape Sets
Spiral (Chang & Yeung, 2008)	2	312	3	Synthetic	Shape Sets
Path based (Chang & Yeung, 2008)	2	300	3	Synthetic	Shape Sets
R15 (Veenman et al., 2002)	2	600	15	Synthetic	Shape Sets
D31 (Veenman et al., 2002)	2	3100	31	Synthetic	Shape Sets

[a] Contributors gratefully acknowledge support of their work by the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

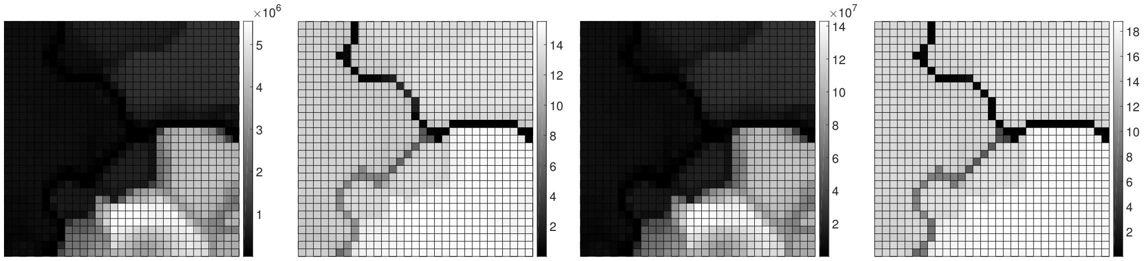
[b] Wisconsin Diagnostic Breast Cancer.

[c] Image courtesy of Eamonn Keogh.

As can be inferred from these figures, the CIP-vis and rCIP-vis are visually similar. The same can be stated about the H*-vis and rH*-vis pair. The computational cost, however, is very different. As expected from the use of representatives to compute the CIP (Araújo et al., 2013a,b), the computation of rCIP-vis/rH*-vis is considerably faster than CIP-vis/H*-vis. For instance, regarding the *Wine* data set, Figure 4 depicts the elapsed time for computing the CIP-vis, the rCIP-vis and the U-matrix (adapted from (Araújo et al., 2013a,b)). The U-matrix is included as a benchmark for baseline comparison purposes with an image-based SOM visualization. Although not representative of an optimized implementation, it still allows for a rough estimate of computational cost measured via elapsed time.

The total elapsed time is the sum of the time needed to compute the data distance matrix, the search for the $MinPts$ (according to the version of k-NN), the statistical parameters (i.e., mean and covariance matrix) and the calculation of the CIP/rCIP for the heat map generation. The elapsed time necessary to compute the rCIP is offset by a constant in comparison to the U-matrix (Figure 4b), which is faster; of course, they are a function of the SOM size but not of $MinPts$, which is the case for the CIP (Figure 4a). The heavy computations required by the representative visualizations are related to the parameters for both k-NN methods (cf. Figs. 4c and 4d), which include all of the tasks mentioned except the computation of the CIP/rCIP for the matrix plot generation. Although more computationally intensive than the standard k-NN, the modified k-NN conforms better with the underlying data structure (Gokcay & Principe, 2002).

Naturally, the computational demand increases with the size of the data, SOM and the $MinPts$ value. Nonetheless, the running times can be improved, for instance, by using fast standard k-NN implementations such as (Garcia et al., 2008; Wang, 2011), which use parallel computing and clustering approaches, respectively. The modified k-NN can be viewed as a call to the standard k-NN with $k = 1$: the samples in the current subset $\mathcal{H}$ are the query points, which are updated by including the closest sample belonging to the complement of $\mathcal{H}$. Additionally, the calculation of the parameters for each neuron are independent and, for large SOMs, can be performed in parallel to further decrease the running times. Therefore, for practical applications and purposes, the representative visualizations are recommended due to the speed boost (Araújo et al., 2013a,b) without compromising visual quality. Moreover, in general, the visualization of the cross-entropy (rH*-vis) generates more appealing visualizations than the cross-information potential (rCIP-vis), as observed in Figure 3.



(a) CIP-vis (b) H*-vis (c) rCIP-vis (d) rH*-vis

Figure 3. SOM IT-vis types for the *Wine* data set using $MinPts = 30$: (a) CIP-vis, (b) H*-vis, (c) rCIP-vis and (d) rH*-vis.

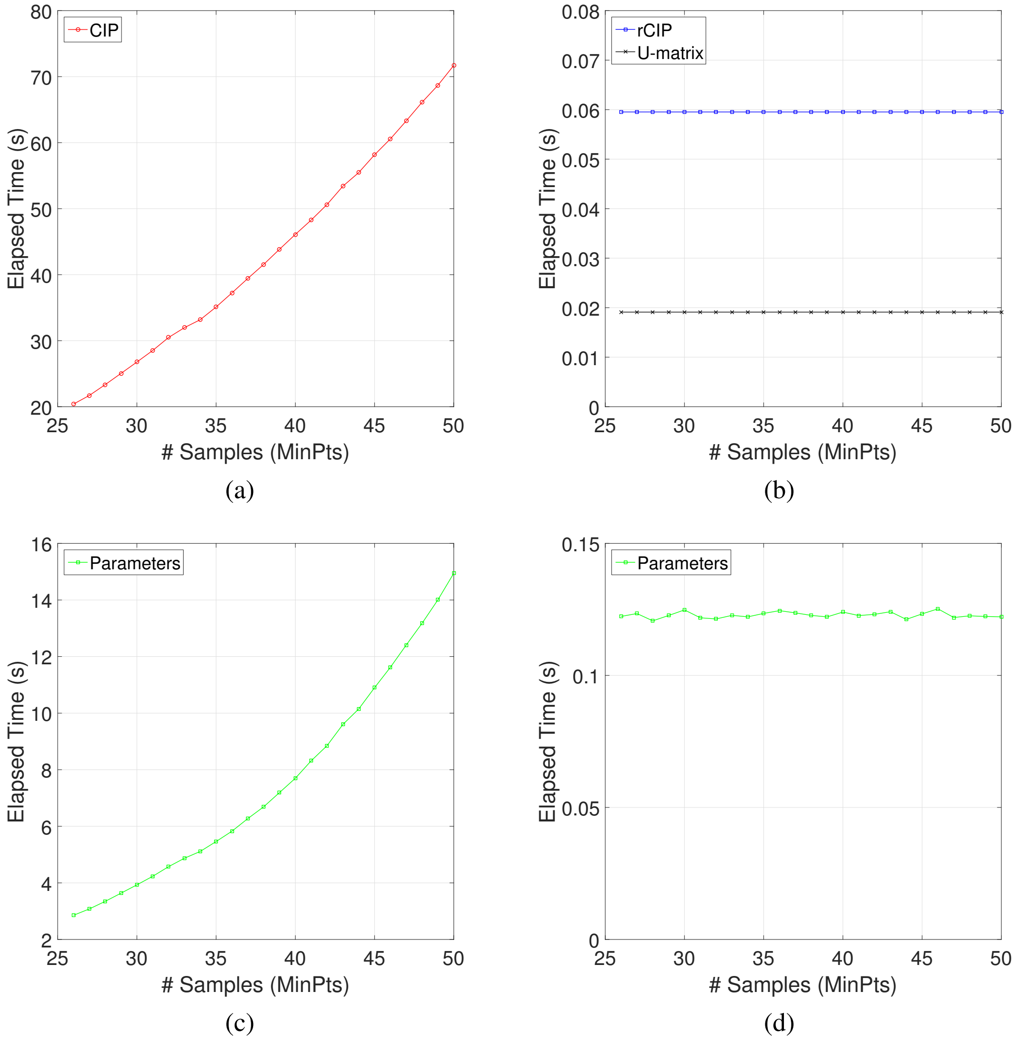


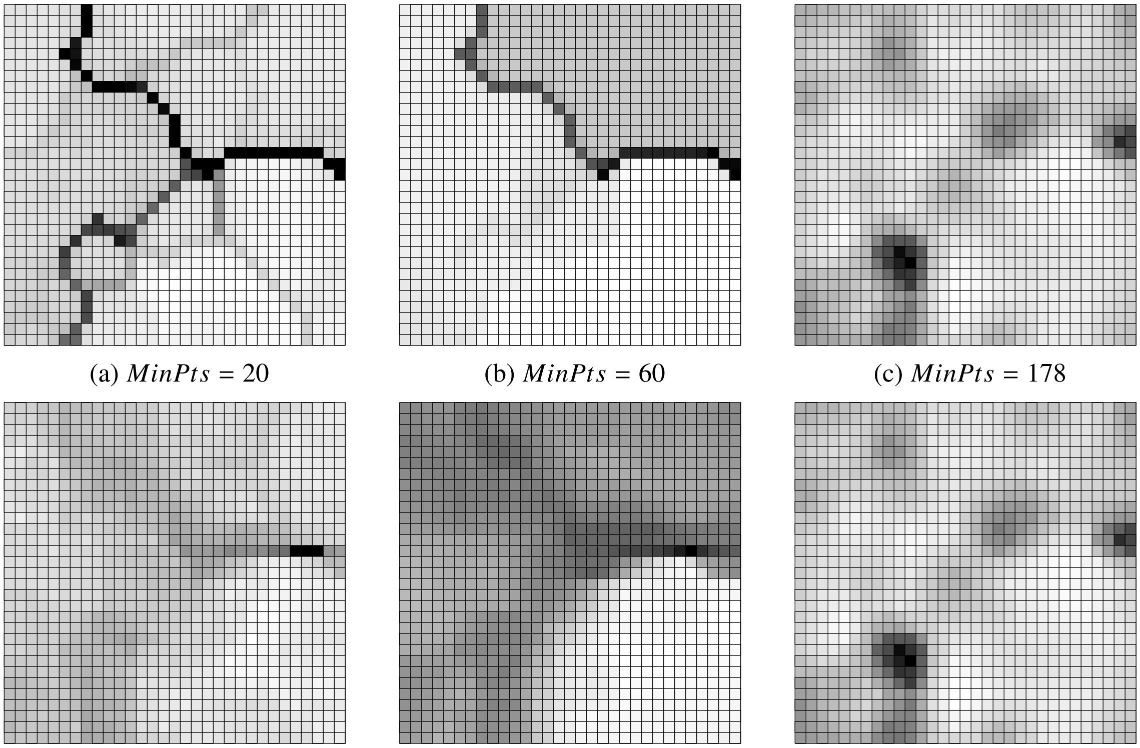
Figure 4. Elapsed time to compute (a) CIP, (b) rCIP and U-matrix (adapted from (Araújo et al., 2013a,b)), and the necessary parameters (subsets $\mathcal{H}_i$ and statistics) using (c) the modified *k*-NN (Gokcay & Principe, 2002) and (d) the standard *k*-NN (Duda et al., 2000).

5.2. *K*-NN METHODS AND SENSITIVITY TO THE *MINPTS* PARAMETER

The generation of the subsets $\mathcal{H}_i$ plays a major role in determining the aspect of the IT-vis. Here, the standard k-NN algorithm (Duda et al., 2000) was used to select the $MinPts$ data samples closer to a given neuron i, as well as the modified k-NN algorithm (Gokcay & Principe, 2002). In the latter k-NN approach, the first sample $\boldsymbol{x}_k$ from the data set X included in the subset $\mathcal{H}_i$ is the closest to neuron $\boldsymbol{w}_i$ (the starting point). The next sample to be selected is the one that has the minimum distance to either $\boldsymbol{x}_k$ or $\boldsymbol{w}_i$ among all remaining samples in a single linkage fashion. The subsequent samples are added analogously: they are the ones closest to the current subset $\mathcal{H}_i$ at time t ($\min\limits_{k,l} \|\boldsymbol{x}_k - \boldsymbol{x}_l\|$, $\boldsymbol{x}_k \in \mathcal{H}_i(t)$, $\boldsymbol{x}_l \in \overline{\mathcal{H}}_i(t) = X - \mathcal{H}_i(t)$). The process is repeated until $|\mathcal{H}_i| = MinPts$.

Regarding the visualization outcome, the difference between both k-NNs lies in the fact that the second approach leads to a higher granularity, in the sense that it captures and displays more strongly and effectively even the smallest differences, i.e. even small non-uniformities within the data are captured, and as such, a very sensitive and sharp visualization is obtained, where strong boundaries between clusters and sub-regions within the clusters can be observed. On the other hand, using standard k-NN, results in a “blurring” effect. In Figure 5, this effect is noticeable on the *Wine* data set visualization using both k-NN methods as the generators of the subsets $\mathcal{H}_i$. The advantage of using the modified k-NN (Figs. 5{a, b, c}) over the standard k-NN (Figs. 5{d, e, f}) is that the clusters' edges are thinner (in most cases one pixel wide), and the regions are sharply separated. In general, due to the sharp visualizations generated, the modified k-NN (Gokcay & Principe, 2002) is recommended when sharp boundaries are desired.

Figure 5 also depicts the variation of the rH*-vis for several values of the $MinPts$ parameter. As shown, there is a reasonably large interval in which the $MinPts$ parameter provides a clear visualization of the *Wine* data set. Both IT-vis computed using the different


(a) $MinPts$ = 20

(b) $MinPts$ = 60

(c) $MinPts$ = 178

(d) $MinPts$ = 20

(e) $MinPts$ = 60

(f) $MinPts$ = 178

Figure 5. The rH*-vis for the *Wine* data set using the modified k-NN (Gokcay & Principe, 2002) (a, b, c) and standard k-NN (Duda et al., 2000) (d, e, f). Different values for the $MinPts$ were used and performance is seen to be robust to the choice of this parameter.

k-NNs exhibit the same hierarchical behavior of smoothed data histograms (Pampalk et al., 2002) and bounday-matrix (Manukyan et al., 2012) visualizations when the smoothing parameter and kernel size are varied, respectively.

5.3. VISUALIZATION EXAMPLES AND SENSITIVITY TO SOM SIZE

In this subsection the effect of the SOM size on the IT-vis is investigated. Following the previous subsections' recommendations, the rH*-vis using the modified k-NN was chosen. Figs. 6 through 11 depict six high-dimensional data sets that were used for proof of concept: *Iris*, *Seeds*, *Wine*, *Synthetic Control*, *dim032* and *WDBC*, respectively. These examples illustrate the rH*-vis characteristics. Additionally, for comparison purposes, other visualization methods are also presented viz., CONNvis2 (Taşdemir, 2010), U-matrix (Ultsch, 1993; Ultsch & Siemon, 1990), as well as the SOM neuron's labels obtained by majority voting. The latter depicts the data appearance on the SOM lattice, and it is used to visually assess the clusters detected by the visualization methods. The CONNvis2 was selected to render the available information from the connectivity matrix CONN on the SOM grid. The connection strength between a pair of neurons was encoded using gray-scale and line width (darker and thicker lines represent stronger connections). Following (Taşdemir, 2010) recommendations, the entries of the CONN matrix were normalized by the mean of the largest value for each neuron; next, values greater than 1 were set to 1. Three SOM sizes were used (small, medium and large), ranging from densely-matched to sparsely-matched regarding the ratio between the number of data samples and neurons.

The *Iris* data set comprises three classes of the iris plant (see Figure 6{a, c, e}): Setosa (+), Versicolor (○) and Virginica (△); the latter two have a high degree of overlap and are linearly separable from the first. Examining Figure 6, the U-matrix is unable to visually convey the existence of all three classes in any of the SOM sizes. Moreover, for the small SOM (Figure 6h), even the boundary between the linearly separable classes is difficult to observe. Disregarding the weak connections, the CONNvis2 reveals the clusters for small and medium SOMs. The rH*-vis was able to reveal all of the clusters with increasing resolution from small to large SOMs. For all of these, the vast majority of the cluster boundaries are one pixel wide and the within-cluster pixel values are very homogeneous.

The *Seeds* data set encompasses three classes of wheat (see Figure 7{a, c, e}): Kama (+), Rosa (○) and Canadian (△). When analyzing Figure 7, it becomes noticeable that the U-matrix does not reveal clear cluster information for any of the trained SOMs. The presence of three major clusters can be clearly observed through the rH*-vis in all SOM sizes, especially for small maps (Figure 7g), for which the data structure information that is provided is clearer than CONNvis2. Figure 8 presents the visualizations for the *Wine* data set, which is composed of three classes of wines. Although well-behaved, these

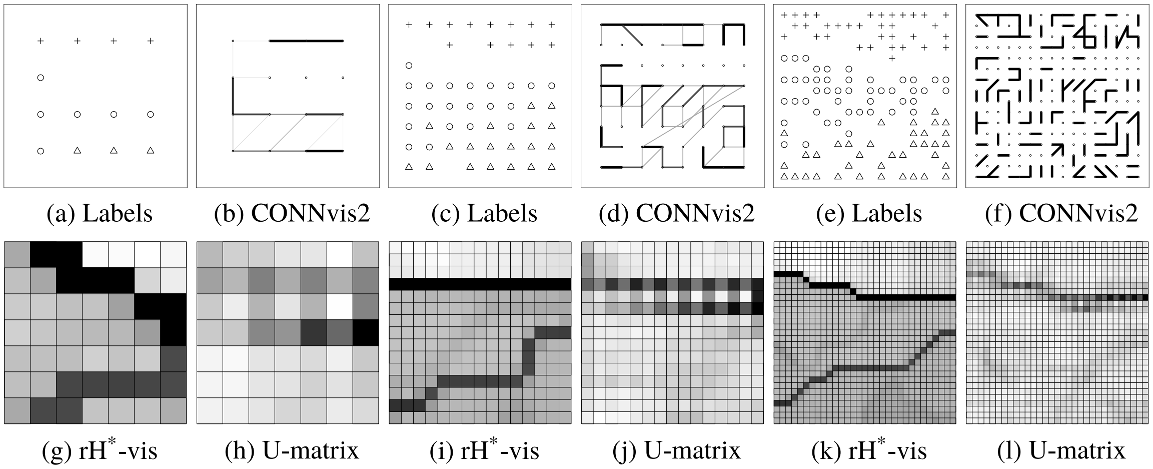


Figure 6. *Iris* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16.

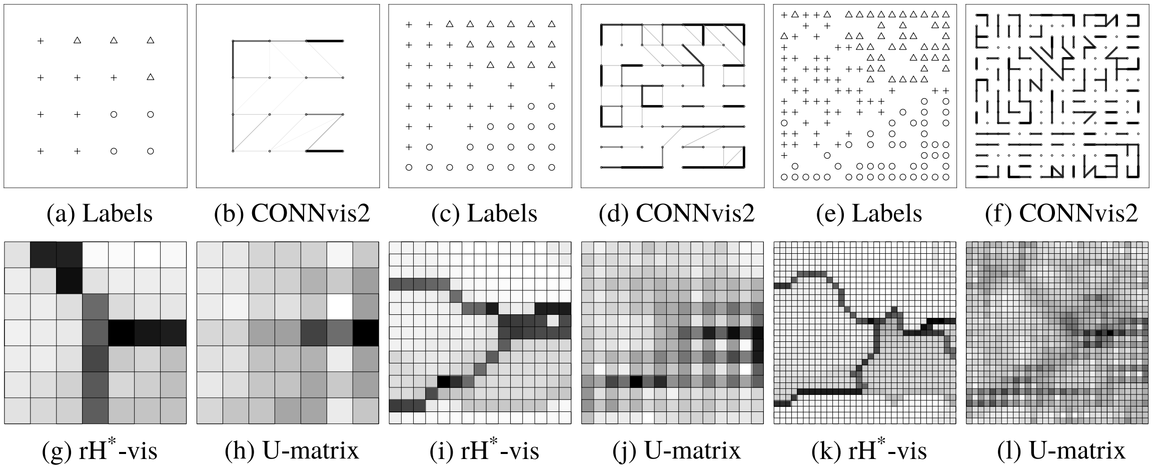
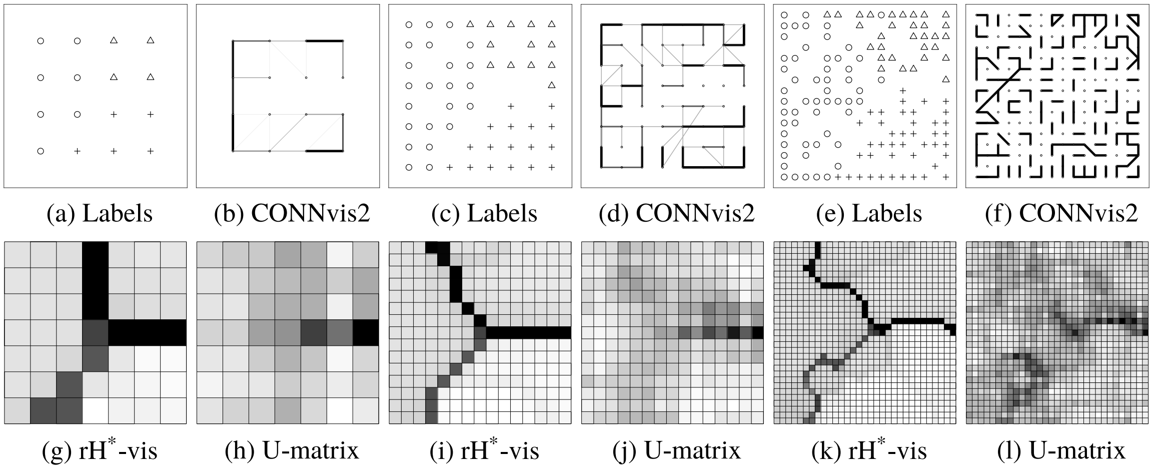


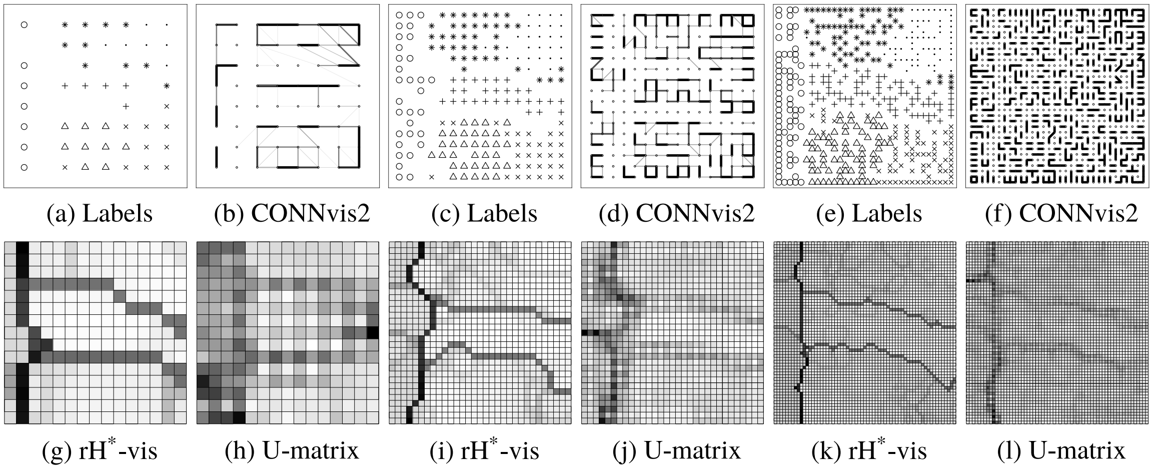
Figure 7. *Seeds* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16.

classes are not well-separated. Inferring the existence of three classes from the rH*-vis is straightforward for all SOMs: the cluster boundaries are sharp and thin (one pixel wide in almost their full extension). The U-matrix does not enable a clear visualization of the classes. The CONNvis2 behaves similarly to the *Iris* data set, i.e., disregarding the weak connections, it reveals the classes for small and medium maps.


(a) Labels (b) CONNvis2 (c) Labels (d) CONNvis2 (e) Labels (f) CONNvis2

(g) rH*-vis (h) U-matrix (i) rH*-vis (j) U-matrix (k) rH*-vis (l) U-matrix

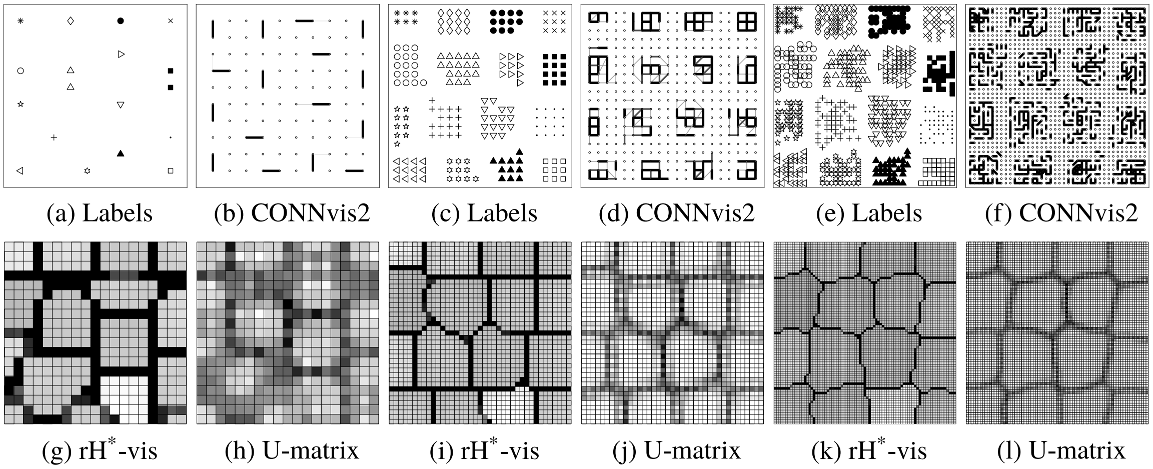
Figure 8. *Wine* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 26, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 4 × 4, (e)-(h) 8 × 8 and (i)-(l) 16 × 16.


(a) Labels (b) CONNvis2 (c) Labels (d) CONNvis2 (e) Labels (f) CONNvis2

(g) rH*-vis (h) U-matrix (i) rH*-vis (j) U-matrix (k) rH*-vis (l) U-matrix

Figure 9. *Synthetic Control* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 61, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 8 × 8, (e)-(h) 16 × 16 and (i)-(l) 32 × 32.

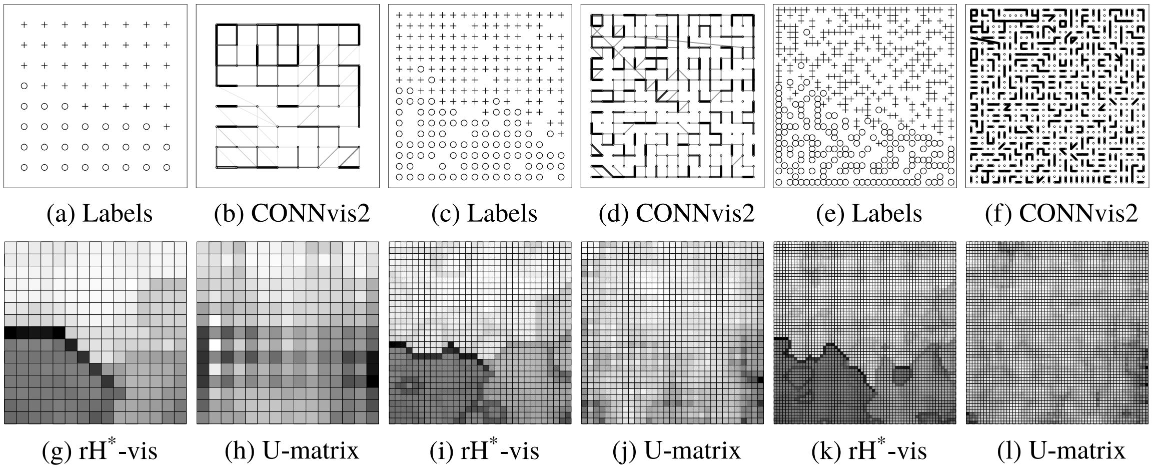
The *Synthetic Control* data set consists of six classes of control charts (see Figure 9{a, c, e}): normal (+), cyclic (○), increasing trend (△), decreasing trend (·), upward shift (×) and downward shift (∗). None of the visualizations in Figure 9 fully reveal all six clusters. When examining the U-matrix, for all SOM sizes, there is a clear division between the cyclic and the remaining clusters; whereas its borders that divide the data set into four



(a) Labels (b) CONNvis2 (c) Labels (d) CONNvis2 (e) Labels (f) CONNvis2

(g) rH*-vis (h) U-matrix (i) rH*-vis (j) U-matrix (k) rH*-vis (l) U-matrix

Figure 10. *dim032* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 33, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 10 × 10, (e)-(h) 20 × 20 and (i)-(l) 40 × 40.



(a) Labels (b) CONNvis2 (c) Labels (d) CONNvis2 (e) Labels (f) CONNvis2

(g) rH*-vis (h) U-matrix (i) rH*-vis (j) U-matrix (k) rH*-vis (l) U-matrix

Figure 11. *Wisconsin Diagnostic Breast Cancer* data set: {a, c, e} SOM neurons' labels according to majority voting, {b, d, f} CONNvis2, {g, i, k} rH*-vis using the modified k-NN and $MinPts$ = 40, {h, j, l} U-matrix. The SOM sizes used were: (a)-(d) 8×8, (e)-(h) 16 × 16 and (i)-(l) 32 × 32.

clusters (normal, cyclic, decreasing trend and downward shift, increasing trend and upward shift), are extremely faint in comparison to the rH*-vis. In the latter, the borders are much more pronounced and thin, although weaker than the boundary that separates the two aforementioned major clusters. In turn, the CONNvis2 provides a clearer definition of four clusters for the small SOM. For the medium SOM, although the decreasing trend and downward shift clusters are separated from each other, the cyclic cluster is further divided into three sub-clusters. The *dim032* contains sixteen well-separated clusters. They were all clearly depicted by all visualization methods. Specifically, regarding the rH*-vis and the U-matrix, the resolution increased with size; again, rH*-vis depicted thinner and sharper boundaries.

Finally, the *Wisconsin Diagnostic Breast Cancer* (WDBC) data set encloses two classes (see Figure 11{a, c, e}): benign (+) and malignant ($\circ$). This data set does not exhibit a clear separation between its classes. For the *WDBC*, none of the U-matrices of the different SOM sizes were able to convey meaningful information about the data structure. On the other hand, the rH*-vis clearly reveals that the rH* values of the benign class are in fact much larger than those of the malignant class, which are mostly represented by the bright and dark regions in Figure 11{g, i, k}, respectively. In fact, this characteristic would be enhanced even further if the rCIP-vis were to be observed and also accentuated by increasing the value of $MinPts$. This data set behaves differently from the previous ones in which there were areas with large within-class cross-information potential similarity that were sufficiently dissimilar to each other, thereby generating sharp and thin boundaries between the clusters. For the WDBC, however, the benign class has a much more compact and defined structure as opposed to the malignant class, in which the samples are more spread out (c.f the PCA projection of this data set in Figure 2d with the additional support of markers and colors to denote the labels of the samples from each class: benign and malignant samples are depicted as blue circles and red triangles, respectively). This fact is portrayed in the rH*-vis. Thus, meaningful information can be extracted from it in all SOM sizes. CONNvis2 for small and medium SOMs depict partial aisles positioned in the boundaries of clusters, which resemble the arrangement of the rH*-vis regions' shapes.

6. A CLUSTERING APPLICATION THROUGH IT-VIS

When considering image-based SOM visualizations, digital image processing segmentation techniques may be used to perform clustering tasks. In order to analyze the performance and behavior of the rCIP-vis and rH*-vis when used for clustering, the clustering methodology discussed in (Costa & Netto, 1999, 2001) is followed, which is a post-processing strategy applied to the U-matrix that aims to automate the clustering process. Briefly, this segmentation strategy is as follows: generate the U-matrix image, filter the image, define the image markers using stable regions to allow the application of the marker-controlled watershed method (Meyer, 1994), and identify each neuron region on the SOM grid. The stable regions consists of a multi-level threshold scanning over the U-matrix in which connected components labeling (CCL) (Haralick & Shapiro, 1992) is used to count the number of clusters; the clustering state with the largest lifetime as a function of this global thresholding operation is selected using the lowest threshold value. Finally, the SOM neurons are assigned labels, according to the U-matrix segmentation. The remaining unlabeled neurons from the watershed ridges may be labeled using standard *k*-NN. These SOM labels are carried back to the samples.

First, the IT-vis were obtained as explained in Section 3. Next, the methods of Otsu's global thresholding (Otsu, 1979) and stable regions (Costa & Netto, 1999, 2001) were used to generate the markers for the application of the watershed segmentation algorithm using a 4-neighborhood connectivity. Regarding the Otsu's generated markers, a low pass Gaussian filter was previously applied to blur the image (kernel width equal to $\sqrt{2}$) and enable an increased Otsu's thresholding performance (Gonzalez & Woods, 2006). Regarding the stable regions markers, as opposed to (Costa & Netto, 1999, 2001), no pre-processing stage using mathematical morphology operators nor other filtering approaches were used to enhance the raw images. The largest threshold value was used from the selected stable region (IT-vis are similarity images as opposed to the U-matrix which is a dissimilarity image).

Here, all of the intensity images were normalized to the range $[0, 1]$; the stable regions' threshold was scanned from 0 to 1 with a step size equal to 0.1 and the largest region with a number of clusters greater than 1, if it existed, would be selected. A CCL with a 4-neighborhood connectivity was used to estimate the number of clusters; after the application of the watershed method, small areas with less than 2 pixels were eliminated (except for small SOMs), and the regions representing the clusters in the image were identified. The remaining border neurons were assigned to the clusters using standard k-NN method with $k = 1$.

As mentioned previously, the segmented image represents the partition of the SOM, and it is ultimately used to recover the clusters present in the data set, by carrying back the labels of the neurons to the data samples. Algorithm 7 sums up the methodology used in the experiments to segment the gray-level images and compare their resulting partitions (Costa & Netto, 1999, 2001). The watershed algorithm is used to generate thin cluster boundaries from the markers (cluster cores) using only the information present in the visualization; the assigning algorithm chosen to label the border neurons may exert a significant influence over the clustering performance, especially when the markers are small regions that need to be grown (Brito da Silva & Ferreira Costa, 2014).

Algorithm 7: Segmentation of image-based SOM visualizations (Costa & Netto, 1999, 2001)

```
Train the SOM.
Generate the matrix plot (IT-vis, U-matrix).
Obtain the markers (Otsu's method or stable regions).
Apply the watershed transform (marker-controlled).
Eliminate small pixel regions (less than 2 pixels).
Identify clusters (CCL).
Assign the edge neurons to clusters using a suitable algorithm ($k$-NN with $k = 1$).
Label the data set samples according to the discovered SOM neurons' labels.
```

The results using rCIP-vis and rH*-vis were compared to the application of the same methods to partition the U-matrix and the logarithm of the U-matrix. Additionally, the results were also compared to k-means (MacQueen, 1967) and CONN linkage clustering (Taşdemir et al., 2011); the latter is a hierarchical clustering algorithm that has the same rationale as the standard average linkage but uses the CONN connectivity strength (6)-(8) as the similarity measure between the SOM neurons. Here, a dissimilarity matrix D was generated from the CONN similarity matrix S as $D = max(S) - S + 1$, with zeros in the main diagonal. Next, the standard average linkage hierarchical clustering method was applied. Regarding k-means, 30 runs with 100 maximum iterations were performed over the SOM neurons and the solution with the lowest quantization error was selected. For all methods, the data sets were partitioned according to the SOM labels.

The quality of the partitions obtained from the SOM were assessed in terms of the normalized mutual information (NMI) (Manning et al., 2008, Sec. 16.3) and the adjusted Rand index (AR) (Hubert & Arabie, 1985) external validity indices,

$$NMI = \frac{I(Y,Z)}{\left[H(Y) + H(Z)\right]/2}, \tag{20}$$

$$AR = \frac{\binom{N}{2}(tp+tn) - \left[(tp+fp)(tp+fn) + (fn+tn)(fp+tn)\right]}{\binom{N}{2}^2 - \left[(tp+fp)(tp+fn) + (fn+tn)(fp+tn)\right]}, \tag{21}$$

where I and H denote mutual information and entropy, respectively, regarding the ground truth Y and the output partition Z. N denotes the cardinality of the data set; tp, tn, fp and fn stand for true positive, true negative, false positive and false negative, when comparing each pair of samples in Y and Z. The ranges of NMI and AR are $[0, 1]$ and $[-1, 1]$, respectively; the optimal value for both validity indices is 1.

For each data set, the clustering task was performed using three sizes of SOM networks (small, medium and large). The medium SOM sizes had their rectangular grid dimensions selected in such a way that their ratios were proportional to the square root of the ratio between the two largest eigenvalues of the data covariance matrix (Kohonen, 2001), and the number of neurons was approximately equal to $5\sqrt{N}$ (Vesanto & Alhoniemi, 2000), where N is the number of samples. The small and large maps are half and double the size of the medium map, respectively. However, here we only report the experimental results using small and large SOMs, since they better emphasize characteristics of the SOM clustering methods. As expected, for each data set and SOM size there is a specific combination of parameters that leads to improved clustering results. Therefore, for consistency, all images received the same treatment and the same methods were applied.

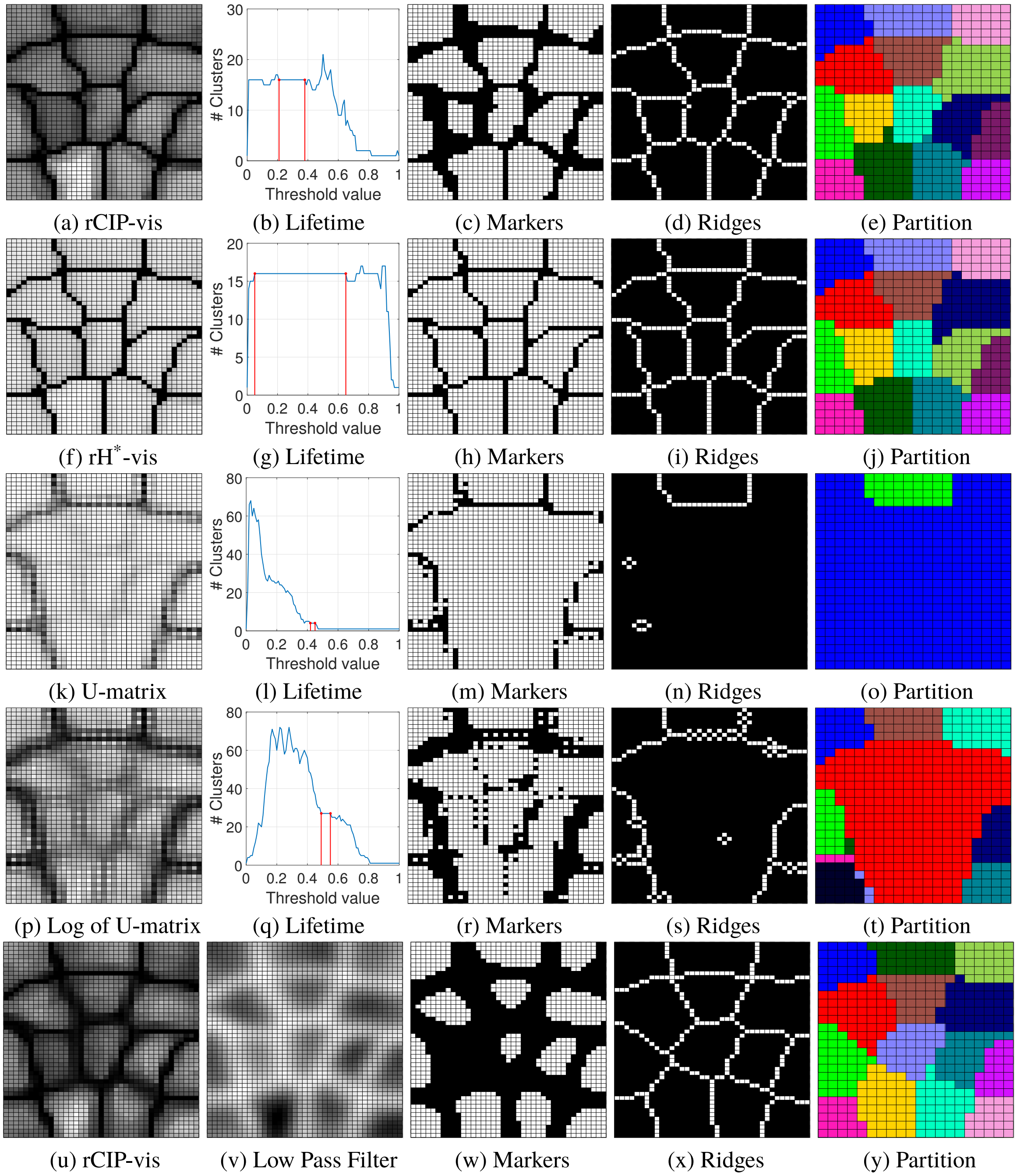
6.1. RESULTS AND DISCUSSIONS

The results obtained with the image segmentation methodology while using the parameters presented in Section 4 are shown in detail for the *R15* data set in Figure 12 using a 24 × 20 SOM. This data set was chosen as a case study to illustrate the segmentation stages. These figures depict the visualizations, the markers used by and the ridges obtained from the watershed algorithm, as well as the final partition that was obtained using each method applied to rCIP-vis, rH*-vis, U-matrix and logarithm of the U-matrix. The low pass filter pre-processing stage and the threshold versus the number of clusters graph are depicted in the cases of Otsu's and stable regions methods, respectively.

After the images were generated using the trained SOM, the pixel values are explicitly divided into two groups using Otsu's algorithm: clusters' cores (within-cluster neurons) and boundaries (between-cluster neurons). The last four rows of Figure 12 depict the output of Otsu's algorithm. Alternatively, the markers were also generated using the stable regions approach (assuming the number of clusters is greater than 1); whose outputs are depicted in the first four rows of Figure 12. Next, the markers generated by either method are used as inputs to the watershed algorithm. Finally, CCL was used to identify the regions. Regions with an area smaller than two pixels were disregarded; and the assignment of non-labeled neurons was performed using standard k-NN ($k = 1$), thereby generating the final partition (last column of Figure 12). Finally, their labels are carried back to the data set and the external validity index of the partition is calculated.

Regarding the clustering of the *R15* data set, it is noticeable that the rH*-vis provided a much larger lifetime plateau of the clustering state (Figs. 12{b, g, l, q}) than the other image-based SOM visualizations; the 16th cluster has only one pixel, which is removed in step 5 of Algorithm 7 - the elimination of small pixel regions is performed for all images. Furthermore, the rH*-vis' thin boundaries are already very similar to the output of the watershed method. As expected, the U-matrix requires more pre and post-processing. For instance, there is a significant amount of connected regions and small pixel groups when applying the stable regions method in the U-matrix-based approaches in Figure 12; additionally, the majority of their best results using large SOMs were obtained by applying Otsu's method, which has a smoothing pre-processing stage.

The peak performance of all of the clustering methods using small and large SOM sizes are depicted in Tables 2 and 3. Regarding the CONN linkage, the peak performance was obtained by scanning all merging levels of the dendrogram and selecting the cut-off value that maximizes the external validity indices. The number of clusters parameter of the k-means algorithm was scanned from 1 to $\sqrt{N}$, where N is the number of data samples, and the value that optimizes the external validity indices was selected. Peak performance for IT-vis images were obtained by scanning the $MinPts$ parameter in the interval $[2 \times dim, 50]$, where dim is the dimensionality of each data set. The exceptions were the *dim032*, *Synthetic Control* and *WDBC* data sets; for the latter, the $MinPts$ was scanned in the range $[dim + 1, 100]$. No parameters were scanned for the U-matrix nor the logarithm of the U-matrix. For all approaches, except for k-means due to random initialization, the results are deterministic for a given set of parameters. The appropriate



(a) rCIP-vis (b) Lifetime (c) Markers (d) Ridges (e) Partition

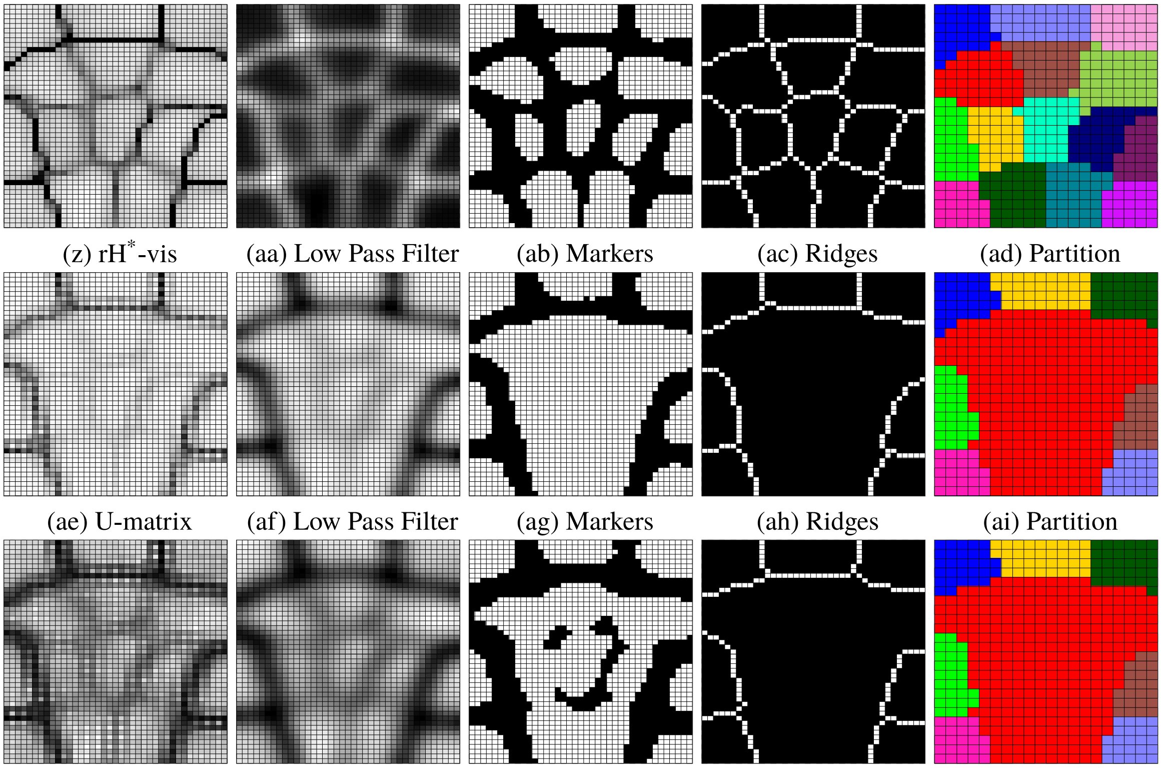
(f) rH*-vis (g) Lifetime (h) Markers (i) Ridges (j) Partition

(k) U-matrix (l) Lifetime (m) Markers (n) Ridges (o) Partition

(p) Log of U-matrix (q) Lifetime (r) Markers (s) Ridges (t) Partition

(u) rCIP-vis (v) Low Pass Filter (w) Markers (x) Ridges (y) Partition

Figure 12. Clustering the *R15* data set with $MinPts = 25$. First four rows: gray-level image, stable region, markers, watershed ridges and final partitions for rCIP-vis (a)-(e), rH*-vis (f)-(j), U-matrix (k)-(o) and logarithm of U-matrix (p)-(t). The IT-vis used the modified *k*-NN (Gokcay & Principe, 2002) to generate $\mathcal{H}$. Last four rows: gray-level image, processed image with Gaussian low pass filter, markers, watershed ridges and final partitions for rCIP-vis (u)-(y), rH*-vis (z)-(ad), U-matrix (ae)-(ai) and logarithm of U-matrix (aj)-(an). The IT-vis used the standard *k*-NN (Duda et al., 2000) to generate $\mathcal{H}$.



(z) rH*-vis (aa) Low Pass Filter (ab) Markers (ac) Ridges (ad) Partition

(ae) U-matrix (af) Low Pass Filter (ag) Markers (ah) Ridges (ai) Partition

(aj) Log of U-matrix (ak) Low Pass Filter (al) Markers (am) Ridges (an) Partition

Figure 12. (Cont.) Clustering the *R15* data set with $MinPts = 25$. First four rows: gray-level image, stable region, markers, watershed ridges and final partitions for rCIP-vis (a)-(e), rH*-vis (f)-(j), U-matrix (k)-(o) and logarithm of U-matrix (p)-(t). The IT-vis used the modified *k*-NN (Gokcay & Principe, 2002) to generate $\mathcal{H}$. Last four rows: gray-level image, processed image with Gaussian low pass filter, markers, watershed ridges and final partitions for rCIP-vis (u)-(y), rH*-vis (z)-(ad), U-matrix (ae)-(ai) and logarithm of U-matrix (aj)-(an). The IT-vis used the standard *k*-NN (Duda et al., 2000) to generate $\mathcal{H}$.

value for the parameters of each clustering method may be selected, for instance, by using heuristics based on the relative validity indices such as the ones discussed in (Gonçalves et al., 2006; Taşdemir et al., 2011).

Regarding the different SOM sizes, *k*-means, CONN linkage, U-matrix and logarithm of the U-matrix, and IT-vis achieved the best performance in $\{4, 5\}$, $\{11, 0\}$, $\{0, 6\}$, $\{9, 16\}$ out of the 18 data sets, respectively. The numbers inside the braces refer to the performance using the small and large SOMs, in that order. Note that some data sets had the same performance for more than one method and the performance of the image-based visualizations were combined across the different image processing methods for each data set (i.e., all U-matrix based images, all IT-vis based images). As expected, CONN linkage outperformed the other methods for small SOM sizes (densely-matched SOMs), whereas clustering through IT-vis was most successful for large SOMs. For IT-vis, the best results were usually obtained using stable regions and the modified *k*-NN, and, in general, performance improved with size. Moreover, the rH*-vis yielded better results overall when compared to the rCIP-vis. For instance, with respect to large SOMs, 16 of the overall best performance for the 18 data sets were achieved by clustering one of the IT-vis varia-

Table 2. Peak performance for CONN linkage, *k*-means, U-matrix and logarithm of U-matrix.

Data set	SOM	CONN		U-matrix				logarithm of U-matrix				K-means	
	network	Linkage		Stable Regions		Otsu's Method		Stable Regions		Otsu's Method		Algorithm	
	size	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR
Small SOM sizes													
Iris	8 × 2	0.7582	0.7302	0.7201	0.5584	0.7337	0.5681	0.7201	0.5584	0.7337	0.5681	0.7507	0.7173
Wine	4 × 4	0.8326	0.8613	0.6304	0.4739	0.0000	0.0000	0.6304	0.4739	0.0000	0.0000	0.8529	0.8685
Seeds	8 × 3	0.6618	0.6699	0.4570	0.3110	0.5942	0.4883	0.4570	0.3110	0.5260	0.4605	**0.7171**	0.7048
Chainlink	9 × 5	**1.0000**	**1.0000**	0.3738	0.2801	0.4195	0.3059	0.3738	0.2801	0.4195	0.3059	0.4772	0.3022
Hepta	5 × 4	**1.0000**	**1.0000**	0.9451	0.8411	0.0000	0.0000	0.9451	0.8411	0.0000	0.0000	**1.0000**	**1.0000**
Tetra	6 × 5	0.9294	0.9348	0.8406	0.8036	0.9402	0.9535	0.7902	0.6961	0.9452	0.9538	0.9452	0.9538
Spiral	5 × 5	**0.2984**	0.1030	0.0280	0.0088	0.0053	0.0022	0.0280	0.0088	0.0088	0.0027	0.1922	0.0507
R15	6 × 5	0.9028	0.8236	0.5109	0.1985	0.0000	0.0000	0.5109	0.1985	0.0000	0.0000	**0.9212**	**0.8548**
Path based	6 × 4	0.4948	0.4024	0.4910	0.3990	0.4910	0.3990	0.3954	0.2321	0.4910	0.3990	0.4910	0.4038
Flame	5 × 4	**0.7581**	**0.8542**	0.5146	0.6371	0.5146	0.6371	0.4124	0.3384	0.5146	0.6371	0.5146	0.6371
D31	10 × 8	**0.8905**	**0.7569**	0.7538	0.4071	0.4036	0.0743	0.7103	0.3039	0.4036	0.0743	0.8553	0.6637
Compound	9 × 3	**0.7736**	**0.7302**	0.5458	0.4155	0.5089	0.4526	0.5458	0.4155	0.5089	0.4526	**0.7736**	0.7223
Aggregation	7 × 5	**0.8905**	**0.8182**	0.8548	0.8162	0.6857	0.5519	0.5588	0.3786	0.6770	0.5445	0.8386	0.7544
Target	7 × 6	**0.9386**	**0.9702**	0.3741	0.3192	0.0000	0.0000	0.0242	0.0035	0.0000	0.0000	0.6520	0.6495
Atom	7 × 5	**0.9937**	**0.9975**	0.1229	0.0260	0.0000	0.0000	0.1229	0.0260	0.0000	0.0000	0.6279	0.5990
dim032	8 × 6	**1.0000**	**1.0000**	0.9083	0.7006	0.0000	0.0000	0.9083	0.7006	0.4380	0.1064	0.9841	0.9361
Synthetic Control	13 × 3	**0.8731**	**0.8028**	0.5243	0.2742	0.4605	0.2672	0.5243	0.2742	0.4605	0.2672	0.8063	0.6261
WDBC	15 × 2	0.5132	0.6367	0.5156	0.5815	0.0000	0.0000	0.5156	0.5815	0.0000	0.0000	0.5132	0.6367
Large SOM sizes													
Iris	32 × 8	0.4161	0.0583	0.7337	0.5681	0.7337	0.5681	0.7337	0.5681	0.7907	0.7323	0.7387	0.7282
Wine	16 × 16	0.4396	0.0902	0.0000	0.0000	0.7801	0.8040	0.0000	0.0000	**0.8336**	**0.8498**	0.8104	0.8203
Seeds	30 × 10	0.3887	0.0499	0.0183	0.0002	0.4727	0.3654	0.0663	0.0025	0.4727	0.3654	0.6954	0.6998
Chainlink	36 × 18	0.3326	0.0815	0.0000	0.0000	**1.0000**	**1.0000**	0.2215	0.2009	**1.0000**	**1.0000**	0.4652	0.2462
Hepta	18 × 16	0.7418	0.4797	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.4799	0.1952	0.0000	0.0000	**1.0000**	**1.0000**
Tetra	22 × 18	0.5543	0.1929	0.0000	0.0000	0.5484	0.3228	0.3805	0.2786	0.8120	0.7061	**1.0000**	**1.0000**
Spiral	20 × 18	0.6093	0.4228	0.0125	0.0052	0.3120	0.2155	0.0125	0.0052	0.2725	0.1308	0.3306	0.1437
R15	24 × 20	0.9013	0.8052	0.1659	0.0196	0.7425	0.2637	0.7425	0.2637	0.7425	0.2637	**0.9942**	**0.9928**
Path based	24 × 14	0.4608	0.1558	0.0000	0.0000	0.0932	−0.0014	0.0000	0.0000	0.4901	0.4157	0.5489	0.4650
Flame	20 × 16	0.2771	0.0349	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9269	0.9666	0.5510	0.4997
D31	38 × 30	0.9305	0.8734	0.0797	0.0044	0.6997	0.2636	0.8956	0.8107	0.7692	0.3258	0.9512	0.9209
Compound	34 × 12	0.6901	0.4330	0.5590	0.4237	0.8197	0.7629	0.7685	0.7524	0.8122	0.7572	0.7836	0.7293
Aggregation	28 × 20	0.9331	0.9012	0.0000	0.0000	0.8359	0.7338	0.0000	0.0000	0.8257	0.7260	0.8782	0.7759
Target	26 × 22	0.5582	0.6033	0.2302	0.0583	**0.9386**	**0.9702**	**0.9386**	**0.9702**	0.2792	0.1801	0.6644	0.6555
Atom	28 × 20	0.5134	0.5237	**1.0000**	**1.0000**	0.3693	0.3008	0.1465	0.0341	0.0000	0.0000	0.6221	0.5971
dim032	30 × 22	0.9832	0.9760	0.1555	0.0173	**1.0000**	**1.0000**	0.9984	0.9979	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Synthetic Control	50 × 10	0.6864	0.4689	0.0000	0.0000	0.5341	0.2838	0.2321	0.0763	0.6922	0.5199	**0.8030**	**0.6250**
WDBC	60 × 8	0.2838	0.0729	0.0000	0.0000	0.6028	0.7120	0.0000	0.0000	0.5584	0.6623	0.6142	0.7179

Note: Bold values indicate the best performance regarding comparable SOM sizes for each data set among the methods in Tables 2 and 3. Underlined values indicate the best performance over both SOM sizes.

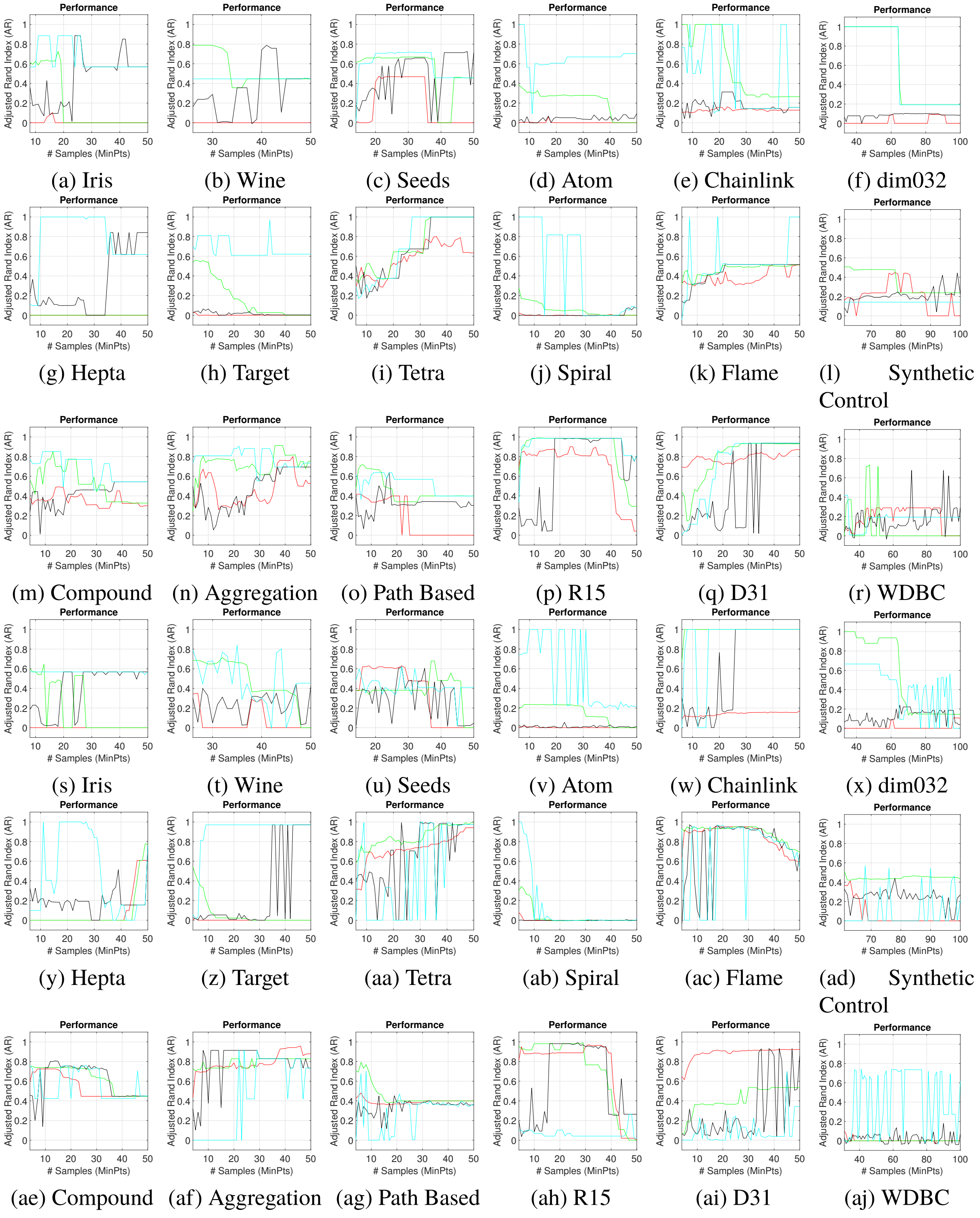
Table 3. Peak performance for IT-vis methods.

Data set	modified *k*-NN								standard *k*-NN							
	Stable Regions + Watershed				Otsu's Method + Watershed				Stable Regions + Watershed				Otsu's Method + Watershed			
	rCIP		rH*		rCIP		rH*		rCIP		rH*		rCIP		rH*	
	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR
Small SOM sizes																
Iris	**0.8465**	**0.8680**	0.7582	0.7302	0.0000	0.0000	0.4730	0.4202	0.7582	0.7302	0.7582	0.7302	0.7582	0.7302	0.7582	0.7302
Wine	**0.9016**	**0.9310**	**0.9016**	**0.9310**	0.0000	0.0000	**0.9016**	**0.9310**	0.7573	0.7269	0.6304	0.4739	0.5862	0.4536	0.6304	0.4739
Seeds	0.6907	0.6599	0.6537	0.6424	0.5326	0.4595	0.6459	0.5942	0.7122	**0.7051**	0.5942	0.4883	0.6219	0.5543	0.6907	0.6595
Chainlink	0.4786	0.4484	0.5151	0.2921	0.4259	0.3335	0.5341	0.4143	0.4458	0.4564	0.7889	0.8389	0.5009	0.6049	0.6542	0.6461
Hepta	0.9451	0.8411	**1.0000**	**1.0000**	0.0000	0.0000	0.3465	0.0990	0.9451	0.8411	0.8858	0.7221	0.5786	0.2315	0.5786	0.2315
Tetra	**0.9579**	**0.9669**	**0.9579**	**0.9669**	0.9452	0.9538	0.9452	0.9539	0.9452	0.9538	0.9452	0.9538	0.9402	0.9535	0.9452	0.9538
Spiral	0.1514	0.0671	0.2131	**0.1310**	0.0066	0.0032	0.0098	0.0041	0.0167	0.0074	0.0536	0.0211	0.0094	0.0056	0.0107	0.0045
R15	0.7553	0.4670	0.7296	0.4551	0.4554	0.1586	0.4864	0.1652	0.6808	0.3749	0.6622	0.3257	0.5329	0.2209	0.5194	0.2006
Path based	0.4948	0.4033	0.4910	0.4033	0.4186	0.3547	0.4910	0.4033	0.4910	0.3990	**0.5021**	**0.4145**	0.4910	0.3990	0.4910	0.3990
Flame	0.5453	0.6512	0.6047	0.6967	0.6185	0.7336	0.5453	0.6512	0.5453	0.6512	0.5453	0.6512	0.5453	0.6512	0.5453	0.6512
D31	0.8452	0.6055	0.8452	0.6055	0.6940	0.3196	0.6937	0.3169	0.6377	0.2493	0.8590	0.6273	0.6234	0.2334	0.5549	0.1604
Compound	**0.7736**	0.7223	**0.7736**	0.7223	0.7400	0.7150	**0.7736**	0.7223	0.7359	0.6959	0.7359	0.6959	0.5746	0.4771	0.5921	0.4526
Aggregation	0.8070	0.6770	0.8631	0.7923	0.6193	0.4346	0.7631	0.6811	0.8832	0.8080	0.8292	0.7474	0.7905	0.6994	0.8175	0.7223
Target	0.2979	0.1597	0.6698	0.6625	0.3153	0.3291	0.5282	0.5477	0.2097	0.0851	0.5921	0.6064	0.0304	0.0231	0.1331	0.0442
Atom	0.2213	0.1853	0.5035	0.4649	0.0000	0.0000	0.1471	0.0396	0.4575	0.3634	0.2260	0.0982	0.0000	0.0000	0.0000	0.0000
dim032	0.3660	0.0944	0.9333	0.7813	0.0000	0.0000	0.7250	0.3543	0.4975	0.1387	0.8772	0.6030	0.0000	0.0000	0.6991	0.3419
Synthetic Control	0.7408	0.5582	0.7216	0.4762	0.7408	0.5582	0.6993	0.5175	0.6616	0.4811	0.5243	0.2742	0.4308	0.2218	0.7408	0.5582
WDBC	0.3390	0.2493	0.5132	0.6367	0.3390	0.2435	0.0000	0.0000	0.2457	0.0607	**0.6160**	**0.6788**	0.0000	0.0000	0.0000	0.0000
Large SOM sizes																
Iris	**0.8705**	**0.8858**	**0.8705**	**0.8858**	0.2599	0.1045	0.7526	0.7264	0.7337	0.5681	0.7337	0.5681	0.0000	0.0000	0.7337	0.6435
Wine	0.7844	0.7882	0.5862	0.4536	0.0000	0.0000	0.7844	0.7882	0.5341	0.4447	0.8252	0.8368	0.4781	0.3504	0.7488	0.7127
Seeds	**0.7292**	**0.7247**	0.7188	0.7180	0.5417	0.4725	0.6887	0.6688	0.6382	0.6158	0.6436	0.6233	0.6455	0.6297	0.6795	0.6787
Chainlink	0.4211	0.3130	**1.0000**	**1.0000**	0.4147	0.1621	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.4135	0.1731	**1.0000**	**1.0000**
Hepta	0.9451	0.8411	**1.0000**	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.8356	0.7603	**1.0000**	**1.0000**	0.8010	0.6092	0.8569	0.7784
Tetra	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.8180	0.8015	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9262	0.9411	0.9899	0.9933
Spiral	0.0858	0.0829	**1.0000**	**1.0000**	0.0727	0.0233	0.4133	0.2781	0.0371	0.0057	**1.0000**	**1.0000**	0.1501	0.0787	0.4482	0.3446
R15	0.9893	0.9857	0.9893	0.9857	0.9490	0.9001	0.9913	0.9892	0.9942	0.9928	0.7425	0.2637	0.9806	0.9715	**0.9942**	**0.9928**
Path based	0.7373	0.6366	0.7373	0.6366	0.4913	0.4186	0.7339	0.7163	0.5379	0.4518	0.7107	0.7023	0.5238	0.4802	**0.7918**	**0.7931**
Flame	0.6715	0.5243	**1.0000**	**1.0000**	0.6491	0.5155	0.6491	0.5155	0.9313	0.9714	0.9355	0.9666	0.8991	0.9501	0.8994	0.9502
D31	**0.9580**	**0.9370**	**0.9580**	**0.9370**	0.9302	0.8734	0.9564	0.9337	0.9551	0.9337	0.9063	0.6990	0.9500	0.9237	0.8624	0.5367
Compound	0.7252	0.5434	0.8847	**0.8519**	0.6710	0.4929	**0.8949**	0.8487	0.8466	0.8073	0.8136	0.7607	0.7732	0.7248	0.8042	0.7531
Aggregation	0.8465	0.6936	0.9367	0.9067	0.8269	0.7999	0.9134	0.9126	0.9538	0.9139	0.9538	0.9139	**0.9634**	**0.9580**	0.9067	0.8286
Target	0.1532	0.0639	**0.9386**	**0.9702**	0.0753	0.0326	0.4997	0.5604	**0.9386**	**0.9702**	**0.9386**	**0.9702**	0.0854	0.0227	0.4844	0.5424
Atom	0.1357	0.0929	**1.0000**	**1.0000**	0.0526	0.0587	0.4005	0.3855	0.0822	0.0552	**1.0000**	**1.0000**	0.0224	0.0152	0.3381	0.2367
dim032	0.3396	0.1024	**1.0000**	**1.0000**	0.3254	0.0972	**1.0000**	**1.0000**	0.5613	0.2398	0.9153	0.6654	0.3570	0.1093	**1.0000**	**1.0000**
Synthetic Control	0.6052	0.4437	0.4019	0.1418	0.6121	0.4490	0.6739	0.5061	0.6035	0.4396	0.7215	0.5707	0.5706	0.4139	0.6791	0.5116
WDBC	0.5562	0.6771	0.4164	0.4198	0.2851	0.2935	0.6594	0.7357	0.2492	0.1798	**0.6730**	**0.7420**	0.1838	0.0999	0.0000	0.0000

Note: Bold values indicate the best performance regarding comparable SOM sizes for each data set among the methods in Tables 2 and 3. Underlined values indicate the best performance over both SOM sizes.

tions using large SOMs, which is the highest count among the clustering methods for both SOM sizes; specifically, 11 using the rH*-vis generated with the modified *k*-NN in Table 3 corresponds to the best overall performance; again, the highest count among the clustering methods for all SOM sizes. Regarding 6 out of the 7 remaining data sets, their best solutions could also be retrieved either using a different IT-vis (5) or different SOM size (1).

The appeal of applying the watershed algorithm is the generation of one pixel-wide boundaries between clusters. This is less significant when using rH*-vis with the modified *k*-NN, as the boundaries between the clusters are already very thin and thus already very similar to the output of the watershed algorithm (cf. Figure 12). Thus, Table 4 depicts



(a) Iris (b) Wine (c) Seeds (d) Atom (e) Chainlink (f) dim032

(g) Hepta (h) Target (i) Tetra (j) Spiral (k) Flame (l) Synthetic Control

(m) Compound (n) Aggregation (o) Path Based (p) R15 (q) D31 (r) WDBC

(s) Iris (t) Wine (u) Seeds (v) Atom (w) Chainlink (x) dim032

(y) Hepta (z) Target (aa) Tetra (ab) Spiral (ac) Flame (ad) Synthetic Control

(ae) Compound (af) Aggregation (ag) Path Based (ah) R15 (ai) D31 (aj) WDBC

Figure 13. AR index for different values of the minimum number of points ($MinPts$) of the subset $\mathcal{H}$ generated using the modified k-NN (Gokcay & Principe, 2002) (first three rows) and the standard k-NN (Duda et al., 2000) (last three rows) for all data sets. In this sweep analysis, the clustering performed using Otsu's method is represented in red and green lines for the rCIP-vis and rH*-vis respectively. The clustering performed using the stable regions approach is represented in black and light blue lines for the rCIP-vis and rH*-vis, respectively.

Table 4. Clustering results for large SOM sizes. Peak performance for IT-vis methods without the watershed algorithm.

Data set	modified *k*-NN								standard *k*-NN							
	Stable Regions				Otsu's Method				Stable Regions				Otsu's Method			
	rCIP		rH*		rCIP		rH*		rCIP		rH*		rCIP		rH*	
	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR	NMI	AR
Iris	0.8705	0.8858	0.8705	0.8858	0.2552	0.1463	0.7304	0.6821	0.7337	0.5681	0.7337	0.5681	0.0000	0.0000	0.7337	0.6100
Wine	0.7728	0.7726	0.5748	0.4464	0.0000	0.0000	0.7844	0.7882	0.6125	0.4687	0.8330	0.8483	0.6281	0.4766	0.7872	0.7847
Seeds	0.7029	0.7024	0.7188	0.7180	0.5685	0.4913	0.6820	0.6795	0.6820	0.6795	0.6886	0.6895	0.6886	0.6895	0.6618	0.6500
Chainlink	0.4211	0.1709	1.0000	1.0000	0.4655	0.2148	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.4667	0.2091	1.0000	1.0000
Hepta	0.9451	0.8411	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.8965	0.7926	1.0000	1.0000	0.8604	0.6183	0.8965	0.7926
Tetra	1.0000	1.0000	1.0000	1.0000	0.8991	0.9027	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9899	0.9933	1.0000	1.0000
Spiral	0.1193	0.1133	1.0000	1.0000	0.0813	0.0472	0.5040	0.3480	0.0198	0.0000	1.0000	1.0000	0.2430	0.1205	0.5246	0.4091
R15	0.9913	0.9892	0.9893	0.9857	0.9648	0.9097	0.9913	0.9892	0.9942	0.9928	0.7425	0.2637	0.9914	0.9892	0.9942	0.9928
Path based	0.7373	0.6366	0.7373	0.6366	0.5286	0.4659	0.7412	0.7184	0.5489	0.4650	0.7239	0.7125	0.5986	0.5477	0.8187	0.8205
Flame	0.6715	0.5243	1.0000	1.0000	0.6493	0.5480	0.6681	0.5156	0.9269	0.9666	0.9635	0.9833	0.9269	0.9666	0.9269	0.9666
D31	0.9563	0.9339	0.9563	0.9339	0.9479	0.9162	0.9611	0.9427	0.9574	0.9377	0.9062	0.6992	0.9579	0.9382	0.8629	0.5373
Compound	0.7252	0.5434	0.8846	0.8552	0.7023	0.5070	0.9003	0.8452	0.8598	0.8200	0.8282	0.7653	0.8103	0.7531	0.8221	0.7630
Aggregation	0.8475	0.6953	0.9367	0.9055	0.8729	0.8457	0.9195	0.9168	0.9538	0.9139	0.9538	0.9139	0.9915	0.9949	0.9067	0.8286
Target	0.1527	0.0401	0.9386	0.9702	0.1312	0.0492	0.5541	0.5921	0.9386	0.9702	0.9386	0.9702	0.0000	0.0000	0.5401	0.5845
Atom	0.0619	0.0501	1.0000	1.0000	0.0333	0.0342	0.4469	0.4323	0.0230	0.0145	1.0000	1.0000	0.0074	0.0068	0.3442	0.2433
dim032	0.3660	0.0944	1.0000	1.0000	0.3660	0.0944	1.0000	1.0000	0.5576	0.2165	0.9153	0.6654	0.3660	0.0944	1.0000	1.0000
Synthetic Control	0.6593	0.4773	0.4019	0.1418	0.7084	0.5342	0.7198	0.5401	0.3830	0.2114	0.7548	0.5861	0.6063	0.4509	0.7272	0.5615
WDBC	0.3899	0.2889	0.4238	0.4296	0.4457	0.3577	0.4804	0.5233	0.0920	0.1279	0.6628	0.7420	0.1878	0.0329	0.0000	0.0000

the performance of clustering large SOMs without the watershed algorithm, in order to analyze its influence regarding the segmentation of IT-vis. Noticeably, the vast majority of the clustering results are similar, and in some cases superior, to those reported in Table 3.

Additionally, in order to analyze the sensitivity of large SOMs regarding $MinPts$, this parameter was varied in the same ranges mentioned previously and the behavior of the adjusted Rand index (AR) of the partitions is illustrated for all data sets in Figure 13 for both *k*-NN methods. As expected, there was no universal value for the minimum number of points parameter ($MinPts$) that can be used for all the data sets to provide the best performance, regardless of the IT-vis type used. For many data sets, an interval or plateau in which partitioning the SOM using one of the IT-vis provides a comparable or superior performance regarding the other clustering methods (cf. Figure 13) can be observed. Additionally, other data sets had good performances for a wide range of values for which $MinPts$ may be effectively used. The best results from Figure 13 are the ones summarized in Table 3. In general, clustering the rH*-vis led to the majority of the best performances using stable regions followed by Otsus's method (cf. light blue and green curves in Figure 13 and Table 3). Specifically, clustering the rH*-vis generated with the modified k-NN is recommended, under which most of the data sets had better performances.

7. CONCLUSION

This work presented information-theoretic-cluster visualizations (IT-vis) for self-organizing maps. The visualizations consist of a gray-level images that follows the structure of the unified distance matrix and displays either Renyi's (representative) cross-information potential or a modified version of Renyi's (representative) cross-entropy: rCIP-vis or CIP-vis and rH*-vis or H*-vis, respectively. The visualizations require only one input parameter ($MinPts$), which must be set by the user. It is the number of samples to compute the statistics (mean and covariance matrix) of the subsets $\mathcal{H}$ of the data set associated with each neuron. This is not a considerable challenge, since performance appears to be robust with respect to $MinPts$.

In order to create the subsets $\mathcal{H}$, the standard k-NN and a modified k-NN were used. In the first case, the visualization is smoother as the subset $\mathcal{H}_i$ is taken inside a hypersphere centered at neuron i. On the other hand, the second case is able to highlight and sharply depict the differences among clusters and the sub-regions within them due to the fact that the statistics are computed using samples that better follow the data structure at the region in which a given neuron is located, with the granularity controlled by the $MinPts$ parameter.

As expected, computing the representative IT-vis is much faster than their original counterparts (CIP-vis and H*-vis), while providing the same level of detail to the visualization. Specifically, the rH*-vis, which displays entropy information, is much more suitable for visualization as it compresses the range of high values of rCIP and unveils subtleties in the regions with low CIP values by expanding its range. From visualization experiments with several data sets and SOMs ranging from densely-matched to sparsely-matched maps, rH*-vis appears to be robust with respect to the SOM dimensions.

The IT-vis were compared and contrasted with the standard U-matrix and CONNvis2. In most cases, they revealed visually clearer cluster information. For instance, compared to the U-matrix, most rH*-vis provided a portrayal of data sets for which the cluster's boundaries were visually sharper and thinner. As expected, the visualization power of CONNvis2 is better harvested for smaller map sizes as it takes advantage of the vector quantization and larger sample to neuron ratio. On the other hand, the resolution of rH*-vis increases with the size of the map, at an additional computational cost. CONNvis2, however, has the unique feature of enabling the visualization of topology violations.

Regarding the clustering task, among the methods compared, the results showed that for the majority of the data sets, superior or comparable peak performances were achieved by using the presented IT-vis for large SOMs; specifically, the rH*-vis generated with the modified *k*-NN clustered via the stable regions approach. The CONN linkage obtained the overall best performance for small SOMs. Similarly to the U-matrix, the clustering methodology based on the digital image processing techniques of Otsu's global thresholding method, stable regions and the watershed algorithm, when employed over the IT-vis, also detects the number of clusters and clusters of complex geometric shapes. Specifically, the rH*-vis generated with the modified *k*-NN clustered by the stable regions approach has the additional benefit of, in many cases, providing larger lifetimes and thinner boundaries without any image pre-processing. Furthermore, for large SOMs, the application of the watershed algorithm appears to be an optional step as the performance was minimally affected.

Finally, rH*-vis is recommended for both visualization and clustering, along with the use of large SOMs, as resolution increases with size. Concerning clustering, the stable regions thresholding method is recommended, as it led to the best results for the majority of the data sets. Finally, it is preferable for the rH*-vis to be generated with the modified *k*-NN; however, the question of when to choose one mode over the other is ultimately application-dependent and user-defined. Naturally, superior performance may be achieved by applying more thorough pre-processing to the images and by using more complex image segmentation algorithms, such as local thresholding, information-theoretic based image segmentation, or simply by evaluating the CEF for the partitions of each segmented image and selecting the one that minimizes it (or using a suitable validity index). In combination, it is expected that these observations will assist in improved visualization for cluster analysis.

ACKNOWLEDGEMENTS

Support from the National Science Foundation, the Missouri University of Science and Technology Intelligent Systems Center, and the Mary K. Finley Endowment is gratefully acknowledged. Leonardo Enzo Brito da Silva was supported by the CAPES Foundation, Ministry of Education of Brazil, under grant number BEX 13494/13-9.

REFERENCES

Araújo, D., Neto, A. D., & Martins, A. (2013a). Information-theoretic clustering: A representative and evolutionary approach. *Expert Systems with Applications*, *40*, 4190–4205.

Araújo, D., Neto, A. D., & Martins, A. (2013b). Representative cross information potential clustering. *Pattern Recognition Letters*, *34*, 2181 – 2191.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Bezdek, J. C., Hathaway, R. J., & Huband, J. M. (2007). Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices. *IEEE Transactions on Fuzzy Systems*, *15*, 890–903.

Brito da Silva, L. E., & Costa, J. A. F. (2013a). A Gravitational Approach for Enhancing Cluster Visualization in Self-Organizing Maps. In *The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE'13)* (pp. 48–54).

Brito da Silva, L. E., & Costa, J. A. F. (2013b). Clustering, Noise Reduction and Visualization Using Features Extracted from the Self-Organizing Map. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, & X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning - IDEAL 2013* (pp. 242–251). Springer Berlin Heidelberg volume 8206 of *Lecture Notes in Computer Science*.

Brito da Silva, L. E., & Costa, J. A. F. (2013c). Clustering the Self-Organizing Map Based on the Neurons' Associated Pattern Sets. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence* (pp. 7–14).

Brito da Silva, L. E., & Costa, J. A. F. (2013d). Clustering the self-organizing map through the identification of core neuron regions. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Brito da Silva, L. E., & Ferreira Costa, J. A. (2014). Clustering of the self-organizing map using particle swarm optimization and validity indices. In *The 2014 International Joint Conference on Neural Networks (IJCNN'14)* (pp. 3798–3806).

Brito da Silva, L. E., & Wunsch II, D. C. (2017). SOM IT-vis. URL: `https://git.mst.edu/acil-group/som-it-vis`.

Chalasani, R., & Principe, J. (2010). Self organizing maps with the correntropy induced metric. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–6).

Chalasani, R., & Principe, J. C. (2015). Self-organizing maps with information theoretic learning. *Neurocomputing*, *147*, 3 – 14. Advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012).

Chang, H., & Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*, 191–203.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. (2010). Complete Gradient Clustering Algorithm for Features Analysis of X-Ray Images. In E. Pietka, & J. Kawa (Eds.), *Information Technologies in Biomedicine* (pp. 15–24). Berlin, Heidelberg: Springer Berlin Heidelberg.

Costa, J. A. F., & Netto, M. L. A. (1999). Estimating the number of clusters in multivariate data by self-organizing maps. *International Journal of Neural Systems*, *9*, 195–202.

Costa, J. A. F., & Netto, M. L. A. (2001). Clustering of complex shaped data sets via Kohonen maps and mathematical morphology. In B. Dasarathy (Ed.), *Proceedings of the SPIE, Data Mining and Knowledge Discovery* (pp. 16–27). volume 4384.

Costa, J. A. F., & Yin, H. (2010). Gradient-based SOM clustering and visualisation methods. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. (2nd ed.). John Wiley & Sons.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Fränti, P., Virmajoki, O., & Hautamaki, V. (2006). Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 1875–1881.

Fränti, Pasi et al. (2015). Clustering datasets. URL: `http://cs.uef.fi/sipu/datasets/` accessed on May 4, 2017.

Fu, L., & Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, *8*, 1–15.

Garcia, V., Debreuve, E., & Barlaud, M. (2008). Fast k Nearest Neighbor Search using GPU. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08)* (pp. 1–6).

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*, 1–30.

Gokcay, E., & Principe, J. (2000). A new clustering evaluation function using renyi's information potential. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on* (pp. 3490–3493). volume 6.

Gokcay, E., & Principe, J. C. (2002). Information theoretic clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*, 158–171.

Gonçalves, M. L., de Andrade Netto, M. L., Costa, J. A. F., & Zullo, J. (2006). Data Clustering using Self-Organizing Maps segmented by Mathematic Morphology and Simplified Cluster Validity Indexes: an application in remotely sensed images. In *The 2006 IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 4421–4428).

Gonzalez, R. C., & Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc.

Haralick, R. M., & Shapiro, L. G. (1992). *Computer and Robot Vision*. Addison-Wesley.

Hastie, T., & Stuetzle, W. (1989). Principal Curves. *Journal of the American Statistical Association*, *84*, 502–516.

Huband, J. M., Bezdek, J. C., & Hathaway, R. J. (2004). Revised Visual Assessment of (Cluster) Tendency (reVAT). In *IEEE Annual Meeting of the Fuzzy Information Processing Society (NAFIPS'04)* (pp. 101–104). volume 1.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Kamimura, R. (2013). Similarity interaction in information-theoretic self-organizing maps. *International Journal of General Systems*, *42*, 239–267.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kohonen, T. (2001). *Self-Organizing Maps* volume 30 of *Springer Series in Information Sciences*. (3rd ed.). Springer-Verlag Berlin Heidelberg New York.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks*, *37*, 52 – 65.

LeBlanc, M., & Tibshirani, R. (1994). Adaptive Principal Surfaces. *Journal of the American Statistical Association*, *89*, 53–64.

Liu, W., Pokharel, P. P., & Príncipe, J. C. (2007). Correntropy: Properties and Applications in Non-Gaussian Signal Processing. *IEEE Transactions on Signal Processing*, *55*, 5286–5298.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). University of California Press volume 1.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Manukyan, N., Eppstein, M. J., & Rizzo, D. M. (2012). Data-driven cluster reinforcement and visualization in sparsely-matched self-organizing maps. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 846–852.

Martinetz, T., & Schulten, K. (1994). Topology representing networks. *Neural Networks*, *7*, 507 – 522.

Martins, A., Doria Neto, A. D., & de Melo, J. D. (2003). Neural networks applied to classification of data based on Mahalanobis metrics. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3071–3076). volume 4.

Martins, A., Doria Neto, A. D., de Melo, J. D., & Costa, J. A. F. (2004a). Clustering using neural networks and Kullback-Leibler divergency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2813–2817). volume 4.

Martins, A., Neto, A., & de Melo, J. (2004b). Comparison Between Mahalanobis Distance and Kullback-Leibler Divergence in Clustering Analisys. In *WSEAS Transactions on Systems* (pp. 501–505). volume 3.

Merkl, D., & Rauber, A. (1997). Alternative Ways for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'97* (pp. 106–111).

Meyer, F. (1994). Topographic Distance and Watershed Lines. *Signal Process.*, *38*, 113–125.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, *9*, 62–66.

Pampalk, E., Rauber, A., & Merkl, D. (2002). Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In J. R. Dorronsoro (Ed.), *Artificial Neural Networks - ICANN 2002* (pp. 871–876). Springer Berlin Heidelberg volume 2415 of *Lecture Notes in Computer Science*.

Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. (1st ed.). Springer Publishing Company, Incorporated.

Rao, S., Martins, A., & Principe, J. C. (2009). Mean shift: An information theoretic perspective. *Pattern Recognition Letters*, *30*, 222 – 230.

Rényi, A. (1961). On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics* (pp. 547–561). Berkeley, Calif.: University of California Press.

Sammon, J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, *C-18*, 401–409.

Taşdemir, K. (2010). Graph based representations of density distribution and distances for self-organizing maps. *IEEE Transactions on Neural Networks*, *21*, 520–526.

Taşdemir, K. (2012). Vector quantization based approximate spectral clustering of large datasets. *Pattern Recognition*, *45*, 3034 – 3044.

Taşdemir, K., & Merényi, E. (2005). Considering topology in the clustering of Self-Organizing Maps. In *WSOM 2005 - 5th Workshop on Self-Organizing Maps* (pp. 439–446).

Taşdemir, K., & Merényi, E. (2007). A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. In *The 2007 International Joint Conference on Neural Networks (IJCNN)* (pp. 2205–2211).

Taşdemir, K., & Merényi, E. (2009). Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *20*, 549–562.

Taşdemir, K., & Merényi, E. (2011). A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*, 1039–1053.

Taşdemir, K., Milenov, P., & Tapsall, B. (2011). Topology-based hierarchical clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *22*, 474–485.

Tapan, M. S. Z., & Siong, T. C. (2008). AC-ViSOM: Hybridising the modified Adaptive Coordinate (AC) and ViSOM for data visualization. In *2008 International Symposium on Information Technology* (pp. 1–8). volume 3.

Tapan, M. S. Z., & Teh, C. S. (2007). Hybridization of Learning Vector Quantization (LVQ) and Adaptive Coordinates (AC) for data classification and visualization. In *International Conference on Intelligent and Advanced Systems (ICIAS'07)* (pp. 505–510).

Teh, C. S., & Sarwar, Z. T. M. (2008). A hybrid supervised ANN for classification and data visualization. In *The 2008 IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 555–562).

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification. In O. Opitz, B. Lausen, & R. Klar (Eds.), *Information and Classification: Concepts, Methods and Applications* (pp. 307–313). Springer Berlin Heidelberg.

Ultsch, A. (2003). Maps for the Visualization of high-dimensional Data Spaces. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)* (pp. 225–230).

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Ultsch, A., & Siemon, H. P. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. International Neural Networks Conference (INNC)* (pp. 305–308).

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1273–1280.

Vesanto, J. (1999). SOM-based data visualization methods. *Intelligent Data Analysis*, *3*, 111 – 126.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*, 586–600.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (1999). Self-Organizing Map in Matlab: the SOM Toolbox. In *Proceedings of the Matlab DSP Conference* (pp. 35–40).

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

Wang, L., Geng, X., Bezdek, J., Leckie, C., & Kotagiri, R. (2008). SpecVAT: Enhanced Visual Cluster Analysis. In *The 2008 IEEE International Conference on Data Mining (ICDM)* (pp. 638–647).

Wang, L., Nguyen, U. T. V., Bezdek, J. C., Leckie, C. A., & Ramamohanarao, K. (2010). iVAT and aVAT: Enhanced Visual Analysis for Cluster Tendency Assessment. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'10)* (pp. 16–27).

Wang, X. (2011). A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality. In *The 2011 International Joint Conference on Neural Networks (IJCNN)* (pp. 1293–1299).

Wu, S., & Chow, T. W. S. (2005). PRSOM: a new visualization method by hybridizing multidimensional scaling and self-organizing map. *IEEE Transactions on Neural Networks*, *16*, 1362–1380.

Xu, L., Chow, T. W. S., & Ma, E. W. M. (2015). Topology-Based Clustering Using Polar Self-Organizing Map. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 798–808.

Xu, L., Xu, Y., & Chow, T. W. S. (2010). PolSOM: A new method for multidimensional data visualization. *Pattern Recognition*, *43*, 1668 – 1675.

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2010). Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, *3*, 120–154.

Xu, Y., Xu, L., & Chow, T. W. S. (2011). PPoSOM: A new variant of PolSOM by using probabilistic assignment for multidimensional data visualization. *Neurocomputing*, *74*, 2018 – 2027.

Yin, H. (2001). Visualisation Induced SOM (ViSOM). In N. Allinson, H. Yin, L. Allinson, & J. Slack (Eds.), *Advances in Self-Organising Maps* (pp. 81–88). Springer London.

Yin, H. (2002a). Data visualisation and manifold mapping using the ViSOM. *Neural Networks*, *15*, 1005 – 1016.

Yin, H. (2002b). ViSOM - a novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks*, *13*, 237–243.

Yin, H. (2008). On multidimensional scaling and the embedding of self-organising maps. *Neural Networks*, *21*, 160 – 169.

Zahn, C. T. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, *C-20*, 68–86.

Zhang, X., & Li, Y. (1993). Self-organizing map as a new method for clustering and data analysis. In *The 1993 International Joint Conference on Neural Networks (IJCNN)* (pp. 2448–2451). volume 3.

VII. INCREMENTAL CLUSTER VALIDITY INDICES FOR HARD PARTITIONS: EXTENSIONS AND COMPARATIVE STUDY

Leonardo Enzo Brito da Silva[1,2], Niklas M. Melton[1] and Donald C. Wunsch II[1]

[1]Applied Computational Intelligence Lab., Dept. of Electrical and Computer Engineering,

Missouri University of Science and Technology, Rolla, MO 65409 USA.

[2]CAPES Foundation, Ministry of Education of Brazil, Brasília, DF 70040-020 Brazil.

Email: leonardoenzo@ieee.org

ABSTRACT

Validation is one of the most important aspects of clustering, but most existing approaches have required batch calculation. Recently, interest has grown in providing online alternatives for data stream applications. This paper extends the incremental cluster validity index (iCVI) family by presenting incremental versions of Calinski-Harabasz (iCH), Pakhira-Bandyopadhyay-Maulik (iPBM), WB index (iWB), Silhouette (iSIL), Negentropy Increment (iNI), Representative Cross Information Potential (irCIP) and Representative Cross Entropy (irH), and Conn_Index (iConn_Index). The sum-of-squares-based iCVIs were realized by incorporating a recently developed incremental update for the clusters' compactness, whereas the information-theoretic-based iCVIs were realized by incorporating a classic incremental update for the clusters' covariance matrices. The multi-prototype representation required by the graph-based iCVI was realized by customizing a fuzzy ART-based neural network and incrementally updating a matrix of connections between prototypes. This paper also provides a thorough comparative study on the effect of correct, under- and over-partitioning on the behavior of these iCVIs, the Partition Separation (PS) index and four recently developed iCVIs: incremental Xie-Beni (iXB), incremental Davies-Bouldin (iDB), and incremental generalized Dunn's indices 43 and 53 (iGD43 and iGD53).

Experiments were carried out using a framework as clustering algorithm agnostic as possible and results on synthetic benchmark data sets showed that while evidence of most under-partitioning cases could be inferred from the behaviors of the majority of these iCVIs, over-partitioning was found to be a more challenging problem detected by a smaller fraction of them. Interestingly, over-partition, rather then under-partition, was more prominently detected on the real world benchmark data sets experiment with in this study. The expansion of iCVIs provides significant novel opportunities for assessing and interpreting in real-time the results of unsupervised lifelong learning, in which samples cannot be reprocessed due to memory and/or application constraints.

Keywords: Clustering, Validation, Incremental Cluster Validity Index (iCVI), Adaptive Resonance Theory (ART), incremental (online) clustering algorithms, data streams.

1. INTRODUCTION

Cluster validation (Gordon, 1998) is a fundamental topic in cluster analysis because it is crucial to assess the quality of partitions detected by clustering algorithms since no class label information is available. Moreover, different clustering solutions may be found by distinct algorithms, or even by the same algorithm subjected to different hyper-parameters or a different input presentation order (Brito da Silva & Wunsch II, 2018a; Xu et al., 2012). *Cluster validity indices* (CVIs) function as evaluators of such solutions by computing some cluster quality measure based on (i) the degree of agreement between the output and the reference partitions (*external* CVIs), or (ii) the data itself and the output partition information (*internal* CVIs). Numerous examples of such criteria have been presented in the literature to evaluate partitions in offline mode; for comprehensive reviews and experimental studies the interested reader may refer to (Arbelaitz et al., 2013; Bezdek et al., 1997; Dimitriadou et al., 2002; Dubes & Jain, 1979; Halkidi et al., 2002a,b; Hämäläinen et al., 2017; Milligan & Cooper, 1985; Vendramin et al., 2010; Vinh et al., 2010; Wang & Zhang, 2007; Xu & Wunsch II, 2005; Xu & Wunsch II, 2009).

Recently, *incremental cluster validity indices* (iCVIs) have been developed to track the effectiveness of online clustering methods over data streams (Ibrahim et al., 2018a,b; Moshtaghi et al., 2018; Moshtaghi et al., 2019). These are online versions of sum-of-squares (SS) based internal CVIs (Zhao & Fränti, 2014), which typically exhibit a trade-off between measures of compactness (a.k.a. dispersion or within-cluster scatter) and isolation (a.k.a. between-cluster separation) (Xu et al., 2012; Zhao & Fränti, 2014). To enable cluster validation in online applications, a recursive formulation of compactness was introduced in (Moshtaghi et al., 2018; Moshtaghi et al., 2019). This strategy has been used to develop incremental versions of Davies-Bouldin (Davies & Bouldin, 1979) (iDB) and Xie-Beni (Xie & Beni, 1991) (iXB) in (Moshtaghi et al., 2018; Moshtaghi et al., 2019) as well as incremental versions of two generalized Dunn's indices (Bezdek & Pal, 1998) (iGDs) in (Ibrahim et al., 2019). Particularly, the behavior of iXB and iDB were analyzed in both accurately and poorly partitioned data sets in (Moshtaghi et al., 2018; Moshtaghi et al., 2019), whereas the studies in (Ibrahim et al., 2018a,b) only investigate the iDB's behavior in cases where the MU streaming clustering (MUSC) (Ibrahim et al., 2016) accurately detected the structures present in the data. In this context, the contributions of this work are two-fold:

1. Presenting 7 additional iCVIs. The incremental versions of Calinski-Harabasz (Caliński & Harabasz, 1974), WB index (Zhao et al., 2009), Pakhira-Bandyopadhyay-Maulik (Pakhira et al., 2004), and Silhouette (Rousseeuw, 1987) were realized by employing the incremental update of compactness developed in (Moshtaghi et al., 2018; Moshtaghi et al., 2019). The incremental versions of Negentropy Increment (Lago-Fernández & Corbacho, 2009; Lago-Fernández & Corbacho, 2010), Representative Cross Information Potential and Representative Cross Entropy (Araújo et al., 2013a,b) were realized using the incremental update of covariance matrices (Duda et al., 2000). Finally, the incremental version of the Conn_Index (Taşdemir & Merényi, 2007, 2011) was realized by storing co-activation counts of multiple prototypes generated using

fuzzy adaptive resonance theory (ART)-based models (Carpenter et al., 1992, 1991). The latter were chosen for their simple parameterization of quantization granularity and other useful properties (Brito da Silva et al., 2019; Wunsch II, 2009).

2. Performing a comparative study among 13 iCVIs in cases of correct, under- and over-partitioning on synthetic and real world benchmark data sets. It is not the focus of this study to contrast the iCVIs' behavior associated with specific online clustering algorithms and their dynamics. Therefore, to explore such scenarios, a framework as clustering algorithm agnostic as possible was used to define the data partitions.

To the best of our knowledge, this work provides the first comprehensive and systematic comparative study on iCVIs. The remainder of this paper is structured as follows: Section 2, provides a brief review of CVIs, iCVIs and ART; Section 3 presents this work's extensions of several other CVIs to the incremental family; Section 4 details the set-up used in the numerical experiments; Section 5 describes and discusses the results; Section 6 compares batch and incremental versions of the Conn_Index; and Section 7 summarizes this paper's findings.

2. BACKGROUND AND RELATED WORK

This section provides an overview of CVIs, iCVIs and ART neural networks used in this study.

2.1. BATCH CLUSTER VALIDITY INDICES (CVIS)

Consider a data set $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ and its hard partition $\Omega = \{\omega_i\}_{i=1}^{k}$ of k disjointed clusters ω_i, such that $\bigcup\limits_{i=1}^{k} \omega_i = \boldsymbol{X}$. In the following CVI overview, $\boldsymbol{v}_i$ is cluster ω_i's prototype (centroid) defined as

$$\boldsymbol{v}_i = \frac{1}{n_i} \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \boldsymbol{x}_j, \tag{1}$$

k is the number of clusters, d is the dimensionality of the data ($\boldsymbol{x}_i \in \mathbb{R}^d$), and N and n_i are the cardinalities of a data set and cluster ω_i, respectively. Additionally, the data geometric center is given by

$$\boldsymbol{\mu}_{data} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i, \tag{2}$$

and the compactness of cluster ω_i with respect to point $\boldsymbol{z}$ is

$$CP_q^p(\boldsymbol{z}, \omega_i) = \sum_{j=1}^{n_i} \|\boldsymbol{x}_j - \boldsymbol{z}\|_q^p, \ \boldsymbol{x}_j \in \omega_i, \tag{3}$$

where $\| \cdot \|_q^p$ is the ℓ_q norm to the p^{th} power.

2.1.1. Calinski-Harabasz (CH). The CH index (Caliński & Harabasz, 1974) is defined as:

$$CH = \frac{BGSS/(k-1)}{WGSS/(N-k)}, \tag{4}$$

where the between group sum of squares (BGSS) and within group sum of squares (WGSS) are computed as:

$$WGSS = \sum_{i=1}^{k} CP_2^2(\boldsymbol{v}_i, \omega_i), \tag{5}$$

$$BGSS = \sum_{i=1}^{k} n_i \|\boldsymbol{v}_i - \boldsymbol{\mu}_{data}\|_2^2, \tag{6}$$

This is an optimization-like criterion (Vendramin et al., 2010) such that larger values of CH indicate better clustering solutions.

2.1.2. WB-Index (WB). The WB index (Zhao et al., 2009) is related to CH as discussed in (Zhao & Fränti, 2014) and is given by:

$$WB = k\frac{WGSS}{BGSS}. \tag{7}$$

Smaller values of WB suggest better data partition quality.

2.1.3. Davies-Bouldin (DB). The DB index (Davies & Bouldin, 1979) averages the similarities R of each cluster i with respect to its maximally similar cluster $j \neq i$:

$$DB = \frac{1}{k}\sum_{i=1}^{k} R_i, \tag{8}$$

where

$$R_i = \max_{i \neq j}\left(\frac{S_i + S_j}{M_{i,j}}\right), \tag{9}$$

$$S_l = \left[\frac{1}{n_l}\sum_{\substack{m=1 \\ \boldsymbol{x}_m \in \omega_l}}^{n_l} \|\boldsymbol{x}_m - \boldsymbol{v}_l\|^q\right]^{\frac{1}{q}}, \ l = \{1, \ldots, k\}, \tag{10}$$

$$M_{i,j} = \left[\sum_{t=1}^{d} |v_{it} - v_{jt}|^p\right]^{\frac{1}{p}}, \ p \geq 1. \tag{11}$$

The variables (p, q) are user-defined parameters, and S_l and $M_{i,j}$ (Minkowski metric) measure compactness and separation, respectively. Smaller values of DB indicate better clustering solutions.

2.1.4. Xie-Beni (XB). The XB index (Xie & Beni, 1991) was originally designed to detect compact and separated clusters in fuzzy c-partitions. A hard partition version is given by the following ratio of compactness to separation (Lamirel & Cuxac, 2015; Lamirel et al., 2016):

$$XB = \frac{WGSS/N}{\min_{i \neq j} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2}. \tag{12}$$

Smaller values of XB indicate better clustering solutions.

2.1.5. Generalized Dunn's Indices (GDs). The GDs (Bezdek & Pal, 1998) comprise a set of 17 variants of the original Dunn's index (Dunn, 1973) devised to address the latter's sensitivity to noise. These CVIs are given by:

$$GD_{rs} = \frac{\min\limits_{i \neq j} \left[\delta_r(\omega_i, \omega_j)\right]}{\max\limits_{k} \left[\Delta_s(\omega_k)\right]}, \tag{13}$$

where $\delta_r(\cdot)$ is a measure of separation and $\Delta_s(\cdot)$ is a measure of compactness. The parameters r and s index the measures' formulations ($r \in \{1, ..., 6\}$ and $s \in \{1, 2, 3\}$). In particular, when employing Euclidean distance, the GD_{43} and GD_{53} variants are formulated using

$$\delta_4(\omega_i, \omega_j) = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2, \tag{14}$$

$$\delta_5(\omega_i, \omega_j) = \frac{CP_2^1(\boldsymbol{v}_i, \omega_i) + CP_2^1(\boldsymbol{v}_j, \omega_j)}{n_i + n_j}, \tag{15}$$

$$\Delta_3(\omega_k) = \frac{2 \times CP_2^1(\boldsymbol{v}_k, \omega_k)}{n_k}. \tag{16}$$

Larger values of these GDs suggest better clustering partitions.

2.1.6. Pakhira-Bandyopadhyay-Maulik (PBM). Consider the I index (Bandyopadhyay & Maulik, 2001) defined as:

$$I = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^p, \ p \geq 1, \tag{17}$$

where

$$E_1 = \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{data}\|_2, \tag{18}$$

$$E_k = \sum_{i=1}^{k} CP_2^1(\boldsymbol{v}_i, \omega_i), \tag{19}$$

$$D_k = \max_{i \neq j} \left(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2\right), \tag{20}$$

The quantities E_k and D_k measure compactness and separation, respectively. This CVI comprises a trade-off among the three competing factors in Eq. (17): $\frac{1}{k}$ decreases with k, whereas both $\frac{E_1}{E_k}$ and D_k increase. By setting $p = 2$ in Eq. (17), the I index reduces to the PBM index (Pakhira et al., 2004). Larger values of PBM indicate better clustering solutions.

2.1.7. Silhouette (SIL). The SIL index (Rousseeuw, 1987) is computed by averaging the silhouette coefficients sc_i across all data samples $\boldsymbol{x}_i$:

$$SIL = \frac{1}{N} \sum_{i=1}^{N} sc_i, \tag{21}$$

where

$$sc_i = \frac{b_i - a_i}{\max\left(a_i, b_i\right)}, \tag{22}$$

$$a_i = \frac{1}{n_i - 1} CP_2^1(\boldsymbol{x}_i, \omega_i), \tag{23}$$

$$b_i = \min_{l, l \neq i} \left[\frac{1}{n_l} CP_2^1(\boldsymbol{x}_i, \omega_l) \right], \tag{24}$$

the variables a_i and b_i measure compactness and separation, respectively. Larger values of SIL (close to 1) indicate better clustering solutions. To reduce computational complexity, some SIL variants, such as (Hruschka et al., 2006, 2004; Luna-Romera et al., 2016; Rawashdeh & Ralescu, 2012), use a centroid-based approach. The simplified SIL (Hruschka et al., 2006, 2004) has been successfully used in clustering data streams processed in chunks, in which the silhouette coefficients are also used to make decisions regarding the centroids' incremental updates (Silva & Hruschka, 2016).

2.1.8. Partition Separation (PS). The PS index (Yang & Wu, 2001) was originally developed for fuzzy clustering; its hard clustering version is given by (Lughofer, 2008):

$$PS = \sum_{i=1}^{k} PS_i, \tag{25}$$

where

$$PS_i = \frac{n_i}{\max\limits_j(n_j)} - exp\left[-\frac{\min\limits_{i\neq j}\left(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2\right)}{\beta_T}\right], \tag{26}$$

$$\beta_T = \frac{1}{k}\sum_{l=1}^{k}\|\boldsymbol{v}_l - \bar{\boldsymbol{v}}\|^2, \tag{27}$$

$$\bar{\boldsymbol{v}} = \frac{1}{k}\sum_{l=1}^{k}\boldsymbol{v}_l, \tag{28}$$

The PS index only comprises a measure of separation between prototypes. Although included in the batch CVI section, it can be readily used to evaluate the partitions identified by unsupervised incremental learners that model clusters using centroids (*e.g.*, (Lughofer, 2008)). Larger values of PS indicate better clustering solutions.

2.1.9. Negentropy Increment (NI). The NI index (Lago-Fernández & Corbacho, 2009; Lago-Fernández & Corbacho, 2010) measures the average normality of the clusters of a given partition Ω via negentropy (Comon, 1994) while avoiding the direct computation of the clusters' differential entropies. Unlike the other CVIs discussed so far, the NI is not explicitly constructed using measures of compactness and separation (Arbelaitz et al., 2013; Lago-Fernández & Corbacho, 2010), thereby being defined as:

$$NI = \frac{1}{2}\sum_{i=1}^{k} p_i \ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}\ln|\boldsymbol{\Sigma}_{data}| - \sum_{i=1}^{k} p_i \ln p_i, \tag{29}$$

where $|\cdot|$ denotes the determinant. The probabilities (p) and covariance matrices ($\boldsymbol{\Sigma}$) are estimated as:

$$p_i = \frac{n_i}{N}, \tag{30}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{n_i - 1}\sum_{\substack{j=1\\ \boldsymbol{x}_j\in\omega_i}}^{n_i}(\boldsymbol{x}_j - \boldsymbol{v}_i)(\boldsymbol{x}_j - \boldsymbol{v}_i)^T, \tag{31}$$

$$\boldsymbol{\Sigma}_{data} = \frac{1}{N-1}\left(\boldsymbol{X}^T\boldsymbol{X} - N\boldsymbol{\mu}_{data}\boldsymbol{\mu}_{data}^T\right), \tag{32}$$

and the means $\boldsymbol{v}$ and $\boldsymbol{\mu}_{data}$ are estimated using Eqs. (1) and (2), respectively. Smaller values of NI indicate better clustering solutions.

2.1.10. Representative Cross Information Potential (rCIP). Cluster evaluation functions (CEFs) based on cross information potential (CIP) (Gokcay & Principe, 2000, 2002) have been consistently used in the literature to evaluate partitions and drive optimization algorithms searching for data structure (Araújo et al., 2013a,b; Gokcay & Principe, 2000, 2002), thus this work includes these CEFs under the CVI category. Precisely, representative approaches (Araújo et al., 2013a,b) replace the sample-by-sample estimation of Renyi's quadratic Entropy (Rényi, 1961) using the Parzen-window method (Duda et al., 2000) (original CIP (Gokcay & Principe, 2000, 2002)) via prototypes and the statistics of their associated Voronoi polyhedrons. The rCIP was devised for prototype-based clustering (*i.e.*, two-step methods: vector quantization followed by clustering of the prototypes) (Ana & Jain, 2003; Cottrell & Rousset, 1997; Karypis et al., 1999; Tyree & Long, 1999; Vesanto & Alhoniemi, 2000). The CEF used here is defined as (Araújo et al., 2013a):

$$CEF = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} rCIP(\omega_i, \omega_j), \tag{33}$$

where

$$rCIP(\omega_i, \omega_j) = \frac{1}{M_i M_j} \sum_{l=1}^{M_i} \sum_{m=1}^{M_j} G(\Delta\boldsymbol{v}_{l,m}, \boldsymbol{\Sigma}_{l,m}), \tag{34}$$

$$G(\Delta\boldsymbol{v}_{l,m}, \boldsymbol{\Sigma}_{l,m}) = \frac{e^{-\frac{1}{2}\Delta\boldsymbol{v}_{l,m}^T \boldsymbol{\Sigma}_{l,m}^{-1} \Delta\boldsymbol{v}_{l,m}}}{\sqrt{(2\pi)^d \left|\boldsymbol{\Sigma}_{l,m}\right|}}, \tag{35}$$

$\Delta\boldsymbol{v}_{l,m} = \boldsymbol{v}_l - \boldsymbol{v}_m$, $\boldsymbol{\Sigma}_{l,m} = \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m$, $\{\boldsymbol{v}_l, \Sigma_l\} \in \omega_i$, $\{\boldsymbol{v}_m, \Sigma_m\} \in \omega_j$, M_i and M_j are the number of prototypes used to represent clusters ω_i and ω_j, respectively. The prototypes and covariance matrices are estimated using Eqs. (1) and (31), respectively. Smaller values of CEF indicate better clustering solutions. Recently, the information potential (Principe, 2010) measure has been used to define a system's state when modeling and analyzing dynamic processes (Oliveira et al., 2018, 2017).

2.1.11. Conn_Index. The Conn_Index (Taşdemir & Merényi, 2007, 2011) was also developed for prototype-based clustering. It is formulated using the connectivity strength matrix ($\boldsymbol{CONN}$), which is a symmetric square similarity matrix that represents local data densities between neighboring prototypes (Taşdemir & Merényi, 2006, 2009). Its $(i, j)^{th}$ entry is formally given by:

$$\boldsymbol{CONN}(i, j) = \boldsymbol{CADJ}(i, j) + \boldsymbol{CADJ}(j, i), \tag{36}$$

where the $(i, j)^{th}$ entry of the non-symmetric cumulative adjacency matrix ($\boldsymbol{CADJ}$) corresponds to the number of samples for which $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are, simultaneously, the first and second closest prototypes (according to some dissimilarity measure $D(\cdot)$, such as Euclidean distance), respectively:

$$\boldsymbol{CADJ}(i, j) = card(RF_{i,j}), \tag{37}$$

$$RF_{i,j} = \{\boldsymbol{x}_k \in RF_i : D(\boldsymbol{x}_k, \boldsymbol{v}_j) \leq D(\boldsymbol{x}_k, \boldsymbol{v}_l)\ \forall l \neq i\}, \tag{38}$$

$$RF_i = \left\{\boldsymbol{x}_k \in \boldsymbol{X} : D(\boldsymbol{x}_k, \boldsymbol{v}_i) \leq D(\boldsymbol{x}_k, \boldsymbol{v}_j)\ \forall j\right\}. \tag{39}$$

where $card(\cdot)$ is the cardinality operator. The Conn_Index is defined as:

$$Conn\_Index = Intra\_Conn \times (1 - Inter\_Conn), \tag{40}$$

where the intra-cluster ($Intra\_Conn$) and inter-cluster ($Inter\_Conn$) connectivities are:

$$Intra\_Conn = \frac{1}{k} \sum_{l=1}^{k} \boldsymbol{Intra}(\omega_l), \tag{41}$$

$$\boldsymbol{Intra}(\omega_l) = \frac{1}{n_l} \sum_{\substack{i,j \\ \boldsymbol{v}_i, \boldsymbol{v}_j \in \omega_l}}^{M} \boldsymbol{CADJ}(i, j), \tag{42}$$

$$Inter\_Conn = \frac{1}{k}\sum_{l=1}^{k}\max_{m,m\neq l}\left[\boldsymbol{Inter}(\omega_l,\omega_m)\right], \tag{43}$$

$$\boldsymbol{Inter}(\omega_l,\omega_m) = \frac{\sum\limits_{\substack{i,j\\ \boldsymbol{v}_i\in\omega_l,\boldsymbol{v}_j\in\omega_m}}^{M} \boldsymbol{CONN}(i,j)}{\sum\limits_{\substack{i,j\\ \boldsymbol{v}_i\in V_{l,m}}}^{M} \boldsymbol{CONN}(i,j)}, \tag{44}$$

$$V_{l,m} = \{\boldsymbol{v}_i : \boldsymbol{v}_i \in \omega_l, \exists \boldsymbol{v}_j \in \omega_m : \boldsymbol{CADJ}(i,j) > 0\}, \tag{45}$$

the variable M is the total number of prototypes, and $\boldsymbol{Inter}(\omega_l,\omega_m) = 0$ if $V_{l,m} = \{\emptyset\}$. Naturally, the quantities $Intra\_Conn$ and $Inter\_Conn$ measure compactness and separation, respectively. Larger values of the Conn_Index (close to 1) indicate better clustering solutions.

2.2. INCREMENTAL CLUSTER VALIDITY INDICES (ICVIS)

The compactness and separation terms commonly found in CVIs are generally computed using data samples and prototypes, respectively (Ibrahim et al., 2018a; Moshtaghi et al., 2018). In order to handle online clustering applications demands (*i.e.*, data streams), an incremental CVI (iCVI) formulation that recursively estimates the compactness term was introduced in (Moshtaghi et al., 2018; Moshtaghi et al., 2019) in the context of fuzzy clustering.

Remark 1. Hereafter the notation CP_q^p is simplified to CP. This notation was changed because only the squared Euclidean norm ($p = q = 2$) will be used for the compactness. Henceforth, CP's subscripts designate *cluster membership*.

Specifically, consider the hard clustering version of cluster i's compactness CP_i (*i.e.*, by setting the fuzzy memberships in (Moshtaghi et al., 2018; Moshtaghi et al., 2019) to binary indicator functions):

$$CP_i = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \|\boldsymbol{x}_j - \boldsymbol{v}_i\|_2^2. \tag{46}$$

In such a case, when a new sample $\boldsymbol{x}$ is presented and encoded by cluster i, then its new compactness value becomes:

$$CP_i^{new} = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i^{new}} \|\boldsymbol{x}_j - \boldsymbol{v}_i^{new}\|_2^2, \tag{47}$$

where

$$n_i^{new} = n_i^{old} + 1, \tag{48}$$

$$\boldsymbol{v}_i^{new} = \boldsymbol{v}_i^{old} + (\boldsymbol{x} - \boldsymbol{v}_i^{old})/n_i^{new}, \tag{49}$$

and

$$N^{new} = N^{old} + 1. \tag{50}$$

The compactness in Eq. (47) can be updated incrementally as (Moshtaghi et al., 2018; Moshtaghi et al., 2019):

$$CP_i^{new} = CP_i^{old} + \|\boldsymbol{z}_i\|_2^2 + n_i^{old}\|\Delta\boldsymbol{v}_i\|_2^2 + 2\Delta\boldsymbol{v}_i^T\boldsymbol{g}_i^{old}, \tag{51}$$

where

$$\boldsymbol{z}_i = \boldsymbol{x} - \boldsymbol{v}_i^{new}, \tag{52}$$

$$\Delta\boldsymbol{v}_i = \boldsymbol{v}_i^{old} - \boldsymbol{v}_i^{new}, \tag{53}$$

and vector $\boldsymbol{g}$, which is formally defined as

$$\boldsymbol{g}_i = \sum_{j=1}^{n_i} \left(\boldsymbol{x}_j - \boldsymbol{v}_i\right), \tag{54}$$

is incrementally updated at each iteration using:

$$\boldsymbol{g}_i^{new} = \boldsymbol{g}_i^{old} + \boldsymbol{z}_i + n_i^{old} \Delta \boldsymbol{v}_i. \tag{55}$$

Using such incremental formulation, the following iCVIs were derived (their hard partition counterparts are shown here (Ibrahim et al., 2019)):

1. incremental Xie-Beni (iXB) (Moshtaghi et al., 2018; Moshtaghi et al., 2019)

$$XB^{new} = \frac{1}{N^{new}} \times \frac{\sum\limits_{i=1}^{k^{new}} CP_i^{new}}{\min\limits_{i \neq j}\left(\|\boldsymbol{v}_i^{new} - \boldsymbol{v}_j^{new}\|_2^2\right)}, \tag{56}$$

2. incremental Davies-Bouldin (iDB) (Moshtaghi et al., 2018; Moshtaghi et al., 2019)

$$DB^{new} = \frac{1}{k^{new}} \sum_{i=1}^{k^{new}} \max_{j, j \neq i} \left(\frac{\frac{CP_i^{new}}{n_i^{new}} + \frac{CP_j^{new}}{n_j^{new}}}{\|\boldsymbol{v}_i^{new} - \boldsymbol{v}_j^{new}\|_2^2} \right), \tag{57}$$

3. incremental generalized Dunn's indices (iGDs) (Ibrahim et al., 2019)

$$GD_{43}^{new} = \frac{\min\limits_{i \neq j}\left(\|\boldsymbol{v}_i^{new} - \boldsymbol{v}_j^{new}\|_2\right)}{\max\limits_{k}\left(\frac{2CP_k^{new}}{n_k^{new}}\right)}, \tag{58}$$

$$GD_{53}^{new} = \frac{\min\limits_{i \neq j}\left(\frac{CP_i^{new} + CP_j^{new}}{n_i^{new} + n_j^{new}}\right)}{\max\limits_{k}\left(\frac{2CP_k^{new}}{n_k^{new}}\right)}. \tag{59}$$

Note that only one prototype $\boldsymbol{v}$, counter n and compactness CP are updated after each input presentation. If a new cluster emerges, then $k^{new} = k^{old} + 1$, and its compactness CP and vector $\boldsymbol{g}$ are initialized as 0 and $\vec{\mathbf{0}}$ (since $\boldsymbol{v} = \boldsymbol{x}$), respectively.

2.3. ADAPTIVE RESONANCE THEORY (ART)

This study uses a neural network implementation of adaptive resonance theory (ART) (Carpenter & Grossberg, 1987) given its fast and stable online learning as well as automatic category recognition capabilities. ART models encompass a rich history with many implementations well-suited to iCVI computation (see (Brito da Silva et al., 2019) for a comprehensive review on ART models); the ones used in this study's experiments are discussed next.

2.3.1. Fuzzy ART. The fuzzy ART model (Carpenter et al., 1991) implements fuzzy logic (Zadeh, 1965) to bound data within hyper-boxes. For a normalized data set $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ ($\boldsymbol{x}_i \in \mathbb{R}^d, 0 \leq x_{i,j} \leq 1$, $j = \{1, ..., d\}$), the fuzzy ART algorithm, with parameters (α, β, ρ), is defined by:

$$\boldsymbol{I} = (\boldsymbol{x}, 1 - \boldsymbol{x}), \tag{60}$$

$$T_j = \frac{\| \min(\boldsymbol{I}, \boldsymbol{w}_j)\|_1}{\alpha + \|\boldsymbol{w}_j\|_1}, \tag{61}$$

$$\| \min(\boldsymbol{I}, \boldsymbol{w}_j)\|_1 \geq \rho \|\boldsymbol{I}\|_1, \tag{62}$$

$$\boldsymbol{w}_j^{new} = \boldsymbol{w}_j^{old}(1 - \beta) + \beta \min(\boldsymbol{I}, \boldsymbol{w}_j^{old}). \tag{63}$$

Equation (60) is the complement coding function, which concatenates sample $\boldsymbol{x}$ and its complement to form an input vector $\boldsymbol{I}$ with dimension $2d$. Equation (61) is the activation function for each fuzzy ART category j, where $\| \cdot \|_1$ is the L_1 norm, $min(\cdot)$ is performed component-wise, and α is a tie breaking constant. Each category is checked for validity against Eq. (62)'s vigilance parameter ρ in a descending order of activation. If no valid category is found during training, then a new category is initialized using $\boldsymbol{I}$ as the new weight vector $\boldsymbol{w}$. Otherwise, the winning category is updated according to Eq. (63) using learning rate β.

2.3.2. Fuzzy ARTMAP. In a fuzzy ARTMAP network (Carpenter et al., 1992), two fuzzy ART modules, A- and B-side, are supplied with separate but dependent data streams. Specifically, in classification settings, these streams consist of data and class labels, respectively. Both ART modules cluster their inputs according to local topology and parameters while an inter-ART module enforces a surjective mapping of the A-side to the B-side, effectively learning the functional map of the A-side to the B-side categories. This model will be *required* to (i) extend the iCVI study to prototype-based CVIs such as the Conn_Index, and (ii) perform the experiments under a clustering agnostic framework (see Section 5), in which the A-side categories represent cluster prototypes and are driven by the B-side true data partition labels (note that we follow a simplified fuzzy ARTMAP design (Kasuba, 1993), in which the B-side is replaced by a stream of class labels).

3. EXTENSIONS OF ICVIS

To compute the CVIs mentioned in Section 2.1 incrementally, employing one of the following approaches is sufficient:

1. The recursive computation of compactness developed in (Moshtaghi et al., 2018; Moshtaghi et al., 2019) (CVIs: CH, WB, PBM, and SIL).
2. The incremental computation of probabilities, means and covariance matrices (CVIs: rCIP and NI). Naturally, if the clustering algorithm of choice already models the clusters using a priori probabilities, means and covariance matrices (such as Gaussian ART (Williamson, 1996) and Bayesian ART (Vigdor & Lerner, 2007)), then, similarly to PS, these CVIs can be readily computed.

3. The incremental building of a multi-prototype representation of clusters using a modified ART model while tracking the density-based connections between neighboring prototypes (CVI: Conn_index). Specifically, the latter is accomplished by updating (incrementing and/or expanding) $\boldsymbol{CADJ}$ and $\boldsymbol{CONN}$ matrices as clusters grow and/or are dynamically created.

In the following iCVIs' extensions (iCH, iWB, iPBM, iSIL, irCIP, iNI, and iConn_index), if a new cluster is formed after sample $\boldsymbol{x}$ is presented, then the total number of clusters is updated to $k^{new} = k^{old} + 1$ (otherwise $k^{new} = k^{old}$), and, unless otherwise noted, the variables associated with this new cluster are initialized as $n_{k^{new}}^{new} = 1$ (number of samples encoded), $\boldsymbol{v}_{k^{new}}^{new} = \boldsymbol{x}$ (this clusters' prototype), $CP_{k^{new}}^{new} = 0$ (initial compactness), $\boldsymbol{g}_{k^{new}}^{new} = \vec{\boldsymbol{0}}$ (initial vector $\boldsymbol{g}$). Naturally, clusters that do not encode the presented sample remain with constant parameter values for the duration of that input presentation. Also note that, where necessary, the Euclidean norm is replaced with the squared Euclidean norm (*i.e.*, $\|\cdot\|^2$) to compute the compactness CP (as per (Moshtaghi et al., 2018; Moshtaghi et al., 2019)). Finally, for iCVIs that require the computation of pairwise (dis)similarity between prototypes, the (dis)similarity matrix is kept in memory, where only the rows and columns corresponding to the prototype that is adapted are modified.

3.1. INCREMENTAL CALINSKI-HARABASZ INDEX (ICH)

The iCH computation is defined as:

$$CH^{new} = \frac{\sum\limits_{i=1}^{k^{new}} SEP_i^{new}}{\sum\limits_{i=1}^{k^{new}} CP_i^{new}} \times \frac{N^{new} - k^{new}}{k^{new} - 1}, \tag{64}$$

where

$$SEP_i^{new} = n_i^{new} \|\boldsymbol{v}_i^{new} - \boldsymbol{\mu}_{data}^{new}\|_2^2. \tag{65}$$

Note that the variables $\{n_1, ..., n_k\}$, $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$, $\{CP_1, ..., CP_k\}$, $\{\boldsymbol{g}_1, ..., \boldsymbol{g}_k\}$, $\boldsymbol{\mu}_{data}$, k, N, and $\{SEP_1, ..., SEP_k\}$ are all kept in memory. These are updated using Eqs. (48) to (55), except for SEP, which is adapted using Eq. (65). The data mean $\boldsymbol{\mu}_{data}$ is updated like the prototypes $\boldsymbol{v}$ (*i.e.*, Eq. (49) using $\boldsymbol{\mu}_{data}$ in place of $\boldsymbol{v}$ and N in place of n).

3.2. INCREMENTAL WB INDEX (IWB)

The iWB computation is very similar to iCH's:

$$WB^{new} = k^{new} \frac{\sum\limits_{i=1}^{k^{new}} CP_i^{new}}{\sum\limits_{i=1}^{k^{new}} SEP_i^{new}}, \tag{66}$$

and the same variable definitions previously mentioned apply.

3.3. INCREMENTAL PAKHIRA-BANDYOPADHYAY-MAULIK INDEX (IPBM)

The iPBM computation is defined as:

$$PBM^{new} = \left[\frac{\max\limits_{i \neq j} \left(\|\boldsymbol{v}_i^{new} - \boldsymbol{v}_j^{new}\|_2^2 \right)}{\sum\limits_{i=1}^{k} CP_i^{new}} \times \frac{CP_0^{new}}{k^{new}} \right]^2, \tag{67}$$

where CP_0 and $\sum\limits_{i=1}^{k} CP_i^{new}$ correspond to E_1 and E_k, respectively. These are updated according to Eqs. (48) to (55) along with the remaining compactness variables. Only the pairwise distances with respect to the updated prototype need to be recomputed at any given iteration.

3.4. INCREMENTAL SILHOUETTE INDEX (ISIL)

The SIL index is inherently batch (offline), since it requires the entire data set to be computed (the silhouette coefficients are averaged across all data samples in Eq. (21)). To remove such a requirement and enable incremental updates, a hard version of the centroid-based SIL variant introduced in (Rawashdeh & Ralescu, 2012) is employed here as well as the squared Euclidean norm (*i.e.*, $\| \cdot \|_2^2$): this is done in order to employ the recurrent formulation of the compactness in Eq. (51). Consider the matrix $\boldsymbol{S}_{k\times k}$, where k prototypes $\boldsymbol{v}_i$ are used to compute the centroid-based SIL (instead of the N samples $\boldsymbol{x}_i$ - which, by definition, are discarded after each presentation in online mode). Define each entry $s_{i,j} = D(\boldsymbol{v}_i, \omega_j)$ (dissimilarity of $\boldsymbol{v}_i$ to cluster ω_j) of $\boldsymbol{S}_{k\times k}$ as:

$$s_{i,j} = \frac{1}{n_j} \sum_{\substack{l=1 \\ \boldsymbol{x}_l \in \omega_j}}^{n_j} \|\boldsymbol{x}_l - \boldsymbol{v}_i\|_2^2 = \frac{1}{n_j} CP(\boldsymbol{v}_i, \omega_j), \tag{68}$$

where $i = \{1, ..., k\}$ and $j = \{1, ..., k\}$. The silhouette coefficients can be obtained from the entries of $\boldsymbol{S}_{k\times k}$ as:

$$sc_i = \frac{\min\limits_{l, l\neq J}(s_{i,l}) - s_{i,J}}{\max\left[s_{i,J}, \min\limits_{l, l\neq J}(s_{i,l})\right]}, \boldsymbol{v}_i \in \omega_J. \tag{69}$$

where $a_i = s_{i,J}$ and $b_i = \min\limits_{l, l\neq J}(s_{i,l})$.

Remark 2. At first, when examining Eq. (68), one might be tempted to store a $k \times k$ matrix of compactness entries along with their accompanying k^2 vectors $\boldsymbol{g}$ (one for each entry) to enable incremental updates of each element of matrix of $\boldsymbol{S}_{k\times k}$; this approach, however, may lead to unnecessarily large memory requirements. A more careful examination shows that it is sufficient to simply redefine CP and $\boldsymbol{g}$ for each cluster i ($i = \{1, ..., k\}$) as:

$$CP_i = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \|\boldsymbol{x}_j - \vec{\boldsymbol{0}}\|_2^2 = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \|\boldsymbol{x}_j\|_2^2, \tag{70}$$

$$\boldsymbol{g}_i = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \left(\boldsymbol{x}_j - \vec{\boldsymbol{0}}\right) = \sum_{\substack{j=1 \\ \boldsymbol{x}_j \in \omega_i}}^{n_i} \boldsymbol{x}_j, \tag{71}$$

which is equivalent to fixing $\boldsymbol{v} = \vec{\boldsymbol{0}}$. Therefore, their incremental update equations become (as opposed to Eqs. (51) and (55)):

$$CP_i^{new} = CP_i^{old} + \|\boldsymbol{x}\|_2^2, \tag{72}$$

$$\boldsymbol{g}_i^{new} = \boldsymbol{g}_i^{old} + \boldsymbol{x}. \tag{73}$$

Using this trick, when a sample $\boldsymbol{x}$ is assigned to cluster ω_J, then the update equations for each entry $s_{i,j}$ of $\boldsymbol{S}_{k \times k}$ are given by Eq. (74). Note that the numerators of the expressions in Eq. (74) update the compactness "as if" the prototype has changed from $\vec{\boldsymbol{0}}$ to $\boldsymbol{v}^{new}$ at every iteration ($\Delta\boldsymbol{v} = -\boldsymbol{v}^{new}$). The remaining variables such as n, N, and $\boldsymbol{v}$ are updated as previously described. This allows $\{CP_1, ..., CP_k\}$ and $\{\boldsymbol{g}_1, ..., \boldsymbol{g}_k\}$ to continue being stored similarly to the previous iCVIs, instead of a $k \times k$ matrix of compactness and the associated k^2 vectors $\boldsymbol{g}$.

Remark 3. In the case where a new cluster ω_{k+1} is created following the presentation of sample $\boldsymbol{x}$, then a new column and a new row are appended to the matrix $\boldsymbol{S}_{k \times k}$. Unlike the other iCVIs, the compactness CP_{k+1} and vector $\boldsymbol{g}_{k+1}$ of this cluster are initialized as $\|\boldsymbol{x}\|_2^2$ and $\boldsymbol{x}$, respectively. Then, the entries of $\boldsymbol{S}_{k \times k}$ are updated using Eq. (75).

Following the incremental updates of the entries of $\boldsymbol{S}_{k \times k}$ (Eq. (74) or (75)), the silhouette coefficients (sc_i) are computed (Eq. (69)), and the iSIL is updated as:

$$s_{i,j}^{new} = \begin{cases} \frac{1}{n_j^{new}} \left(CP_j^{old} + \|\boldsymbol{z}_i\|_2^2 + n_j^{old}\|\boldsymbol{v}_i^{old}\|_2^2 - 2\boldsymbol{v}_i^{old\,T} \boldsymbol{g}_j^{old}\right) & , (i \neq J, j = J) \\ \frac{1}{n_j^{old}} \left(CP_j^{old} + n_j^{old}\|\boldsymbol{v}_i^{new}\|_2^2 - 2\boldsymbol{v}_i^{new\,T} \boldsymbol{g}_j^{old}\right) & , (i = J, j \neq J) \\ \frac{1}{n_j^{new}} \left(CP_j^{old} + \|\boldsymbol{z}_j\|_2^2 + n_j^{old}\|\boldsymbol{v}_j^{new}\|_2^2 - 2\boldsymbol{v}_j^{new\,T} \boldsymbol{g}_j^{old}\right) & , (i = J, j = J) \\ s_{i,j}^{old} & , (i \neq J, j \neq J) \end{cases} \tag{74}$$

$$s_{i,j}^{new} = \begin{cases} CP_{k+1} + \|\boldsymbol{v}_i^{old}\|_2^2 - 2\boldsymbol{v}_i^{old\,T}\boldsymbol{g}_{k+1} & , (i \neq k+1, j = k+1) \\ \frac{1}{n_j^{old}}\left(CP_j^{old} + n_j^{old}\|\boldsymbol{v}_i^{new}\|_2^2 - 2\boldsymbol{v}_i^{new\,T}\boldsymbol{g}_j^{old}\right) & , (i = k+1, j \neq k+1) \\ 0 & , (i = k+1, j = k+1) \\ s_{i,j}^{old} & , (i \neq k+1, j \neq k+1) \end{cases} \tag{75}$$

$$SIL^{new} = \frac{1}{k^{new}} \sum_{i=1}^{k^{new}} sc_i^{new}. \tag{76}$$

3.5. INCREMENTAL NEGENTROPY INCREMENT (INI)

The iNI computation is defined as:

$$NI^{new} = \sum_{i=1}^{k} p_i^{new} \ln\left(\frac{\sqrt{|\boldsymbol{\Sigma}_i^{new}|}}{p_i^{new}}\right) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_{data}| \tag{77}$$

where $p_i^{new} = n_i^{new}/N^{new}$, and $\boldsymbol{\Sigma}_i^{new}$ is computed using the following recursive formula (Duda et al., 2000):

$$\boldsymbol{\Sigma}^{new} = \frac{n^{new} - 2}{n^{new} - 1}\left(\boldsymbol{\Sigma}^{old} - \delta I\right) + \frac{1}{n^{new}}\left(\boldsymbol{x} - v^{old}\right)\left(\boldsymbol{x} - v^{old}\right)^T + \delta I \tag{78}$$

This work's authors set $\delta = 10^{-\frac{\epsilon}{d}}$ to avoid numerical errors, where ϵ is a user-defined parameter. If a new cluster is created, then $\boldsymbol{\Sigma} = \delta I$ and $|\boldsymbol{\Sigma}| = 10^{-\epsilon}$.

3.6. INCREMENTAL REPRESENTATIVE CROSS INFORMATION POTENTIAL (IRCIP) AND CROSS-ENTROPY (IRH)

Section 5 will show that using the representative cross-entropy rH for computing the CEF makes it easier to observe the behavior of the incremental clustering process (this corroborates a previous study in which rH was deemed more informative than rCIP for multivariate data visualization (Brito da Silva & Wunsch II, 2018b)):

$$rH(\omega_i, \omega_j) = -\ln\left[rCIP(\omega_i, \omega_j)\right], \tag{79}$$

$$CEF = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} rH(\omega_i, \omega_j). \tag{80}$$

Note that, as opposed to the rCIP-based CEF, larger values of rH-based CEF indicate better clustering solutions. Concretely, since the CEF only measures separation, then, like iNI, it is only necessary to update the means and the covariance matrices online in order to construct the incremental CEF (iCEF). This is also done using Eqs. (49) and (78), respectively. The iCEFs, based on rCIP and rH, are hereafter referred to as irCIP and irH, respectively.

3.7. INCREMENTAL CONN_INDEX (ICONN_INDEX)

The Conn_Index is an inherently batch CVI formulated around the $\boldsymbol{CADJ}$ and $\boldsymbol{CONN}$ matrices. Each element (i, j) of the $\boldsymbol{CADJ}$ matrix requires the count of the samples in the data set with the first and second closest prototypes, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ respectively, while the symetric $\boldsymbol{CONN}$ matrix is equal to the sum of the $\boldsymbol{CADJ}$ matrix with its transpose. When clustering data online, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ may change for previously presented samples as prototypes are continuously modified or created. However, for the purpose of building and incrementing $\boldsymbol{CADJ}$ and $\boldsymbol{CONN}$ matrices online (with only one matrix entry changing per sample presentation), it is assumed that the trends exhibited over time by the iConn_Index do not differ dramatically from its offline counterpart. Batch calculation can be eliminated entirely by keeping the values of Eqs. (42) and (44) in memory and updating only the entries that depend on prototypes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$.

In this study, the multi-prototype cluster representation required by the Conn_Index was generated using a modified fuzzy ARTMAP, whose modules A and B are used for prototype and cluster definition, respectively. Fuzzy ARTMAP's module A was modified in such a way that it forcefully creates two prototypes using the first two samples of every emerging cluster in module B. By enforcing this dynamic, each cluster always possesses at least two prototypes for the computation of the iConn_Index. This strategy addresses two problems: first, it allows $\boldsymbol{CADJ}$ to be created from the second presented sample and onward; second, it prevents cases in which well-separated clusters are strongly connected simply because one of them does not have another prototype to assume the role of the second winner.

Remark 4. Fuzzy ART neural networks represent prototypes by the categories' weight vectors $\boldsymbol{w}$ (see Section 2.3). Thus, the two highest ranked resonant categories (i.e., the ones with the largest activation function values according to Eq. (61) that also satisfy Eq. (62)'s resonance criteria) constitute the first and second winner pair. Note that the second winning prototype for a sample ($\boldsymbol{w}_j$) is the winning A-side category when the first winning prototype ($\boldsymbol{w}_i$) has been removed from the A-side category set. Moreover, if no second resonant category is found during search, then the second winning category defaults to the highest activated one.

Upon receiving the very first sample input, we can only form a single viable cluster and prototype and, therefore, we cannot calculate the iConn_Index. We remedy this by introducing a counter separate from the $\boldsymbol{CADJ}$ matrix. This counter is incremented to count the number of times a sample has been presented while only a single prototype exists, thus preserving these otherwise troublesome samples. Upon creation of the second prototype $\boldsymbol{w}_2$ in fuzzy ARTMAP's module A, the $\boldsymbol{CADJ}$ matrix will be incremented for the first time at element $(2, 1)$. At this point, the element $(1, 2)$ will be incremented by the value of the instance counter. When this instance counting technique is combined with the forcible splitting of prototypes previously mentioned, the result is that all samples will be taken into account when computing iConn_Index. For all subsequent samples, the instance counter will remain unused, the $\boldsymbol{CONN}$ and $\boldsymbol{CADJ}$ incrementing will be streamlined, and the iConn_Index will be calculable.

Remark 5. The iConn_Index boundary conditions are listed below:

1. Cluster represented by a single prototype (singleton), e.g., immediately following the creation of a new cluster: the $\boldsymbol{Intra}$ entry for that cluster, given by Eq. (42), defaults to a value of 0, since $\boldsymbol{CADJ}(i, i) = 0\ \forall i$.

2. A single non-singleton cluster exists (i.e., a unique cluster represented by multiple prototypes): $\boldsymbol{Intra} = 1$ for this cluster.

3. Like the remaining iCVIs in this study, iConn_Index is not defined for a single cluster, since $\boldsymbol{Inter}$ (Eq. (43)) cannot be computed.

4. Instead of the original constraint $\boldsymbol{CADJ}(i, j) > 0$ imposed by Eq. (45), this paper's iConn_Index implementation uses $\boldsymbol{CONN}(i, j) > 0$, as this seemed to make its behavior smoother in our experiments.

Note that items (1)-(3) arise directly from the Conn_Index definitions (Taşdemir & Merényi, 2011), whereas item (4) follows from the step-by-step illustrative example in (Taşdemir & Merényi, 2007). For further clarity, the pseudo-code for the iConn_index is provided in Algorithm 8.

4. NUMERICAL EXPERIMENTS DESIGN

The behaviors of 13 iCVIs (namely iCH, iSIL, iPBM, iWB, iXB, iDB, iGD_{43}, iGD_{53}, PS, iNI, irCIP, irH, and iConn_Index) were analyzed using the benchmark data sets summarized in Table 1. These synthetic and real world data sets are also depicted in Figure 1's scatter plots and encompass a diverse set of properties, such as unbalanced classes, high dimensionality, levels of overlap and number of samples.

Algorithm 8: iConn_Index

```
/* Initialization */
$\boldsymbol{CADJ}$ := [] ;
$\boldsymbol{CONN}$ := [] ;
$\boldsymbol{Inter}$ := [];
$\boldsymbol{Intra}$ := [];
$Inter\_Conn$ := 0;
$Intra\_Conn$ := 0;
$S_{counter}$ := 0;
/* iConn_Index computation */
while streaming samples do
    $\boldsymbol{x}$ := new sample;
    Process $\boldsymbol{x}$ with an ART-based model to obtain the first $\boldsymbol{w}_i \in \omega_k$ and second
      $\boldsymbol{w}_j \in \omega_l$ best matching prototypes;
    if $\boldsymbol{w}_j = \{\emptyset\}$ then
        $S_{counter} := S_{counter} + 1$;
    else if $S_{counter} > 0$ then
        $CADJ(\boldsymbol{w}_j, \boldsymbol{w}_i) := CADJ(\boldsymbol{w}_j, \boldsymbol{w}_i) + S_{counter}$;
        $S_{counter} := 0$;
    if $S_{counter} = 0$ then
        $CADJ(\boldsymbol{w}_i, \boldsymbol{w}_j) := CADJ(\boldsymbol{w}_i, \boldsymbol{w}_j) + 1$;
        Update $\boldsymbol{CONN}$ using Eq. (36);
        Update $\boldsymbol{Intra}(\omega_k)$ using Eq. (42);
        if $\omega_k \neq \omega_l$ then
            Update $\boldsymbol{Inter}(\omega_k, \omega_l)$ and $\boldsymbol{Inter}(\omega_l, \omega_k)$ using Eq. (44);
        else
            Update $\boldsymbol{Inter}(\omega_k, \omega_m)$, $\forall m$ using Eq. (44);
        end
        Recompute $Intra\_Conn$ using Eq. (41);
        Recompute $Inter\_Conn$ using Eq. (43);
        Recompute $Conn\_Index$ using Eq. (40);
end
```

Like (Ibrahim et al., 2018a; Ibrahim et al., 2019; Ibrahim et al., 2018b; Moshtaghi et al., 2018; Moshtaghi et al., 2019), a natural ordering, *i.e.*, meaningful temporal information is assumed. To emulate such scenarios, the samples were presented in a *cluster-by-cluster* fashion (samples within a given cluster were randomized), and thus this experiment setup is suitable for change point detection (Ibrahim et al., 2019). All iCVIs were subjected to the same 10 random orders of clusters (and order of samples within each cluster) per data set per experiment (see Sections 5.1 to 5.3).

The following discussion is relative to the data sets used in the experiments, their respective order of cluster and sample presentation as well the application of linear normalization. The latter assumes knowledge of the minimum and maximum data statistics, since the vector quantization required by the iConn_Index is realized via fuzzy ARTMAP. Therefore, for consistency, all data sets were normalized to the unit cube $[0, 1]^d$. Additionally, note that the fuzzy ARTMAP dynamics were performed with the additional application of complement coding (Carpenter et al., 1992). Finally, note that this study does not employ multi-prototype representations for irCIP or irH, *i.e.*, $M_i = M_j = 1, \forall i, j$ in Eq. (34), since, as opposed to iConn_Index, such representations are not mandatory for their computation. Moreover, in these experiments, $\epsilon = 12$ in Eq. (78) for the incremental computation of the covariance matrices used by irCIP, irH and iNI.

The numerical experiments and the statistical analysis were carried out using the MATLAB software environment and the scmamp R package (Calvo & Santafé, 2016), respectively. The source code of the (i)CVIs, ART models' algorithms, and experiments is provided by the *iCVI MATLAB Toolbox* at the Applied Computational Intelligence Laboratory public GitHub repository[1].

5. A COMPARATIVE STUDY

This section discusses the behavior of the iCVIs in three general cases when assessing the quality of clustering solutions in real-time: (1) correct partitions, (2) under-partitions, and (3) over-partitions. It should be emphasized that this analysis is not focused on evaluating the performance or capabilities of specific online clustering algorithms, but instead the purpose of this study is to observe the behavior of the iCVIs in these different scenarios to gain insight on their applicability. Similar to (Ibrahim et al., 2019), in each of these

[1] https://github.com/ACIL-Group/iCVI-toolbox

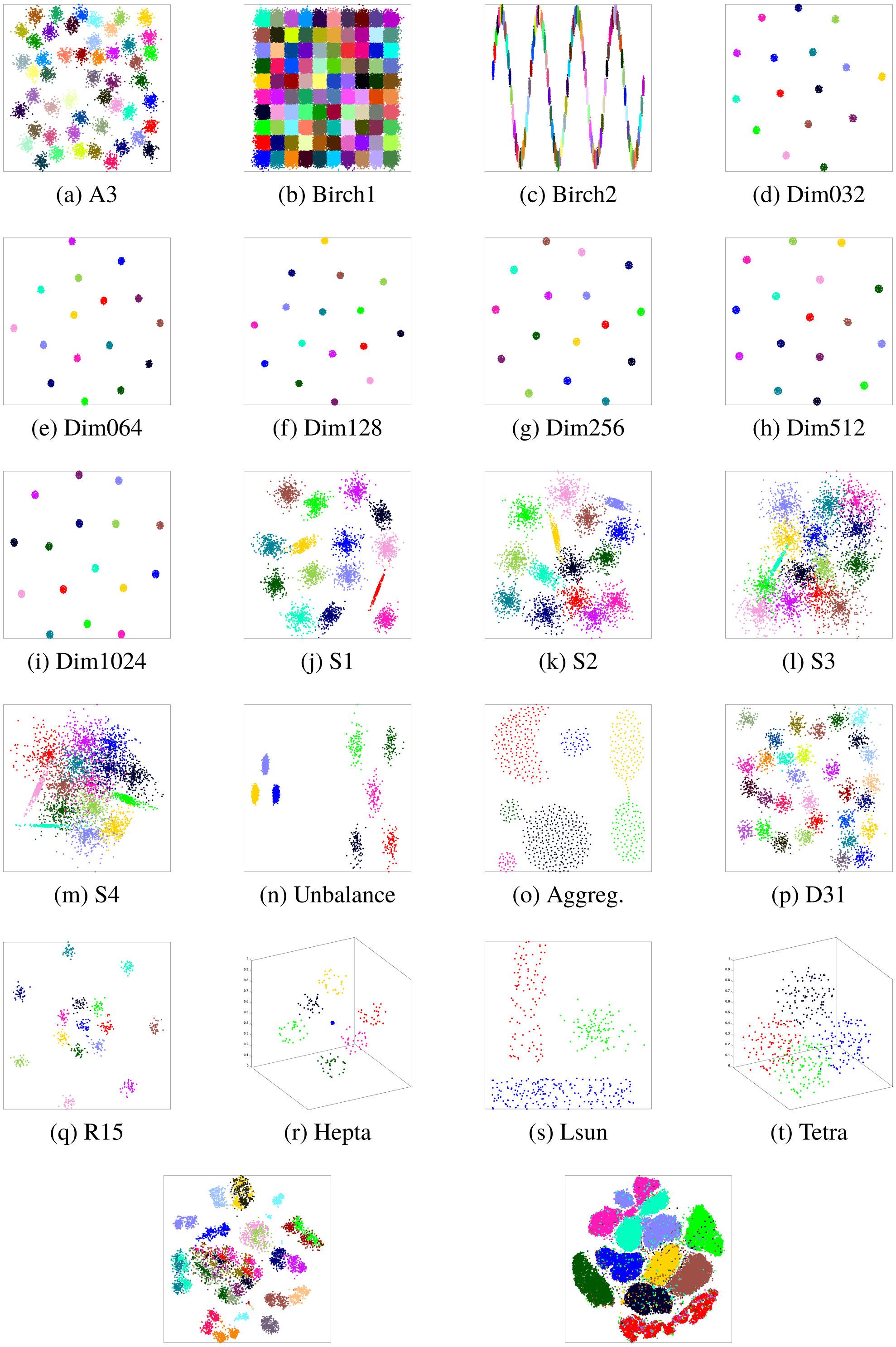
Table 1. Summary of the data sets' characteristics.

Data set	# Samples	# Features	# Clusters	Reference(s)
Synthetic data sets				
A3[a]	7500	2	50	(Fränti & Sieranoja, 2018; Kärkkäinen & Fränti, 2002)
Birch1[a]	100000	2	100	(Fränti & Sieranoja, 2018; Zhang et al., 1997)
Birch2[a]	100000	2	100	(Fränti & Sieranoja, 2018; Zhang et al., 1997)
Dim032[a]	1024	32	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
Dim064[a]	1024	64	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
Dim128[a]	1024	128	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
Dim256[a]	1024	256	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
Dim512[a]	1024	512	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
Dim1024[a]	1024	1024	16	(Fränti & Sieranoja, 2018; Fränti et al., 2006)
S1[a]	5000	2	15	(Fränti & Sieranoja, 2018; Fränti & Virmajoki, 2006)
S2[a]	5000	2	15	(Fränti & Sieranoja, 2018; Fränti & Virmajoki, 2006)
S3[a]	5000	2	15	(Fränti & Sieranoja, 2018; Fränti & Virmajoki, 2006)
S4[a]	5000	2	15	(Fränti & Sieranoja, 2018; Fränti & Virmajoki, 2006)
Unbalance[a]	6500	2	8	(Fränti & Sieranoja, 2018; Rezaei & Fränti, 2016)
Aggregation[a]	788	2	7	(Gionis et al., 2007)
D31[a]	3100	2	31	(Veenman et al., 2002)
R15[a]	600	2	15	(Fränti & Sieranoja, 2018; Veenman et al., 2002)
Hepta[b]	212	3	7	(Ultsch, 2005)
Lsun[b]	400	2	3	(Ultsch, 2005)
Tetra[b]	400	3	4	(Ultsch, 2005)
Real world data sets				
Isolet[c]	7797	617	26	(Cai et al., 2011; Cai et al., 2011; Dua & Graff, 2017)
MNIST[c]	70000	784	10	(Cai et al., 2011; Cai et al., 2011; Lecun et al., 1998)

[a] *Clustering basic benchmark* and *Other clustering datasets*, available at `http://cs.uef.fi/sipu/datasets`.

[b] *Fundamental Clustering Problem Suite*, available at `https://www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/data?language_sync=1`.

[c] MATLAB processed data sets, available at `http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html`.



(a) A3 (b) Birch1 (c) Birch2 (d) Dim032

(e) Dim064 (f) Dim128 (g) Dim256 (h) Dim512

(i) Dim1024 (j) S1 (k) S2 (l) S3

(m) S4 (n) Unbalance (o) Aggreg. (p) D31

(q) R15 (r) Hepta (s) Lsun (t) Tetra

(u) Isolet (v) MNIST

Figure 1. (a)-(t) Synthetic data sets. (u)-(v) Real world data sets. High dimensional data sets are shown using a 2-dimensional t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) projection.

scenarios, we investigate the iCVIs' dynamics triggered by the two following events: (a) the creation of a new cluster (for scenarios (1) and (3)) or the merging of two clusters (for scenario (2)) and (b) the assignment of samples to the current (existing) cluster.

Note that this is not an exhaustive study of all possible permutations of clusters and samples (which is $k!$ for clusters, and $n_i!$ within each cluster i). Nonetheless, we seek to find typical behaviors that would allow the inference of specific problems that may arise during incremental unsupervised learning: iCVIs should help the practitioner to identify issues by yielding good values when correctly partitioning and bad values when problems occur. Particularly, the observations from case (1) are used as a reference behavior (or default) to which cases (2) and (3) are compared. The overarching goal is to observe the capabilities of the iCVIs in identifying anomalous behaviors caused by *deliberately* generated problems (under- and over-partitions).

5.1. CORRECT PARTITIONS

Assume that a suitable clustering algorithm was selected and optimally parameterized, thus yielding correct data partitions when presenting samples in a given cluster-by-cluster ordering. Since, this study's goal is not to compare the merits of any particular incremental clustering algorithm used for data streams, then to emulate the scenario previously described and make the experiments *clustering algorithm agnostic*, we simply cluster, or in reality classify, each sample based on their respective labels and recompute the iCVIs incrementally. This experimental setup relies on the assumption that, if there exists a subset of clustering algorithms that can perfectly cluster a given data set, then at each point in time they must make the same, and correct, sample assignment to clusters. Furthermore, such correct assignments should be reflected by good iCVI values.

For brevity, Figure 2 shows the iCVIs' behaviors when correctly partitioning only the *R15* data set. Note that these figures depict the iCVIs' behaviors immediately following the creation of a second cluster because they usually cannot be computed for a single cluster.

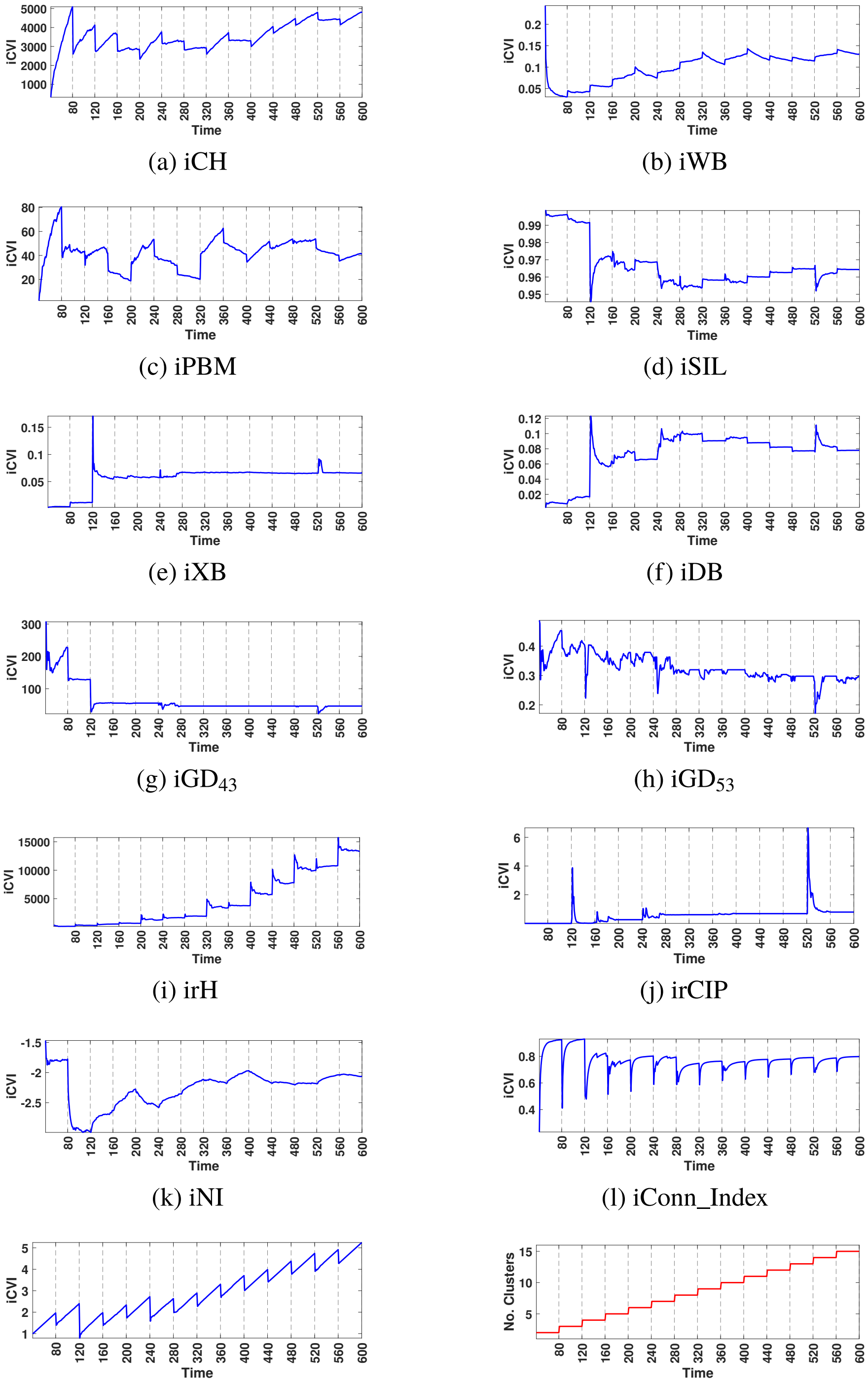
Note how iConn_Index behavior tends to follow an exponential of the form $A(1 - e^{-Bt})$ during the presentation of each cluster in well-behaved data sets. Such response attempts to approach the somewhat step-like behavior of its batch counterpart (see Section 6).

Figure 2 also shows that sudden changes in many iCVI values follow the emergence of new clusters (as expected from previous studies (Ibrahim et al., 2018a; Ibrahim et al., 2019; Ibrahim et al., 2018b; Moshtaghi et al., 2018; Moshtaghi et al., 2019)). During the presentation of samples belonging to a particular cluster, different behaviors can be observed. To identify trends among the iCVIs in a principled manner, in each run of each data set, the following experimental data was collected:

1. The number of times the iCVI increased, decreased and remained constant immediately following the creation of a new cluster (hereafter referred to as immediate behavior).

2. The number of times the iCVI increased, decreased and remained constant during the assignment of samples to the current existing cluster (hereafter referred to as medium term behavior). Particularly, in each time interval corresponding to the presentation of samples belonging to an existing cluster, a simple linear regression model (Kutner et al., 2004) was fit and a t-test was performed for the first order coefficient (slope). If the null hypothesis could be rejected under a 5% significance level, then we observed the first order coefficient's sign: if positive then it was counted as an increasing trend, if negative as a decreasing trend. Otherwise, if the t-test result was not deemed statistically significant then the behavior was accounted for as constant (i.e., no iCVI change).

Both experimental data (1) and (2) were then averaged across 10 runs for each data set. Next, both data were analyzed by adapting the methodology discussed in (Calvo & Santafé, 2016; Trawiński et al., 2012) to our problem. In particular,



(a) iCH

(b) iWB

(c) iPBM

(d) iSIL

(e) iXB

(f) iDB

(g) iGD$_{43}$

(h) iGD$_{53}$

(i) irH

(j) irCIP

(k) iNI

(l) iConn_Index

(m) PS

(n) Number of clusters

Figure 2. (a)-(m) Behaviors of iCVIs (blue curves) when correctly partitioning the data set *R15*. (n) The number of clusters is depicted by the step-like red curve. Each discrete time instant (x-axis) corresponds to the presentation of one sample. The dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another.

1. We performed the Iman–Davenport's version of Friedman's rank sum test to check the hypothesis that these trends are equally typical/probable.
2. If the null hypothesis of the previous test was rejected then we proceeded with a post-hoc test (Bergmann–Hommel's method) to identify the most typical/probable trend.

This analysis was repeated for all iCVIs and the results are summarized in Table 2. We emphasize that the behaviors listed in Table 2 are typical, *not exclusive*. The only iCVIs that consistently behaved following solely the trends showed in Table 2 (i.e., for all data sets, without exceptions) were the iCH, iWB, iConn_Index, iGD$_{53}$, irH and PS for experimental data (1). The iCVIs generally exhibited different trends, but if a single one of them was frequent enough to be deemed as statistically significant, then it is reported in Table 2.

5.2. UNDER-PARTITIONS

Consider a scenario in which a suboptimal clustering algorithm is selected or an appropriate one is badly parameterized such that it yields an under-partition of the data set at hand. For instance, Figure 3a shows an under-partition of the *R15* data set yielded by a fuzzy ART trained under a suboptimal parameter setting and when clusters are presented in the order depicted in Figure 3b. We are interested in how similar scenarios would reflect in the iCVIs behaviors (ideally they should yield poor values) and how strikingly these would deviate from the reference (*i.e.*, according to Table 2). Therefore, we *deliberately* under-partition each data set by randomly merging two close clusters: these are selected with probability proportional to the Euclidean distance between their centroids. In particular, the probability of selecting clusters i and j for merging is given by:

$$p_{i,j} = \frac{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^6}{\sum\limits_{m,n}^{\binom{k}{2}} \|\boldsymbol{v}_m - \boldsymbol{v}_n\|_2^6}, \tag{81}$$

Table 2. Summary of the iCVIs' behaviors when correctly, under, and over-partitioning the synthetic data sets used in the experiments.

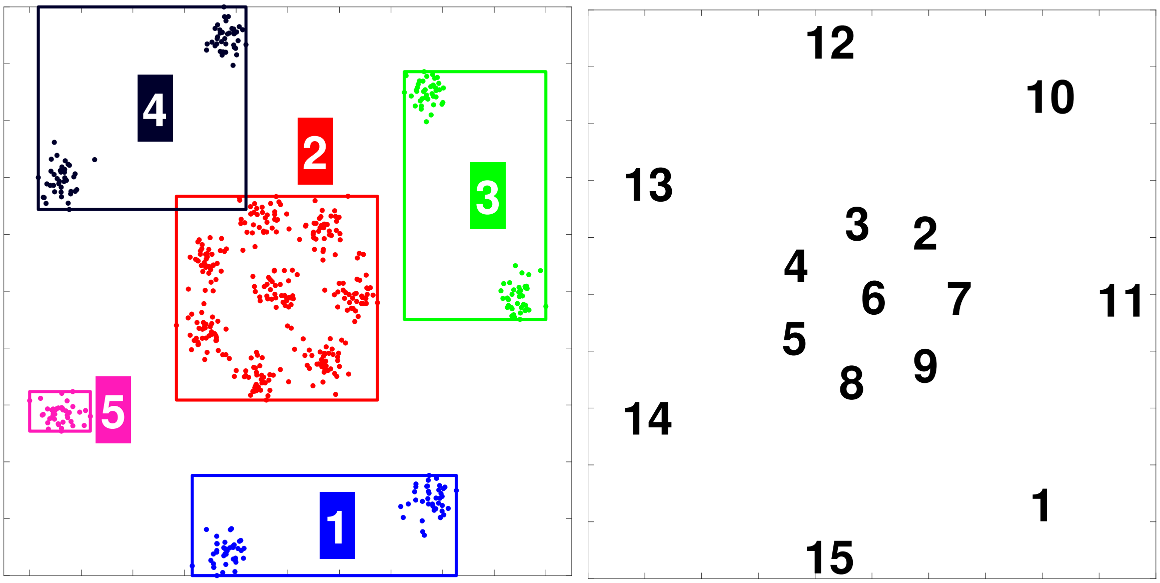
iCVI	Type	Optimality	Correct partition		Under-partition		Over-partition	
			immediate	medium term	immediate	merging	immediate	split
iCH	SS-based	max-optimal	Decrease	Increase	Decrease	**Decrease**	Decrease	Increase
iPBM		max-optimal	Decrease	Increase	Decrease	**Decrease**	Decrease	-
iSIL		max-optimal	Increase	Decrease	Decrease	*Decrease*	**Decrease**	*Decrease*
iWB		min-optimal	Increase	Decrease	Increase	**Increase**	Increase	Decrease
iXB		min-optimal	Decrease	-	Increase	*Increase*	**Increase**	*Increase*
iDB		min-optimal	Decrease	Increase	Increase	*Increase*	**Increase**	*Increase*
iGD$_{43}$		max-optimal	Constant	Increase	-	**Decrease**	**Decrease**	**Decrease**
iGD$_{53}$		max-optimal	Decrease	Increase	-	**Decrease**	Decrease	Increase
PS		max-optimal	Decrease	Increase	-	**Decrease**	Decrease	Increase
iNI[a]	IT-based	min-optimal	Decrease	-	Increase	*Increase*	-	*Increase*
irCIP[a]		min-optimal	Increase	-	Increase	*Increase*	Increase	*Increase*
irH[a]		max-optimal	Increase	Decrease	Decrease	*Decrease*	Increase	Decrease
iConn_Index	graph-based	max-optimal	Decrease	Increase	Decrease	Increase	Decrease	**Decrease**

[a] The experiments with these iCVIs did not include the high dimensional synthetic data sets of *Dim512* and *Dim1024* given the issues associated with the reliable estimation of covariance matrices in high dimensional spaces under small sample sizes.

Empty cells ("-"): these stand for inconclusive; no single trend stood out in a statistically significant manner (i.e., no statistical difference was observed among either the two highest ranked behaviors or among all three behaviors).

Under-partition: behaviors in **boldface** are useful indicators of problems occurring during online clustering, as they differ from their respective counterparts when correctly partitioning the synthetic data sets used in this study; behaviors in *italic* are also useful indicators when they "dominate" the "natural" tendencies of their respective iCVIs.

Over-partition: behaviors in **boldface** are useful indicators of problems occurring during online clustering, as they differ from their respective counterparts when correctly partitioning the synthetic data sets used in this study; behaviors in *italic* are also useful indicators when very pronounced.



(a) Data partition

(b)

Figure 3. (a) An under-partition of the data set *R15* by fuzzy ART. (b) Presentation order of the clusters.

where the 6^{th} power is used for contrast enhancement. After a cluster pair is selected, they are assigned the same label during the online computation of the iCVIs. It is reasonable to assume that a clustering algorithm might allocate samples from close clusters together rather than those from clusters farther apart. Equation (81) is used to avoid repeatedly merging the same two closest clusters in all runs.

For brevity, Figure 4 shows the iCVIs' behaviors when under-partitioning only the *R15* data set. The gray shaded areas shown in these figures correspond the exact time interval in which samples from different clusters are merged, and thus the total number of clusters remains constant. Note that the merged clusters are not necessarily consecutive, given that the sequence of clusters is randomized.

To identify under-partitioning trends among the iCVIs in a principled manner, in each run of each data set, the following data was collected:

1. The number of times the iCVI increased, decreased and remained constant immediately following the first merged sample (hereafter referred to as immediate behavior).

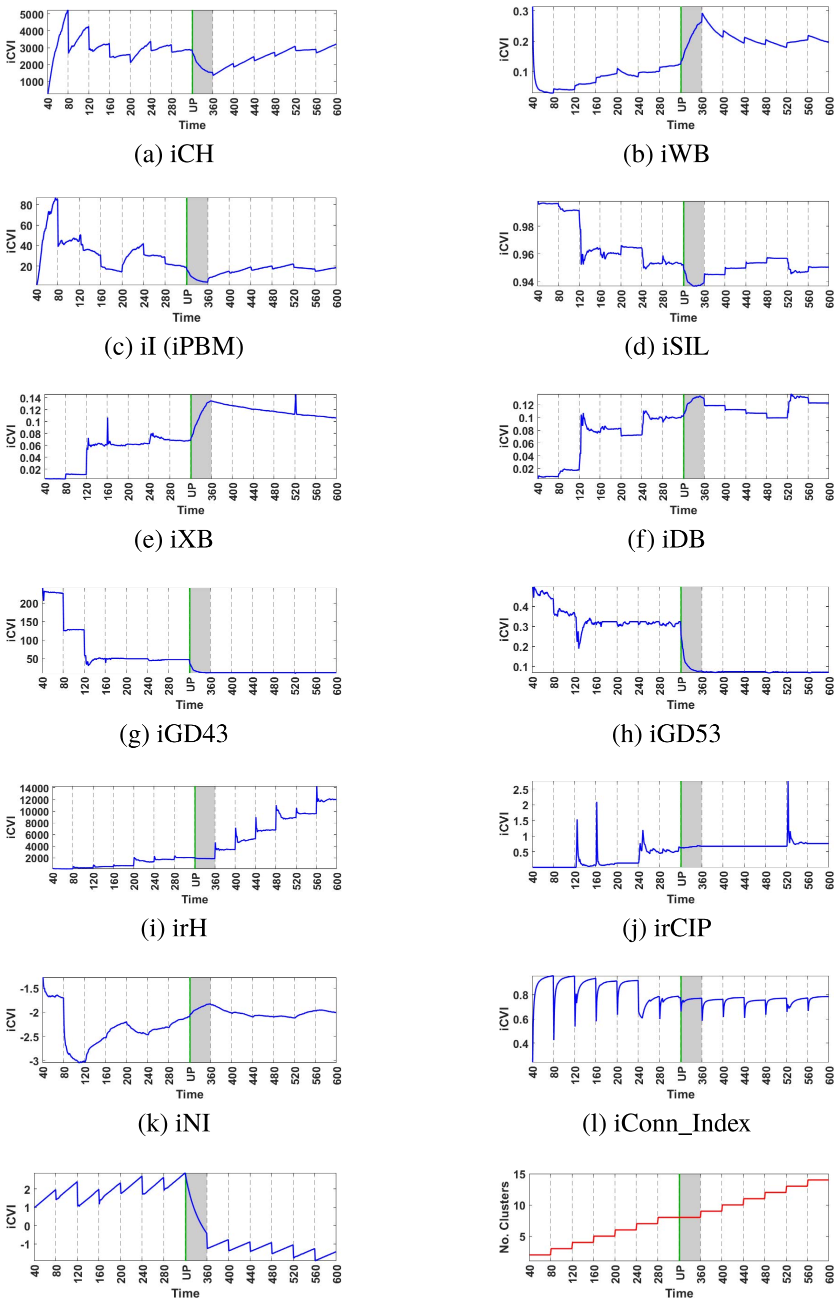
2. The number of times the iCVI increased, decreased and remained constant during the incorrect assignment of samples, *i.e.*, during merging (hereafter referred to as medium term behavior).

The procedures discussed in Section 5.1 were used to obtain the experimental data (2) and to perform the statistical comparison among trends. The results obtained, which are summarized in Table 2, show that:

1. All iCVIs consistently worsened while the algorithm incorrectly agglomerated samples from different clusters (behavior during merging). The exception is the iConn_Index, for which an overall increasing trend was deemed statistically significant. Additionally, compared to the correct partition experiment *while under constant number of clusters*, the iCH, iPBM, iWB, iGD_{43}, iGD_{53} and PS have opposite behavior, which is a strong indication of the occurrence of this problem in the clustering process.

2. Immediately after starting to incorrectly merge clusters (*i.e.*, first merged sample), the performances of most iCVIs are typically accompanied by a change toward worse values under constant number of clusters. The exceptions are iGD_{43}, iGD_{53} and PS, which did not exhibit a statistically significant immediate behavior across our experiments.

3. Although iSIL's and iDB's trends during merging are similar to the correct partition case under constant number of clusters, it is still possible to infer the under-partition issue, since in many cases a sudden and pronounced worsening of these iCVIs was observed as a defining characteristic following such problem. Many of these worsening trends during merging "dominate" the "natural" worsening tendencies of



(a) iCH (b) iWB

(c) iI (iPBM) (d) iSIL

(e) iXB (f) iDB

(g) iGD43 (h) iGD53

(i) irH (j) irCIP

(k) iNI (l) iConn_Index

(m) PS (n) Number of clusters

Figure 4. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *R15* during the under-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the under-partition (UP) problem starts: the samples delimited by the gray shaded interval are assigned to an existing cluster, instead of forming a new one. (a)-(m) Behaviors of iCVIs (blue curves). (n) Number of clusters (step-like red curve).

these iCVIs. However, there are instances in which the challenge relies in patently identifying, without any external knowledge, how much the relative worsening would actually constitute a problem. The latter issue can potentially affect many iCVIs; for instance it is also present in irCIP and irH. The statistically significant trends of the latter iCVI is also the same as the ones for correct partitions. Although no direct comparison to the correct partitioning case is possible for iXB, a similar aggressive worsening behavior was observed in many cases; thus, analogous conclusions and caveats apply.

In summary, a worsening iCVI trend *under constant number of clusters* is an indication that the clustering algorithm might be mistakenly grouping the samples under the same cluster umbrella, and thus should trigger the practitioner's attention. However, it is important to be cautious with respect to false positives because even when a correct partition was retrieved in the experiments of Section 5.1, some iCVIs exhibited large fluctuations while assigning samples of some data sets to their correct cluster (number of clusters is constant in that interval), as well as false negatives, givent that the behaviors listed in Table 2 are typical, *not exclusive*. As a general recommendation, abrupt changes toward worse values of an iCVI under constant number of cluster should be carefully examined. Also, as pointed out in (Ibrahim et al., 2019), it is recommended to observe more than one iCVI. This is even more important to reliably detect under-partition.

5.3. OVER-PARTITIONS

Finally, consider a scenario in which a suboptimal clustering algorithm is selected or an appropriate one is badly parameterized such that the data set at hand is over-partitioned. For instance, Figure 5a shows an over-partition of the *unbalance* data set yielded by standard fuzzy ART (the clusters were presented in the order depicted in Figure 5b), which is suboptimal given that the global vigilance parameter (ρ) assumes equally sized clusters. We are interested in how over-partition would reflect in the iCVIs behaviors (ideally they should yield poor values) and how strikingly these would deviate from their expected behaviors when correct partitions are detected (*i.e.*, the "references" according to Table 2). Therefore, we *deliberately* over-partition each data set by splitting one of its clusters. A cluster is chosen for splitting with probability proportional to its size, thus favoring the selection of large clusters. It is reasonable to assume that certain clustering algorithms, such as standard ART-based ones, would split large clusters according to their parameterization (e.g., the problem depicted in Figure 5). A cluster size is measured by the smallest hyperrectangle that encloses all of its points. Thus, cluster i's hyperrectangle size R_i is measured as (Carpenter et al., 1991):

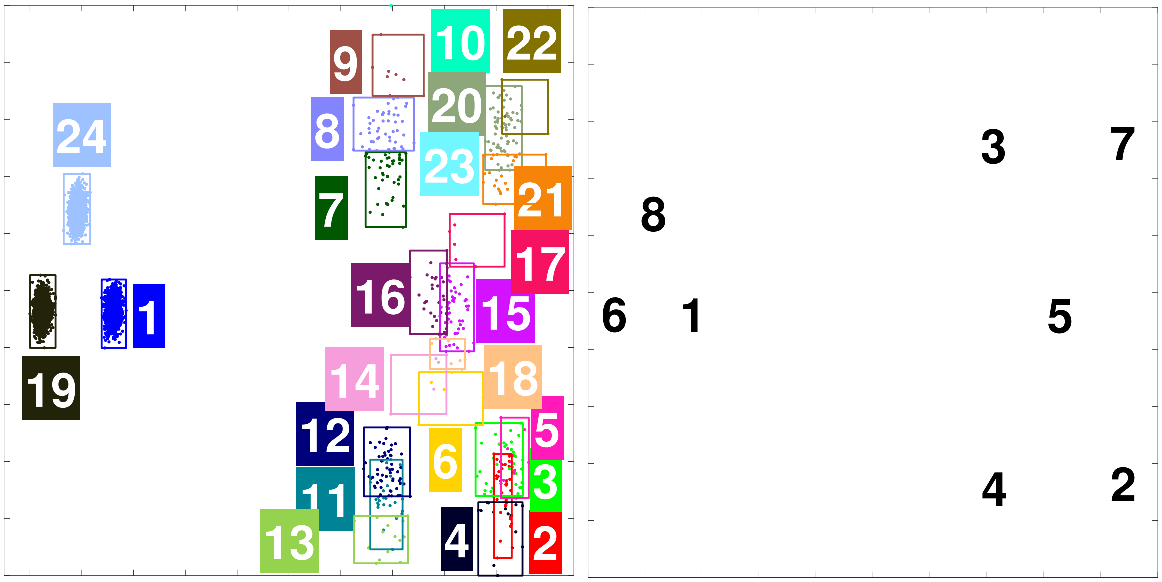
$$R_i = d - \| \bigwedge_{\boldsymbol{I}_j \in \omega_i} \boldsymbol{I}_j \|_1, \tag{82}$$

where $\boldsymbol{I}_j$ is the complement coded version of $\boldsymbol{x}_j$ (see Eq. (60)). To avoid splitting large clusters with small number of samples (n) and consequently permit a better observation of the iCVIs behaviors during over-partition, if $n_i < 10$ then R_i was set to 0.

Naturally, some method must be employed to split a cluster. That is why the over-partition experiment is not completely clustering algorithm agnostic: fuzzy ART was used to create the over-partition. Therefore, results might be somewhat biased toward fuzzy ART solutions. For clarity, the selected clusters were split only into two sub-clusters. In particular, for each selected cluster, its samples were shuffled and fed to fuzzy ARTs trained for 1 epoch (*i.e.*, online mode) with progressively larger vigilance parameter (ρ) values until a solution with 3 clusters was found, in which case the vigilance parameter sweep was stopped. The vigilance values for the fuzzy ART trained with that specific sample order were successively increased using

$$\rho(t+1) = \frac{1}{C(t+1)}(\rho(0) + C(t+1) - 1), \tag{83}$$

$$C(t+1) = C(t) + \delta, \tag{84}$$



(a) Data partition

(b)

Figure 5. (a) An overpartition of the data set *unbalance* by fuzzy ART. (b) Presentation order of the clusters.

where $\rho(0) = 1 - \frac{R_s}{d}$, R_s is the size of the selected cluster s (Eq. (82)), $C(0) = 1$ and δ is the step size which was set to 0.001. From Eqs. (83) and (84) the constraint on the subclusters' sizes becomes increasingly more strict as ρ increases. For instance, $C = 2$ would correspond to a maximum category size equal to half the size of the selected cluster (Tscherepanow, 2010). However given the ordering effects, the value selected for C is not necessarily equal to 2; hence the necessity of the vigilance parameter sweep, which is defined following the strategy described in Eqs. (83) and (84). This process was repeated for 10 random orders and the clustering solution that yielded the most balanced two subclusters was used in the over-partition experiment. This strategy was followed to (i) create a realistic over-partition case for that cluster in online unsupervised learning mode, (ii) facilitate the observation of over-partition behaviors and (iii) avoid the creation of singletons. The over-parition experiment then proceeds like the previous sections, but using fuzzy ART's labels for the split cluster during the online computation of the iCVIs.

For brevity, Figure 6 shows the iCVIs' behaviors when over-partitioning only the *R15* data set. The gray shaded areas shown in these figures correspond to the time interval in which samples belonging to the same cluster are split into two subclusters. Note that the subclusters' samples are randomly presented,*i.e*., they are *not* presented in a subcluster-by-subcluster manner.

To identify over-partitioning trends among the iCVIs in a principled manner, in each run of each data set, the following data was collected

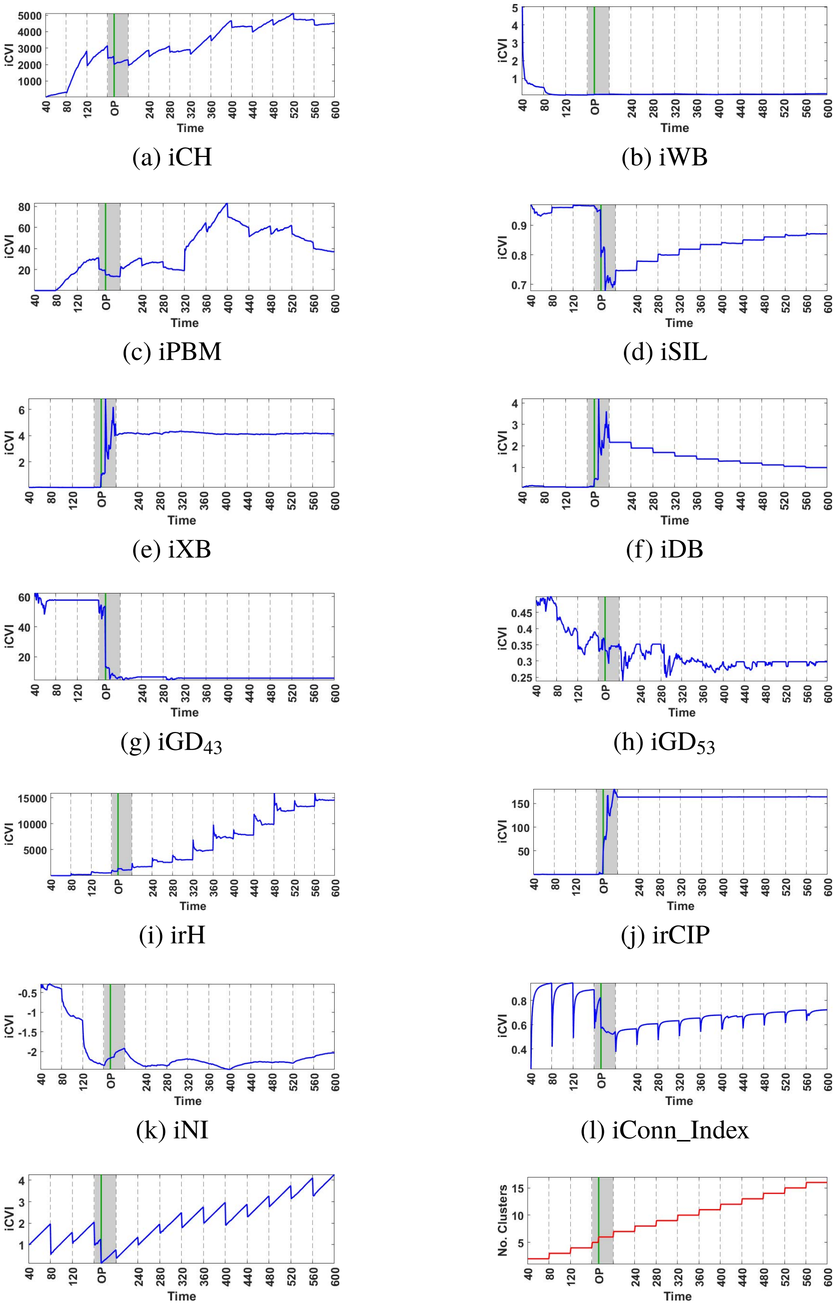
1. The number of times the iCVI increased, decreased and remained constant immediately following a clusters' split (hereafter referred to as immediate behavior).

2. The number of times the iCVI increased, decreased and remained constant following the over-partition of the large cluster (hereafter referred to as medium term behavior).

The procedures discussed in Section 5.1 were used to obtain the experimental data (ii) and to perform the statistical comparison among trends. The results are summarized in Table 2, and show that:

1. The iCH, iWB, PS, iGD_{53} and irH typical behaviors are usually indiscernible from the the ones expected when accurately partitioning during both the (incorrect) creations of new clusters as well as during the presentation of samples belonging to the current cluster. Additionally, the iPBM only exhibited one typical behavior, namely for the creation of a new cluster event, which was again identical to the correct partition case. Therefore, these iCVIs did not seem suitable to identify over-partitions.

2. The iSIL, iXB and iDB only deviate partially, ı.e. they deviate for one trend, particularly the creation of a new cluster when incorrectly splitting a cluster. Although iSIL and iDB typical trends during the cluster split are identical to the correct partition case, and no direct comparison for iXB is possible, for many data sets they underwent a pronounced worsening of these iCVIs values during the split. Similarly to



(a) iCH

(b) iWB

(c) iPBM

(d) iSIL

(e) iXB

(f) iDB

(g) iGD$_{43}$

(h) iGD$_{53}$

(i) irH

(j) irCIP

(k) iNI

(l) iConn_Index

(m) PS

(n) Number of clusters

Figure 6. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *R15* during the over-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the over-partition (OP) problem starts in the cluster delimited by the gray shaded interval. (a)-(m) Behaviors of iCVIs (blue curves). (n) Number of clusters (step-like red curve).

the discussion in the under-partitioning case, defining how much worsening would become a problem can be subjective, especially in borderline cases and with no additional information. Nevertheless, these iCVIs show indication of over-partitioning problems.

3. The irCIP exhibits the same typical trend following the presentation of the first sample of the second subcluster and no direct comparison to the correct partition scenario is possible. However, like the iSIL, iXB and iDB, the irCIP usually undergoes a noticeable worsening during the splitting of the cluster.

4. The iGD$_{43}$ and iConn_Index were the only iCVIs that exhibited trends opposite to their correct partition experiments counterparts, thereby providing a strong indication of over-partition over time. Moreover, when clustering well-behaved data sets such as *dim032* through *dim1024*, the iConn_Index does not follow its characteristic exponential curve (expected from correct partitions) after the erroneous creation of a new cluster and subsequent incorrect assignment of samples. In turn, the iGD$_{43}$ was the only iCVI that exhibited opposite tendencies for both the emergence of a new cluster and the posterior assignment of samples.

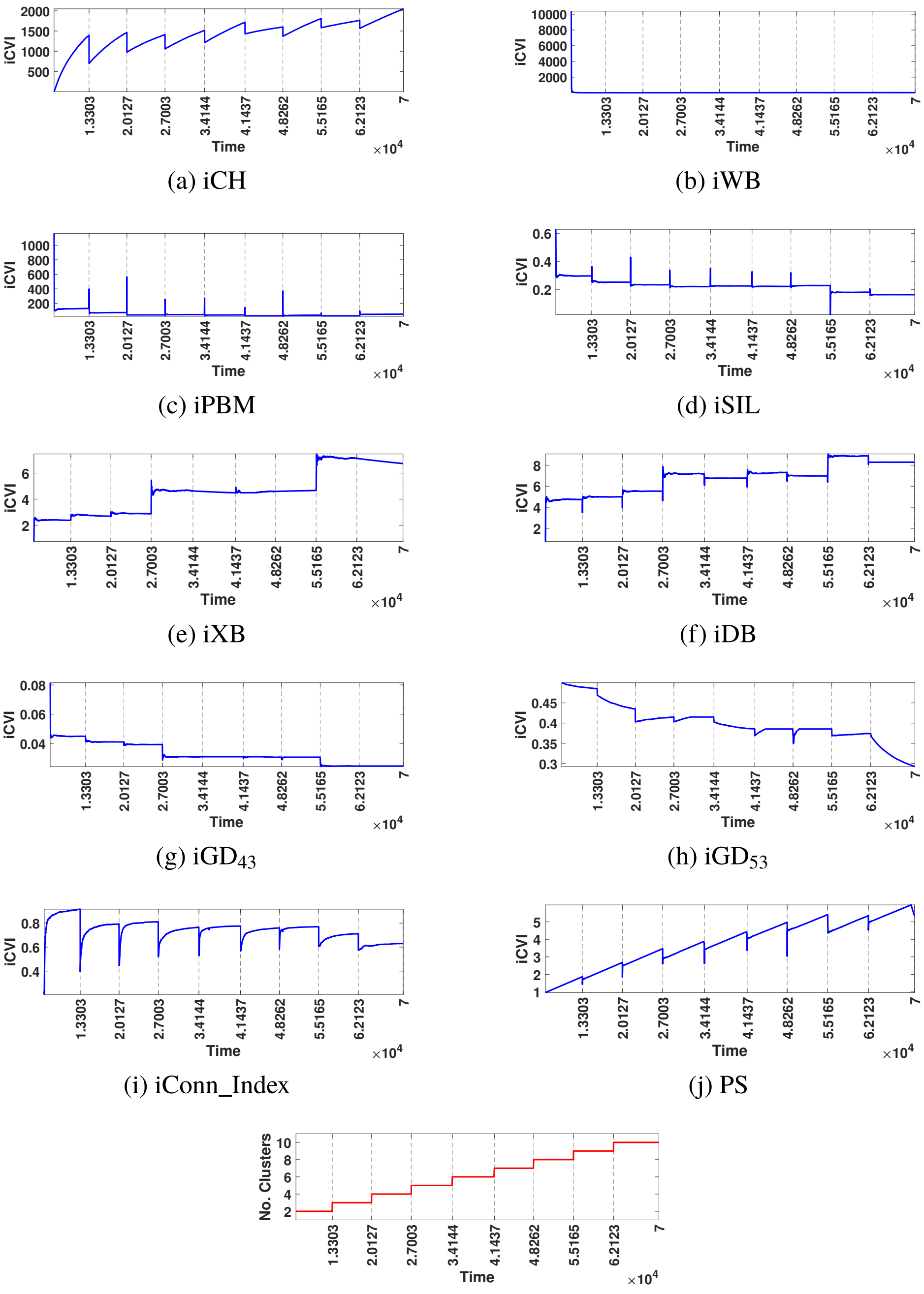
In summary, 6 out of the 13 of the iCVIs (iCH, iPBM, iWB, iGD$_{53}$, PS, and irH) did not provide distinctive insights to definitively detect over-partition problem. In this scenario, unless there was additional a priori information (*e.g.*, the cardinality of clusters) to detect a premature partition, these iCVIs were unable to patently identify over-partition based on their immediate and/or medium term behaviors. On the other hand, 5 iCVIs (iSIL, iXB, iDB, irCIP, and iNI) hinted on over-partition in regard to either immediate behavior and/or a considerable worsening of their medium term behaviors (which were the same as the correct partition scenario for some of these iCVIs or a direct comparison was not possible). Finally, 2 iCVis (iGD$_{43}$ and iConn_Index) showed the opposite tendencies to what was expected during the split, thus providing the strongest evidences for this particular problem.

Note that although there is a natural order for the presentation of clusters, the presentation of samples within each cluster is random. Consequently, samples of the over-partitioned cluster are *not* presented in a *subcluster-by-subcluster* manner. This adds another layer of complexity and thus makes this problem even more challenging. Also note that the vast majority of behaviors are typical, not deterministic, thus we strongly recommend the practitioner to observe a number of iCVIs in order to avoid detection of over-partition false positives/negatives.

5.4. EXPERIMENTS WITH REAL-WORLD DATA SETS

In light of the results obtained for the synthetic data sets, in this section we analyze the scenarios of correct, under- and over-partition performed with the real world data sets of *MNIST* and *Isolet*. The experiments were carried out under the same settings previously described. The discussion in this section is based on the observation of each trend's frequency of occurrence for these two data sets across 10 runs. For brevity, Figs. 7 through 9 illustrate the iCVIs' behaviors under correct, under-, and over-partition only for the *MNIST* data set. We note that the iNI, irCIP and irH were not observed, given the issues associated with the reliable estimation of covariance matrices in high dimensional spaces.

All iCVIs followed the tendencies described in Table 2 when correctly partitioning both real world data sets, with the exception of iPBM and iGD$_{43}$. The former did not consistently follow either expected trend, whereas the latter only followed the immediate behavior trend. Regarding under-partition scenarios, the iPBM, iSIL, iWB, iDB, iGD$_{43}$, and PS consistently followed the trends listed in Table 2, wheras the iXB, iGD$_{53}$, and iConn_Index behaved with varying degrees of agreement; notably, iXB's medium term behavior (merging interval) was coherent with the findings for the synthetic data sets. The



(a) iCH

(b) iWB

(c) iPBM

(d) iSIL

(e) iXB

(f) iDB

(g) iGD$_{43}$

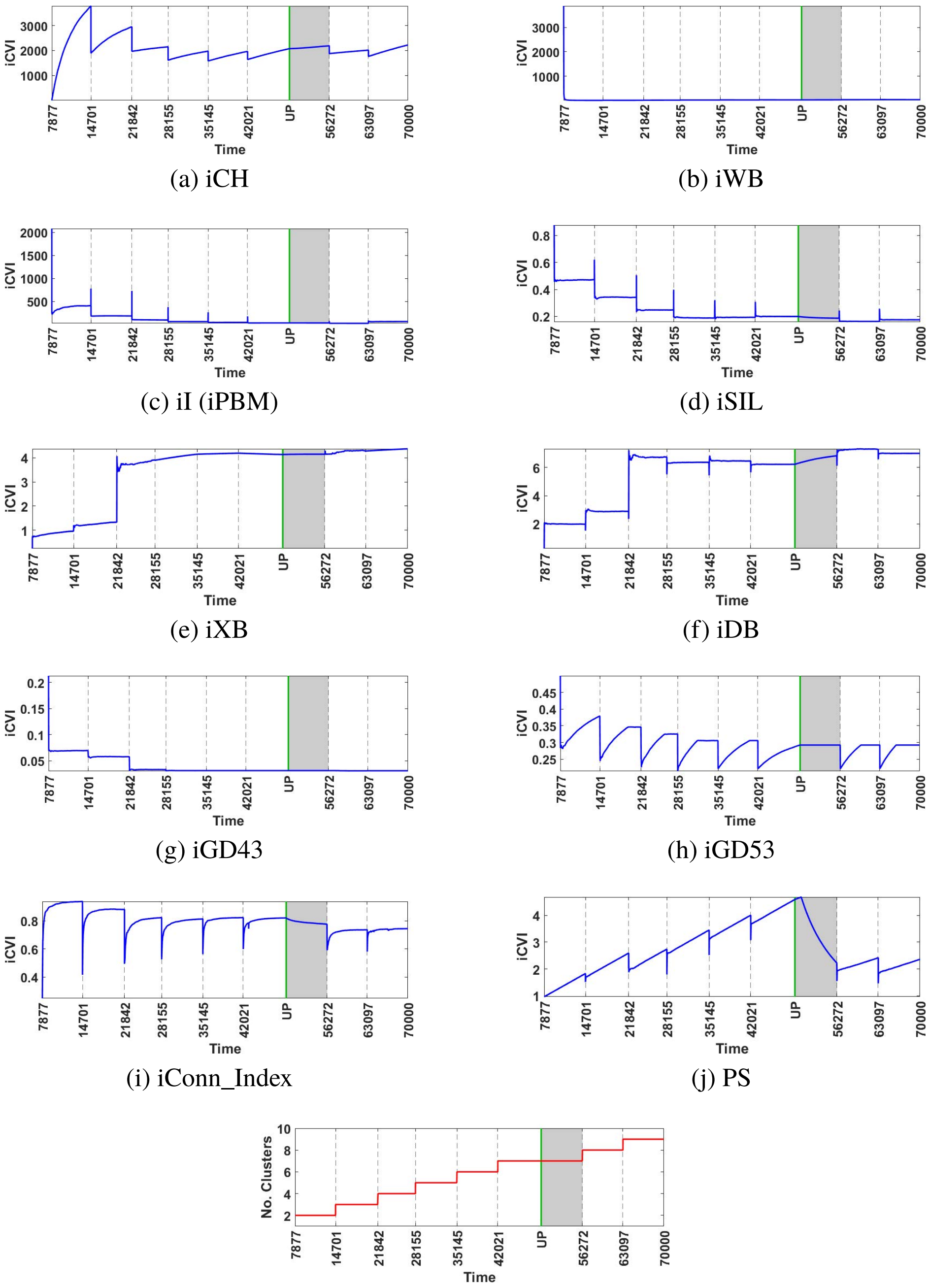
(h) iGD$_{53}$

(i) iConn_Index

(j) PS

(k) Number of clusters

Figure 7. (a)-(j) Behaviors of iCVIs (blue curves) when correctly partitioning the data set *MNIST*. (k) The number of clusters is depicted by the step-like red curve. Each discrete time instant (x-axis) corresponds to the presentation of one sample. The dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another.



(a) iCH

(b) iWB

(c) iI (iPBM)

(d) iSIL

(e) iXB

(f) iDB

(g) iGD43

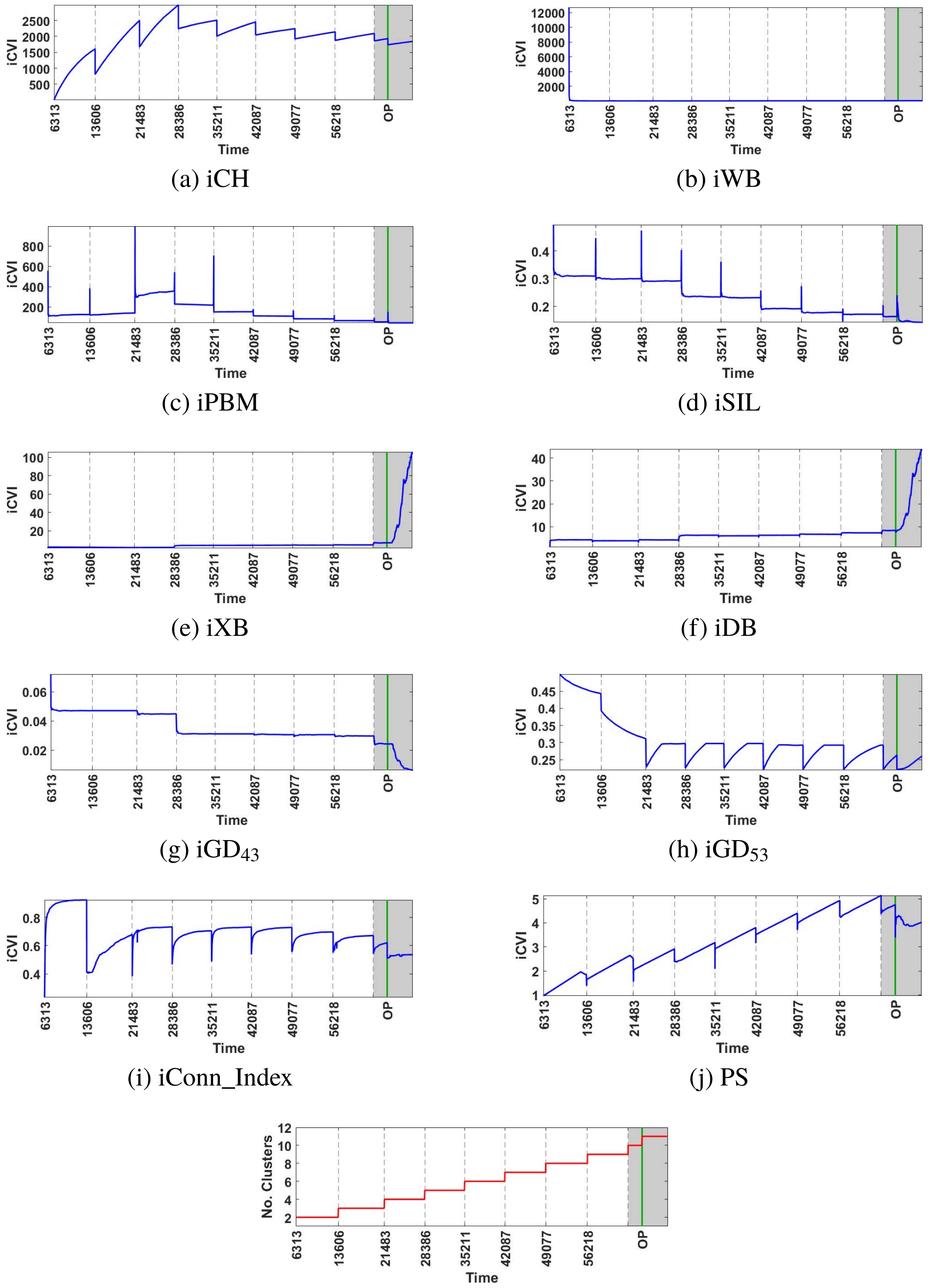
(h) iGD53

(i) iConn_Index

(j) PS

(k) Number of clusters

Figure 8. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *MNIST* during the under-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the under-partition (UP) problem starts: the samples delimited by the gray shaded interval are assigned to an existing cluster, instead of forming a new one. (a)-(j) Behaviors of iCVIs (blue curves). (k) Number of clusters (step-like red curve).



(a) iCH

(b) iWB

(c) iPBM

(d) iSIL

(e) iXB

(f) iDB

(g) iGD$_{43}$

(h) iGD$_{53}$

(i) iConn_Index

(j) PS

(k) Number of clusters

Figure 9. Each discrete time instant (x-axis) corresponds to the presentation of one sample of the data set *MNIST* during the over-partitioning experiment. The black dashed vertical lines delimit consecutive clusters (ground truth), *i.e.*, samples before a dashed line belong to one cluster whereas samples after it belong to another. The green continuous vertical lines indicate the instant in which the over-partition (OP) problem starts in the cluster delimited by the gray shaded interval. (a)-(j) Behaviors of iCVIs (blue curves). (k) Number of clusters (step-like red curve).

iCH was the only iCVI that did not behave as expected. Interestingly, the constant trend was observed for both data sets regarding the immediate behavior of iGD$_{43}$ and iGD$_{53}$. Finally, for the over-partition experiment, the iCH and iGD$_{53}$ followed their expected trends. The remaining iCVIs only partially exhibited the trends on Table 2. In particular, iSIL, iXB, iDB and iGD$_{43}$ were only consistent with their medium term behavior, whereas iWB, iConn_Index and PS were only coherent with their immediate behavior.

Interestingly, for the real world data sets experimented with, over-partition was prominently detected by more iCVIs than under-partition. The latter issue was only patently flagged by the PS CVI. Regarding over-partitions the most visually useful iCVIs were iXB, iDB, iGD$_{43}$ and iConn_Index. We note that although an increasing trend was observed for the latter, as opposed to the synthetic data set findings, the behavior following a cluster split usually does not follow the familiar exponential curve; instead a sharp drop generally follows the split with a small improvement/recovery afterwards. This behavior suggests that there might be an issue with the clustering solution. In such case, the challenge lies in discriminating between correct clusters that naturally don't follow an exponential curve during its evolution and/or determining a drop threshold that would constitute a problem. Similarly, note that iGD$_{43}$ also decreases following the correct creation of some new clusters, thus discriminating among these two events might also be a challenge in some instances.

Finally, note that disagreements to the synthetic data sets' trends listed on Table 2 are to be expected, since those behaviors are typical but not unique. Such variance encourages the observation of several iCVIs to reliably make inferences about the quality of streaming data partitions in real-time.

5.5. VISUALIZATION POWER

In this section we examine a practical aspect of the iCVIs, namely their visualization power in terms of clear hints to problems occurring during the online clustering process, including, but not limited to, substantial variations of their values (in a global scale) over time. Briefly, a useful iCVI behavior should make problems easier to spot. To accomplished this, we visually inspected the iCVIs' curves to gauge their usefulness to detect the under- and over-partition issues that were artificially generated and intentionally inserted in the experiments described in the previous sections. The results of such visual inspection are summarized in Table 3.

For under-partition problems, the PS index consistently provided visually striking cues for both synthetic and real world benchmark data sets. Moreover it was the most robust CVI to increasing levels of cluster overlap (*S1* to *S4* data sets) and number of samples/clusters (*Birch1* and *Birch2*). The iGD$_{43}$, iGD$_{53}$, iXB, iDB, iSIL, iPBM, iCH and iWB (to a lesser extent) were also visually informative for the synthetic data sets, in which tendencies associated to this problem were clearly observable.

Regarding the over-partition problem, the iXB and iDB clearly flagged all over-partition issues. These were followed by iGD$_{43}$, iSIL, iConn_Index and irCIP were also able to flag the majority of cases. As previously discussed, a potential challenge associated with iConn_Index consists of determining which cases not following an exponential behavior during the evolution of samples' assignments should signal a problem and which cases a cluster does not naturally follow such function, since this characteristic is used to detect problems. Similarly, iGD$_{43}$'s caveat is related to determining whether its value decrease is associated with an over-partition problem or a correct emergence of a new cluster: in some correct partition instances the creation a new clusters was also followed by a decrease on this iCVI's value.

6. A CLOSER LOOK AT ICONN_INDEX

When evaluated over time, most iCVIs discussed in this study yield the same values as their batch counterparts (*e.g.*, the the recursive formulation of compactness used in sum-of-squares-based iCVIs is an exact computation, not an approximation (Moshtaghi et al., 2018; Moshtaghi et al., 2019)). The iConn_Index is an exception, and thus is the subject of

Table 3. The iCVIs' that conveyed the most visually informative behavior across the experiments are indicated by a black dot symbol "•" for each data set. The dash symbol "-" indicates that the iCVI was not computed for the corresponding data set.

data set	Under-partition													Over-partition												
	iCH	iPBM	iSIL	iWB	iXB	iDB	iGD_{43}	iGD_{53}	PS	iNI	irCIP	irH	iConn_Index	iCH	iPBM	iSIL	iWB	iXB	iDB	iGD_{43}	iGD_{53}	PS	iNI	irCIP	irH	iConn_Index
Synthetic																										
A3	•	•	•	•	•	•	•	•	•							•		•	•	•				•		
Birch1							•	•	•							•		•	•	•				•		
Birch2									•		•							•	•	•				•		
Dim032	•	•	•	•	•	•	•	•	•							•		•	•	•						•
Dim064	•	•	•	•	•	•	•	•	•							•		•	•	•						•
Dim128	•	•	•	•	•	•	•	•	•		•					•		•	•	•				•		•
Dim256	•	•	•	•	•	•	•	•	•		•					•		•	•	•				•		•
Dim512	•	•	•	•	•	•	•	•	•	-	-	-				•		•	•	•			-	-	-	•
Dim1024	•	•	•	•	•	•	•	•	•	-	-	-				•		•	•	•			-	-	-	•
S1	•	•	•		•	•	•	•	•							•		•	•	•				•		•
S2	•		•		•	•	•	•	•							•		•	•	•				•		•
S3	•	•					•	•	•							•		•	•	•				•		
S4									•							•		•	•	•				•		
Unbalance	•	•	•		•	•			•	•						•		•	•	•				•		•
Aggregation			•		•	•				•						•		•	•					•		•
D31	•		•		•	•	•	•	•							•		•	•	•				•		•
R15	•	•	•	•	•	•	•	•	•	•						•		•	•	•				•		•
Hepta	•	•	•	•	•	•	•	•	•	•	•					•		•	•	•				•		•
Lsun	•	•	•		•	•	•			•	•					•		•	•	•			•	•		•
Tetra	•	•	•		•	•	•	•	•							•		•	•	•			•	•		•
Real World																										
Isolet									•	-	-	-						•	•				-	-	-	•
MNIST									•	-	-	-						•	•	•		•	-	-	-	•

analysis of this section. To obtain the batch Conn_Index values, all first and second winning fuzzy ART prototypes were recomputed after the presentation of each sample based on their activation function values.

For illustration purposes, Figs. 10 through 12 show the evolution of both Conn_Index and iConn_Index for data set *R15* in all three scenarios described in Section 5 in one of the ten experiments. Moreover, they also show a simple linear regression plot of Conn_Index and iConn_Index as well as the final prototypes (hyperrectagles) and their connectivity visualization (CONNvis (Taşdemir & Merényi, 2009)). These show that iConn_Index smoothly follows the overall trends of its batch counterpart (with Pearson correlation coefficients (Bain & Engelhardt, 1992) of 0.80, 0.74 and 0.94 for correct, under- and over-partition scenarios, respectively) which has a more jagged behavior and many plateaus. Also note the faint and permanent connections between several different clusters: these are an artifact of the online learning process since the second closest prototype of a sample that originated a new cluster always belongs to another existing cluster.

Table 4 reports the correlation coefficients and the mean square errors between the incremental and batch versions of the Conn_Index for all data sets averaged across the 10 experiments. For the majority of them, the average correlation between both Conn_Index versions is above (a) 0.75 (correct partitions), (b) 0.70 (under-partitions), and (c) 0.85 (over-partitions). Moreover, for most of data sets, the average mean square error is below 0.02 in all scenarios. Some exceptions include the data sets *Aggregation*, *Lsun*, and *D31* for the correct, under-, and over-partition scenarios, respectively. These have smaller correlation coefficients. Therefore, the effect of fuzzy ARTMAP module A's quantization level on the similarity of the batch and incremental implementations was investigated. This was accomplished by varying modulue A's vigilance parameter ρ_A in the closed interval $[\rho_{min}, \rho_{max}]$, where ρ_{max} is the value listed in Table 4 for the respective data set and $\rho_{min} = \min_i \left(1 - \frac{R_i}{d}\right)$ (i.e., ρ_{min} is computed based on the largest cluster of a given data set, see Eq. (82)), since the interval $[0, \rho_{min}]$ would yield identical results (the vigilance test would always be satisfied). Note that larger values of ρ_A produce finer granularity of clusters' prototypes.

The correlation coefficients and mean squared errors (averaged across 10 runs) depicted in Figure 13 show that carefully tuning the vigilance parameter (granularity level) may improve the average correlation (from 0.5962 to 0.7977 when correctly partitioning the *Aggregation* data set, 0.5792 to 0.6810 when under-partitioning the *Lsun* data set, and 0.8337 to 0.9609 when over-partitioning the *D31* data set); however, its effect on this iCVIs' visualization power when clustering data streams requires further investigation. All these results support the original assumption, stated in Section 3.7, that both versions of the Conn_Index would behave similarly. Therefore, iConn_Index is suitable for assessing the partitions generated by incremental clustering methods.

7. CONCLUSION

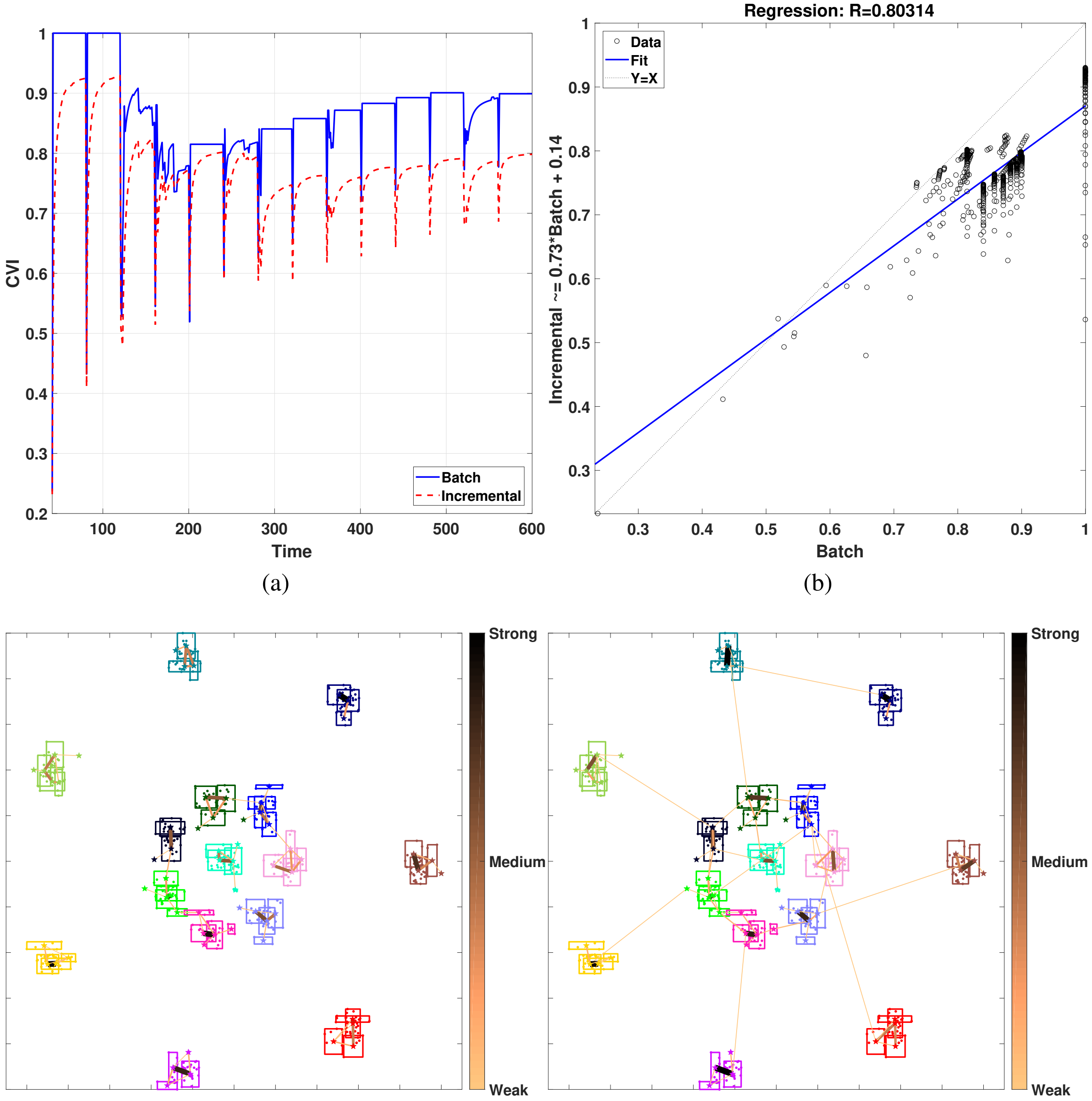
This paper presented incremental versions of 7 cluster validity indices (CVIs), namely, incremental Calinski-Harabasz (iCH), incremental Pakhira-Bandyopadhyay-Maulik (iPBM), incremental Silhouette (iSIL), incremental Negentropy Increment (iNI), incremental Representative Cross Information Potential (irCIP) and Cross Entropy (irH), and incremental Conn_Index (iConn_Index). These and previously developed incremental cluster validity indices (iCVIs) are essential tools at the practitioner disposal: they allow the assessment of the quality of data streams' partitions. By definition, data streams require real-time processing of incoming samples because iterating over the entire data set is either prohibitive or unsuitable for the application.

Furthermore, using an experimental framework as clustering algorithm agnostic as possible and synthetic and real world benchmark data sets, the dynamics of 13 iCVIs were analyzed in 3 different clustering scenarios: correct, under- and over-partitioning. Specifically, a thorough comparative study was performed among the presented iCVIs,

Table 4. Vigilance parameter (ρ) values used in in this study and the average correlation coefficient (R_{avg}) and mean square error (MSE_{avg}) between the incremental and batch Conn_Index curves.

Data set	Vigilance	Correct partition		Under-partition		Over-partition	
	ρ	R_{avg}	MSE_{avg}	R_{avg}	MSE_{avg}	R_{avg}	MSE_{avg}
A3	0.900	0.9511	0.0172	0.9681	0.0125	0.9559	0.0088
Birch1	0.920	0.9387	0.0020	0.8724	0.0026	0.8830	0.0028
Birch2	0.960	0.9869	0.0188	0.9890	0.0195	0.9885	0.0148
Dim032	0.930	0.7682	0.0040	0.7014	0.0068	0.9286	0.0045
Dim064	0.950	0.7654	0.0037	0.6996	0.0084	0.9299	0.0046
Dim128	0.960	0.7855	0.0028	0.7277	0.0056	0.9282	0.0040
Dim256	0.975	0.7838	0.0031	0.6999	0.0072	0.9296	0.0054
Dim512	0.980	0.7852	0.0028	0.7312	0.0044	0.9224	0.0046
Dim1024	0.988	0.7854	0.0030	0.7573	0.0071	0.9252	0.0044
S1	0.900	0.7379	0.0100	0.6884	0.0059	0.8592	0.0060
S2	0.900	0.7675	0.0409	0.7197	0.0402	0.8676	0.0249
S3	0.950	0.9166	0.0098	0.9107	0.0136	0.9083	0.0094
S4	0.950	0.8894	0.0185	0.8810	0.0101	0.8647	0.0153
Unbalance	0.880	0.6699	0.0033	0.7660	0.0034	0.9354	0.0078
Aggregation	0.750	0.5962	0.0114	0.6736	0.0214	0.9133	0.0077
D31	0.900	0.8448	0.0109	0.8977	0.0167	0.8337	0.0140
R15	0.950	0.7037	0.0179	0.7011	0.0154	0.8357	0.0197
Hepta	0.800	0.8089	0.0109	0.7821	0.0134	0.9267	0.0116
Lsun	0.900	0.7882	0.0550	0.5792	0.0350	0.9140	0.0328
Tetra	0.800	0.7459	0.0286	0.7101	0.0155	0.9307	0.0208
Isolet	0.500	0.8072	0.0167	0.7648	0.0167	0.8494	0.0117

the Partition Separation (PS), the incremental Xie-Beni (iXB), the incremental Davies-Bouldin (iDB) and the incremental generalized Dunn's indices 43 and 53 (iGD$_{43}$ and iGD$_{53}$) in order to observe how these iCVIs are affected by the aforementioned problems and thus provide guidelines to aid the practitioner in identifying when these occur during online unsupervised learning. Additionally, it was shown that, although not equal to its batch counterpart, the iConn_Index follows the same general trends.



(a)

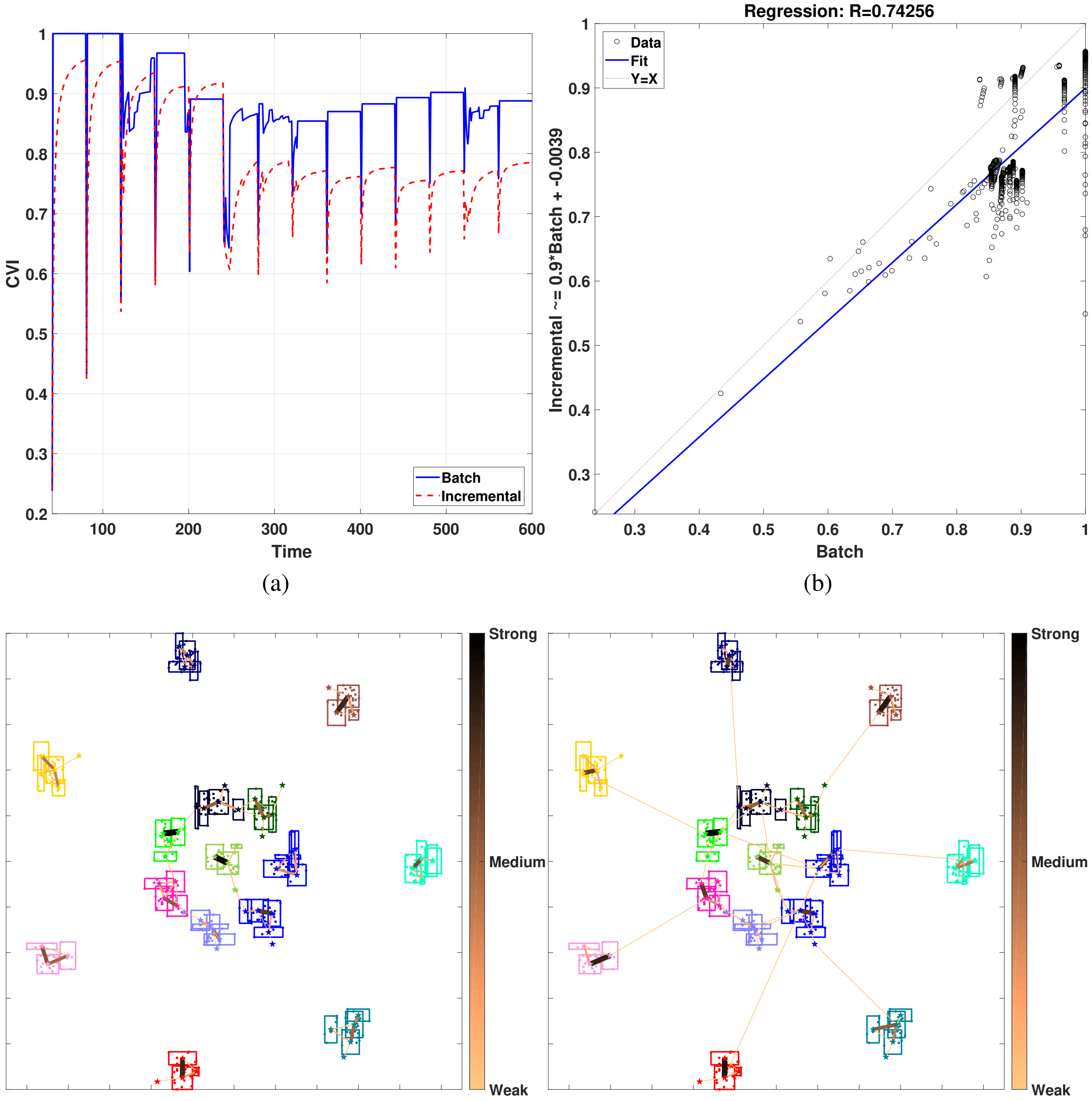
(b)

(c)

(d)

Figure 10. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when correctly partitioning the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental $\boldsymbol{CONN}$ matrices.

As expected from previous studies, most iCVIs undergo abrupt changes following the creation of a new cluster. When samples from an existing cluster are presented, however, each iCVI exhibits a particular behavior, which was taken as a reference to compare the

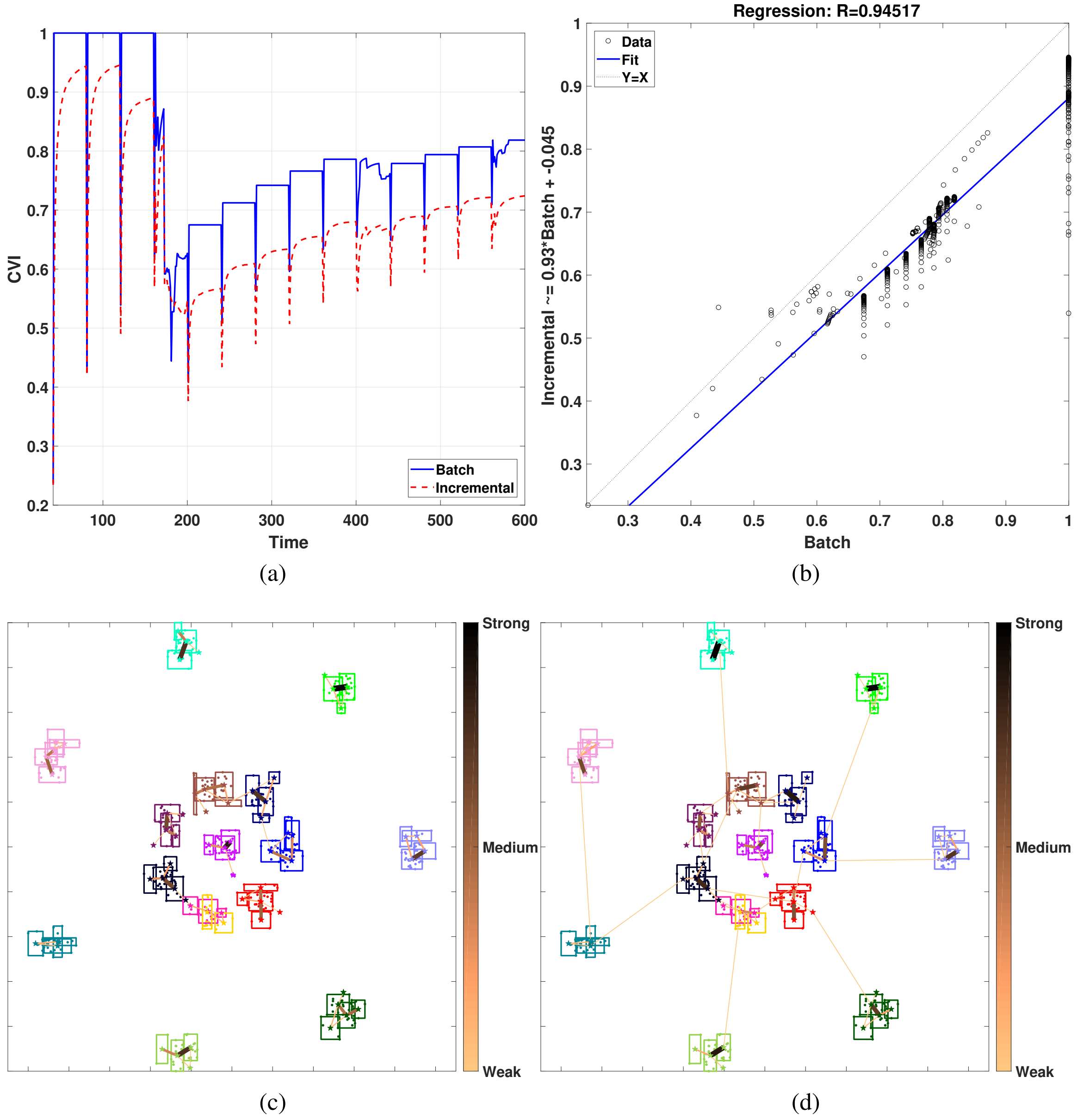


(a) (b)

(c) (d)

Figure 11. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when under-partitioning of the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental ***CONN*** matrices.

cases of under- and over-partitioning a data set. Most iCVIs detected under-partitioning of the synthetic data sets during the incremental clustering process, whereas only a smaller subset of them provided insight to indicate over-partitioning problems. Interestingly, the

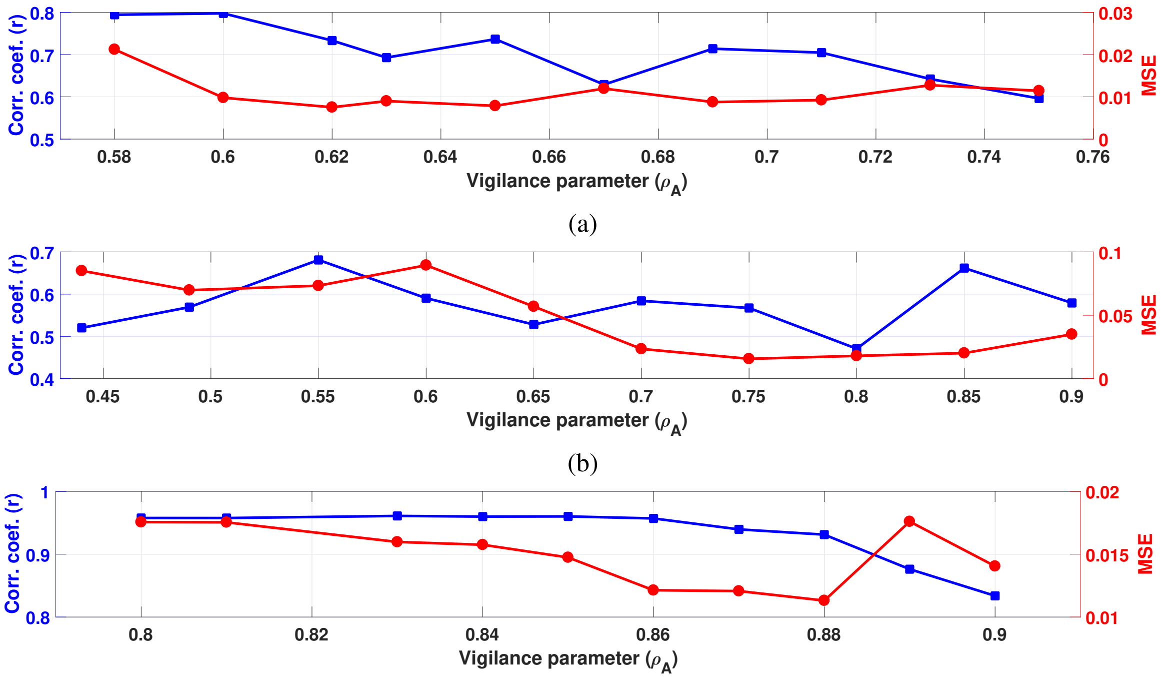


(a) (b)

(c) (d)

Figure 12. (a) Behaviors of Conn_Index (continuous blue line) and iConn_Index (dashed red line) when over-partitioning of the *R15* data set. (b) Regression plot between Conn_Index and iConn_Index in (a). Fuzzy ARTMAP's A-side categories and CONNvis (thicker and darker lines indicate stronger connections) generated with the (c) batch and (d) incremental $\boldsymbol{CONN}$ matrices.

opposite was observed for the real world data sets. According to this study's findings, if the practitioner is expecting under-partition, the PS index was particularly useful for the detection of this type of problem, as well as the following CVIs: iCH, iPBM, iSIL, iWB,



(a)

(b)

(c)

Figure 13. Correlation coefficients and MSEs between the batch and incremental versions of Conn_Index as a function of Fuzzy ARTMAP's module A vigilance parameter. The values shown were averaged across 10 runs for (a) *Aggregation* data set under correct partition experiment, (b) *Lsun* data set under under-partition experiment, and (c) *D31* data set under over-partition experiment.

iXB, iDB, iGD$_{43}$, and iGD$_{53}$. On the other hand, if over-partition issues are of concern, then we recommend iXB, iDB, iGD$_{43}$, iSIL, iConn_Index and irCIP. In any case, we corroborate previous studies' recommendations regarding iCVIs: like their batch counterparts, it is good practice to observe a number of iCVIs' dynamics at any given time, rather than relying on the assessment of solely one. It is expected that the observations from the study presented here will assist in incremental clustering applications such as data streams, as well as the iCVIs MATLAB toolbox package provided.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. James M. Keller and his coauthors for providing an early copy of reference (Ibrahim et al., 2018b). The authors would also like to thank Prof. Gayla Olbricht for the helpful discussions on the statistical analysis performed in this study. This research was sponsored by the Missouri University of Science and Technology Mary K. Finley Endowment and Intelligent Systems Center; the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance code BEX 13494/13-9; the U.S. Dept. of Education Graduate Assistance in Areas of National Need program; and the Army Research Laboratory (ARL), and it was accomplished under Cooperative Agreement Number W911NF-18-2-0260. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Ana, L. N. F., & Jain, A. K. (2003). Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (pp. II–II). volume 2.

Araújo, D., Neto, A. D., & Martins, A. (2013a). Information-theoretic clustering: A representative and evolutionary approach. *Expert Systems with Applications*, *40*, 4190–4205.

Araújo, D., Neto, A. D., & Martins, A. (2013b). Representative cross information potential clustering. *Pattern Recognition Letters*, *34*, 2181 – 2191.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*, 243 – 256.

Bain, L. J., & Engelhardt, M. (1992). *Introduction to Probability and Mathematical Statistics*. (2nd ed.). Brooks/Cole, Cengage Learning.

Bandyopadhyay, S., & Maulik, U. (2001). Nonparametric genetic clustering: comparison of validity indices. *IEEE Trans. Syst., Man, Cybern. C*, *31*, 120–125.

Bezdek, J. C., Li, W. Q., Attikiouzel, Y., & Windham, M. (1997). A geometric approach to cluster validity for normal mixtures. *Soft Computing*, *1*, 166–179.

Bezdek, J. C., & Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *28*, 301–315.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019). A survey of adaptive resonance theory neural network models for engineering applications. *Neural Networks*, .

Brito da Silva, L. E., & Wunsch II, D. C. (2018a). A study on exploiting VAT to mitigate ordering effects in Fuzzy ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2351–2358).

Brito da Silva, L. E., & Wunsch II, D. C. (2018b). An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2595–2613.

Cai, D., He, X., & Han, J. (2011). Speed up kernel discriminant analysis. *The VLDB Journal*, *20*, 21–33.

Cai, D., He, X., Han, J., & Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 1548–1560.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*, 1–27.

Calvo, B., & Santafé, G. (2016). scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems. *The R Journal*, *8*, 248–256.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Comon, P. (1994). Independent component analysis, A new concept? *Signal Processing*, *36*, 287 – 314.

Cottrell, M., & Rousset, P. (1997). The kohonen algorithm: A powerful tool for analysing and representing multidimensional quantitative and qualitative data. In J. Mira, R. Moreno-Díaz, & J. Cabestany (Eds.), *Biological and Artificial Computation: From Neuroscience to Technology* (pp. 861–871). Berlin, Heidelberg: Springer Berlin Heidelberg.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*, 224–227.

Dimitriadou, E., Dolničar, S., & Weingessel, A. (2002). An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, *67*, 137–159.

Dua, D., & Graff, C. (2017). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Dubes, R., & Jain, A. K. (1979). Validity studies in clustering methodologies. *Pattern Recognition*, *11*, 235 – 254.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. (2nd ed.). John Wiley & Sons.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, *3*, 32–57.

Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, *48*, 4743–4759.

Fränti, P., & Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, *39*, 761 – 775.

Fränti, P., Virmajoki, O., & Hautamaki, V. (2006). Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 1875–1881.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*, 1–30.

Gokcay, E., & Principe, J. (2000). A new clustering evaluation function using renyi's information potential. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on* (pp. 3490–3493). volume 6.

Gokcay, E., & Principe, J. C. (2002). Information theoretic clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*, 158–171.

Gordon, A. D. (1998). Cluster Validation. In C. Hayashi, K. Yajima, H.-H. Bock, N. Ohsumi, Y. Tanaka, & Y. Baba (Eds.), *Data Science, Classification, and Related Methods* (pp. 22–39). Tokyo: Springer Japan.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002a). Cluster Validity Methods: Part I. *SIGMOD Rec.*, *31*, 40–45.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002b). Clustering Validity Checking Methods: Part II. *SIGMOD Rec.*, *31*, 19–27.

Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, *10*.

Hruschka, E. R., Campello, R. J., & de Castro, L. N. (2006). Evolving clusters in gene-expression data. *Information Sciences*, *176*, 1898 – 1927.

Hruschka, E. R., de Castro, L. N., & Campello, R. J. G. B. (2004). Evolutionary algorithms for clustering gene-expression data. In *Proc. IEEE Int. Conf. Data Mining (ICDM)* (pp. 403–406).

Ibrahim, O. A., Keller, J. M., & Bezdek, J. C. (2018a). Analysis of streaming clustering using an incremental validity index. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8).

Ibrahim, O. A., Keller, J. M., & Bezdek, J. C. (2019). Evaluating Evolving Structure in Streaming Data With Modified Dunn's Indices. *IEEE Transactions on Emerging Topics in Computational Intelligence*, (pp. 1–12).

Ibrahim, O. A., Shao, J., Keller, J. M., & Popescu, M. (2016). A temporal analysis system for early detection of health changes. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 186–193).

Ibrahim, O. A., Wang, Y., & Keller, J. M. (2018b). Analysis of incremental cluster validity for big data applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *26*, 47–62.

Kärkkäinen, I., & Fränti, P. (2002). *Dynamic local search algorithm for the clustering problem*. Technical Report A-2002-6 Department of Computer Science, University of Joensuu Joensuu, Finland.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, *32*, 68–75.

Kasuba, T. (1993). Simplified Fuzzy ARTMAP. *AI Expert*, *8*, 18–25.

Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2004). *Applied Linear Statistical Models*. (5th ed.).

Lago-Fernández, L. F., & Corbacho, F. (2009). Using the negentropy increment to determine the number of clusters. In J. Cabestany, F. Sandoval, A. Prieto, & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009. Proceedings, Part I* (pp. 448–455). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lago-Fernández, L. F., & Corbacho, F. (2010). Normality-based validation for crisp clustering. *Pattern Recognition*, *43*, 782 – 795.

Lamirel, J., & Cuxac, P. (2015). New quality indexes for optimal clustering model identification with high dimensional data. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (pp. 855–862).

Lamirel, J., Dugué, N., & Cuxac, P. (2016). New efficient clustering quality indexes. In *Proc. Int. Joint Conf. Neural Netw. (IJCNN)* (pp. 3649–3657).

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

Luna-Romera, J. M., del Mar Martínez-Ballesteros, M., García-Gutiérrez, J., & Riquelme-Santos, J. C. (2016). An Approach to Silhouette and Dunn Clustering Indices Applied to Big Data in Spark. In O. Luaces, J. A. Gámez, E. Barrenechea, A. Troncoso, M. Galar, H. Quintián, & E. Corchado (Eds.), *Advances in Artificial Intelligence* (pp. 160–169). Cham: Springer International Publishing.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*, 159–179.

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., & Bailey, J. (2018). Online Cluster Validity Indices for Streaming Data. *ArXiv e-prints*, . ArXiv:1801.02937v1 [stat.ML].

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., & Bailey, J. (2019). Online cluster validity indices for performance monitoring of streaming data clustering. *International Journal of Intelligent Systems*, *34*, 541–563.

Oliveira, A. G., Martins, A., & Neto, A. D. (2018). Information state: A representation for dynamic processes using information theory. In *Proc. Int. Joint Conf. Neural Netw. (IJCNN)* (pp. 1–8).

Oliveira, A. G., Neto, A. D., & Martins, A. (2017). An analysis of information dynamic behavior using autoregressive models. *Entropy*, *19*, 1–13.

Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern Recognition*, *37*, 487 – 501.

Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. (1st ed.). Springer Publishing Company, Incorporated.

Rawashdeh, M., & Ralescu, A. (2012). Center-wise intra-inter silhouettes. In E. Hüllermeier, S. Link, T. Fober, & B. Seeger (Eds.), *Scalable Uncertainty Management* (pp. 406–419). Berlin, Heidelberg: Springer.

Rényi, A. (1961). On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics* (pp. 547–561). Berkeley, Calif.: University of California Press.

Rezaei, M., & Fränti, P. (2016). Set matching measures for external cluster validity. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 2173–2186.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53 – 65.

Silva, J. d. A., & Hruschka, E. R. (2016). A support system for clustering data streams with a variable number of clusters. *ACM Trans. Auton. Adapt. Syst.*, *11*, 11:1–11:26.

Taşdemir, K., & Merényi, E. (2006). Data topology visualization for the Self-Organizing Maps. In *Proc. 14th European Symposium on Artificial Neural Networks (ESANN 2006)* (pp. 277–282).

Taşdemir, K., & Merényi, E. (2007). A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. In *The 2007 International Joint Conference on Neural Networks (IJCNN)* (pp. 2205–2211).

Taşdemir, K., & Merényi, E. (2009). Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *20*, 549–562.

Taşdemir, K., & Merényi, E. (2011). A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*, 1039–1053.

Trawiński, B., Smętek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, *22*, 867 – 881.

Tscherepanow, M. (2010). TopoART: A Topology Learning Hierarchical ART Network. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 157–167). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tyree, E. W., & Long, J. A. (1999). The use of linked line segments for cluster representation and data reduction. *Pattern Recognition Letters*, *20*, 21 – 29.

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1273–1280.

Vendramin, L., Campello, R. J. G. B., & Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, *3*, 209–235.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*, 586–600.

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*, 1628–1644.

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, *11*, 2837–2854.

Wang, W., & Zhang, Y. (2007). On fuzzy cluster validity indices. *Fuzzy Sets and Systems*, *158*, 2095 – 2117.

Williamson, J. R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, *9*, 881 – 897.

Wunsch II, D. C. (2009). ART properties of interest in engineering applications. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 3380–3383).

Xie, X. L., & Beni, G. (1991). A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, *13*, 841–847.

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., Xu, J., & Wunsch, D. C. (2012). A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*, 1243–1256.

Yang, M.-S., & Wu, K.-L. (2001). A new validity index for fuzzy clustering. In *10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297)* (pp. 89–92). volume 1.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338 – 353.

Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A New Data Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery*, *1*, 141–182.

Zhao, Q., & Fränti, P. (2014). WB-index: A sum-of-squares based index for cluster validity. *Data & Knowledge Engineering*, *92*, 77 – 89.

Zhao, Q., Xu, M., & Fränti, P. (2009). Sum-of-Squares Based Cluster Validity Index and Significance Analysis. In M. Kolehmainen, P. Toivanen, & B. Beliczynski (Eds.), *Adaptive and Natural Computing Algorithms* (pp. 313–322). Berlin, Heidelberg: Springer Berlin Heidelberg.

SECTION

2. SUMMARY AND CONCLUSIONS

This dissertation examined the unsupervised learning tasks of data visualization, clustering, and validation. Specifically, the first part investigated the following aspects of performing clustering with adaptive resonance theory (ART) neural networks: (1) input order dependency, (2) distributed representations, (3) detection of clusters with complex structures, (4) clustering with a reinforcement signal, and (5) category proliferation. In order to handle the first problem, frameworks were presented in terms of pre-processing by profiting from VAT sorting and in terms of post-processing by cascading a Merge ART module. To address the second problem, two novel fuzzy ART-based architectures were engineered, namely the dual vigilance fuzzy ART and and the distributed dual vigilance fuzzy ART, where the latter possesses higher-order distributed representation according to hierarchical agglomerative clustering. Finally, cluster validity indices were embedded in the fuzzy ART model via a second vigilance test, thus enabling immediate feedback to direct the dynamics of the system during offline learning. When employed, all of these augmentations were able to improve performance, and most of them create more compact networks (i.e., reduce category proliferation) compared to baseline ART models. Furthermore, the performance of the novel architectures that were designed were either on par or superior to current state-of-the-art fuzzy ART-based models conceived for the clustering task. The second part of this dissertation explored multivariate data visualization with self-organizing map (SOM) neural networks. Particularly, an image-based visualization following the unified distance matrix structure was devised by combining Renyi's quadratic cross-entropy and a single-linkage-based k-nearest neighbors. These enhancements were capable of improving between-cluster contrast and within-cluster uniformness. Furthermore, the visualization capabilities of the novel approach were either comparable or superior to classic and state-of-the-art SOM-based visualizations.

The third and final part of this dissertation expanded the current set of incremental cluster validity indices (iCVIs) by developing online versions of 1 graph-, 3 information-theoretic- and 4 sum-of-squares-based iCVIs. These, along with other existing iCVIs, were tested for their performance tracking capabilities when deliberately creating erroneous partitions. After establishing as reference the iCVIs' trends associated with correct partitions, experimental results suggested that the Partition Separation (PS) as well as the incremental versions of Calinski-Harabasz (iCH), Pakhira-Bandyopadhyay-Maulik (iPBM), Silhouette (iSIL), WB index (iWB), Xie-Beni (iXB), Davies-Bouldin (iDB), and generalized Dunn's indices 43 and 53 (iGD_{43} and iGD_{53}) are the most useful to visually detect under-partition problems when clustering data streams. On the other hand, the iXB, iDB, iGD_{43}, iSIL, incremental Conn_Index and incremental Representative Cross Information Potential (irCIP) are the most useful to flag over-partition issues.

REFERENCES

Al-Daraiseh, A., Georgiopoulos, M., Anagnostopoulos, G., Wu, A. S., & Mollaghasemi, M. (2006). GFAM: A Genetic Algorithm Optimization of Fuzzy ARTMAP. In *IEEE International Conference on Fuzzy Systems* (pp. 315–322).

Amis, G. P., & Carpenter, G. A. (2007). Default ARTMAP 2. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 777–782).

Amis, G. P., & Carpenter, G. A. (2010). Self-supervised ARTMAP. *Neural Networks*, *23*, 265 – 282.

Amorim, D. G., Delgado, M. F., & Ameneiro, S. B. (2007). Polytope ARTMAP: Pattern Classification Without Vigilance Based on General Geometry Categories. *IEEE Transactions on Neural Networks*, *18*, 1306–1325.

Amorim, D. G., Delgado, M. F., Ameneiro, S. B., & Amorim, R. R. (2011). Evolução das Redes ART e suas Funcionalidades. *Revista OPARA*, *1*, 40 – 59.

Ana, L. N. F., & Jain, A. K. (2003). Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (pp. II–II). volume 2.

Anagnostopoulos, G. C., Bharadwaj, M., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2003). Exemplar-based pattern recognition via semi-supervised learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2782–2787). volume 4.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 59–64). volume 6.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001a). Ellipsoid ART and ARTMAP for incremental clustering and classification. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1221–1226). volume 2.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001b). Ellipsoid ART and ARTMAP for incremental unsupervised and supervised learning. In *Aerospace/Defense Sensing, Simulation, and Controls* (pp. 293–304). International Society for Optics and Photonics.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2002). Category regions as new geometrical concepts in Fuzzy-ART and Fuzzy-ARTMAP. *Neural Networks*, *15*, 1205 – 1221.

Anagnostopoulos, G. C., & Georgiopoulos, M. (2003). Putting the Utility of Match Tracking in Fuzzy ARTMAP Training to the Test. In V. Palade, R. J. Howlett, & L. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 1–6). Berlin, Heidelberg: Springer Berlin Heidelberg.

Anagnostopoulos, G. C., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2002a). Boosted ellipsoid ARTMAP. In *Proc. SPIE*. volume 4739.

Anagnostopoulos, G. C., Georgiopoulos, M., Verzi, S. J., & Heileman, G. L. (2002b). Reducing generalization error and category proliferation in ellipsoid ARTMAP via tunable misclassification error tolerance: boosted ellipsoid ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2650–2655). volume 3.

Andonie, R. (1990). A Converse H-theorem for Inductive Processes. *Comput. Artif. Intell.*, *9*, 161–167.

Andonie, R., & Sasu, L. (2003). A Fuzzy ARTMAP Probability Estimator with Relevance Factor. In *Proc. of the 11th European Symposium on Artificial Neural Networks (ESANN)* (pp. 367–372).

Andonie, R., & Sasu, L. (2006). Fuzzy ARTMAP with input relevances. *IEEE Transactions on Neural Networks*, *17*, 929–941.

Andonie, R., Sasu, L., & Beiu, V. (2003a). A Modified Fuzzy ARTMAP Architecture for Incremental Learning Function Approximation. In *Proc. IASTED Int. Conf. Neural Networks and Computational Intelligence (NCI)* (pp. 124–129).

Andonie, R., Sasu, L., & Beiu, V. (2003b). Fuzzy ARTMAP with relevance factor. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1975–1980). volume 3.

Araújo, D., Neto, A. D., & Martins, A. (2013a). Information-theoretic clustering: A representative and evolutionary approach. *Expert Systems with Applications*, *40*, 4190–4205.

Araújo, D., Neto, A. D., & Martins, A. (2013b). Representative cross information potential clustering. *Pattern Recognition Letters*, *34*, 2181 – 2191.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*, 243 – 256.

Asfour, Y. R., Carpenter, G. A., Grossberg, S., & Lesher, G. W. (1993). Fusion ARTMAP: an adaptive fuzzy network for multi-channel classification. In *Proc. Third International Conference on Industrial Fuzzy Control and Intelligent Systems* (pp. 155–160).

Auda, G., & Kamel, M. (1998). Modular Neural Network Classifiers: A Comparative Study. *J. Intell. Robot. Syst.*, *21*, 117–129.

Auda, G., & Kamel, M. (1999). Modular Neural Networks: A Survey. *International Journal of Neural Systems*, *09*, 129–151.

Auda, G., Kamel, M., & Raafat, H. (1996). Modular Neural Network Architectures for Classification. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1279–1284). volume 2.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Baek, J., Lee, H., Lee, B., Lee, H., & Kim, E. (2014). An efficient genetic selection of the presentation order in simplified fuzzy ARTMAP patterns. *Applied Soft Computing*, *22*, 101–107.

Bain, L. J., & Engelhardt, M. (1992). *Introduction to Probability and Mathematical Statistics*. (2nd ed.). Brooks/Cole, Cengage Learning.

Bandyopadhyay, S., & Maulik, U. (2001). Nonparametric genetic clustering: comparison of validity indices. *IEEE Trans. Syst., Man, Cybern. C*, *31*, 120–125.

Bartfai, G. (1994). Hierarchical clustering with ART neural networks. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 940–944). volume 2.

Bartfai, G. (1995). A comparison of two ART-based neural networks for hierarchical clustering. In *Proc. Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems* (pp. 83–86).

Bartfai, G. (1996). An ART-based modular architecture for learning hierarchical clusterings. *Neurocomputing*, *13*, 31 – 45.

Bartfai, G., & White, R. (1997a). A fuzzy ART-based modular neuro-fuzzy architecture for learning hierarchical clusterings. In *Proc. 6th International Fuzzy Systems Conference* (pp. 1713–1718). volume 3.

Bartfai, G., & White, R. (1997b). Adaptive Resonance Theory-based Modular Networks for Incremental Learning of Hierarchical Clusterings. *Connection Science*, *9*, 87–112.

Bartfai, G., & White, R. (1998). Learning and optimisation of hierarchical clusterings with ART-based modular networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2352–2356). volume 3.

Bartfai, G., & White, R. (2000). Incremental Learning and Optimization of Hierarchical Clusterings with ART-Based Modular Networks. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 87–131). Heidelberg: Physica-Verlag HD.

Béjar, J., Cortés, U., & Poch, M. (1993). *LINNEO+: A classification methodology for ill-structured domains*. Technical Report LSI-93-22-R Facultat d’InformÃătica de Barcelona.

Benites, F., & Sapozhnikova, E. (2017). Improving scalability of ART neural networks. *Neurocomputing*, *230*, 219–229.

Bezdek, J. C. (2017). *A Primer on Cluster Analysis: 4 Basic Methods that (usually) Work*. First Edition Design Publishing.

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2225–2230). volume 3.

Bezdek, J. C., Hathaway, R. J., & Huband, J. M. (2007). Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices. *IEEE Transactions on Fuzzy Systems*, *15*, 890–903.

Bezdek, J. C., Li, W. Q., Attikiouzel, Y., & Windham, M. (1997). A geometric approach to cluster validity for normal mixtures. *Soft Computing*, *1*, 166–179.

Bezdek, J. C., & Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *28*, 301–315.

Blume, M., & Esener, S. (1995). Optoelectronic Fuzzy ARTMAP processor. *Optical Computing*, *10*, 213–215.

Blume, M., & Van Blerkom, D. A. (2000). Fuzzy ARTMAP Modifications for Intersecting Class Distributions. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 27–54). Heidelberg: Physica-Verlag HD.

Bouchachia, A., & Mittermeir, R. (2003). A neural cascade architecture for document retrieval. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1915–1920). volume 3.

Brannon, N., Conrad, G., Draelos, T., Seiffertt, J., & Wunsch II, D. C. (2006). Information Fusion and Situation Awareness using ARTMAP and Partially Observable Markov Decision Processes. In *Proc. IEEE International Joint Conference on Neural Network (IJCNN)* (pp. 2023–2030).

Brannon, N., Seiffertt, J., Draelos, T., & Wunsch II, D. C. (2009). Coordinated machine learning and decision support for situation awareness. *Neural Networks*, *22*, 316 – 325. Goal-Directed Neural Systems.

Brito da Silva, L. E., & Costa, J. A. F. (2013a). A Gravitational Approach for Enhancing Cluster Visualization in Self-Organizing Maps. In *The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE'13)* (pp. 48–54).

Brito da Silva, L. E., & Costa, J. A. F. (2013b). Clustering, Noise Reduction and Visualization Using Features Extracted from the Self-Organizing Map. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, & X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning - IDEAL 2013* (pp. 242–251). Springer Berlin Heidelberg volume 8206 of *Lecture Notes in Computer Science*.

Brito da Silva, L. E., & Costa, J. A. F. (2013c). Clustering the Self-Organizing Map Based on the Neurons' Associated Pattern Sets. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence* (pp. 7–14).

Brito da Silva, L. E., & Costa, J. A. F. (2013d). Clustering the self-organizing map through the identification of core neuron regions. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019a). Dual vigilance fuzzy adaptive resonance theory. *Neural Networks*, *109*, 1–5.

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2019b). A survey of adaptive resonance theory neural network models for engineering applications. *Neural Networks*, .

Brito da Silva, L. E., Elnabarawy, I., & Wunsch II, D. C. (2020). Distributed dual vigilance fuzzy adaptive resonance theory learns online, retrieves arbitrarily-shaped clusters, and mitigates order dependence. *Neural Networks*, *121*, 208 – 228.

Brito da Silva, L. E., & Ferreira Costa, J. A. (2014). Clustering of the self-organizing map using particle swarm optimization and validity indices. In *The 2014 International Joint Conference on Neural Networks (IJCNN'14)* (pp. 3798–3806).

Brito da Silva, L. E., M. Melton, N., & Wunsch II, D. C. (2019c). Incremental Cluster Validity Indices for Hard Partitions: Extensions and Comparative Study. *CoRR*, *abs/1902.06711*. arXiv:1902.06711v1 [cs.LG].

Brito da Silva, L. E., & Wunsch II, D. C. (2015). Multi-prototype local density-based hierarchical clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–9).

Brito da Silva, L. E., & Wunsch II, D. C. (2017a). SOM IT-vis. URL: `https://git.mst.edu/acil-group/som-it-vis`.

Brito da Silva, L. E., & Wunsch II, D. C. (2017b). Validity Index-based Vigilance Test in Adaptive Resonance Theory Neural Networks. In *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Brito da Silva, L. E., & Wunsch II, D. C. (2017c). VI-based Fuzzy ART. URL: `https://git.mst.edu/acil-group/CVI-Fuzzy-ART`.

Brito da Silva, L. E., & Wunsch II, D. C. (2018a). A study on exploiting VAT to mitigate ordering effects in Fuzzy ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2351–2358).

Brito da Silva, L. E., & Wunsch II, D. C. (2018b). An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2595–2613.

Brito da Silva, L. E., & Wunsch II, D. C. (2018). VAT + Fuzzy ART Framework. URL: `https://git.mst.edu/acil-group/VAT-Fuzzy-ART`.

Cacoullos, T. (1966). Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, *18*, 179–189.

Cai, D., He, X., & Han, J. (2011). Speed up kernel discriminant analysis. *The VLDB Journal*, *20*, 21–33.

Cai, D., He, X., Han, J., & Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 1548–1560.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*, 1–27.

Calvo, B., & Santafé, G. (2016). scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems. *The R Journal*, *8*, 248–256.

Carpenter, G. A. (1994). A distributed outstar network for spatial pattern learning. *Neural Networks*, *7*, 159 – 168.

Carpenter, G. A. (1996a). Distributed activation, search, and learning by ART and ARTMAP neural networks. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 244–249).

Carpenter, G. A. (1996b). Distributed ART networks for learning, recognition, and prediction. In *Proc. World Congress on Neural Networks (WCNN)* (pp. 333 – 344).

Carpenter, G. A. (1997). Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks. *Neural Networks*, *10*, 1473 – 1494.

Carpenter, G. A. (2003). Default ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1396–1401). volume 2.

Carpenter, G. A. (2019). Looking to the future: Learning from experience, averting catastrophe. *Neural Networks*, *118*, 204 – 207.

Carpenter, G. A., & Gaddam, S. C. (2010). Biased ART: A neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. *Neural Networks*, *23*, 435 – 451.

Carpenter, G. A., & Gjaja, M. N. (1994). Fuzzy ART Choice Functions. *Proc. World Congress on Neural Networks (WCNN)*, (pp. 713–722).

Carpenter, G. A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*, 54 – 115.

Carpenter, G. A., & Grossberg, S. (1987b). ART 2: self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.*, *26*, 4919–4930.

Carpenter, G. A., & Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, *21*, 77–88.

Carpenter, G. A., & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, *3*, 129–152.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*, 565 – 588.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1995). A fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems. *IEEE Transactions on Neural Networks*, *6*, 1330–1336.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991b). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, *4*, 493 – 504.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991c). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*, 759 – 771.

Carpenter, G. A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, *11*, 323 – 336.

Carpenter, G. A., Milenova, B. L., & Noeske, B. W. (1998). Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks*, *11*, 793 – 813.

Carpenter, G. A., & Ross, W. D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, *6*, 805–818.

Carpenter, G. A., & Tan, A.-H. (1995). Rule Extraction: From Neural Architecture to Symbolic Representation. *Connection Science*, *7*, 3–27.

Caudell, T. P. (1992). Hybrid optoelectronic adaptive resonance theory neural processor, ART1. *Appl. Opt.*, *31*, 6220–6229.

Caudell, T. P., Smith, S. D., Escobedo, R., & Anderson, M. (1994). NIRS: Large scale ART-1 neural architectures for engineering design retrieval. *Neural Networks*, *7*, 1339 – 1350.

Caudell, T. P., Smith, S. D., Johnson, G. C., & Wunsch II, D. C. (1991). Application of neural networks to group technology. In *Proceedings of SPIE - The International Society for Optical Engineering* (pp. 612–621). volume 1469.

Chalasani, R., & Principe, J. (2010). Self organizing maps with the correntropy induced metric. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–6).

Chalasani, R., & Principe, J. C. (2015). Self-organizing maps with information theoretic learning. *Neurocomputing*, *147*, 3 – 14. Advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012).

Chang, H., & Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*, 191–203.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. (2010). Complete Gradient Clustering Algorithm for Features Analysis of X-Ray Images. In E. Pietka, & J. Kawa (Eds.), *Information Technologies in Biomedicine* (pp. 15–24). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, P. P., & Lin, W.-C. (2001). Multi-resolution distributed ART neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. A19–A24). volume Addendum.

Chen, P. P., Lin, W.-C., & Hung, H.-L. (1999). Multi-resolution fuzzy ART neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1973–1978). volume 3.

Chin, W. H., Loo, C. K., Seera, M., Kubota, N., & Toda, Y. (2016). Multi-channel Bayesian Adaptive Resonance Associate Memory for on-line topological map building. *Applied Soft Computing*, *38*, 269 – 280.

Chou, C.-H., Hsieh, S.-C., & Qiu, C.-J. (2017). Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction. *Applied Soft Computing*, *56*, 298 – 316.

Clapper, J. P., & Bower, G. H. (1994). Category invention in unsupervised learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 443 – 460.

Comon, P. (1994). Independent component analysis, A new concept? *Signal Processing*, *36*, 287 – 314.

Cornuéjols, A. (1993). Getting order independence in incremental learning. In P. B. Brazdil (Ed.), *Machine Learning: ECML-93* (pp. 196–212). Berlin, Heidelberg: Springer Berlin Heidelberg.

Costa, J. A. F., & Netto, M. L. A. (1999). Estimating the number of clusters in multivariate data by self-organizing maps. *International Journal of Neural Systems*, *9*, 195–202.

Costa, J. A. F., & Netto, M. L. A. (2001). Clustering of complex shaped data sets via Kohonen maps and mathematical morphology. In B. Dasarathy (Ed.), *Proceedings of the SPIE, Data Mining and Knowledge Discovery* (pp. 16–27). volume 4384.

Costa, J. A. F., & Yin, H. (2010). Gradient-based SOM clustering and visualisation methods. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Cottrell, M., & Rousset, P. (1997). The kohonen algorithm: A powerful tool for analysing and representing multidimensional quantitative and qualitative data. In J. Mira, R. Moreno-Díaz, & J. Cabestany (Eds.), *Biological and Artificial Computation: From Neuroscience to Technology* (pp. 861–871). Berlin, Heidelberg: Springer Berlin Heidelberg.

da Silva, A. R., & Goes, L. F. W. (2018). HearthBot: An Autonomous Agent Based on Fuzzy ART Adaptive Neural Networks for the Digital Collectible Card Game HearthStone. *IEEE Transactions on Games*, *10*, 170–181.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1998). Ordered fuzzy ARTMAP: a fuzzy ARTMAP algorithm with a fixed order of pattern presentation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1717–1722). volume 3.

Dagher, I., Georgiopoulos, M., Heileman, G. L., & Bebis, G. (1999). An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance. *IEEE Transactions on Neural Networks*, *10*, 768–778.

Dagli, C. H., & Huggahalli, R. (1993). A neural network approach to group technology. In J. Wang, & Y. Takefuji (Eds.), *Neural Networks in Design and Manufacturing* (pp. 1–55). Singapore: World Scientific.

Dakshayani, G., Srinivasulu, A., & Haneesh, K. (2016). Study on Clustering Large Data Using Fuzzy Adaptive Resonance Theory. *International Research Journal Of Engineering And Technology (IRJET)*, *3*, 1673–1677.

Davenport, M. P., & Titus, A. H. (2004). Multilevel category structure in the ART-2 network. *IEEE Transactions on Neural Networks*, *15*, 145–158.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*, 224–227.

DeClaris, N., & Su, M.-C. (1991). A novel class of neural networks with quadratic junctions. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1557–1562). volume 3.

DeClaris, N., & Su, M.-C. (1992). Introduction to the theory and applications of neural networks with quadratic junctions. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1320–1325). volume 2.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., & Zupan, B. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, *14*, 2349–2353.

Dimitriadou, E., Dolničar, S., & Weingessel, A. (2002). An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, *67*, 137–159.

Du, K.-L. (2010). Clustering: A neural network approach. *Neural Networks*, *23*, 89 – 107.

Dua, D., & Graff, C. (2017). UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Dubes, R., & Jain, A. K. (1979). Validity studies in clustering methodologies. *Pattern Recognition*, *11*, 235 – 254.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. (2nd ed.). John Wiley & Sons.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, *3*, 32–57.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing*. (2nd ed.). Springer Publishing Company, Incorporated.

Elnabarawy, I., Tauritz, D. R., & Wunsch II, D. C. (2017). Evolutionary Computation for the Automated Design of Category Functions for Fuzzy ART: An Initial Exploration. In *Proc. Genetic and Evolutionary Computation Conference Companion* GECCO'17 (pp. 1133–1140). New York, NY, USA: ACM.

Elnabarawy, I., Wunsch II, D. C., & Abdelbar, A. M. (2016). Biclustering ARTMAP Collaborative Filtering Recommender System. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2986–2991).

Esposito, F., Ferilli, S., Fanizzi, N., Basile, T. M. A., & Mauro, N. D. (2003). Incremental multistrategy learning for document processing. *Applied Artificial Intelligence*, *17*, 859–883.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 226–231). AAAI Press.

Fisher, D. (1993). Ordering Effects in Incremental Learning. In *Proc. of the 1993 AAAI Spring Symposium on Training Issues in Incremental Learning* SS-93-06 (pp. 35–42). Menlo Park, California: The AAAI Press.

Fisher, D. (1996). Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*, *4*, 147–179.

Fisher, D., Xu, L., & Zard, N. (1992). Ordering Effects in Clustering. In D. Sleeman, & P. Edwards (Eds.), *Machine Learning Proceedings 1992* (pp. 163 – 168). San Francisco (CA): Morgan Kaufmann.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*, 139–172.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, *48*, 4743–4759.

Fränti, P., & Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, *39*, 761 – 775.

Fränti, P., Virmajoki, O., & Hautamaki, V. (2006). Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 1875–1881.

Fränti, Pasi et al. (2015). Clustering datasets. URL: `http://cs.uef.fi/sipu/datasets/` accessed on May 4, 2017.

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*, 972–976.

Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, *32*, 675–701.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, *11*, 86–92.

Fritzke, B. (1994). Growing cell structures–A self-organizing network for unsupervised and supervised learning. *Neural Networks*, *7*, 1441 – 1460.

Fritzke, B. (1995). A Growing Neural Gas Network Learns Topologies. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7* (pp. 625–632). MIT Press.

Fu, L., & Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, *8*, 1–15.

Fudenberg, D., & Tirole, J. (1991). *Game Theory*. Cambridge, MA: MIT Press.

Fukumizu, K., Song, L., & Gretton, A. (2013). Kernel Bayes' Rule: Bayesian Inference with Positive Definite Kernels. *Journal of Machine Learning Research*, *14*, 3753–3783. URL: `http://jmlr.org/papers/v14/fukumizu13a.html`.

Fung, W. K., & Liu, Y. H. (1999). A game-theoretic formulation on adaptive categorization in ART networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1081–1086). volume 2.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *19*, 90 – 106.

Gabrys, B., & Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. *IEEE Transactions on Neural Networks*, *11*, 769–783.

Garcia, V., Debreuve, E., & Barlaud, M. (2008). Fast k Nearest Neighbor Search using GPU. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08)* (pp. 1–6).

Georgiopoulos, M., Fernlund, H., Bebis, G., & Heileman, G. L. (1996). Order of Search in Fuzzy ART and Fuzzy ARTMAP: Effect of the Choice Parameter. *Neural Networks*, *9*, 1541 – 1559.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1990). Convergence Properties of Learning in ART1. *Neural Computation*, *2*, 502–509.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1991). Properties of learning related to pattern diversity in ART1. *Neural Networks*, *4*, 751 – 757.

Georgiopoulos, M., Heileman, G. L., & Huang, J. (1992). The N-N-N conjecture in ART1. *Neural Networks*, *5*, 745 – 753.

Georgiopoulos, M., Huang, J., & Heileman, G. L. (1994). Properties of learning in ARTMAP. *Neural Networks*, *7*, 495 – 506.

Georgiopoulos, M., Koufakou, A., Anagnostopoulos, G. C., & Kasparis, T. (2001). Overtraining in fuzzy ARTMAP: Myth or reality? In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1186–1190). volume 2.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*, 1–30.

Giraud-Carrier, C. (2000). A Note on the Utility of Incremental Learning. *AI Communications*, *13*, 215–223.

Gokcay, E., & Principe, J. (2000). A new clustering evaluation function using renyi's information potential. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on* (pp. 3490–3493). volume 6.

Gokcay, E., & Principe, J. C. (2002). Information theoretic clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*, 158–171.

Gomez-Sanchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Lopez-Coronado, J. (2001). Safe-μARTMAP: a new solution for reducing category proliferation in fuzzy ARTMAP. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 1197–1202). volume 2.

Gomez-Sanchez, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Lopez-Coronado, J. (2002). μARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP. *IEEE Transactions on Neural Networks*, *13*, 58–69.

Gonçalves, M. L., de Andrade Netto, M. L., Costa, J. A. F., & Zullo, J. (2006). Data Clustering using Self-Organizing Maps segmented by Mathematic Morphology and Simplified Cluster Validity Indexes: an application in remotely sensed images. In *The 2006 IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 4421–4428).

Gonzalez, R. C., & Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gordon, A. D. (1998). Cluster Validation. In C. Hayashi, K. Yajima, H.-H. Bock, N. Ohsumi, Y. Tanaka, & Y. Baba (Eds.), *Data Science, Classification, and Related Methods* (pp. 22–39). Tokyo: Springer Japan.

Granger, E., Savaria, Y., Lavoie, P., & Cantin, M.-A. (1998). A comparison of self-organizing neural networks for fast clustering of radar pulses. *Signal Processing*, *64*, 249 – 269.

Grossberg, S. (1968). A prediction theory for some nonlinear functional-differential equations i. learning of lists. *Journal of Mathematical Analysis and Applications*, *21*, 643 – 694.

Grossberg, S. (1969). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns, i. *Journal of Mathematics and Mechanics*, *19*, 53–91.

Grossberg, S. (1972). Neural expectation: cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik*, *10*, 49–57.

Grossberg, S. (1976a). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, *23*, 187–202.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, *87*, 1–51.

Grossberg, S. (2013). Adaptive Resonance Theory: how a brain learns to consciously attend, learn, and recognize a changing world. *Neural networks*, *37*, 1–47.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An Efficient Clustering Algorithm for Large Databases. In *Proc. ACM SIGMOD Int. Conf. Manag. Data* (pp. 73–84). New York, NY, USA: ACM.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002a). Cluster Validity Methods: Part I. *SIGMOD Rec.*, *31*, 40–45.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002b). Clustering Validity Checking Methods: Part II. *SIGMOD Rec.*, *31*, 19–27.

Halkidi, M., & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, *29*, 773 – 786.

Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, *10*.

Haralick, R. M., & Shapiro, L. G. (1992). *Computer and Robot Vision*. Addison-Wesley.

Hastie, T., & Stuetzle, W. (1989). Principal Curves. *Journal of the American Statistical Association*, *84*, 502–516.

Hathaway, R. J., Bezdek, J. C., & Huband, J. M. (2006). Scalable visual assessment of cluster tendency for large data sets. *Pattern Recognition*, *39*, 1315 – 1324.

Havens, T. C., & Bezdek, J. C. (2012). An Efficient Formulation of the Improved Visual Assessment of Cluster Tendency (iVAT) Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *24*, 813–822.

Havens, T. C., Bezdek, J. C., Keller, J. M., & Popescu, M. (2009a). Clustering in ordered dissimilarity data. *International Journal of Intelligent Systems*, *24*, 504–528.

Havens, T. C., Bezdek, J. C., Keller, J. M., Popescu, M., & Huband, J. M. (2009b). Is vat really single linkage in disguise? *Annals of Mathematics and Artificial Intelligence*, *55*, 237–251.

Havens, T. C., Bezdek, J. C., & Palaniswami, M. (2013). Scalable single linkage hierarchical clustering for big data. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (pp. 396–401).

Haykin, S. S. (2009). *Neural networks and Learning Machines*. (3rd ed.). Upper Saddle River, NJ: Pearson Education.

Healy, M. J., & Caudell, T. P. (1998). Guaranteed two-pass convergence for supervised and inferential learning. *IEEE Transactions on Neural Networks*, *9*, 195–204.

Healy, M. J., & Caudell, T. P. (2006). Ontologies and Worlds in Category Theory: Implications for Neural Systems. *Axiomathes*, *16*, 165–214.

Healy, M. J., & Caudell, T. P. (2019). Episodic Memory: A Hierarchy of Spatiotemporal Concepts. *Neural Networks*, *this issue*.

Healy, M. J., Caudell, T. P., & Smith, S. D. G. (1993). A neural architecture for pattern sequence verification through inferencing. *IEEE Transactions on Neural Networks*, *4*, 9–20.

Healy, M. J., Olinger, R. D., Young, R. J., Taylor, S. E., Caudell, T., & Larson, K. W. (2009). Applying category theory to improve the performance of a neural architecture. *Neurocomputing*, *72*, 3158 – 3173.

Heileman, G. L., Georgiopoulos, M., & Juxin Hwang (1994). A survey of learning results for ART1 networks. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 1222–1225). volume 2.

Henniges, P., Granger, E., & Sabourin, R. (2005). Factors of overtraining with fuzzy ARTMAP neural networks. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1075–1080). volume 2.

Ho, C. S., Liou, J. J., Georgiopoulos, M., Heileman, G. L., & Christodoulou, C. (1994). Analogue circuit design and implementation of an adaptive resonance theory (ART) neural network architecture. *International Journal of Electronics*, *76*, 271–291.

Hruschka, E. R., Campello, R. J., & de Castro, L. N. (2006). Evolving clusters in gene-expression data. *Information Sciences*, *176*, 1898 – 1927.

Hruschka, E. R., de Castro, L. N., & Campello, R. J. G. B. (2004). Evolutionary algorithms for clustering gene-expression data. In *Proc. IEEE Int. Conf. Data Mining (ICDM)* (pp. 403–406).

Huang, J., Georgiopoulos, M., & Heileman, G. L. (1994). Properties of learning in fuzzy ART. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 756–761). volume 2.

Huang, J., Georgiopoulos, M., & Heileman, G. L. (1995). Fuzzy ART properties. *Neural Networks*, *8*, 203 – 213.

Huang, Y.-T., Cheng, F.-T., Shih, Y.-H., & Chen, Y.-L. (2014). Advanced ART2 scheme for enhancing metrology-data-quality evaluation. *Journal of the Chinese Institute of Engineers*, *37*, 1064–1079.

Huband, J. M., Bezdek, J. C., & Hathaway, R. J. (2004). Revised Visual Assessment of (Cluster) Tendency (reVAT). In *IEEE Annual Meeting of the Fuzzy Information Processing Society (NAFIPS'04)* (pp. 101–104). volume 1.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Hung, H.-L., Liao, H.-Y. M., Lin, S.-J., Lin, W.-C., & Fan, K.-C. (1996a). Cascade fuzzy ART: a new extensible database for model-based object recognition. *Proc. SPIE*, *2727*, 187–198.

Hung, H.-L., Liao, H.-Y. M., Sze, C.-J., Lin, S.-J., Lin, W.-C., & Fan, K.-C. (1996b). CFART: A Multi-Resolutional Adaptive Resonance System. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 1312–1317). volume 2.

Ibrahim, O. A., Keller, J. M., & Bezdek, J. C. (2018a). Analysis of streaming clustering using an incremental validity index. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8).

Ibrahim, O. A., Keller, J. M., & Bezdek, J. C. (2019). Evaluating Evolving Structure in Streaming Data With Modified Dunn's Indices. *IEEE Transactions on Emerging Topics in Computational Intelligence*, (pp. 1–12).

Ibrahim, O. A., Shao, J., Keller, J. M., & Popescu, M. (2016). A temporal analysis system for early detection of health changes. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 186–193).

Ibrahim, O. A., Wang, Y., & Keller, J. M. (2018b). Analysis of incremental cluster validity for big data applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *26*, 47–62.

Ilc, N. (2013). Datasets package. URL: `https://www.researchgate.net/publication/239525861_Datasets_package` accessed on Sep 11, 2017.

Ilc, N., & Dobnikar, A. (2011). Gravitational Clustering of the Self-Organizing Map. In A. Dobnikar, U. Lotrič, & B. Šter (Eds.), *Adaptive and Natural Computing Algorithms* (pp. 11–20). Berlin, Heidelberg: Springer volume 6594.

Ilc, N., & Dobnikar, A. (2012). Generation of a clustering ensemble based on a gravitational self-organising map. *Neurocomputing*, *96*, 47–56.

Illetskova, M., Elnabarawy, I., Brito da Silva, L. E., Tauritz, D. R., & Wunsch II, D. C. (2019). Nested Monte Carlo Search Expression Discovery for the Automated Design of Fuzzy ART Category Choice Functions. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* GECCO'19 (pp. 171–172). New York, NY, USA: ACM.

Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics - Theory and Methods*, *9*, 571–595.

Isawa, H., Matsushita, H., & Nishio, Y. (2008a). Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3595–3600).

Isawa, H., Matsushita, H., & Nishio, Y. (2008b). Improved Fuzzy Adaptive Resonance Theory Combining Overlapped Category in Consideration of Connections. In *IEEE Workshop on Nonlinear Circuit Networks (NCN)* (pp. 8–11).

Isawa, H., Matsushita, H., & Nishio, Y. (2009). Fuzzy ART Combining Overlapped Categories Using Variable Vigilance Parameters. In *Proc. International Workshop on Nonlinear Circuits and Signal Processing (NCSP)* (pp. 661–664).

Isawa, H., Tomita, M., Matsushita, H., & Nishio, Y. (2007). Fuzzy Adaptive Resonance Theory with Group Learning and its Applications. In *Proc. International Symposium on Nonlinear Theory and its Applications (NOLTA)* (pp. 292–295).

Ishihara, S., Hatamoto, K., Nagamachi, M., & Matsubara, Y. (1993). ART1.5SSS for Kansei engineering expert system. In *Proc. International Conference on Neural Networks (IJCNN)* (pp. 2512–2515). volume 3.

Ishihara, S., Ishihara, K., Nagamachi, M., & Matsubara, Y. (1995). arboART: ART based hierarchical clustering and its application to questionnaire data analysis. In *Proc. IEEE International Conference on Neural Networks (ICNN)* (pp. 532–537). volume 1.

Izquierdo, J. M. C., Almonacid, M., Pinzolas, M., & Ibarrola, J. (2009). dFasArt: Dynamic neural processing in FasArt model. *Neural Networks*, *22*, 479 – 487.

Izquierdo, J. M. C., Dimitriadis, Y. A., Araúzo, M., & Coronado, J. L. (1996). FasArt: A New Neuro-Fuzzy Architecture for Incremental Learning in System Identification. In *IFAC Proceedings Volumes* (pp. 2532 – 2537). volume 29.

Izquierdo, J. M. C., Dimitriadis, Y. A., & Coronado, J. L. (1997). FasBack: matching-error based learning for automatic generation of fuzzy logic systems. In *Proc. International Fuzzy Systems Conference* (pp. 1561–1566). volume 3.

Izquierdo, J. M. C., Dimitriadis, Y. A., Sánchez, E. G., & Coronado, J. L. (2001). Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, *14*, 407 – 425.

Jain, A. K., & Law, M. H. C. (2005). Data Clustering: A User's Dilemma. In S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds.), *Pattern Recognition and Machine Intelligence* (pp. 1–10). Berlin, Heidelberg: Springer Berlin Heidelberg volume 3776 of *Lecture Notes in Computer Science*.

Jain, L. C., Lazzerini, B., & Halici, U. (Eds.) (2000). *Innovations in ART Neural Networks* volume 43. (1st ed.). Dordrecht: Physica-Verlag Heidelberg.

Jain, L. C., Seera, M., Lim, C. P., & Balasubramaniam, P. (2014). A review of online learning in supervised neural networks. *Neural Computing and Applications*, *25*, 491–509.

Kamimura, R. (2013). Similarity interaction in information-theoretic self-organizing maps. *International Journal of General Systems*, *42*, 239–267.

Kärkkäinen, I., & Fränti, P. (2002). *Dynamic local search algorithm for the clustering problem*. Technical Report A-2002-6 Department of Computer Science, University of Joensuu Joensuu, Finland.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, *32*, 68–75.

Kasuba, T. (1993). Simplified Fuzzy ARTMAP. *AI Expert*, *8*, 18–25.

Kaylani, A., Georgiopoulos, M., Mollaghasemi, M., & Anagnostopoulos, G. C. (2009). AG-ART: An adaptive approach to evolving ART architectures. *Neurocomputing*, *72*, 2079 – 2092.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1942–1948). volume 4.

Kim, B., Ban, S.-W., & Lee, M. (2011). Growing fuzzy topology adaptive resonance theory models with a push-pull learning algorithm. *Neurocomputing*, *74*, 646 – 655.

Kim, S. (2016). *Novel approaches to clustering , biclustering algorithms based on adaptive resonance theory and intelligent control*. Ph.D. thesis Missouri University of Science and Technology.

Kim, S., & Wunsch II, D. C. (2011). A GPU based Parallel Hierarchical Fuzzy ART clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2778–2782).

Knuth, D. E. (1964). Backus Normal Form vs. Backus Naur Form. *Communications of the ACM*, *7*, 735–736.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kohonen, T. (1990). Improved versions of learning vector quantization. In *1990 IJCNN International Joint Conference on Neural Networks* (pp. 545–550). volume 1.

Kohonen, T. (2001). *Self-Organizing Maps* volume 30 of *Springer Series in Information Sciences*. (3rd ed.). Springer-Verlag Berlin Heidelberg New York.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks*, *37*, 52 – 65.

Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, *47*, 1902–1914.

Kondadadi, R., & Kozma, R. (2002). A modified fuzzy ART for soft document clustering. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2545–2549). volume 3.

Kosko, B. (1986). Fuzzy entropy and conditioning. *Information Sciences*, *40*, 165 – 174.

Koufakou, A., Georgiopoulos, M., Anagnostopoulos, G., & Kasparis, T. (2001). Cross-validation in Fuzzy ARTMAP for large databases. *Neural Networks*, *14*, 1279 – 1291.

Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2004). *Applied Linear Statistical Models*. (5th ed.).

Lago-Fernández, L. F., & Corbacho, F. (2009). Using the negentropy increment to determine the number of clusters. In J. Cabestany, F. Sandoval, A. Prieto, & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009. Proceedings, Part I* (pp. 448–455). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lago-Fernández, L. F., & Corbacho, F. (2010). Normality-based validation for crisp clustering. *Pattern Recognition*, *43*, 782 – 795.

Lam, D., Wei, M., & Wunsch II, D. C. (2015). Clustering Data of Mixed Categorical and Numerical Type With Unsupervised Feature Learning. *IEEE Access*, *3*, 1605–1613.

Lamirel, J., & Cuxac, P. (2015). New quality indexes for optimal clustering model identification with high dimensional data. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (pp. 855–862).

Lamirel, J., Dugué, N., & Cuxac, P. (2016). New efficient clustering quality indexes. In *Proc. Int. Joint Conf. Neural Netw. (IJCNN)* (pp. 3649–3657).

Langley, P. (1995). Order Effects in Incremental Learning. In P. Reimann, & H. Spada (Eds.), *Learning in Humans and Machines: Towards an Interdisciplinary Learning Science* (pp. 154–165). Emerald Group Publishing Limited.

Lavoie, P. (1999). Choosing a choice function: granting new capabilities to ART. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1988–1993). volume 3.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1997). Multiple categorization using fuzzy ART. In *Proceedings of International Conference on Neural Networks (ICNN'97)* (pp. 1983–1988). volume 3.

Lavoie, P., Crespo, J.-F., & Savaria, Y. (1999). Generalization, discrimination, and multiple categorization using adaptive resonance theory. *IEEE Transactions on Neural Networks*, *10*, 757–767.

Le, Q., Anagnostopoulos, G. C., Georgiopoulos, M., & Ports, K. (2005). An experimental comparison of semi-supervised ARTMAP architectures, GCS and GNG classifiers. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3121–3126). volume 5.

LeBlanc, M., & Tibshirani, R. (1994). Adaptive Principal Surfaces. *Journal of the American Statistical Association*, *89*, 53–64.

Leconte, F., Ferland, F., & Michaud, F. (2014). Fusion Adaptive Resonance Theory Networks Used as Episodic Memory for an Autonomous Robot. In B. Goertzel, L. Orseau, & J. Snaider (Eds.), *Artificial General Intelligence* (pp. 63–72). Cham: Springer International Publishing.

Leconte, F., Ferland, F., & Michaud, F. (2016). Design and integration of a spatio-temporal memory with emotional influences to categorize and recall the experiences of an autonomous mobile robot. *Autonomous Robots*, *40*, 831–848.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Lee, C. J., Yoon, C. G., & Lee, C. W. (1995). A new learning method to improve the category proliferation problem in fuzzy ART. In *Proc. International Conference on Neural Networks (ICNN)* (pp. 1393–1396). volume 3.

Lee, J. S., Yoon, C. G., & Lee, C. W. (1998). Learning method for fuzzy ARTMAP in a noisy environment. *Electronics Letters*, *34*, 95–97.

Lee, S. W., Palmer-Brown, D., & Roadknight, C. M. (2004). Performance-guided neural network for rapidly self-organising active network management. *Neurocomputing*, *61*, 5 – 20. Hybrid Neurocomputing: Selected Papers from the 2nd International Conference on Hybrid Intelligent Systems.

Lee, S. W., Palmer-Brown, D., Tepper, J. A., & Roadknight, C. M. (2003). Snap-drift: real-time, performance-guided learning. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* (pp. 1412–1416 vol.2). volume 2.

Lerner, B., & Guterman, H. (2008). Advanced Developments and Applications of the Fuzzy ARTMAP Neural Network in Pattern Classification. In L. C. Jain, M. Sato-Ilic, M. Virvou, G. A. Tsihrintzis, V. E. Balas, & C. Abeynayake (Eds.), *Computational Intelligence Paradigms: Innovative Applications* (pp. 77–107). Berlin, Heidelberg: Springer Berlin Heidelberg.

Levine, D. S. (2019). *Introduction to Neural and Cognitive Modeling*. (3rd ed.). New York: Routledge.

Levine, D. S., & Penz, P. A. (1990). ART 1.5–A simplified adaptive resonance network for classifying low-dimensional analog data. In *Proc. of International Conference on Neural Networks (IJCNN)* (pp. 639–642). volume 2.

Lim, C. P., & Harrison, R. F. (1997a). An Incremental Adaptive Network for On-line Supervised Learning and Probability Estimation. *Neural Networks*, *10*, 925 – 939.

Lim, C. P., & Harrison, R. F. (1997b). Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration. *Neural Networks*, *10*, 755 – 774.

Lim, C. P., & Harrison, R. F. (2000a). ART-Based Autonomous Learning Systems: Part I — Architectures and Algorithms. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 133–166). Heidelberg: Physica-Verlag HD.

Lim, C. P., & Harrison, R. F. (2000b). ART-Based Autonomous Learning Systems: Part II — Applications. In L. C. Jain, B. Lazzerini, & U. Halici (Eds.), *Innovations in ART Neural Networks* (pp. 167–188). Heidelberg: Physica-Verlag HD.

Lin, T.-H., & Soo, V.-W. (1997). Pruning fuzzy ARTMAP using the minimum description length principle in learning from clinical databases. In *Proc. Ninth IEEE International Conference on Tools with Artificial Intelligence* (pp. 396–403).

Liu, B., Wan, C., & Wang, L. (2006). An efficient semi-unsupervised gene selection method via spectral biclustering. *IEEE Transactions on NanoBioscience*, *5*, 110–114.

Liu, W., Pokharel, P. P., & Príncipe, J. C. (2007). Correntropy: Properties and Applications in Non-Gaussian Signal Processing. *IEEE Transactions on Signal Processing*, *55*, 5286–5298.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition*, *41*, 995 – 1011.

Luna-Romera, J. M., del Mar Martínez-Ballesteros, M., García-Gutiérrez, J., & Riquelme-Santos, J. C. (2016). An Approach to Silhouette and Dunn Clustering Indices Applied to Big Data in Spark. In O. Luaces, J. A. Gámez, E. Barrenechea, A. Troncoso, M. Galar, H. Quintián, & E. Corchado (Eds.), *Advances in Artificial Intelligence* (pp. 160–169). Cham: Springer International Publishing.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

MacGregor, J. N. (1988). The Effects of Order on Learning Classifications by Example: Heuristics for Finding the Optimal Order. *Artificial Intelligence*, *34*, 361 – 370.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). University of California Press volume 1.

Majeed, S., Gupta, A., Raj, D., & Rhee, F. C.-H. (2018). Uncertain fuzzy self-organization based clustering: interval type-2 fuzzy approach to adaptive resonance theory. *Information Sciences*, *424*, 69 – 90.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Manukyan, N., Eppstein, M. J., & Rizzo, D. M. (2012). Data-driven cluster reinforcement and visualization in sparsely-matched self-organizing maps. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 846–852.

Marriott, S., & Harrison, R. F. (1995). A modified fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural Networks*, *8*, 619 – 641.

Martinetz, T., & Schulten, K. (1994). Topology representing networks. *Neural Networks*, *7*, 507 – 522.

Martinetz, T. M., & Shulten, K. J. (1991). A “Neural-Gas” Network Learns Topologies. In T. Kohonen, K. Mäkisara, O. Simula, & J. Kangas (Eds.), *Artificial Neural Networks* (pp. 397–402).

Martínez-Zarzuela, M., Díaz Pernas, F. J., Díez Higuera, J. F., & Rodríguez, M. A. (2007). Fuzzy ART Neural Network Parallel Computing on the GPU. In F. Sandoval, A. Prieto, J. Cabestany, & M. Graña (Eds.), *Computational and Ambient Intelligence* (pp. 463–470). Berlin, Heidelberg: Springer Berlin Heidelberg.

Martínez-Zarzuela, M., Díaz-Pernas, F. J., de Pablos, A. T., Perozo-Rondón, F., Antón-Rodríguez, M., & González-Ortega, D. (2011). Fuzzy ARTMAP Based Neural Networks on the GPU for High-Performance Pattern Recognition. In J. M. Ferrández, J. R. Álvarez Sánchez, F. de la Paz, & F. J. Toledo (Eds.), *New Challenges on Bioinspired Applications* (pp. 343–352). Berlin, Heidelberg: Springer Berlin Heidelberg.

Martínez-Zarzuela, M., Pernas, F. J. D., de Pablos, A. T., Rodríguez, M. A., Higuera, J. F. D., Giralda, D. B., & Ortega, D. G. (2009). Adaptative Resonance Theory Fuzzy Networks Parallel Computation Using CUDA. In J. Cabestany, F. Sandoval, A. Prieto, & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence* (pp. 149–156). Berlin, Heidelberg: Springer Berlin Heidelberg.

Martins, A., Doria Neto, A. D., & de Melo, J. D. (2003). Neural networks applied to classification of data based on Mahalanobis metrics. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3071–3076). volume 4.

Martins, A., Doria Neto, A. D., de Melo, J. D., & Costa, J. A. F. (2004a). Clustering using neural networks and Kullback-Leibler divergency. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2813–2817). volume 4.

Martins, A., Neto, A., & de Melo, J. (2004b). Comparison Between Mahalanobis Distance and Kullback-Leibler Divergence in Clustering Analisys. In *WSEAS Transactions on Systems* (pp. 501–505). volume 3.

Massey, L. (2009). Discovery of hierarchical thematic structure in text collections with adaptive resonance theory. *Neural Computing and Applications*, *18*, 261–273.

Masuyama, N., Loo, C. K., & Dawood, F. (2018a). Kernel Bayesian ART and ARTMAP. *Neural Networks*, *98*, 76 – 86.

Masuyama, N., Loo, C. K., Ishibuchi, H., Nojima, Y., & Lin, Y. (2018b). Topological Kernel Bayesian ARTMAP. In *2018 World Automation Congress (WAC)* (pp. 1–5).

Masuyama, N., Loo, C. K., & Wermter, S. (2019). A Kernel Bayesian Adaptive Resonance Theory with A Topological Structure. *International journal of neural systems*, *29*, 1850052–1 – 1850052–19.

Matias, A. L. S., & Neto, A. R. R. (2018). OnARTMAP: A Fuzzy ARTMAP-based Architecture. *Neural Networks*, *98*, 236 – 250.

Matias, A. L. S., Neto, A. R. R., & Rocha, A. (2017). Opposite-to-Noise ARTMAP Neural Network. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in Computational Intelligence* (pp. 507–519). Cham: Springer International Publishing.

Mauro, N. D., Esposito, F., Ferilli, S., & Basile, T. M. A. (2004). A Backtracking Strategy for Order-independent Incremental Learning. In *Proceedings of the 16th European Conference on Artificial Intelligence* ECAI'04 (pp. 460–464). Amsterdam, The Netherlands, The Netherlands: IOS Press.

Mauro, N. D., Esposito, F., Ferilli, S., & Basile, T. M. A. (2005). Avoiding Order Effects in Incremental Learning. In S. Bandini, & S. Manzoni (Eds.), *AI*IA 2005: Advances in Artificial Intelligence* (pp. 110–121). Berlin, Heidelberg: Springer Berlin Heidelberg.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In G. H. Bower (Ed.), *Psychology of Learning and Motivation* (pp. 109 – 165). Academic Press volume 24.

Meng, L., & Tan, A. H. (2012). *Heterogeneous Learning of Visual and Textual Features for Social Web Image Co-Clustering*. Technical Report School of Computer Engineering, Nanyang Technological University.

Meng, L., Tan, A.-H., Leung, C., Nie, L., Chua, T.-S., & Miao, C. (2015). Online Multimodal Co-indexing and Retrieval of Weakly Labeled Web Image Collections. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* ICMR '15 (pp. 219–226). New York, NY, USA: ACM.

Meng, L., Tan, A.-H., & Wunsch II, D. (2013). Vigilance adaptation in adaptive resonance theory. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7).

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2016). Adaptive scaling of cluster boundaries for large-scale social media data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 2656–2669.

Meng, L., Tan, A.-H., & Wunsch II, D. C. (2019). *Adaptive Resonance Theory in Social Media Data Clustering: Roles, Methodologies, and Applications*. Cham: Springer International Publishing.

Meng, L., Tan, A. H., & Xu, D. (2014). Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 2293–2306.

Merkl, D., & Rauber, A. (1997). Alternative Ways for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'97* (pp. 106–111).

Mermillod, M., Bugaiska, A., & Bonin, P. (2013). The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, *4*, 1–3.

Meuth, R. J. (2009). *Meta-Learning Computational Intelligence Architectures*. Ph.D. thesis Missouri University of Science and Technology.

Meyer, F. (1994). Topographic Distance and Watershed Lines. *Signal Process.*, *38*, 113–125.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*, 159–179.

Moore, B. (1989). ART 1 and pattern clustering. In *Proceedings of the 1988 connectionist models summer school* (pp. 174–185). Morgan Kaufmann Publishers San Mateo, CA.

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., & Bailey, J. (2018). Online Cluster Validity Indices for Streaming Data. *ArXiv e-prints*, . ArXiv:1801.02937v1 [stat.ML].

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., & Bailey, J. (2019). Online cluster validity indices for performance monitoring of streaming data clustering. *International Journal of Intelligent Systems*, *34*, 541–563.

Mulder, S. A., & Wunsch II, D. C. (2003). Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, *16*, 827–832.

Nasir, J., Kim, D.-H., & Kim, J.-H. (2019). ART neural network-based integration of episodic memory and semantic memory for task planning for robots. *Autonomous Robots*, .

Nasir, J., Yoo, Y., Kim, D., & Kim, J. (2018). User Preference-Based Dual-Memory Neural Model With Memory Consolidation Approach. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2294–2308.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Ph.D. thesis Princeton University.

Nooralishahi, P., Loo, C. K., & Seera, M. (2018). Semi-supervised topo-Bayesian ARTMAP for noisy data. *Applied Soft Computing*, *62*, 134 – 147.

Oliveira, A. G., Martins, A., & Neto, A. D. (2018). Information state: A representation for dynamic processes using information theory. In *Proc. Int. Joint Conf. Neural Netw. (IJCNN)* (pp. 1–8).

Oliveira, A. G., Neto, A. D., & Martins, A. (2017). An analysis of information dynamic behavior using autoregressive models. *Entropy*, *19*, 1–13.

Oong, T. H., & Isa, N. A. M. (2014). Feature-Based Ordering Algorithm for Data Presentation of Fuzzy ARTMAP Ensembles. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 812–819.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, *9*, 62–66.

Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern Recognition*, *37*, 487 – 501.

Palaniappan, R., & Eswaran, C. (2009). Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. *Applied Soft Computing*, *9*, 100–106.

Palmer-Brown, D., & Lee, S. W. (2005). Continuous reinforced snap-drift learning in a neural architecture for proxylet selection in active computer networks. *International Journal of Simulation: Systems, Science and Technology*, *6*, 11–21.

Palmero, G. I. S., Dimitriadis, Y. A., Izquierdo, J. M. C., Sánchez, E. G., & Hernández, E. P. (2000). ART-Based Model Set for Pattern Recognition: FasArt Family. In H. Bunke, & A. Kandel (Eds.), *Neuro-Fuzzy Pattern Recognition* (pp. 145–175). World Scientific.

Pampalk, E., Rauber, A., & Merkl, D. (2002). Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In J. R. Dorronsoro (Ed.), *Artificial Neural Networks - ICANN 2002* (pp. 871–876). Springer Berlin Heidelberg volume 2415 of *Lecture Notes in Computer Science*.

Park, G., Choi, J., & Kim, J. (2019). Developmental Resonance Network. *IEEE Transactions on Neural Networks and Learning Systems*, *30*, 1278–1284.

Park, G., & Kim, J. (2016). Deep Adaptive Resonance Theory for learning biologically inspired episodic memory. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 5174–5180).

Park, G., Yoo, Y., Kim, D., & Kim, J. (2018). Deep ART Neural Model for Biologically Inspired Episodic Memory and Its Application to Task Performance of Robots. *IEEE Transactions on Cybernetics*, *48*, 1786–1799.

Park, G.-M., Yoo, Y.-H., & Kim, J.-H. (2015). REM-ART: Reward-based electromagnetic adaptive resonance theory. In *Proc. Int. Conf. Artif. Intell. (ICAI)* (pp. 805–811). volume 1.

Parrado-Hernández, E., Gómez-Sánchez, E., & Dimitriadis, Y. A. (2003). Study of distributed learning as a solution to category proliferation in Fuzzy ARTMAP based neural systems. *Neural Networks*, *16*, 1039 – 1057.

Parrado-Hernández, E., Gómez-Sánchez, E., Dimitriadis, Y. A., & Coronado, J. L. (1999). A neuro-fuzzy system that uses distributed learning for compact rule set generation. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 441–446). volume 3.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, *33*, 1065–1076.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pourpanah, F., Lim, C. P., & Saleh, J. M. (2016). A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction. *Expert Systems with Applications*, *49*, 74 – 85.

Prim, R. C. (1957). Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal*, *36*, 1389–1401.

Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. (1st ed.). Springer Publishing Company, Incorporated.

Raijmakers, M. E., & Molenaar, P. C. (1997). Exact ART: A Complete Implementation of an ART Network. *Neural Networks*, *10*, 649 – 669.

RamaKrishna, K., Ramam, V. A., & Rao, R. S. (2014). Mathematical Neural Network (MaNN) Models Part III: ART and ARTMAP in OMNI_METRICS. *Journal of Applicable Chemistry*, *3*, 919 – 989.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, *66*, 846–850.

Rao, H. A., & Gu, P. (1995). A multi-constraint neural network for the pragmatic design of cellular manufacturing systems. *International Journal of Production Research*, *33*, 1049–1070.

Rao, S., Martins, A., & Principe, J. C. (2009). Mean shift: An information theoretic perspective. *Pattern Recognition Letters*, *30*, 222 – 230.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, *97*, 285 – 308.

Rawashdeh, M., & Ralescu, A. (2012). Center-wise intra-inter silhouettes. In E. Hüllermeier, S. Link, T. Fober, & B. Seeger (Eds.), *Scalable Uncertainty Management* (pp. 406–419). Berlin, Heidelberg: Springer.

Rényi, A. (1961). On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics* (pp. 547–561). Berkeley, Calif.: University of California Press.

Rezaei, M., & Fränti, P. (2016). Set matching measures for external cluster validity. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 2173–2186.

Roure, J., & Talavera, L. (1998). Robust Incremental Clustering with Bad Instance Orderings: A New Strategy. In H. Coelho (Ed.), *Progress in Artificial Intelligence — IBERAMIA 98* (pp. 136–147). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53 – 65.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1* (pp. 318–362). Cambridge, MA, USA: MIT Press.

Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. Technical Report CUED/F-INFENG/TR 166 Engineering Department, Cambridge University.

Sammon, J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, *C-18*, 401–409.

Sanchez, E. G., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & Coronado, J. L. (2000). MicroARTMAP: use of mutual information for category reduction in fuzzy ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 47–52). volume 6.

Santamaria, I., Pokharel, P. P., & Principe, J. C. (2006). Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Transactions on Signal Processing*, *54*, 2187–2197.

Sapozhnikova, E. P. (2009). ART-Based Neural Networks for Multi-label Classification. In N. M. Adams, C. Robardet, A. Siebes, & J.-F. Boulicaut (Eds.), *Advances in Intelligent Data Analysis VIII* (pp. 167–177). Berlin, Heidelberg: Springer volume 5772.

Sasu, L. M., & Andonie, R. (2012). Function Approximation with ARTMAP Architectures. *International Journal of Computers, Communications & Control*, *7*, 957–967.

Sasu, L. M., & Andonie, R. (2013). Bayesian ARTMAP for regression. *Neural Networks*, *46*, 23 – 31.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.*, *42*, 19:1–19:21.

Seiffertt, J., & Wunsch II, D. C. (2010). *Unified Computational Intelligence for Complex Systems* volume 6 of *Evolutionary Learning and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Serrano-Gotarredona, T., & Linares-Barranco, B. (1996). A Modified ART 1 Algorithm more Suitable for VLSI Implementations. *Neural Networks*, *9*, 1025 – 1043.

Serrano-Gotarredona, T., Linares-Barranco, B., & Andreou, A. G. (1998). *Adaptive Resonance Theory Microchips: Circuit Design Techniques*. Norwell, MA, USA: Kluwer Academic Publishers.

Shang, F., Jiao, L., Shi, J., Wang, F., & Gong, M. (2012). Fast affinity propagation clustering: A multilevel approach. *Pattern Recognition*, *45*, 474 – 486.

Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, *16*, 30–34.

Silva, J. d. A., & Hruschka, E. R. (2016). A support system for clustering data streams with a variable number of clusters. *ACM Trans. Auton. Adapt. Syst.*, *11*, 11:1–11:26.

Simpson, P. K. (1992). Fuzzy min-max neural networks. i. classification. *IEEE Transactions on Neural Networks*, *3*, 776–786.

Simpson, P. K. (1993). Fuzzy min-max neural networks - part 2: Clustering. *IEEE Transactions on Fuzzy Systems*, *1*, 32–.

Sit, W. Y., Mak, L. O., & Ng, G. W. (2009). Managing Category Proliferation in Fuzzy ARTMAP Caused by Overlapping Classes. *IEEE Transactions on Neural Networks*, *20*, 1244–1253.

Sledge, I. J., & Keller, J. M. (2008). Growing neural gas for temporal clustering. In *2008 19th International Conference on Pattern Recognition* (pp. 1–4).

Sledge, I. J., Keller, J. M., & Alexander, G. L. (2008). Emergent trend detection in diurnal activity. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 3815–3818).

Smith, C., & Wunsch II, D. C. (2015). Particle Swarm Optimization in an adaptive resonance framework. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–4).

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109 – 118.

Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, *2*, 568–576.

Srinivasa, N. (1997). Learning and generalization of noisy mappings using a modified probart neural network. *IEEE Transactions on Signal Processing*, *45*, 2533–2550.

Srinivasulu, A., & Dakshayani, G. (2016). Clustering Large Data with Mixed Values Using Extended Fuzzy Adaptive Resonance Theory. *Indonesian Journal of Electrical Engineering and Computer Science*, *4*, 617–628.

Su, M.-C., DeClaris, N., & Liu, T.-K. (1997). Application of neural networks in cluster analysis. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1–6). volume 1.

Su, M.-C., & Liu, T.-K. (2001). Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, *37*, 165 – 175.

Su, M.-C., & Liu, Y.-C. (2002). A hierarchical approach to ART-like clustering algorithm. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 788–793). volume 1.

Su, M.-C., & Liu, Y.-C. (2005). A new approach to clustering data with arbitrary shapes. *Pattern Recognition*, *38*, 1887 – 1901.

Subagdja, B., & Tan, A.-H. (2012). iFALCON: A neural architecture for hierarchical planning. *Neurocomputing*, *86*, 124 – 139.

Subagdja, B., & Tan, A.-H. (2015). Neural modeling of sequential inferences and learning over episodic memory. *Neurocomputing*, *161*, 229 – 242.

Subagdja, B., Wang, W., Tan, A.-H., Tan, Y.-S., & Teow, L.-N. (2012). Memory Formation, Consolidation, and Forgetting in Learning Agents. In *Proc. 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 1007–1014). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems volume 2 of *AAMAS'12*.

Sutton, R. S., & Barto, A. G. (2018). *Introduction to Reinforcement Learning*. (2nd ed.). Cambridge, MA, USA: MIT Press.

Swope, J. A. (2012). ARTdECOS, adaptive evolving connectionist model and application to heart rate variability. *Evolving Systems*, *3*, 95–109.

Taşdemir, K. (2010). Graph based representations of density distribution and distances for self-organizing maps. *IEEE Transactions on Neural Networks*, *21*, 520–526.

Taşdemir, K. (2012). Vector quantization based approximate spectral clustering of large datasets. *Pattern Recognition*, *45*, 3034 – 3044.

Taşdemir, K., & Merényi, E. (2005). Considering topology in the clustering of Self-Organizing Maps. In *WSOM 2005 - 5th Workshop on Self-Organizing Maps* (pp. 439–446).

Taşdemir, K., & Merényi, E. (2006). Data topology visualization for the Self-Organizing Maps. In *Proc. 14th European Symposium on Artificial Neural Networks (ESANN 2006)* (pp. 277–282).

Taşdemir, K., & Merényi, E. (2007). A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. In *The 2007 International Joint Conference on Neural Networks (IJCNN)* (pp. 2205–2211).

Taşdemir, K., & Merényi, E. (2009). Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *20*, 549–562.

Taşdemir, K., & Merényi, E. (2011). A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*, 1039–1053.

Taşdemir, K., Milenov, P., & Tapsall, B. (2011). Topology-based hierarchical clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, *22*, 474–485.

Talavera, L., & Roure, J. (1998). A buffering strategy to avoid ordering effects in clustering. In C. Nédellec, & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (pp. 316–321). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H. (1995). Adaptive Resonance Associative Map. *Neural Networks*, *8*, 437 – 446.

Tan, A.-H. (1997). Cascade ARTMAP: integrating neural computation and symbolic knowledge processing. *IEEE Transactions on Neural Networks*, *8*, 237–250.

Tan, A.-H. (2004). FALCON: a fusion architecture for learning, cognition, and navigation. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 3297–3302). volume 4.

Tan, A.-H. (2006). Self-organizing Neural Architecture for Reinforcement Learning. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 470–475). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence Through Interaction: Towards a Unified Theory for Learning. In D. Liu, S. Fei, Z.-G. Hou, H. Zhang, & C. Sun (Eds.), *Advances in Neural Networks - ISNN 2007* (pp. 1094–1103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tan, A.-H., Feng, Y.-H., & Ong, Y.-S. (2010). A self-organizing neural architecture integrating desire, intention and reinforcement learning. *Neurocomputing*, *73*, 1465 – 1477. Advances in Computational Intelligence and Learning.

Tan, A.-H., Lu, N., & Xiao, D. (2008). Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning With Delayed Evaluative Feedback. *IEEE Transactions on Neural Networks*, *19*, 230–244.

Tan, A.-H., Ong, Y.-S., & Tapanuj, A. (2011). A hybrid agent architecture integrating desire, intention and reinforcement learning. *Expert Systems with Applications*, *38*, 8477 – 8487.

Tan, A.-H., Subagdja, B., Wang, D., & Meng, L. (2019). Self-organizing neural networks for universal learning and multimodal memory encoding. *Neural Networks*, *this issue*.

Tan, A.-H., & Xiao, D. (2005). Self-organizing cognitive agents and reinforcement learning in multi-agent environment. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 351–357).

Tan, S. C., Rao, M. V. C., & Lim, C. P. (2009). An online pruning strategy for supervised ARTMAP-based neural networks. *Neural Computing and Applications*, *18*, 387–395.

Tang, X.-l., & Han, M. (2010). Semi-supervised Bayesian ARTMAP. *Applied Intelligence*, *33*, 302–317.

Tapan, M. S. Z., & Siong, T. C. (2008). AC-ViSOM: Hybridising the modified Adaptive Coordinate (AC) and ViSOM for data visualization. In *2008 International Symposium on Information Technology* (pp. 1–8). volume 3.

Tapan, M. S. Z., & Teh, C. S. (2007). Hybridization of Learning Vector Quantization (LVQ) and Adaptive Coordinates (AC) for data classification and visualization. In *International Conference on Intelligent and Advanced Systems (ICIAS'07)* (pp. 505–510).

Teh, C. S., & Sarwar, Z. T. M. (2008). A hybrid supervised ANN for classification and data visualization. In *The 2008 IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 555–562).

Tou, J. T., & Gonzalez, R. C. (1974). *Pattern recognition principles*. Addison-Wesley,.

Trawiński, B., Smętek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, *22*, 867 – 881.

Tsay, S. W., & Newcomb, R. W. (1991). VLSI implementation of ART1 memories. *IEEE Transactions on Neural Networks*, *2*, 214–221.

Tscherepanow, M. (2010). TopoART: A Topology Learning Hierarchical ART Network. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 157–167). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2011). An Extended TopoART Network for the Stable On-line Learning of Regression Functions. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *International Conference on Neural Information Processing (ICONIP)* (pp. 562–571). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tscherepanow, M. (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In P. Perner (Ed.), *Proc. Industrial Conference on Data Mining (ICDM)* (pp. 22–34). ibai-publishing.

Tscherepanow, M., Kortkamp, M., & Kammer, M. (2011). A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*, *24*, 906 – 916.

Tscherepanow, M., Kühnel, S., & Riechers, S. (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 24–29).

Tscherepanow, M., & Riechers, S. (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In V. Lemaire, J.-C. Lamirel, & P. Cuxac (Eds.), *Proceedings of the ECAI Workshop on Active and Incremental Learning (AIL)* (pp. 18–23).

Tyree, E. W., & Long, J. A. (1999). The use of linked line segments for cluster representation and data reduction. *Pattern Recognition Letters*, *20*, 21 – 29.

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification. In O. Opitz, B. Lausen, & R. Klar (Eds.), *Information and Classification: Concepts, Methods and Applications* (pp. 307–313). Springer Berlin Heidelberg.

Ultsch, A. (2003). Maps for the Visualization of high-dimensional Data Spaces. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)* (pp. 225–230).

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps (WSOM)* (pp. 75–82).

Ultsch, A., & Siemon, H. P. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. International Neural Networks Conference (INNC)* (pp. 305–308).

Uray, M., Skočaj, D., Roth, P. M., & Bischof, H. (2007). Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches. In *Proceedings of the British Machine Vision Conference 2007, University of Warwick, UK* (pp. 1–10).

Vakil-Baghmisheh, M.-T., & Pavešić, N. (2003). A Fast Simplified Fuzzy ARTMAP Network. *Neural Processing Letters*, *17*, 273–316.

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1273–1280.

Vendramin, L., Campello, R. J. G. B., & Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, *3*, 209–235.

Versace, M., Kozma, R. T., & Wunsch, D. C. (2012). Adaptive Resonance Theory Design in Mixed Memristive-Fuzzy Hardware. In R. Kozma, R. E. Pino, & G. E. Pazienza (Eds.), *Advances in Neuromorphic Memristor Science and Applications* (pp. 133–153). Dordrecht: Springer Netherlands.

Verzi, S. J., Heileman, G. L., & Georgiopoulos, M. (2006). Boosted ARTMAP: Modifications to fuzzy ARTMAP motivated by boosting theory. *Neural Networks*, *19*, 446 – 468.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Anagnostopoulos, G. (2002). Off-line structural risk minimization and BARTMAP-S. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 2533–2538). volume 3.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Anagnostopoulos, G. C. (2003). Universal approximation with Fuzzy ART and Fuzzy ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1987–1992). volume 3.

Verzi, S. J., Heileman, G. L., Georgiopoulos, M., & Healy, M. J. (1998). Boosted ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 396–401). volume 1.

Verzi, S. J., Heileman, G. L., Georgiopoulus, M., & Healy, M. J. (2001). Rademacher penalization applied to fuzzy ARTMAP and boosted ARTMAP. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1191–1196). volume 2.

Vesanto, J. (1999). SOM-based data visualization methods. *Intelligent Data Analysis*, *3*, 111 – 126.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*, 586–600.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (1999). Self-Organizing Map in Matlab: the SOM Toolbox. In *Proceedings of the Matlab DSP Conference* (pp. 35–40).

Vigdor, B., & Lerner, B. (2006). Accurate and Fast Off and Online Fuzzy ARTMAP-Based Image Classification With Application to Genetic Abnormality Diagnosis. *IEEE Transactions on Neural Networks*, *17*, 1288–1300.

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*, 1628–1644.

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, *11*, 2837–2854.

Vitter, J. S. (1985). Random Sampling with a Reservoir. *ACM Trans. Math. Softw.*, *11*, 37–57.

Švaco, M., Jerbić, B., & Šuligoj, F. (2014). ARTgrid: A Two-level Learning Architecture Based on Adaptive Resonance Theory. *Advances in Artificial Neural Systems*, *2014*, 1–9.

Wang, D., Subagdja, B., Tan, A.-H., & Ng, G.-W. (2009a). Creating human-like autonomous players in real-time first person shooter computer games. In *Proc. Twenty-First Innovative Applications of Artificial Intelligence Conference* (pp. 173 – 178).

Wang, D., & Tan, A. (2015). Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game. *IEEE Transactions on Computational Intelligence and AI in Games*, *7*, 123–138.

Wang, K., Wang, B., & Peng, L. (2009b). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93.

Wang, L. (1997). On competitive learning. *IEEE Transactions on Neural Networks*, *8*, 1214–1217.

Wang, L., Geng, X., Bezdek, J., Leckie, C., & Kotagiri, R. (2008). SpecVAT: Enhanced Visual Cluster Analysis. In *The 2008 IEEE International Conference on Data Mining (ICDM)* (pp. 638–647).

Wang, L., Leckie, C., Kotagiri, R., & Bezdek, J. (2011). Approximate pairwise clustering for large data sets via sampling plus extension. *Pattern Recognition*, *44*, 222 – 235.

Wang, L., Nguyen, U. T. V., Bezdek, J. C., Leckie, C. A., & Ramamohanarao, K. (2010a). iVAT and aVAT: Enhanced Visual Analysis for Cluster Tendency Assessment. In M. J. Zaki, J. X. Yu, B. Ravindran, & V. Pudi (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 16–27). Berlin, Heidelberg: Springer Berlin Heidelberg.

Wang, L., Nguyen, U. T. V., Bezdek, J. C., Leckie, C. A., & Ramamohanarao, K. (2010b). iVAT and aVAT: Enhanced Visual Analysis for Cluster Tendency Assessment. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'10)* (pp. 16–27).

Wang, W., Subagdja, B., Tan, A.-H., & Starzyk, J. A. (2010). A self-organizing approach to episodic memory modeling. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Wang, W., Subagdja, B., Tan, A.-H., & Starzyk, J. A. (2012a). Neural Modeling of Episodic Memory: Encoding, Retrieval, and Forgetting. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1574–1586.

Wang, W., Subagdja, B., Tan, A.-H., & Tan, Y.-S. (2012b). A self-organizing multi-memory system for autonomous agents. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Wang, W., Tan, A., & Teow, L. (2017). Semantic Memory Modeling and Memory Interaction in Learning Agents. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*, 2882–2895.

Wang, W., & Zhang, Y. (2007). On fuzzy cluster validity indices. *Fuzzy Sets and Systems*, *158*, 2095 – 2117.

Wang, X. (2011). A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality. In *The 2011 International Joint Conference on Neural Networks (IJCNN)* (pp. 1293–1299).

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.

Wenzel, S., & Förstner, W. (2009). *The Role of Sequences for Incremental Learning*. Technical Report TR-IGG-P-2009-04 Department of Photogrammetry, University of Bonn.

Wenzel, S., & Hotz, L. (2010). The Role of Sequences for Incremental Learning. In J. Filipe, A. L. N. Fred, & B. Sharp (Eds.), *ICAART 2010 - Proceedings of the International Conference on Agents and Artificial Intelligence* (pp. 434–439). Valencia, Spain: INSTICC Press volume 1.

Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis Harvard University.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*, 1550–1560.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*, 80–83.

Williamson, J. R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*, *9*, 881 – 897.

Wu, S., & Chow, T. W. S. (2005). PRSOM: a new visualization method by hybridizing multidimensional scaling and self-organizing map. *IEEE Transactions on Neural Networks*, *16*, 1362–1380.

Wunsch II, D. C. (1991). *An optoelectronic learning machine: invention, experimentation, analysis of first hardware implementation of the ART 1 neural network*. Ph.D. thesis University of Washington.

Wunsch II, D. C. (2009). ART properties of interest in engineering applications. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 3380–3383).

Wunsch II, D. C., Caudell, T. P., Capps, C. D., Marks, R. J., & Falk, R. A. (1993). An optoelectronic implementation of the adaptive resonance neural network. *IEEE Transactions on Neural Networks*, *4*, 673–684.

Xiao, D., & Tan, A. (2007). Self-Organizing Neural Architectures and Cooperative Learning in a Multiagent Environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *37*, 1567–1580.

Xie, X. L., & Beni, G. (1991). A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, *13*, 841–847.

Xu, L., Chow, T. W. S., & Ma, E. W. M. (2015). Topology-Based Clustering Using Polar Self-Organizing Map. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 798–808.

Xu, L., Xu, Y., & Chow, T. W. S. (2010). PolSOM: A new method for multidimensional data visualization. *Pattern Recognition*, *43*, 1668 – 1675.

Xu, R., & Wunsch II, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Xu, R., & Wunsch II, D. C. (2009). *Clustering*. Wiley-IEEE Press.

Xu, R., & Wunsch II, D. C. (2010). Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, *3*, 120–154.

Xu, R., & Wunsch II, D. C. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, *24*, 709–716.

Xu, R., Wunsch II, D. C., & Kim, S. (2012a). Methods and systems for biclustering algorithm. U.S. Patent 9,043,326 Filed January 28, 2012, claiming priority to Provisional U.S. Patent Application, January 28, 2011, issued May 26, 2015.

Xu, R., Xu, J., & Wunsch, D. C. (2012b). A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*, 1243–1256.

Xu, Y., Xu, L., & Chow, T. W. S. (2011). PPoSOM: A new variant of PolSOM by using probabilistic assignment for multidimensional data visualization. *Neurocomputing*, *74*, 2018 – 2027.

Yang, M.-S., & Wu, K.-L. (2001). A new validity index for fuzzy clustering. In *10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297)* (pp. 89–92). volume 1.

Yap, K. S., Lim, C. P., & Abidin, I. Z. (2008). A Hybrid ART-GRNN Online Learning Neural Network With a ε-Insensitive Loss Function. *IEEE Transactions on Neural Networks*, *19*, 1641–1646.

Yap, K. S., Lim, C. P., & Au, M. T. (2011). Improved GART Neural Network Model for Pattern Classification and Rule Extraction With Application to Power Systems. *IEEE Transactions on Neural Networks*, *22*, 2310–2323.

Yap, K. S., Lim, C. P., & Mohamad-Saleh, J. (2010). An enhanced generalized adaptive resonance theory neural network and its application to medical pattern classification. *Journal of Intelligent & Fuzzy Systems*, *21*, 65–78.

Yavaş, M., & Alpaslan, F. N. (2009). Behavior categorization using Correlation Based Adaptive Resonance Theory. In *Proc. 17th Mediterranean Conference on Control and Automation* (pp. 724–729).

Yavaş, M., & Alpaslan, F. N. (2012). Hierarchical behavior categorization using correlation based adaptive resonance theory. *Neurocomputing*, *77*, 71 – 81.

Yin, H. (2001). Visualisation Induced SOM (ViSOM). In N. Allinson, H. Yin, L. Allinson, & J. Slack (Eds.), *Advances in Self-Organising Maps* (pp. 81–88). Springer London.

Yin, H. (2002a). Data visualisation and manifold mapping using the ViSOM. *Neural Networks*, *15*, 1005 – 1016.

Yin, H. (2002b). ViSOM - a novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks*, *13*, 237–243.

Yin, H. (2008). On multidimensional scaling and the embedding of self-organising maps. *Neural Networks*, *21*, 160 – 169.

Yousuf, A., & Murphey, Y. L. (2010). A Supervised Fuzzy Adaptive Resonance Theory with Distributed Weight Update. In L. Zhang, B.-L. Lu, & J. Kwok (Eds.), *Advances in Neural Networks - ISNN 2010* (pp. 430–435). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338 – 353.

Zahn, C. T. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, *C-20*, 68–86.

Zhang, L., Wang, G., & Wang, W. (2006). A New Fuzzy ART Neural Network Based on Dual Competition and Resonance Technique. In J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, & H. Yin (Eds.), *Advances in Neural Networks - ISNN 2006* (pp. 792–797). Berlin, Heidelberg: Springer volume 3971.

Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A New Data Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery*, *1*, 141–182.

Zhang, X., & Li, Y. (1993). Self-organizing map as a new method for clustering and data analysis. In *The 1993 International Joint Conference on Neural Networks (IJCNN)* (pp. 2448–2451). volume 3.

Zhang, Y., Ji, H., & Zhang, W. (2014). TPPFAM: Use of threshold and posterior probability for category reduction in fuzzy ARTMAP. *Neurocomputing*, *124*, 63 – 71.

Zhao, Q., & Fränti, P. (2014). WB-index: A sum-of-squares based index for cluster validity. *Data & Knowledge Engineering*, *92*, 77 – 89.

Zhao, Q., Xu, M., & Fränti, P. (2009). Sum-of-Squares Based Cluster Validity Index and Significance Analysis. In M. Kolehmainen, P. Toivanen, & B. Beliczynski (Eds.), *Adaptive and Natural Computing Algorithms* (pp. 313–322). Berlin, Heidelberg: Springer Berlin Heidelberg.

VITA

Leonardo Enzo Brito da Silva received both his bachelor degree in Electrical Engineering and master of science degree in Electrical and Computer Engineering from Universidade Federal do Rio Grande do Norte (UFRN), Natal-RN, Brazil, in 2011 and 2013, respectively. At UFRN, he was the recipient of the 2006 "Comenda Joel Dantas" Award for ranking first place in the admissions exam for the Electrical Engineering undergraduate program. In 2009, he was granted an undergraduate fellowship by BRAFITEC-CAPES Foundation under the Brazilian Ministry of Education to attend a semester at the State University of New York at Buffalo (SUNY-UB) in a non-degree undergraduate exchange program in Electrical Engineering.

In 2014, he was granted a scholarship by the CAPES Foundation under the Brazilian Ministry of Education to pursue his Ph.D. degree in Computer Engineering at the Missouri University of Science and Technology (Missouri S&T). He developed his research under the supervision of Prof. Donald C. Wunsch II and became a member of the Applied Computational Intelligence Laboratory (ACIL), where he was a Graduate Research Assistant. At Missouri S&T, Leonardo was one of the recipients of the 2019 "Dean's Ph.D. Scholar" Award "*In recognition of scholarship excellence, academic success, and service contributions to Missouri University of Science and Technology*." He received his Ph.D. degree in Computer Engineering from Missouri S&T in December 2019.

Leonardo's research focused on unsupervised machine learning methods, and his findings were published in peer-reviewed conference proceedings and journal articles under the umbrella of the Institute of Electrical and Electronics Engineers (IEEE) and the International Neural Network Society (INNS). He served as a reviewer for journals such as *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Signal Processing Letters*, and Elsevier's *Neural Networks*.