machine learning & knowledge extraction

MDPI

Article

Assessing the Robustness of Cluster Solutions in Emotionally-Annotated Pictures Using Monte-Carlo Simulation Stabilized K-Means Algorithm

Marko Horvat [1,*], Alan Jović [2] and Kristijan Burnik [3]

[1] Department of Applied Computing, Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia
[2] Department of Electronics, Microelectronics, Computer and Intelligent Systems, Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia; alan.jovic@fer.hr
[3] Independent Researcher, HR-10000 Zagreb, Croatia; kristijan.burnik@gmail.com
* Correspondence: marko.horvat3@fer.hr

Abstract: Clustering is a very popular machine-learning technique that is often used in data exploration of continuous variables. In general, there are two problems commonly encountered in clustering: (1) the selection of the optimal number of clusters, and (2) the undecidability of the affiliation of border data points to neighboring clusters. We address both problems and describe how to solve them in application to affective multimedia databases. In the experiment, we used the unsupervised learning algorithm k-means and the Nencki Affective Picture System (NAPS) dataset, which contains 1356 semantically and emotionally annotated pictures. The optimal number of centroids was estimated, using the empirical elbow and silhouette rules, and validated using the Monte-Carlo simulation approach. Clustering with k = 1–50 centroids is reported, along with dominant picture keywords and descriptive statistical parameters. Affective multimedia databases, such as the NAPS, have been specifically designed for emotion and attention experiments. By estimating the optimal cluster solutions, it was possible to gain deeper insight into affective features of visual stimuli. Finally, a custom software application was developed for study in the Python programming language. The tool uses the scikit-learn library for the implementation of machine-learning algorithms, data exploration and visualization. The tool is freely available for scientific and non-commercial purposes.

Keywords: multimedia; clustering; k-means; Monte-Carlo simulation; cluster distribution; emotion; affective computing

Citation: Horvat, M.; Jović, A.; Burnik, K. Assessing the Robustness of Cluster Solutions in Emotionally-Annotated Pictures Using Monte-Carlo Simulation Stabilized K-Means Algorithm. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 435–452. https://doi.org/10.3390/make3020022

Academic Editor: Andreas Holzinger

Received: 31 March 2021
Accepted: 1 May 2021
Published: 4 May 2021

1. Introduction

Clustering can be broadly described as the task of dividing the population, or data points, or observations, as they are also called, into a number of groups such that data points in the same groups, given a chosen set of attributes and metrics to compare them, are more similar to other data points in the same group than to those in other groups or clusters [1]. Clustering is an unsupervised process, which means that we are given unlabeled data and we need to put similar samples in one group (called cluster) and dissimilar samples in another, different cluster. Each cluster has maximum within-cluster similarity and minimum between-cluster similarity based on a certain similarity measure [2]. The aim of any clustering procedure is to identify groups of observations with at least one similar trait or attribute and assign them to different clusters. The similarity measure between observations is defined by some inter-observation distance measures or correlation-based distance measures. Simply put, clustering is an unsupervised machine-learning technique that divides the original dataset into disjoint subsets based on the relatedness between the members of the subsets. It can be applied to any dimension of data, be it one-dimensional data points or multifaceted individuals with many different features. When applying

clustering in practice, one often encounters several problems: (1) the selection of the cluster similarity measure, (2) the selection of the optimal number of clusters, (3) the undecidability of the affiliation of border data points to neighboring clusters, and (4) the lack of correct group labels, which limits the applicability of the clustering model (unclear interpretation) [3,4].

Evaluating the robustness of clustering solutions is a very important and common task in data exploration. Many clustering-evaluation measures have been proposed in the published literature [1–7]. Clustering validation is a concept used to describe the process of testing the performance of a clustering algorithm. Classes or categories of clustering algorithms are centroid-based (e.g., k-means and m-medoids), distribution-based (e.g., expectation–maximization algorithm), hierarchical (e.g., agglomerative) and distribution-based (e.g., DBSCAN) [8,9]. Importantly, for a particular clustering algorithm, such as k-means, k-medoids and the expectation–maximization algorithm, there is a parameter known as k that specifies the number of clusters to be detected [10]. In some other clustering algorithms, it is not necessary to specify the parameter k in advance, and hierarchical clustering methods avoid this problem completely.

A novel affective multimedia dataset was selected to test efficient and essential clustering methods. The descriptors of multimedia documents usually refer to the file name, type, size, time of creation and last modification, owner, access privileges, etc. In addition, the semantics of multimedia documents may be described using informal and formal methods, such as keywords from unsupervised vocabularies and ontologies, respectively. However, multimedia can also be used to efficiently elicit a wide range of emotional responses. To this end, the set of document descriptors is extended to include explicitly structured information about the expected emotional response upon exposure to the given multimedia content. Affective multimedia databases are standardized digital repositories that store auditory, linguistic, and visual materials for emotion research. The affective multimedia dataset used for the experiments in this work is described in the next section.

The aim of the research conducted was to investigate the clustering of affective multimedia and the relationship between semantics and emotions. The interaction of semantics and emotions is important because their relationship is not random but depends on personal knowledge, personal experience, cultural conditioning, collective memory, and other similar factors that cumulatively direct the articulation of particular semantics toward a particular set of emotions. The difference in the articulation of images by pixel-defined content and semantic content is referred to as the semantic gap [11]. This coupling of semantics and affect in emotionally annotated multimedia documents can be defined as a deterministic interaction between the semantics of a document and the effect that its semantics evoke. In this regard, the practical goal of the presented research is to develop an intelligent system that can infer the emotional content of a multimedia document from the evaluation of its semantics, and conversely, estimate the dominant semantics from the affective annotations when such information is available. The system could potentially have many useful applications, such as supported construction of affective multimedia databases, video recommendation, or emotion estimation. The reported results on emotion clustering are essential for the success of such a future system.

The remainder of the paper is organized as follows; the next section brings forward emotion models used in affective multimedia databases, in particular the dimensional model, which is utilized in the presented research. Properties of the NAPS affective multimedia database, as the best generic representative of affective multimedia databases, are also described in the next section. Related work is covered in Section 3 and differences between this research and other previously published studies are briefly explained. After that, in Section 4, unsupervised learning with the k-means algorithm is formally defined. Advantages and shortcomings of k-means are described. The evaluation of pictures clustering based on a dimensional emotion model using the Monte-Carlo simulation-stabilized k-means algorithm, with estimation of the optimal number of clusters, are presented in Section 5. Finally, the conclusion is presented in the final section at the end of the paper.

2. Affective Multimedia Databases

All multimedia documents, such as texts, pictures, videos and sounds, generate emotional responses in humans. This effect can be characterized by polarity and intensity. Some document content will provoke intense reactions, while others will produce no obvious feedback at all. Additionally, different individuals or homogeneous groups of individuals will show different responses, depending on the stimulation context. Multimedia documents specifically prepared for controlled stimulation of emotions in laboratory settings are stored in affective multimedia databases. They are often referred to as stimuli. In addition to the study of human emotion mechanisms, affective multimedia databases have many other practical applications in the study of perception, memory, attention, and reasoning [12]. Some of the experimental areas involved are cognitive science, psychology, neuroscience, and interdisciplinary studies, such as human–computer interaction.

2.1. Models of Affect in Affective Multimedia Databases

In contemporary affective multimedia databases, the structure of emotion is described with at least one of the two prevalent models of affect: discrete and dimensional [13]. These two models can both effectively describe emotion in digital systems but are not mutually exclusive. While most repositories use only one model, most frequently the dimensional one, some already incorporate both theories of emotion—for example, [14,15]. Having at the disposal annotations according to both theories is useful because they provide a more complete characterization of multimedia affect content.

The discrete or categorical model classifies emotions into specific labels. The number of these basic emotions is a point of contention. However, most researchers agree that six primary emotions (joy, sadness, surprise, anger, fear, disgust) are invariant among different cultures and universal [16].

The dimensional model, shown in Figure 1, is also often called the circumplex model of emotion, the Russell model of emotion, or the PAD (Pleasure–Arousal–Dominance) model [17]. This simple yet efficient model describes each emotion as a tuple $e_i = \{val,ar,dom\}$ where *val*, *ar*, *dom* are continuous variables representing valence, arousal and dominance emotional dimensions [17]. These three emotional dimensions form mutually orthogonal axes and their values are normalized in interval [1,9]. $val \in [1,9] \in Val$, $ar \in [1,9] \in Ar$, $dom \in [1,9] \in Dom$. Dominance (*dom*) is frequently omitted from the description of the emotion space because it was shown to be the least informative measure of the elicited affect [18]. Thus, following the dimensional model of affect and for all practical purposes, a single emotionally annotated multimedia document can be represented as a coordinate in a two-dimensional space of emotion $\Omega_{Emo} = Val \times Ar$. Russell estimated the approximate central coordinates of specific discrete emotions in the dimensional model's space [17]. He hypothesized that these locations are not fixed but rather change during a person's lifetime, and also differ from one person to another, or between homogenous groups of persons based on their character traits. An illustration of the circumplex model of emotion, as proposed in [17], is shown in Figure 1. Emotionally annotated pictures, listed in Table 1 and shown in Figure 2, are projected on the two-dimensional space of emotion Ω_{Emo} with each point representing one picture.

Table 1. An extract of the sub sample of the NAPS dataset annotations used in the experiment.

ID	Description	Valence (Avg)	Arousal (Avg)
Animals_002_v	lion	6.45	6.86
Animals_003_h	snake	5.02	5.51
Animals_004_v	wolf	4.54	7.10
Animals_005_h	bat	5.57	5.73
Faces_001_h	children with a dog	7.80	4.97
Faces_242_h	man and woman smiling	6.66	3.76

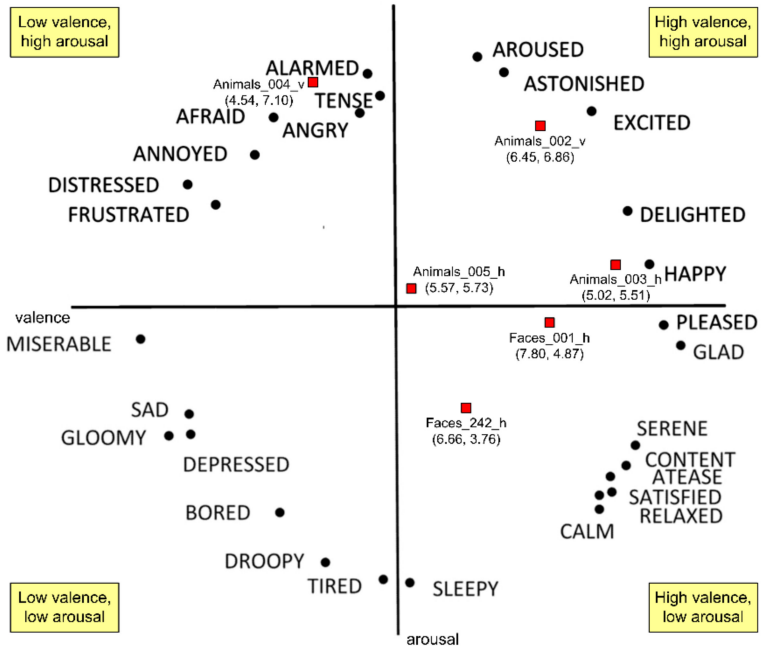

Figure 1. The circumplex model of emotion as described in [17]. Valence (*val*) represents *x*-axis and arousal (*ar*) *y*-axis. Red points mark pictures from the experimental dataset listed in Table 1. Approximate (*val*, *ar*) coordinates of basic emotions in the dimensional emotion model space Ω_{Emo} are indicated.

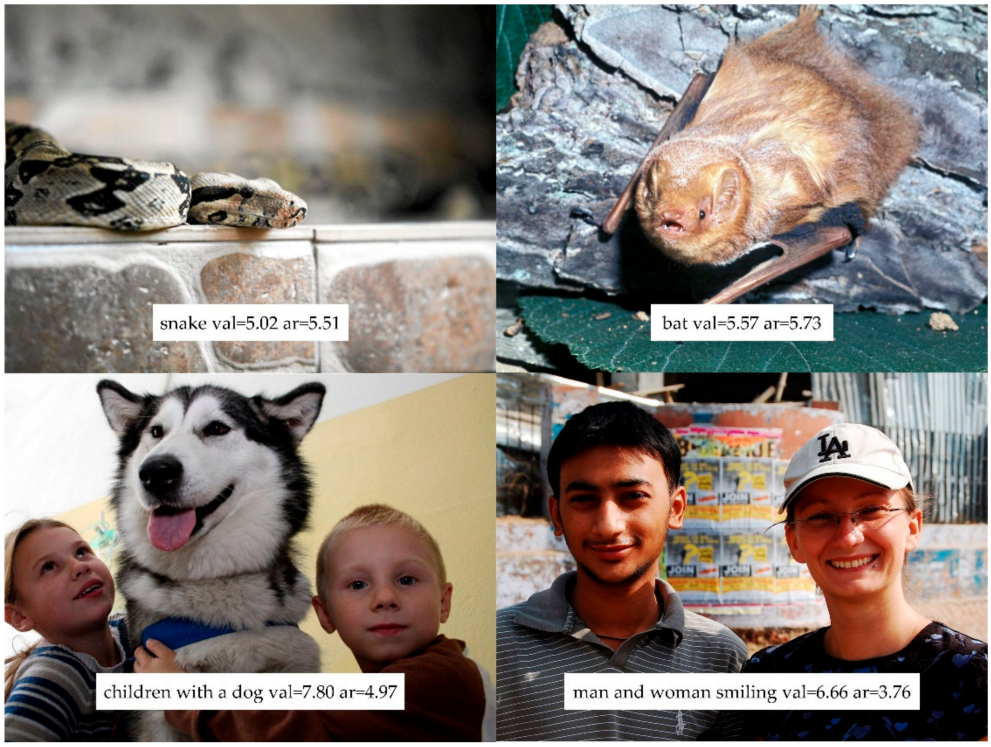

Figure 2. Example pictures from the NAPS dataset. Reproduced with permission from Marchewka, A.; Żurawski, Ł.; Jednorog, K.; Grabowska, A. The Nencki Affective Picture System (NAPS): Introduction to a novel, standardized, wide-range, high-quality, realistic picture database (2014), Springer.

2.2. The NAPS Affective Picture Database

Currently there are many databases indexing affective information in multimedia [13], but to the best of our knowledge, the Nencki Affective Picture System (NAPS) is currently

the largest database of visual stimuli with a comprehensive set of accompanying normative ratings and physical parameters of images (e.g., luminance) [14]. Furthermore, the NAPS has a typical architecture and file structure and employs data model generic to almost all affective picture databases [13]. It has also been relatively recently developed. As such, it is the best common representative of affective picture repositories using the dimensional model of emotion. Therefore, it was the optimal choice for the presented study.

The NAPS is the result of research conducted by the Polish Nencki Institute of Experimental Biology [14]. This database was constructed in response to limitations of existing databases, such as a limited number of stimuli in specific categories or poor picture quality of visual stimuli. The NAPS can be used in various areas of affective research and consists of 1356 realistic, high-quality photographs divided into five disjoint categories: people, faces, animals, objects, and landscapes. The photos were chosen to elicit a specific emotional response in the general population, i.e., not all photos are content-neutral. More recently, two extensions of the NAPS were developed. The first one is NAPS Basic Emotions (NAPS BE), which contains a subset of 510 images from the original NAPS [15] with normative ratings according to categorical models of emotions in addition to dimensional ratings. Additionally, an erotic subset for the Nencki Affective Picture System (NAPS ERO) was introduced with 200 stimuli pictures accompanied by self-report ratings of emotional valence and arousal by homosexual and heterosexual men and women (N = 80, divided into four equally sized subgroups) [19].

One of the main features of the NAPS compared to other emotion-elicitation databases is that it combines a large number of pictures together with normative ratings elicited in accordance with both dimensional and categorical (discrete) emotion theories [14], in addition to having additional multi-word and categorical semantic descriptions. By combining these features, it is possible to define faceted query patterns using different emotion and semantic dimensions. Examples of four pictures from the NAPS along with their semantic and emotional annotations are shown in Figure 2. These examples are also listed in Table 1.

The entire NAPS dataset with N = 1356 picture stimuli was used for the experiment. Each picture in the dataset is described with short, descriptive text (e.g., "dead animal", "bat", "wolf", "children with a dog", etc.) and two main emotional dimensions. The two dimensions are valence, which describes the attractiveness (positive valence) or aversiveness (negative valence) of stimuli along a continuum (negative–neutral–positive), and arousal, which refers to the perceived intensity of an event ranging from very calming to highly exciting or agitating [20]. Both emotional dimensions are described as numerical variables with a range of values from 1.0 to 9.0, where a lower value indicates a lower level of valence and arousal, and vice versa for a higher value. Examples of semantic and emotional picture annotations from the NAPS dataset are shown in Table 1.

3. Related Work

Thus far, the grouping of affective values in pictures and other formats has not been extensively investigated. In our previous work, we wanted to establish whether a statistically significant relationship between semantics and emotions in pictures is possible. The goal of this line of research would be to create a decision support system for the facilitated construction of affective multimedia databases. We envisioned a scenario where new documents could be added to stimuli repositories based only on sparse keyword annotations. Given a priori prepared stimuli clusters, it could be then possible to assign a range of affective values from stimuli in the cluster closest to the new document. In the continuation of this scenario, a domain expert would be asked to confirm the assigned values and correct them if necessary [21]. We also conducted a methodical survey among expert users of affective multimedia databases and confirmed that such a system is strongly needed in practice. Indeed, survey participants unequivocally expressed a necessity for an intelligent stimuli retrieval application that would assist them in experimentation. Almost all experts agreed that such applications would be useful in their work [22].

In the mentioned preliminary exploratory investigation of the NAPS dataset, we applied k-means algorithm on the two-dimensional (valence/arousal) model of affect [21]. Clustering with k = 1–94 centroids was reported, together with dominant picture keywords and descriptive statistical parameters. The optimal number of centroids was estimated using the minimum cumulative error rule. Results of this initial study were inconclusive. Using the elbow rule, we found that the optimal number of clusters is probably between 2 and 9. We noticed that cluster semantics began to homogenize with $k \sim 7$. Cluster stability was not investigated or verified. Nonetheless, we recommended that the exact value should be determined with further research into the coupling of semantics and emotion in the NAPS dataset. We expressed our belief that semantic distance between picture keywords in a cluster is an important indicator of the group's homogeneity.

In a third related study [23], researchers have also noticed that—despite their extensive use—retrieval of picture stimuli for a research project is strenuous and tailored to individual studies. Therefore, they have proposed a standard method for stimulus selection based on cluster analysis. Their motivation was to re-create group structures that are most likely to produce valence and arousal norms associated with the IAPS images. The method included outlier analysis, the identification of suitable clustering solutions, and the extraction of stimuli based on their level of certainty of belonging to the assigned cluster. Another feature of their method was maintaining statistical power in studies by maximizing the likelihood that the stimuli belong to the cluster structure fitted to them, and by filtering stimuli according to their certainty of cluster membership. This study used a different affective multimedia database (i.e., IAPS) [18].

Generally, compared to previously published studies on the clustering of pictures, and specifically, to even more closely related studies in affective computing, the experiment presented in this paper has three novel differences: (1) the largest available dataset of affective pictures was utilized in the experiment, (2) cluster robustness was assessed with the Monte-Carlo method, and (3) the employed dataset makes use of both discrete and dimensional models of affect, thereby allowing subsequent investigations of deeper statistical relationships and correlations between stimuli emotions and semantics.

4. Unsupervised Machine Learning Methods

Unsupervised machine learning is a common name for methods that automatically attempt to find a particular structure, relation, or other statistical feature from a set of inputs that are not necessarily obvious, or the relevant conclusion cannot be reached by traditional statistical methods. Clustering algorithms are often associated with unsupervised learning, that is, finding groups or clusters of data. The purpose of this partitioning is to divide the data into representative groups to facilitate analysis and, in some cases, to find certain interesting properties of such groups that are not initially obvious. However, this division is often only abstract and has no meaning, and it usually depends on the nature of the given measurements and the distribution of these data.

4.1. k-Means Algorithm

The most common and basic unsupervised machine-learning method, or in other words, the typical first choice for the data distribution algorithm, is the k-means clustering algorithm [8], which attempts to automatically partition the input data set into k partitions.

Details of how this algorithm works are described in [8], and only the necessary steps and interface of the algorithm are described here. The procedure reduces to determining the mid-points (so-called centroids) and assigning each input point to one of these centroids, using the square distance reduction criterion. Each of the centroids, and hence the associated distributions, is given its own unique index from 0 to $k - 1$, and the indexes of the associated (nearest) centroids are then associated with the input data points.

The result is a vector column with associated indexes (from 0 to $k - 1$) in the same number and order as the supplied input points. The algorithm can also be implemented

for multidimensional data—not just for $D = 2$ or $D = 3$. For $D > 3$, visualization is possible only by the chosen projection.

A typical implementation, often referred to as Lloyd's algorithm [24], behaves linearly in terms of time asymptotic complexity if the given data structure is such that it already naturally contains data clusters through its distribution. In general, the worst case is expressed as $O(nkdi)$ where n is the number of points (examples), k is the number of given groups (data clusters), d is the number of dimensions of the points and finally, I is the number of iterations of the algorithm.

This distribution method was chosen for the purpose of this study. The k-means algorithm is simple to implement programmatically, while its algorithmic complexity level is low. In the next section, some shortcomings of the k-means algorithm are described with possible solutions.

4.2. Disadvantages of the k-Means Algorithm and the Solutions Used

The k-means algorithm is a popular choice due to its simplicity and good performance, but shortcomings are known in practice that can lead to errors, confusion, or erroneous inference.

Most standard implementations have an inherent lack of deterministic behavior, while one of the core values in scientific and engineering practice is precisely the consistency and reproducibility of the performed investigations. The reason for the non-deterministic behavior is the way in which the solution is reached. The solution to k-means is an NP-hard problem [25]. A solution for the clustering task can be reached based on the expectation maximization algorithm, which is an efficient yet approximate algorithm [26]. Obtaining non-deterministic results in k-means is caused by the way the centroids are initialized at the very beginning of the algorithm, through the pseudorandom selection of the starting points. In addition, the assigned distribution indexes may differ when comparing the outputs after several runs of the clustering procedure with the k-means algorithm.

Another drawback, which is also indicated by the name of the algorithm, is the parameter k, which defines the given number of distributions and must be determined in advance. The optimal parameter k is not always easy to determine, and justification must be given as to why a particular number or range of values was chosen.

The difficulties described are analyzed in the remainder of this chapter—the chosen solutions are described, together with the key parts of the implementation, all with the aim of achieving consistent behavior and obtaining meaningful results in the distribution of the data.

4.2.1. Unstable Cluster Indexes

A minor difficulty in the output of the k-means algorithm, specifically in the implementation of the scikit-learn program library in the Python programming language used to develop the software for this study, is unstable distribution indexes. For example, if distribution indexes 0, 1, 2, and 3 are associated with the specific colors purple, blue, yellow, and red, respectively, then the corresponding output graphs in Figure 3 shows how the color order changes with respect to the position distribution. In other words, the distribution indexes are not correlated with the centroid positions but are permuted randomly.

In order for the distribution indexes to be assigned to the centroids in a stable way, it is necessary to make index substitutions after starting the k-means algorithm. The proposed way is to assign the original unstable distribution index from the k-means algorithm to each centroid, then the centroid coordinates are lexicographically sorted with the original indexes, and finally the original indexes are replaced with new centroid indexes in the sorted field. With this method, the distributions are always stably distributed and, following the example in Figure 3, show that the color order in the output graph is consistent after each run.

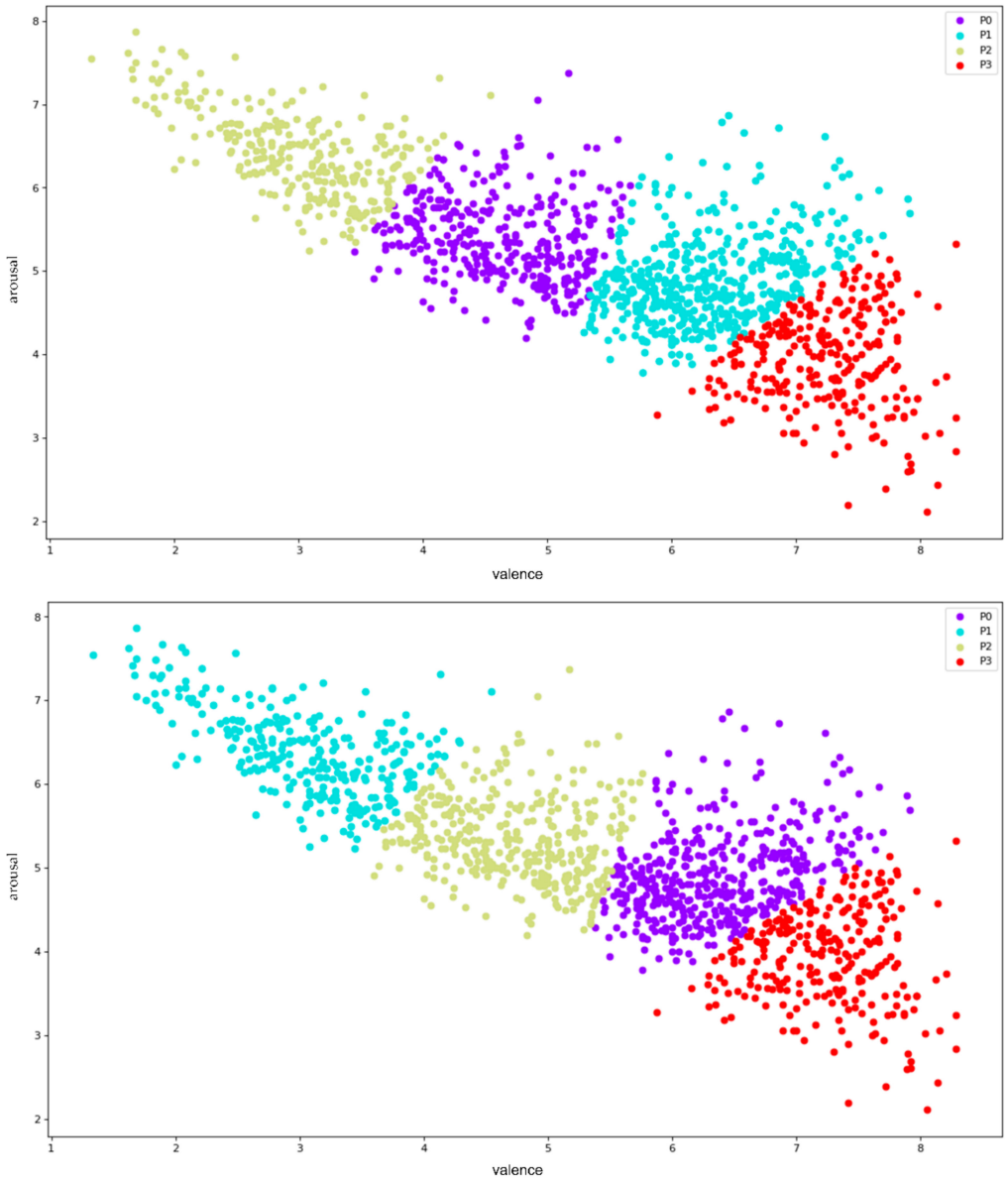

Figure 3. Examples of unstable distribution indexes and cluster order permutations for four possible distribution indexes.

4.2.2. Statistical Distribution Undecidability

A more significant problem with the k-means algorithm is that its behavior is nondeterministic in most implementations. This problem is illustrated in Figure 4. The issue is easily noticed after several successive program executions, and by comparing the output graphs, one can detect indecision in the affiliation of points between two adjacent groups. The probability of occurrence of this indecision is higher the closer the point is to the average distribution boundary. This is due to the fact that the assigned distribution indexes, as well as the cluster boundaries themselves, depend on the initial positions of the centroids.

Typically, the initial positions of centroids are chosen pseudorandomly so as not to give particular importance to individual points, and the practice is similar in controlled machine-learning algorithms, e.g., when initializing the parameters of a neural network.

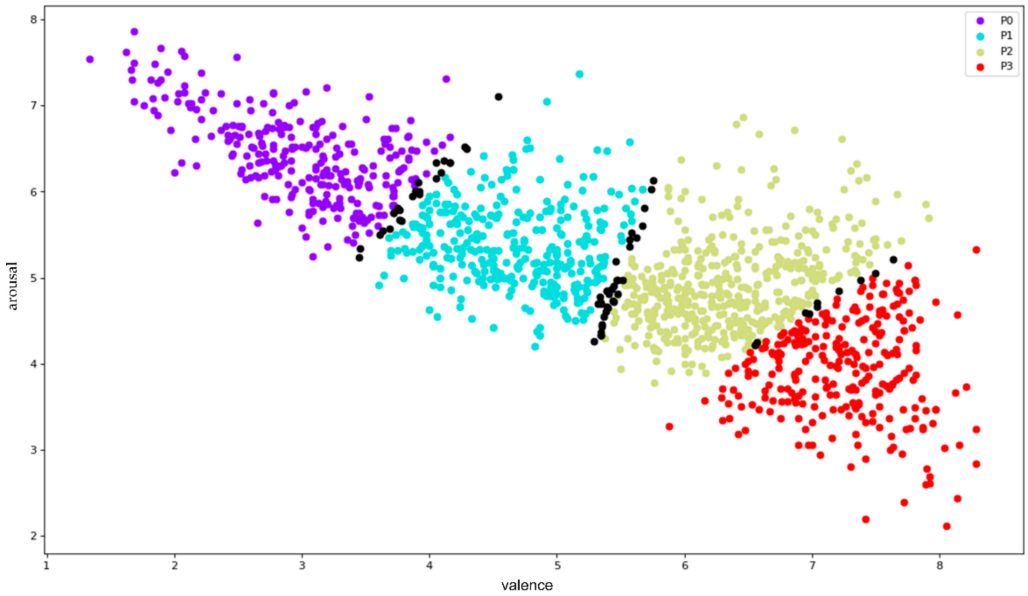

Figure 4. An example of the undecidability of the distribution. Black dots represent the unstable cluster affiliation of pictures in the feature space (valence, arousal).

To initialize the pseudorandom number generator, practical implementations often use an operating system clock (typically the system clock or network server timer) or some other source of entropy in the system (e.g., voltage fluctuations of a device, events at network interfaces, or user interactions), especially when randomness is much more important for security, as is the case with cryptographic functions.

In order to be able to reproduce the result of the algorithm in a stable way, a solution was chosen by means of a Monte-Carlo simulation [27]. For each point (valence–arousal pair), a histogram of affiliation to the distributions was calculated. The indexes (columns) in each histogram are also stable distribution indexes, and the values are the number of occurrences of a point in this cluster index (frequency, i.e., probability of belonging to the distribution). The histograms are filled in by a certain number of iterations of the k-means algorithm (e.g., $p = 1000$). Finally, all histograms are read with the argmax function, which selects the index with the highest frequency and declares it the stable distribution index for the corresponding point.

However, the described method has a hidden flaw, namely the probability that the distribution in the histogram is (at least partially) uniform (e.g., 50/50), especially for points closer to the average distribution boundary and very close to some of the other points. Ignoring this phenomenon, the indecision would be automatically resolved by the distribution index, i.e., the lexicographic order of the centroid, since argmax always selects the smallest index with the maximum value. However, this is not a desirable outcome because it unjustifiably introduces an arbitrary heuristic about the method of data distribution.

The empirical method, which involved successive simulations for $k = 4$ with the choice of the number of iterations, showed that on average, there is 0.3% (i.e., only 4 out of 1356 points from the entire NAPS database) of the occurrence of uniform distribution (50/50) after 50 iterations, and 0% after 1000 iterations. In other words, all points stabilize with more than 50% probability in one of the data clusters.

By successive runs with 2000 iterations each, it was shown that a smaller number of points (less than 5, i.e., 0.37%) can still behave as undecidable at the distribution boundaries. In other words, picture cluster memberships of a two-simulation run can differ, but only by a marginal amount. Therefore, the method was observed to behave satisfactorily statistically stable.

4.3. Defining the Optimal Number of Clusters (Parameter k)

A well-known limitation of the k-means algorithm is the need to determine the number of distributions (parameter k) in advance. Often, this number is relatively small (e.g., 2 to 10), and the choice is arbitrary in the absence of a pre-defined distribution goal.

Choosing a favorable number k in terms of optimizing a given property, such as maximizing variation, can be done by several methods, and two commonly used methods have been evaluated [28]: (1) the "elbow" method [29] and (2) the silhouette method [30].

The elbow method is probably the best-known method, where the values of the parameter k are iterated in a given range and the value of the distortion or inertia for each k is calculated and visualized in a graph. The distortion is defined as the average of the squared distances to the cluster centers of each cluster. On the other hand, the inertia is the sum of the squared distances of the samples to their nearest cluster center. Typically, the Euclidean distance metric is used. To determine the optimal number of clusters, the value of k must be chosen at the "elbow" (a change in graph slope from steep to flat), or in other words, at the point after which the distortion or inertia begins to decrease linearly. The elbow method is inaccurate in practice, but still potentially useful.

The other popular method for analyzing the separation interval between the resulting clusters is the silhouette method. The silhouette method attempts to estimate for each data point how strongly it belongs to the assigned cluster and, at the same time, how weakly it belongs to other clusters. The corresponding silhouette plot shows a measure of how close each point in a cluster is to points in neighboring clusters, providing a way to visually assess parameters, such as the number of clusters. The output value of this metric is a silhouette coefficient, or silhouette score, in the range [–1, 1]. Values close to +1 indicate that the sample is far from the neighboring clusters. A value of 0 means that the sample is at or above the decision boundary between two adjacent clusters, while negative values mean that the samples have been assigned to the wrong cluster.

5. Experiment and Results

The experimental dataset consisted of 1356 pictures from the NAPS repository, semantically and emotionally annotated with an unsupervised keyword vocabulary and two orthogonal emotional dimensions: valence (*val*) and arousal (*ar*). The robustness of the cluster solutions was assessed using a Monte-Carlo simulation stabilized k-means algorithm. Input features for k-means were vectors [*val*, *ar*], one for each picture in the dataset.

The approach described in Section 4 used the argmax function over the histograms of the Monte-Carlo simulation with $p = 2000$ iterations. The results are shown in Figures 5–8 and discussed in Sections 5.1 and 5.2.

5.1. The Optimal Number of Clusters

Using the elbow method on the NAPS database distribution, the graph shown in Figure 5 was obtained.

The elbow method shows a very smooth transition. Preferably, the bend of the "elbow" should be much more distinct. Nevertheless, it is noticeable that after $k = 8$, the bending angle of the curve exceeds 45°. This suggests that after more than 8 distributions, there is not much variation between cluster data.

Calculating the silhouette curve produced the graph shown in Figure 6, with the higher value reflecting the stronger average affiliation of the points to their clusters for a given k.

It can be noticed that the evaluation by the silhouette method gives a similar result as the elbow method—when there are more than eight distributions, the evaluation is consistently low. Both methods suggest that a range for $k = 2$–8 is sufficient for further analysis of the NAPS database. Additional confirmation for this choice is the previous research with a similar IAPS affective multimedia database described in Section 3 [21–23] where this range of values was also chosen as the target number of data clusters.

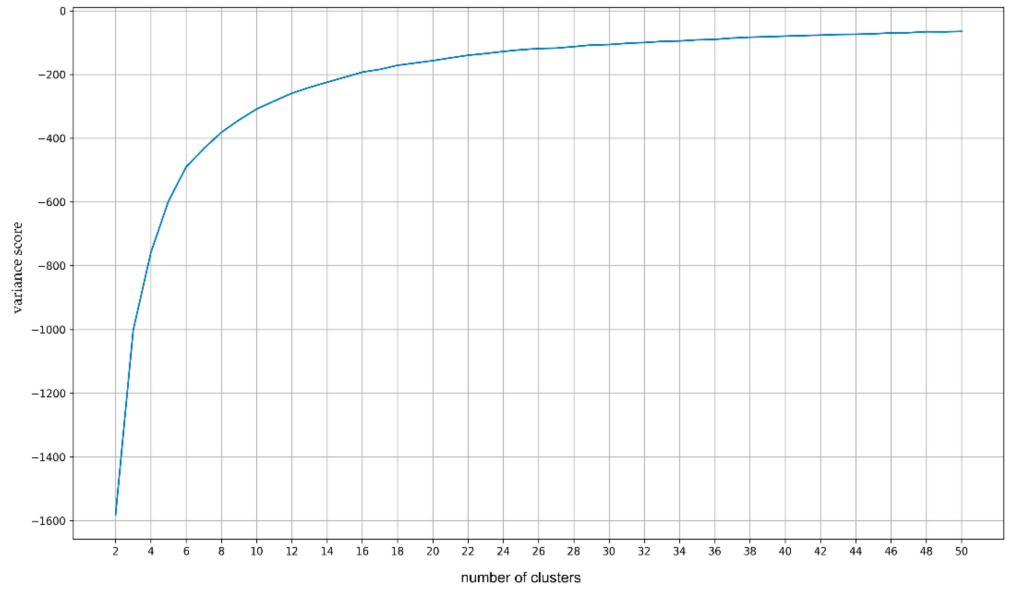

Figure 5. Estimation of variation by number of distributions using the elbow method.

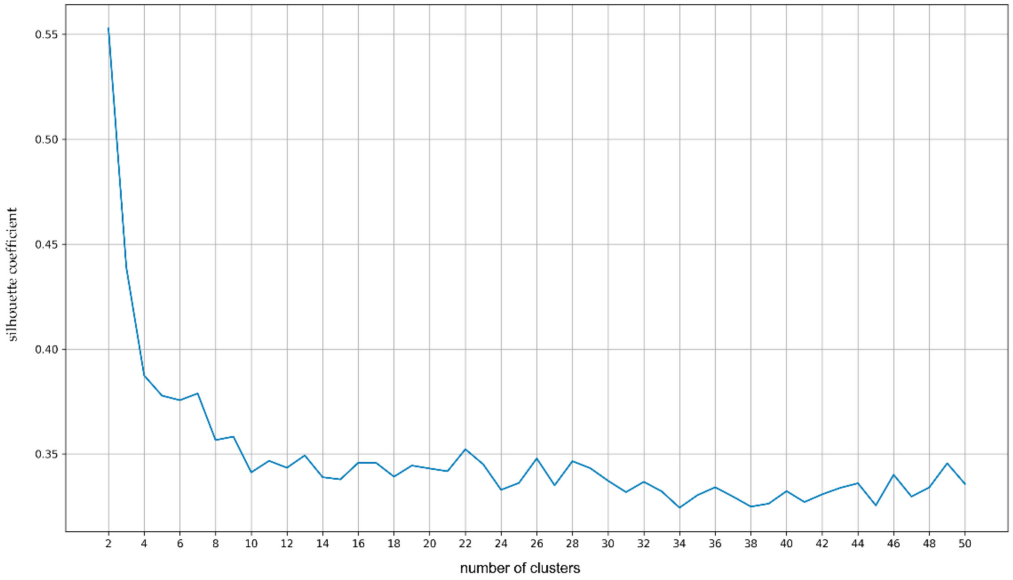

Figure 6. Estimation of variation by the number of distributions using the silhouette method.

5.2. Reliability of the Stable Distribution Method

The proposed method for stable data distribution has been shown to be sufficiently accurate for application to target datasets, but it is not completely deterministic. Although it is possible to obtain different results when the Monte-Carlo simulation is run in succession, the observed difference is almost negligible.

The reliability of the method is quantitatively evaluated by measuring the total error based on 10 simulation runs. The error is first estimated as a function of the number of iterations $e(p)$ with $k = 4$, and then as a function of the number of clusters $e(k)$ with $p = 100$.

The error function is calculated from the histogram of cluster affiliation according to the algorithm described below where the defined variables are as follows: n is the number of examples (data points), k is the number of distributions, p is the number of iterations of the Monte-Carlo method, s is the number of repetitions for the error measurement, and e is the total error.

The procedure for calculating the total stability error is defined as:

1. Calculate the histogram, i.e., the matrix of cluster affiliation (n x k) through s simulations.
2. All elements of the matrix that are equal to s are reset to zero because these points are stable.
3. For each row (example) in the matrix, count columns other than zero.
4. Subtract 1 from each such row (one column is considered correct).
5. The total error e is then the sum of all the rows from Step 4.

For example, in the experiment $n = 6$, $k = 3$ and $s = 10$, a possible ideal histogram h_{ideal} looks like this:

$$h_{ideal} = \begin{bmatrix} 10 & 0 & 0 \\ 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \\ 0 & 0 & 10 \end{bmatrix}$$

Ideally, all points belong to their data cluster in a completely stable way so that the total error is 0. When errors are present, as shown in the following example, the histogram contains elements smaller than the number of simulation repetitions s:

$$\begin{bmatrix} 10 & 0 & 0 \\ 9 & 1 & 0 \\ 3 & 5 & 2 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \\ 0 & 0 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 9 & 1 & 0 \\ 3 & 5 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow 3$$

From this case, it can be observed that the second sample has a split affiliation between two clusters (typical error distribution) in the ratio of 90% and 10%, while the third sample is even more scattered—between three clusters, with a 30%, 50% and 20% probability. Taking NAPS as an example, the error distribution in three clusters can be observed starting from $k = 5$, since the k-means algorithm for a smaller number only generates boundaries between two adjacent clusters.

According to the procedure described, the total error is actually the number of unstable clusters, i.e., the number of elements not equal to 0 and 10 reduced by one "correct" cluster per line. In the example above, the total error is $e = 3$.

The error function counts unstable clusters, so it does not depend on the number of simulation runs s and by increasing s, a more reliable error is reported in Appendix A. The error for a sample, i.e., the row, can be interpreted as the degree of scattering where the degree zero indicates non-scattering, i.e., the case where the point statistically belongs to only one cluster.

The Figure 7 below shows the overall stability error as a function of the selected number of iterations of the Monte-Carlo simulation for a stable distribution of points from the NAPS database with $k = 4$. It can be observed that the stability displays more variance when in the 0 to 200 iteration range.

More precisely, the graph shown in Figure 7 is composed of two iteration intervals (p) with different point densities: an interval from 10 to 190 with step 20, and an interval from 200 to 2400 with step 200 for which the total error values are shown on the y-axis. Further, measurements were performed with $p = 2000$ simulations, where the total error is $e = 4$. Since in simulations with $k = 4$ (i.e., four clusters or groups) there are only boundaries between two adjacent clusters, it can be concluded that the error affects, at most, 4 out of 1356 points from the NAPS database or 0.29% of the dataset. This suggests that the cluster affiliations of 99.71% of pictures remained unchanged in comparison with the two starts of the stable distribution algorithm.

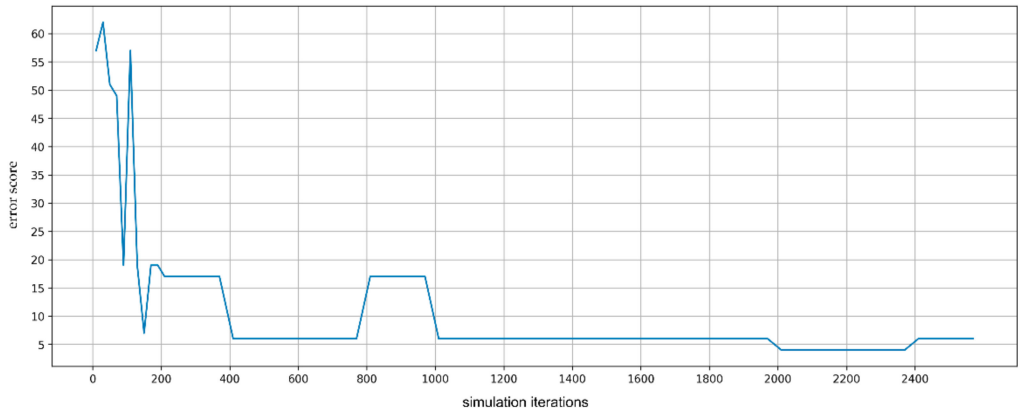

Figure 7. Overall stability error of the distribution method with respect to the number of simulation iterations.

Figure 8 below shows the overall stability error as a function of the chosen number of data clusters (k) for the stable distribution of points from the NAPS database, where $p = 100$ is given.

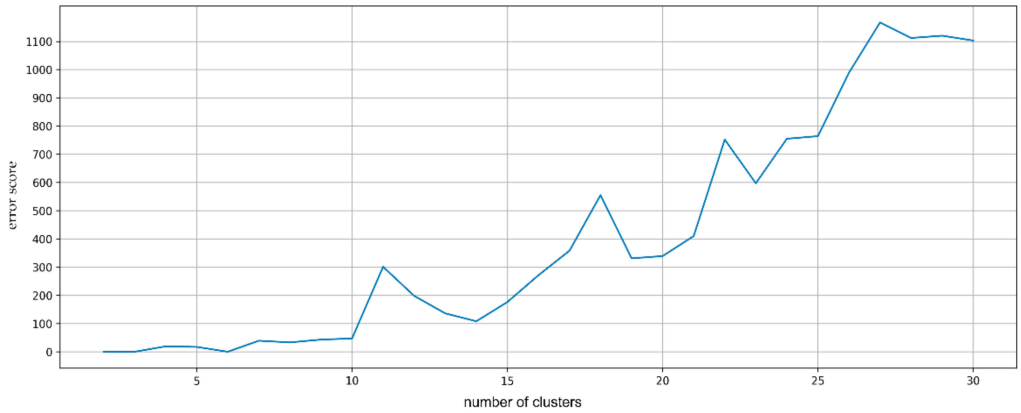

Figure 8. Overall stability error of the distribution method with respect to the number of clusters.

The graph shows that the error increases linearly as the number of clusters increases, and the processed parameter range $k = 2$–8 causes a small relative stability error ($e < 40$), even for a small number of iterations. The measured error values for this parameter range are in accordance with the results of the methods used for selecting the parameter k described in Section 5.2. and with conclusions from a previously published research with the NAPS dataset [21], which once again confirms that the parameter k is properly constrained in the range $k = 2$–8.

6. Conclusions

The relationships of affective data in the NAPS database were researched and presented, and an in-depth analysis of the stable cluster distribution method was performed. The research allows data scientists to find similarities within the affective picture data, draw inferences, find hidden patterns and also to reconstruct a possibly underlying data structure. As a method of unsupervised learning, the k-means algorithm was selected as the most common clustering algorithm, well-known and frequently used in many applications. Analysis of k-means shortcomings and solutions were put forward and practically applied to the largest emotionally annotated dataset available. In addition, novelties of this research compared to the other related studies have been explained.

The NAPS database, which contains 1356 semantically and emotionally annotated pictures with unsupervised vocabulary and two emotional dimensions, was processed by

a range of k-means parameter k. The derived picture clusters were the most stable for k in the range 2 to 8. The optimal values of the parameter were selected by calculating the variation using the elbow method and the silhouette method, and the range correctness was further confirmed by stability error analysis.

The nondeterministic behavior of the k-means algorithm was solved by Monte-Carlo simulation with the selection of the highest frequencies of affiliation in the distributions, so that the results could be reproduced without introducing arbitrary initial conditions.

The chosen method proved to be highly reliable. The stability error was measured to be only 0.29% of the total number of the NAPS database data points for the chosen number of data clusters $k = 4$ and $p = 2000$ iterations. Uncertainty of this magnitude is acceptable for most studies in which the method is to be used.

For the purpose of the presented research, a custom software application for data analysis was created in the Python programming language, using the scikit-learn machine learning library. The developed application was used as a computational tool in each step of the evaluation to arrive at the presented experimental results. The developed software tool may be freely used for academic and non-commercial purposes for experimentation in unsupervised machine learning, interactive viewing of graphs and semantic labels with pairs of selected emotion dimensions.

The described approach to validate cluster solutions in emotionally annotated pictures, using Monte-Carlo simulation, the developed software tool for unsupervised machine learning and the obtained results form the basis for further research in the field of affective multimedia, but also for the comparison of results of similar research, in-depth analysis in different aspects and application of other methods of unsupervised machine-learning in many similar fields.

Author Contributions: Conceptualization, M.H. and K.B.; methodology, M.H. and K.B.; software, K.B.; validation, M.H., A.J. and K.B.; formal analysis, M.H. and A.J.; investigation, K.B.; resources, M.H.; data curation, K.B.; writing—original draft preparation, M.H. and K.B.; writing—review and editing, A.J. and K.B.; visualization, M.H. and K.B.; supervision, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the Laboratory of Brain Imaging of the Nencki Institute of Experimental Biology for granting access to the NAPS dataset.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The entire work was fully supported by developing a Python application to produce tables and graphs, run the stable k-means algorithm, compute errors and perform other computation that can easily be reproduced. The architecture, although mostly procedural, is such that it allows for simple changes in code (often just configuration) and running multiple analyses with varying hyperparameters. By using object-oriented abstractions, primarily concerned with data input and preprocessing, the entire software solution can easily be applied to different data sources of similar shape.

The Python software tool is divided in five functional class modules:

(1) Analysis—the main program that runs the selected computation (snippet) and produces a graph or textual output. These outputs were directly used for analysis and are included in the paper as figures or tables.

(2) Runner—a class with all the computation and plotting logic on the higher abstraction level, e.g., for computing stable argmax partitions, plotting stability error curves, and computing silhouette scores. Lib—implements the lower-level library functions and abstractions, contains the following classes:

(3) InputData—abstraction for data input and output for the NAPS or other affective picture datasets with similar architectures;

(4) Config—class for configuring the k-means algorithm and evaluation parameters, other methods, such as dataset partitioning;
(5) PlotAnnotator—a class module that provides support for rendering interactive data plots in the tool's graphical user interface.

As described in Section 5.2, a specific metric was used to measure the degree of scattering of pictures between individual clusters. In the following program excerpt in pseudocode, the mentioned error function is implemented in the class Runner:

```
struct Point: (x, y)

# Assuming stable index coloring.
function KMeans(data: Vector of Point, k) -> Vector of Int

function MonteCarloKMeans(data, k, p) -> Vector of Int:
  n := length of data
  histogram := Array[n, k] of Int initialized to 0
  repeat p times:
    for i, c in KMeans(data, k):
      histogram[i, c] += 1
  stable_clusters := Vector[n] of Int
  for i := 0 to n-1:
    stable_clusters[i] := argmax of histogram[i]
  return stable_clusters

function StabilityError(data, k, p, s) -> Int:
  n := length of data
  histogram := Array[n, k] of Int initialized to 0
  repeat s times:
    for i, c in MonteCarloKMeans(data, k, p):
      histogram[i, c] += 1
  e := 0
  for i := 0 to n − 1:
    scattering := −1
    for c := 0 to k − 1:
      if histogram[i, c] > 0 and histogram[i, c] < s:
        scattering += 1
    if scattering > 0:
      e += scattering
  return e
```

The internal class structure of the software source code is presented as a UML class diagram in Figure A1.

The implemented procedure using Monte-Carlo simulation stable cluster coloring k-means is described with UML activity diagrams in Figure A2.

In this procedure, first the raw k-means algorithm (partition_naps function from the InputData class) is invoked to obtain the cluster vector which consists of the image index and the cluster index. The cluster index is not stable in color. Second, the reindex_partitions function from the InputData class is called with the input data (x, y) and the cluster vector from the previous step. Third, within the index_partitions function, in the partitions dictionary (size *K*, the key is the color index of an unstable cluster), the sum and number of points to determine the centroids are calculated as the arithmetic mean of the individual coordinates within the same cluster. The centroids are associated with the cluster indexes from the first step. Then, in the ordering field (size *K*), each partition is lexicographically sorted by the centroid coordinates. In the new_ordering field (size *K*), old unstable cluster indexes are mapped to new stable cluster indexes. Finally, the correct cluster vector is determined by replacing the old (unstable color) cluster vector with the new stable color determined by the centroid index.

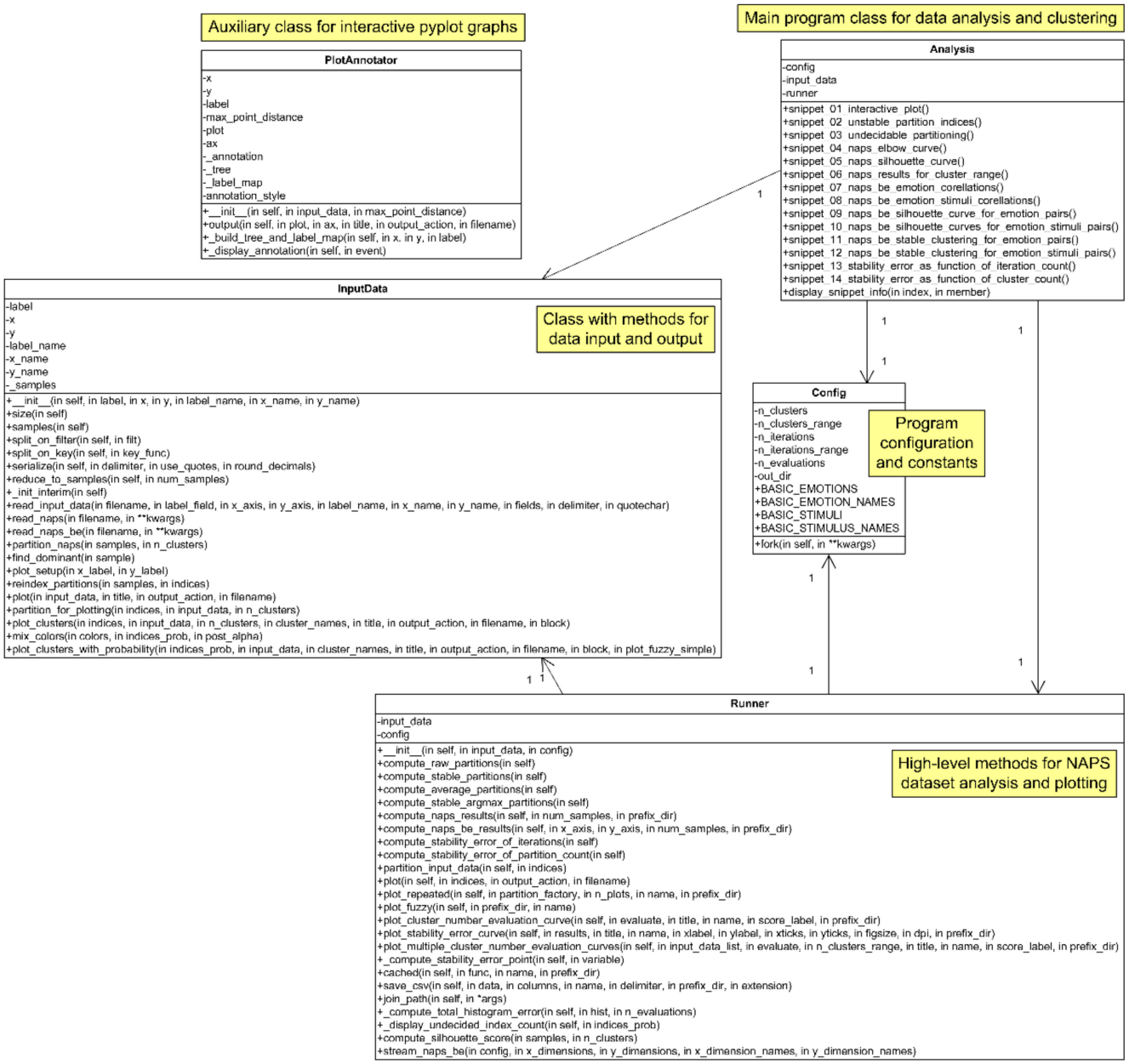

Figure A1. UML class diagram showing the software tool's five functional class modules (Analysis, Runner, InputData, Config, PlotAnnotator), their attributes, operations and mutual relationships.

The Monte-Carlo simulation for determining stable clusters is encapsulated in the function MonteCarloKMeans (compute_average_partitions method of the Runner class) and can be described as follows; let N be the size of the input data (number of sample points), K the target number of clusters, and P the number of Monte-Carlo iterations. First an array H of size $N \times K$ is initialized. The array represents a histogram in which the statistical behavior, i.e., a non-deterministic choice of cluster $0 \ldots K-1$ of each sample in range $0 \ldots N-1$ is recorded.

Then, function StableColoredKMeans (compute_stable_partitions method in the class Runner) is run P times, and at each iteration updates the histogram H (increment the "bar" of the chosen cluster) for all N samples respectively. Note that StableColoredKmeans returns a vector of size N in which all values are between $0 \ldots K-1$ (i.e., the choice of cluster for each of the N samples).

In the following step, the software tool computes argmax on each row of H (compute_stable_argmax_partitions in the class Runner). By doing this, the index of the most frequently chosen cluster of each sample is selected. The resulting vector of size N is then the stable interpretation of KMeans.

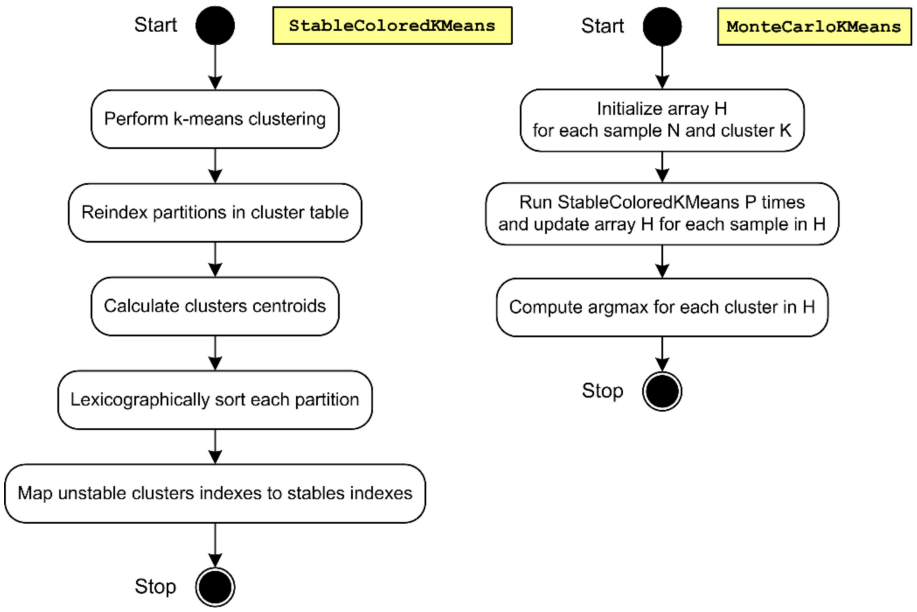

Figure A2. The clustering procedure using Monte-Carlo simulation stabilized k-means implemented in the Python software tool. UML activity diagrams illustrating functions StableColoredKMeans (**left**) and MonteCarloKMeans (**right**).

It should be noted, as previously described, argmax has a built-in bias toward lower indexes. If two columns hold the same value $\max(H[i])$ then the one appearing earlier is selected. This can be accounted for by scanning through the sample's histogram, and then choosing how to resolve it upon detection (e.g., ignore it, log the occurrence, remove the point from the data, etc.) For a more robust solution, a threshold may also be applied in the detection, such as if $H[i] > 0.95 * \max(H[i])$ for each j in [0, k $-$ 1] and j!= argmax($H[i]$), then the sample is undecidable (with a threshold of 5%).

The software tool is freely available for scientific and non-commercial purposes at https://github.com/kburnik/naps-clustering (accessed on 30 April 2021). For all inquiries, please contact the third author. The archive does not contain the NAPS repository. To request the NAPS for non-profit academic research purposes, contact Nencki Institute of Experimental Biology, Laboratory of Brain Imaging (LOBI) at https://lobi.nencki.gov.pl/research/8/ (accessed on 30 April 2021).

References

1. Omran, M.G.H.; Engelbrecht, A.P.; Salman, A. An overview of clustering methods. *Intell. Data Anal.* **2007**, *11*, 583–605. [CrossRef]
2. Alelyani, S.; Tang, J.; Liu, H. Feature Selection for Clustering: A Review. In *Data Clustering: Algorithms and Applications*; Aggarwal, C., Reddy, C., Eds.; CRC Press: Boca Raton, FL, USA, 2013.
3. de Amorim, R.C.; Hennig, C. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Inf. Sci.* **2015**, *324*, 126–145. [CrossRef]
4. Calvo-Zaragoza, J.; Valero-Mas, J.J.; Rico-Juan, J.R. Prototype generation on structural data using dissimilarity space representation. *Neural Comput. Appl.* **2017**, *28*, 2415–2424. [CrossRef]
5. Cios, K.J.; Swiniarski, R.W.; Pedrycz, W.; Kurgan, L.A. Unsupervised learning: Clustering. In *Data Mining*; Springer: Boston, MA, USA, 2007; pp. 257–288.
6. Celebi, M.E.; Aydin, K. (Eds.) *Unsupervised Learning Algorithms*; Springer: Berlin, Germany, 2016.
7. Kameshwaran, K.; Malarvizhi, K. Survey on clustering techniques in data mining. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 2272–2276.

8. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [CrossRef]
9. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *InKdd* **1996**, *96*, 226–231.
10. Sinaga, K.P.; Yang, M.S. Unsupervised K-means clustering algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
11. Horvat, M.; Popović, S.; Ćosić, K. Towards semantic and affective coupling in emotionally annotated databases. In Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2012, Opatija, Croatia, 21–25 May 2012; pp. 1003–1008.
12. Colden, A.; Bruder, M.; Manstead, A.S. Human content in affect-inducing stimuli: A secondary analysis of the international affective picture system. *Motiv. Emot.* **2008**, *32*, 260–269. [CrossRef]
13. Horvat, M. A Brief Overview of Affective Multimedia Databases. In *Central European Conference on Information and Intelligent Systems*; Faculty of Organization and Informatics: Varaždin, Croatia, 2017; pp. 3–9.
14. Marchewka, A.; Żurawski, Ł.; Jednorog, K.; Grabowska, A. The Nencki Affective Picture System (NAPS): Introduction to a novel, standardized, wide-range, high-quality, realistic picture database. *Behav. Res. Methods* **2014**, *46*, 596–610. [CrossRef] [PubMed]
15. Riegel, M.; Żurawski, Ł.; Wierzba, M.; Moslehi, A.; Klocek, Ł.; Horvat, M.; Grabowska, A.; Michałowski, J.; Marchewka, A. Characterization of the Nencki Affective Picture System by discrete emotional categories (NAPS BE). *Behav. Res. Methods* **2016**, *48*, 600–612. [CrossRef] [PubMed]
16. Peter, C.; Herbon, A. Emotion representation and physiology assignments in digital systems. *Interact. Comput.* **2006**, *18*, 139–170. [CrossRef]
17. Posner, J.; Russell, J.A.; Peterson, B.S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Dev. Psychopathol.* **2005**, *17*, 715. [CrossRef] [PubMed]
18. Lang, P.J.; Bradley, M.M.; Cuthbert, B.N. *International Affective Picture System (IAPS): Affective Ratings of Pictures and Instruction Manual*; Technical Report A-8; University of Florida: Gainesville, FL, USA, 2008.
19. Wierzba, M.; Riegel, M.; Pucz, A.; Leśniewska, Z.; Dragan, W.Ł.; Gola, M.; Jednorog, K.; Marchewka, A. Erotic subset for the Nencki Affective Picture System (NAPS ERO): Cross-sexual comparison study. *Front. Psychol.* **2015**, *6*, 1336. [CrossRef] [PubMed]
20. Kensinger, E.A.; Schacter, D.L. Processing emotional pictures and words: Effects of valence and arousal. *Cogn. Affect. Behav. Neurosci.* **2006**, *6*, 110–126. [CrossRef] [PubMed]
21. Horvat, M.; Jednoróg, K.; Marchewka, A. Clustering of Affective Dimensions in Pictures: An exploratory analysis of the NAPS database. In Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2016, Opatija, Croatia, 30 May–3 June 2016; pp. 1496–1501.
22. Horvat, M.; Popović, S.; Ćosić, K. Multimedia stimuli databases usage patterns: A survey report. In Proceedings of the 36th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2013, Opatija, Croatia, 20–24 May 2013; pp. 993–997.
23. Constantinescu, A.C.; Wolters, M.; Moore, A.; MacPherson, S.E. A cluster-based approach to selecting representative stimuli from the International Affective Picture System (IAPS) database. *Behav. Res. Methods* **2017**, *49*, 896–912. [CrossRef] [PubMed]
24. Hamerly, G.; Drake, J. Accelerating Lloyd's algorithm for k-means clustering. In *Partitional Clustering Algorithms*; Springer: Cham, Switzerland, 2015; pp. 41–78.
25. Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. The planar k-means problem is NP-hard. *Theor. Comput. Sci.* **2012**, *442*, 13–21. [CrossRef]
26. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2000.
27. Kroese, D.P.; Brereton, T.; Taimre, T.; Botev, Z.I. Why the Monte Carlo method is so important today. *Wiley Interdiscip. Rev. Comput. Stat.* **2014**, *6*, 386–392. [CrossRef]
28. Cluster Validation Essentials. Available online: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/ (accessed on 31 March 2021).
29. Ketchen, D.J.; Shook, C.L. The application of cluster analysis in strategic management research: An analysis and critique. *Strateg. Manag. J.* **1996**, *17*, 441–458. [CrossRef]
30. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]