Robust clustering methods for incomplete and erroneous data

T. Kärkkäinen & S. Äyrämö
University of Jyväskylä, Finland

Abstract

In this paper, reliable methods for clustering erroneous and incomplete data *per se* (e.g. without imputation) are considered. For this purpose, the usual K-means algorithm is generalized by using robust location estimates and special projection technique. Numerical comparison of the resulting methods with simulated data are presented and analyzed.
Keywords: robust clustering, erroneous and incomplete data, K-means.

1 Introduction

Clustering, by definition, is a descriptive technique, which is widely used, for example, in statistics, machine learning, pattern recognition, data mining (DM) and Knowledge Discovery in Databases (KDD) [1, 2, 3, 4, 5, 6]. Undoubtedly, it can be considered as a core method of DM and KDD, but the number of different clustering methods is huge. The main idea behind all of these methods is to group similar objects into the same cluster and dissimilar objects into separate clusters. Similarity (or dissimilarity) is measured by a suitable distance function. However, clustering is a challenging task since it includes many choices, such as, the decision between basic approaches (hierarchical, partitioning, density-based, model-based, grid-based, fuzzy etc.), the selection of an initialization method, the choice of a distance measure, and fixing of a cluster representation technique. These all are dependable on the nature of that particular context, in which the method is intended to be applied. The variety of application fields is also remarkable.

Although efficient systems for data gathering have been developed, most of collected real-world data sets use to be incomplete and erroneous [7, 6]. Robust techniques are by construction more suitable to such data sets and better quality of results compared to traditional clustering methods can be expected. However, as

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

usually, nothing is for free, since the price to pay for the robustness is usually increased costs in computation.

The well-known K-means algorithm (see, e.g., [1]) works as a reference method for this study. It is a prototype-based partitioning clustering method, whose popularity is based on simplicity and implementability. Although it is computationally efficient, it is also very sensitive to all kind of defects in the data and to initial conditions. K-means has also been applied to initialization of computationally more expensive algorithms (e.g., EM-algorithm [8]). Also many variants of K-means have been developed, for example, *fuzzy k-means* [3], *GKA* [9], *FGKA* [10], *K-Harmonic Means* [11] and *the global k-means algorithm* [12].

Based on K-means, two robust algorithms will be derived in this paper by replacing the sample mean with a robust multivariate estimator. Since missing data values are present in most real-data sets, restricting all computation only to available data values (c.f. [13]) was chosen as a strategy to handle computation in the presence of missing data. Notice that a similar approach was also presented by Estivill-Castro and Yang in [14], but with different algorithmic details and without taking into account the possibility of missing data values.

2 Estimation of location and robust statistics

One of the main problems in prototype-based clustering is the correct and accurate estimation of cluster prototype location in a multivariate space. The estimation of a prototype for a cluster consisting of one or more data points means that for the whole cluster, one value is computed, which is supposed to represent all data within the cluster. A number of different estimates are presented in the statistical literature (see, e.g., [15, 16, 17]).

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a sample of multivariate random variable $\mathbf{x} \in \mathbb{R}^n$. Throughout the paper, we denote by $(\mathbf{v})_i$ the ith component of a vector $\mathbf{v} \in \mathbb{R}^n$. Without parenthesis, $\mathbf{v}_i$ represents one element in the set of vectors $\{\mathbf{v}_i\}$. The l_q-norm of a vector $\mathbf{v}$ is given by

$$\|\mathbf{v}\|_q = \left(\sum_{i=1}^{n} |(\mathbf{v})_i|^q \right)^{1/q}, \quad q < \infty. \tag{2.1}$$

The most frequently used choice is the l_2-norm, which is also known as the Euclidean norm.

Next we will turn our attention to the robust estimation of a location parameter. The presentation follows the one given by Kärkkäinen and Heikkola [18]. Now, let's consider the following family of optimization problems:

$$\min_{\mathbf{u} \in \mathbb{R}^n} \mathcal{J}_q^{\alpha}(\mathbf{u}), \quad \text{for } \mathcal{J}_q^{\alpha}(\mathbf{u}) = \frac{1}{\alpha} \sum_{i=1}^{N} \|\mathbf{u} - \mathbf{x}_i\|_q^{\alpha}. \tag{2.2}$$

By restricting the consideration to the three cases $q = \alpha = 2$, $q = \alpha = 1$ and $q = 2\alpha = 2$, three optimization problems yielding three different estimates will

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

be formulated. As pointed out in [18], the two latter cases lead to nonsmooth optimization problems [19], which means that they can not be described by using the classical (C^1) differential calculus.

By choosing $q = \alpha = 2$, the problem returns to the quadratic least-squares problem. In this case, the unique solution for problem eqn. (2.2) is found by enforcing the gradient $\nabla \mathcal{J}_2^2(\mathbf{u}) = \sum_{i=1}^{N}(\mathbf{u} - \mathbf{x}_i)$ to zero, in which case the coordinatewise mean (average) $\mathbf{m} = \frac{1}{N}\sum_{i \in N} \mathbf{x}_i$ of a given sample is obtained. Problems with the sample mean are due to its high sensitivity to outliers, which means that it may be significantly disturbed by even small amounts of inconsistent data.

2.1 Robust estimation of location

The need for robust estimates results from the existence of outlying and contaminated data values in real-world data sets. They use to be a stumbling block of the sample mean because of its lack of robustness. Non-robust estimates are typically sensitive to outliers and erroneous values.

Let us define the key concepts first. According to Huber [15] robustness signifies *"insensitivity to small deviations from the assumptions"*. Barnett and Lewis [20] define an *outlier* as *"An observation (or a subset of observations) which appears to be inconsistent with the remainder of that set of data"*. A *contaminant* is defined as *"an observation from some other distribution"*[20]. There are many, although quite obvious, reasons why real-world data sets contain contaminants and outliers. Human errors during data acquisition or faults in measurement devices and data storage systems are probably the most typical ones. Probability of these aforementioned defects must be taken into account by system developers while developing reliable DM and KDD tools.

An important measure of robustness is *breakdown point*. Barnett and Lewis [20] define the breakdown point as *"the smallest proportion of contamination, which can carry the value of the estimator over all bounds"*. More formally it can be defined by the following equation (e.g., [16]):

$$\varepsilon(\hat{\mu}; \mathbf{X}) = \inf\{\frac{m}{N} : Bias(m, \hat{\mu}) = \infty\}, \tag{2.3}$$

where m is the number of contaminated data points and $\hat{\mu}(\mathbf{X})$ an estimator from $\mathbf{X}$.

For example, the breakdown point of the multivariate sample mean is 0% [17]. This means that only one outlying data value can have arbitrarily large effect on the estimator, which, in turn, clearly reflects the high sensitivity to outliers. The breakdown point can never be higher than 50%, because for larger fraction of contaminated data it becomes impossible to know, which part of the data is "good" or "bad".

2.1.1 Sample median

To formulate the problem of the sample median, we choose $q = \alpha = 1$ in eqn. (2.2) leading to the minimization of the sum of l_1-norms. As mentioned above, this

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

is actually a nonsmooth optimization problem and the subdifferential of the cost function is given by:

$$\partial \mathcal{J}_1^1(\mathbf{u}) = \sum_{i=1}^{N} \xi_i \quad \text{where} (\xi_i)_j = \text{sign}((\mathbf{u} - \mathbf{x}_i)_j). \tag{2.4}$$

The sign-function $\text{sign}(u)$ is defined such that $\text{sign}(u) = -1$ for $u < 0$, $sign(u) = 1$ for $u > 0$ and $sign(u) = [-1, 1]$ for $u = 0$. The solution of the problem is the coordinatewise median. In practice, it is a set of middle values taken from a coordinatewise ordered sample set. The solution is unique for odd $\mathcal{N}$, but for even $\mathcal{N}$, all points in the closed interval between the middle values satisfy eqn. (2.4). In this case, an appropriate single choice for a prototype (used e.g., in MATLAB) is the average of the middle values.

The coordinatewise sample median is a nonparametric estimate of the population median [17]. As shown, the idea is to estimate the median of each variable separately. Although the sample median is a robust estimate of location (breakdown point as high as 50%) and straightforward to compute, it has some undesirable properties. First, it suffers from the lack of rotational invariance. Another remarkable drawback is that the coordinatewise median does not have to lie in the convex hull when samples are drawn from $\mathbb{R}^n$ for $n \geq 3$. An illustrative example is a case of three n-dimensional data vectors $\mathbf{x}_1 = 1, 0, \ldots, 0$, $\mathbf{x}_2 = 0, 1, \ldots, 0$ and $\mathbf{x}_3 = 0, 0, \ldots, 1$ [17, p.250]. The coordinatewise median of the given sample is now $\mathbf{m} = 0, 0, \ldots, 0$, which is neither inside the convex hull of the sample nor very representative estimate for the data vectors.

2.1.2 Spatial median

Now we choose $q = 2\alpha = 2$ that leads to the computation of the spatial median (a.k.a multivariate L1-median or Weber point [16]), which is also a nonsmooth optimization problem. The gradient of the convex cost function $f(\mathbf{u}, \mathbf{x}_i) = \|\mathbf{u} - \mathbf{x}_i\|_2$ is well-defined and unique for all $\mathbf{u} \neq \mathbf{x}_i$. However, the case $\mathbf{u} = \mathbf{x}_i$ leads to the use of a subgradient, which is characterized by the condition $\|\xi\|_2 \leq 1$. The whole subgradient for eqn. (2.5) reads as

$$\partial \mathcal{J}_2^1(\mathbf{u}) = \sum_{i=1}^{N} \xi_i, \quad \text{with} \begin{cases} (\xi_i)_j = \frac{(\mathbf{u}-\mathbf{x}_i)_j}{\|\mathbf{u}-\mathbf{x}_i\|_2}, & \text{for } \|\mathbf{u} - \mathbf{x}_i\|_2 \neq 0 \\ \|\xi_i\|_2 \leq 1, & \text{for } \|\mathbf{u} - \mathbf{x}_i\|_2 = 0 \end{cases} \tag{2.5}$$

When datapoints $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ belong to Euclidean space and they are not collinear (not concentrated on a straight line), the spatial median is unique [21]. In the collinear cases, the spatial median collapses to the one-dimensional coordinatewise median, which means that it still exists, but it may not be unique.

As seen from the eqn. (2.5), the spatial median is obtained by projecting the data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ on a candidate unit sphere centered at $\mathbf{u}$ and moving $\mathbf{u}$ so that finally the average of the projected values lies at the center of the sphere (the sum of the projected vectors is zero). By projecting the data on the unit sphere, only the directions of the data vectors effect on the result and each data point will be

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

equally weighted. This decreases the sensitivity of a location estimator to outliers and requirements for the amount and quality of data.

3 Prototype-based partitioning clustering

The basic idea of prototype-based partitioning clustering is to assign all data points to their closest prototypes, update the prototype locations and use them as new representatives for the clusters. K-means is the best-known method of this class. It is used as a reference method for other algorithms in this paper. Its popularity is based on low computational costs and simplicity. The time complexity of K-means is $\mathcal{O}(ndKT)$ where n is the number of data points, d is the number of features, K is the number of clusters, and T is the number of iterations [3]. It also consumes less memory than, for example, hierarchical clustering methods, because it does not exploit a dissimilarity matrix in computation [22].

The K-means algorithm consist of four main steps. First, it must be initialized with cluster centers. The basic iteration proceeds such that each data point is first assigned to its closest center and secondly, the cluster centers are updated by computing the sample mean for each cluster. This is repeated until the selected stopping criteria is met.

Actually the K-means algorithm is a splitting method for solving an optimization problem. K-means partitions the sample $\mathbf{X}$ into K disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ such that for each $\mathbf{x}_i$, which is assigned to a particular subset $\mathcal{C}_k$ for $k \in \{1, \ldots, K\}$, the Euclidean distance $d(\mathbf{x}_i, \mathbf{m}_k)$ between the data point and the mean $\mathbf{m}_k$ of the subset $\mathcal{C}_k$ is minimized. A common definition for the K-means optimization problem is [23, 24]:

$$\min_{\mathbf{m}_{1,\ldots,K} \in \mathbb{R}^n} \mathcal{J}_e, \text{ for } \mathcal{J}_e(\mathbf{m}_1, \ldots, \mathbf{m}_K) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathcal{R}(\mathbf{x}_i)\|^2, \tag{3.1}$$

where

$$\mathcal{R}(\mathbf{x}_i) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \|\mathbf{x}_i - \mathbf{m}_k\|^2. \tag{3.2}$$

Function $\mathcal{R}(\mathbf{x}_i) \in \{\mathbf{m}_1, \ldots, \mathbf{m}_K\} \subset \mathbb{R}^n$ determines which cluster prototype $\mathbf{m}_k$ is closest to $\mathbf{x}_i$.

Although K-means is a popular algorithm with some good properties, it suffers from a few significant drawbacks [25, 8]. The most significant is the lack of robustness related to the estimation of cluster prototypes. It presumes symmetrical and Gaussian shape of all cluster density functions. From this it follows that large amounts of clean data are needed. Therefore, the focus of this study is on the estimates of cluster prototypes. By substituting the sample mean with a robust alternative, such as the coordinatewise median or the spatial median, new prototype-based clustering methods with more robust behavior are developed.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

3.1 Robust prototype-based partitioning clustering algorithms

To put into use the aforementioned robust estimators the K-means algorithm is first generalized. The general algorithm does not define the used distance measures and prototype estimators. The algorithm includes the following steps:

1. Choose a distance function denoted by $d(\mathbf{u}, \mathbf{v})$.
2. Choose a cost function denoted by $\mathcal{J}_q^\alpha(\mathbf{u})$ for the cluster prototype estimation.
3. Initialize cluster prototypes $\mathbf{u}_i, \forall i \in 1, \ldots, K$.
4. Assign each data item in data set $\mathbf{X}$ to its closest cluster center according to distance function $d(\mathbf{u}, \mathbf{v})$.
5. Recompute the cluster prototypes by minimizing cost function $\mathcal{J}_q^\alpha(\mathbf{u}_i) \quad \forall i \in \{1, \ldots, K\}$.
6. If the stopping criterion is satisfied then stop. Otherwise repeat from step 4.

Some alternatives for the stopping criterion are:

- $\underset{k\in\{1,\ldots,K\}}{\arg\max} |\mathbf{m}_k^{iter} - \mathbf{m}_k^{iter-1}|_\infty \leq \epsilon$
- Number of reassignments between clusters.
- Decrease in the overall error measured using a suitable norm.

Two robust algorithms, namely K-medians and K-spatialmedians, are developed by applying the aforementioned robust estimators to the general algorithm. As a measure of distance between two data vectors $\mathbf{u}$ and $\mathbf{v}$ the Euclidean distance, given by $\|\mathbf{u} - \mathbf{v}\|_2$, is utilized.

In order to deal with incomplete data sets, a strategy for the missing data treatment must be embedded into the formulae. Since we do not want to be involved in making hypotheses on the distributions of unknown clusterwise data, we chose to apply a strategy, which employs only available data values in the calculation of distances and location estimators [13, 26]. From this it follows that all computations of distances are restricted to existing fields of the original data. Therefore, the optimization problem eqn. (2.2) will be generalized for missing data cases by using a projector technique.

A convenient way to indicate the available data is to define a projector 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 redefine the family of optimization problems given in eqn. (2.2):

$$\min_{\mathbf{u}\in\mathbb{R}^n} \mathcal{J}_q^\alpha(\mathbf{u}), \quad \text{for } \mathcal{J}_q^\alpha(\mathbf{u}) = \frac{1}{\alpha}\sum_{i=1}^{N} \|\mathbf{P}_i(\mathbf{u} - \mathbf{x}_i)\|_q^\alpha. \tag{3.3}$$

Exactly similar projector technique can be applied to distance calculation.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

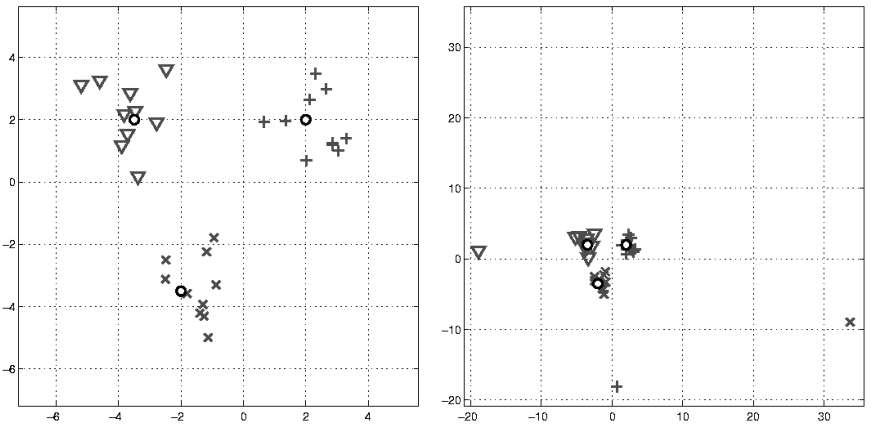

Figure 1: Left: A plot of the test data set, in which three clusters of size ten are well-separated. Right: The dirty test data set, in which four data values (appr. 6.67 percent of the whole data) are randomly distorted.

4 Experiments

All the algorithms were tested and compared on a small artificial 2-d data set. The test data consists of 30 data points that were obtained by taking three samples of size ten from the Gaussian distribution such that each sample composed a well-separated data cluster (see Figure 1). To make sure the correctness of the algorithms in optimal conditions, they all were tested on the complete and clean data containing well-separated clusters. To compare robustness against outliers and missing data, four of the overall 60 data values (30 observations times 2 variables) were disturbed and thus transformed into outliers (Figure 1) and, moreover, 10%, 30% and 50% of data values, in turn, were removed.

The tests were run on MATLAB 6.1. environment. To solve the optimization problem for the spatial median estimator, conjugate gradient (CG) and golden section (GS) method were implemented [27]. The update of search direction in CG was realized using Polak-Ribiere method. To find the exact solutions for the final cluster prototypes Nelder-Mead algorithm [28] was utilized.

The maximum norm of cluster displacement was used as a stopping criterion in each algorithm (see Section 3.1). The tolerance was set at 10^{-3}. The stopping criteria in CG was defined as $\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_\infty \leq 10^{-6}$ and, in GS, as $\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_2 \leq 10^{-8}$, where $\mathbf{u}^k$ is the solution after k^{th} iteration.

Each algorithm was tested with 100 different random initialization for cluster prototypes on each data sets. Accuracy and correctness of the results was compared and evaluated visually by presenting the error distributions for all results (for all 100 test runs) using histograms. The errors were calculated by assigning each of the obtained cluster prototypes to the closest original cluster center. Note that each

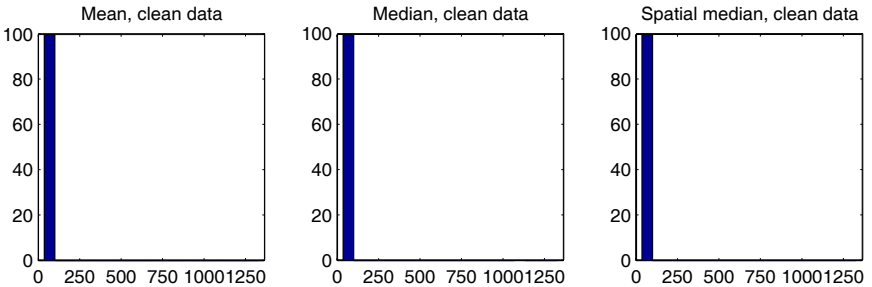

Figure 2: Error distributions from 100 test runs on the original complete data set.

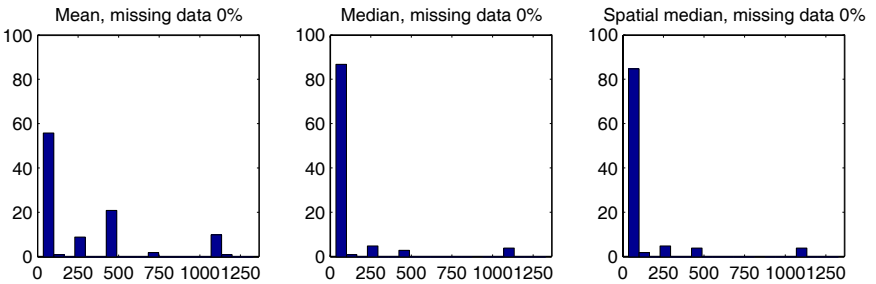

Figure 3: Error distributions from 100 test runs on the complete data set with outliers.

original prototype captures one, and only one, estimated prototype.

4.1 Results

Let's finally consider the results. As it is shown in Figure 2, no significant difference in the accuracy of K-means, K-medians and K-spatialmedians was found for clean data. All the algorithms give very good results for such data set.

Next, the clustering algorithms were tested on the data set containing the outliers, but no missing data. The results are illustrated in Figure 3. As it can be seen, the accuracy of the algorithms was decreased due to addition of the outliers. However, clear differences can be observed between K-means and the robust algorithms. K-means produces good results with only 50% probability, whereas for K-medians and K-spatialmedians the same probability is almost 90% and they perform almost identically.

Removing 10 % of data seems to impair increasingly the accuracy of K-means (see Figure 4). Approximately half of the test runs produced significant errors. The accuracy of K-medians and K-spatialmedians also decreased in comparison to the test runs on the complete data with the outliers, but approximately 80 %

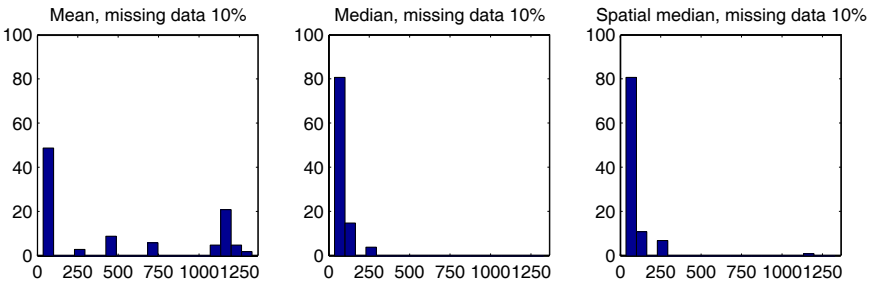

Figure 4: Error distributions from 100 test runs on the incomplete data set (10% of values missing) with outliers.

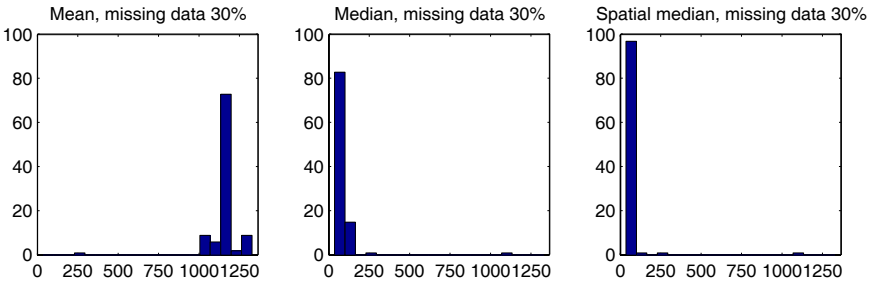

Figure 5: Error distributions from 100 test runs on the incomplete data set (30% of values missing) with outliers.

of the results were still at the best level and almost none of them were poor. No significant difference was found for the robust algorithms.

To further stress the robust algorithms, the portion of missing data was increased to 30%. The results are illustrated in Figure 5. Obviously, the efficiency of K-means is not anymore feasible. Almost all of the results are unsatisfactory. The results of K-medians and K-spatialmedians were again quite identical. The results were even better when compared to the two previous cases, but it can be interpreted as a coincidence. However, it shows that even third of the data may be lost without significant influence on the accuracy of the robust algorithms. Finally, Figure 6 presents the averages and medians of the errors produced by the algorithms. Considering the median error of K-means one can observe that one half of the results suffers from significant error, when more than 10% of data is missing and outliers are present.

The definite contribution of the outliers to the weak performance of K-means may be deduced by comparing the errors produced by K-means against the errors of the two robust algorithms. Figure 6 shows that for K-means the average error,

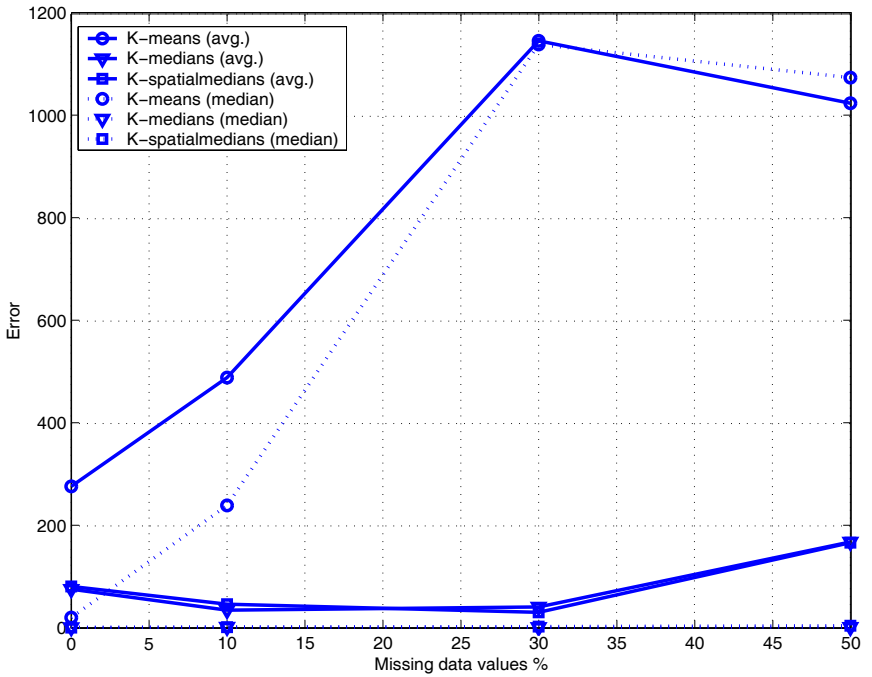

Figure 6: Averages and medians of the error for the methods.

when no data is missing, is greater than the error of K-medians and K-spatialmedians when 10%-50% of data is missing.

Although the median errors of K-medians and K-spatialmedians do not vary much according to percentage of missing data, the average quality of the results impairs somewhat.

5 Conclusions

The data mining algorithms contains inherently a number of adjustable parameters that are difficult to understand by an ordinary end user. Therefore, such algorithms that are adjusted to the target environment in advance are needed. Our long-term goal is to develop robust clustering algorithms that are able to handle noisy and incomplete data sets with a minimal number of user interventions. Although estimation of the correct number of clusters was not considered in this paper, it is one of the main issues related to data clustering [1].

The main goal of this paper was to apply and test robust estimators in prototype-based partitioning clustering algorithms. The results were encouraging. The robust variants of the clustering algorithms performed much better in the cases of dirty and missing data. Since large part of the test runs with the robust procedures produced good results, even if 10%-50% of the data were missing and 7% distorted,

we expect that careful initialization of the algorithms allows one to achieve appropriate clusters with high probability. It is also noticeable that the performance of K-spatialmedians and K-medians distinguished slightly from each other. However, this result is not considered totally unexpected, since the test data was sampled from two-dimensional distributions. The weakness of the coordinatewise sample median may not appear until in higher dimensional problems (see Section 2.1.1). To deal with large data sets, relative to the number of dimensions and observations, the current algorithm for computing the spatial median is inefficient, but the results encourage us to further develop faster variants for this purpose.

References

[1] Kaufman, L. & Rousseeuw, P.J., *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons, 1990.

[2] Jain, A., Murty, M. & Flynn, P., Data clustering: a review. *ACM Computing Surveys*, **31(3)**, pp. 264–323, 1999.

[3] Duda, R.O., Hart, P.E. & Stork, D.G., *Pattern classification*. John Wiley & Sons, Inc., 2001.

[4] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R., *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, 1996.

[5] Hastie, T., Tibshirani, R. & Friedman, J., *The elements of statistical learning: Data mining, inference and prediction*. Springer-Verlag, 2001.

[6] Han, J. & Kamber, M., *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, Inc., 2001.

[7] Kim, W., Choi, B.J., Hong, E.K., Kim, S.K. & Lee, D., A taxonomy of dirty data. *Data Mining and Knowledge Discovery*, **7(1)**, pp. 81–99, 2003.

[8] Bradley, P.S. & Fayyad, U.M., Refining initial points for K-Means clustering. *Proc. 15th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp. 91–99, 1998.

[9] Bandyopadhyay, S. & Maulik, U., An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Inf Sci Appl*, **146(1-4)**, pp. 221–237, 2002.

[10] Lu, Y., Lu, S., Fotouhi, F., Deng, Y. & Brown, S., FGKA: A fast genetic K-means clustering algorithm. *Proc. of the 19th ACM Symposium on Applied Computing*, ACM Press, 2004. To appear.

[11] Zhang, B., Generalized K-harmonic means – boosting in unsupervised learning. Technical Report 137, Hewlett Packard, 2000.

[12] Likas, A., Vlassis, N. & Verbeek, J.J., The global k-means clustering algorithm. *Pattern Recognition*, **36(2)**, pp. 451–461, 2003.

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

[14] Estivill-Castro, V. & Yang, J., Fast and robust general purpose clustering algorithms. *Data Mining and Knowledge Discovery*, **8(2)**, pp. 127–150, 2004.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

[15] Huber, P., *Robust statistics*. John Wiley & Sons, 1981.
[16] Small, C., A survey on multidimensional medians. *Internat Statist Rev*, **58**, pp. 263–277, 1990.
[17] Rousseeuw, P.J. & Leroy, A.M., *Robust regression and outlier detection*. John Wiley & Sons, Inc., 1987.
[18] Kärkkäinen, T. & Heikkola, E., Robust formulations for training multilayer perceptrons. *Neural Computation*, **16(4)**, pp. 837–862, 2004.
[19] Mäkelä, M. & Neittaanmäki, P., *Nonsmooth Optimization; Analysis and Algorithms with Applications to Optimal Control*. World Scientific: Singapore, 1992.
[20] Barnett, V. & Lewis, T., *Outliers in statistical data*. John Wiley & Sons, 2nd edition, 1984.
[21] Vardi, Y. & Zhang, C.H., The multivariate l1-median and associated data depth. *Proceedings of the National Academy of Science*, National Academy of Sciences: USA, volume 97, pp. 1423–1426, 2000.
[22] Berkhin, P., Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
[23] Wan, S.J., Wong, S.K.M. & Prusinkiewicz, P., An algorithm for multidimensional data clustering. *ACM Trans Math Softw*, **14(2)**, pp. 153–162, 1988.
[24] Estivill-Castro, V. & Murray, A.M., Hybrid optimization for clustering in data mining. Technical Report 2000-01, Callaghan 2308, Australia, 2000.
[25] Pena, J.M., Lozano, J.A. & Larranaga, P., An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recogn Lett*, **20(10)**, pp. 1027–1040, 1999.
[26] Everitt, B.S., Landau, S. & Leese, M., *Cluster analysis*. Arnolds, a member of the Hodder Headline Group, 2001.
[27] Bazaraa, M.S., Sherali, H.D. & Shetty, C.M., *Nonlinear programming: Theory and algorithms*. John Wiley & Sons, Inc., 1993.
[28] Lagarias, J., Reeds, J.A., Wright, M.H. & Wright, P.E., Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, **9(1)**, pp. 112–147, 1998.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)