2015-11-01

CVIC: Cluster Validation Using Instance-Based Confidences

Dean M. LeBaron
Brigham Young University - Provo

CVIC: Cluster Validation Using Instance-Based Confidences

Dean M. LeBaron

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Tony Martinez, Chair
Christophe Giraud-Carrier
Seth Holladay

Department of Computer Science

Brigham Young University

November 2015

ABSTRACT

CVIC: Cluster Validation Using Instance-Based Confidences

Dean M. LeBaron
Department of Computer Science, BYU
Master of Science

As unlabeled data becomes increasingly available, the need for robust data mining techniques increases as well. Clustering is a common data mining tool which seeks to find related, independent patterns in data called clusters. The cluster validation problem addresses the question of how well a given clustering fits the data set. We present CVIC (**c**luster **v**alidation using **i**nstance-based **c**onfidences) which assigns confidence scores to each individual instance, as opposed to more traditional methods which focus on the clusters themselves. CVIC trains supervised learners to recreate the clustering, and instances are scored based on output from the learners which corresponds to the confidence that the instance was clustered correctly. One consequence of individually validated instances is the ability to direct users to instances in a cluster that are either potentially misclustered or correctly clustered. Instances with low confidences can either be manually inspected or reclustered and instances with high confidences can be automatically labeled. We compare CVIC to three competing methods for assigning confidence scores and show results on CVIC's ability to successfully assign scores that result in higher average precision and recall for detecting misclustered and correctly clustered instances across five clustering algorithms on twenty data sets including handwritten historical image data provided by Ancestry.com.

Keywords: clustering, validation, cluster confidence, supervised learners

ACKNOWLEDGMENTS

I guess this is where I get to break from technical white paper voice and say whatever I feel like. Radical. I'd like to thank my wife, Katie, for her patience, love, support, sympathy, and most importantly believing in me. I'd like to thank Jonathan Willis and my dad, Lynn LeBaron, for getting me into computer science; one through enthuastic persistance and the other not so much.

I'd also like to acknowledge my advisor Dr. Martinez and my commitee members for their input, guidance and feedback into this whole process. Thank you.

– Dean

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Because of the ill-defined nature of clustering there are several cluster validity indices which seek to inform the user of the goodness or quality of a clustering [10, 11, 23, 42]. In general, these indices are used to select the optimal number of clusters, k, for algorithms that are parametrized by k. This is known as the cluster validity problem. While this is an important question to answer, one application of validity indices that has not been well explored is instance level confidence scores. Such instance level scores can validate a clustering by detecting both high and low confidence points, which in turn informs the end-user of the quality of the clustering at a more granular level.

We present CVIC: a **C**luster **V**alidation method using **I**nstance-based **C**onfidences. CVIC is a technique that looks at every individual instance in a given clustering and assigns a confidence score to each instance. These scores represent the belief that a given instance belongs to a cluster and therefore has the same class as the rest of the instances in the cluster. This is accomplished in a pseudo-supervised fashion using the clustering itself to assign targets to the data which can then be used to train any traditional supervised model such as a neural network, support vector machine, decision tree, etc. The output of the learner reflects the confidence that a given instance should have a particular label. We show that CVIC assigns confidence scores to clustered instances that can subsequently be used to discover misclustered or correctly clustered instances with higher precision and recall rates than other competing distance based techniques from the literature. Moreover, we show that

CVIC provides a unique approach to cluster validation which is partly responsible for its success.

In clustering applications there are no ground-truth labels and clustering is generally performed in order to create or discover classes. The application that we are primarily concerned with is when clustering is used to efficiently assign labels to clusters of data as opposed to manually labeling each instance. For example, handwritten images of U.S. census data can be clustered and the data can be labeled by simply labeling the clusters as opposed to each individual image. This process could save countless man-hours of indexing said images, but requires that the clusterings are acceptable. In Fig. 1.1, a clustering of census data has been performed. The six images in the figure all belong to the same cluster, and a human operator, only seeing this subset of a larger cluster, would label this cluster as "Nebraska", thereby labeling all the instances as "Nebraska" including the "Ger-German" image which should not be a member of this cluster. Therefore, a validation technique which focuses on individual instances is also needed, particularly one that can assign a confidence score to each instance allowing the discovery of the "Ger-German" instance and other similarly mislabeled instances regardless of how they became misclustered-whether through noise, outliers, poor distance metrics, or poor features. CVIC addresses this problem and its confidence scores can be thresholded so that points falling below the threshold, such as the "Ger-German" instance are tagged as needing further inspection and points above some threshold can be automatically labeled with confidence thus saving the work of manually labeling a data set. Our work stems from this type of word-spotting application, but has also been applied to more general clustering tasks, including documents and human faces.

Traditional validity indices do not typically assign individual scores, but rather give scores to the clusters themselves, potentially missing out on the advantages of considering each instance individually and calculating the confidence of whether it belongs to its current cluster or not.

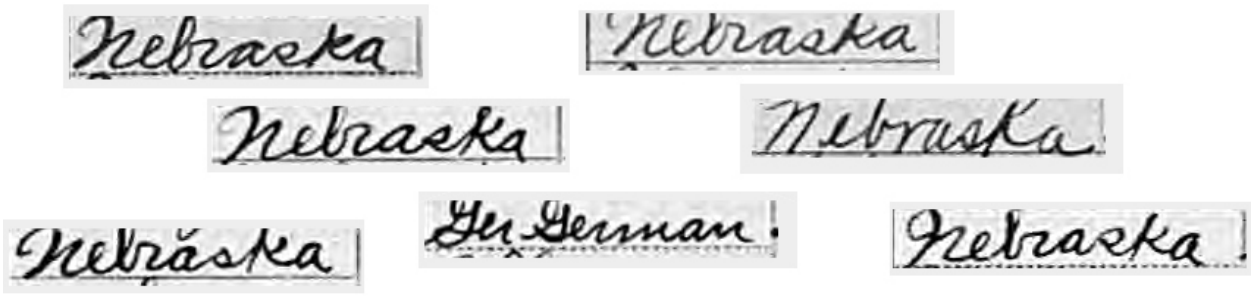


Figure 1.1: Part of a cluster of handwritten census data, including "Nebraska" and "Ger-German".

Chapter 2

Related Work

In this section we discuss work related to cluster validation and automatic cluster labeling, such as word-spotting. We introduce the notion of confidence scores and review indices in the literature that assign scores to individual instances.

2.1 Automated Cluster Labeling

One of the most common applications of using clustering to automatically label data is word spotting. Word spotting involves using handwritten images to index documents.
The main idea is that images are extracted from historical documents and then the images are clustered. The clusters are then manually labeled according to the text that is most common in the cluster. An operator labels a cluster by seeing either just one instance which is nearest the centroid or perhaps a few instances from the cluster. The label is then propagated to all the instances in the cluster. These labels are then used to build an index that allows retrieval of the documents. For example, a cluster of images containing the word congress can be labeled and then each document that the images were segmented from can now be accessed by a query containing the word congress similar to the example in Fig. 1.1. This approach would benefit from instance level confidence scores in order to aid the automated labelings.

Rath and Manmatha [31] used dynamic time warping to measure distances between images, and used this as a metric for k-means and hierarchical agglomerative clustering (HAC). They report that the word error rates of clustering 4860 images into 1365 clusters

were 41.58% for k-means. Using HAC with Ward linkage [41] the word error rate was 31.50%. They mention that these percentages are acceptable, but CVIC could conceivably be used to detect some erroneous instances from the clusterings, therefore lowering the error rates.

Sankar et al. [33] used a k-nearest neighbor (knn) classifier to automatically label cluster centroids, which are then used to label the rest of the cluster as opposed to having a human labeler. The knn classifier uses a training set which consists of labeled data similar to the data to be clustered. A test set, composed of unlabeled images which is the target of indexing, is then clustered using hierarchical k-means and the centroids are labeled by using the trained knn classifier to predict labels. Similar to our work, they are combining clustering and supervised learning, but their learner is trained with labeled data which reflects ground-truth that is not the same data set to be indexed and clustered, whereas CVIC does not train on labeled data that reflects ground truth, but rather training targets come from the clusters themselves. They also employ a threshold to remove unimportant clusters. This threshold is related to the score from knn, but since they are only classifying centroids, the entire cluster is removed if it falls below the threshold. We are interested instead in thresholding individual points, as opposed to clusters.

2.2 Confidence Scores

Our work relies on a confidence score, which is essentially a score of how confident our model is that an instance belongs in its cluster. Validity indices which assign scores specifically to individual instances–as opposed to scoring the entire clustering–are rare in the current literature. The silhouette index [32] is one which is commonly used, however, and is described below:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_i)\}}$$

where x_i is a single instance belonging to a cluster i, and $a(x_i)$ and $b(x_i)$ are defined as follows:

$$a(x_i) = \frac{1}{|C_i|} \sum_{y \in C_i} \|x_i - y\|$$

$$b(x_i) = min_{j=1,\ldots,k;j\neq i}(\frac{1}{|C_j|} \sum_{y \in C_j} \|x_i - y\|)$$

where C_i is the cluster that instance x_i belongs to, C_j is a cluster the instance is not a member of, and $y \in C$ is a point in the cluster.

The silhouette index scores instances between -1 and 1, where a 1 indicates that the instance was clustered correctly and a -1 indicates that the instance should belong to a different cluster. The score compares average dissimilarity within an instance's own cluster to the lowest average dissimilarity to any other cluster. This is a prime example of an index that rewards a well-separated cluster which is also dense. The main weakness with the silhouette is that in favoring well separated and dense clusters it therefore struggles with oddly shaped clusters, or clusterings with some dense clusters and some that are less dense. Specifically, in applications where the data follows a power law, or some other skewed distribution, silhouette favors the more densely populated clusters over the less dense ones; depending on the application. For example, with census images dense and sparse clusters can both have high purity, and an index which is more robust to density is needed.

[27] use a k-nearest neighbors approach to assign scores to individual instances. They created the isolation index which is described as follows:

$$I(x) = \frac{1}{k}\, v_k(x)$$

where $v_k(x)$ is the number of k-nearest neighbors of x that have the same cluster label as x. This score is simply the fraction of x's nearest neighbors that are in the same cluster as x. The isolation index therefore gives lower confidences to points that are near the border between another cluster and the cluster containing x. The index does not however give low confidences to points near the border of the containing cluster, unless another cluster is nearby. This can also be seen in the silhouette index, since a point on the far side of a cluster that is not near any other clusters will have a higher confidence score. This observation is further discussed in section 5.

Another score that can be used to determine how well an individual instance is clustered is the fuzzy membership score used in fuzzy clustering such as Fuzzy C-Means [4]. The membership is calculated as follows:

$$w(x_{i,j}) = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$$

where w_{ij} is the score for instance i belonging to cluster j, and c is the cluster centroid and m is a fuzzifier. Fuzzy clustering allows instances to have membership in multiple clusters, and this score could be interpreted as a type of confidence.

In our experiments we compare silhouette, isolation, and fuzzy membership scores to CVIC as a means to assign scores to the instances in a given clustering. Although silhouette, isolation, and fuzzy membership were not necessarily intended to be used as a basis for individual instance confidence, but rather as cluster validation measures, we can still make a valid comparison to CVIC since they assign some score to each individual instance. However, to the best of our knowledge, this type of instance-based *confidence* score does not exist

in the literature, and this is one of the main contributions of our work: a score which is concerned with determining if any given instance has the same label as the rest of its cluster. Silhouette, isolation, and fuzzy clustering do address this to a greater extent than any other methods we are aware of which is why we have included them for comparison.

2.3 Cluster Validation

The cluster validity problem involves determining the correct number of clusters which can be approximated by several different indices [20, 29]. Typically, a clustering is performed several times with increasing k values, and the index is calculated after each run. The correct k corresponds to the maximum, minimum, or elbow value of the index depending on which index is used. These values can also be used to compare different clusterings with the same k. Many indices struggle with oddly shaped clusters, or clusters with differing densities, and therefore newer indices have been designed to handle varying densities [15, 17, 43], but do not assign scores to individual instances.

In addition, most indices still face the problem of the curse of dimensionality [2] because of their reliance on a distance metric between points. In a high dimensional setting, the curse of dimensionality begins to be a concern as all points are essentially the same distance from each other [3]. CVIC uses a supervised classifier, which does not suffer from high dimensional data in the same way that something which measures distances would. However, it is generally understood that a supervised learner needs much more training data to work well with high dimensional data sets. [40].

Chapter 3

CVIC - Cluster Validation using Instance-based Confidences

In this section we describe the basic outline and implementation of CVIC. Figure 3.1 outlines CVIC and the following subsections explain the process.

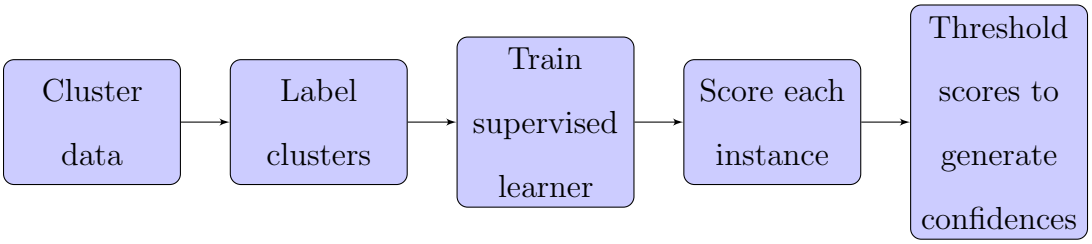


Figure 3.1: The first three nodes are modular in the sense that different clustering algorithms, labeling methods, and supervised learning models can be used.

3.1 Initial Clustering and Labeling

CVIC essentially sits on top of a clustering algorithm and is therefore invariant to the clustering algorithm itself. Our approach to instance-based confidences is through supervised models, though we do not use ground truth labels to train the models, but rather treat the given clustering as a sort of pseudo-labeling which can be used to train a model. Specifically, we treat each cluster as a label for all points contained in the cluster, e.g., points in some cluster A can be arbitrarily labeled as "A". This allows the supervised classifier to be trained. However, we have found that arbitrary labels do not work nearly as well as human assigned ones. This is due to the fact that in many real-world clustering applications there will be more than one cluster with the same ground truth label. For instance, census birthplace data

tends to be highly skewed toward the country or state in which the census took place. For example, the CANADA data set has roughly 54% of its instances with Canada as a ground truth label, and more than one cluster ends up being labeled as Canada as well. This is partly due to the fact that many different writers contributed to the census so one Canada image may look quite different than another image. The inherent difficulty of this type of clustering task contributes as well. However, the phenomenon of multiple clusters with the same label simplifies the learner because instead of having k output classes, where k is the number of clusters, there are m outputs, where m represents the number of uniquely labeled clusters. Though more common in census data, handwritten text, and object classification, we expect that often m will be less than k and this is one advantage of casting the problem into supervised classification with human assigned labels as opposed to traditional cluster validation. Namely, the classifier is learning the difference between points labeled as one class and points labeled otherwise. In the case of using human assigned labels CVIC is concerned with whether an instance belongs to a given class, regardless of how it was clustered. Although we have found that human labeled clusters help CVIC perform better than using completely arbitrary labels human assigned labels are not necessary to CVIC's approach.

However, in applications like word spotting, the manual labelling process will already be taking place since ground truth values are being discovered through clustering. Furthermore, the task of labelling the clusters by hand is efficient since the labeler only needs to label each cluster which is much smaller than the amount of total data. For labelling clusters, the operator is only shown the n closest images to the cluster's centroid, where n is small, and simply labels the cluster according to the majority class. These labels are then propagated to each point in the cluster. For example, the labeler could be shown the top 5 closest points to a centroid and assign the label "Germany" which is then automatically assigned to the remaining points in the cluster which may number in the thousands.

3.2 Training and Initial Scoring

The training of a supervised classifier is at the heart of CVIC's approach. While theoretically any classifier could be used we have implemented CVIC with multi-layer perceptrons, support vector machines, C4.5 trees, random forests, and k-nearest neighbor classifiers. These were selected for their relative success in classification problems, their relative independence from distance metrics (except KNN), their representation of a diverse set of algorithms, and especially for their ability to easily produce a confidence in the classification. We have also implemented a simple ensemble of these methods. It should be noted that all hyperparameters to the models were selected through trial-and-error analysis.

MLP

The training of the MLP follows a non-traditional setup because we want to expose the network to every single instance. In smaller data sets, a validation set for a stopping criteria is not feasible because the network is attempting to recreate the clustering and smaller clusters would possibly be under or over represented in a validation set. Therefore, instead of using a validation set, we include all instances in the training set and stop training after 5 epochs of no significant improvement where significance is defined as the mean sum squared error of the network decreasing relative to the previous epoch by at least .01.

Overfit is a very real problem that must be addressed to avoid creating high confidences in points that were actually clustered poorly. We regularize using dropout [37] at a rate of 10% and weight decay according to the following:

$$\Delta w_{ij}(t+1) = n\delta_j o_i - \lambda w_{ij}(t)$$

where Δw_{ij} is the change of weight from node i to j, δ_j is the error of node j, o_i is the output of node i, n is the learning rate, λ is a constant which we experimentally set to .0001, and t is time. After the network has been trained we calculate a confidence score for each instance

by running them through the network and getting a final classification. For each instance the score is calculated as:

$$conf(x) = \frac{1}{2} * \frac{e^{o(h)} + e^{o_{true}(x)}}{\sum_i^{|Y|} e^{o_i(x)}}$$

where $|Y|$ is the size of the output layer, *o(h)* is the output of the node with the highest activation, $o_{true(x)}$ is the output of the node which corresponds to x's label and $o_{i(x)}$ is the output of every node i in the output layer. Note that if the network correctly classifies the instance then $o(h) = o_{true(x)}$ and the score is simply a normalization of the activation, but if the instance was misclassified the confidence is the average of these two values. Essentially, for misclassified instances we are lowering the score even more than a simple normalization would.

SVM

The training of the SVM follows a similar philosophy to the MLP, namely studiously avoid overfit and obtain confidence scores that take into account the use of the labels to penalize misclassified instances even more. For our SVMs we leverage the popular LibSVM library [7]. We use degree 2 polynomial kernels, and change the epsilon parameter which is responsible for the tolerance of the stopping criteria from the default of .0001 to .1, again in an attempt to avoid the possibility of overfit.

Confidence scores from the SVM are obtained through Platt scaling (a logistic regression based on the output values), which is provided by the library [28]. Similar to the MLP, the Platt value from the predicted classification is averaged with the value from the SVM that corresponds to the instance's true label (we are averaging results from different SVMs due to their binary output limitations). The two values will only be different if the model misclassifies the instance.

C4.5 Tree

Our C4.5 tree follows the basic algorithm outlined in Quinlan [30]. Splitting is done according to the information gain criterion, and the tree is not allowed to exceed a number of leaf nodes equal to the number of features in the data set. This can be thought of as a type of pruning, which is accomplished by splitting nodes with the most gain until the number of leaf nodes is reached. The confidence score is simply the purity of the leaf node that a given instance falls into.

Random Forest

50 Classification and Regression Trees (CART) [6] make up the random forest implementation of CVIC. These trees are trained on a random subset of features with a size equal to the square root of the number of features. CART uses Gini impurity as its splitting criteria which is a measure of the probability that an instance would be misclassified given the current probability distribution at a node. Pruning of trees follows the same method outlined above for the C4.5 tree. The confidence score is similar to that of the MLP in that it is the average of two numbers; the fraction of trees with the majority prediction and the fraction of trees with the correct prediction. These two numbers may be the same, but in the case of a misclassified instance the confidence score will be lower.

KNN

For our knn we use the five nearest neighbors to classify instances. There is no feature weighting that takes place and nearest neighbors are calculated according to manhattan distance. We use a cover tree to speed up the search and the confidence score is simply the fraction of neighbors that share the same label. This is slightly different than the isolation index because isolation uses cluster membership whereas our knn uses the labels assigned by

a user and therefore points may come from different clusters but have the same label. If we were to use arbitrary labellings the two methods would perform the same.

Ensemble

We implemented an ensemble of all of the learners described above taking inspiration from Smith's work on instance hardness [36]. The basic idea of instance hardness is that if several different learners misclassify or struggle to correctly classify a given instance then that instance can be thought of as "hard". Hard instances in our problem setting could be potentially misclustered instances, and easy instances are potentially clustered correctly. In order to get a confidence score using this type of ensemble we simply ran each instance through each of the different learners for a total of five models and set the confidence score for the instance equal to the fraction of learners that correctly classified the instance according to the cluster labels.

We also tried a "pseudo-ensemble" which simply normalizes the confidence scores from each different learner between zero and one. A single instance's confidence score was then calculated by normalizing across all instances the sum of the five separately normalized confidence scores from each learner. We found that this second approach for ensembling produces better results and it is used in our experiments.

3.3 Thresholding

With confidence scores for each instance, the end-user of CVIC now has an idea of the quality of the initial clustering and can be made aware of points which may merit human-inspection. However, these values do not directly correlate to the probability that an instance is clustered correctly, indeed the scores are relative and need to be considered in the context of the rest of the scores in a given cluster. In order to use these scores to find points which are misclustered,

we calculate a threshold τ as some fraction of standard deviations above or below the mean of all instance's confidence scores on a per cluster basis as follows:

$$\tau = \mu(con(\mathbf{X})) - \alpha\sigma(con(\mathbf{X}))$$

where $\mathbf{X}$ is a vector of all points in a given cluster, σ is the standard deviation, μ is the arithmetic mean, and α is set by the user, and represents the number of standard deviations above or below the mean the threshold value should be. A reasonable default for α is 1. Each cluster now has its own τ which can be used to mark all points falling below τ as potentially misclustered. These points can either be reclustered, or manually inspected/labeled.

3.4 Choosing K

We also present a solution to the problem of choosing the correct number of clusters. CVIC's confidence scores can be used in aggregate to suggest a good number of clusters. The formula for calculating an index based on confidence scores is presented below:

$$\frac{1}{k} * \sigma * |S| \qquad \text{(Eq. 1)}$$

where σ is the standard deviation, k is the number of clusters, and $|S|$ is the number of points with confidence scores below one standard deviation from the mean.

This index can be used similarly to other popular indices, simply by creating several clusterings with different k values and then choosing the value of k that minimizes the index. Intuitively, the index works as follows: If the standard deviation of the scores is low, then the clustering has little noise, and most instances are easy for the supervised learner to classify, meaning the σ term brings the score down. Similarly in a good clustering one would expect the size of S to be small since fewer points would be removed. This portion of the index also lowers the score. It is generally true that more clusters are preferred to fewer clusters and the *1/k* term serves to penalize small clusterings with only one or two clusters.

Chapter 4

Experiments and Results

This section describes our experimental setup and presents results on 20 different data sets using 5 different clustering algorithms. We compare CVIC to silhouette, fuzzy membership and isolation as three additional methods for assigning confidence scores to individual instances.

We show results for the average f-score of each method for finding correctly clustered points and misclustered points on each data set using neural gas clustering and include similar tables in the appendix for the other four clustering algorithms. We show results for the average number of points uniquely indentified by each method. We show results on choosing k on 12 synthetic data sets compared to several other validity methods. We also include a brief discussion on the time-completixy of the various algorithms.

We find that on average CVIC outperforms other instance-based validation techniques, and offers a unique perspective on cluster validation.

4.1 Clustering Algorithms

The five clustering algorithms described below were used in our experiments and were specifically chosen to represent a broad range of clustering approaches and especially for their speed. Seeing as some data sets used contain upwards of 50,000 points in hundreds of dimensions, non-quadratic time and space complexity of the algorithms used is highly desirable. However, the BIRCH and spectral clustering algorithms did cluster subsets of some

of the larger data sets rather than the entire data set to speed performance. In these cases the points that were not included in the clustered subset were mapped to the clusters so that every instance was still assigned a cluster.

Kmeans++

Kmeans++ [1] is a straightforward improvement to the classical k-means clustering algorithm. Specifically, it is an algorithm for choosing inital values, or "seeds", for the k-means algorithm in such a way as to avoid poor clusterings. It works by probabilistically choosing as initial points instances that are far from each other. Then standard k-means is run. This setup requires $O(k^2)$ time, but significantly decreases the time it takes for k-means to converge.

Neural Gas

Neural gas is a competitive clustering algorithm inspired by self-organizing maps [19]. During learning each neuron in the network sorts itself according to its distance from the current input signal. The sorting is then used to drive an adaptation step where each neuron is changed incrementally as in gradient descent. This process of adapting all neurons leads to a more robust convergence compared to k-means. It has a run time of $O(nki)$ where i is the number of iterations until convergence [22].

Growing Neural Gas

Growing neural gas is an extension to the neural gas algorithm, which can add and delete nodes during execution. This growth is based on competitive Hebbian learning. Additionally, growing neural gas has edge connections between nodes. Age of an edge helps the algorithm make decisions about which nodes to delete, and the edges also serve as indicators of where distinct clusters are. It runs in $O(nlog(n)i)$ time [12].

Spectral Clustering

Spectral clustering is a term for graph based clustering algorithms that use eigenvalues of the similarity matrix to first perform a dimensionality reduction (similar to PCA) and then cluster in the smaller space. For our experiments we use the random walk normalized Laplacian matrix for eigenvalue calculation [24], which has been shown to be mathmatically equivalent to the normalized cuts algorithm [34]. This type of decomposition naively takes $O(n^3)$ time, but approximation methods and sub-sampling of the matrix can be used to speed up this process to $O(k^3)$ where $k < n$ is the number of samples. Even so, we found it necessary to cluster a subset of the data for the larger data sets and then map the remaining points to the given clusters.

BIRCH

BIRCH, or balanced iterative reducing and clustering using hierarchies, is designed to perform hierarchical clustering on large data sets [44]. It first creates a clustering feature tree from a single scan of the data set. It then clusters the leaves of the tree in a hierarchical manner until a user specified k value is reached, or until a single cluster remains. In all, it makes two passes through the data set and has time complexity of $O(nm^2)$ where m is the number of leaf nodes to be clustered by hierarchical agglomerative clustering, or HAC.

4.2 Data sets

Table 4.1 details the 20 data sets used in the experiments. These data sets were chosen specifically because the resulting clusters can easily be labeled by a human, i.e., each data set is image based and contains distinguishable objects such as human faces, handwritten text, digits, etc. In the case of the BANKNOTES, SEEDS, and LEAF data sets a labeler with some domain knowledge of the visual differences between classes would be required. Of the

data sets tested some contain high-dimensional data which serves to emphasize the need for non-distance based measures and some contain lower dimensional data which helps to test the generalizability of CVIC. We manually labeled each cluster by viewing the 5 points closest to the centroid and assigning a label. Feature type is indicated in the table; CONFIRM is a line-based dynamic programming approach for feature extraction in documents [39], LEAF and SEEDS data sets both contain botanical features [8, 35], eye-tracking refers to features based on human visual tracking patterns when shown an image [26], HOG refers to Histogram of Oriented Gradients [9], Rath refers to the features used in [31] for word spotting, and wavelets refer to various features of the shift-invariant wavelet transformed images [13]. PADEATHS, WALES, WASHPASS, CANADA, MISSOURI, NEBRASKA, and NEVADA are all data sets provided by Ancestry.com and represent historical census-like data. COIL [25], UMIST [14], USPS [14], CVL [18], and IAM [21] images were all scaled down to create a smaller features space, since the input features are simply the raw grayscale pixel values. The final image sizes are 32x32, 23x28, 16x16, 40x20, and 20x15 respectively. It should be noted that the IAM data set is a subset of the original IAM data set, including only images that had less than 10,000 total pixels. This was done due to the higher complexity of the spectral and birch clustering algorithms, but is a representative subset of the original. The table (and subsequent tables) is sorted by the number of classes.

4.3 Experiment 1 - F-score of High Confidence Points

The goal of CVIC is to assign confidences to individual instances and therefore provide the end-user with a tool to find both instances which can be automatically labeled with confidence and which are potentially misclustered. We conduct experiments on both the success of finding high confidence points and low confidence points. One way to measure this is to calculate the f-score–the harmonic average of precision and recall–of the selected points. For our first experiment we consider high confidence points. The threshold τ is set to one

Name	Classes	Instances	Feature dimensions	Feature Type	Domain
IAM	6444	75213	300	pixels	handwritten text
CVL	390	99705	800	pixels	handwritten text
NEVADA_DTW	146	38453	931	Rath	handwritten text
NEVADA_HOG	146	38453	224	HOG	handwritten text
NEBRASKA_DTW	101	62446	956	Rath	handwritten text
NEBRASKA_HOG	101	62446	224	HOG	handwritten text
MISSOURI_DTW	93	51036	947	Rath	handwritten text
MISSOURI_HOG	93	51036	224	HOG	handwritten text
CANADA_DTW	91	5553	956	Rath	handwritten text
CANADA_HOG	91	5553	240	HOG	handwritten text
LEAF	36	340	16	domain-specific	leaf images
COIL	20	1440	1024	pixels	object images
UMIST	20	575	644	pixels	face images
WALES	11	4800	1715	CONFIRM	document images
POET	10	6258	16	eye-tracking	object images
USPS	10	2007	256	pixels	handwritten digits
PADEATHS	5	4974	403	CONFIRM	document images
SEEDS	3	210	7	domain-specific	seed images
BANKNOTES	2	1372	4	wavelets	banknote images
WASHPASS	2	2000	268	CONFIRM	document images

Table 4.1: Real-world data sets.

standard deviation above the mean confidence score for each individual cluster. Precision is calculated by counting the number of high confidence points that fall above the threshold τ that are correctly clustered and dividing by the total number of high confidence points above τ. Recall is calculated by taking the number of correctly clustered points above τ and dividing by all correct points. Each of the data sets tested have ground-truth values available and we use these in order to measure precision and recall, though in real applications of clustering these values would not be known. The methods used for learning confidence scores do not see these ground-truth labels; they are simply used to evaluate the performance of the respective algorithms for detecting misclustered and correctly clustered instances.

Table 4.2 below shows results for the average f-score (over five runs) for each data set using neural gas clustering. The CVIC-MLP, CVIC-SVM, CVIC-C4.5, CVIC-RForest, CVIC-KNN and CVIC-Ensemble columns are all different implementations of CVIC using the respective learners. We individually compare each one against the three distance based methods of silhouette, fuzzy, and isolation. We have taken this approach for two reasons. One is to perform a comparison between a single implementation of CVIC and the three other methods to help illustrate that the framework of CVIC is viable, namely that different supervised learners can be plugged in. The other reason is to try and discover if there exists a single supervised learner that performs most successfully in CVIC. Therefore, a bold and underlined CVIC value represents statistical significance over every distance-based method according to a Wilcoxon signed-rank test at α =.05. A distance-based method with a statistically significant value over all CVIC values is represented similarly. The median and average values of each column are shown at the bottom. Similar tables for the remaining four clustering methods are contained in the supplemental material section and neural gas was chosen as a representative of the others.

The results for this experiment show the power of CVIC to detect a much larger number of points than the other methods. In the case of the high dimensional handwritten

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.350	0.315	0.000	0.313	0.266	**0.505**	0.340	**0.567**	0.362
CVL	0.200	0.225	0.240	0.236	**0.336**	**0.287**	0.249	**0.314**	**0.268**
NV_DTW	0.225	0.269	0.260	**0.400**	**0.434**	**0.428**	**0.409**	**0.399**	**0.396**
NV_HOG	0.288	0.284	0.379	**0.523**	**0.546**	**0.504**	0.330	**0.580**	**0.460**
NB_DTW	0.254	0.267	0.270	**0.679**	**0.682**	**0.595**	**0.680**	**0.599**	**0.686**
NB_HOG	0.298	0.298	0.366	**0.753**	**0.765**	**0.754**	**0.395**	**0.772**	**0.723**
MO_DTW	0.272	0.291	0.310	**0.810**	**0.813**	**0.735**	**0.778**	**0.702**	**0.792**
MO_HOG	0.295	0.318	0.365	**0.845**	**0.845**	**0.842**	**0.769**	**0.847**	**0.817**
CAN_DTW	0.275	0.291	0.301	**0.705**	**0.759**	**0.753**	**0.462**	**0.679**	**0.637**
CAN_HOG	0.293	0.283	0.384	**0.839**	**0.818**	**0.838**	0.313	**0.761**	**0.704**
LEAF	0.471	0.351	**0.659**	0.524	0.338	0.570	0.529	0.594	0.555
COIL	0.199	0.298	**0.833**	0.733	0.570	0.702	0.381	0.724	0.470
UMIST	0.318	0.361	0.676	0.636	0.379	0.599	0.425	0.666	0.632
WALES	0.313	0.000	**0.657**	0.493	0.633	0.635	0.517	0.612	0.564
POET	0.246	0.194	0.357	0.265	0.372	0.341	0.373	0.331	0.351
USPS	0.316	0.374	0.783	**0.817**	0.801	0.802	0.729	**0.821**	**0.805**
PADEATHS	0.274	0.298	0.768	**0.838**	**0.844**	**0.846**	**0.793**	0.638	**0.809**
SEEDS	0.158	0.270	0.938	0.732	0.924	0.750	0.904	0.682	0.888
BANK	0.389	0.334	0.727	0.238	0.722	0.732	0.719	0.510	0.718
WASHPASS	0.168	0.222	**0.998**	0.947	0.934	0.943	0.851	0.666	0.900
MEDIAN	0.281	0.290	0.381	**0.692**	**0.701**	**0.717**	**0.489**	**0.651**	**0.661**
AVERAGE	0.280	0.277	0.513	**0.616**	**0.638**	**0.658**	**0.547**	**0.623**	**0.626**

Table 4.2: Average f-score of high confidence points over five runs for each method and data set using neural gas clustering. Bolded and underlined values indicate statistical significance of CVIC values greater than silhouette, fuzzy, or isolation and vice versa.

data sets the recall values for the CVIC implementations are much higher than for the distance based measures, and the precision values are slightly higher. This creates an f-score that is significantly better for these data sets, and the cause appears to be tied to the idea brought up in section 3.1 with the CANADA data set. Namely, the supervised learners are learning what it means for an instance to belong to a given class since multiple clusters end up with the same label and therefore CVIC can confidently select points that from a distance perspective might be quite far from the centroid, therefore lowering the confidence for a method like silhouette. The threshold τ can be manipulated through adjusting the number of standard deviations away from the mean to consider, but although this increases the recall of the distance methods it similarly will increase recall for CVIC.

On four of the data sets isolation significantly performs the best, but except in the case of WASHPASS, it has lower precision than CVIC methods. Thus the better f-score is a result of higher recall. We believe this is due in part to the fact that the learners have less data to work with. This is easy to see since the tables are ordered by number of classes which is closely tied to the size of the data set. The class distributions and number of points seem to contribute to the results in subsequent experiments as well. Also influential to isolation's performance is the simplicity of its bias. Isolation finds points near the border of other clusters and these points have lower confidences while points nearer the centroid have higher confidences. This type of approach works well for the six more uniform data sets whereas for more complicated clusterings and more complicated class distributions with long tails the biases of supervised models are more appropriate.

4.4 Experiment 2 - F-score of Low Confidence Points

While the previous section focused on the scoring methods' ability to determine high confidence points, this section looks at the other half of the problem, namely, are low confidence points actually misclustered? Depending on the end-user's goals this could be just as important as

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.058	0.002	0.000	**0.252**	**0.189**	**0.413**	**0.316**	**0.267**	**0.375**
CVL	0.194	0.105	0.000	**0.226**	**0.286**	**0.451**	**0.290**	0.155	**0.305**
NV_DTW	0.257	0.166	0.000	0.193	0.169	0.106	0.275	0.172	0.225
NV_HOG	0.272	0.136	0.252	**0.350**	**0.367**	**0.293**	**0.316**	**0.327**	**0.350**
NB_DTW	0.222	0.186	**0.384**	0.114	0.068	0.033	0.206	0.106	0.151
NB_HOG	0.286	0.155	0.259	**0.364**	**0.371**	0.250	**0.342**	**0.354**	**0.384**
MO_DTW	0.225	0.158	0.073	0.124	0.072	0.059	0.214	0.116	0.203
MO_HOG	0.327	0.203	0.272	0.315	0.323	0.261	0.344	0.294	**0.379**
CAN_DTW	0.247	0.127	**0.409**	0.300	0.242	0.190	0.336	0.264	0.344
CAN_HOG	0.308	0.264	0.322	0.337	0.204	0.241	0.270	0.316	0.315
LEAF	0.298	0.000	0.376	0.272	0.315	0.244	0.343	0.381	0.324
COIL	0.409	0.000	**0.454**	0.199	0.229	0.310	0.321	0.401	0.363
UMIST	0.364	0.039	**0.391**	0.196	0.241	0.169	0.323	0.342	0.310
WALES	**0.258**	0.000	0.010	0.068	0.172	0.075	0.181	0.010	0.160
POET	0.279	0.130	0.174	0.265	0.085	0.012	0.110	0.080	0.184
USPS	0.295	0.000	0.449	0.342	0.344	0.248	0.462	**0.500**	0.469
PADEATHS	**0.387**	0.135	0.368	0.106	0.118	0.049	0.290	0.157	0.207
SEEDS	0.526	0.548	**0.597**	0.542	0.487	0.074	0.411	0.490	0.527
BANK	**0.358**	**0.341**	0.103	0.251	0.099	0.041	0.094	0.051	0.121
WASHPASS	0.008	0.008	0.222	0.047	0.022	0.080	0.008	**0.400**	0.022
MEDIAN	0.282	0.132	0.265	0.251	0.216	0.179	**0.302**	0.280	**0.312**
AVERAGE	0.278	0.135	0.255	0.243	0.220	0.179	0.272	0.258	**0.285**

Table 4.3: Average f-score of low confidence points over five runs for each method and data set using neural gas clustering. Underlines indicate statistical significance in a row, and bold indicates CVIC values greater than silhouette, fuzzy, or isolation and vice versa.

finding correct points. In this case we detect points by defining the threshold as one standard deviation below the mean.

Table 4.3 shows the average f-score over five runs for each method on each data set using neural gas clustering, and similar tables for the other clustering algorithms can be found in the appendix. The formatting is the same as Table 4.2 from the previous section.

Table 4.3 highlights the difficulty in detecting misclustered points, and shows that the different methods perform well on different domains. In the case of detecting misclustered points CVIC is well suited for high dimensional tasks with many clusters, particularly the ones using HOG features. Interestingly, the data sets which used word-spotting features had much lower recall which in turn lowered the f-score. This seems to be due to the fact that the features are not as discriminating as HOG and the models tended to have higher confidences

in the instances, so relatively few points were selected. Silhouette and isolation perform well on the other tasks, and again this is more due to having higher recall (as opposed to precision) than CVIC. However, median and average are important measures as well since in a typical clustering task the class distributions, data set complexity, and appropriate clustering algorithm are not often known. Therefore, with no specific knowledge of the data set to be validated a choice between silhouette, isolation, fuzzy, or CVIC can in part be determined by the average or median across several data sets and domains. In Table 4.3 CVIC-Ensemble has the best value for each.

Table 4.4 summarizes the results for detecting correctly clustered and misclustered points. We rank the methods for each data set on all five clustering algorithms. For detecting low confidence points the ensemble method and random forest perform best, and for high confidence points the support vector machine performs best. If, however, we consider the application of CVIC to include detecting high and low confidence points at the same time the final section of the table suggests that for the more spherical based clustering algorithms such as k-means and neural gas that an ensemble of the learners performs best where as with spectral and birch clustering knn or isolation perform best. However, it is easy to make the argument that to detect low confidence points a CVIC-RForest could be run, and to detect high confidence points an CVIC-SVM could be run.

Although the f-score is a useful measure, another informative indicator for this type of comparison study is the number of *unique* points found by each scoring method. Figures 4.1 and 4.2 graph the average number of unique points found by each method. This is defined as misclustered or correctly clustered points that were found only by a single method. When considering uniqueness we compare a single CVIC implementation to just the silhouette, isolation and fuzzy methods. Again this is done in an attempt to demonstrate the potential of the framework and that different supervised learners can be plugged-in. It is also done to try and find a single learner thay may be the best "out of the box" implementation of CVIC.

Low	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
KMEANS	4.65	5.90	4.15	5.10	5.35	7.05	3.80	5.75	**3.25**
GAS	4.20	6.70	4.85	4.55	5.60	6.70	4.25	4.95	**3.25**
GROWING	3.90	4.55	5.05	4.05	5.55	7.40	4.65	7.00	**3.20**
SPECTRAL	3.80	6.40	4.65	6.05	4.75	6.35	**3.50**	4.65	5.00
BIRCH	4.55	6.75	4.50	4.40	5.56	6.30	**4.15**	4.90	4.30
High									
KMEANS	7.85	7.75	4.90	3.95	**3.75**	3.85	4.55	4.40	4.05
GAS	7.95	7.85	4.80	4.45	**3.15**	3.30	5.05	4.05	4.40
GROWING	6.65	7.35	4.50	4.70	3.85	**3.45**	5.40	4.40	4.80
SPECTRAL	8.35	7.45	4.20	4.80	**3.15**	3.40	4.65	3.20	5.80
BIRCH	7.05	6.60	4.15	5.60	**3.45**	3.55	4.65	4.20	5.95
Both									
KMEANS	6.25	6.82	4.52	4.52	4.55	5.45	4.17	5.07	**3.65**
GAS	6.07	7.27	4.82	4.50	4.37	5.00	4.65	4.50	**3.82**
GROWING	5.27	5.95	4.77	4.37	4.70	5.42	5.02	5.70	**4.00**
SPECTRAL	6.07	6.95	4.42	5.42	3.95	4.87	4.07	**3.93**	5.40
BIRCH	5.80	6.67	**4.32**	5.00	4.50	4.92	4.40	4.55	5.12

Table 4.4: Average rank of f-score for low and high confidence points over each data set and clustering algorithm. Bold indicates best value in a row. The section titled "Both" shows the average of the low and high confidence ranks.

The data shown for silhouette, isolation and fuzzy is the average number of unique points found compared to each different CVIC implementation.

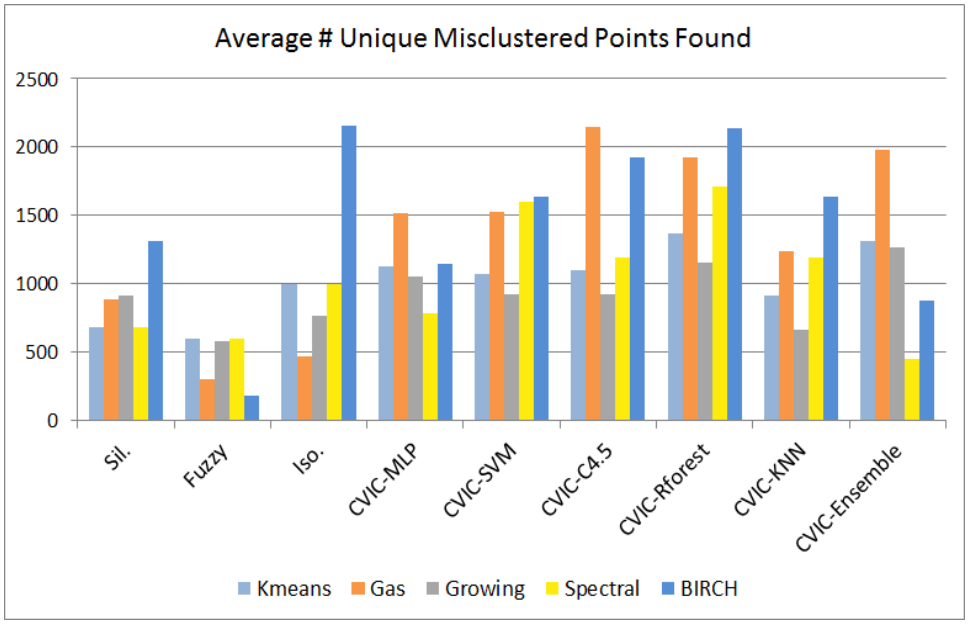


Figure 4.1: Average number of unique misclustered points found by each method.

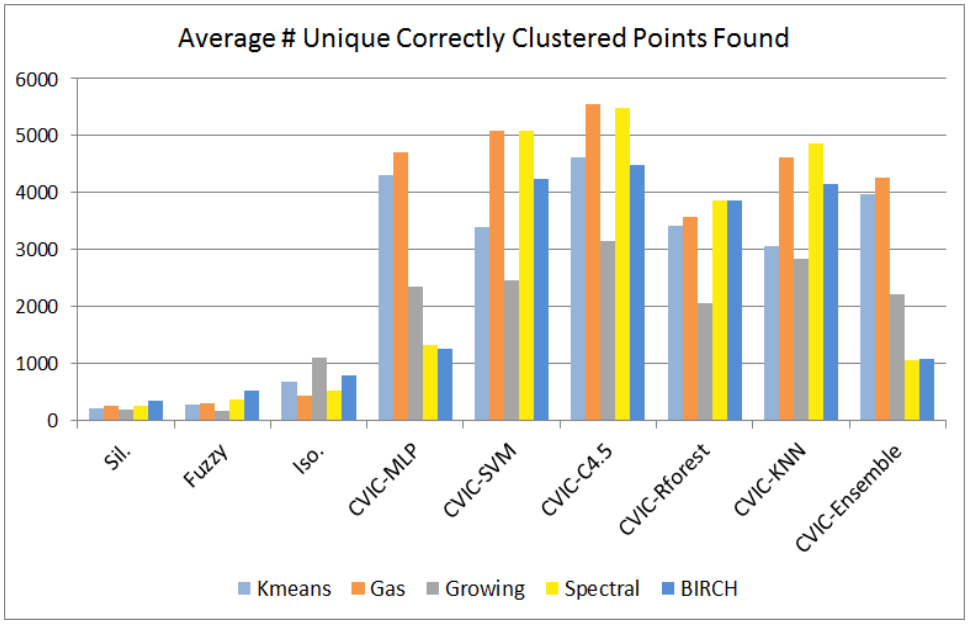


Figure 4.2: Average number of unique correctly clustered points found by each method.

This was done to reduce the number of comparisons shown, but accurately reflects the number of unique points found by the methods when compared to just a single CVIC implementation. Figures 4.1 and 4.2 demonstrate the distinct properties of CVIC. Nearly every implementation of CVIC finds more unique misclustered points than the distance based measures across all clustering algorithms. In the case of finding correctly clustered points, each CVIC method finds more unique points. The ability to find unique points supports the notion that non-distance based approaches have value and focusing on the question of what it means to belong to a cluster as seen by a supervised learner provides a distinct approach to discovering points that may or may not belong to a given cluster. We discuss these findings further in section 5.

When considering individual clustering algorithms and data sets, our results indicate that with some domain knowledge an end-user may select CVIC, silhouette or isolation to obtain good instance-based confidences. However, in clustering tasks, the class distributions, number of clusters, or difficulty in clustering may not be known ahead of time. This study shows that on average CVIC performs well and is therefore a viable choice for confidence

scores. Our results also suggest that a combination of the distance based techniques and CVIC would yield good results. We have performed a few preliminary tests of different combinations and found that although precision or recall could be raised it is difficult to raise both simultaneously. Future work will consider the difficult problem of successfully combining the distance based and supervised methods.

4.5 Experiment 3 - Choosing K

Lastly, we show results for determining the correct k value for the 12 synthetic data sets. We use Eq. 1 with our confidence scores to produce an index which when minimized suggests the correct k. Each dataset is clustered several times with a different k value each time ranging from 2 to 9. We compare with several indices from the literature (Davies-Bouldin (DB), XieBeni, I, SV, Silhouette, and Dunn's) that follow the same pattern, namely run several clusterings and calculate the index, then select the k which corresponds to the maximum or minimum index. As shown in Table 4.3 CVIC competes well with other indices, but silhouette does the best getting 11 out of 12 k values correct.

Index	**DB**	**XieBeni**	***I***	**SV**	**Silhouette**	**Dunn**	**CVIC**
# Correct	**1**	**9**	**6**	**6**	**11**	**7**	**9**

Figure 4.3: Table of number of data sets out of 12 in which the correct k was chosen

This experiment, while not the main focus of our work, shows the flexibility of CVIC and provides a solution to the question of how many clusters exist in a data set–a common question to answer when considering cluster validity. Again, we emphasize that using supervised learners provides a viable alternative to cluster validation and merits further study.

4.6 Experiment 4 - Run-Time Complexity

We provide a brief complexity analysis of the various methods. Silhouette runs in $O(n^2)$ time because each point uses the distances to all other points. Similarly, the isolation index performs a nearest neighbor search which naively is $O(n^2)$, but might be improved to $O(n\ logn)$ time with the use of a binary structure such as a k-d tree or cover tree. However it has been shown that in high dimensional spaces these structures do not provide any gains [16]. The fuzzy membership calculation is simply based on centroids requiring $O(kn)$ and is the fastest algorithm. The neural network is $O(in)$ where i is the number of iterations until training stops, but it also has large constants tied to the complexity of the net. SVMs involve solving a quadratic optimization that takes between $O(n^2)$ and $O(n^3)$ time for each iteration depending on the soft margin parameter [5]. The C4.5 tree runs in $O(nm^2)$ where m is the number of features [38]. A random forest trains each tree on a subset of the features and data and each tree therefore runs in $O(nm)$ where m is the square root of the number of features. K-nearest neighbors has the same analysis as isolation. Overall the CVIC framework has run times as follows. The human labelling portion requires k labelling queries where k is the number of clusters and is much smaller than n. These queries are easily and efficiently answered. The calculation of the threshold requires calculating the standard deviation for each cluster which is $O(km)$ where k is number of clusters and m is the number of points in a given cluster. Lastly, selecting points which are high and low confidence takes $O(n)$ time. Therefore the complexity of CVIC is only limited by its supervised learner and as such the runtimes are better or equal to silhouette and isolation except in the case of the SVM.

Chapter 5

Discussion

Our results show that on the data sets tested CVIC on average assigns confidence scores that more accurately reflect both misclustered and correctly clustered points compared to the other methods. When comparing each of these methods it should be kept in mind the trade-off between detecting low confidence points and high confidence points. CVIC's ensemble has the most consistent results across each performance metric. Combining CVIC with other methods does come at the cost of longer run times and the difficulty of deciding how best to combine the methods. An obvious candidate for ensembling that was not explored in the results section is a combination of the distance based measures with the supervised learners in CVIC. We found, however, that this approach did not gain any advantages except on the LEAF and SEEDS data sets. This is partly due to the difficulty of robustly combining the scores, and partly due to the fact that simply taking the union of the methods results in poor precisions. This idea is further discussed in the future work section.

High confidence points seem to be easier to succesfully detect than low confidence points. This makes sense because given a clustering that is generally accurate, a liberal detection method, which accepts all points, will have high recall and a precision that is near the baseline clustering accuracy. The SVM seems to perform best here because of the maximum margin used by SVMs. The confidence score that comes from the SVM is influenced by the distance a point is to the margin and therefore the SVMs scores tend to be higher for more points than they would be for another supervised model. In the case of finding correctly

clustered points this is favorable because the initial clustering is likely to have a fair number of the points already correctly clustered and the SVM being more liberal with its scores is favorable. However, for low confidence points there are multiple reasons for misclustering and it is harder to detect all of the misclustered points because several different approaches may be necessary. This is one argument for the success of the ensemble or random forest in detecting misclustered points since they consider several different biases. By recasting the problem from unsupervised validation to a supervised classification problem we have introduced a different inductive bias to the problem. Viewed another way, we have essentially created a non-distance based method, where we look at both output from a clustering algorithm *and* a supervised learner (or ensemble of learners) to validate an individual instance. The learner serves to either reinforce the given clustering, or to point to instances that should be reconsidered. We have shown this technique to be robust to the selection of data set or clustering algorithm.

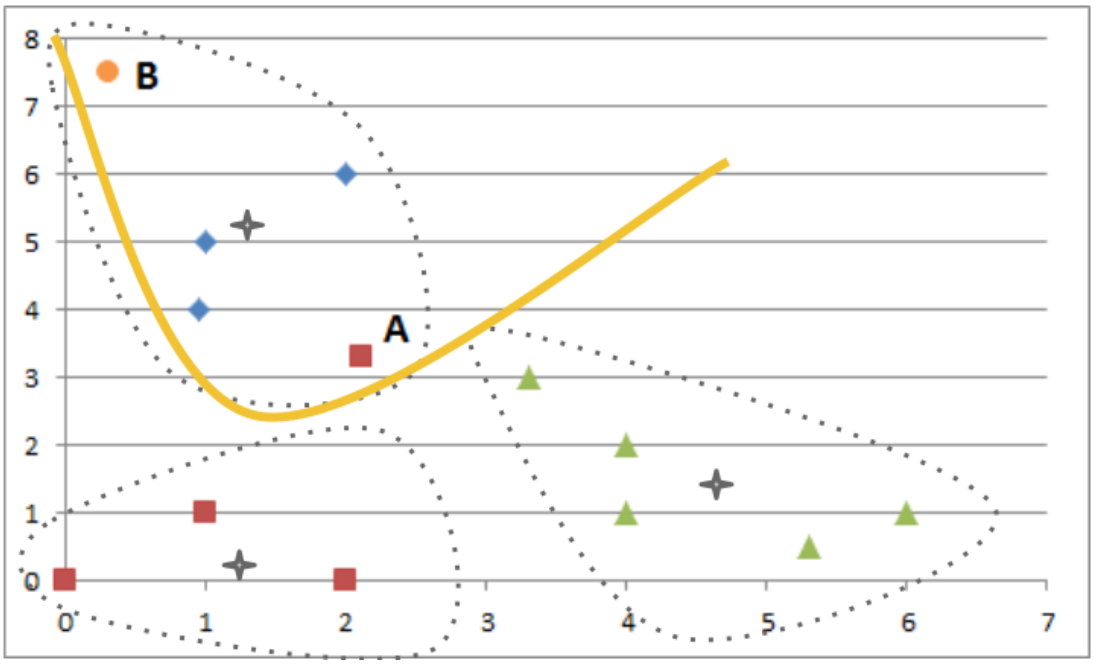


Figure 5.1: Example of 3 clusters, dotted black lines denote clusters, stars represent centroids, thick yellow line represents the decision boundary.

Figure 5.1 helps to illustrate the gains CVIC provides for finding unique points. In the example there are three distict clusters, but points A and B have been misclustered. Point A could be considered an outlier and atypical of its class and point B is a singleton. All the methods can easily find point A and label it as low-confidence, but only CVIC is able to find point B as well. This occurs because silhouette and fuzzy are concerned with distances to other points and the idea that points are lower confidence if they are near borders to other clusters. Isolation follows the same philosophy, point A's nearest neighbors are not all the same class, but point B's nearest neighbors are. If we draw hypothetical decision surfaces for a supervised learner we can see that point B has lower confidence because it is less typical of a point that belongs to said cluster. Though this example is admittedly contrived due to its simplistic and two-dimensional nature, it illustrates the point that the cluster, distance, and centroid focused framework for solving validation problems used by silhouette and others may be insufficient in some cases, and that at higher dimensions this problem is exacerbated. Therefore the perspective offered by a supervised learner can add significant new information to these types of problems. One negative aspect of CVIC is the fact that we are using supervised models such as multilayer perceptrons, support vector machines, and decision trees and as such the model is tied to hyperparameter selection. We performed trial-and-error analysis to determine good hyperparameter values, e.g., learning rate, stopping criteria, weight decay, and dropout rates for our MLP and other learners as outlined in section 3.2. This type of hyperparameter optimization is an ongoing area of research and in the case where trial-and-error analysis is not an affordable approach, reasonable heuristics must be used.

Chapter 6

Conclusion and Future Work

We have developed a method of cluster validation that departs from more traditional techniques and employs supervised learning to recast the problem into a nonlinear feature space which addresses the problem in a way that incorporates different biases than those of traditional distance metric based methods. We have shown on the data sets tested that on average CVIC performs better than other distance based methods for assigning robust confidence scores to clustered instances and that a simple combination of learners can provide even more accuracy.

Silhouette, isolation, fuzzy membership, and CVIC all produce scores which can be used to confidently detect points which are either misclustered or correctly clustered. However, if one simply looks at the numbers assigned, in most cases they are unitless. For example, a point may receive a score of .6 from any of the methods above, and this does not mean that we are 60% confident in the point or that there is a 60% chance it was clustered correctly. Looking at the score relative to other points in the cluster does give us an idea that there is low or high probability of being correct, but a way to actually map scores to an intuitive notion of confidence or probability is an area for future work. Our method of labelling points some number of standard deviations below the mean "low-confidence" is sufficient to find a fair amount of misclustered points, but does not really give the end-user a simple way to ask for all points that are less than 80% confident. Simply normalizing the scores or trying to fit some distribution to them seems to be insufficient to solve this problem.

Another potential area for future work is in more complex supervised learners such as deep networks or ensembles. Some preliminary work with ensembles of MLPs was done, but the gains wre insignificant and the additional run time costs were severe. The only requirement for a supervised learner to be used in CVIC is that a confidence score for prediction must be extractable.

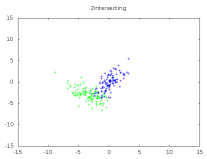
Finally, work with ensembling the various scoring methods could be further explored. This option has appeal due to the success of the current ensemble method. However, determining the best way to combine the scores is yet unknown. For example, silhouette scores range between -1 and 1 whereas CVIC scores are tied to the normalization of the network's activation layer, which in our case ends up being between 0 and 1. Although further normalization could be done to appropriately combine scores, they still would not reflect a true confidence. For example, a point that receives a score of 0 does not necessarily indicate that there in no confidence in that point belonging to its cluster. Therefore, a more principled approach to assiging intuitive scores will lead to improvements when using combined methods.

Appendix A

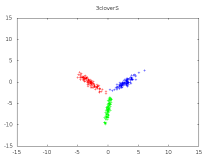
Supplemental Material

A.1 Synthetic Data sets

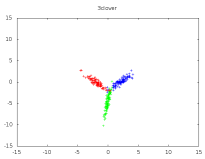
The following are images of the 12 synthetic data sets used for selection of k in experiment 3.



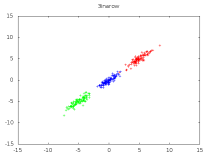
(a)



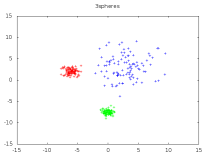
(b)



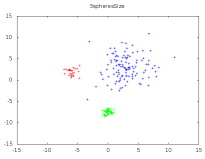
(c)



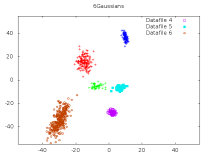
(d)



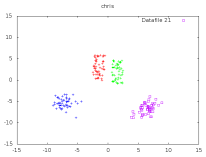
(e)



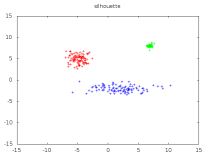
(f)



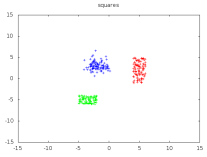
(g)



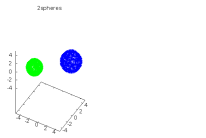
(h)



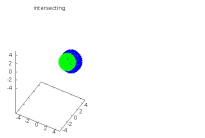
(i)



(j)



(k)



(l)

A.2 Tables for Various Clusterings

The following are tables for the remaining four clustering algorithms mentioned in experiments 1 and 2. Average f-scores for high and low confidence points. Numbers in bold and underlined indicate indicate statistical significance as described in section 4.3. Zeros indicate that no points were found for that particular threshold τ, but running at a less strict threshold did not significantly change outcomes.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.148	0.048	0.000	**0.263**	**0.187**	**0.377**	**0.324**	**0.320**	**0.360**
CVL	0.248	0.265	**0.375**	0.281	0.296	0.220	0.298	0.160	0.293
NV_DTW	0.225	0.254	**0.361**	0.058	0.029	0.026	0.179	0.012	0.096
NV_HOG	0.255	0.096	0.291	**0.324**	**0.325**	0.274	**0.312**	**0.315**	**0.335**
NB_DTW	0.211	0.236	**0.249**	0.025	0.120	0.002	0.039	0.002	0.037
NB_HOG	0.272	0.121	0.306	**0.348**	**0.361**	0.253	**0.342**	**0.363**	**0.385**
MO_DTW	0.190	0.206	**0.260**	0.037	0.169	0.002	0.046	0.000	0.048
MO_HOG	0.296	0.215	0.293	0.312	0.296	0.231	**0.349**	0.294	**0.374**
CAN_DTW	0.161	0.146	0.240	0.186	0.206	0.078	**0.284**	0.153	0.227
CAN_HOG	0.265	0.278	0.331	0.327	0.202	0.251	0.277	0.331	0.332
LEAF	0.333	0.006	0.384	0.259	0.304	0.286	0.346	0.383	0.350
COIL	**0.392**	0.036	0.330	0.173	0.236	0.294	0.351	0.311	0.359
UMIST	0.349	0.093	**0.403**	0.204	0.289	0.188	0.327	0.342	0.326
WALES	**0.302**	0.000	0.029	0.055	0.153	0.110	0.186	0.027	0.153
POET	**0.265**	0.133	0.203	0.132	0.109	0.018	0.120	0.054	0.121
USPS	0.278	0.105	0.380	0.300	0.319	0.223	**0.437**	**0.414**	**0.413**
PADEATHS	0.245	0.106	0.364	0.337	0.261	0.131	**0.450**	0.214	**0.403**
SEEDS	0.528	**0.539**	0.515	0.491	0.435	0.067	0.357	0.436	0.475
BANK	**0.333**	0.309	0.086	0.216	0.068	0.029	0.111	0.042	0.094
WASHPASS	0.362	0.348	0.309	0.335	0.136	0.367	0.379	**0.600**	0.377
MEDIAN	0.268	0.139	0.307	0.261	0.220	0.203	**0.318**	0.302	**0.333**
AVERAGE	0.282	0.176	**0.285**	0.233	0.225	0.171	0.275	0.238	0.277

Table A.1: Average f-score for low confidence points over five runs using kmeans.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.058	0.002	0.000	**0.252**	**0.188**	**0.413**	**0.316**	**0.267**	**0.375**
CVL	0.207	0.255	0.248	0.246	0.208	0.142	0.246	0.089	0.252
NV_DTW	0.267	**0.279**	0.114	0.054	0.052	0.006	0.087	0.002	0.076
NV_HOG	0.239	0.176	**0.389**	0.331	0.324	0.193	0.290	0.280	0.324
NB_DTW	0.201	**0.240**	0.062	0.042	0.120	0.004	0.031	0.004	0.051
NB_HOG	0.180	0.189	0.264	0.250	0.212	0.089	**0.309**	0.232	0.270
MO_DTW	0.206	0.214	**0.227**	0.066	0.128	0.002	0.062	0.000	0.079
MO_HOG	0.233	0.259	0.350	**0.381**	0.283	0.178	0.332	0.273	**0.401**
CAN_DTW	0.256	0.107	**0.345**	0.298	0.215	0.135	0.324	0.219	0.312
CAN_HOG	0.263	0.235	0.277	**0.325**	0.207	0.235	0.266	0.296	**0.331**
LEAF	**0.371**	0.000	0.073	0.258	0.230	0.000	0.000	0.069	0.245
COIL	**0.405**	0.358	0.347	0.301	0.244	0.321	0.307	0.242	0.356
UMIST	**0.322**	0.131	0.315	0.243	0.225	0.140	0.296	0.285	0.282
WALES	**0.298**	0.000	0.012	0.120	0.099	0.055	0.163	0.008	0.136
POET	**0.244**	0.235	0.031	0.159	0.050	0.006	0.067	0.012	0.124
USPS	**0.403**	0.348	0.296	0.324	0.285	0.143	0.377	0.338	0.363
PADEATHS	**0.364**	0.105	0.071	0.102	0.066	0.020	0.198	0.023	0.128
SEEDS	0.519	0.452	0.450	**0.544**	0.529	0.262	0.434	0.451	**0.542**
BANK	0.379	**0.414**	0.000	0.161	0.062	0.000	0.029	0.000	0.065
WASHPASS	0.192	**0.212**	0.000	0.059	0.028	0.025	0.071	0.000	0.068
MEDIAN	0.259	0.224	0.237	0.247	0.207	0.112	0.255	0.154	**0.261**
AVERAGE	**0.280**	0.210	0.193	0.225	0.187	0.118	0.210	0.154	0.239

Table A.2: Average f-score for low confidence points over five runs using growing gas.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.113	0.000	0.000	**0.236**	**0.183**	**0.406**	**0.328**	**0.328**	**0.343**
CVL	0.194	0.000	0.000	**0.235**	**0.358**	**0.297**	**0.307**	**0.266**	0.170
NV_DTW	0.250	0.173	0.000	0.024	0.087	0.062	0.252	0.100	0.006
NV_HOG	0.247	0.131	**0.468**	0.091	0.360	0.285	0.293	0.338	0.002
NB_DTW	0.188	0.114	0.000	0.056	0.070	0.031	**0.225**	0.108	0.026
NB_HOG	0.260	0.130	**0.416**	0.162	0.379	0.244	0.336	0.369	0.008
MO_DTW	0.181	0.124	**0.299**	0.020	0.069	0.027	0.240	0.139	0.006
MO_HOG	0.333	0.169	**0.387**	0.157	0.335	0.197	0.362	0.318	0.006
CAN_DTW	0.196	0.089	0.000	**0.357**	**0.256**	**0.226**	**0.326**	**0.322**	**0.368**
CAN_HOG	0.301	0.320	0.304	**0.359**	0.206	0.248	0.282	0.332	0.338
LEAF	0.387	0.000	**0.501**	0.299	0.353	0.314	0.383	0.439	0.390
COIL	0.335	0.107	0.324	0.284	0.236	**0.379**	**0.356**	0.305	**0.374**
UMIST	0.243	0.012	**0.346**	0.187	0.258	0.147	0.302	0.308	0.275
WALES	**0.301**	0.000	0.035	0.019	0.127	0.099	0.152	0.033	0.143
POET	**0.285**	0.199	0.230	0.223	0.222	0.039	0.183	0.108	0.211
USPS	0.314	0.000	0.434	0.322	0.337	0.253	0.446	0.450	0.434
PADEATHS	**0.356**	0.132	0.286	0.073	0.118	0.047	0.295	0.104	0.217
SEEDS	**0.471**	0.400	0.452	0.364	0.445	0.179	0.387	0.412	0.443
BANK	**0.264**	0.240	0.095	0.239	0.113	0.018	0.096	0.046	0.172
WASHPASS	0.197	**0.213**	0.000	0.020	0.029	0.010	0.058	0.000	0.051
MEDIAN	0.262	0.127	0.292	0.205	0.229	0.188	**0.298**	**0.306**	0.191
AVERAGE	0.270	0.127	0.228	0.186	0.227	0.175	**0.280**	0.240	0.199

Table A.3: Average f-score for low confidence points over five runs using spectral.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.149	0.000	0.000	**0.174**	0.162	**0.408**	**0.316**	**0.283**	0.124
CVL	0.224	0.000	**0.374**	0.149	0.308	0.321	0.294	0.238	0.071
NV_DTW	0.258	0.145	0.000	0.215	0.108	0.080	0.251	0.167	0.231
NV_HOG	0.236	0.144	0.308	**0.330**	0.316	0.273	0.304	0.312	**0.330**
NB_DTW	0.218	0.155	**0.311**	0.002	0.025	0.008	0.114	0.039	0.016
NB_HOG	0.244	0.180	**0.429**	0.033	0.358	0.236	0.349	0.342	0.002
MO_DTW	0.207	0.147	0.276	0.004	0.077	0.035	0.260	0.086	0.002
MO_HOG	0.252	0.162	0.291	**0.369**	0.303	0.217	**0.325**	**0.320**	**0.384**
CAN_DTW	0.215	0.109	0.000	**0.389**	0.222	0.178	**0.344**	**0.255**	**0.371**
CAN_HOG	0.271	0.317	0.355	**0.440**	0.302	0.283	0.341	**0.406**	**0.435**
LEAF	0.402	0.000	0.410	0.373	0.206	0.303	0.347	0.384	0.407
COIL	**0.408**	0.000	0.315	0.244	0.225	0.308	0.274	0.354	0.301
UMIST	0.337	0.000	**0.419**	0.308	0.307	0.188	0.344	0.347	0.336
WALES	**0.238**	0.000	0.016	0.115	0.121	0.092	0.141	0.014	0.130
POET	**0.307**	0.219	0.212	0.224	0.149	0.029	0.136	0.076	0.217
USPS	0.236	0.000	0.401	0.366	0.296	0.228	0.378	0.362	0.387
PADEATHS	0.155	0.047	0.195	0.107	0.132	0.033	0.208	0.056	0.136
SEEDS	**0.552**	0.492	0.490	0.511	0.397	0.158	0.425	0.364	0.495
BANK	0.238	0.271	0.059	**0.455**	0.000	0.000	**0.455**	**0.455**	**0.455**
WASHPASS	0.049	0.054	0.200	0.205	0.125	**0.395**	0.086	**0.250**	0.165
MEDIAN	0.237	0.126	0.299	0.234	0.213	0.202	**0.310**	0.297	0.265
AVERAGE	0.259	0.122	0.252	0.250	0.206	0.188	**0.284**	**0.255**	0.249

Table A.4: Average f-score for low confidence points over five runs using BIRCH.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.331	0.325	0.000	0.323	0.299	**0.503**	**0.373**	**0.579**	**0.387**
CVL	0.157	0.165	0.130	**0.193**	0.172	**0.193**	**0.190**	0.184	**0.192**
NV_DTW	0.234	0.277	0.333	**0.428**	**0.423**	**0.421**	**0.433**	0.335	**0.432**
NV_HOG	0.282	0.290	0.352	**0.534**	**0.537**	**0.488**	0.367	**0.551**	**0.492**
NB_DTW	0.255	0.255	0.462	**0.682**	0.386	**0.549**	**0.693**	0.125	**0.684**
NB_HOG	0.298	0.305	0.359	**0.762**	**0.770**	**0.752**	**0.460**	**0.778**	**0.741**
MO_DTW	0.289	0.294	0.531	**0.806**	0.405	**0.662**	**0.812**	0.008	**0.808**
MO_HOG	0.291	0.322	0.356	**0.844**	**0.842**	**0.839**	**0.772**	**0.846**	**0.821**
CAN_DTW	0.280	0.283	0.408	**0.703**	**0.696**	**0.679**	**0.615**	**0.691**	**0.673**
CAN_HOG	0.296	0.283	0.400	**0.848**	**0.816**	**0.839**	0.326	**0.789**	**0.736**
LEAF	0.465	0.340	**0.666**	0.451	0.411	0.561	0.545	0.616	0.553
COIL	0.140	0.273	**0.764**	0.726	0.572	0.660	0.434	0.706	0.532
UMIST	0.340	0.374	0.678	0.645	0.446	0.525	0.494	0.661	0.628
WALES	0.274	0.000	0.636	0.464	**0.657**	0.594	0.528	0.553	0.521
POET	0.217	0.199	0.355	0.294	0.368	0.349	0.368	0.352	0.356
USPS	0.475	0.462	0.704	**0.750**	**0.742**	**0.724**	0.681	**0.765**	0.715
PADEATHS	0.297	0.319	0.763	**0.872**	**0.872**	**0.856**	**0.801**	0.612	**0.845**
SEEDS	0.163	0.298	0.930	0.621	0.913	0.634	0.901	0.854	0.889
BANK	0.379	0.335	0.714	0.207	0.706	0.723	0.701	0.532	0.718
WASHPASS	0.080	0.228	0.839	0.824	0.805	0.757	0.781	0.732	0.781
MEDIAN	0.285	0.292	0.496	**0.663**	**0.614**	**0.646**	**0.536**	**0.614**	**0.678**
AVERAGE	0.277	0.281	0.519	**0.598**	**0.591**	**0.615**	**0.563**	**0.563**	**0.624**

Table A.5: Average f-score for high confidence points over five runs using kmeans.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.352	0.315	0.000	0.310	0.266	**0.505**	0.340	**0.567**	0.362
CVL	0.154	0.152	0.103	0.118	0.141	0.172	0.113	0.159	0.119
NV_DTW	0.240	0.253	0.410	0.424	0.425	0.416	0.421	0.087	0.425
NV_HOG	0.231	0.240	0.423	0.441	**0.480**	**0.472**	0.329	**0.496**	0.397
NB_DTW	0.266	0.252	0.688	0.690	0.403	0.695	0.692	0.695	0.688
NB_HOG	0.247	0.297	0.465	**0.729**	**0.729**	**0.700**	**0.689**	**0.706**	**0.723**
MO_DTW	0.258	0.283	0.748	**0.809**	0.662	**0.796**	**0.812**	**0.782**	**0.810**
MO_HOG	0.296	0.324	0.407	**0.837**	**0.838**	**0.840**	**0.787**	**0.858**	**0.831**
CAN_DTW	0.274	0.291	0.416	**0.664**	**0.735**	**0.723**	**0.573**	**0.708**	**0.648**
CAN_HOG	0.267	0.284	0.646	**0.763**	**0.820**	**0.842**	0.297	**0.832**	**0.673**
LEAF	0.483	0.420	**0.666**	0.355	0.586	0.000	0.000	0.369	0.357
COIL	0.448	0.262	**0.726**	0.426	0.421	0.595	0.364	0.538	0.348
UMIST	0.350	0.360	0.653	0.597	0.377	0.593	0.472	0.634	0.562
WALES	0.467	0.000	**0.518**	0.262	0.406	0.473	0.383	0.312	0.291
POET	0.172	0.199	0.362	0.350	0.360	0.356	0.361	0.374	0.358
USPS	0.500	0.400	0.713	0.726	0.721	0.725	0.608	0.718	0.719
PADEATHS	0.229	0.303	0.896	0.852	0.881	0.887	0.819	0.632	0.833
SEEDS	0.542	0.321	0.945	0.741	0.922	0.934	0.873	0.863	0.867
BANK	0.748	0.200	0.732	0.753	0.744	0.000	0.727	0.000	0.743
WASHPASS	0.012	0.210	0.701	0.670	0.635	0.697	0.684	0.000	0.664
MEDIAN	0.270	0.283	0.649	**0.667**	0.610	0.645	0.522	0.599	**0.655**
AVERAGE	0.326	0.268	0.560	**0.575**	**0.577**	**0.571**	0.517	0.516	**0.570**

Table A.6: Average f-score for high confidence points over five runs using growing gas.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.312	0.356	0.668	0.653	0.401	0.560	0.467	0.655	0.607
CVL	0.124	0.102	0.116	0.134	0.143	0.138	0.142	0.142	0.132
NV_DTW	0.210	0.276	0.238	0.263	**0.429**	**0.411**	**0.443**	**0.399**	0.245
NV_HOG	0.250	0.285	0.347	**0.382**	**0.555**	**0.497**	0.317	**0.575**	0.310
NB_DTW	0.266	0.252	0.688	0.690	0.403	0.695	0.692	0.695	0.688
NB_HOG	0.247	0.297	0.465	**0.729**	**0.729**	**0.700**	**0.689**	**0.706**	**0.723**
MO_DTW	0.231	0.290	0.279	0.246	**0.817**	**0.779**	**0.765**	**0.798**	0.199
MO_HOG	0.257	0.308	0.333	0.305	**0.844**	**0.836**	**0.765**	**0.852**	0.218
CAN_DTW	0.250	0.285	0.292	**0.717**	**0.767**	**0.781**	**0.365**	**0.726**	**0.515**
CAN_HOG	0.281	0.284	0.381	**0.844**	**0.811**	**0.851**	0.284	**0.807**	**0.732**
LEAF	0.501	0.352	0.657	0.414	0.343	0.621	0.544	0.643	0.582
COIL	0.250	0.296	0.437	**0.469**	0.380	0.421	0.271	**0.485**	0.355
UMIST	0.312	0.356	0.668	0.653	0.401	0.560	0.467	0.655	0.607
WALES	0.408	0.000	**0.624**	0.416	0.597	0.571	0.516	0.518	0.550
POET	0.231	0.196	0.353	0.273	0.362	0.343	0.358	0.353	0.323
USPS	0.289	0.368	0.788	0.796	0.795	0.791	0.757	**0.831**	0.790
PADEATHS	0.250	0.310	0.751	**0.837**	**0.846**	**0.792**	**0.799**	0.500	**0.815**
SEEDS	0.051	0.294	0.919	0.712	0.904	0.833	0.851	0.931	0.843
BANK	0.391	0.336	0.762	0.747	0.768	0.766	0.731	0.602	0.742
WASHPASS	0.016	0.231	0.701	0.695	0.665	0.695	0.682	0.000	0.677
MEDIAN	0.250	0.291	0.409	**0.442**	**0.630**	**0.657**	**0.528**	**0.621**	**0.532**
AVERAGE	0.256	0.274	0.495	**0.501**	**0.605**	**0.625**	**0.532**	**0.586**	0.485

Table A.7: Average f-score for high confidence points over five runs using spectral.

	Silhouette	Fuzzy	Isolation	CVIC-MLP	CVIC-SVM	CVIC-C4.5	CVIC-RForest	CVIC-KNN	CVIC-Ensemble
IAM	0.385	0.391	0.000	0.343	0.312	**0.424**	0.288	**0.537**	0.346
CVL	0.172	0.167	0.131	0.144	**0.200**	0.183	0.182	0.168	0.139
NV_DTW	0.197	0.260	0.225	**0.387**	**0.418**	**0.420**	**0.402**	**0.396**	**0.394**
NV_HOG	0.242	0.271	0.334	**0.425**	**0.488**	**0.467**	0.302	**0.522**	**0.416**
NB_DTW	0.189	0.253	0.243	0.193	**0.695**	**0.564**	**0.686**	**0.605**	0.164
NB_HOG	0.261	0.300	0.368	0.241	**0.758**	**0.747**	**0.645**	**0.775**	0.217
MO_DTW	0.225	0.295	0.307	0.204	0.074	0.031	**0.524**	0.004	0.152
MO_HOG	0.250	0.324	0.345	0.220	**0.834**	**0.832**	**0.763**	**0.842**	0.164
CAN_DTW	0.249	0.285	0.263	**0.523**	**0.713**	**0.717**	**0.417**	**0.652**	**0.568**
CAN_HOG	0.255	0.298	0.414	**0.756**	**0.806**	**0.832**	0.331	**0.804**	**0.680**
LEAF	0.482	0.394	**0.684**	0.533	0.339	0.536	0.565	0.584	0.555
COIL	0.417	0.302	**0.818**	0.428	0.465	0.568	0.346	0.498	0.291
UMIST	0.399	0.396	0.678	0.666	0.445	0.586	0.495	**0.712**	0.647
WALES	0.462	0.000	**0.656**	0.512	0.595	0.625	0.486	0.411	0.484
POET	0.184	0.203	0.368	0.337	0.361	0.340	0.368	0.311	0.349
USPS	0.256	0.343	0.802	0.737	0.770	0.788	0.661	0.780	0.713
PADEATHS	0.250	0.295	0.888	0.869	0.891	0.895	0.816	0.625	0.852
SEEDS	0.823	0.324	**0.930**	0.767	0.885	0.808	0.873	0.879	0.862
BANK	0.203	0.320	**0.715**	0.442	0.000	0.000	0.442	0.442	0.441
WASHPASS	0.170	0.224	**0.998**	0.902	0.936	0.886	0.850	0.666	0.891
MEDIAN	0.250	0.296	0.390	**0.435**	**0.541**	**0.577**	**0.490**	**0.594**	**0.428**
AVERAGE	0.303	0.282	0.508	0.481	**0.549**	**0.562**	**0.522**	**0.560**	0.466

Table A.8: Average f-score for high confidence points over five runs using BIRCH.

References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] Richard Bellman, Richard Ernest Bellman, Richard Ernest Bellman, and Richard Ernest Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[3] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *Database TheoryICDT99*, pages 217–235. Springer, 1999.

[4] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[5] Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, pages 301–320, 2007.

[6] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[7] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[8] Małgorzata Charytanowicz, Jerzy Niewczas, Piotr Kulczycki, Piotr A Kowalski, Szymon Łukasik, and Sławomir Żak. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*, pages 15–24. Springer, 2010.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] David L Davies and Donald W Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.

[11] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[12] Bernd Fritzke et al. A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7:625–632, 1995.

[13] Stefan Glock, Eugen Gillich, Johannes Schaede, and Volker Lohweg. Feature extraction algorithm for banknote textures based on incomplete shift invariant wavelet packet transform. In *Pattern Recognition*, pages 422–431. Springer, 2009.

[14] Daniel B Graham and Nigel M Allinson. Characterising virtual eigensignatures for general purpose face recognition. In *Face Recognition*, pages 446–456. Springer, 1998.

[15] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment using multi representatives. In *Proceedings of the Hellenic Conference on Artificial Intelligence, SETN*, pages 237–249, 2002.

[16] P. Indyk. Nearest neighbors in high-dimemsional spaces. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 2004.

[17] Young-Il Kim, Dae-Won Kim, Doheon Lee, and Kwang H Lee. A cluster validation index for gk cluster analysis based on relative degree of sharing. *Information Sciences*, 168(1):225–242, 2004.

[18] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 560–564. IEEE, 2013.

[19] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[20] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[21] U-V Marti and Horst Bunke. A full english sentence database for off-line handwriting recognition. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 705–708. IEEE, 1999.

[22] Thomas Martinetz, Klaus Schulten, et al. *A" neural-gas" network learns topologies*. University of Illinois at Urbana-Champaign, 1991.

[23] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1650–1654, 2002.

[24] Marina Meilpa and Jianbo Shi. Learning segmentation by random walks. In *Neural Information Processing Systems*, 2001.

[25] S Nayar, Sammeer A Nene, and Hiroshi Murase. Columbia object image library (coil 100). *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*, 1996.

[26] Dim P. Papadopoulos, Alasdair D. F. Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 361–376, 2014.

[27] Eric J Pauwels and Greet Frederix. Finding salient regions in images: nonparametric clustering for image segmentation and grouping. *Computer Vision and Image Understanding*, 75(1):73–85, 1999.

[28] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[29] Haiyan Qiao and Brandon Edwards. A data clustering tool with cluster validity indices. In *Computing, Engineering and Information, 2009. ICC'09. International Conference on*, pages 303–309. IEEE, 2009.

[30] J Ross Quinlan. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 1993.

[31] Tony M Rath and Rudrapatna Manmatha. Word spotting for historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):139–152, 2007.

[32] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[33] K Pramod Sankar, R Manmatha, and CV Jawahar. Large scale document image retrieval by automatic word annotation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(1):1–17, 2014.

[34] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[35] Pedro FB Silva, André RS Marçal, and Rubim M Almeida da Silva. Evaluation of features for leaf discrimination. In *Image Analysis and Recognition*, pages 197–204. Springer, 2013.

[36] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. An instance level analysis of data complexity. *Machine learning*, 95(2):225–256, 2014.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[38] Jiang Su and Harry Zhang. A fast decision tree learning algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 500. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[39] Chris Tensmeyer and Tony Martinez. Confirm - clustering of noisy form images using robust metrics. In *15th Workshop on Family History Technology*, Provo, UT, USA, 2015.

[40] Michel Verleysen, Damien Francois, Geoffroy Simon, and Vincent Wertz. On the effects of dimensionality on data analysis with neural networks. In *Artificial Neural Nets Problem solving methods*, pages 105–112. Springer, 2003.

[41] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[42] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 13(8):841–847, 1991.

[43] Krista Rizman Žalik and Borut Žalik. Validity index for clusters of different sizes and densities. *Pattern Recognition Letters*, 32(2):221–234, 2011.

[44] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM, 1996.