Reports of the Department of Mathematical Information Technology
Series C. Software and Computational Engineering
No. C. 1/2006

Introduction to partitioning-based clustering methods with a robust example

Sami Äyrämö   Tommi Kärkkäinen

University of Jyväskylä
Department of Mathematical Information Technology
P.O. Box 35 (Agora)
FI–40014 University of Jyväskylä
FINLAND
fax +358 14 260 2731
http://www.mit.jyu.fi/

ISBN 951-39-2467-X
ISSN 1456-4378

Introduction to partitioning-based clustering methods with a robust example*

Sami Äyrämö[†] Tommi Kärkkäinen[‡]

Abstract

Data clustering is an unsupervised data analysis and data mining technique, which offers refined and more abstract views to the inherent structure of a data set by partitioning it into a number of disjoint or overlapping (fuzzy) groups. Hundreds of clustering algorithms have been developed by researchers from a number of different scientific disciplines. The intention of this report is to present a special class of clustering algorithms, namely partition-based methods. After the introduction and a review on iterative relocation clustering algorithms, a new robust partitioning-based method is presented. Also some illustrative results are presented.

1 Introduction

Data clustering, by definition, is an exploratory and descriptive data analysis technique, which has gained a lot of attention, e.g., in statistics, data mining, pattern recognition etc. It is an explorative way to investigate multivariate data sets that contain possibly many different data types. These data sets differ from each other in size with respect to a number of objects and dimensions, or they contain different data types etc. Undoubtedly, the data clustering belongs to the core methods of data mining, in which one focuses on large data sets with unknown underlying structure. The intention of this report is to be an introduction into specific parts of this methodology called cluster analysis. So called partitioning-based clustering methods are flexible methods based on iterative relocation of data points between clusters. The quality of the solutions is measured by a clustering criterion. At each iteration, the iterative relocation algorithms reduce the value of the criterion function until convergence. By changing the clustering criterion, it is possible to construct robust clustering methods that are more insensitive to erroneous and missing

*This work was supported by National Technology Agency of Finland under the project no. 40280/05.

[†]Agora Center, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, sami.ayramo@mit.jyu.fi

[‡]Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, tka@mit.jyu.fi

data values than classical methods. Surprisingly, most of ”real-data” is of this form [81, 13, 54]. Hence, in the end of this report, an example of robust partitioning-based cluster analysis techniques is presented.

Next to this introduction, various definitions for cluster analysis and clusters are discussed. Thereafter, in the third section, a principle of partitioning-based clustering is presented with numerous examples. A special treatment is given for the well-known K-means algorithm. The fourth chapter consists of discussion about robust clustering methods. In the sixth section, a novel partitioning-based method, which is robust against outliers and based on the iterative relocation principle including the treatment for missing values, is introduced. The last section contains the final summary for the report.

2 What is cluster analysis?

Cluster analysis is an important element of exploratory data analysis. It is typically directed to study the internal structure of a complex data set, which can not be described only through the classical second order statistics (the sample mean and covariance). Already in 1967, MacQueen [92] stated that clustering applications are considered more as an aid for investigators to obtain qualitative and quantitative understanding of a large amount of multivariate data than only a computational process that finds some unique and definitive grouping for the data. Later, due to its unsupervised, descriptive and summarizing nature, data clustering has also become a core method of data mining and knowledge discovery. Especially during the last decade, the increasing number of large multidimensional data collections have stepped up the development of new clustering algorithms [54, 56, 118].

Generally speaking, the classification of different things is a natural process for human beings. There exist numerous natural examples about different classifications for living things in the world. For example, various animal and plant species are the results of unsupervised categorization processes made by humans (more precisely, domain experts), who have divided objects into separate classes by using their observable characteristics [45]. There were no labels for the species before someone generated them. A child classifies things in an unsupervised manner as well. By observing similarities and dissimilarities between different objects, a child groups those objects into the same or different group.

At the time before the computers came available, clustering tasks had to be performed manually. Although it is easy to visually perceive groups from a two- or three-dimensional data set, such ”human clustering” is not likely an inconsistent procedure, since different individuals see things in different ways. The measure of similarity, or the level and direction one is looking at the data, are not consistent between different individuals. By direction we mean the set of features (or combinations of features) that one exploits when classifying objects. For example, people can be classified into a number of groups according to the economical status or the annual alcohol consumption etc. These groupings will not necessarily capture the same individuals [37]. The direction where the user is looking at the data set depends, for example, on her/his background (position, education, profession, culture etc.). It is clear that such things vary a lot among different individuals [70].

Numerous definitions for cluster analysis have been proposed in the literature. The definitions differ slightly from each other in the way to emphasize the different aspects of the methodology. In one of the earliest books on data clustering, Anderberg [5] defines cluster analysis as a task, which aims to *"finding of "natural groups" from a data set, when little or nothing is known about the category structure*". Bailey [8], who surveys the methodology from the sociological perspective, defines that "*cluster analysis seeks to divide a set of objects into a small number of relatively homogeneous groups on the basis of their similarity over $\mathcal{N}$ variables."* $\mathcal{N}$ is the total number of variables in this case. Moreover, Bailey notes that "*Conversely variables can be grouped according to their similarity across all objects.*". Hence, the interest of cluster analysis may be in either grouping of objects or variables, or even both (see also [37, p.154-155]). On the other hand, it is not rare to reduce the number of variables before the actual object grouping, because the data can be easily compressed by substituting the correlating variables with one summarizing and representative variable. From the statistical pattern recognition perspective, Jain et al. [69] define cluster analysis as "*the organization of collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity*". Hastie et al. [58] define the goal of cluster analysis from his statistical perspective as a task "*to partition the observations into groups ("clusters") such that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters*". Tan et al. [118] states from data mining point of view that "*Cluster analysis divides data into groups that are meaningful, useful, or both.*". By meaningful they refer to clusters that capture the natural structure of a data set, whereas the useful clusters serve only as an initial setting for some other method, such as *PCA* (*principal component analysis*) or regression methods. For these methods, it may be useful to summarize the data sets beforehand.

The first definition emphasizes the unknown structure of the target data sets, which is one of the key assumptions in cluster analysis. This is the main difference between clustering (*unsupervised classification*) and classification (*supervised classification*). In a classification task the category structure is known a priori, whereas the cluster analysis focuses on the object collections, whose class labels are unknown. Jain et al. [71] suggest that the class labels and all other information about data sources, have an influence to the result interpretation, but not to the cluster formation process. On the other hand, the domain understanding is often of use during the configuration of initial parameters or correct number of clusters.

The second and third definitions stress the multi-dimensionality of the data objects (observations, records etc.). This is an important notion, since the grouping of objects that possess more than three variables is no easy matter for a human being without automated methods. Naturally, most of the aforementioned definitions address the notion of similarity. Similarity is one of the key issues of cluster analysis, which means that one of the most influential elements of cluster analysis is the choice of an appropriate similarity measure. The similarity measure selection is a data-dependent problem. Anderberg [5] does not use term "similarity", but instead he talk about the degree of "natural association" among objects. Based on the aforementioned definitions and notions, the cluster analysis is summarized as "*analysis of the unknown structure of a multidimensional data set by determining a (small) number of meaningful groups of objects or variables according to a chosen (dis)similarity measure*". In this definition, the term meaningful is understood identically with Tan et al. [118].

Even though the visual perception of data clusters is a suitable method up to three dimensions, in more than three dimensional space the visual perception turns to a complex task and computers become indispensable. As we know that a human classifier is an inconsistent classifier, also different algorithms produce different groupings even for the same data set. Hence, there exist not any universally best clustering algorithm [4, 71]. On this basis, Jain et al. [71] advise one to try several clustering algorithms when trying to obtain the best possible understanding about data sets. Based on the authors experience and theoretical considerations, Kaufman et al. [79] propose six clustering algorithms (*PAM, CLARA, FANNY, AGNES, DIANA* and *MONA*) that they believe to cover a major part of the applications. PAM is a partitioning-based K*-medoid* method that divides the data into a given number disjoint clusters. CLARA, which also partitions a data set with respect to medoid points, scales better to large data sets than PAM, since the computational cost is reduced by sub-sampling the data set. FANNY is a fuzzy clustering method, which gives a degree for memberships to the clusters for all objects. AGNES, an agglomerative hierarchical clustering method produce a tree-like cluster hierarchy using successive fusions of clusters. The result provides a solution for different values of K. DIANA is also a hierarchical method, but it proceeds in an inverse order with respect to AGNES. At the beginning, DIANA puts all objects into one cluster and continues by splitting each cluster up to two smaller ones at each step. MONA is also a divisive algorithm, but the separation of objects into groups is carried out by using a single variable. The set of methods, which was just presented, should give a quite overall view to the internal structure of any data set. As mentioned earlier, the result interpretation step is a human process, in which one may utilize different visualization techniques (e.g., PCA and *MDS* (*multidimensional scaling*) [56]). After the interpretation, priori domain knowledge and any other problem related information are integrated to the clusters.

The development of clustering methods is very interdisciplinary. Contributions have been made, for example, by psychologists [96], biologists [121, 72], statisticians [42], social scientists [8], and engineers [70]. Naturally, various names for cluster analysis have emerged, e.g., numerical taxonomy, automatic classification, botryology, and typological analysis [79, p.3]. Also unsupervised classification [118, 69], data segmentation [58], and data partition [106] are used as synonyms for data clustering. Later, when data mining and knowledge discovery have grown further off the other original fields, and constituted its own separate scientific discipline, it has also contributed in a great amount to the development of clustering methods. The special focus has been on the computationally efficient algorithms for large data sets [54, 56, 118, 34]. Perhaps due to the interdisciplinary nature of the cluster analysis, the same methods are often invented with different names on different disciplines.

There exist huge amount of clustering applications from many different fields, such as, biological sciences, life sciences, medical sciences, behavioral and social sciences, earth sciences, engineering and information, policy and decision sciences to mention just a few [5, 70, 79, 69, 37, 123]. This emphasizes the importance of data clustering as a key technique of data mining and knowledge discovery [56, 54, 34, 48, 12, 47, 118], pattern recognition [120, 31, 32, 71, 45] and statistics [29, 58].

The range of clustering applications is very wide. It may be analysis of software modules and procedures [93, 127, 128], grouping customers of similar behavior in marketing research [13], classification of unknown radar emitters from received radar pulse samples [87], optimal placement of radioports in cellular networks [1], identification of subtypes of schizophrenia [60], archeological applications [3], peace science applications (identification of international conflicts [122]), P2P-networks [104] etc. The list above could be almost endless. It also contains some quite exotic examples.

2.1 The main elements of cluster analysis

Although the intuitive idea behind cluster analysis is simple, the successful completion of the tasks presume a large number of correct decisions and choices from several alternatives. Anderberg [5] states that there appears to be at least nine major elements in a cluster analysis study before the final results can be attained. Because the current real-world data sets contain missing values as well, we complete this element list with data presentation and missing data strategy [86, 75].

1. Data presentation.
2. Choice of objects.
3. Choice of variables.
4. What to cluster: data units or variables.
5. Normalization of variables.
6. Choice of (dis)similarity measures.
7. Choice of clustering criterion (objective function).
8. Choice of missing data strategy.
9. Algorithms and computer implementation (and their reliability, e.g., convergence)
10. Number of clusters.
11. Interpretation of results.

These are the most significant parts of the general clustering process. Jain et al. [71] suggest that the strategies used in data collection, data representation, normalization and cluster validity are as important as the clustering strategy itself. According to Hastie et al. [58, p.459], choice of the best (dis)similarity measure is even more important than the choice of clustering algorithms. This list could be also completed by validation of the resulting cluster solution [70, 4]. Validation is, on the other hand, closely related to the estimation of the number of clusters and to the result interpretation. For example, the visual exploration of the obtained solutions can be considered a kind of validation technique.

2.2 What is a cluster?

"Do not forget that clusters are, in large part, on the eye of the beholder."[35]

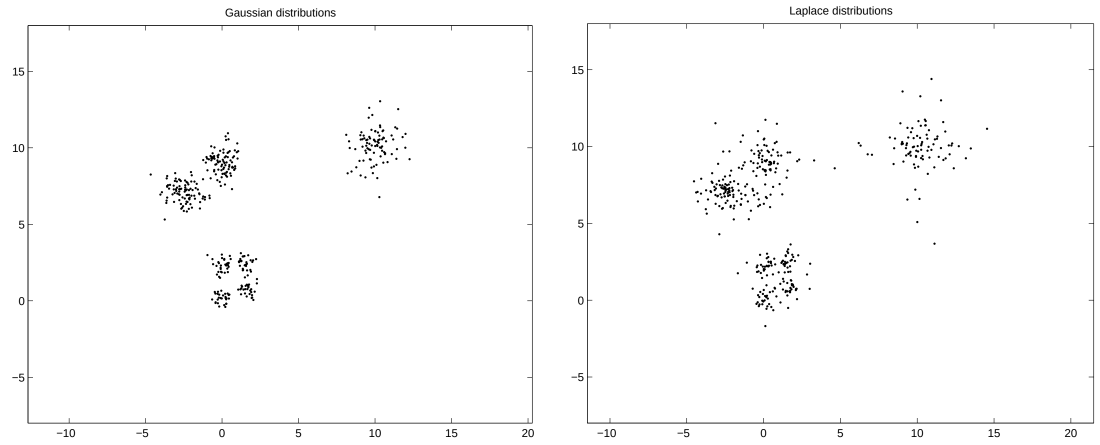

Figure 1: Illustration about the ambiguous number of clusters. On the left, there are seven clusters that are generated from a normal distribution using a set of different location and scatter parameters. Correspondingly, on the right, there are seven clusters that are drawn from the Laplace distribution using the same location and scatter parameters as in the normal case. It is not straightforward to say how many clusters there are, especially in the Laplace-case, because the clusters are inherently more spread in all directions.

Although the visual recognition of clusters from a two-dimensional view is usually easy, it is hard to give a formal definition for a cluster. Many authors with contributions in the clustering literature address the lack of the universal and formal definition of a cluster. However, at the same time, they agree that giving one is an intractable problem [118, 123, 37, 79, 4, 25]. The notion of a cluster depends on the application and it is usually weakly defined [118]. The goal of the cluster analysis task effects to the definition as well. Depending on the application, clusters have different shapes and size. Moreover, even the number of inherent clusters in the data is not unambiguous, because it depends on the resolution (local versus global) one is looking at the data [70, 118]. See Figure 1.

Typically clustering methods yield a data description in terms of clusters that possess strong internal similarities [31]. Often one defines the cluster in terms of internal cohesion (homogeneity) and external isolation (separation). Hence, the cluster is often simply considered as a collection of objects, which are similar to one another within the same cluster and dissimilar to the objects in other clusters [54]. An interesting connection to the software engineering is recognized, when we notice that the principle is very similar with the common software architecture rule on *"loose coupling and strong cohesion"*. Such architecture aims to localize effects caused by code modifications (see, e.g., Bachmann et al. [7]). The software components with a large number of mutual links can be considered close to each other. Hence, a good software architecture should contain clearly separated "component clusters".

Some common definitions are collected from the clustering literature and given below [70, 4].

- "A Cluster is a set of entities which are alike, and entities from different clusters are not alike."
- "A cluster is an aggregation of points in the space such that the *distance* between two points in the cluster is less than the distance between any point in the cluster and any point not in it."
- "Clusters may be described as connected regions of a multidimensional space containing a relatively *high density* of points, separated from other such regions by a region containing a relatively low density of points."

Although the cluster is an application dependent concept, all clusters are compared with respect to certain properties: density, variance, dimension, shape, and separation [4]. The cluster should be a tight and compact high-density region of data points when compared to the other areas of space. From compactness and tightness, it follows that the degree of dispersion (variance) of the cluster is small. The shape of the cluster is not known a priori. It is determined by the used algorithm and clustering criteria. Separation defines the degree of possible cluster overlap and the distance to each other. Fuzzy clustering methods produce overlapping clusters by assigning the degree of the membership to the clusters for each point [10, 11]. Traditional partitioning clustering methods, such as K-Means, and hierarchical methods produce separated clusters , which means that each data point is assigned to only one cluster. A cluster is defined in a dimension of its variables and, if having a round shape, it is possible to determine its radius. These are the measurable features for any cluster, but it is not possible to assign universal values or relations to them. Perhaps, the most problematic features are shape and size.

3 Partitioning-based clustering algorithms

Perhaps the most popular class of clustering algorithms is the combinatorial optimization algorithms a.k.a. iterative relocation algorithms. These algorithms minimize a given clustering criterion by iteratively relocating data points between clusters until a (locally) optimal partition is attained. In a basic iterative algorithm, such as K-means- or K-medoids, convergence is local and the globally optimal solution can not be guaranteed. Because the number of data points in any data set is always finite and, thereby, also the number of distinct partitions is finite, the problem of local minima could be avoided by using exhaustive search methods. However, this is truth only in theory, since finding the globally optimal partition is known to be NP-hard problem and exhaustive methods are not useful in practice. The number of different partitions for n observations into K groups is a Stirling number of the second kind, which is given by

$$S_n^{(K)} = \frac{1}{K!} \sum_{i=0}^{i=K} (-1)^{K-i} \binom{K}{i} i^n.$$

This shows that enumeration of all possible partitions is impossible for even relatively small problems. The problem is even more demanding when additionally the number of clusters is unknown. Then the number of different combinations is the sum of the Stirling numbers of the second kind:

$$\sum_{i=1}^{i=K_{max}} S_n^{(i)},$$

where K_{max} is the maximum number of cluster and it is obvious that $K_{max} <= n$. The fact is that exhaustive search methods are far too time consuming even with modern computing systems. Moreover, it seems be an infinite race between computer power and amount of data, which both have increased constantly during the last years. Therefore, more practical approach than exhaustive search is the iterative optimization.

3.1 Iterative relocation algorithm

The iterative optimization clustering starts with an initial partition. Quality of this partition is then improved by applying a local search algorithm to the data. Several methods of this type are often categorized as a partitioning cluster method (a.k.a. non-hierarchical or flat methods [32]). A general iterative relocation algorithm, which provides a baseline for partitioning-based (iterative relocation) clustering methods is given in Algorithm 3.1 (see, [55],[37, pp.99-100] or [4, p.45]).

Algorithm 3.1. Iterative relocation algorithm

Input: The number of clusters K, and a database containing n objects from some $\mathbb{R}^p$.

Output: A set of K clusters, which minimizes a criterion function $\mathcal{J}$.

Step 1. Begin with an initial K centers/distributions as the initial solution.

Step 2. (Re)compute memberships for the data points using the current cluster centers.

Step 3. Update some/all cluster centers/distributions according to new membersips of the data points.

Step 4. Repeat from Step 2. until no change to $\mathcal{J}$ or no data points change cluster.

Using this framework, iterative methods compute the estimates for cluster centers, which are rather referred to as prototypes or centroids. The prototypes are meant to be the most representative points for the clusters. The mean and median are typical choices for the estimates. On the other hand, some methods, such as the EM-algorithm [15, 28], estimate a set of parameters that maximizes the likelihood of the chosen distribution model for a data. The best-known of the prototype-based algorithms are K-means and K-medoids, whereas the EM-algorithm is probably the most popular distribution-based algorithm [55]. The methods differ in the way they represent clusters, but they are mostly based on the same general algorithmic principle, which is given by Algorithm 3.1. K-means is discussed more thoroughly later in this work.

Three changeable elements compose the general relocation algorithm: 1) initialization, 2) reassignment of data points into clusters and 3) update of the cluster parameters. Although the heuristical, computational and statistical properties are defined through the realization of these elements, there are also other influencing factors, such as treatment of missing data values that must be considered in algorithm development.

3.1.1 Initialization

Due to the non-convex nature of criterion functions, the iterative relocation methods are often trapped into one of the local minima. Then the quality of a final clustering solution depends on the initial partition. A simple approach is to run the partition-based algorithms by starting from with several initial conditions. Another, more sophisticated way is to use some heuristic for finding an optimal initialization. In general, the initialization of the partition-based clustering algorithms is an important matter of interest (previous studies, e.g., [5, 95, 102, 16, 78, 80, 2, 59, 39, 116]).

3.1.2 Main iteration

The reassignment of data points and the update of prototypes (or parameters) construct the pass through data that improves the quality of the clustering. There are two main types of passes: *nearest centroid sorting pass* (a.k.a. K-means pass) and *hill-climbing pass* [4] or [5, p.160-162]. Let us further refer to the passes by NCS-pass and HC-pass, respectively.

The NCS-pass simply assigns data points to the cluster with the nearest prototype. Aldenfelder [4] divides the NCS-passes into *combinatorial* and *noncombinatorial* cases. In the former case, cluster centers are recomputed immediately after the reassignment of a data point (c.f. MacQueen's K-means and its variant [5]). In the latter case, the cluster centers are recomputed only after all data points are reassigned to the closest cluster centers (c.f. Forgy's K-means and Jancey's variant [41, 5]). The NCS-pass approach implicitly optimizes a particular statistical criterion (e.g., $tr(\mathbf{W})$ for K-means) by moving data points between the clusters based on distance to the cluster centers, whereas the HC-pass moves the points from a cluster to another only if the move optimizes the value of a statistical criterion.

3.1.3 Problem of unknown number of clusters

The name of a partitioning-based clustering method is usually of form K-"estimates" (sometimes, mostly in context of fuzzy clustering, also C-"estimates" is used, see [113, 10, 11, 123], and articles therein), which is due to the tendency to partition a data set into a fixed number K clusters. Another well-known class of clustering algorithms, namely hierarchical algorithms, produce a set of solutions with different numbers of clusters, which are then presented by a hierarchical graphical structure called dendrogram. See Figure 2. Although, the hierarchical methods provide some information about the number of clusters, they are not very feasible for data mining problems. First, quadratic memory requirement of the dissimilarity matrix is intractable for large data sets. Secondly, construction of the dissimilarity matrix is troublesome for incomplete data, because distances between data points lying in different subspaces are not directly comparable. This opens up another interesting problem: estimation of the correct number of clusters for partitioning-based algorithms. Several techniques are developed and tested for the estimation of the number of clusters, see e.g., [33, 30, 96, 5, 37, 43, 57, 82, 74, 119, 105, 50, 101, 114, 117]. A graphical technique based on the comparison of cluster tightness and separation was presented by Rousseeuw [107].

3.2 K-means clustering

Basically *K-means* is an iterative process that divides a given data set into K disjoint groups. K-means is perhaps the most widely used clustering principle, and especially, the best-known of the partitioning-based clustering methods that utilize prototypes for cluster presentation (a.k.a representative-based algorithm by Estivill-Castro [35]). Quality of K-means clustering is measured through the within-cluster squared error criterion (e.g., [5, p.165] or [58])

$$\min_{\mathbf{c}\in\mathbb{N}^n,\{\mathbf{m}_k\}_{i=1}^{K}\in\mathbb{R}^p} \mathcal{J}, \text{ for } \mathcal{J}(\mathbf{c},\{\mathbf{m}_k\}_{k=1}^{K},\{\mathbf{x}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{m}_{(\mathbf{c})_i}\|^2 \tag{3.1}$$

subject to

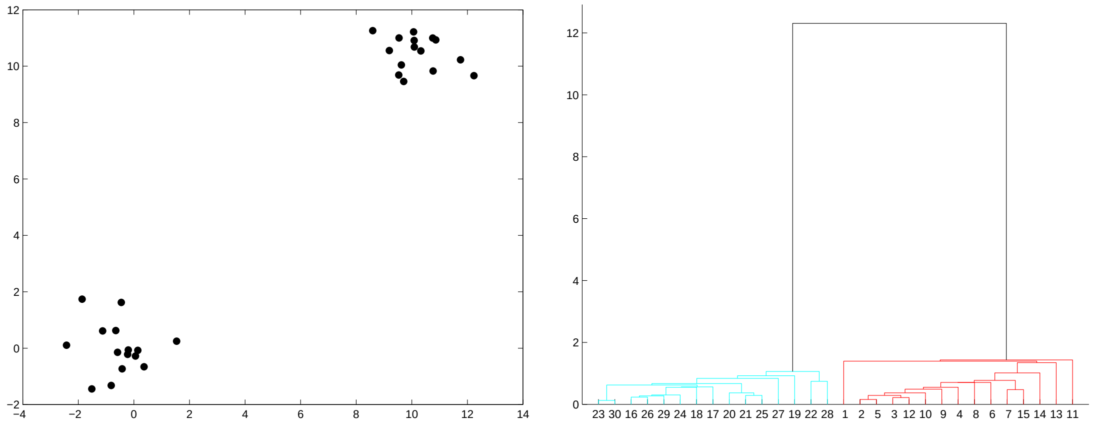

Figure 2: A sample ($n = 30$) from a two dimensional normal distribution $f(\mathbf{x}) = \frac{1}{2}N_2((0 \quad 0)^T, \mathbf{I}_2) + \frac{1}{2}N_2((10 \quad 10)^T, \mathbf{I}_2)$ in the left figure is clustered using the hierarchical single-linkage method. The result is visualized using a dendrogram tree.

$$(\mathbf{c})_i \in \{1, \ldots, K\} \quad \text{for all} \quad i \in \{1, \ldots, n\},$$

where $\mathbf{c}$ is a code vector for partition that contains the cluster membership for each object. $\mathbf{m}_{(\mathbf{c})_i}$ is the mean of the cluster, where the data point $\mathbf{x}_i$ is assigned. The sample mean leads to a unique minimum of the within-cluster variance, from which it follows that the problem actually corresponds to the minimization of $\sum_{i=1}^{K} \text{trace}(\mathbf{W}_i)$, where $\mathbf{W}_i$ is the within-group covariance matrix of the i^{th} cluster. Thus, the K-means clustering is also referred to as a variance minimization technique [79, p.112]. Actually in 1963, before the invention of any K-means algorithm, the minimum variance optimization technique was used by Ward [73], who proposed an hierarchical algorithm that begins with each data points as its own cluster and proceed by combining points that result in the minimum increase in the error sum of squares value (This method is later referred to both as the Ward's method, e.g., [37, 5] and the pairwise nearest neighbor algorithm (PNN), e.g., [62]).

As such, K-means clustering tends to produce compact clusters, but not take into account the between-cluster distances. The use of the squared l_2-norm makes the problem formulation extremely sensitive towards large errors, which means that the formulation is non-robust in statistical sense (see, e.g. [65]). However, due to its implementational simplicity and computational efficiency, K-means has remained its position as an extremely popular principle for many kind of cluster analysis tasks. It also requires less memory resources than, for instance, hierarchical methods, in which computation is often based on the dissimilarity matrix. By courtesy of its computational efficiency, K-means is also applied to initialization of other more expensive methods (e.g., EM-algorithm [14, 16]). The K-means algorithm, which is used to minimize the problem of K-means, has a large number of variants which are described next.

3.2.1 K-means algorithms

K-means type grouping has a long history. For instance, already in 1958, Fisher [40] investigated this problem in one-dimensional case as a *grouping problem*. At that time, algorithms and computer power were still insufficient for larger-scale problems, but the problem was shown to be interesting with concrete applications. Hence, more efficient procedures than exhaustive search was needed. The seminal versions of the K-means procedure were introduced in the Sixties by Forgy [41] (c.f. discussion in [92]) and MacQueen [92] (see also [5] and [12]). These are perhaps the most widely used versions of the K-means algorithms [79, p.112]. The main difference between Forgy's and MacQueen's algorithms is the order, in which the data points are assigned to the clusters and the cluster centers are updated. The MacQueen's K-means algorithm updates the "winning" cluster center immediately after every assignment of a data point and all cluster centers one more time after all data points have become assigned to the clusters. The Forgy's method updates the cluster centers only after all data points are assigned to the closest cluster centers. Moreover, another difference is that the Forgy's method iterates until converged while the MacQueen's basic algorithm performs only one complete pass through data. The starting points of the MacQueen's algorithm are often the first K data points in the data set.

In 1982, Lloyd [88] presented a quantization algorithms for pulse-code modulation (PCM) of analog signals. The algorithm is often referred to as *Lloyd's algorithm* and it is actually equivalent with the Forgy's K-means algorithm in a scalar case. Although Lloyd's paper was published not until 1982, the unpublished manuscript from 1957 is referred, for example, in articles from 1977 and 1980 by Chen [22] and Linde et al. [85], respectively[1]. A basically similar algorithm for multidimensional cases was presented by Chen in [22]. Linde et al. generalized the Lloyd's algorithm to a vector quantization algorithm [85]. This algorithm is often referred to as the *Generalized Lloyd's Algorithm* (GLA) in signal and image processing context. Hence, two main types of the K-means method have been discovered more than once on different disciplines.

The time complexity of the Forgy's K-means is $\mathcal{O}(npKt)$ (t is the number of iterations) [32]. A convergence proof for the algorithm is given by Selim et al. [110]. Because of the algorithmic details, the MacQueen's and Forgy's algorithms are also referred to as online- and batch-K-means algorithms, respectively (see, e.g., [14, 111]). One should note that many times, as in [14, 111], the convergent variant [5] of the MacQueen's K-means algorithm is behind the online clustering, although the MacQueen's basic K-means algorithm is referred. In [14], the numerical experiments suggest that the online K-means algorithm converges faster during the first few passes through the data and, thereafter, batch version outperforms it. However, the online clustering may be useful in real-time applications, which have to respond to inputs in extremely short time, or receive the data in a stream of unknown length,

[1] According to a author's note in [88], the manuscript of this article was written and circulated for comments at Bell Laboratories already in 1957.

or if there is not enough memory available to store a data set as a complete block [10].

3.2.2 Drawbacks

Despite the wide popularity of the ordinary K-means algorithms, there are some significant defects that have led to development of numerous alternative versions during the past years (see, e.g., [102, 16]):

- *Sensitivity to initial configuration.* Since the basic algorithms are local search heuristics and K-means cost function is non-convex, it is very sensitive to the initial configuration and the obtained partition is often only suboptimal (not the globally best partition).
- *Lack of robustness.* As the sample mean and variance are very sensitive estimate against outliers. So-called breakdown point is zero, which means that one gross errors may distort the estimate completely. The obvious consequent is that the k-means problem formulation is highly non-robust as well.
- *Unknown number of clusters.* Since the algorithm is a kind "flat" or "non-hierarchical" method [32], it does not provide any information about the number of clusters.
- *Empty clusters.* The Forgy's batch version may lead to empty clusters on unsuccessful initialization.
- *Order-dependency.* The MacQueen's basic and converging variants are sensitive to the order in which the points are relocated. This is not the case for the batch versions.
- *Only spherical clusters.* K-means presumes the symmetric Gaussian shape for cluster density functions. From this it follows that a large amount of clean data is usually needed for successful clustering.
- *Handling of nominal values.* The sample mean is not defined for nominal values.

In order to solve the previous problems many variants for the original versions have been developed.

3.2.3 Enhanced variants of K-means algorithm

It seems that the development of the clustering algorithms has been very intensive during the sixties. As we know, the rapid development of PC computer systems during the eighties and still growing data storages led to the invention of knowledge discovery and data mining concepts. It seems that this developed has led again to the growing interest in clustering algorithms. Hence, a lot of variants for the traditional K-means algorithms have emerged during the last ten years. Many of these try to solve the known drawbacks of the K-means procedures.

The general version of the iterative relocation algorithm (Algorithm 3.1) provides a lot of optional elements to be implemented in different ways, when building an iterative relocation algorithm for solving the problem of K-means. First, there are many ways to generate an initial partition or cluster prototypes for a data set. Secondly, there are many ways to arrange the relocation of the data points and update of the prototypes (for example, see [37, p.100]). The data points can be assigned to the nearest cluster or to the one that leads to the largest reduction in the value of the objective function. The cluster prototypes can be updated, either after every single reassignment of a data point, or alternatively after a fixed number of reassignments. Therefore, it is not necessarily surprising that the K-means clustering algorithm receives a somewhat many-sided treatment in the clustering literature. Although the differences among the variants were not always would not seem remarkable, the algorithms may well produce different final partitions for one data despite starting with the equal initial conditions.

Together with the basic K-means procedure, MacQueen [92] presented a ”coarsening-refining” variant that estimates also the correct number of clusters. It starts with a user specified number of clusters, and then coarsens and refines clustering according to input parameters during the process. After each assignment, all pairs of cluster means whose distance to each other is less than the coarsening parameter will be merged. On the other hand, every time a data point is processed in order to make an assignment to a cluster, its distance to the closest cluster mean is compared with the refining parameter. If the distance exceeds the parameter value, a new cluster must be created.

Another variant that also tries to determine the number of clusters is called ISODATA[2]. This is a quite elaborate algorithm for which several versions exist [5, pp.170–171]. The weakness of the method is the set of complex input parameters, e.g., the desired number of clusters, standard deviation parameter, lumping parameters etc. that are required from the users. If the user is looking at the data from data mining perspective, which means a minimal amount of prior assumptions and information, the use of this algorithm may prove to be complicated.

Jancey’s variant is a modification for the Forgy’s K-means method [5, p.161–162], which is expected to accelerate convergence and avoid inferior local minima. In this variant the new cluster center is not the mean of the old and added points, but the new center is updated by reflecting the old center through the mean of the new cluster.

In order to avoid poor suboptimal solutions, a number of different initialization methods for K-means(-type) methods have been developed and evaluated through numerical experiments (c.f., the references in 3.1.1). Zhang et al. [125] suggested to run so called *K-Harmonic Means* algorithm prior to K-means. They reported that in comparison to the K-means algorithm, K-Harmonic Means is more insensitive to initial configurations, but slower convergence near the solution. An accelerated, $k-d$-tree-based variant for the K-means clustering is proposed by Pelleg et al. [100]. The

[2] This is not the same procedure as the the one called Basic Isodata in [31, p.201], e.g., [5, 120]. Basic Isodata is actually the same as the Forgy’s K-means.

authors suggest this method for initialization of the ordinary K-means algorithm as well. As a problem the authors report the scalability with respect to the number of dimensions, which is due to the use of the kd-tree structure. One of the most interesting approaches for avoiding poor quality minima in clustering problems, is *LBG-U method*, which is presented as a vector quantization method [44]. Since the LBG-algorithm [85] is equivalent with the Forgy's K-means clustering algorithm, LBG-U method can also be used as a clustering method. The idea of the LBG-U is to repeat the LBG-algorithm until convergence. After each run, the cluster center with the minimum utility is moved to a new point. The mean vector that possesses the minimum utility is the one that contributes least to the overall sum of squared errors when removed. The new point is chosen close to the mean of the cluster that generates most of the distortion error for clustering. LBG-U is good from data mining point of view, because it does not require extra parameters for any operation when compared to the basic K-means algorithms. It also converges since the algorithm will be terminated if the last run do not produce reduction to the value of the error function.

The increase of the computer power has enabled also the use of more intensive methods for solving the drawbacks of the K-means-type algorithms. In order to avoid poor local solutions, a number of genetic algorithm based methods have been developed [83, 9, 91]. Likas et al. propose *the global k-means* clustering algorithm [84], which is a deterministic and incremental global optimization method. It is also independent on any initial parameters and employs K-means procedure as a local search procedure. Since the exhaustive global K-means method is computationally relatively expensive, a faster variant of the algorithm and a $k-d$-tree-based initialization method are also given in the paper in order to reduce the computational cost.

As the requirements for dealing with very large data sets that are often even too large to be loaded to RAM[3] have been constantly growing, the scalability of the K-means algorithms have become an important issue. A scalable single-pass version of the K-means algorithm, which is based on identification of data that can be compressed, the region of data that must be maintained and regions that can be discarded, is introduced in [18]. An enhanced version of the previous is proposed in [38]. These methods can be used efficiently for searching multiple solutions using different initial conditions, since the information about the compressible regions, retained and discarded data can be reused. An efficient "disk-based" algorithm, *Relational K-means*, for clustering large high-dimensional data sets inside a relational database is given in [99]. Disk-based refers to efficient organization of data matrices to a hard disk drive.

A number of methods for estimation of K, the correct number of clusters, have been developed and experimented for partition-based methods in general. It is not so much algorithm-specific issue, but a more general problem covering all partition-based methods that are based on solving the clustering problems for a specific K. A common approach is to use some validity measure to evaluate the goodness of the

[3] RAM (random access memory) is a temporary fast access memory of a computer system.

obtained solution (see references in Section 3.1.3).

The problem of empty clusters may occur with the Forgy's batch-type algorithms. One cluster center may become empty when all points are closer to the other centers. Consequently the empty cluster is never updated during the process thereafter. The risk of empty clusters exists also for many other batch-type partitioning-based clustering methods, but not for the MacQueen's type single pass algorithms. However, on large data sets with a small number of clusters, the probability of empty clusters is quite small. The problem is worse, for example, in sub-sampling methods due to the reduced number of data points. In order to avoid this problem in a sub-sampling-based initialization method, Bradley et al. [16] introduce a variant for the K-means algorithm, which is called KMeansMod. KMeansMod repeats the K-means process until all K clusters are non-empty. The empty clusters are re-initialized with the worst fitting points.

The order-dependency problem concerns mainly MacQueen's type sequential K-means algorithms that perform the assignment and recomputation steps for one point at a time. It is not a problem for the batch-type K-means algorithms, because all points are processed at once. The tendency to spherical-shaped clusters depends on the problem setting. In prototype-based clustering we are usually interested in finding such clusters, in which points are close to each other and presented by a one central/representative point. Any connected and arbitrary shaped cluster may not necessarily be of interest. If the cohesiveness of cluster areas is the only criterion for cluster structure, points in the same cluster can be anyhow very distant to each other. For example, some of the hierarchical (e.g., single-link) and density-based methods are biased to connected, possibly arbitrary shape, cluster areas.

Extensions of the K-means-type algorithms to deal with mixed and categorical data types are represented in [103, 63, 64, 49, 109]. Algorithms for clustering binary data streams are represented in [98].

The lack of robustness is due to the sensitive squared l_2-norm in the cluster center estimation, that is, the sample mean is chosen as the location estimator of each cluster center, see Equation 3.1. The sample mean is straightforward and fast to compute (closed-form solution exists), but at the same time, it is extremely sensitive to any gross errors, which inevitably exist in real-world data sets. This questions the usability of K-means on noisy and incomplete data sets. A solution to this problem is discussed more thoroughly in this report.

4 Robust clustering

A problem of many classical clustering methods is their sensitivity to erroneous and missing values that unfortunately exist in real-world data sets. For example, in the K-means methods, each cluster center is represented by the sample mean. The sample mean is extremely sensitive toward outlying values. Because the sample mean suffers from the lack of robustness, only a few erroneous or missing data values may distort the estimates so completely that the inherent cluster structure of the data set will not be uncovered. By robustness we mean that the estimator tolerates small deviations from the assumed model and avoids "catastrophes" in the case of larger deviations [65]. Robustness and efficiency are contradicting goals for statistical estimators (see, e.g., in [65, 124, 24]). The relative efficiency says which one of two consistent estimators gives correct values in probability.

Definition 4.1. Let $\{x_1, \ldots, x_n\}$ be a sample of a random variable X and θ an interesting parameter. $T_1(x_1, \ldots, x_n)$ and $T_2(x_1, \ldots, x_n)$ are estimators of θ. *Relative efficiency* of T_1 and T_2 is defined by ratio

$$eff_R(T_1, T_2) = \frac{\mathcal{E}[(T_1(x_1, \ldots, x_n) - \theta)^2]}{\mathcal{E}[(T_2(x_1, \ldots, x_n) - \theta)^2]},$$

where $\mathcal{E}$ is the expected value.

Hence, the smaller the variance of an estimator, the more efficient the estimator. On the other hand, as the robustness against deviations gets higher, the variance of the estimator tends to grow. For example, in comparison to the sample mean the relative efficiency of the trimmed sample mean estimator is poor, because the tails of a sample are ignored in the computation. The trimming leads to the loss of data, which again causes larger variance to the estimates and, thereby, higher risk for inaccurate results. The estimators possessing the highest practical breakdown point value (50%) will not be completely "broken down" even though nearly half of the data were contaminated.

Robust partitioning-based algorithms are, for example, K-medoids [58], PAM, CLARA [79] and CLARANS [97]. These are called as medoids-based clustering algorithms, since the prototypes are restricted to be chosen among the data points. K-medoids-based algorithms are more insensitive to outliers and noise when compared to K-means, but also computationally more expensive. In order to reduce the computational load of the basic PAM algorithm on large data sets, enhanced algorithms CLARA and CLARANS are proposed. CLASA [23] is another variant, which exploits the simulated annealing method for finding better cluster medoids. The K-medoids algorithms are invariant to translations and orthogonal transformations of objects, but not invariant to affine transformations that change the inter-object distances [79, p.119]. K clusters are always found, so that the problem of empty clusters is avoided. The sensitive estimate of the K-means procedures can be substituted also with other robust estimates, such as the coordinate-wise median [5, p.166]) or the spatial median [75, 36]. In [19, 17], Bradley et al. reformulate the K-coordinate-wise medians problem into a bilinear programming form. The last two variants do not restrict the prototypes to be chosen among the data points. As a coordinate-wise order-statistic, the coordinate-wise median together with l_1-distance ('city-block'-distance) in the assignment step, is inherently more appropriate for discrete (e.g., questionnaire data) than continuous data. The spatial median is more appropriate for continuous and high-dimensional problems in $\mathbb{R}^p$, since its statistical efficiency improves as the number of dimensions grows [20]. We have also developed a computationally efficient algorithm that scales well with respect to the number of dimensions [76]. Later, we will present a spatial median clustering algorithm with missing data treatment. Robust clustering algorithms are proposed, e.g., [46, 112, 6, 68]. Robust methods for fuzzy clustering are presented in [27]. During the last years, robust clustering techniques have also been integrated into some statistical software products, such as S-PLUS [115].

4.1 Measure of robustness

Robustness measures the sensitivity of an estimators to any observation in a data set. Let $\mathbf{X}$ represent a data set of size $n \times p$. Then functional $T = T(\mathbf{X})$ is any arbitrary estimator for an interesting parameter on a data set $\mathbf{X}$. The most common measure of robustness is the breakdown point, which defines the smallest fraction of contamination that can cause the estimator T to take values arbitrarily far from $T(X)$ [108]. A formal definition for the finite-sample breakdown point is given by

$$\varepsilon_n^*(T, \mathbf{X}) = \min\{\frac{m}{n} : b_{\max}(m, T, \mathbf{X}) = \infty\}. \tag{4.1}$$

$b_{\max}(m, T, \mathbf{X})$ defines the maximum bias caused by contamination as follows

$$b_{\max}(m, T, \mathbf{X}) = \sup_{\mathbf{X}'} \|T(\mathbf{X}') - T(\mathbf{X})\|, \tag{4.2}$$

where the supremum is over all possible $\mathbf{X}'$s. Here $\mathbf{X}'$ denotes all contaminated samples obtained by replacing m data points by arbitrary values (this allows also extremely distant outliers) [65, 108]. This definition of the breakdown point is independent on any probability distribution. The less robust an estimator is, the closer to zero its breakdown point gets (e.g., for the sample mean $\varepsilon_n^* = 1/(n+1)$) [66]. On the other hand, the highest possible breakdown point for any translation equivariant estimator is $\frac{1}{2}$ (cf. Theorem 4.1). The upper limit is clear due to the obvious truth that if more than half of the data is contaminated, it is impossible to decide which part of data is good and which part is bad.

Theorem 4.1. (Lopuhaä et al. [89]). Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a collection of n points in $\mathbb{R}^p$. When T_n is translation equivariant, then $\varepsilon^*(T_n, \mathbf{X}) \leq \lfloor (n+1)/2 \rfloor / n$, where $\lfloor u \rfloor$ denotes the nearest integer less than or equal to u.

As we are interested to diminish the effect of extremely outlying points, the breakdown point of the estimator is our main interest. The coordinate-wise sample median and the spatial median are estimators with a bounded influence function and high breakdown value (50%), which makes them insensitive to a finite amount of gross errors. Alternative tools for measuring robustness are influence curves, sensitivity curves, local shift-sensitivity and rejection point [126, 52, 61, 65, 53, 51].

4.2 Spatial median

The spatial median is one of the multivariate generalizations for the univariate (population or sample) median. In univariate case, the (marginal) median is known as a robust multivariate estimate of location, which possess the highest possible breakdown point (50%). In geometric sense, the spatial median can be defined as a point of Euclidean space $\mathbb{R}^p$, from which the sum of absolute distances to a given set of n points attains its minimum value. Thus, it also provides the solution for a well-known facility location problem, which gains a lot of interest among the location science and operations research communities (e.g., [90]). A typical problem in this context is, for example, finding the optimal location of a warehouse with respect to a set of retail stores. The spatial median is referred to by several names, such as, multivariate L_1-estimator, multivariate L_1-median, Weber point, or Fermat-Weber point [89, 108, 26].

The spatial median is one special case of the M-estimators [65].

Definition 4.2. Any estimate T_n, defined by a minimum problem of the form

$$T_n = \arg\min_{T_n} \sum_{i=1}^{n} \rho(\mathbf{x}_i; T_n), \tag{4.3}$$

or by an implicit equation

$$\sum_{i=1}^{n} \psi(\mathbf{x}_i; T_n) = 0, \tag{4.4}$$

where ψ is an arbitrary function, $\psi(\mathbf{x}, \theta) = (\partial/\partial\theta)\rho(\mathbf{x}, T_n)$, is said to be an *M-estimate* (or a maximum likelihood type estimate).

The location M-estimator is defined as follows:

$$T_n = \arg\min_{T_n} \sum_{i=1}^{n} \rho(\mathbf{x}_i - T_n), \tag{4.5}$$

or

$$\sum_{i=1}^{n} \psi(x_i - T_n) = 0. \tag{4.6}$$

By altering ρ a large assortment of location M-estimators can be constructed. Usually ρ is strictly convex so that ψ becomes strictly monotone and correspondingly T_n unique.

Hence, the problem of the spatial median is given by

$$\min_{\mathbf{u}\in\mathbb{R}^p} \mathcal{J}_2^1(\mathbf{u}), \quad \text{for } \mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{n} \|\mathbf{u} - \mathbf{x}_i\|_2, \tag{4.7}$$

which clearly satisfies the conditions of definition (4.3).

The gradient of the convex cost function $\mathbf{f}(\mathbf{u}, \mathbf{x}_i) = \|\mathbf{u} - \mathbf{x}_i\|_2$ is well-defined and unique for all $\mathbf{u} \neq \mathbf{x}_i$. However, case $\mathbf{u} = \mathbf{x}_i$ leads to the use of a sub-gradient, which is characterized by condition $\|\xi\|_2 \leq 1$ (cf. inliers in statistics [21]). Thus the (local) extremity of (4.7) is characterized by means of a sub-gradient, which reads as

$$\partial\mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{n} \xi_i, \quad \text{with} \begin{cases} (\xi_i)_j = \frac{(\mathbf{u}-\mathbf{x}_i)_j}{\|\mathbf{u}-\mathbf{x}_i\|_2}, & \text{for } \|\mathbf{u} - \mathbf{x}_i\|_2 \neq 0, \\ \|\xi_i\|_2 \leq 1, & \text{for } \|\mathbf{u} - \mathbf{x}_i\|_2 = 0. \end{cases} \tag{4.8}$$

As pointed out in [77] problem (4.7) is a non-smooth optimization problem [94], which means that it can not be described by using the classical (C^1) differential calculus. In a recent paper by the authors of this report, a reformulated problem and an efficient iterative SOR-based algorithm are presented [76]. The algorithm is easily generalized to missing data.

The spatial median is orthogonally equivariant, which makes it insensitive to all orthogonal transformations, such as rotation of a data set, that preserve the Euclidean distances between data points. The breakdown point (50%) of the estimator is independent on the number of dimensions (see Theorem 4.2 and Figures 3 and 4). In univariate case the spatial median corresponds to the coordinate-wise sample median.

Theorem 4.2. (Lopuhaä et al. [89]). Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$ be a random sample. Then the L_1 estimator has breakdown point $\varepsilon^*(T_n, \mathbf{X}) = \lfloor (n+1)/2 \rfloor / n$.

The asymptotic normality of the spatial median is shown by Brown [20]. The normal efficiency in the univariate case is ($\frac{2}{\pi} = 0.637$), which is reasonable as the estimator is equivalent with the coordinate-wise sample median in this case. Brown shows that the efficiency improves as the number of dimensions grows. The relative normal efficiency equals with the sample mean as the number of dimensions goes to infinity. Hence, it is justified to characterize the spatial median as an inherently multivariate estimate of location.

In order to make a choice between the coordinate-wise and spatial median, we can compare their influence curves. A continuous influence function provides safety against inliers, which means that not any single observation can fully determine the value of an estimator. The influence function of the coordinate-wise sample median is not continuous at zero, from which it follows that the estimate is very sensitive to one or two middle values of a data set. This leads to extreme sensitivity to rounding or grouping of these points [21]. In other words, the local-shift sensitivity of the coordinate-wise median is asymptotically infinite at the central part of the influence curve. The robustness of the spatial median improves against such inliers as the dimension of the problem space grows. Unlike the coordinate-wise median, the spatial median lies always inside the convex hull of a sample as well.

5 A robust clustering algorithm with treatment for missing data

A robust partitioning-based clustering method is proposed by Estivill-Castro and Yang [36]. The method computes the spatial median when estimating the most representative point of a cluster. It is a robust variant of the K-means method. A similar method with different algorithmic details and treatment for missing data is derived by Kärkkäinen and Äyrämö in [75].

A convenient way to indicate the available data is to define a projector matrix,

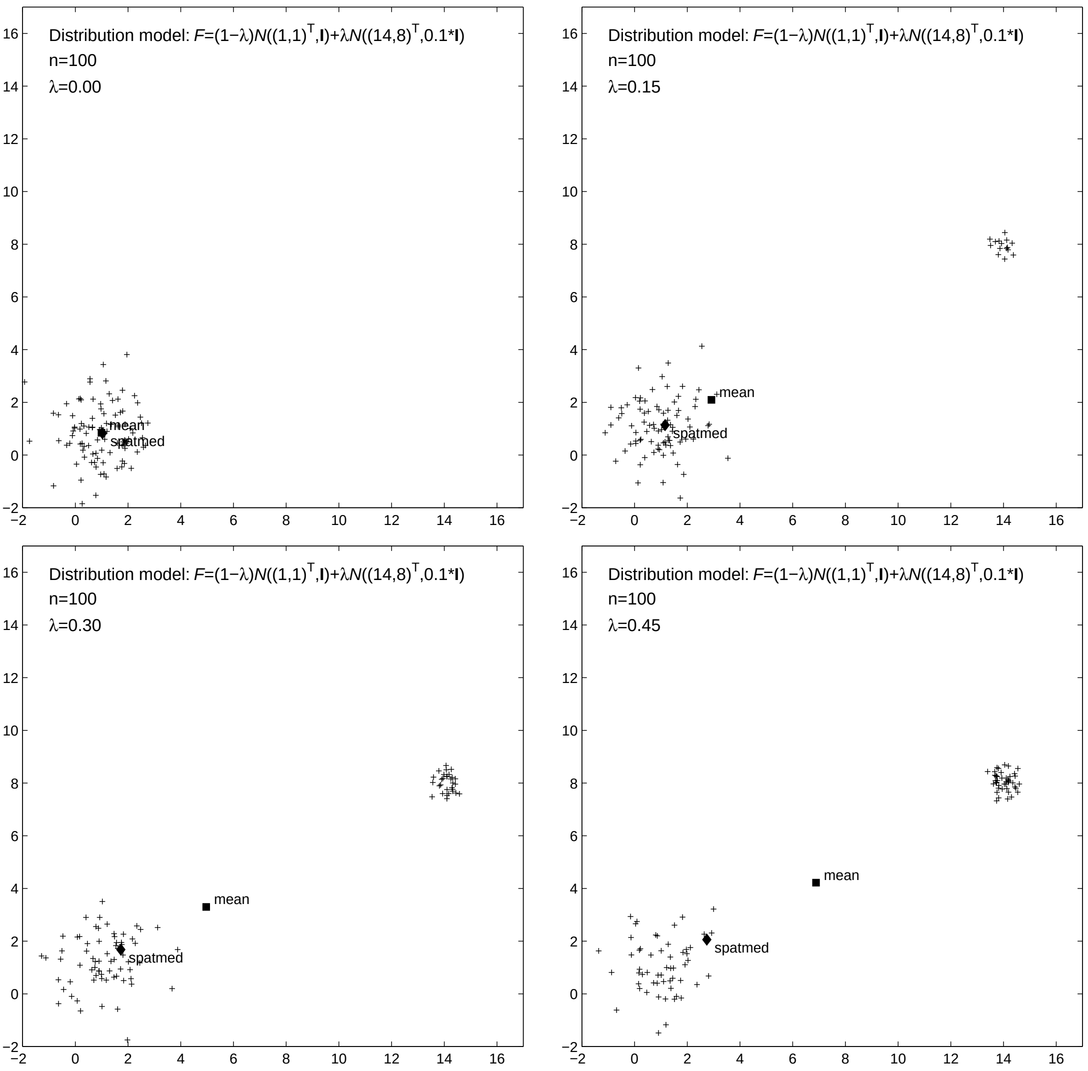

Figure 3: Illustration about the sensitive of the sample mean towards the contaminated part of the data. One can see that larger the proportion (λ) of the outliers, more distorted the sample mean becomes whereas the spatial median remains relatively stable.

which separates the missing and existing values:

$$(\mathbf{p}_i)_j = \begin{cases} 1, & \text{if } (\mathbf{x}_i)_j \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

By further denoting $\mathbf{P}_i = \mathrm{Diag}\{\mathbf{p}_i\}$ we can, for example, redefine the problem of the spatial median given in (4.7):

$$\min_{\mathbf{u}\in\mathbb{R}^p} \mathcal{J}_2^1(\mathbf{u}), \quad \text{for } \mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{n} \|\mathbf{P}_i(\mathbf{u}-\mathbf{x}_i)\|_2, \tag{5.1}$$

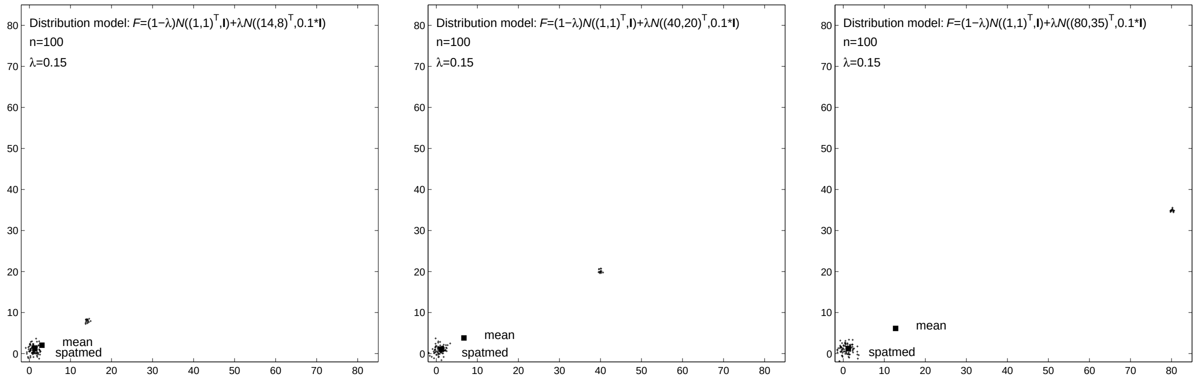

Figure 4: The remoteness of the bunch of outliers (15% in this case) does not influence the spatial median whereas the sample mean becomes significantly broken down.

Exactly similar projector technique can be applied to distance calculation.

The algorithm follows the principles of the general iterative relocation algorithm 3.1 in Section 3.1.

Algorithm 5.1. K-spatialmedians algorithm

Required input parameters: Data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$, the number of clusters K and the maximum number of iterations maxit.

Optional input parameters: Initial cluster centers $\{\mathbf{m}_k^0\}_{k=1}^K$ or $n \times 1$ code vector $\mathbf{c}^0 \in \{1, \ldots, K\}$, which defines an initial partition.

Output parameters: Final cluster centers $\{\mathbf{m}_k^*\}_{k=1}^K$ and code vector (partition) $\mathbf{c}^*$.

Step 1.(*Initialization.*) If centers $\{\mathbf{m}_k^0\}_{k=1}^K$ are given then go to Step 2. Else if an initial partition $\mathbf{c}^0$ is given then go to Step 3. If neither centers nor partition is given as input then initialize centers $\{\mathbf{m}_k^0\}_{k=1}^K$, assign each data point $\{\mathbf{x}_i\}_{i=1}^n$ to the closest center and go to Step 3. Set $t = 0$.

Step 2.(*Reassignment.*) Reassign each data point in $\{\mathbf{x}_i\}_{i=1}^n$ to a new cluster $\mathcal{C}_r$ if

$$\min_{r \neq (\mathbf{c}^t)_i} \|\mathbf{x}_i - \mathbf{m}_r^t\| < \|\mathbf{x}_i - \mathbf{m}_{(\mathbf{c}^t)_i}^t\|.$$

Update $(\mathbf{c}^t)_i$ for all relocated points.

Step 3.(*Recomputation.*) Update the cluster centers of the current partition by minimizing (5.1) for all $k \in \{1, \ldots, K\}$.

Step 4.(*Stopping.*) If no reassignments of data points between cluster centers then stop. Otherwise $t = t + 1$ repeat from step 2.

Because we are mainly calculating in continuous space, an obvious choice of the dissimilarity measure is the Euclidean distance. In the case of discrete data, the coordinate-wise median for the prototype estimation and l_1-distance for distance are more appropriate.

In Step 3. problem (4.7) is solved, for example, using SOR-based algorithm given in [76]. In order to deal with incomplete data sets, a strategy for the missing data treatment must be included in the algorithms. Since we do not want to be involved in making hypotheses on the distributions of unknown cluster-wise data, we chose to apply a strategy, which employs only available data values in the calculation of the distances and estimates [86]. From this it follows that all computations are restricted to the existing fields of the original data.

5.1 Comparison of the new robust method to K-means

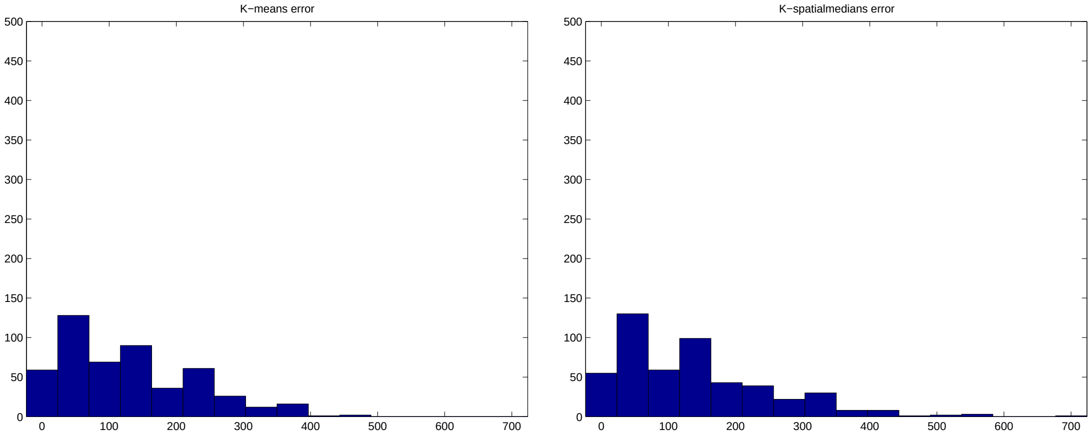

Figure 5: This figure shows the error distributions for K-Means (left) and K-spatialmedians (right) methods when they were used to cluster the data represented in the right plot of Figure 1. There are no missing values in the data set.

In this section some comparison on the behavior of the Forgy's K-means and the new K-spatialmedians clustering methods are presented. As a test data, we use the bivariate data sets represented in Figure 1. The clusters in the data sets are generated from normal and laplace distribution.

Figure 6 shows the four outliers that are added to the data by randomly disturbing four observations. One can see that it is difficult to figure out the seven distinct clusters since the view is more distant from the data. Error distributions in Figure 7 show that the performance of K-means is significantly reduced by disturbing only one percent of the data whereas the spatial median clustering produce relatively good clusterings.

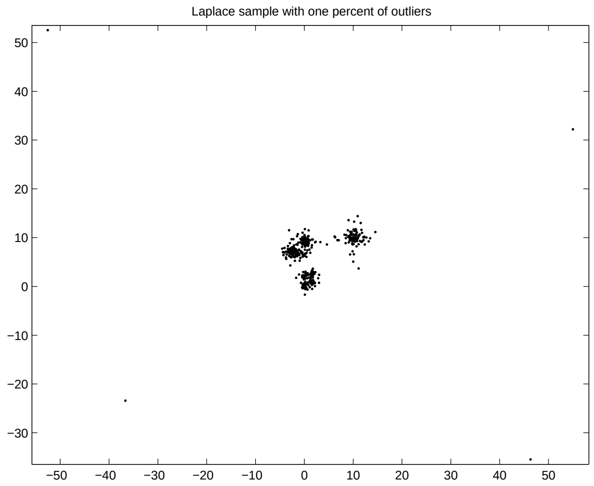

Figure 6: The data represented in the right plot of Figure 1 after disturbing one percent of the observations. There are no missing values in the data set.

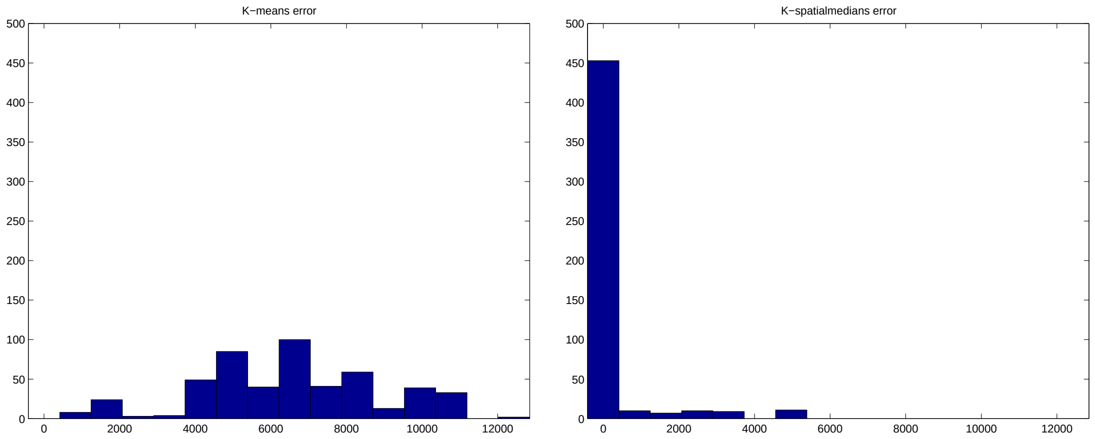

Figure 7: This figure shows the error distributions for K-Means (left) and K-spatialmedians (right) methods when they were used to cluster the data represented in the right plot of Figure 1 after disturbing one percent of the observations. There are no missing values in the data set.

6 Discussion

In this report we have given an introduction to the cluster analysis and reviewed partition-based clustering algorithms. A new robust algorithm is introduced as an example. An important field of data mining applications is the data mining context. As the data mining concentrates on large real-world data sets, missing and erroneous values are often encountered. Hence, we need automated clustering methods that compress the data into prototypes with minimal user inputs. This diminishes the need of statistical and computational skill of the end users. The algorithmic details that we have chosen for treatment of missing data and robust estimation are a step toward automated clustering procedures. The approach relieves the end users from most of the priori manipulations of the data sets, for example, estimation of the missing values or removing outliers. The algorithms (iterative clustering and SOR-based prototype estimation) are also fast enough to solve the problems on large data sets as well. As future work, we should find a way to obtain the globally optimal solutions. This is mainly an initialization problem. Moreover, a heuristic for estimation of the natural number of clusters may be useful. Although, as it was shown in this report, there is not always the correct number of clusters. The number of clusters depends also on the way one looking at the data. In order to efficiently utilize the results, visualization methods are needed as well. A review of the multidimensional visualization methods is given in [67].

References

[1] B. Abolhassani, J. Salt, and D. Dodds, *A two-phase genetic k-means algorithm for placement of radioports in cellular networks*, IEEE Transactions on Systems, Man and Cybernetics, Part B, 34 (2004), pp. 533–538.

[2] M. B. Al-Daoud and S. A. Roberts, *New methods for the initialisation of clusters*, Pattern Recognition Letters, 17 (1996), pp. 451–455.

[3] M. S. Aldenderfer, *Methods of cluster validation for archaeology*, World Archaeology, 14 (1982), pp. 61–72.

[4] M. S. Aldenderfer and R. K. Blashfield, *Cluster analysis*, Sage Publications, London, England, 1984.

[5] M. R. Anderberg, *Cluster analysis for applications*, Academic Press, Inc., London, 1973.

[6] L. Ángel García-Escudero and A. Gordaliza, *Robustness properties of k means and trimmed k means*, Journal of the American Statistical Association, 94 (1999), pp. 956–969.

[7] F. Bachmann and L. Bass, *Managing variability in software architectures*, in SSR '01: Proceedings of the 2001 symposium on Software reusability, New York, USA, 2001, ACM Press, pp. 126–132.

[8] K. D. Bailey, *Cluster analysis*, Sociological Methodology, 6 (1975), pp. 59–128.

[9] S. Bandyopadhyay and U. Maulik, *An evolutionary technique based on k-means algorithm for optimal clustering in $\mathbb{R}^N$*, Information Sciences, 146 (2002), pp. 221–237.

[10] A. BARALDI AND P. BLONDA, *A survey of fuzzy clustering algorithms for pattern recognition I*, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 29 (1999), pp. 778–785.

[11] ——, *A survey of fuzzy clustering algorithms for pattern recognition II*, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 29 (1999), pp. 786–801.

[12] P. BERKHIN, *Survey of clustering data mining techniques*, tech. report, Accrue Software, San Jose, CA, 2002.

[13] M. J. BERRY AND G. S. LINOFF, *Mastering data mining: The art and science of customer relationship management*, John Wiley & Sons, Inc., 2000.

[14] L. BOTTOU AND Y. BENGIO, *Convergence properties of the K-means algorithms*, in Advances in Neural Information Processing Systems, G. Tesauro, D. Touretzky, and T. Leen, eds., vol. 7, The MIT Press, 1995, pp. 585–592.

[15] P. BRADLEY, C. REINA, AND U. FAYYAD, *Clustering very large databases using EM mixture models*, in Proceedings of 15th International Conference on Pattern Recognition (ICPR'00), vol. 2, 2000, pp. 76–80.

[16] P. S. BRADLEY AND U. M. FAYYAD, *Refining initial points for K-Means clustering*, in Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1998, pp. 91–99.

[17] P. S. BRADLEY, U. M. FAYYAD, AND O. L. MANGASARIAN, *Mathematical programming for data mining: formulations and challenges*, INFORMS Journal on Computing, 11(3) (1999), pp. 217–238.

[18] P. S. BRADLEY, U. M. FAYYAD, AND C. REINA, *Scaling clustering algorithms to large databases*, in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, R. Agrawal and P. Stolorz, eds., AAAI Press, 1998, pp. 9–15.

[19] P. S. BRADLEY, O. L. MANGASARIAN, AND W. N. STREET, *Clustering via concave minimization*, in Advances in Neural Information Processing Systems, M. C. Mozer, M. I. Jordan, and T. Petsche, eds., vol. 9, The MIT Press, 1997, p. 368.

[20] B. M. BROWN, *Statistical uses of the spatial median*, J. Roy. Statist. Soc. Ser. B, 45 (1983), pp. 25–30.

[21] B. M. BROWN, P. HALL, AND G. A. YOUNG, *On the effect of inliers on the spatial median*, Journal of Multivariate Analysis, 63 (1997), pp. 88–104.

[22] D. CHEN, *On two or more dimensional optimum quantizers*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '77., vol. 2, Telecommunication Training Institute, Taiwan, Republic of China, May 1977, pp. 640–643.

[23] S.-C. CHU, J. F. RODDICK, AND J. S. PAN, *A comparative study and extension to K-medoids algorithms*, in Proc. 5th International Conference on Optimization : Techniques and Applications (ICOTA 2001), Hong Kong, December 2001.

[24] C. W. COAKLEY AND T. P. HETTMANSPERGER, *A bounded influence, high breakdown, efficient regression estimator*, J. Amer. Statist. Assoc., 88 (1993), pp. 872–880.

[25] R. M. CORMACK, *A review of classification*, Journal of the Royal Statistical Society. Series A (General), 134 (1971), pp. 321–367.

[26] C. CROUX AND A. RUIZ-GAZEN, *High breakdown estimators for principal components: the projection-pursuit approach revisited*, Journal of Multivariate Analysis, 95 (2005), pp. 206–226.

[27] R. N. DAVÉ AND R. KRISHNAPURAM, *Robust clustering methods: A unified view*, IEEE Transactions on Fuzzy Systems, 5 (1997), pp. 270–293.

[28] A. DEMPSTER, N. LAIRD, AND D. RUBIN, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society. Series B (Methodological), 39 (1977), pp. 1–38.

[29] W. R. DILLON AND M. GOLDSTEIN, *Multivariate analysis : methods and applications*, Wiley series in probability and mathematical statistics, Applied probability and statistics, Wiley, New York, 1984.

[30] R. C. DUBES, *How many clusters are best? - an experiment.*, Pattern Recognition, 20 (1987), pp. 645–663.

[31] R. DUDA AND P. HART, *Pattern Classification and Scene analysis*, John Wiley & Sons, Inc., NY, 1973.

[32] R. O. DUDA, P. E. HART, AND D. G. STORK, *Pattern classification*, John Wiley & Sons, Inc., 2001.

[33] S. DUDOIT AND J. FRIDLYAND, *A prediction-based resampling method for estimating the number of clusters in a dataset*, Genome Biology, 3 (2002), pp. research0036.1–research0036.21.

[34] M. H. DUNHAM, *Data mining introductory and advanced topics*, Pearson Education Inc, Upper Saddle River, New Jersey, USA, 2003.

[35] V. ESTIVILL-CASTRO, *Why so many clustering algorithms: A position paper*, SIGKDD Explorations Newsletter, 4 (2002), pp. 65–75.

[36] V. ESTIVILL-CASTRO AND J. YANG, *Fast and robust general purpose clustering algorithms*, Data Mining and Knowledge Discovery, 8 (2004), pp. 127–150.

[37] B. S. Everitt, S. Landau, and M. Leese, *Cluster analysis*, Arnolds, a member of the Hodder Headline Group, 2001.

[38] F. Farnstrom, J. Lewis, and C. Elkan, *Scalability for clustering algorithms revisited*, SIGKDD Explor. Newsl., 2 (2000), pp. 51–57.

[39] U. M. Fayyad, C. Reina, and P. S. Bradley, *Initialization of iterative refinement clustering algorithms*, in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD98), AAAI Press, 1998, pp. 194–198.

[40] W. D. Fisher, *On grouping for maximum homogeneity*, Journal of the American Statistical Association, 53 (1958), pp. 789–798.

[41] E. Forgy, *Cluster analysis of multivariate data: Efficiency versus interpretability of classifications*, Biometrics, 21 (1965), pp. 768–769. Abstracts in Biometrics.

[42] C. Fraley and A. Raftery, *Model-based clustering, discriminant analysis, and density estimation*, Journal of the American Statistical Association, 97 (2002), pp. 611–631.

[43] J. Fridlyand and S. Dudoit, *Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method*, Technical report 600, Department of Statistics, University of California, Berkeley, September 2001.

[44] B. Fritzke, *The LBG-U method for vector quantization an improvement over LBG inspired from neural networks*, Neural Processing Letters, 5 (1997), pp. 35–45.

[45] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc, 1972.

[46] M. T. Gallegos and G. Ritter, *A robust method for cluster analysis*, The Annals of Statistics, 33 (2005), pp. 347–380.

[47] J. Ghosh, *Scalable clustering methods for data mining*, Lawrence Ealbaum Associates, Inc., Publishers, Mahwah, New Jersey, USA, 2003, ch. 10, pp. 247–277.

[48] J. Grabmeier and A. Rudolph, *Techniques of cluster algorithms in data mining*, Data mining and knowledge discovery, 6 (2002), pp. 303–360.

[49] S. K. Gupta, K. S. Rao, and V. Bhatnagar, *K-means clustering algorithm for categorical attributes*, in DaWaK '99: Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, London, UK, 1999, Springer-Verlag, pp. 203–208.

[50] G. Hamerly and C. Elkan, *Learning the k in k-means*, in Advances in Neural Information Processing Systems 16, S. Thrun, L. Saul, and B. Schölkopf, eds., MIT Press, Cambridge, MA, 2004.

[51] F. R. HAMPEL, *The influence curve and its role in robust estimation*, Journal of the American Statistical Association, 69 (1974), pp. 383–393.

[52] F. R. HAMPEL, E. M. RONCHETTI, P. J. ROUSSEEUW, AND W. A. STAHEL, *Robust statistics: The approach based on influence functions*, John Wiley & Sons, 1986.

[53] F. R. HAMPEL, P. J. ROUSSEEUW, AND E. RONCHETTI, *The change-of-variance curve and optimal redescending M-estimators*, J. Amer. Statist. Assoc., 76 (1981), pp. 643–648.

[54] J. HAN AND M. KAMBER, *Data mining: concepts and techniques*, Morgan Kaufmann Publishers, Inc., 2001.

[55] J. HAN, M. KAMBER, AND A. K. H. TUNG, *Spatial Clustering Methods in Data Mining: A Survey*, Taylor and Francis, 1 ed., December 2001, ch. 8, pp. 188–217.

[56] D. HAND, H. MANNILA, AND P. SMYTH, *Principles of Data Mining*, MIT Press, 2001.

[57] A. HARDY, *On the number of clusters*, Computational Statistics and Data Analysis, 23 (1996), pp. 83–96.

[58] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The elements of statistical learning: Data mining, inference and prediction*, Springer-Verlag, 2001.

[59] J. HE, M. LAN, C.-L. TAN, S.-Y. SUNG, AND H.-B. LOW, *Initialization of cluster refinement algorithms: A review and comparative study*, in Proceedings of International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, July 2004.

[60] E. HELMES AND J. LANDMARK, *Subtypes of schizophrenia: A cluster analytic approach*, The Canadian Journal of Psychiatry, 48 (2003), pp. 702–708.

[61] D. C. HOAGLIN, F. MOSTELLER, AND J. W. TUKEY, *Understanding robust and exploratory data analysis*, John Wiley & Sons, Inc., 1983.

[62] C.-M. HUANG AND R. HARRIS, *A comparison of several vector quantization codebook generation approaches*, IEEE Transactions on Image Processing, 2 (1993), pp. 108–112.

[63] Z. HUANG, *A fast clustering algorithm to cluster very large categorical data sets in data mining*, in DMKD'97 Pre-Conf. Data Mining Workshop: Research Issues on Data Mining and Knowledge Discovery, 1997.

[64] ——, *Extensions to the k-means algorithm for clustering large data sets with categorical values.*, Data Mining and Knowledge Discovery, 2 (1998), pp. 283–304.

[65] P. HUBER, *Robust statistics*, John Wiley & Sons, 1981.

[66] P. J. HUBER, *Finite sample breakdown of M- and P-estimators*, Ann. Statist., 12 (1984), pp. 119–126.

[67] M. HUUMONEN, T. KÄRKKÄINEN, AND S. ÄYRÄMÖ, *Tiedonlouhinnan visualisointiohjelmistoista*. Tuotanto 2010 -projektiraportti, Jyväskylän Yliopisto - Agora Center, 2006.

[68] A. G. J. A. CUESTA-ALBERTOS AND C. MATRÁN, *Trimmed k-means: an attempt to robustify quantizers*, The Annals of Statistics, 25 (1997), pp. 553–576.

[69] A. JAIN, M. MURTY, AND P. FLYNN, *Data clustering: a review*, ACM Computing Surveys, 31 (1999), pp. 264–323.

[70] A. K. JAIN AND R. C. DUBES, *Algorithms for clustering data*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[71] A. K. JAIN, R. P. W. DUIN, AND J. MAO, *Statistical pattern recognition: A review*, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), pp. 4–37.

[72] D. JIANG, C. TANG, AND A. ZHANG, *Cluster analysis for gene expression data: A survey*, IEEE Transactions on Knowledge and Data Engineering, 16 (2004), pp. 1370–1386.

[73] J. JOE H. WARD, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association, 58 (1963), pp. 236–244.

[74] R. JÖRNSTEN, Y. VARDI, AND C.-H. ZHANG, *A Robust Clustering Method and Visualization Tool Based on Data Depth*, Birkhäuser Verlag, Switzerland, 2002, pp. 67–76.

[75] T. KÄRKKÄINEN AND S. ÄYRÄMÖ, *Robust clustering methods for incomplete and erroneous data*, in Proceedings of the Fifth Conference on Data Mining, 2004, pp. 101–112.

[76] ——, *On computation of spatial median for robust data mining*, in Proceedings of Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems (EUROGEN 2005), R. Schilling, W. Haase, J. Periaux, and H. Baier, eds., Munich, Germany, September 2005, FLM, TU Munich.

[77] T. KÄRKKÄINEN AND E. HEIKKOLA, *Robust formulations for training multilayer perceptrons*, Neural Computation, 16 (2004), pp. 837–862.

[78] I. KATSAVOUNIDIS, C.-C. JAY KUO, AND Z. ZHANG, *A new initialization technique for generalized lloyd iteration*, Signal Processing Letters, 1 (1994), pp. 144–146.

[79] L. KAUFMAN AND P. J. ROUSSEEUW, *Finding groups in data: An introduction to cluster analysis*, John Wiley & Sons, 1990.

[80] S. Khan and A. Ahmad, *Cluster center initialization algorithm for k-means clustering*, Pattern Recognition Letters, 25 (2004), pp. 1293–1302.

[81] W. Kim, B.-J. Choi, E.-K. Hong, S.-K. Kim, and D. Lee, *A taxonomy of dirty data*, Data Mining and Knowledge Discovery, 7 (2003), pp. 81–99.

[82] R. Kothari and D. Pitts, *On finding the number of clusters.*, Pattern Recognition Letters, 20 (1999), pp. 405–416.

[83] K. Krishna and M. Narasimha Murty, *Genetic k-means algorithm*, IEEE Transactions on Systems, Man and Cybernetics, Part B, 29 (1999), pp. 433–439.

[84] A. Likas, N. Vlassis, and J. J. Verbeek, *The global k-means clustering algorithm*, Pattern Recognition, 36 (2003), pp. 451–461.

[85] Y. Linde, A. Buzo, and R. Gray, *An algorithm for vector quantizer design*, IEEE Transactions on Communications, 28 (1980), pp. 84–95.

[86] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, 1987.

[87] J. Liu, J. P. Lee, L. Li, Z.-Q. Luo, and K. M. Wong, *Online clustering algorithms for radar emitter classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 1185–1196.

[88] S. P. Lloyd, *Least squares quantization in PCM.*, IEEE Transactions on Information Theory, 28 (1982), pp. 129–136.

[89] H. P. Lopuhaä and P. J. Rousseeuw, *Breakdown points of affine equivariant estimators of multivariate location and covariance matrices*, Ann. Statist., 19 (1991), pp. 229–248.

[90] R. Love, J. Morris, and G. Wesolowsky, *Facilities Location. Models and Methods*, North Holland Publishing Company, 1988.

[91] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, *FGKA: a fast genetic k-means clustering algorithm*, in SAC '04: Proceedings of the 2004 ACM symposium on Applied computing, New York, NY, USA, 2004, ACM Press, pp. 622–623.

[92] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[93] R. Maitra, *Clustering massive datasets with applications in software metrics and tomography*, Technometrics, 43 (2001), pp. 336–346.

[94] M. M. Mäkelä and P. Neittaanmäki, *Nonsmooth Optimization; Analysis and Algorithms with Applications to Optimal Control*, World Scientific, Singapore, 1992.

[95] M. Marina and H. David, *An experimental comparison of several clustering and initialization methods*, in Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98), San Francisco, CA, 1998, Morgan Kaufmann Publishers, pp. 386–395.

[96] G. Milligan and M. Cooper, *An examination of procedures for determining the number of clusters in a data set*, Psychometrika, 50 (1985), pp. 159–179.

[97] R. T. Ng and J. Han, *CLARANS: A method for clustering objects for spatial data mining*, IEEE Transactions on Knowledge and Data Engineering, 14 (2002), pp. 1003–1016.

[98] C. Ordonez, *Clustering binary data streams with k-means*, in DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, New York, NY, USA, 2003, ACM Press, pp. 12–19.

[99] C. Ordonez and E. Omiecinski, *Efficient disk-based k-means clustering for relational databases*, IEEE Transactions on Knowledge and Data Engineering, 16 (2004), pp. 909–921.

[100] D. Pelleg and A. Moore, *Accelerating exact k-means algorithms with geometric reasoning*, in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, S. Chaudhuri and D. Madigan, eds., New York, NY, August 1999, AAAI Press, pp. 277–281. An extended version is available as Technical Report CMU-CS-00-105.

[101] D. Pelleg and A. W. Moore, *X-means: Extending k-means with efficient estimation of the number of clusters*, in ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA, 2000, Morgan Kaufmann Publishers Inc., pp. 727–734.

[102] J. M. Pena, J. A. Lozano, and P. Larranaga, *An empirical comparison of four initialization methods for the k-means algorithm*, Pattern Recognition Letters, 20 (1999).

[103] H. Ralambondrainy, *A conceptual version of the k-means algorithm*, Pattern Recognition Letters, 16 (1995), pp. 1147–1157.

[104] L. Ramaswamy, B. Gedik, and L. Liu, *A distributed approach to node clustering in decentralized peer-to-peer networks*, IEEE Transactions on Parallel and Distributed Systems, 16 (2005), pp. 814–829.

[105] S. Ray and R. H. Turi, *Determination of number of clusters in k-means clustering and application in colour image segmentation*, in Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99), New Delhi, India, December 1999, Narosa Publishing House, pp. 137–143.

[106] S. J. Roberts, R. Everson, and I. Rezek, *Minimum entropy data partitioning*, in Proceedings of International Conference on Artificial Neural Networks, vol. 2, 1999, pp. 844–849.

[107] P. J. Rousseeuw, *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, Journal of Computational and Applied Mathematics, 20 (1987), pp. 53–65.

[108] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, John Wiley & Sons, Inc., 1987.

[109] O. San, V. Huynh, and Y. Nakamori, *An alternative extension of the k-means algorithm for clustering categorical data*, International Journal of Applied Mathematics and Computer Science, 14 (2004), pp. 241–247.

[110] S. Selim and M. Ismail, *K-means-type algorithms: A generalized convergence theorem and characterization of local optimality*, PAMI, 6 (1984), pp. 81–87.

[111] A. Shademan and M. Zia, *Adaptive vector quantization of mr images using on-line k-means algorithm*, in Proceedings of SPIE, 46th Annual Meeting, Application of Digital Image Processing XXIV Conference, vol. 4472, San Diego, CA, USA, July-August 2001, pp. 463–470.

[112] W. Sheng and X. Liu, *A hybrid algorithm for k-medoid clustering of large data sets*, Congress on Evolutionary Computation, 2004. CEC2004., 1 (2004), pp. 77–82.

[113] P. Simpson, *Fuzzy min-max neural networks – part 2: Clustering*, Fuzzy Systems, IEEE Transactions on, 1 (1993), pp. 32–44.

[114] S. Still and W. Bialek, *How many clusters? an information-theoretic perspective*, Neural Computation, 16 (2004), pp. 2483–2506.

[115] A. Struyf, M. Hubert, and P. J. Rousseeuw, *Integrating robust clustering techniques in S-PLUS*, Comput. Stat. Data Anal., 26 (1997), pp. 17–37.

[116] T. Su and J. G. Dy, *A deterministic method for initializing k-means clustering.*, in ICTAI, 2004, pp. 784–786.

[117] C. Sugar and G. James, *Finding the number of clusters in a data set : An information theoretic approach*, Journal of the American Statistical Association, 98 (2003), pp. 750–763.

[118] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Addison-Wesley, 2005.

[119] R. Tibshirani, G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001), pp. 411–423.

[120] J. TOU AND R. GONZALEZ, *Pattern Recognition Principles*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1974.

[121] G. C. TSENG AND W. H. WONG, *Tight clustering: A resampling-based approach for identifying stable and tight patterns in data*, Biometrics, 61 (2005), pp. 10–16.

[122] M. WOLFSON, Z. MADJD-SADJADI, AND P. JAMES, *Identifying National Types: A Cluster Analysis of Politics, Economics, and Conflict*, Journal of Peace Research, 41 (2004), pp. 607–623.

[123] R. XU AND D. W. II, *Survey of clustering algorithms*, IEEE Transactions on Neural Networks, 16 (2005), pp. 645–678.

[124] V. J. YOHAI AND R. H. ZAMAR, *High breakdown-point estimates of regression by means of the minimization of an efficient scale*, J. Amer. Statist. Assoc., 83 (1988), pp. 406–413.

[125] B. ZHANG, *Generalized k-harmonic means – boosting in unsupervised learning*, Tech. Report 137, Hewlett Packard, October 2000.

[126] J. ZHANG AND G. LI, *Breakdown properties of location M-estimators*, Ann. Statist., 26 (1998), pp. 1170–1189.

[127] S. ZHONG, T. M. KHOSHGOFTAAR, AND N. SELIYA, *Analyzing software measurement data with clustering techniques*, IEEE Intelligent Systems, 19 (2004), pp. 20–27.

[128] ——, *Unsupervised learning for expert-based software quality estimation*, in Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering (HASE'04), Tampa, FL, USA, March 2004, IEEE Computer Society, pp. 149–155.