A fast algorithm for robust constrained clustering*

Heinrich Fritz
Department of Statistics and Probability Theory
Vienna University of Technology
Luis A. García-Escudero and Agustín Mayo-Iscar
Department of Statistics and Operations Research
University of Valladolid

Abstract

The application of "concentration" steps is the main principle behind Forgy's k-means algorithm and Rousseeuw and van Driessen's fast-MCD algorithm. Although they share this principle, it is not completely straightforward to combine both algorithms for developing a clustering method which is not affected by a certain proportion of outlying observations and that is able to cope with non spherical groups or with groups with different weights. However, these approaches can be successfully combined by additionally controlling the relative cluster scatters in the concentration steps. In this way, the appearance of uninteresting spurious clusters is avoided. An algorithm which implements such "constrained concentration" steps in a computationally efficient way will be presented in this work.

Key words: Cluster Analysis; Robustness; Trimming; k-means; MCD; Trimmed k-means.

1 Introduction

It is easy to see certain relations between Forgy's k-means algorithm (Forgy, 1965) and the fast-MCD algorithm (Rousseeuw and Van Driessen, 1999). These two widely applied algorithms play a very important role in Cluster Analysis and in Robust Statistics, respectively. The connection between these methods mainly refers to the application of so called"concentration" steps which will be explained later in Section 3.

This relation gets clearer when comparing the fast-MCD algorithm to the trimmed k-means algorithm (García-Escudero *et al.*, 2003), since trimming (outlying) data is an important characteristic of both methods. Notice, that the trimmed k-means algorithm simplifies to Forgy's k-means algorithm when the trimming level α is set to 0. More information on

*This research was partially supported by Ministerio de Ciencia y Tecnología and FEDER grant BFM2002-04430-C02-01 and by Consejería de Educación y Cultura de la Junta de Castilla y León grant PAPIJCL VA074/03.

Trimmed k-means	**Fast-MCD**
...	...
– Randomly draw k centers.	– Randomly draw a center and a scatter matrix.
...	...
– Trim a proportion α of the most remote observations to these k centers, considering Euclidean distances.	– Trim a proportion α of the most remote observations to the center, considering Mahalanobis distances.
– Compute k new centers using the non-trimmed observations.	– Compute a new center and scatter matrix using the non-trimmed observations.
...	...
– Return the k centers leading to the "best" value of the target function.	– Return the center and scatter matrix leading to the "best" value of the target function.

Table 1: Schematic description of the differences between the trimmed k-means and fast-MCD algorithms.

the trimmed k-means procedure can be found in Cuesta-Albertos *et al.* (1997) and García-Escudero and Gordaliza (1999). A very simplified comparison of the concentration steps for trimmed k-means and fast-MCD is given in Table 1.

The main drawback of using k-means and trimmed k-means is that they ideally search for spherically scattered groups and for clusters with equal size, whereas in many clustering problems the clusters we are looking for do not necessarily follow these assumptions. Thus, in this work, we focus on general "heterogeneous" clustering problems where elliptically contoured clusters can also be expected. Further we expect the data to contain a certain fraction α of outlying observations which would negatively affect classical clustering procedures (see García-Escudero *et al.*, 2010). In this setup, it seems logical to combine the clustering capabilities of k-means with the ability to robustly estimate covariance structures provided by the fast-MCD algorithm. Thus, we can think of applying the trimmed k-means algorithm, but considering Mahalanobis distances when identifying the closest cluster center to each observation. The centers and scatter matrices are updated by computing the sample means and sample covariance matrices of the observations assigned to each cluster. Unfortunately, this "naive" combination of both algorithms does not provide sensible clustering results, since large groups sometimes tend to "eat" smaller ones, and the algorithm ends up finding spurious groups with few, almost collinear observations. This problem has already been described in Maronna and Jacovkis (1974).

A sensible way to address this issue is to impose constraints which control the relative difference among cluster scatters. In fact, many well-know clustering methods implement (implicitly and explicitly) such constraints on the relative cluster sizes, as for example the k-means method assumes spherical clusters with similar scatter. With this idea in mind, García-Escudero *et al.* (2008) introduced the TCLUST method, which is based on a relative size constraint on the eigenvalues of the scatter matrices defining the shape of the elliptically contoured groups. The idea of using restrictions of this type goes back to Hathaway (1985)

where related constraints were proposed in a mixture fitting framework.

From a computational point of view, solving the TCLUST problem is not an easy task. One of the most critical issues in this algorithm is how to enforce the relative size constraints. Unfortunately, this is the computational bottle-neck of the algorithm, because a complex optimization problem must be solved in each concentration step. In this work, we present a computational efficient algorithm for such "constrained concentration" steps, which clearly speeds up the TCLUST algorithm and makes it computationally feasible for practical applications.

It is also important to note that the idea of such constrained concentration steps can be easily extended to other constrained clustering methods like Gallegos (2002) and Gallegos and Ritter (2005).

The methodology of the discussed approach is explained in Section 2, whereas in Section 3 the corresponding algorithm is presented. Section 4 contains a simulation study, investigating the performance of the algorithm and Section 5 concludes.

2 Constrained robust clustering and TCLUST

Given a sample of observations $\{x_1, ..., x_n\}$ in $\mathbb{R}^p$ and $\phi(\cdot; \mu, \Sigma)$ the probability density function of a p-variate normal distribution with mean μ and covariance matrix Σ, we consider the following general *robust clustering problem*:

> Search for a partition $R_0, R_1, ..., R_k$ of the indices $\{1, ..., n\}$ with $\#R_0 = \lceil n\alpha \rceil$, centers $m_1, ..., m_k$, symmetric positive semidefinite scatter matrices $S_1, ..., S_k$ and weights $p_1, ..., p_k$ with $p_j \in [0, 1]$ and $\sum_{j=1}^{k} p_j = 1$, which maximizes
>
> $$\sum_{j=1}^{k} \sum_{i \in R_j} \log \big(p_j \phi(x_i; m_j, S_j) \big). \tag{2.1}$$

Depending on the constraints imposed on the weights p_j and scatter matrices S_j, the maximization of (2.1) for $\alpha = 0$ leads to well established clustering procedures. For instance, a very constrained setup, assuming equal weights $p_1 = ... = p_k$ and scatter matrices $S_1 = ... = S_k = \sigma^2 I$ with I being the identity matrix and $\sigma > 0$, yields the k-means method. The determinantal criterion introduced by Friedman and Rubin (1967) is obtained when assuming $p_1 = ... = p_k$ and $S_1 = ... = S_k = S$ with S being a positive definite matrix. In general, the "log-likelihood" in (2.1) when $\alpha = 0$ and $p_1 = ... = p_k$ corresponds to the Classification-Likelihood (see e.g. Scott and Symons, 1971). The use of (2.1) assuming different weights p_j goes back to Symons (1981) and Bryant (1991) and is also known as the Complete-Data-Likelihood approach to Cluster Analysis.

Trimmed alternatives to the previously commented approaches can be constructed by introducing a trimming level $\alpha > 0$ to (2.1), which yields "trimmed log-likelihoods". This way, the trimmed k-means method in Cuesta-Albertos *et al.* (1997) extends k-means and the trimmed determinantal criterion in Gallegos and Ritter (2005) extends the determinantal criterion. Notice that $\lceil n\alpha \rceil$ observations (R_0) are not taken into account when computing (2.1), and thus the harmful effect of outlying observations, up to a contamination α, can be avoided. Gallegos and Ritter (2005) introduce the so-called "spurious outlier model" that theoretically justifies the use of trimmed log-likelihoods like in (2.1).

It is also important to note that the robust clustering problem reduces to the fast-MCD method when assuming $k = 1$ (i.e. only partitioning the data into $\lceil n\alpha \rceil$ outliers and $\lfloor n\,(1-\alpha) \rfloor$ regular observations). The fact that the same target function defines both problems emphasizes the relation between robust clustering methods and the MCD estimator.

It is straightforward to see, that the direct maximization of (2.1) without any constraint on the scatter matrices is not a well defined problem, since a single cluster scatter matrix S_j with $\det(S_j) \to 0$ causes (2.1) to tend to infinity. Thus partitions containing spurious clusters are quite likely and preferred to more sensible solutions. This explains why the previously described "naive" algorithm (combining the two algorithms from Table 1) does not work appropriately.

In order to make the maximization of (2.1) a well defined problem, García-Escudero *et al.* (2008) propose to additionally consider the following eigenvalue-ratio constraint on the scatter matrices $S_1, ..., S_k$:

$$\frac{\max_{j,l} \lambda_{j,l}}{\min_{j,l} \lambda_{jl}} \leq c, \tag{2.2}$$

with $\lambda_{j,l}$ as the eigenvalues of the corresponding scatter matrices S_j (for $j = 1, ..., k$ and $l = 1, ..., p$) and $c \geq 1$ as a constant which controls the strength of the constraint (2.2). The maximization of (2.1) under the eigenvalue-ratio constraint (2.2) leads to the TCLUST problem introduced by García-Escudero *et al.* (2008). The smaller the value of c, the stronger is the restriction imposed on the solution, yielding the strongest constraint $c = 1$, which corresponds to the k-means procedure with different cluster weights.

The TCLUST method has good theoretical and robustness properties but no practically applicable algorithm is available yet when $k \cdot p$ is moderately large. With this in mind, a feasible algorithm for efficiently implementing this method will be described in the following section.

3 Algorithm

An algorithm for approximately maximizing (2.1) under the constraint (2.2) has been presented in García-Escudero *et al.* (2008), whereas a significantly faster approach will be presented here. Further, an inaccuracy in the presentation of the algorithm in García-Escudero *et al.* (2008) will be corrected here.

Both algorithms can be seen as a trimmed version of the Classification Expectation-Maximization (EM) algorithms proposed in Schroeder (1976) and Celeux and Govaert (1992).

In the *E-step*, at a given iteration, each observation x_i is assigned to the cluster with closest center. Since we are considering different weights and scatter matrices, the distance of an observation x_i to the center of cluster j is proposed to be quantified by a so-called "discriminant function":

$$D_j(x_i; \theta) = p_j \phi(x_i; m_j, S_j).$$

with $\theta = (p_1, ..., p_k, m_1, ..., m_k, S_1, ..., S_k)$ as the set of cluster parameters in the current iteration of the algorithm. The smaller $D_j(x_i; \theta)$ is, the larger is the distance of observation x_i to a center m_j. Further,

$$D(x_i; \theta) = \max\{D_1(x_i; \theta), ..., D_k(x_i; \theta)\} \tag{3.1}$$

defines an overall measure for outlyingness.

Notice that if $k = 1$, observations with largest (3.1) are those with smallest Mahalanobis distances

$$(x_i - m_1)' S_1^{-1} (x_i - m_j). \tag{3.2}$$

These observations are taken into account in the concentration steps of the fast-MCD algorithm.

Further, when assuming $p_1 = ... = p_k$ and $S_1 = ... = S_k = \sigma^2 I$, observations with largest (3.1) are those with smallest values of

$$\min_{j=1,...,k} \|x_i - m_j\|^2,$$

as considered in the concentration steps of the (trimmed) k-means algorithm.

With this notation, the $\lceil n\alpha \rceil$ observations x_i with smallest values of $D(x_i; \theta)$ can be discarded as possible outliers (trimmed), whereas $D_j(x_i; \theta)$ is used to assign the remaining observations to one of the j groups. Notice, that in contrast to mixture clustering approaches, this approach fully assigns each (non-trimmed) observation to a cluster and thus is a "crisp" clustering method.

In a second step, the *M-step*, the cluster parameters are updated, based on the non-trimmed observations and the corresponding cluster assignments. At this point it is crucial to constrain the cluster scatter matrices for avoiding spurious clusters.

A more detailed presentation of the proposed algorithm is given as follows:

1. *Initialization:* The procedure is initialized `nstart` times by selecting different $\theta^0 = (p_1^0, ..., p_k^0, m_1^0, ..., m_k^0, S_1^0, ..., S_k^0)$. For this purpose we propose to randomly select $k(p+1)$ observations and to accordingly compute k cluster centers m_j^0 and scatter matrices S_j^0 from the chosen data points. Afterwards the cluster scatter matrix constraints are applied to these S_j^0, as described in Step 2.2. Weights $p_1^0, ..., p_k^0$ in the interval $(0, 1)$ and summing up to 1 are also randomly chosen.

2. *Concentration step:* The following steps are executed until convergence (i.e. $\theta^l = \theta^{l-1}$) or a maximum number of iterations `iter.max` is reached.

   2.1. *Trimming and cluster assignment (E-step):* Based on the current parameter set $\theta^l = (p_1^l, ..., p_k^l, m_1^l, ..., m_k^l, S_1^l, ..., S_k^l)$ the $\lceil n\alpha \rceil$ observations with smallest values of $D_j(x_i, \theta^l)$ are trimmed. Each remaining observation x_i is then assigned to a cluster j, such that $D_j(x_i, \theta^l) = D(x_i, \theta^l)$. This yields a partition $R_0, R_1, ..., R_k$ of the indices $\{1, ..., n\}$ holding the trimmed observations in R_0 and all observations belonging to cluster j in R_j for $j = 1, ..., k$.

   2.2. *Update parameters (M-step):* Given $n_j = \#R_j$, the weights are updated by

   $$p_j^{l+1} = n_j / [n(1 - \alpha)]$$

   and the centers by the sample means

   $$m_j^{l+1} = \frac{1}{n_j} \sum_{i \in R_j} x_i.$$

Updating the scatter estimates is more difficult, as the sample covariance matrices

$$T_j = \frac{1}{n_j} \sum_{i \in R_j} (x_i - m_j^{l+1})(x_i - m_j^{l+1})',$$

may not satisfy the specified eigenvalue-ratio constraint. In this case, the singular-value decomposition of $T_j = U_j' D_j U_j$ is considered, with U_j being an orthogonal matrix and $D_j = \text{diag}(d_{j1}, d_{j2}, ..., d_{jp})$ a diagonal matrix. Let us define the truncated eigenvalues as

$$d_{jl}^m = \begin{cases} d_{jl} & \text{if } d_{jl} \in [m, cm] \\ m & \text{if } d_{jl} < m \\ cm & \text{if } d_{jl} > cm \end{cases} \tag{3.3}$$

with m as some threshold value. The scatter matrices are updated as

$$S_j^{l+1} = U_j' D_j^* U_j,$$

with $D_j^* = \text{diag}\left(d_{j1}^{m_{\text{opt}}}, d_{j2}^{m_{\text{opt}}}, ..., d_{jp}^{m_{\text{opt}}}\right)$ and m_{opt} minimizing

$$m \mapsto \sum_{j=1}^{k} n_j \sum_{l=1}^{p} \left(\log\left(d_{jl}\right) + \frac{d_{jl}^m}{d_{jl}} \right). \tag{3.4}$$

As shown in Remark 3, this expression has to be evaluated only $2kp+1$ times for exactly finding this minimum.

3. *Evaluate target function:* After the concentration steps the value of the target function (2.1) is computed. The set of parameters yielding the highest value of this target function is returned as the algorithm's output.

The proposed algorithm can be used to solve the maximization of (2.1) when assuming equal weights $p_1 = ... = p_k$, by simply setting all weights constantly to $p_j^l = 1/k$ within each iteration.

Remark 1 *The number of random starts* `nstart` *and the maximum number of constrained-concentration steps* `iter.max` *depends on the complexity of the processed data set. Experience shows that not excessively large values of* `nstart` *and* `iter.max` *are needed to obtain a proper solution if, apart from outliers, the cluster structure is easy to be discovered (see also Section 4).*

García-Escudero et al. *(2011) provides some graphical tools which help to make appropriate choices for the number of groups k and the trimming level α.*

If the constraints on the eigenvalues are not considered, the algorithm essentially coincides with the method proposed in Neykov et al. *(2007), which is also based on trimmed likelihoods. However, as already mentioned, explicitly stating relative cluster scatter constraints and providing a computational procedure for solving them is very important in this approach to robust clustering.*

Remark 2 *The main novelty of this algorithm compared to García-Escudero* et al. *(2008) is how the constraint on the eigenvalue ratio is imposed. Equation* (3.4) *in García-Escudero* et al. *(2008) constrains eigenvalues by solving the minimization problem*

$$(d^*_{11}, d^*_{12}, ..., d^*_{jl}, ..., d^*_{kp}) \mapsto \sum_{j=1}^{k} n_j \sum_{l=1}^{p} \left(\log\left(d_{jl}\right) + \frac{d^*_{jl}}{d_{jl}} \right), \tag{3.5}$$

under the restriction

$$(d^*_{11}, d^*_{12}, ..., d^*_{jl}, ..., d^*_{kp}) \in \Lambda, \tag{3.6}$$

with Λ *as the cone*

$$\Lambda = \left\{ d^*_{jl} : d^*_{jl} \leq c \cdot d^*_{rs} \text{ for every } j, r \in \{1, ..., k\} \text{ and } l, s \in \{1, ..., p\} \right\}. \tag{3.7}$$

This is clearly a more complex problem than minimizing (3.4) *because its complexity tremendously increases with the number of groups* k *and the dimension* p*. In García-Escudero* et al. *(2008), the problem of minimizing* (3.5) *in* Λ *was translated into a quadratic programming problem which can be approximately solved by recursive projections onto cones (Dykstra, 1983). However, as this computationally intensive problem must be solved in each concentration step, the algorithm becomes extremely slow and even unfeasible for high values of* $k \cdot p$*. Moreover, there was a mistake in García-Escudero* et al. *(2008), as the term* n_j *in (3.5) was omitted, and thus the algorithm proposed there can only be applied onto similarly sized clusters.*

Remark 3 *There is a closed form for obtaining* m_{opt} *(and thus, the constrained eigenvalues) just by evaluating function* (3.4) $2pk+1$ *times. Let us consider* $e_1 \leq e_2 \leq ... \leq e_{2kp}$ *obtained by ordering the following* $2pk$ *values:*

$$d_{11}, d_{12}, ..., d_{jl}, ..., d_{kp}, d_{11}/c, d_{12}/c, ..., d_{jl}/c, ..., d_{kp}/c.$$

After that, let us consider any $2pk+1$ *values* $f_1, ..., f_{2kp+1}$ *satisfying:*

$$f_1 < e_1 \leq f_2 \leq e_2 \leq ... \leq f_{2kp} \leq e_{2kp} < f_{2kp+1},$$

and, compute

$$m_i = \frac{\sum_{j=1}^{k} n_j \left(\sum_{l=1}^{p} d_{jl}(d_{jl} < f_i) + \frac{1}{c} \sum_{l=1}^{p} d_{jl}(d_{jl} > c f_i) \right)}{\sum_{j=1}^{k} n_j \left(\sum_{l=1}^{p} ((d_{jl} < f_i) + (d_{jl} > c f_i)) \right)},$$

for $i = 1, ..., 2kp+1$*. Finally, choose* m_{opt} *as the value of* m_i *which yields the minimum value of* (3.4)*.*

Remark 4 *An implementation of the algorithm described in this work has been made available through the* ***R*** *package* `tclust` *at* `http://CRAN.R-project.org/package=tclust`*. Further, little changes to this algorithm yield a generalized version of a robust clustering method introduced by Gallegos (2002), who constrains the scatter matrices' determinants instead of their eigenvalues.*

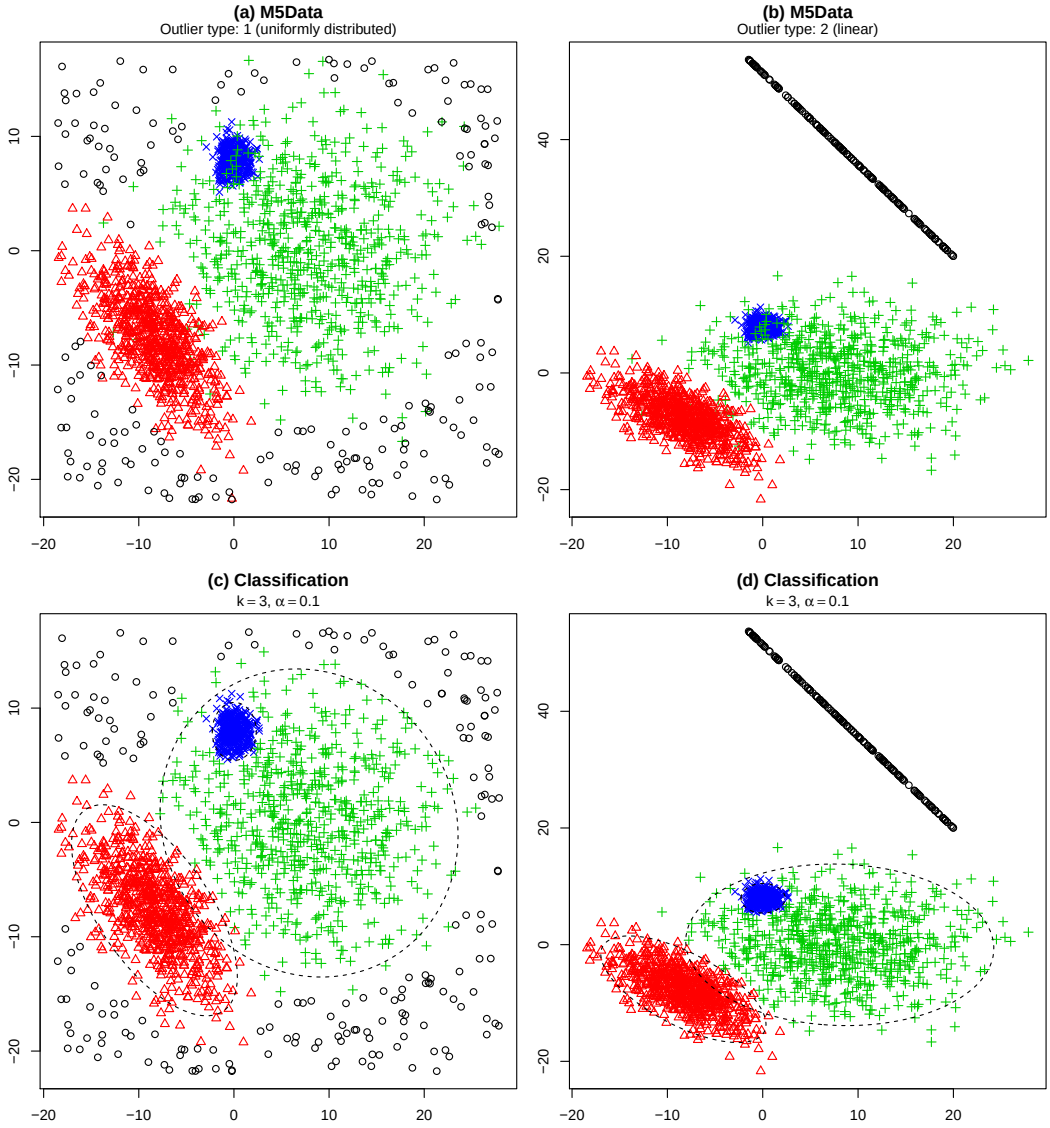

Figure 1: An M5 type data set in two dimensions with uniformly distributed outliers (a) and outliers restricted to a line (b). Plots (c) and (d) show the corresponding clustering results obtained by `tclust`.

4 Simulation Study

As the discussed algorithm has already been compared to other robust clustering approaches on simulated and real data sets (see Fritz *et al.*, 2011), this work concludes with a simulation study investigating the effect of the choice of parameters `iter.max` (number of concentration steps) and `nstart` (number of random initializations) on the performance of the algorithm.

In this simulation study, a so-called M5 type data set is considered, which is based on the "M5 scheme" as introduced in García-Escudero *et al.* (2008). These simulated $p \geq 2$ dimensional data sets consist of three partly overlapping clusters generated from three p-variate normal distributions with means

$$\mu_1 = (0, \beta, 0, \ldots, 0)\,, \mu_2 = (\beta, 0, \ldots, 0) \text{ and } \mu_3 = (-\beta, -\beta, 0, \ldots, 0)\,,$$

with $\beta \in \mathbb{R}^+$ and covariance matrices

$$\Sigma_1 = \operatorname{diag}(1, \ldots, 1), \Sigma_2 = \operatorname{diag}(45, 30, 1, \ldots, 1) \text{ and}$$

$$\Sigma_3 = \begin{pmatrix} 15 & -10 & 0 & \ldots & 0 \\ -10 & 15 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The parameter β specifies how strong the clusters overlap, i.e. smaller values (e.g. 6) yield heavily overlapping clusters, whereas larger values (e.g. 10) yield a better separation of the clusters and thus a problem which is easier to solve. Theoretical cluster weights are fixed as $(0.2, 0.4, 0.4)$, implying that the first cluster size is half of the size of clusters two and three. Further two different types of outliers are considered which are added to the data:

- Type 1: Uniformly distributed outliers in the bounding box of the data.
- Type 2: Uniformly distributed outliers restricted to a random hyperplane of dimension $p - 1$.

All outliers are drawn under the restriction that the squared Mahalanobis distance (see Equation 3.2) of each outlier with respect to all three clusters must be larger than the 0.975 quantile of the chi-squared distribution with p degrees of freedom.

Choosing a number of observations $n = 2000$, parameters $p = 2$ and $\beta = 8$ and a 10% outlier portion results in data sets as shown in Figure 4 (a) and (b) with outlier types 1 and 2 respectively. Considering outlier type 2 in a two dimensional data set reduces the space of the outliers to a line as seen in the mentioned figure. Panels (c) and (d) in the same figure show the corresponding cluster results computed with an R implementation of the described algorithm from package `tclust`. Apparently the cluster structure is captured nicely by the algorithm, only at the boundaries and overlapping regions of the clusters some differences between the theoretical and the computed cluster assignment can be noticed.

For the simulation study the algorithm has been applied on data sets of dimension $p = (2, 6, 10)$, with separation of the cluster determined by $\beta = (6, 8, 10)$ and the two described outlier types on a data set with $n = 2000$, split into three clusters of sizes 360, 720 and 720 and a 10% outlier portion yielding 200 contaminated observations. For each possible combination of these parameters 100 samples have been drawn. Further the `tclust` algorithm has been applied on each of these samples with values $(2, 4, 6, 8, 12, 16, 24, 32, 64)$ for parameters `iter.max` and `nstart`. Moreover, for each of these settings, a very precise "reference result" has been computed with parameters `iter.max = 10000` and `nstart = 200`. All simulations were run on an AMD Phenom II X6 1055T at 2.8GHz.

Figure 2 shows the box plots of the classification errors in percent and runtimes for different values of `iter.max` and the two outlier types, using `nstart` $= 32$, $p = 10$ and $\mu = 6$. The label "X" at the very right of each plot represents the "reference result", which is assumed as to be very close to the theoretically optimal solution. Differences between the outlier types can be seen, as in panel (a) a value of `iter.max` $= 24$ already gives a result very similar to the reference. On the other hand in panel (b), with the outliers restricted

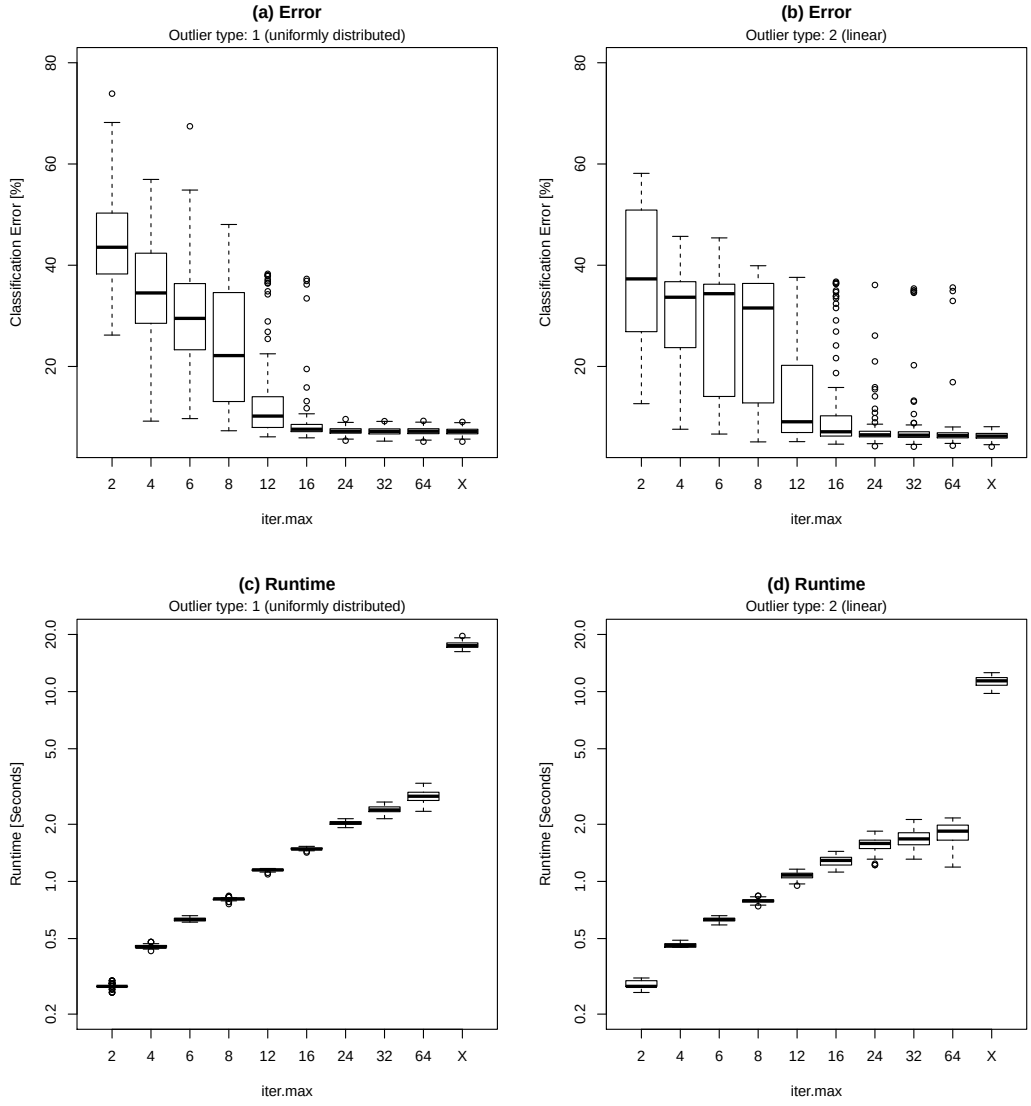

Figure 2: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of `iter.max` and `nstart` $= 32$ when $p = 10$ and $\beta = 6$ are fixed.

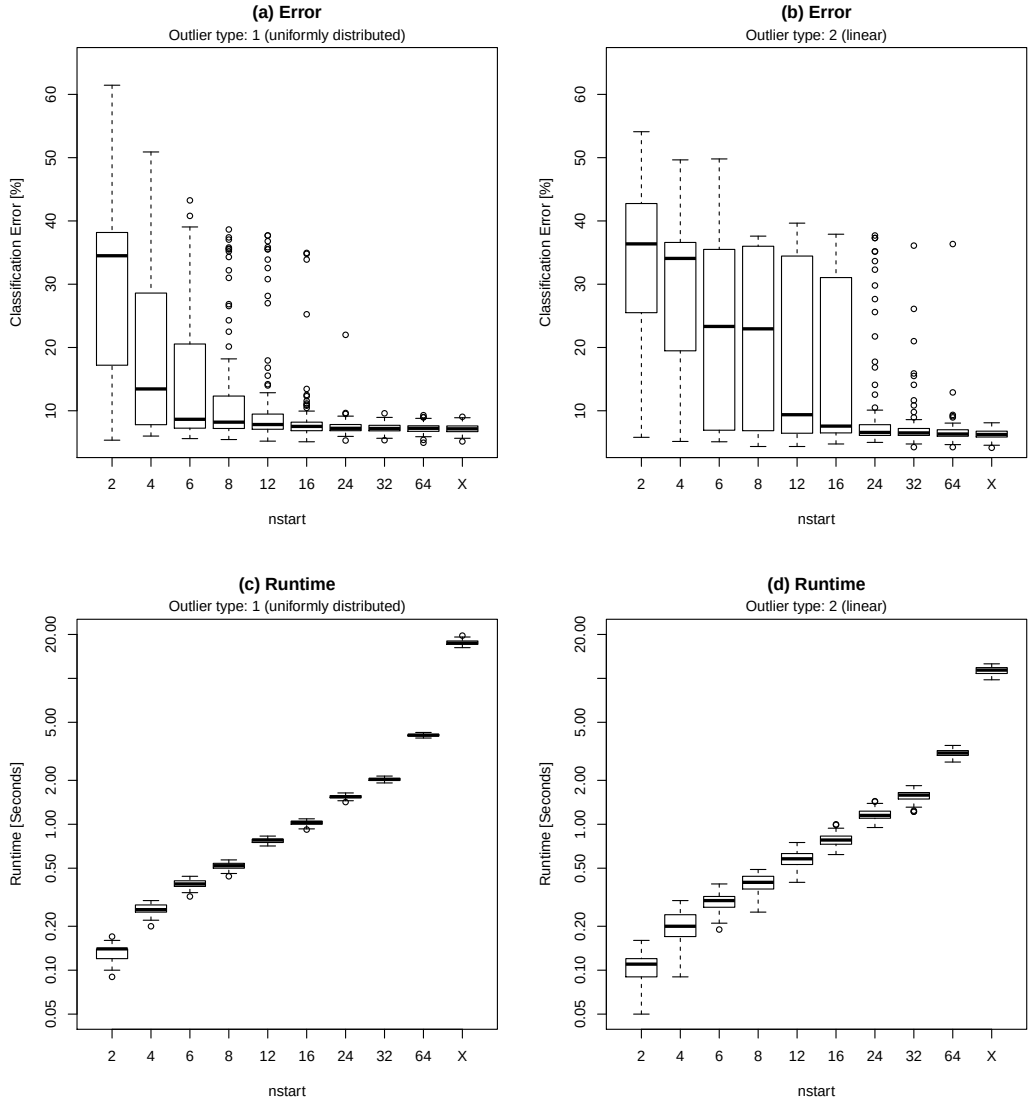

Figure 3: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of `nstart` and `iter.max` $= 24$ when $p = 10$ and $\beta = 6$ are fixed.

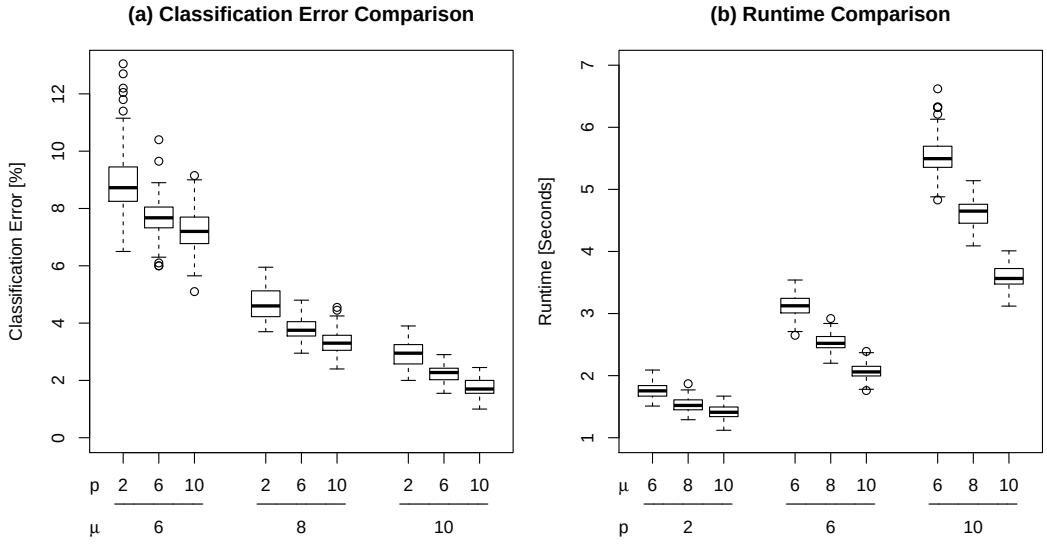

Figure 4: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of p and β and when values `nstart` = 64 and `iter.max` = 64 are fixed.

to a hyperplane of dimension $p - 1$, even a value `iter.max` = 64 yields three out of 100 solutions, which apparently differ from the results in the reference solution "X".

When considering the runtimes in panels (c) and (d) a general pattern can be observed, as at a certain point the runtimes do not increase linearly with the parameter `iter.max` anymore. This is apparently caused by the convergence criterion in Step 2 of the algorithm, which stops the iterations earlier than specified by the chosen value of `iter.max` as soon as the same parameters are obtained within two consecutive concentration steps. The runtimes are quite similar for the different outlier types, however for values `iter.max` larger than 16 the algorithm applied to data contaminated by the second outlier type seems to converge slightly faster. This can be explained, as for the majority of the samples, the second outlier type is easier to grasp. As soon as the cluster structure has been found approximately, the outliers can be identified easily, as most of them do not overlap with the actual clusters. This is not the case with the first outlier scheme. Although the cluster structure can be found quickly, and most of the observations are assigned correctly, the outliers located in the outer regions of the clusters make it more difficult for the algorithm to converge.

Figure 3 shows a similar scenario, but here the parameter `nstart` is varied and the parameters `iter.max` = 24, $p = 10$ and $\mu = 6$ are fixed. When applying the algorithm on data contaminated with the first outlier type, results computed with `nstart = 24` are almost equal to the reference solution "X" as shown in panel (a). However, when the second outlier type is considered, even `nstart` = 64 is not sufficient for obtaining a completely converged solution. The corresponding runtimes as shown in Figure 3 (c) and (d) depend linearly on the parameter `nstart`, as expected. Due to the earlier convergence of the algorithm, when contamination of the second type is present (as commented before), runtimes in panel (d) are slightly lower than in panel (a).

Figure 4 gives classification errors (a) and runtimes (b) for different values of β and p, the first outlier type and values `iter.max` = 64 and `nstart` = 64 fixed. As with increasing

β the clusters are better separable, a larger value of β yields smaller classification errors. Due to the better separation of the clusters the algorithm converges faster when β is large, resulting in lower runtimes.

Also larger values of p decrease the classification error, as in higher dimensional space the clusters are separated more clearly. Further an increase of the number of dimensions clearly increases the runtimes, which is expected due to the algorithms's structure.

5 Conclusions

A feasible algorithm for robust heterogeneous clustering has been presented. The keystone of the algorithm are the constrained concentration steps which successfully combine the concentration steps of the fast-MCD algorithm with Forgy's k-means algorithm. The discussed algorithm computes these constrained concentration steps, only by additionally evaluating an explicit function at $2kp + 1$ values within each iteration. A complete implementation is available in the **R** package `tclust` which is available through the `CRAN` repository.

Appendix: Justification of the algorithm

E-step: Assuming the optimal set of parameters θ to be known, Equation (2.1) implies that the optimal cluster assignments of observations $\{x_1, ..., x_n\}$ can be obtained by using the discriminant functions $D_j(x_i, \theta)$ as described in Step 2.1 of the algorithm.

M-step: Further, it is clear that depending on the cluster assignments (i.e. given $R_0, R_1, ..., R_k$), the values of p_j and m_j maximizing (2.1) are given by

$$p_j = n_j / \lfloor n(1-\alpha) \rfloor \text{ with } n_j = \#R_j \tag{5.1}$$

and

$$m_j = \sum_{i \in R_j} x_i / n_j. \tag{5.2}$$

Let us consider the singular-value decomposition of the sample covariance matrices of observations x_i in group j, given by

$$U_j' D_j U_j = \frac{1}{n_j} \sum_{i \in R_j} (x_i - m_j)(x_i - m_j)', \tag{5.3}$$

where U_j are orthogonal matrices and $D_j = \text{diag}(d_{j1}, d_{j2}, ..., d_{jp})$ are diagonal matrices.

Let S_j be the optimally constrained scatter matrices maximizing (2.1) under restriction (2.2) when $R_0, R_1, ..., R_k$ are known and parameters m_j and p_j are given by (5.1) and (5.2). Analogously to the previous decomposition of the sample covariance matrices, matrices S_j can be split up into $S_j = V_j' D_j^* V_j$, with V_j orthogonal matrices and $D_j^* = \text{diag}(d_{j1}^*, d_{j2}^*, ..., d_{jp}^*)$ diagonal matrices. It can be shown (see García-Escudero *et al.*, 2008) that the eigenvectors of the optimal constrained matrices S_j must be exactly the same as the eigenvectors of the unrestricted sample covariance matrices in (5.3) (i.e., we can set $U_j = V_j$). Thus, we just need to search for the optimal eigenvalues $\{d_{j,l}^*\}$ to obtain the optimal constrained scatter matrices $S_j = U_j' D_j^* U_j$.

Given the eigenvalues $\{d_{j,l}\}$ of the sample covariance matrices in (5.3), the optimal $\{d^*_{j,l}\}$ are obtained by minimizing expression (3.5) when $\{d^*_{j,l}\} \in \Lambda$ with Λ as defined in (3.7). The proof of this claim is almost identical to the proof of Proposition 4 in García-Escudero *et al.* (2008), with the only difference that expression (3.5) in the present work contains the cluster sizes n_j, whereas Equation (3.4) in the mentioned article wrongly did not.

Moreover, notice that Λ can be written as

$$\Lambda = \bigcup_{m \geq 0} \Lambda_m \text{ with } \Lambda_m = \bigcup_{m \geq 0} \big\{ d^*_{jl} : m \leq d^*_{jl} \leq cm \big\}.$$

Thus, for globally minimizing expression (3.5) in Λ, we need to be able to minimize (3.5) when $\{d^*_{j,l}\} \in \Lambda_m$ for every possible value $m > 0$. However, the minimization (for a fixed value of m) can be simplified significantly by considering truncated eigenvalues $d^*_{jl} = d^m_{jl}$ like those in (3.3) which leads us to the minimization of the following target function:

$$\begin{aligned} f : m \ \mapsto \ \sum_{j=1}^{k} n_j \Bigg[& \sum_{l=1}^{p} (\log(m) + d_{jl}/m)(d_{jl} < m) \\ & + \sum_{l=1}^{p} (\log(d_{jl}) + 1)(m \leq d_{jl} < cm) \\ & + \sum_{l=1}^{p} (\log(cm) + d_{jl}/cm)(d_{jl} > cm) \Bigg], \end{aligned} \tag{5.4}$$

which coincides with the target function in (3.4).

Further, f is a continuous differentiable function minimizing in one of its critical values, which satisfy the following fixed point equation:

$$m^* = \frac{\sum_{j=1}^{k} (s_j(m^*) + t_j(m^*)/c)}{\sum_{j=1}^{k} n_j r_j(m^*)}$$

with

$$r_j(m) = \sum_{l=1}^{p} ((d_{jl} < m) + (d_{jl} > cm)),$$

$$s_j(m) = \sum_{l=1}^{p} d_{jl}(d_{jl} < m) \text{ and } t_j(m) = \sum_{l=1}^{p} d_{jl}(d_{jl} > cm).$$

Functions r_j,s_j and t_j take constant values in the intervals $(-\infty, e_1], (e_1, e_2], ..., (e_{2k}, \infty)$. Therefore, we only need to evaluate (5.4) at the $2kp+1$ values $m_1, ..., m_{2kp+1}$ given in Remark 3.

If m_{opt} is the value of m minimizing function f, we finally set the optimal eigenvalues as $d^*_{jl} = d^{m_{\text{opt}}}_{jl}$ to obtain the optimally constrained scatter matrices S_j.

References

Bryant P (1991). "Large-sample Results for Optimization-based Clustering Methods." *Journal of Classification*, **8**(1), 31–44.

Celeux G, Govaert G (1992). "A Classification EM algorithm for clustering and two stochastic versions." *Computational Statistics & Data Analysis*, **14**(3), 315–332.

Cuesta-Albertos J, Gordaliza A, Matrán C (1997). "Trimmed k-means: an Attempt to Robustify Quantizers." *Annals of Statistics*, **25**(2), 553–576.

Dykstra R (1983). "An Algorithm for Restricted Least Squares Regression." *Journal of the American Statistical Association*, **78**(384), 837–842.

Forgy E (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications." *Biometrics*, **21**, 768–780.

Friedman H, Rubin J (1967). "On Some Invariant Criterion for Grouping Data." *Journal of the American Statistical Association*, **63**(320), 1159–1178.

Fritz H, García-Escudero LA, Mayo-Iscar A (2011). "tclust: An R Package for a Trimming Approach to Cluster Analysis." Preprint available at http://cran.r-project.org/web/packages/tclust/vignettes/tclust.pdf.

Gallegos MT (2002). "Maximum likelihood clustering with outliers." In K Jajuga, A Sokolowski, H Bock (eds.), *Classification, Clustering and Data Analysis: Recent advances and applications*, pp. 247–255. Springer-Verlag.

Gallegos MT, Ritter G (2005). "A Robust Method for Cluster Analysis." *Annals of Statistics*, **33**(1), 347–380.

García-Escudero LA, Gordaliza A (1999). "Robustness properties of k-means and trimmed k-means." *Journal of the American Statistical Association*, **94**(447), 956–969.

García-Escudero LA, Gordaliza A, Matrán C (2003). "Trimming Tools in Exploratory Data Analysis." *Journal of Computational and Graphical Statistics*, **12**(2), 434–449.

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2008). "A General Trimming Approach to Robust Cluster Analysis." *Annals of Statistics*, **36**(3), 1324–1345.

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2010). "A Review of Robust Clustering Methods." *Advances in Data Analysis and Classification*, **4**(2-3), 89–109.

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2011). "Exploring the number of groups in robust model-based clustering." *Statistics and Computing*, **Forthcoming**. Preprint available at `http://www.eio.uva.es/infor/personas/langel.html`.

Hathaway RJ (1985). "A Constrained Formulation of Maximum Likelihood Estimation for Normal Mixture Distributions." *Annals of Statistics*, **13**(2), 795–800.

Maronna R, Jacovkis PM (1974). "Multivariate Clustering Procedures with Variable Metrics." *Biometrics*, **30**(3), 499–505.

Neykov N, Filzmoser P, Dimova R, Neytchev P (2007). "Robust Fitting of Mixtures Using the Trimmed Likelihood Estimator." *Computational Statistics & Data Analysis*, **52**(1), 299–308.

Rousseeuw PJ, Van Driessen K (1999). "A Fast Algorithm for the Minimum Covariance Determinant Estimator." *Technometrics*, **41**, 212–223.

Schroeder A (1976). "Analyse d'un mélange de distributions de probabilits de même type." *Rev. Statist. Appl.*, **24**, 39–62.

Scott AJ, Symons MJ (1971). "Clustering Methods Based on Likelihood Ratio Criteria." *Biometrics*, **27**(2), 387–397.

Symons M (1981). "Clustering criteria and multivariate normal mixtures." *Biometrics*, **37**, 35–43.