A fast algorithm for robust constrained clustering

Heinrich Fritz[a], Luis A. García-Escudero[b], Agustín Mayo-Iscar[b]

[a]*Department of Statistics and Probability Theory*
Vienna University of Technology
[b]*Department of Statistics and Operations Research and IMUVA*
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. Despite this coincidence, it is not completely straightforward to combine both algorithms for developing a clustering method which is not severely affected by few outlying observations and being able to cope with non spherical clusters. A sensible way of combining them relies on controlling the relative cluster scatters through constrained concentration steps. With this idea in mind, a new algorithm for the TCLUST robust clustering procedure is proposed which implements such constrained concentration steps in a computationally efficient fashion.

Keywords: Cluster Analysis, Robustness, Impartial trimming, Classification EM algorithm, TCLUST.

1. Introduction

It is easy to realize that there are clear 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 clear key role in Cluster Analysis and in Robust Statistics, respectively. The connection between them mainly refers to the application of the so-called "concentration" steps. Roughly speaking, in these concentration steps, the closest observations to a given center are considered in order to update this center estimate, such that the algorithm searches for regions with a high concentration of observations.

A great drawback when using the k-means method is that it ideally searches for spherically scattered clusters with similar sizes. Further, the presence of a certain fraction of outlying observations could negatively affect its performance (see, e.g., García-Escudero et al., 2010).

Under the previous premises, it seems quite logical to try to combine the clustering ability of k-means with the ability to robustly estimate covariance structures provided by the fast-MCD algorithm.

The trimmed k-means algorithm (García-Escudero et al., 2003) can be seen as a simple combination of k-means and fast-MCD algorithms, where spherical clusters are still assumed. In each concentration step, the proportion α of the most remote observations (considering Euclidean distances) to the previous k centers are discarded. Subsequently, k new centers are obtained by using the group means of the non-discarded observations. Note that the approach simplifies to the well-known Forgy's k-means algorithm when the trimming level α is set to 0. More information on the trimmed k-means approach can be found in Cuesta-Albertos et al. (1997) and García-Escudero and Gordaliza (1999).

It is also a logical step to think about the trimmed k-means algorithm but

considering Mahalanobis distances $(\boldsymbol{x}_i - \boldsymbol{m}_j)' \boldsymbol{S}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{m}_j)$ (as the fast-MCD algorithm does) instead of Euclidean distances. In this case, the centers $\boldsymbol{m}_j$ and the "scatter matrices" $\boldsymbol{S}_j$ (for $j = 1, \ldots, k$) would be updated by computing sample means and sample covariance matrices of the non-discarded observations assigned to each group. Unfortunately, this "naive" combination of algorithms does not provide sensible clustering results, since large clusters tend to "eat" smaller ones. This problem was already noticed in Maronna and Jacovkis (1974) in the untrimmed case ($\alpha = 0$).

For avoiding this drawback, additional constraints are introduced, which limit the difference between the cluster scatters. In fact, many well-known clustering methods implement (implicitly and explicitly) such constraints. For example, the k-means method assumes the same spherical scatter for all the clusters.

Hathaway (1985), in a pioneering work on the mixture fitting framework, proposed constraining the relative differences between cluster scatters through a constant c that controls the strength of the constraints. With this idea in mind, García-Escudero et al. (2008) introduces the TCLUST method which is based on controlling the relative sizes of the eigenvalues of the cluster scatter matrices.

The TCLUST method has good robustness behavior and nice theoretical properties (the existence of solutions for both sample and population problems, together with the consistency of sample solutions to population ones). Unfortunately, from a computational viewpoint, solving the TCLUST problem is not an easy task. Although an algorithm for solving this problem was given in García-Escudero et al. (2008), the most critical issue there was how

to enforce the eigenvalue ratio constraints. This is clearly its computational bottle-neck because a complex optimization problem must be solved in each concentration step. To be more precise, a maximization of a $(k \times p)$-variate function with $\binom{k \times p}{2}$ constraints needs to be solved (k stands for the number of clusters and p for the data dimension). This makes the algorithm computationally unfeasible even for moderate values of k and/or p.

In this work, we present an algorithm for implementing the constrained concentration steps, which clearly speeds up the previous TCLUST algorithm and makes it computationally feasible for practical applications. This algorithm only requires the evaluation of a not very complex function $2pk+1$ times in each concentration step.

The proposed algorithm can be seen as a Classification EM algorithm (Schroeder, 1976; Celeux and Govaert, 1992) and, more generally, as a generalized k-means algorithm (Bock, 2007). Note that the proposed algorithm allows to exactly solve the (constrained) maximization step, which forces the trimmed likelihood target function to increase monotonically through the iterations.

An implementation of the algorithm described in this work is available through the `R` package `tclust` available at `http://CRAN.R-project.org/package=tclust`. A description of how this `R` package can be used in practical applications can be found in Fritz et al. (2012). In this work, we detail the algorithms internally applied by this package.

The methodology behind the discussed approach is explained in Section 2, while the algorithm is presented in Section 3. Section 4 contains a brief simulation study, investigating the performance of the algorithm, and it is

compared to other closely related ones in Section 5. Section 6 explains how this algorithm allows the practical application of exploratory tools which help us to decide on the number of clusters and the trimming level. Section 7 finally presents concluding thoughts.

2. Constrained robust clustering and TCLUST

Given a sample of observations $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ in $\mathbb{R}^p$ and $\phi(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, the probability density function of a p-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, we consider the following general *robust constrained clustering problem* for a fixed trimming level α:

> Search for a partition $R_0, R_1, \cdots, R_k$ of the indices $\{1, \cdots, n\}$ with $\#R_0 = \lceil n\alpha \rceil$, centers $\boldsymbol{m}_1, \cdots, \boldsymbol{m}_k$ in $\mathbb{R}^p$, symmetric positive semidefinite $p \times p$ scatter matrices $\boldsymbol{S}_1, \cdots, \boldsymbol{S}_k$ and weights $p_1, \cdots, p_k$ with $p_j \in [0, 1]$ and $\sum_{j=1}^{k} p_j = 1$, which maximizes
>
> $$\sum_{j=1}^{k} \sum_{i \in R_j} \log \big(p_j \phi(\boldsymbol{x}_i; \boldsymbol{m}_j, \boldsymbol{S}_j) \big). \tag{2.1}$$

Depending on the constraints imposed on the weights p_j and on the scatter matrices $\boldsymbol{S}_j$, the maximization of (2.1) for $\alpha = 0$ leads to well established clustering procedures. For instance, assuming equal weights $p_1 = \cdots = p_k$ and scatter matrices $\boldsymbol{S}_1 = \cdots = \boldsymbol{S}_k = \sigma^2 \boldsymbol{I}$ with $\boldsymbol{I}$ 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 $p_1 = \cdots = p_k$ and $\boldsymbol{S}_1 = \cdots = \boldsymbol{S}_k = \boldsymbol{S}$ with $\boldsymbol{S}$ being a positive definite matrix. In general, the "likelihood" in (2.1) when $\alpha = 0$ and $p_1 = \cdots = p_k$ is often referred to as 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 known as the penalized Classification-Likelihood criterion.

Trimmed alternatives to the previously commented approaches can be constructed by introducing a trimming level $\alpha > 0$ to (2.1), which yields "trimmed likelihoods". This way, for instance, 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. Note 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 level α, can be avoided. Gallegos and Ritter (2005) introduce the so-called "spurious outlier model" that theoretically justifies the use of trimmed likelihoods. It is also important to note that this robust clustering problem reduces to the fast-MCD method when assuming $k = 1$ (i.e. only partitioning the data into $\lceil n\alpha \rceil$ trimmed and $\lfloor n\,(1-\alpha) \rfloor$ regular observations).

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. For instance, a single cluster j made up of only one observation $\boldsymbol{x}_i$ causes (2.1) to tend to infinity by taking a center $\boldsymbol{m}_j = \boldsymbol{x}_i$ and a scatter matrix $\boldsymbol{S}_j$ with $\det(\boldsymbol{S}_j) \to 0$. Thus, partitions containing spurious clusters are quite likely and even preferred to more sensible clustering partitions. This also explains why the previously described "naive" algorithm, combining trimmed k-means and the fast-MCD, does not always work properly.

In order to make the maximization of (2.1) a well defined problem, García-Escudero et al. (2008) proposed to additionally consider an eigenvalue ratio constraint on the scatter matrices $\boldsymbol{S}_1, \cdots, \boldsymbol{S}_k$:

$$\frac{\max_{j,l} \lambda_l(\boldsymbol{S}_j)}{\min_{j,l} \lambda_l(\boldsymbol{S}_j)} \leq c, \tag{2.2}$$

with $\lambda_l(\boldsymbol{S}_j)$ for $l = 1, \cdots, p$, as the set of eigenvalues of the scatter matrix $\boldsymbol{S}_j$ 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 is, the stronger the restriction imposed on the solution, yielding the strongest constraint when $c = 1$.

García-Escudero et al. (2008) proves the existence of solutions for the previously stated robust constrained clustering problem whenever some pathological (non-interesting for robust clustering) data configurations are excluded. To be more precise, data configurations where all data points are concentrated in k points after deleting a fraction α of data points are excluded.

The TCLUST method has good theoretical and robustness properties but no practically applicable algorithm is available yet when $k \times p$ is moderately large. To overcome this drawback, a computationally efficient algorithm for implementing this method will be described in the next section.

3. Algorithm

An algorithm for approximately maximizing (2.1) under the constraint (2.2) was presented in García-Escudero et al. (2008), whereas a significantly

faster approach will be presented here. Further, an inaccuracy in the presentation of this algorithm will be corrected.

Starting with the three steps (E, C and M-) for the Classification EM algorithm described in Celeux and Govaert (1992), we propose the following algorithm:

E-step. For each observation $\boldsymbol{x}_i$ and $D_j(\boldsymbol{x}_i;\theta) = p_j\phi(\boldsymbol{x}_i;\boldsymbol{m}_j,\boldsymbol{S}_j)$, the posterior probabilities

$$\frac{D_j(\boldsymbol{x}_i;\theta)}{\sum_{j=1}^{k} D_j(\boldsymbol{x}_i;\theta)} \text{ for } j = 1,\cdots,k,$$

are computed, with $\theta = (p_1,\cdots,p_k,\boldsymbol{m}_1,\cdots,\boldsymbol{m}_k,\boldsymbol{S}_1,\cdots,\boldsymbol{S}_k)$ as the set of cluster parameters in the current iteration of the algorithm.

C-step. Each non-trimmed observation $\boldsymbol{x}_i$ will be assigned to the cluster which provides maximum posterior probability. In order to implement the trimming procedure, the $\lceil n\alpha \rceil$ observations $\boldsymbol{x}_i$ with smallest values of

$$D(\boldsymbol{x}_i;\theta) = \max\{D_1(\boldsymbol{x}_i;\theta),\cdots,D_k(\boldsymbol{x}_i;\theta)\} \tag{3.1}$$

are discarded as possible outliers (for this iteration).

M-step. The parameters are updated, based on the non-discarded observations and their cluster assignments. At this point, it is crucial to properly enforce the constraints on the cluster scatter matrices.

Note that the distance of an observation $\boldsymbol{x}_i$ to the center of cluster j is quantified throughout $D_j(\boldsymbol{x}_i;\theta)$. The smaller $D_j(\boldsymbol{x}_i;\theta)$, the larger the distance of observation $\boldsymbol{x}_i$ to a center $\boldsymbol{m}_j$. Further, $D(\boldsymbol{x}_i;\theta)$ defines an overall measure for outlyingness. If $k = 1$, then the observations with

the largest (3.1) values are those with the smallest Mahalanobis distances $(\boldsymbol{x}_i - \boldsymbol{m}_j)' \boldsymbol{S}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{m}_j)$ (as considered in the concentration steps of the fast-MCD algorithm). When $k > 1$, $p_1 = \cdots = p_k$ and $\boldsymbol{S}_1 = \cdots = \boldsymbol{S}_k = \sigma^2 \boldsymbol{I}$, the observations with the largest (3.1) values are those with the smallest values of $\min_{j=1,\cdots,k} \|\boldsymbol{x}_i - \boldsymbol{m}_j\|^2$ (as considered in the (trimmed) k-means algorithm).

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

1. *Initialization:* The procedure is initialized `nstart` times by selecting different $\theta^0 = (p_1^0, \cdots, p_k^0, \boldsymbol{m}_1^0, \cdots, \boldsymbol{m}_k^0, \boldsymbol{S}_1^0, \cdots, \boldsymbol{S}_k^0)$. For this purpose, we propose to randomly select $k \times (p+1)$ observations and to accordingly compute k cluster centers $\boldsymbol{m}_j^0$ and k scatter matrices $\boldsymbol{S}_j^0$ from the chosen data points. If needed, the cluster scatter matrix constraints are applied to these $\boldsymbol{S}_j^0$ (as will be described in Step 2.2). Weights $p_1^0, \cdots, p_k^0$ 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., $\theta^{l+1} = \theta^l$) or a maximum number of iterations `iter.max` is reached.
   2.1. *Trimming and cluster assignments (E and C-steps):* Based on the current parameters $\theta^l = (p_1^l, \cdots, p_k^l, \boldsymbol{m}_1^l, \cdots, \boldsymbol{m}_k^l, \boldsymbol{S}_1^l, \cdots, \boldsymbol{S}_k^l)$ the $\lceil n\alpha \rceil$ observations with the smallest values of $D(\boldsymbol{x}_i, \theta^l)$ are discarded. Each remaining observation $\boldsymbol{x}_i$ is then assigned to a cluster j such that $D_j(\boldsymbol{x}_i, \theta^l) = D(\boldsymbol{x}_i, \theta^l)$. This yields a partition $R_0, R_1, \cdots, R_k$ of $\{1, \cdots, n\}$ holding the indexes of the trimmed observations in R_0 and the indexes of the observations belonging to cluster j in R_j for $j = 1, \cdots, k$.

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

$$\boldsymbol{m}_j^{l+1} = \frac{1}{n_j}\sum_{i\in R_j}\boldsymbol{x}_i.$$

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

$$\boldsymbol{T}_j = \frac{1}{n_j}\sum_{i\in R_j}(\boldsymbol{x}_i - \boldsymbol{m}_j^{l+1})(\boldsymbol{x}_i - \boldsymbol{m}_j^{l+1})',$$

may not satisfy the specified eigenvalue ratio constraint. In this case, the spectral decomposition of $\boldsymbol{T}_j = \boldsymbol{U}_j'\boldsymbol{D}_j\boldsymbol{U}_j$ is considered, with $\boldsymbol{U}_j$ being an orthogonal matrix and $\boldsymbol{D}_j = \text{diag}(d_{j1}, d_{j2}, \cdots, d_{jp})$ a diagonal matrix. Let us consider truncated eigenvalues defined as

$$d_{jl}^m = \begin{cases} d_{jl} & \text{if } d_{jl} \in [m, cm] \\ m & \text{if } d_{jl} < m \\ cm & \text{if } d_{jl} > cm \end{cases}, \tag{3.2}$$

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

$$\boldsymbol{S}_j^{l+1} = \boldsymbol{U}_j'\boldsymbol{D}_j^*\boldsymbol{U}_j,$$

with $\boldsymbol{D}_j^* = \text{diag}\left(d_{j1}^{m_{\text{opt}}}, d_{j2}^{m_{\text{opt}}}, \cdots, d_{jp}^{m_{\text{opt}}}\right)$ and m_{opt} minimizing

$$m \mapsto \sum_{j=1}^{k} n_j \sum_{l=1}^{p}\left(\log\left(d_{jl}^m\right) + \frac{d_{jl}}{d_{jl}^m}\right). \tag{3.3}$$

As it will be shown in Proposition 3.2, this expression has to be evaluated only $2kp + 1$ times to exactly find the minimum.

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

The proposed algorithm can be used to solve the maximization of (2.1) when assuming equal weights $p_1 = \cdots = p_k$, by simply setting all weights constantly to $p_j^l = 1/k$ within each iteration. Little changes to this algorithm would also yield a generalized version of a robust clustering method introduced by Gallegos (2002) but relaxing the constraint $\det(\boldsymbol{S}_1) = \cdots = \det(\boldsymbol{S}_k)$ thereby considering determinant ratio constraints as in Section 3.9.1 of McLachlan and Peel (2000). In a similar way, the presented algorithm can also be adapted to develop EM algorithms for constrained mixture fitting problems.

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. The larger the values of `nstart` and `iter.max` are (which of course increase the computational effort), the higher the probability that the algorithm ends up close to the global optimum. 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. Moreover, note that the concentration steps are stopped when the convergence of parameters is achieved in Step 2 of the algorithm. Thus, choosing a higher value than needed of `iter.max` is not too problematic. More insights about how parameters `nstart` and `iter.max` affect the per-

formance of the proposed algorithm will be given in Section 4.

The main novelty of this algorithm, compared to García-Escudero et al. (2008), is how constraints on the eigenvalue ratio are enforced. Equation (3.4) in García-Escudero et al. (2008) constrains eigenvalues by solving the minimization problem

$$(d^*_{11}, d^*_{12}, \cdots, d^*_{jl}, \cdots, d^*_{kp}) \mapsto \sum_{j=1}^{k} n_j \sum_{l=1}^{p} \left(\log\left(d^*_{jl}\right) + \frac{d_{jl}}{d^*_{jl}} \right), \tag{3.4}$$

under the restriction

$$(d^*_{11}, d^*_{12}, \cdots, d^*_{jl}, \cdots, d^*_{kp}) \in \Lambda, \tag{3.5}$$

with Λ as the cone

$$\Lambda = \left\{ d^*_{jl} : d^*_{jl} \leq c \cdot d^*_{rs} \text{ for every } j, r \in \{1, \cdots, k\} \text{ and } l, s \in \{1, \cdots, p\} \right\}. \tag{3.6}$$

This is clearly a more complex problem than minimizing (3.3), as its complexity tremendously increases with the number of clusters k and the dimension p. The problem of minimizing (3.4) in Λ was translated into a quadratic programming problem, which was approximately solved by recursive projections onto cones (Dykstra, 1983). These recursive projections must be carried out in each concentration step and, thus, the algorithm becomes extremely slow and even unfeasible for moderately high values of k and/or p. Moreover, there was a mistake in García-Escudero et al. (2008), as the term n_j in (3.4) was omitted and, thus, the algorithm proposed there can only be applied to similarly sized clusters.

The following proposition serves to justify the new M-step considered in the proposed algorithm:

Proposition 3.1. *If the sets R_j, $j = 1, \cdots, k$, are kept fixed, the maximum of* (2.1) *under constraint (2.2) can be obtained through the following steps:*

(i) The best choice of p_j is $p_j = n_j/\lfloor n(1-\alpha)\rfloor$ with $n_j = \#R_j$.

(ii) Fixed p_j as given in (i), the best choice for $\boldsymbol{m}_j$ is $\boldsymbol{m}_j = \sum_{i\in R_j} \boldsymbol{x}_i/n_j$.

(iii) Fixed the eigenvalues for the matrix $\boldsymbol{S}_j$ and the optimum values given in (i) and (ii) for p_j and $\boldsymbol{m}_j$, the best choice for the set of unitary eigenvectors is the set of unitary eigenvectors of the sample covariance matrix of the observations in R_j.

(iv) With the optimal selections from (i), (ii) and (iii), if d_{jl} are the eigenvalues of the sample covariance matrix, the best choice for the truncated eigenvalues $d_{jl}^{m_{opt}}$ is as in (3.2) *with m_{opt} minimizing function* (3.3)*. Then, the best choice for the scatter matrix $\boldsymbol{S}_j$ is obtained with the eigenvectors of the sample covariance matrix of the observations in R_j and with the optimally truncated eigenvalues.*

Proof. The proofs of statements (i), (ii) and (iii) are included in the proof of Proposition 4 in García-Escudero et al. (2008).

Let us consider the spectral decomposition of the sample covariance matrices of observations given by:

$$\boldsymbol{T}_j = \frac{1}{n_j}\sum_{i\in R_j}(\boldsymbol{x}_i - \boldsymbol{m}_j)(\boldsymbol{x}_i - \boldsymbol{m}_j)' = \boldsymbol{U}_j'\boldsymbol{D}_j\boldsymbol{U}_j, \tag{3.7}$$

where $\boldsymbol{U}_j$ are orthogonal matrices and $\boldsymbol{D}_j = \mathrm{diag}(d_{j1}, d_{j2}, \cdots, d_{jp})$ are diagonal matrices.

Let $\boldsymbol{S}_j$ be the optimally constrained scatter matrices maximizing (2.1) under restriction (2.2) when $R_0, R_1, \cdots, R_k$ are known and parameters $\boldsymbol{m}_j$

and p_j are those given by (i) and (ii). Analogously to the previous decomposition of the sample covariance matrices, matrices $\boldsymbol{S}_j$ can be split up into $\boldsymbol{S}_j = \boldsymbol{V}'_j \boldsymbol{D}^*_j \boldsymbol{V}_j$ with $\boldsymbol{V}_j$ orthogonal matrices and $\boldsymbol{D}^*_j = \text{diag}(d^*_{j1}, d^*_{j2}, \cdots, d^*_{jp})$ diagonal matrices. Statement (iii) tells us that eigenvectors of the optimal constrained matrices $\boldsymbol{S}_j$ must be exactly the same as the eigenvectors of the unrestricted sample covariance matrices in (3.7) (i.e., we can set $\boldsymbol{U}_j = \boldsymbol{V}_j$). We just need to search for the optimal eigenvalues $\{d^*_{j,l}\}$ to obtain the optimally constrained scatter matrices $\boldsymbol{S}_j = \boldsymbol{U}'_j \boldsymbol{D}^*_j \boldsymbol{U}_j$.

Given the eigenvalues $\{d_{j,l}\}$, the optimal $\{d^*_{j,l}\}$ are obtained by minimizing expression (3.4) when $\{d^*_{j,l}\} \in \Lambda$ with Λ as defined in (3.6). The proof of this claim follows from the proof of Proposition 4 in García-Escudero et al. (2008), with the only difference that expression (3.4) now contains the cluster sizes n_j, whereas Equation (3.4) in the mentioned article wrongly did not.

Note that Λ can be written as

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

Thus, for globally minimizing expression (3.4) in Λ, we need to be able to minimize it when $\{d^*_{j,l}\} \in \Lambda_m$ for every possible value $m > 0$. The minimization (for a fixed value of m) can be significantly simplified by considering truncated eigenvalues $d^*_{jl} = d^m_{jl}$ like those in (3.2).

Possible singularities in $\boldsymbol{T}_j$ are not a problem, provided that not all values of d_{jl} are 0 at the same time. Under this mild assumption, it is easy to see that $m > 0$ and this prevents that any value of d^*_{jl} drops to 0 (i.e. no singular clusters are obtained after the truncation of the eigenvalues).

There is a closed form for obtaining m_{opt} (and thus, the constrained

eigenvalues) just by evaluating the function (3.3) $2pk+1$ times:

Proposition 3.2. *Let us consider* $e_1 \leq e_2 \leq \cdots \leq e_{2kp}$ *obtained by ordering the following* $2pk$ *values:*

$$d_{11}, d_{12}, \cdots, d_{jl}, \cdots, d_{kp}, d_{11}/c, d_{12}/c, \cdots, d_{jl}/c, \cdots, d_{kp}/c,$$

and, $f_1, \cdots, f_{2kp+1}$ *any values satisfying:*

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

We can choose m_{opt} *as the value of:*

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

$i = 1, \cdots, 2kp+1$, *yielding the minimum value of* (3.3).

Proof. Firstly, let us rewrite the target function (3.3) as

$$\begin{aligned} f : m \;\mapsto\; \sum_{j=1}^{k} n_j \Bigg[& \sum_{l=1}^{p} (\log(m) + d_{jl}/m)(d_{jl} < m) \\ & + \sum_{l=1}^{p} (\log(d_{jl}) + 1)(m \leq d_{jl} < cm) \\ & + \sum_{l=1}^{p} (\log(cm) + d_{jl}/cm)(d_{jl} > cm) \Bigg]. \end{aligned} \tag{3.8}$$

Since f is a continuously differentiable function, it minimizes in one of its critical values, which satisfies 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 $(-\infty, e_1], (e_1, e_2]$, $\cdots, (e_{2k}, \infty)$. Therefore, we only need to evaluate (3.8) at the $2kp+1$ values $m_1, \cdots, m_{2kp+1}$.

4. Simulation study

In this section, a small simulation study is presented, 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.

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

$$\boldsymbol{\mu}_1 = (0, \beta, 0, \ldots, 0), \boldsymbol{\mu}_2 = (\beta, 0, \ldots, 0) \text{ and } \boldsymbol{\mu}_3 = (-\beta, -\beta, 0, \ldots, 0),$$

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

$$\boldsymbol{\Sigma}_1 = \text{diag}(1, \ldots, 1), \boldsymbol{\Sigma}_2 = \text{diag}(45, 30, 1, \ldots, 1) \text{ and}$$

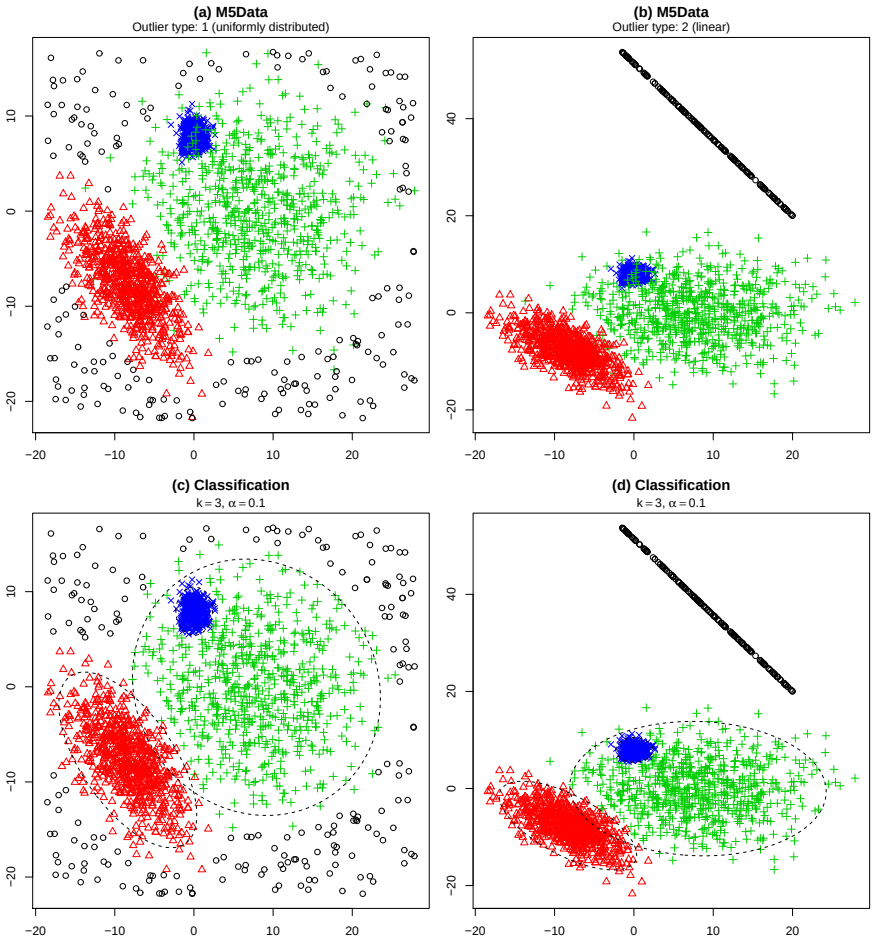

Figure 1: An M5 type data set in two dimensions with uniformly distributed outliers (a) and outliers restricted to a line (b). Plots (c) and (d) show the corresponding clustering results obtained by `tclust`.

$$\mathbf{\Sigma}_3 = \begin{pmatrix} 15 & -10 & 0 & \dots & 0 \\ -10 & 15 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

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 the size of clusters two and three. Two further 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 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 1 (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; however, 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 to 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 $n = 2000$,

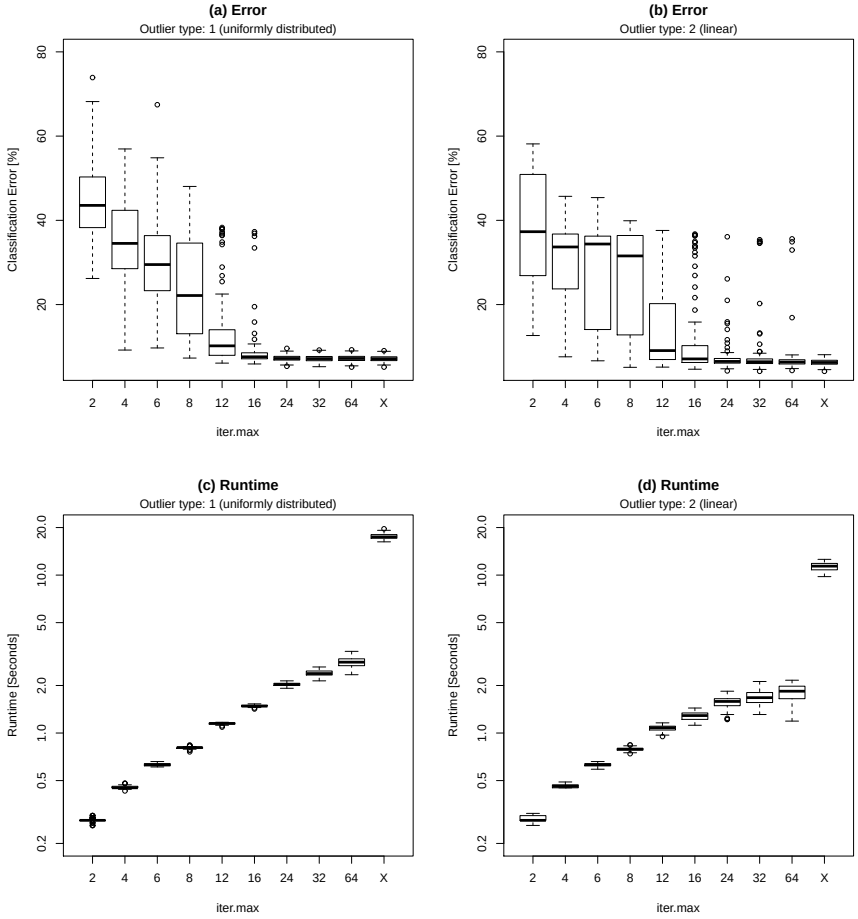

Figure 2: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of `iter.max` and `nstart` $= 32$ when $p = 10$ and $\beta = 6$ are fixed.

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. In addition, the `tclust` algorithm has been applied to 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 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 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 no longer increase linearly with the parameter `iter.max`. 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

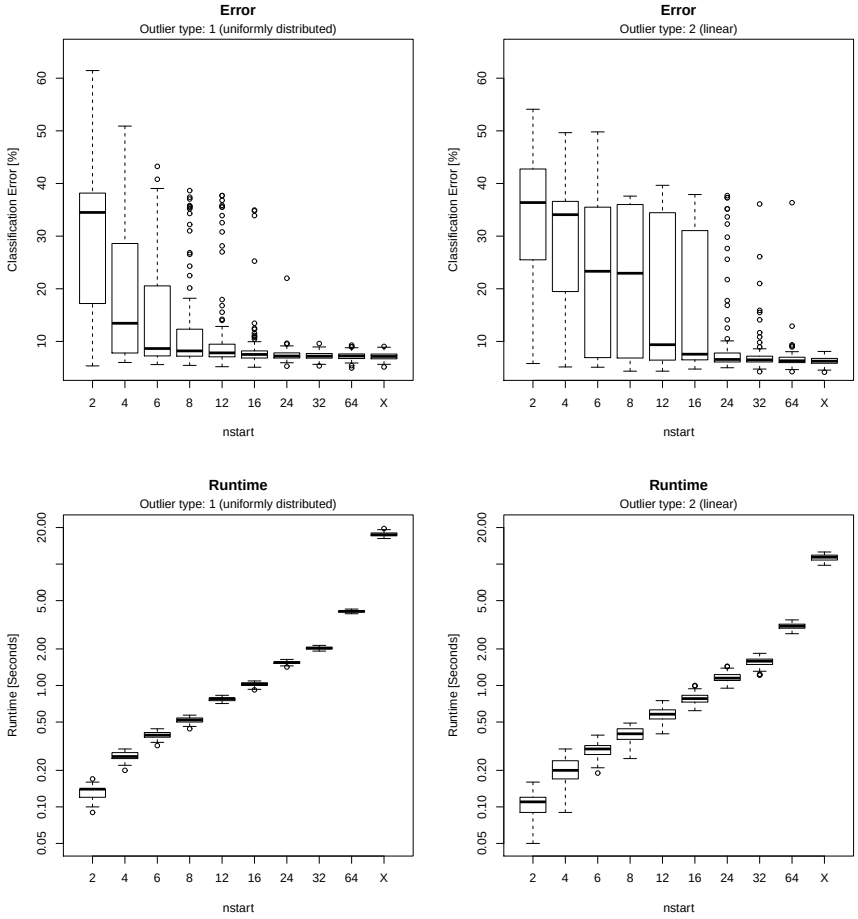

Figure 3: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of `nstart` and `iter.max` $= 24$ when $p = 10$ and $\beta = 6$ are fixed.

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 to obtain 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 more easily 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. In addition, an increase in the number of dimensions clearly increases the runtimes, which is expected due to the algorithm structure.

5. Relationships with other approaches

If the constraints on the eigenvalues were not considered (for instance fixing a very large value of constant c) and equal weights $p_1 = \cdots = p_k$ were assumed, the algorithm would essentially coincide with the one proposed by

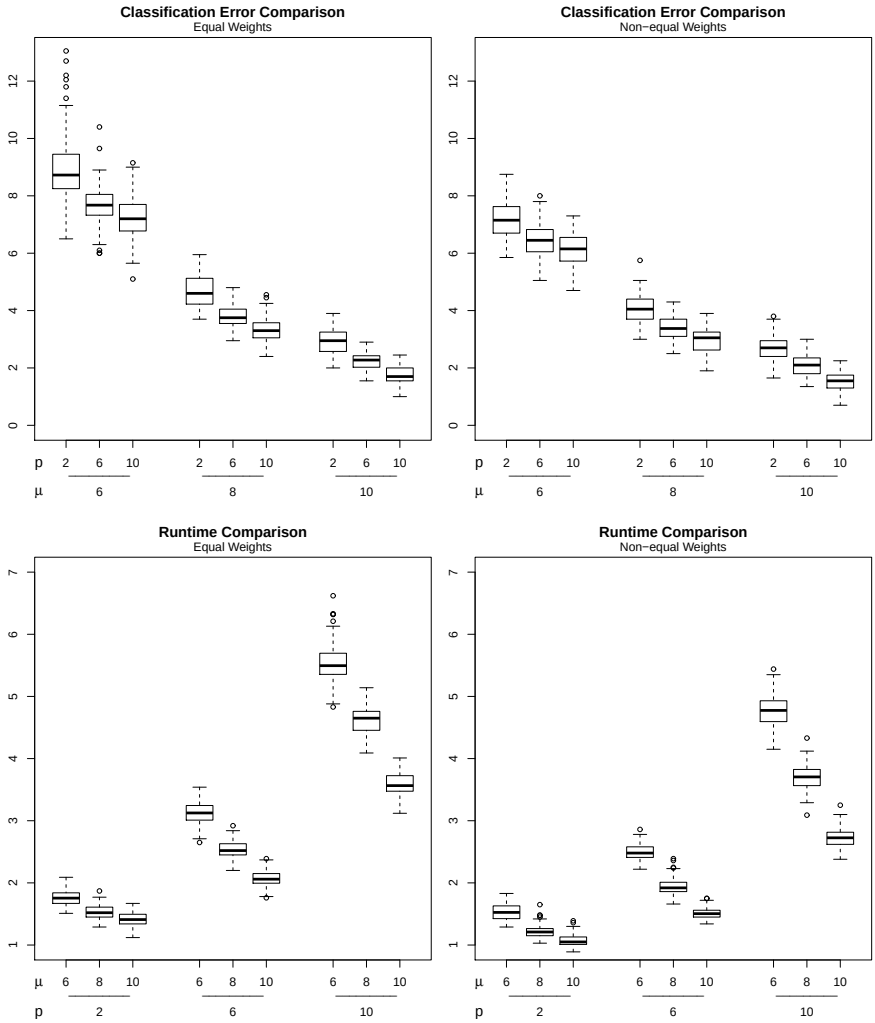

Figure 4: Classification errors and runtimes of the `tclust` algorithm applied to simulated M5 type data sets for different values of p and β and when values `nstart` $= 64$ and `iter.max` $= 64$ are fixed.

Neykov et al. (2007) when the Classification EM approach (not the mixture fitting one) is applied there. As already mentioned, explicitly stating relative cluster scatter constraints and providing a computationally efficient

procedure for solving them have key importance in the presented approach to robust clustering. Note that this yields a wide range of different clustering solutions depending on the choice of constant c. These solutions range from almost spherical clusters when c is close to 1 to more unrestricted ones when c is huge. The researcher may choose from among them, depending on the clustering application in mind. Moreover, since both approaches share a similar type of target function (i.e., (2.1) with $p_1 = \cdots = p_k$), it is easy to see that Neykov et al. (2007)'s target function is unbounded too and, if no constraints are posed, certain precautions in the associated algorithm are clearly needed in order to avoid the degeneration of the clustering solution. Neykov et al. (2007) also considers other interesting finite mixtures statistical problems where trimmed likelihoods can be successfully applied if robustness is a major concern.

Gallegos and Ritter (2009) have also considered a trimmed likelihood approach with scatter matrix constraints. They propose to apply Hathaway (1985)'s original extension of his univariate constraints to multivariate problems by constraining

$$\min_{l} \min_{1 \leq h \neq j \leq k} \lambda_l(\boldsymbol{S}_h \boldsymbol{S}_j^{-1}) \geq \frac{1}{c} \text{ with } c \geq 1. \tag{5.1}$$

However, these constraints were not directly enforced by the algorithm. They propose to obtain all possible local maxima of the trimmed likelihood (with a similar algorithm to that found in Neykov et al. (2007) or ours without considering any constraints) and, afterwards, the ratio in (5.1) and the value of the trimmed likelihood for these local maxima are monitored in order to choose sensible candidate clustering solutions. Gallegos and Ritter (2010) also deal with the unboundedness of the trimmed likelihood by controlling

the smaller cluster sizes, which implies solving a λ-assignment problem in the concentration steps.

In a mixture fitting framework without trimming, Ingrassia and Rocci (2007) proposed algorithms for addressing constrained mixture likelihood maximization. They give an interesting discussion starting from the constraint (5.1) and ending with the same type of constraints as in (2.2). They propose an algorithm based on truncating scatter matrices eigenvalues when lower and upper bounds on these eigenvalues are known. Relaxing this assumption, when no suitable external information is available for bounding them, they also consider a bound on the ratio of the eigenvalues. However, their algorithm for this last proposal does not directly maximize the likelihood as is done in Step 2.2 of our algorithm. Rather, it is based on obtaining iterative estimates of a lower bound on the scatter matrices eigenvalues η needed in order to properly truncate the eigenvalues.

Another possibility for dealing with noise in Cluster Analysis is based on trying to "fit" noisy observations through the consideration of additional mixture components, as for instance, the approach followed by the MCLUST method (see, e.g., Fraley and Raftery, 1998). This well established approach is based on adding a uniformly distributed mixture component to accommodate the presence of background noise. Although this approach provides clear robustness in many problems, its performance may depend on whether the "uniformly distributed" assumption for the noise approximately holds. For instance, in a very extreme case, this approach breaks down with only one observation placed in a very remote position as Hennig (2004) showed. On the other hand, the TCLUST procedure does not impose a specific model

for noise, which explains its good robustness performance (Ruwet et al., 2012a,b).

6. Computation of Classification Trimmed Likelihood curves

One of the main motivations for a fast algorithm lies in the graphical tools introduced in García-Escudero et al. (2011) (see also Fritz et al., 2012), which help to make appropriate choices for the number of clusters k and the trimming level α. The practical application of these tools, as e.g. the Classification Trimmed Likelihood curves, implies solving many TCLUST problems for different values of α and k. This clearly turns out to be unfeasible without a computationally efficient algorithm at hand.

Of course, the determination of k and α is not a well-defined problem and, usually, several choices are arguable for these two parameters. For instance, the question of whether small subsets of isolated observations are actually clusters or only outliers is in many cases philosophical and usually related to the context in which a clustering problem is considered. In any case, the Classification Trimmed Likelihood curves provide helpful guidance for obtaining a small set of sensible choices for k and α which have to be carefully evaluated by the researcher.

To exemplify this, let us consider a data set like in Figure 1,(b) and (d), and the Classification Trimmed Likelihood curves in Figure 5,(a), which plot the chosen trimming level α against the maximum value attained by the objective function (2.1) for a set of values of k (see García-Escudero et al., 2011). If there was a reason to fix the number of clusters to $k = 3$, 10% of observations which are restricted to a line may be considered as outliers.

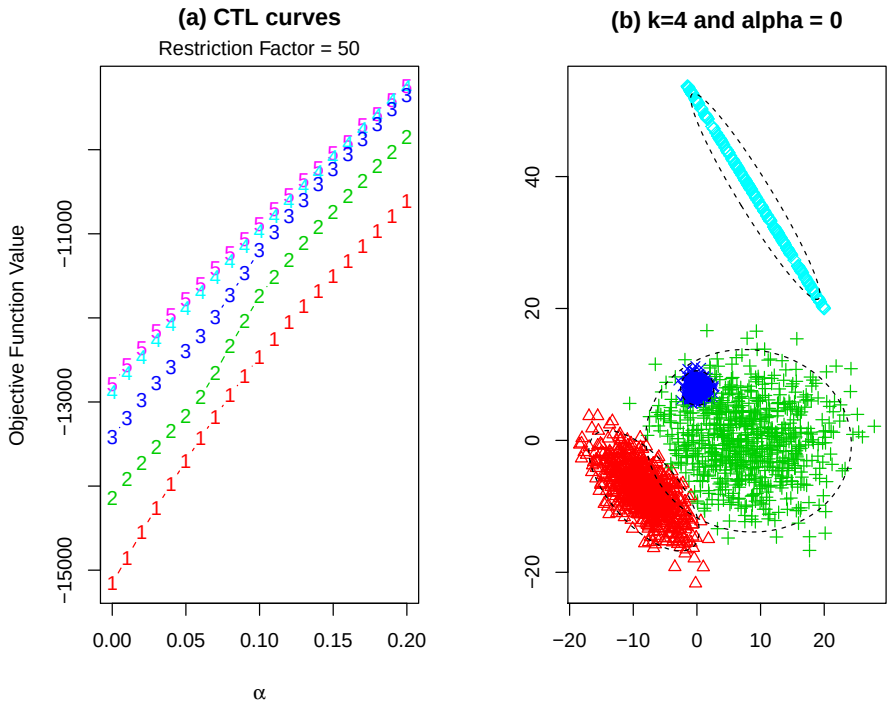

Figure 5: (a) Classification Trimmed Likelihood curves for a data set like in 1,(b) and (d). (b) Clustering solution for this data set when $k = 4$ and $\alpha = 0$.

Further, if no trimming is allowed, the Classification Trimmed Likelihood curves imply that $k = 4$ is a more sensible choice in this example than $k = 3$ whereas considering $k > 4$ does not further increase the solution's quality. This can be seen in Figure 5,(a), as the objective function's value is almost equal for $k = 4$ and $k = 5$ under the restriction $\alpha = 0$. The solution given by the proposed algorithm for $k = 4$ and $\alpha = 0$ is shown in Figure 5,(b). A large value of $c = 50$ is considered, such that the "cluster" made up by the observations restricted to a line (collinear data points) is properly detected. Further, the Classification Trimmed Likelihood curves show that increasing k from 3 to 4 would not result in a better solution, if a trimming level of

$\alpha = 0.1$ is considered. A detailed discussion on the interpretation of the Classification Trimmed Likelihood is given in García-Escudero et al. (2011) and Fritz et al. (2012).

7. Conclusions

A computationally feasible algorithm for robust heterogeneous clustering has been presented. The keystone of the proposed algorithm is the consideration of constrained concentration steps which combine elements of the fast-MCD algorithm with Forgy's k-means algorithm, but enforce constraints on the ratio of the scatter matrices' eigenvalues. This is done by additionally evaluating an explicit function at $2kp + 1$ values within each concentration step. The presented algorithm was implemented in the `R` package `tclust`.

References

Bock, H.-H., 2007. Clustering methods: A history of k-means algorithms. In: Brito, P., Bertrand, B., Cucumel, G., de Carvalho, F. (Eds.), Selected Contributions in Data Analysis and Classification. Studies in Classification, Data Analysis, and Knowledge Organization, Berlin, Heidelberg, pp. 161–172.

Bryant, P., 1991. Large-sample results for optimization-based clustering methods. J Classif 8, 31–44.

Celeux, G., Govaert, A., 1992. A classification em algorithm for clustering and two stochastic versions. Comput Stat Data An 14, 315–332.

Cuesta-Albertos, J., Gordaliza, A., Matrán, C., 1997. Trimmed k-means: an attempt to robustify quantizers. Ann Stat 25, 553–576.

Dykstra, R., 1983. An algorithm for restricted least squares regression. J Am Stat Assoc 78, 837–842.

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

Fraley, C., Raftery, A. E., 1998. How many clusters? which clustering method? answers via model-based cluster analysis. The Computer J 41 (8), 578–588.

Friedman, H., Rubin, J., 1967. On some invariant criterion for grouping data. J Am Stat Assoc 63, 1159–1178.

Fritz, H., García-Escudero, L., Mayo-Iscar, A., 2012. tclust: An r package for a trimming approach to cluster analysis. J Stat Softw 47 (12).
URL `http://www.jstatsoft.org/v47/i12`

Gallegos, M., Ritter, G., 2005. A robust method for cluster analysis. Ann Stat 33, 347–380.

Gallegos, M., Ritter, G., 2009. Trimming algorithms for clustering contaminated grouped data and their robustness. Adv Data Anal Classif 10, 135–167.

Gallegos, M., Ritter, G., 2010. Using combinatorial optimization in model-based trimmed clustering with cardinality constraints. Comput Stat Data An 54, 637–654.

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

García-Escudero, L., Gordaliza, A., 1999. Robustness properties of k-means and trimmed k-means. J Am Stat Assoc 94, 956–969.

García-Escudero, L., Gordaliza, A., Matrán, C., 2003. Trimming tools in exploratory data analysis. J Comput Graph Stat 12, 434–449.

García-Escudero, L., Gordaliza, A., Matrán, C., Mayo-Iscar, A., 2008. A general trimming approach to robust cluster analysis. Ann Stat 36, 1324–1345.

García-Escudero, L., Gordaliza, A., Matrán, C., Mayo-Iscar, A., 2010. A review of robust clustering methods. Adv Data Anal Classif 4, 89–109.

García-Escudero, L., Gordaliza, A., Matrán, C., Mayo-Iscar, A., 2011. Exploring the number of groups in robust model-based clustering. Stat Comput 21, 585–599.

Hathaway, R., 1985. A constrained formulation of maximum likelihood estimation for normal mixture distributions. Ann Stat 13, 795–800.

Hennig, C., 2004. Breakdown points for maximum likelihood-estimators of location-scale mixtures. Ann Stat 32 (4), 1313–1340.

Ingrassia, S., Rocci, R., 2007. Constrained monotone em algorithms for finite mixture of multivariate gaussians. Comput Stat Data An 51, 5339–5351.

Maronna, R., Jacovkis, P., 1974. Multivariate clustering procedures with variable metrics. Biometrics 30, 499–505.

McLachlan, G., Peel, D., 2000. Finite mixture models. Wiley Series in Probability and Statistics, New York.

Neykov, N., Filzmoser, P., Dimova, R., Neytchev, P., 2007. Robust fitting of mixtures using the trimmed likelihood estimator. Comput Stat Data An 52, 299–308.

Rousseeuw, P., Van Driessen, K., 1999. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41, 212–223.

Ruwet, C., García-Escudero, L. A., Gordaliza, A., Mayo-Iscar, A., 2012a. The influence function of the tclust robust clustering procedure. Adv. Data Anal. Classif 6 (2), 107–130.

Ruwet, C., García-Escudero, L. A., Gordaliza, A., Mayo-Iscar, A., 2012b. On the breakdown behavior of robust constrained clustering procedures. Submitted, preprint avilable at `http://orbi.ulg.ac.be/handle/2268/104215`.
URL `http://orbi.ulg.ac.be/handle/2268/104215`

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

Scott, A., Symons, M., 1971. Clustering methods based on likelihood ratio criteria. Biometrics 27, 387–397.

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