tclust: An R Package for a Trimming Approach to Cluster Analysis

Heinrich Fritz
Department of Statistics and Probability Theory,
Vienna University of Technology
and
Luis A. García-Escudero
Departamento de Estadística e Investigación Operativa
Universidad de Valladolid
and
Agustín Mayo-Iscar
Departamento de Estadística e Investigación Operativa
Universidad de Valladolid

April 30, 2011

Abstract

Outlying data can heavily influence standard clustering methods. At the same time, clustering principles can be useful when robustifying statistical procedures. These two reasons motivate the interest in developing feasible robust model-based clustering approaches. With this in mind, an R package for performing non-hierarchical robust clustering, called **tclust** is presented here. Instead of trying to "fit" noisy data, a proportion α of the most outlying observations is trimmed. The **tclust** package efficiently handles different cluster scatter constraints. Graphical exploratory tools are also implemented to help the user make sensible choices for the trimming proportion as well as the number of clusters to search for.

Keywords: Model-based clustering, trimming, heterogeneous clusters

1 Introduction to robust clustering and tclust

Methods for cluster analysis are basically aimed at detecting homogeneous clusters with large heterogeneity among them. As happens with other (non-robust) statistical procedures, clustering

methods may be heavily influenced by even a small fraction of outlying data. For instance, due to outlying observations, two or more clusters might artificially be joined or "spurious" non-informative clusters may be made up by only a few outlying observations (see, e.g. García-Escudero and Gordaliza, 1999; García-Escudero *et al.*, 2010b). Therefore, the application of robust methods in this context is very advisable, especially in fully automatized clustering (unsupervised learning) problems. Certain relations between cluster analysis and robust methods (Rocke and Woodruff, 2002; Hardin and Rocke, 2004; García-Escudero *et al.*, 2003; Woodruff and Reiners, 2004) are also a motivation for the interest of robust clustering techniques. For instance, robust clustering techniques can be used to handle "clusters" of highly concentrated outliers which are specially dangerous in (robust) estimation. García-Escudero *et al.* (2010b) provides a recent survey of robust clustering methods.

The **t**clust package for the **R** environment for statistical computing (**R** Development Core Team, 2010) implements different robust non-hierarchical clustering algorithms where trimming plays a key role. This package is available at `http://CRAN.R-project.org/package=tclust`. As trimming allows to remove a fraction α of the "most outlying" data, the strong influence of outlying observations can be avoided and robustness naturally arises. This trimming approach to clustering has been introduced in Cuesta-Albertos *et al.* (1997), Gallegos (2002), Gallegos and Ritter (2005) and García-Escudero *et al.* (2008). Trimming also serves to highlight interesting anomalous observations.

Trimming is not a new concept in statistics. For instance, the widely used trimmed mean for one dimensional data removes a proportion $\alpha/2$ of the largest, and a proportion $\alpha/2$ of the smallest observations before computing the mean. However, it is not straightforward to extend this philosophy to cluster analysis, because most of these problems are of multivariate nature. Moreover, it is often the case that "bridge points" lying between clusters ought to be trimmed. Instead of forcing the statistician to define the regions to be trimmed in advance, the procedures implemented in **t**clust take the whole data structure into account in order to decide which parts of the sample should be discarded. By considering this type of trimming, these procedures are even able to trim outlying bridge points. The "self-trimming" philosophy behind these procedures is exactly the same as adopted by some well-known high breakdown-point methods (see, e.g., Rousseeuw and Leroy, 1987).

As a first example of this trimming approach, let us consider the trimmed k-means method introduced in Cuesta-Albertos *et al.* (1997). The function `tkmeans` from the **t**clust package implements this method. In the following example, this function is applied to a bivariate data set based on the Old Faithful geyser called `geyser2` that accompanies the **t**clust package. The code given below creates Figure 1:

```
R > library ("tclust")
R > data ("geyser2")
R > clus <- tkmeans (geyser2, k = 3, alpha = 0.03)
R > plot (clus)
```

In the data set `geyser2`, we are searching for $k = 3$ clusters and a proportion $\alpha = 0.03$ of the data is trimmed. The clustering results are shown in Figure 1. Among this 3% of trimmed data, we can see 6 anomalous "short followed by short" eruptions lengths. Notice that an observation situated between the clusters is also trimmed.

The package presented here adopts a "crisp" clustering approach, meaning that each observation is either trimmed or fully assigned to a cluster. In comparison, mixture approaches estimate a cluster pertinence probability for each observation. Robust mixture alternatives have also been proposed where noisy data is tried to be fitted through additional mixture components. For instance, package **m**clust (Banfield and Raftery, 1993; Fraley and Raftery, 1998) and package **f**lexmix (Leisch, 2004; McLachlan and Peel, 2000) implement such robust mixture fitting approaches. Mixture fitting results can be easily converted into a "crisp" clustering result by converting the cluster pertinence probabilities into 0-1 probabilities. Contrary to these mixture fitting approaches, the procedures implemented in the **t**clust package simply remove outlying observations and do not intend to fit them at all. Package **t**lemix (see Neykov *et al.*, 2007) also implements a closely related trimming approach. As described in Section 3, the **t**clust package focuses on offering adequate cluster scatter matrix constraints to avoid the occurrence of spurious non-interesting clusters. In contrast, the **t**lemix mainly controls the minimum number of observations in a cluster. More comments explaining the differences of the approach followed in the **t**clust package with respect to other alternatives can be found in García-Escudero *et al.* (2010a).

The outline of the paper is as follows: In Section 2 we briefly review the so-called "spurious outliers" model and show how to derive two different clustering criteria from it. Different constraints

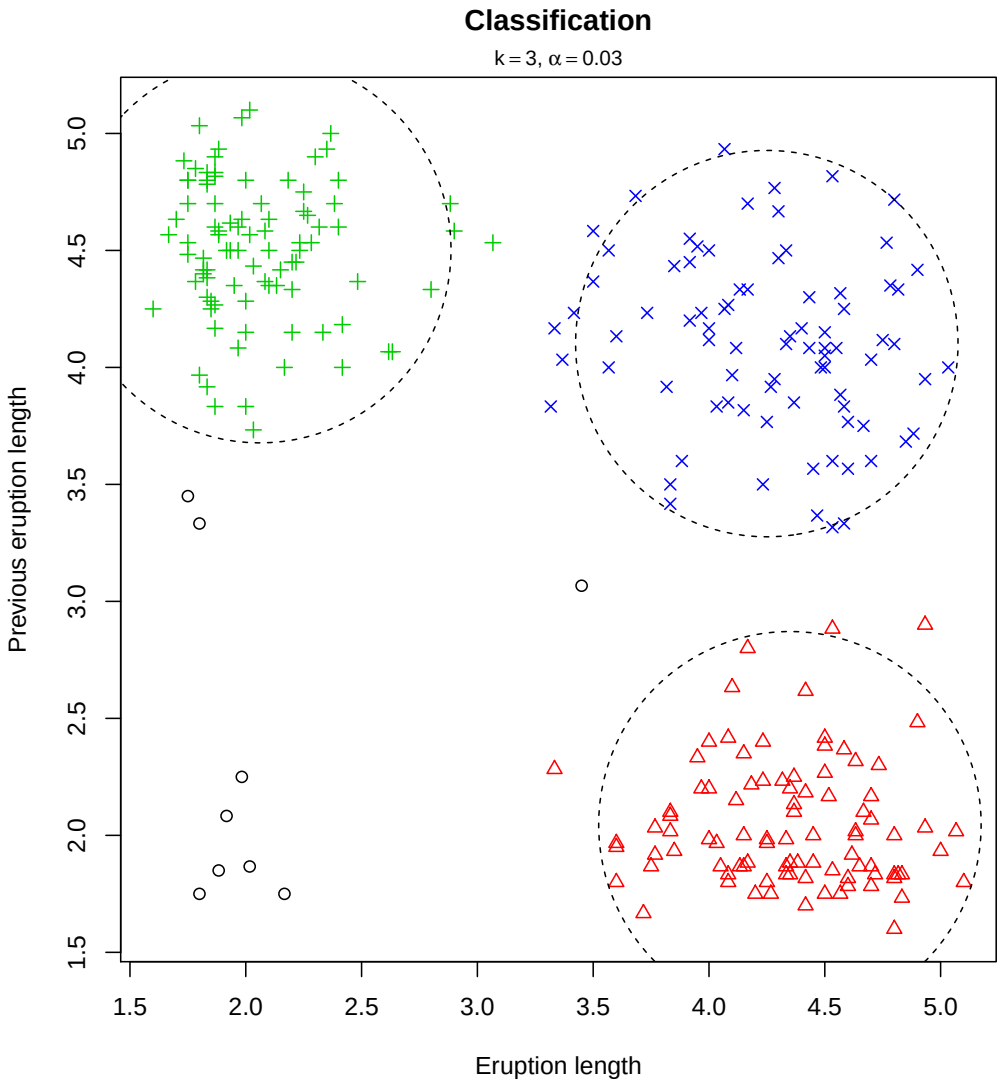

Figure 1: Trimmed k-means results with $k = 3$ and $\alpha = 0.03$ for the bivariate Old Faithful Geyser data. Trimmed observations are denoted by the symbol "○".

on the cluster scatter matrices and their implementation in the **tclust** package are commented in Section 3. Section 4 presents the numerical output returned by this package. Some brief comments concerning the implemented algorithms are given in Section 5. Section 6 shows some graphical outputs that help us to make sensible choices for the number of clusters and trimming proportion. Other useful plots summarizing the robust clustering results are shown in Section 7. Finally, Section 8 applies the **tclust** package to a well-know real data set.

2 Trimming and the spurious outliers model

Gallegos (2002) and Gallegos and Ritter (2005) propose the "spurious outliers model" as a probabilistic framework for robust crisp clustering. Let $f(\cdot;\mu,\Sigma)$ denote the probability density function of the p-variate normal distribution with mean μ and covariance matrix Σ. The "spurious-outlier model" is defined through "likelihoods" like

$$\left[\prod_{j=1}^{k}\prod_{i\in R_j} f(x_i;\mu_j,\Sigma_j)\right]\left[\prod_{i\in R_0} g_i(x_i)\right] \tag{1}$$

with $\{R_0,...,R_k\}$ being a partition of the set of indices $\{1,2,...,n\}$ such that $\#R_0 = \lceil n\alpha\rceil$. R_0 are the indices of the "non-regular" observations generated by other density functions g_i. "Non-regular" observations can be clearly considered as "outliers" if we assume certain sensible assumptions for the g_i (see details in Gallegos, 2002; Gallegos and Ritter, 2005). Under these assumptions, the search of a partition $\{R_0,...,R_k\}$ with $\#R_0 = \lceil n\alpha\rceil$, vectors μ_j and positive definite matrices Σ_j maximizing (1) can be simplified to the same search but just maximizing the simpler expression

$$\sum_{j=1}^{k}\sum_{i\in R_j} \log f(x_i;\mu_j,\Sigma_j). \tag{2}$$

Notice that observations x_i with $i \in R_0$ are not taken into account in (2). Maximizing (2) with $k = 1$ yields the Minimum Covariance Determinant (MCD) estimator (Rousseeuw, 1985).

Unfortunately, the direct maximization of (2) is not a well-defined problem when $k > 1$. It is easy to see that (2) is unbounded without any constraint on the cluster scatter matrices Σ_j. The `tclust` function from the **tclust** package approximately maximizes (2) under different cluster scatter matrix constraints which will be shown in Section 3.

The maximization of (2) implicitly assumes equal cluster weights. In other words, we are ideally searching for clusters with equal sizes. The function `tclust` provides this option by setting the argument `equal.weights = TRUE`. Alternatively different cluster sizes or cluster weights can be considered by searching for a partition $\{R_0,...,R_k\}$ (with $\#R_0 = \lceil n\alpha\rceil$), vectors μ_j, positive definite matrices Σ_j and weights $\pi_j \in [0,1]$ maximizing

$$\sum_{j=1}^{k}\sum_{i\in R_j} (\log \pi_j + \log f(x_i;\mu_j,\Sigma_j)). \tag{3}$$

The (default) option `equal.weights = FALSE` is used in this case. Again, the scatter matrices have to be constrained in the same way.

	`equal.weights = TRUE`	`equal.weights = FALSE`
`restr = "eigen"`	*k-means* Cuesta-Albertos *et al.* (1997)	García-Escudero *et al.* (2008)
`restr = "deter"`	Gallegos (2002)	This work
`restr = "sigma"`	*Friedman and Rubin* (1967) Gallegos and Ritter (2005)	This work

Table 1: Clustering methods handled by **t**clust. Names in cursive letters are untrimmed ($\alpha = 0$) methods.

3 Constraints on the cluster scatter matrices

As already mentioned, the function `tclust` implements different algorithms aimed at approximately maximizing (2) and (3) under different types of constraints which can be applied on the scatter matrices Σ_j. The type of constraint is specified by the argument `restr` of the `tclust` function. Table 1 gives an overview of the different clustering approaches implemented by the `tclust` function depending on the chosen type of constraints.

Imposing constraints is compulsory because maximizing (2) or (3) without any restriction is not a well-defined problem. Notice that an almost degenerated scatter matrix Σ_j would cause trimmed log-likelihoods (2) and (3) to tend to infinity. This issue can cause a (robust) clustering algorithm of this type to end up finding "spurious" clusters almost lying in lower dimensional subspaces. Moreover, the resulting clustering solutions might heavily depend on the chosen constraint. The strength of the constraint is controlled by the argument `restr.fact` ≥ 1 in the `tclust` function. The smaller `restr.fact`, the stronger the scatter matrices are restricted. Values of `restr.fact` close to 1 imply very "equally scattered" clusters.

Also arising from the spurious outlier model, other types of constraints have recently been introduced by Gallegos and Ritter (2009, 2010). These (closely related) constraints also serve to avoid degeneracy of trimmed likelihoods but they are not implemented in the current version of the **t**clust package.

3.1 Constraints on the eigenvalues

Based on the eigenvalues of the cluster scatter matrices, a scatter similarity constraint may be defined. With $\lambda_l(\Sigma_j)$ as the eigenvalues of the cluster scatter matrices Σ_j and

$$M_n = \max_{j=1,...,k} \max_{l=1,...,p} \lambda_l(\Sigma_j) \text{ and } m_n = \min_{j=1,...,k} \min_{l=1,...,p} \lambda_l(\Sigma_j) \tag{4}$$

as the maximum and minimum eigenvalues, the restriction `restr = "eigen"` constrains the ratio M_n/m_n to be smaller than a fixed value `restr.fact`. A theoretical study of the properties of this approach with `equal.weights = FALSE` can be found in García-Escudero *et al.* (2008). This type of constraints on the eigenvalues goes back to those applied by Hathaway (1985) for univariate mixture modeling.

Setting `equal.weights = TRUE`, `restr = "eigen"` and `restr.fact = 1` implies the most constrained case. In this case, the `tclust` function tries to solve the trimmed k-means problem as introduced by Cuesta-Albertos *et al.* (1997). This problem simplifies to the well-known k-means clustering criterion when no trimming is done (i.e. `alpha = 0`). The `tkmeans` function directly implements this most constrained application of the `tclust` function.

3.2 Constraints on the determinants

Another way of restricting cluster scatter matrices is constraining their determinants. Thus, if

$$M_n = \max_{j=1,...,k} |\Sigma_j| \text{ and } m_n = \min_{j=1,...,k} |\Sigma_j|$$

are the maximum and minimum determinants, we attempt to maximize (2) or (3) by constraining the ratio M_n/m_n to be smaller than a fixed value `restr.fact`. This is done in the function `tclust` by using the option `restr = "deter"`.

The untrimmed case `alpha = 0`, `restr = "deter"` and `restr.fact = 1` was already outlined in Maronna and Jacovkis (1974), as the only sensible way to avoid (Mahalanobis distance modified) k-means type algorithms to return clusters of a few almost collinear observations. The possibility of trimming data is also considered in Gallegos (2002) who implicitly assumes $|\Sigma_1| = ... = |\Sigma_k|$ (and so `restr.fact = 1`). The package presented here extends her approach to more general cases (`restr.fact` ≥ 1).

3.3 Equal scatter matrices

Among the methods considered, **t**clust also implements a stronger type of constraint by setting `restr = "sigma"` which forces all cluster scatter matrices to be the same: $\Sigma_1 = ... = \Sigma_k$. This is known as the "determinantal" criterium and it goes back to Friedman and Rubin (1967). The trimmed version of this approach was introduced by Gallegos and Ritter (2005). The argument `restr.fact` is ignored when applying this type of constraint.

3.4 Example

In this example, we compare the clustering results obtained by the `tclust` function and different constraints applied to the so-called `M5data` data set. This data set, that also accompanies the **t**clust package, has been generated following the simulation scheme M5 introduced in García-Escudero *et al.* (2008). Thus it is a bivariate mixture of three simulated gaussian components with very different scatters and a clear overlap between two of these components. A 10% proportion of outliers is also added in the outer region of the bounding rectangle enclosing the 3 gaussian components. See Figure 2 for a graphical representation and García-Escudero *et al.* (2008) for more details on the structure of this `M5data` data set. Executing the following code yields Figure 3.

```
R > data ("M5data")
R > x <- M5data[, 1:2]
R > res.a <- tclust (x, k = 3, alpha = 0.1, restr.fact = 1,  restr = "eigen",
  + equal.weights = TRUE)
R > res.b <- tclust (x, k = 3, alpha = 0.1, restr.fact = 1,  restr = "sigma",
  + equal.weights = TRUE)
R > res.c <- tclust (x, k = 3, alpha = 0.1, restr.fact = 1,  restr = "deter",
  + equal.weights = TRUE)
R > res.d <- tclust (x, k = 3, alpha = 0.1, restr.fact = 50, restr = "eigen",
  + equal.weights = FALSE)
R > plot (res.a, main = "(a) Trimmed k-means")
R > plot (res.b, main = "(b) Gallegos and Ritter (2005)")
```

```
R > plot (res.c, main = "(c) Gallegos (2002)")
R > plot (res.d, main = "(d) Garcia-Escudero et al. (2008)")
```

Although different constraints are imposed, we are searching for $k = 3$ clusters and the trimming proportion is set to $\alpha = 0.1$ in all the cases. Note that only the clustering procedure introduced in García-Escudero *et al.* (2008), shown in Figure 3,(d), with a sufficiently large value of `restr.fact` approximately returns the three original clusters in spite of the very different clus-

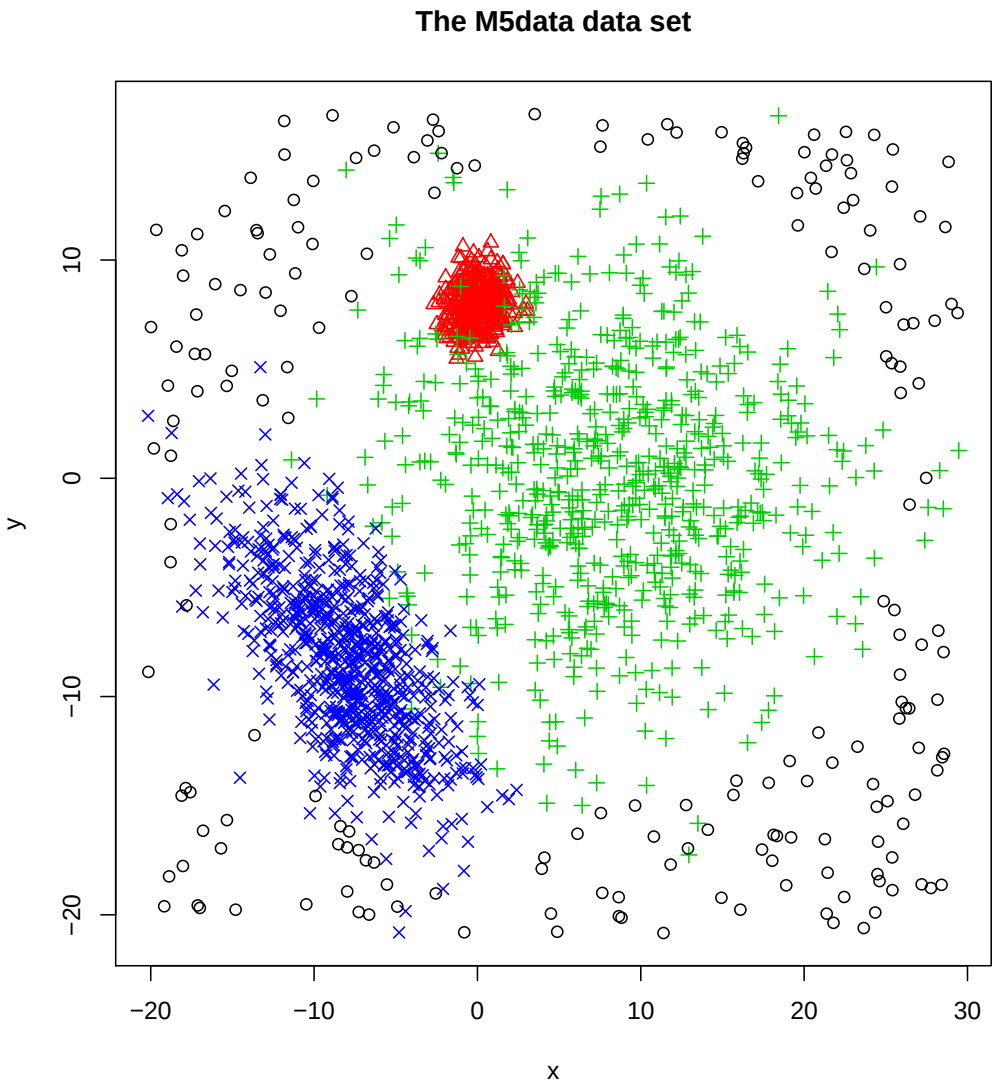

Figure 2: A scatter plot of the `M5data` data set. Different symbols are used for the data points generated by each of the three bivariate normal components and "○" for the added outliers.

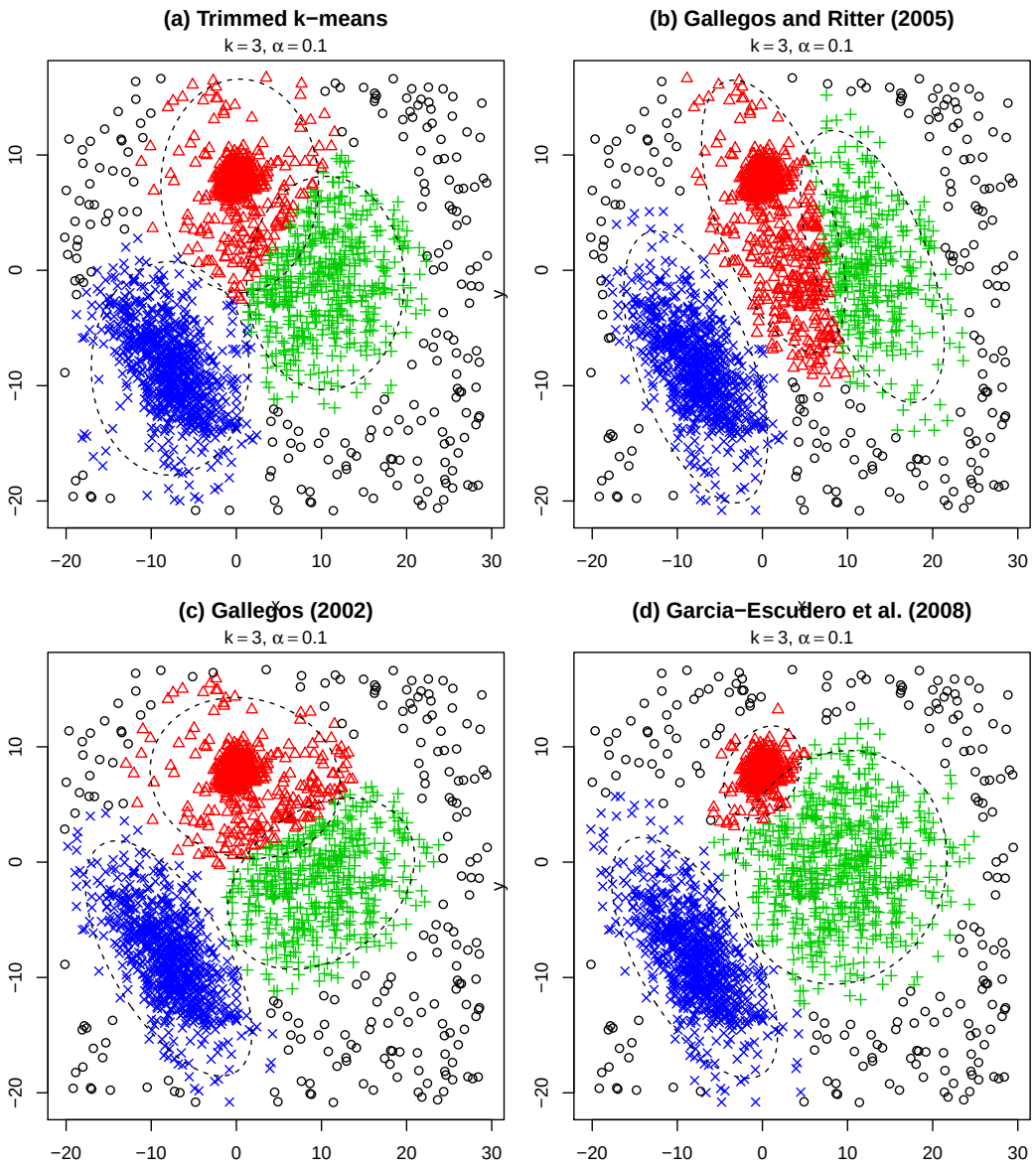

Figure 3: Results of the clustering processes for the `M5data` data set for different constraints on the cluster scatter matrices.

ter scatters and different cluster sizes. Moreover, this clustering procedure adequately handles the severe overlap of two clusters. The value `restr.fact = 50` has been chosen in this case because the eigenvectors of the covariance matrices of the three gaussian components satisfy restriction (4) for this value. Due to their underlying assumptions, the other three clustering methods (trimmed k-means in Figure 3,(a), Gallegos and Ritter (2005) in (b), Gallegos (2002) in (c)) return rather similarly structured clusters. In fact, we found spherical clusters in (a), clusters with the same

scatter matrix in (b) and clusters with the same cluster scatter matrix determinant in (c).

4 Numerical output

The function `tclust` returns an S3 object containing the cluster centers μ_j by columns (`$centers`), scatter matrices Σ_j as an array (`$cov`) and the maximum value found for the trimmed log-likelihood objective function (2) or (3) (`$obj`). The vector `$cluster` provides the cluster assignment of each observation, whereas an artificial cluster "`0`" (without location and scatter information) is introduced which holds all trimmed data points.

Sometimes equations (2) and (3) maximize with some clusters remaining empty. In this case, only information on the non-empty groups is returned. Notice that, if we are searching for k clusters, empty clusters can be found when a clustering solution for a number of clusters strictly smaller than k attains a higher value for (2) or (3) than the best solution found with k clusters. In this case, artificial empty clusters may be defined by considering sufficiently remote centers μ_j and scatter matrices Σ_j satisfying the desired constraints.

Let us consider the following code

```
R > set.seed (10)
R > x <- rbind (rmvnorm (200, c (0, 0), diag (2)),
  +           rmvnorm (200, c (5, 0), diag (2)))
R > plot (tclust (x, k = 3, alpha = 0, restr.fact = 1))
```

Although we are searching for $k = 3$ clusters, we can see in Figure 4 that only 2 clusters are found. Notice that $k = 2$ is surely a more sensible choice for the number of clusters than $k = 3$ for this generated data set. Therefore, the detection of empty clusters, or clusters with few data points, can be helpful providing valuable tools for making sensible choices for k as we will see in Section 6. On the other hand, the detection of empty clusters is very unlikely to happen when the argument `equal.weights = TRUE` is provided in the call to `tclust`.

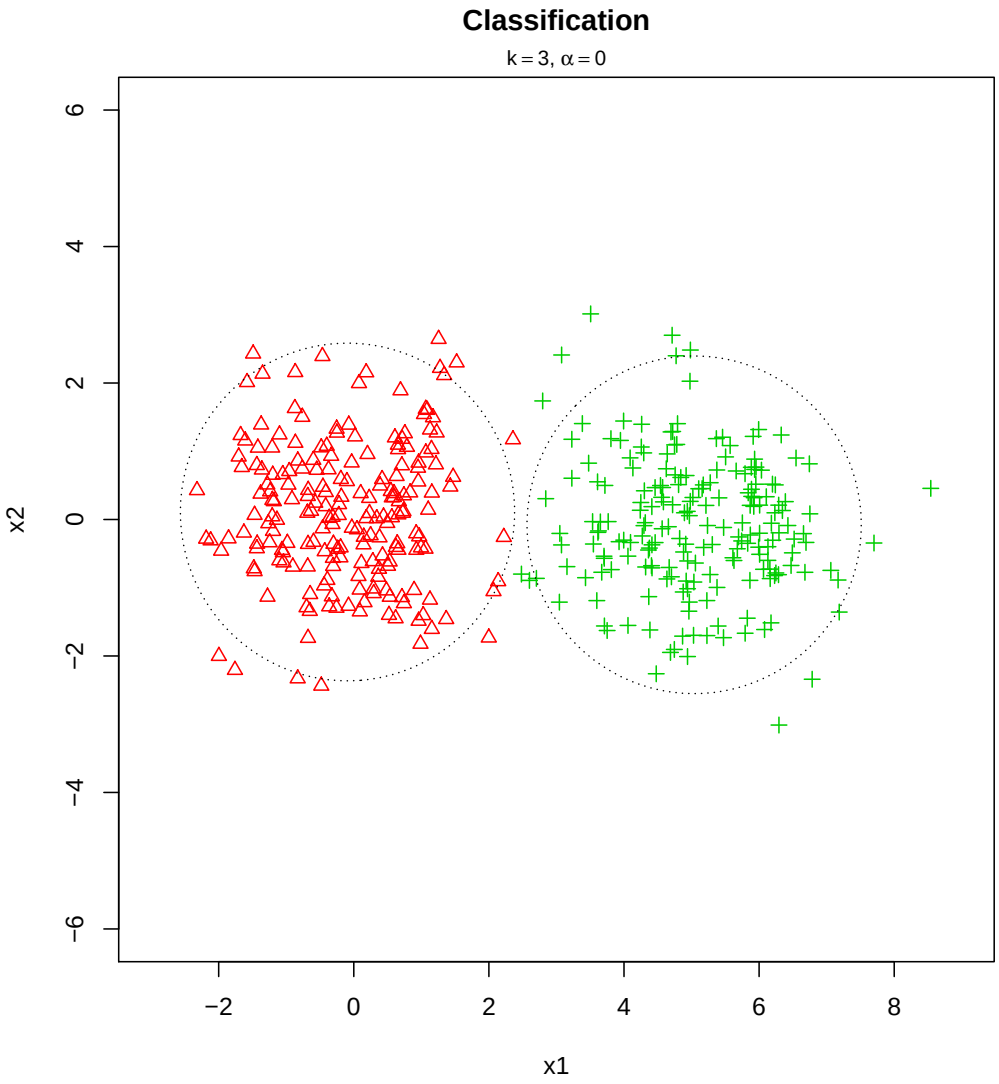

Figure 4: Applying `tclust` with `k = 3` and `alpha = 0` on a simulated data set which originally consists of 2 clusters.

5 Algorithms

The maximization of (2) or (3) considering different cluster scatter matrix constraints is not straightforward because of the combinatorial nature of the associated maximization problems. The algorithm presented in García-Escudero *et al.* (2008) can be adapted to approximately solve all these problems. The methods implemented in **tclust** could be seen as Classification EM algorithms (Celeux and Govaert, 1992), whereas a certain type of "concentration" steps (see the

fast-MCD algorithm in Rousseeuw and Van Driessen, 1998) is applied. In fact, the concentration steps applied by the package **t**clust can be considered as an extension of those applied by the Forgy (1965) algorithm used in k-means clustering. It can be seen that the target function always increases throughout the application of concentration steps, whereas several random start configurations are needed in order to avoid ending trapped in local minima. Therefore, `nstart` random initializations and `iter.max` to the global optimum maximizing (2) or (3) increases with larger values of `nstart` and `iter.max`. The drawback of high values concentration steps are considered. The probability that the algorithm converges close of `nstart` and `iter.max` obviously is the increasing computational effort.

In the concentration step, the centers and scatter matrices are updated by considering the cluster sample means and cluster sample covariance matrices. New cluster assignments are obtained by gathering the "closest" observations to the new centers, whereas the cluster sample covariance matrices are taken into account. If needed, in the updating step, the cluster sample covariance matrices are modified as little as possible but in such a way that they satisfy the desired constraints (see more details in García-Escudero *et al.*, 2008).

Notice that the weights should be taken all equal to $\pi_j = 1/k$ (`equal.weights = TRUE`) when maximizing (2).

6 Selecting the number of groups and the trimming size

Perhaps one of the most complex problems when applying cluster analysis is the choice of the number of clusters, k. In some cases one might have an idea of the number of clusters in advance, but usually k is completely unknown. Moreover, in the approach proposed here, the trimming proportion α has also to be chosen without knowing the true contamination level.

As we will see through the following example, the choices for k and α are related problems that should be addressed simultaneously. It is important to see that a particular trimming level implies a specific number of clusters and vice versa. This dependency can be explained as entire clusters tend to be trimmed completely when increasing α. On the other hand, when choosing α too low, groups of outliers might form new spurious clusters and thus it appears that the number of clusters found in the data set is quite high. Moreover, the simultaneous choice of k and α depends on the allowed differences between cluster scatter sizes, which is controlled by the argument `restr.fact`.

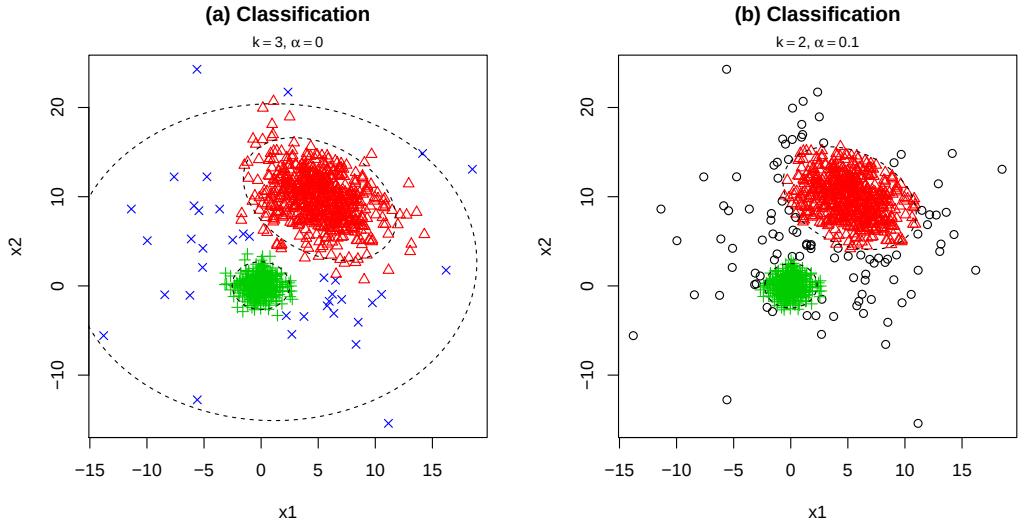

Figure 5: Clustering results for the simulated data set `mixt` with $k = 3$, $\alpha = 0$ and `restr.fact` $= 50$ (a) and $k = 2$, $\alpha = 0.1$ and `restr.fact` $= 12$ (b).

To demonstrate the relation between α, k and `restr.fact`, let us consider the data set in Figure 5 which could either be interpreted as a mixture of three components (a) or a mixture of two components (b) with a 10% outlier proportion. Both clustering solutions shown in Figure 5 are perfectly sensible and the final choice of α and k only depends on the value given to `restr.fact`. The code used to obtain Figure 5 is the following:

```
R > mixt <- rbind (
  + rmvnorm (360, c (0.0,  0), matrix (c ( 1,  0,  0,  1), ncol = 2)),
  + rmvnorm (540, c (5.0, 10), matrix (c ( 6, -2, -2,  6), ncol = 2)),
  + rmvnorm (100, c (2.5,  5), matrix (c (50,  0,  0, 50), ncol = 2)))
R > plot (tclust (mixt, k = 3, alpha = 0.0, restr.fact = 50))
R > plot (tclust (mixt, k = 2, alpha = 0.1, restr.fact = 12))
```

In general, we assume that the value of `restr.fact` has been fixed in advance by the researcher who applies the robust clustering method. Thus, the choice of `restr.fact` should depend on prior knowledge of the type of clusters the researcher is looking for. Large values of `restr.fact` lead to rather unrestricted solutions, while smaller values of `restr.fact` yield more similar structured clusters.

Even when specifying a single type of constraint and assuming $\alpha = 0$, choosing the appropriate number of clusters is not an easy task. The careful monitoring of the maximum value attained by log-likelihoods like those in (2) and (3) while changing k has traditionally been applied as a method for choosing the number of clusters when $\alpha = 0$. Moreover Bryant (1991) stated that the use of "weighted" log-likelihoods (3) is preferred to the use of log-likelihoods assuming equal weights (2). Notice that increasing k always causes the maximized log-likelihood (2) to increase too, and this could lead to "overestimate" the appropriate number of clusters (see García-Escudero *et al.*, 2010a).

In this trimming framework, let us consider $\mathcal{L}^{\Pi}_{\texttt{rest.fact}}(\alpha, k)$ as the maximum value reached by (3) for each combination of a given set of values for k and α. García-Escudero *et al.* (2010a) propose to monitor the "classification trimmed likelihoods" functionals

$$(\alpha, k) \mapsto \mathcal{L}^{\Pi}_{\texttt{rest.fact}}(\alpha, k)$$

while altering α and k, which yields an exploratory graphical tool for making sensible choices for parameters α and k. In fact, it is proposed to choose the number of clusters as the smallest value of k such that

$$\mathcal{L}^{\Pi}_{\texttt{rest.fact}}(\alpha, k+1) - \mathcal{L}^{\Pi}_{\texttt{rest.fact}}(\alpha, k) \tag{5}$$

is always (close to) 0 except for small values of α. Once the number of clusters is fixed, a good choice for the trimming level is the first α_0 such that (5) is (close to) 0 for every $\alpha \geq \alpha_0$. Although we are convinced that monitoring the classification trimmed likelihoods functionals is very informative, no theoretical statistical procedures are available yet for determining when (5) can be formally considered as "close to 0".

The function `ctlcurves` in package **tclust** approximates the classification trimmed likelihoods by successively applying the `tclust` function for a sequence of values of k and α. The default value `restr.fact` is set to 50 because we are allowing the method high flexibility for determining extra clusters but, if desired, smaller values of `restr.fact` can be passed to `tclust` via `ctlcurves` too. For instance, the following code applied to the previously simulated `mixt` data set

```
R > plot (ctlcurves (mixt, k = 1:4, alpha = seq (0, 0.2, by = 0.05)))
```

results in Figure 6 and shows that increasing k from 2 to 3 is clearly needed when $\alpha = 0$, as the objective functions value differs noticeably between $k = 2$ and $k = 3$. On the other hand,

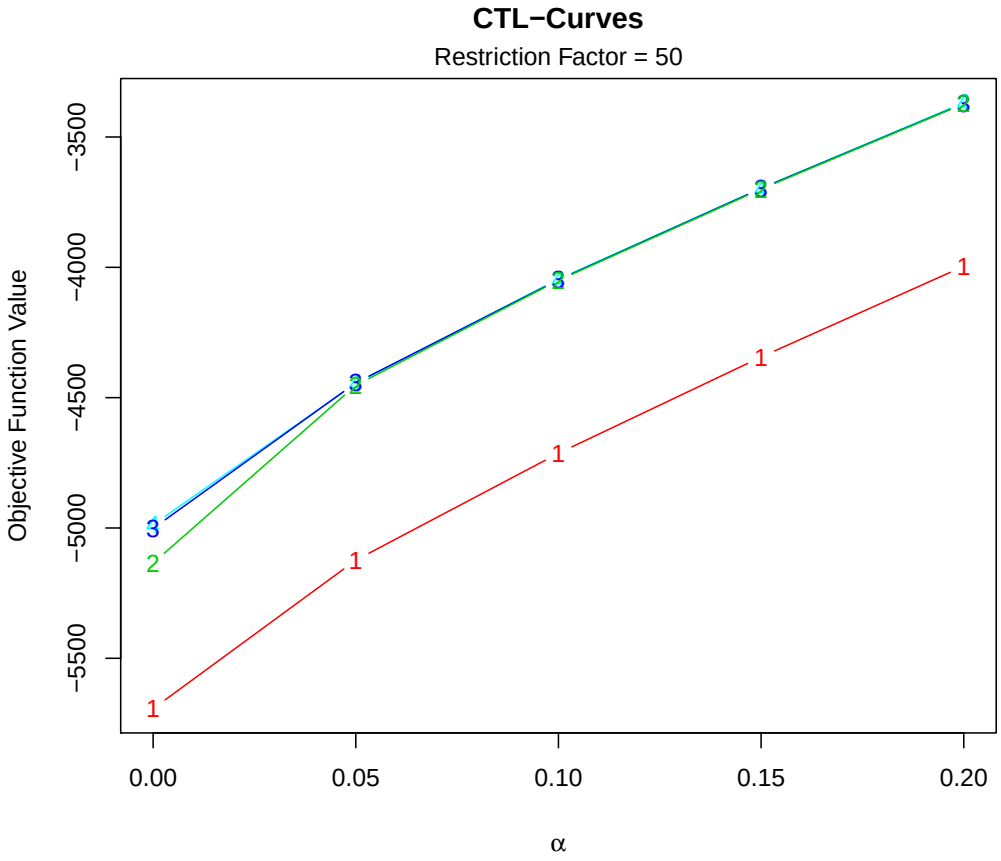

Figure 6: Classification trimmed likelihoods with $k = 1, ..., 4$, $\alpha = 0$, 0.05, ..., 0.2 and `restr.fact` = 50 for the `mixt` data set in Figure 5.

increasing k from 2 to 3 is not needed anymore as the third (more scattered) "cluster" vanishes when trimming 5% of the most outlying observations. Thus, there is no difference of the objective functions value with $\alpha \geq \alpha_0 = 0.05$ and $k \geq 2$. Note that, although the true contamination level was actually 10%, a 5% trimming level is enough in this case because the contaminated observations partially overlap with the two main clusters. Increasing k from 3 to 4 is not needed in any case.

The curves presented in García-Escudero *et al.* (2003) can be considered as precedents of those we obtain by using the `ctlcurves` function. Trimmed likelihoods have also been taken into account in Neykov *et al.* (2007) for choosing k and α by using a BIC criterium.

Note that if arguments `nstart` and `iter.max` are provided in the call to `ctlcurves`, they are

internally passed to function `tclust`.

7 Graphical displays

As seen in previous examples, the package **t**clust provides functions for visualizing the computed cluster assignments. One dimensional, two dimensional and higher dimensional cases are visualized differently:

$p = 1$**:** The one-dimensional data set with the corresponding cluster assignments is displayed along the x-axis. Setting the argument `jitter = TRUE` jitters the data along the y-axis in order to increase the visibility of the actual data structure. Additionally, a (robust) scatter estimation of each cluster is also displayed.

$p = 2$**:** Tolerance ellipsoids are plotted additionally in order to visualize the estimated cluster scatter matrices.

$p > 2$**:** The first two Fisher's canonical coordinates are displayed in this case, which are computed based on the estimated cluster scatter matrices (notice that trimmed observations are not taken into account when computing these coordinates). The implementation of these canonical coordinates is derived from the function `discrcoord` as implemented in the package **f**pc (Hennig, 2010).

A simple example demonstrates how the `plot` function works in different dimensions. The code

```
R > geyser1 <- geyser2[, 1, drop = FALSE]
R > geyser3 <- cbind (geyser2, rnorm (nrow (geyser2)))
R > plot (tkmeans (geyser1, k = 2, alpha = 0.03), jitter = TRUE)
R > plot (tkmeans (geyser3, k = 3, alpha = 0.03))
```

yields Figure 7. We have selected some variables of the `geyser2` data to obtain a one-dimensional and a three-dimensional data set and plotted the results of the trimmed k-means robust clustering method.

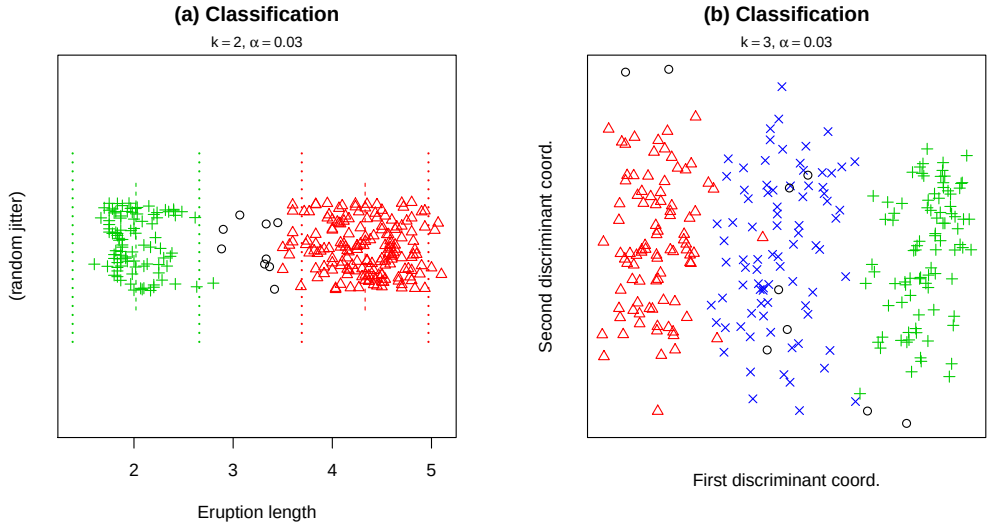

Figure 7: Trimmed k-means clustering results for a one-dimensional (a) and a three-dimensional (b) data set based on the `geyser2` data. In the one-dimensional setting (a) $k = 2$ clusters are considered, whereas in the three-dimensional setting (b) the number of clusters has been increased to $k = 3$. A trimming proportion $\alpha = 0.03$ is fixed in both cases.

Given a `tclust` object, some additional exploratory graphical tools can be applied in order to evaluate the quality of the cluster assignments and the trimming decisions. This is done by applying the function `DiscrFact`.

Let $\widehat{R} = \{\widehat{R_0}, \widehat{R_1}, ..., \widehat{R_k}\}$, $\widehat{\theta} = (\widehat{\theta_1}, ..., \widehat{\theta_k})$ and $\widehat{\pi} = (\widehat{\pi_1}, ..., \widehat{\pi_k})$ be the values obtained by maximizing (3). $D_j(x_i; \widehat{\theta}, \widehat{\pi}) = \widehat{\pi_j}\phi(x_i, \widehat{\theta_j})$ is a measure of the degree of affiliation of observation x_i with cluster j. These values can be ordered as $D_{(1)}(x_i; \widehat{\theta}, \widehat{\pi}) \leq ... \leq D_{(k)}(x_i; \widehat{\theta}, \widehat{\pi})$. Thus the quality of the assignment decision of a non trimmed observation x_i to cluster j can be evaluated by comparing its degree of affiliation with cluster j to the best second possible assignment through $\mathrm{DF}(i) = \log\left(D_{(k)}(x_i; \widehat{\theta}, \widehat{\pi})/D_{(k-1)}(x_i; \widehat{\theta}, \widehat{\pi})\right)$.

It is easy to see that the $\lceil n\alpha \rceil$ observations with smallest values for $D_{(k)}(x_i; \widehat{\theta}, \widehat{\pi})$ are the trimmed ones. Considering for a trimmed observation x_i, the quality of the trimming decision can be evaluated by comparing $D_{(k)}(x_i; \widehat{\theta}, \widehat{\pi})$ and $D_{(k)}(x_l; \widehat{\theta}, \widehat{\pi})$ with x_l being the non-trimmed observation with smallest value of $D_{(k)}(x_l; \widehat{\theta}, \widehat{\pi})$ by using $\mathrm{DF}(i) = \log\left(D_{(k)}(x_l; \widehat{\theta}, \widehat{\pi})/\ D_{(k)}(x_i; \widehat{\theta}, \widehat{\pi})\right)$. Following this approach, discriminant factors $\mathrm{DF}(i)$ are obtained for every observation in the data set, whether trimmed or not.

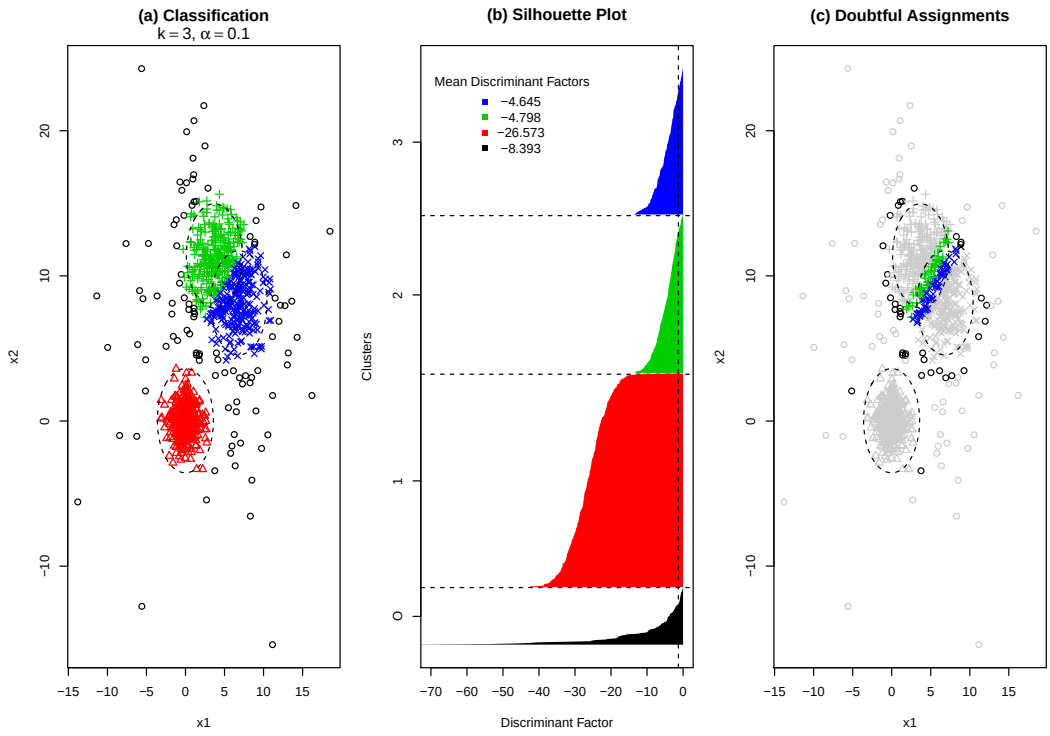

Figure 8: Graphical displays based on the DF(i) values for a `tclust` cluster solution obtained with $k = 3$, $\alpha = 0.1$, `restr.fact` $= 1$ and `equal.weights = TRUE` for the `mixt` data set.

Observations with large DF(i) values indicate doubtful assignments or trimming decisions. The use of this type of discriminant factors was already suggested in Van Aelst *et al.* (2006) in a clustering problem without trimming. "Silhouette" plots (Rousseeuw, 1987) can be used for summarizing the obtained ordered discriminant factors. Clusters in the silhouette plot with many large DF(i) values indicate the existence of not very "well-determined" clusters. The most "doubtful" assignments with DF(i) larger than a log(`threshold`) value are highlighted by the function `DiscrFact`.

```
R > clus.w <- tclust (mixt, k = 3, alpha = 0.1, restr.fact = 1,
  + equal.weights = TRUE)
R > discr.clus.w <- DiscrFact (clus.w, threshold = 0.1)
R > plot (discr.clus.w)
```

Figure 8 shows a clustering solution for the `mixt` data set shown in Figure 5. Although Figure 6 suggests to choose $k = 2$, k has been increased to 3 in order to show how such a change

leads to doubtful cluster assignment decisions which can be visualized by `DiscrFact`. Figure 8,(a) simply illustrates the cluster assignments and trimming decisions. The mentioned silhouette plot is presented in (b), whereas the doubtful decisions are marked in (c). All observations with $\mathrm{DF}(i) \geq \log(0.1)$ are highlighted as they are plotted darker/in color. Most of the doubtful decisions are located in the overlapping area of the two artificially found clusters (highlighted symbols "$\times$" and "+"). Some doubtfully trimmed observations (highlighted symbol "$\circ$") are located in the boundaries of these two clusters.

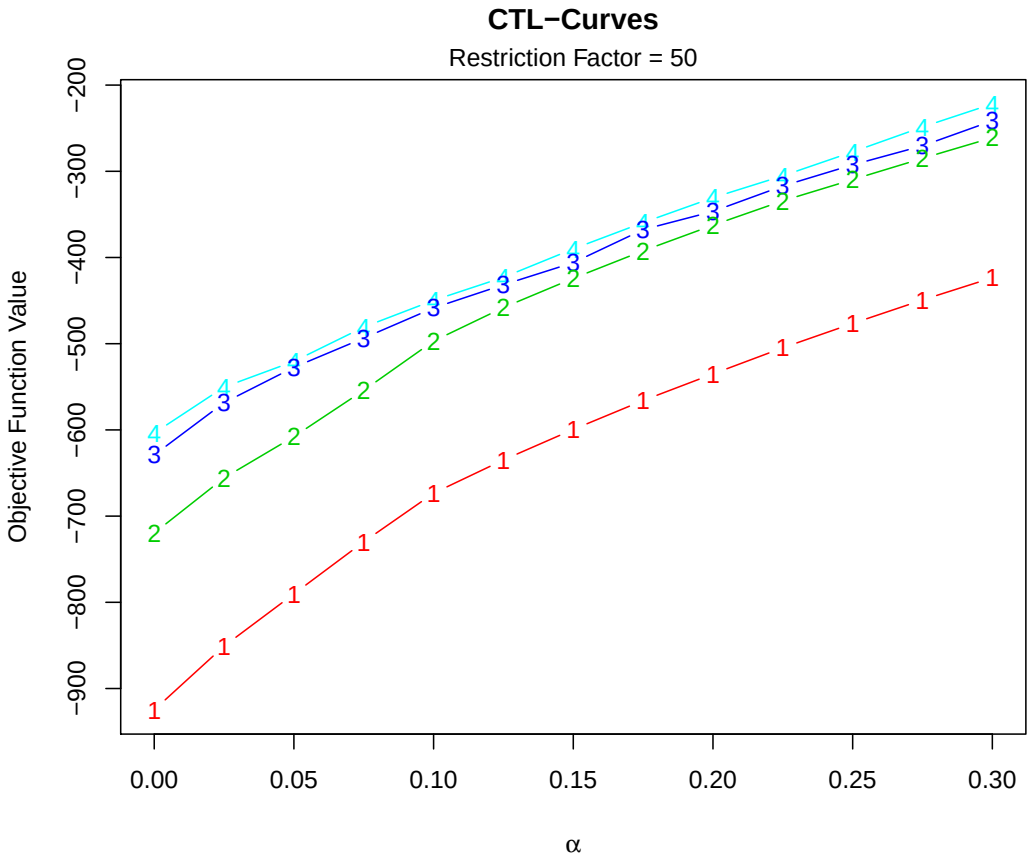

Figure 9: Classification trimmed likelihoods for $k = 1, ..., 4$ and $\alpha = 0, .025, ..., .3$ when `restr.fact` $= 50$ for the "Swiss Bank notes" data set.

8 Swiss Bank notes data

The well-known "Swiss Bank notes" data set includes 6 numerical measurements (six-dimensional data set) made on 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes (Flury and Riedwyl, 1988). The following code can be used to obtain the classification trimmed likelihoods shown in Figure 9.

```
R > data ("swissbank")
R > plot (ctlcurves (swissbank, k = 1:4, alpha = seq (0, 0.3, by = 0.025)))
```

This figure indicates the clear existence of $k = 2$ main clusters ("genuine" and "forged" bills). Moreover, considering the clear difference between $\mathcal{L}^{\Pi}_{50}(0, 3)$ and $\mathcal{L}^{\Pi}_{50}(0, 2)$, we can see that a further cluster, i.e. $k = 3$, is needed when no trimming is allowed. This extra cluster can be justified by the inhomogeneity of the group of forgeries (perhaps due to the presence of different sources of forged bills).

Considering Figure 9, the choice $k = 2$ and a value of α close to 0.1 also seem sensible. Notice that $\mathcal{L}^{\Pi}_{50}(\alpha, 3)$ is clearly larger than $\mathcal{L}^{\Pi}_{50}(\alpha, 2)$ for $\alpha < 0.1$ while these differences are not so big when $\alpha \geq 0.1$. We can even see smaller differences in the classification trimmed likelihood curves when increasing k from 3 to 4. However, these differences are less significant than those previously commented. More spurious clusters can be surely found but they have less entity and importance.

Figure 10 shows the clustering results with $k = 2$, $\alpha = 0.1$ and `restr.fact = 50` obtained by executing the code:

```
R > clus <- tclust (swissbank, k = 2, alpha = 0.1, restr.fact = 50)
R > plot (DiscrFact (clus, threshold = .000125))
```

The value `restr.fact = 50` has been considered because this was also the (default) value used for `ctlcurves`. Notice also that variables in this data set are not standardized and, thus, we do not expect to find very "spherically" shaped clusters and a large value of `restr.fact` is needed.

We use the function `DiscrFact` to summarize the obtained clustering results. The two first Fisher's canonical coordinates derived from the final cluster assignments are plotted. The threshold value 0.000125 is chosen in order to highlight the 7 most doubtful decisions.

Finally, Figure 11 shows a scatterplot of the fourth ("Distance of the inner frame to lower border") against the sixth variable ("Length of the diagonal") with the corresponding cluster as-

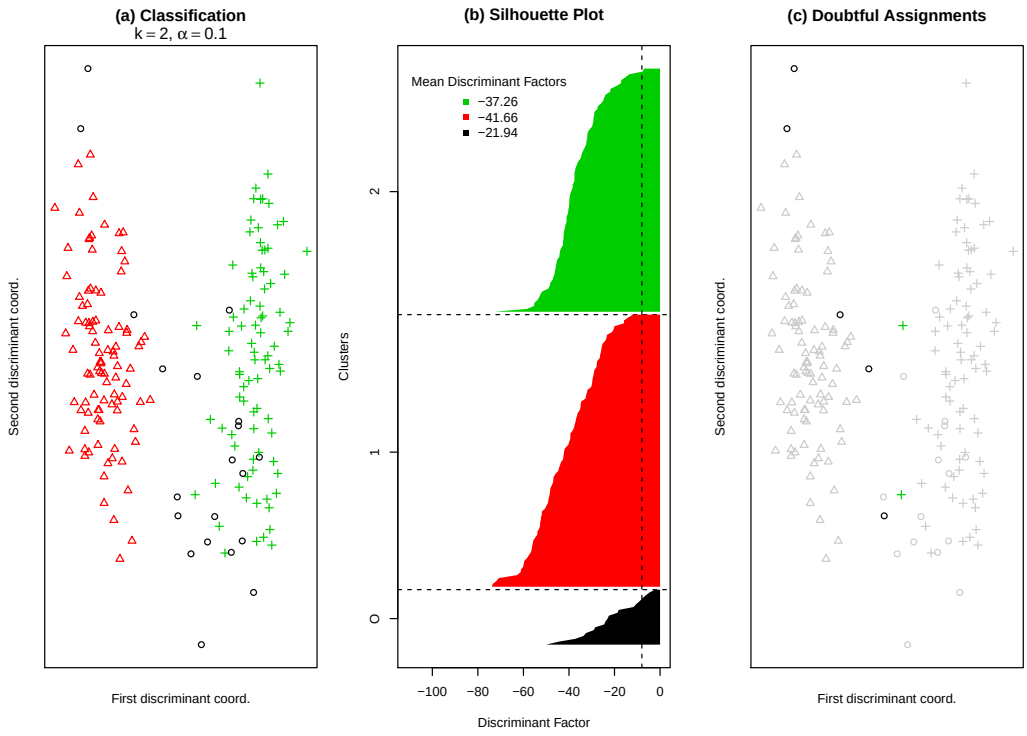

Figure 10: Clustering results with $k = 2$, $\alpha = 0.1$ and `restr.fact` $= 50$ summarized by the use of `DiscrFact` function for the "Swiss Bank notes" data set. The threshold value is chosen in order to highlight the 7 most doubtful cluster assignments.

signments. We use the symbols "`G`" for the genuine bills and "`F`" for the forged ones. The 7 most doubtful decisions (i.e., the observations with largest DF(i) values that were highlighted in Figure 10,(c)) are surrounded by circles in this figure.

We can see that "Cluster 1" essentially includes most of the "forged" bills while "Cluster 2" includes most of the "genuine" ones. Among the trimmed observations, we can find a subset of 15 forged bills following a clearly different forgery pattern that has been previously commented by other authors (see, e.g. Flury and Riedwyl, 1988; Cook, 1999). These most doubtful assignments include 5 "genuine" bills that have perhaps been wrongly trimmed.

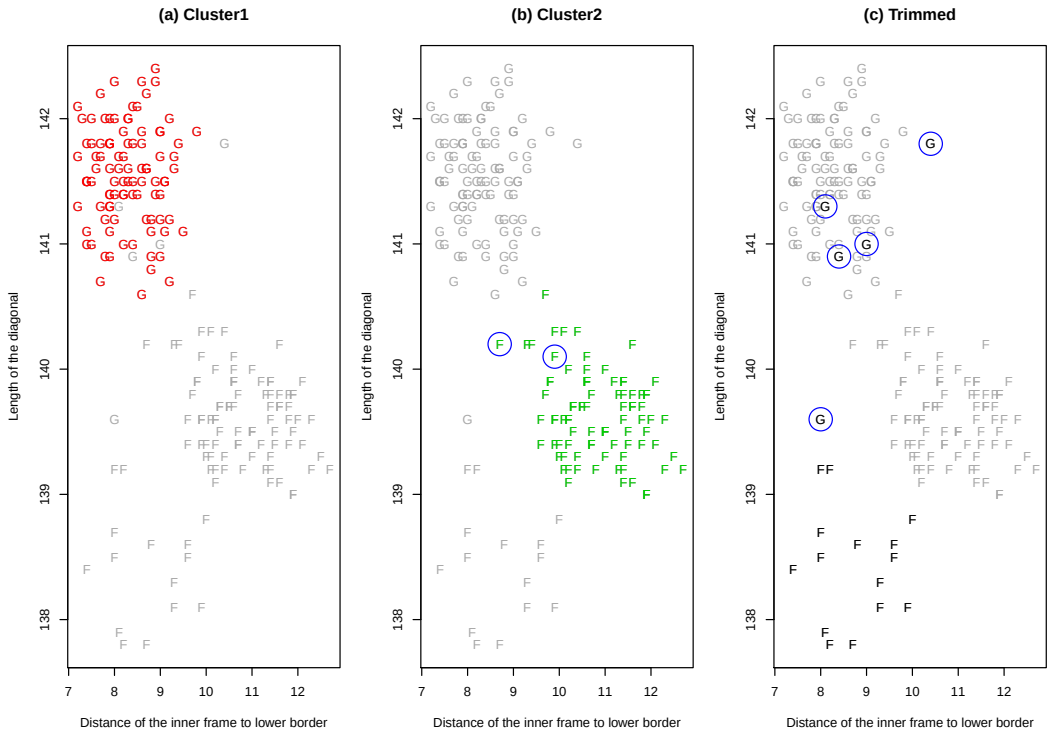

Figure 11: Clustering results with $k = 2$, $\alpha = .1$ and `restr.fact` $= 50$ for the "Swiss Bank notes" data set. Only the fourth and sixth variables are plotted. The 7 most doubtful decisions are rounded by a circle symbol.

9 Conclusion

We have presented a package called **tclust** for robust (non-hierarchical) clustering. As the package is implemented in a flexible manner, only the restrictions on the cluster scatters have to be changed in order to carry out different robust clustering algorithms. Robustness is achieved by trimming a specific amount of observations which are identified as the "most outlying" ones.

This **R**-package implements robust clustering approaches which have already been described in the literature, whereas some of these approaches are extended to gain flexibility. The package also provides some graphical tools which on the one hand help to chose appropriate parameters (`ctlcurves`) and on the other hand help to estimate the adequacy of a particular clustering solution (`DiscrFact`).

The future work on this package focuses on implementing further types of scatter restrictions, making the algorithm even more flexible and on providing more numerical tools for automatically

choosing the number of clusters and the trimming proportion.

Acknowledgements:

This research is partially supported by the Spanish Ministerio de Ciencia e Innovación, grant MTM2008-06067-C02-01, and 02 and by Consejería de Educación y Cultura de la Junta de Castilla y León, GR150.

References

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**(3), 803–821.

Bryant, P. (1991). Large-sample results for optimization-based clustering methods. *Journal of Classification*, **8**(1), 31–44.

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

Cook, R. D. (1999). Graphical detection of regression outliers and mixtures. In *Proceedings of the International Statistical Institute 1999*, Finland. ISI.

Cuesta-Albertos, J., Gordaliza, A., and Matrán, C. (1997). Trimmed k-means: an attempt to robustify quantizers. *Annals of Statistics*, **25**(2), 553–576.

Flury, B. and Riedwyl, H. (1988). *Multivariate Statistics. A Practical Approach*. Chapman and Hall, London.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, **21**, 768Ű769.

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

Friedman, H. and Rubin, J. (1967). On some invariant criterion for grouping data. *Journal of the American Statistical Association*, **63**(320), 1159–1178.

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

Gallegos, M. T. and Ritter, G. (2005). A robust method for cluster analysis. *Annals of Statistics*, **33**(1), 347–380.

Gallegos, M. T. and Ritter, G. (2009). Trimming algorithms for clustering contaminated grouped data and their robustness. *Advances in Data Analysis and Classification*, **3**(2), 135–167.

Gallegos, M. T. and Ritter, G. (2010). Using combinatorial optimization in model-based trimmed clustering with cardinality constraints. *Computational Statistics & Data Analysis*, **54**(3), 637–654.

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

García-Escudero, L. A., Gordaliza, A., and Matrán, C. (2003). Trimming tools in exploratory data analysis. *Journal of Computational and Graphical Statistics*, **12**(2), 434–449.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008). A general trimming approach to robust cluster analysis. *Annals of Statistics*, **36**(3), 1324–1345.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2010a). 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`.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2010b). A review of robust clustering methods. *Advances in Data Analysis and Classification*, **4**(2-3), 89–109.

Hardin, J. and Rocke, D. M. (2004). Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, **44**(4), 625 – 638.

Hathaway, R. J. (1985). A constrained formulation of maximum likelihood estimation for normal mixture distributions. *Annals of Statistics*, **13**(2), 795–800.

Hennig, C. (2010). ***fpc**: Flexible procedures for clustering. Version 2.0-2*.

Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software*, **11**(8), 1–18.

Maronna, R. and Jacovkis, P. M. (1974). Multivariate clustering procedures with variable metrics. *Biometrics*, **30**(3), 499–505.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. John Wiley & Sons, New York.

Neykov, N., Filzmoser, P., Dimova, R., and Neytchev, P. (2007). Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics & Data Analysis*, **52**(1), 299–308.

R Development Core Team (2010). ***R**: A Language and Environment for Statistical Computing*. **R** Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rocke, D. M. and Woodruff, D. L. (2002). Computational connections between robust multivariate analysis and clustering. In W. Härdle and R. B., editors, *COMPSTAT 2002 Proceedings in Computational Statistics*, pages 255–260.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. In I. V. W. Grossmann, G. Pflug and W. Wertz, editors, *Mathematical Statistics and Applications*, volume B, pages 283–297, Dordrecht. Reidel.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, **20**(1), 53–65.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York.

Rousseeuw, P. J. and Van Driessen, K. (1998). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, **41**, 212–223.

Van Aelst, S., Wang, X., Zamar, R. H., and Zhu, R. (2006). Linear grouping using orthogonal regression. *Computational Statistics & Data Analysis*, **50**(5), 1287–1312.

Woodruff, D. and Reiners, T. (2004). Experiments with, and on, algorithms for maximum likelihood clustering. *Computational Statistics & Data Analysis*, **47**(2), 237–253.