ROBUST IMPROPER MAXIMUM LIKELIHOOD:
TUNING, COMPUTATION, AND A COMPARISON WITH
OTHER METHODS FOR ROBUST GAUSSIAN CLUSTERING

Pietro Coretto∗
Università di Salerno, Italy
e-mail: pcoretto@unisa.it

Christian Hennig†
University College London, UK
e-mail: c.hennig@ucl.ac.uk

Abstract. The two main topics of this paper are the introduction of the “optimally
tuned improper maximum likelihood estimator” (OTRIMLE) for robust clustering
based on the multivariate Gaussian model for clusters, and a comprehensive simula-
tion study comparing the OTRIMLE to Maximum Likelihood in Gaussian mixtures
with and without noise component, mixtures of t-distributions, and the TCLUST ap-
proach for trimmed clustering. The OTRIMLE uses an improper constant density
for modelling outliers and noise. This can be chosen optimally so that the non-noise
part of the data looks as close to a Gaussian mixture as possible. Some deviation
from Gaussianity can be traded in for lowering the estimated noise proportion. Co-
variance matrix constraints and computation of the OTRIMLE are also treated. The
ideas used for computing the OTRIMLE were applied with benefit to the competing
methods, too, in order to make the approaches comparable. In the simulation study,
all methods are confronted with setups in which their model assumptions were not
exactly fulfilled (even apart from outliers), and in order to evaluate the experiments
in a standardized way by misclassification rates, a new model-based definition of “true
clusters” is introduced that deviates from the usual identification of mixture compo-
nents with clusters. In the study, every method turns out to be superior for one or
more setups, but the OTRIMLE achieves the most satisfactory overall performance.

MSC2010 code. 62H30, 62F35, 62P25.

Keywords. Cluster analysis, EM-algorithm, improper density, maximum likelihood,
mixture models, model-based clustering, robustness.

1 Introduction

In this paper we introduce and investigate the “optimally tuned robust improper maximum
likelihood estimator” (OTRIMLE), a method for robust clustering with clusters that can
be approximated by multivariate Gaussian distributions. The one-dimensional version of
OTRIMLE was introduced in Coretto and Hennig (2010). We also present a simulation
study comparing OTRIMLE and other currently popular approaches for (mostly robust)
model-based clustering, which is, to our knowledge, the most comprehensive study in the

∗The author gratefully acknowledges support from the Università di Salerno grant program “Sistema
di calcolo ad alte prestazioni per l’analisi economica, finanziaria e statistica (HPC - High Performance
Computing) - prot. ASSA098434, 2009”

†The author gratefully acknowledges support from the EPSRC grant EP/K033972/1.
field and involves a careful discussion of the issue of comparing methods based on different model assumptions.

The basic idea of OTRIMLE is to fit an improper density to the data that is made up by a Gaussian mixture density and a “pseudo mixture component” defined by a small constant density, which is meant to capture outliers in low density areas of the data. This is inspired by the addition of a uniform “noise component” to a Gaussian mixture suggested by Banfield and Raftery (1993). Hennig (2004) showed that using an improper density improves the breakdown robustness of this approach. The OTRIMLE has been found to work well for one-dimensional data in the comparative simulation study in Coretto and Hennig (2010).

As in many other statistical problems, violations of the model assumptions and particularly outliers may cause problems in cluster analysis. Our general attitude to the use of statistical models in cluster analysis is that the models should not be understood as reflecting some underlying but in practice unobservable “truth”, but rather as thought constructs implying a certain behaviour of methods derived from them (e.g., maximizing the likelihood), which may or may not be appropriate in a given application (more details on the general philosophy of clustering can be found in Hennig and Liao (2013)). Using a model such as a mixture of multivariate Gaussian distributions, interpreting every mixture component as a “cluster”, implies that we look for clusters that are approximately “Gaussian-shaped”, but we do not want to rely on whether the data really were generated i.i.d. by an at least approximate Gaussian mixture. Therefore we are interested in the performance of such methods in situations where one may legitimately look for Gaussian-shaped clusters, even if some data do not belong to such clusters (such data points will be called “noise” in the following), and even if the clusters are not precisely Gaussian. This reflects the fact that in practice for example mixtures of $t$-distributions are used for clustering the same data sets to which Gaussian mixtures (possibly involving special treatment of outliers) are fitted as well, interpreting the resulting clusters in the same way.

For illustration of the outlier problem in model-based clustering, we use a data set in which the 170 districts of the German city of Dortmund are characterized by a number of variables. The original data set was used for a clustering competition at the 2004 conference of the German classification society GfKl and is described in Sommerer and Weihs (2005). The number of original variables is large, concerning a heterogeneous multitude of characteristics of the districts. We used five sociological key variables and transformed them in such a way that fitting Gaussian distributions within clusters makes sense. The resulting variables are

- **unemployment**: logarithm of unemployment rate,
- **birth.death**: birth/death balance divided by number of inhabitants,
- **moves.in.out**: migration balance divided by number of inhabitants,
- **soc.ins.emp**: logarithm of rate of employees paying social insurance,
- **inhabitants**: square root of number of inhabitants.

All variables were centered and standardised by the median absolute deviation. Figure 1 shows a scatterplot of birth.death and moves.in.out. In order to deal with overplotting, an existing extreme outlier with values $\approx (-200, 50)$ is not shown. Figure 2 shows a scatterplot of unemployment and soc.ins.emp. Figure 2 shows some more moderate outliers. The left side of Figure 1 shows a clustering that was produced by fitting a plain
Gaussian mixture with $G = 4$ to all five variables by R’s MCLUST package (Fraley and Raftery (2006)). Cluster 4 is a one-point cluster consisting only of the extreme outlier. Clusters 1 and 3 basically fit two different varieties of moderate outliers on these two variables, whereas all the more than 120 districts that are not extreme regarding these two variables are put together in a single cluster. Clearly, it would be more desirable to have a clustering that is not dominated so much by a few odd districts, given that there is some meaningful structure among the other districts. Such a clustering is produced by the OTRIMLE method, shown on the right side of Figure 1 and on the left side of Figure 2. The clustering is nicely interpretable with cluster no. 3 collecting a group of districts with higher migration balance and very scattered birth/death rate, cluster no. 1 being a high variation cluster characterized by high unemployment or high number of employees paying social insurance, cluster no. 2 being a homogeneous group with medium number of employees paying social insurance and rather high but not very high unemployment, and cluster no. 4 collecting most districts with low values on both of these variables.

A number of model-based clustering methods that can deal with outliers have been pro-
posed in recent years. An overview of these methods is given in Section 2. The OTRIMLE is introduced and discussed in Section 3, starting from the “RIMLE”, in which the level of the improper constant density is a tuning constant. We then introduce a method for optimal tuning so that the points classified as “non-noise” are as close as possible to a proper Gaussian mixture, and discuss the computation of the resulting method. Section 5 presents a comprehensive comparative simulation study involving the OTRIMLE and methods introduced in Section 2. One of the features of this study is that we do not only look at precisely Gaussian clusters, and that all the methods are confronted with some simulation setups for which their assumptions are not precisely fulfilled (even apart from the noise/outliers). This requires a unified approach for defining elliptically shaped clusters with noise/outliers when the reference cluster prototype is given by the Gaussian model, which is presented in Section 4. Furthermore, computational ideas such as eigenvalue constraints for the covariance matrices, initialization of the EM-algorithm and tuning are applied across several methods, see Section 5.2. Some further issues are discussed in Section 6. Additional details about the example dataset (including full scatterplots), the simulation study, and computation of methods are provided in an online supplement (Coretto and Hennig (2014)). Coretto and Hennig (2013) is a twin of the present paper with a focus on theoretical properties (including the breakdown point) of the RIMLE; we divided them in order to produce papers of more digestible length. Remarks regarding the choice of the number of mixture components are also given there, whereas in the current paper we focus on situations where this number is fixed.

2 Methods from the literature

This section gives an account of existing methods from the literature that aim at finding clusters with approximately Gaussian shapes. In the following, assume an observed sample $x_n = \{x_1, x_2, \ldots, x_n\}$, where $x_i$ is the realization of a random variable $X_i \in \mathbb{R}^p$ with $p \geq 1$; $X_1, \ldots, X_n$ i.i.d. The goal is to cluster the sample points into $G$ distinct groups.

**Maximum Likelihood (ML) for Gaussian mixtures (gmix).** Let $\phi(x; \mu, \Sigma)$ be the density of a multivariate Gaussian distribution with mean vector $\mu \in \mathbb{R}^p$ and $p \times p$ covariance matrix $\Sigma$. Assume that the observed sample is iid drawn from the finite Gaussian mixture distribution having density

$$m(x; \theta) = \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \Sigma_j),$$

(2.1)

where $\pi_j \in [0, 1]$ for all $j = 1, 2, \ldots, G$ and $\sum_{j=1}^{G} \pi_j = 1$, $\theta$ is the parameter vector containing the triplets $\pi_j, \mu_j, \Sigma_j$ for all $j = 1, 2, \ldots, G$. Clustering coincides with assigning points to the mixture components based on ML parameter estimates. $\pi_j$ can be interpreted as the expected proportion of points originated from the $j$th component. Let $\theta_n^{ml}$ be the ML estimator for $\theta$. This is usually computed by the EM algorithm (Dempster et al., 1977). The ML estimator under (2.1) exists only under appropriate constraints on the covariances matrices. These constraints will be discussed in detail in Section 3. Let $\tau_{ij}^{ml}$ be the estimated posterior probability that the observed point $x_i$ has been drawn from the
The point $x_i$ can then be assigned to the $k$th cluster if $k = \arg \max_{j=1,2,...,G} \tau_{ij}^{ml}$. This assignment method is common to all model-based clustering methods. $\text{gmix}$ is very popular and implemented in R’s MCLUST package (Fraley and Raftery, 2006). As illustrated in Section 1 and proven in Hennig (2004), the method can be strongly affected by outliers and deviations from the model assumptions, and we now turn to approaches that attempt to deal with this problem.

**ML–type estimator for Gaussian mixtures with uniform noise (gmix.u).** Banfield and Raftery (1993) suggested to accommodate “noise” such as outliers by adding a uniform mixture component to (2.1), calling it “noise component”, which is implemented in the \texttt{mclust} package mentioned above. The resulting density for the data is

$$m_u(x; \theta) := \pi_0 1 \{x \in S\} \frac{1}{V_n} + \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \Sigma_j),$$

(2.3)

where $S \subset \mathbb{R}^p$ with volume $V_n$, $\pi_0 \in [0,1)$, $\pi_0 + \sum_{j=1}^{G} \pi_j = 1$, $\theta$ now also includes $\pi_0$. The parameter $\pi_0$ represents the proportion of the uniform noise component. Banfield and Raftery (1993) proposed to fix $S$ as the convex hull of the data, although the MCLUST package actually uses a hyperrectangle that includes $S$ (Coretto and Hennig (2011) noted that this does not normally yield a proper ML-estimator). $\theta$ is estimated by maximizing the log-likelihood function corresponding to (2.3). Denoting this estimator $\theta_n^{ml}$, the estimated posterior probability that a point $x_i$ belongs to the noise component is given by

$$\tau_{i0}^{ml} = \left(\frac{\pi_0}{V_n}\right) / m_u(x_i; \theta_n^{ml}).$$

Points are then assigned to clusters or the noise component as above. A drawback is that $V_n$ may seriously be affected by outlying points, reducing the formal breakdown point of the method to zero (Hennig, 2004).

**ML for mixtures of Student–t distributions (tmix).** McLachlan and Peel (2000b) propose to replace the Gaussian densities in (2.1) with multivariate Student-t densities, because they have heavier tails and can therefore accommodate outliers in a better way. Observations can be declared “noise” if they lie so far away from their cluster center that they are in a low density area with a probability lower than some pre-specified small $\alpha$. Let $f(x; \mu, S, v)$ be the non-central Student-t density in $\mathbb{R}^p$ with expectation $\mu$, positive definite scale matrix $S$, and $v$ degrees of freedom. Consider the density

$$m_t(x; \theta) := \sum_{j=1}^{G} \pi_j f(x; \mu_j, S_j, v_j),$$

(2.4)

where $\theta$ contains all mixture component parameters to be fitted by ML. Degrees of freedom can be fixed a priori or can be estimated. The unknown parameter vector is estimated by ML. A point $x_i$ assigned to the $j$th mixture component can be defined as outlier if

$$(x_i - \mu_j^{ml})' (S_j^{ml})^{-1} (x_i - \mu_j^{ml}) \geq q(\alpha),$$

(2.5)

where $\mu_j^{ml}$ and $S_j^{ml}$ are the resulting ML estimates for the model (2.4). McLachlan and Peel (2000a) suggest $q(\alpha) = \chi^2_{p,(1-\alpha)}$, the $(1-\alpha)$-quantile of the $\chi^2$ distribution with $p$ degrees of freedom.
degrees of freedom. Hennig (2004) showed that this method is not breakdown-robust.

**TCLUST.** Gallegos (2002) and Gallegos and Ritter (2005) proposed the “spurious outliers model” as a probabilistic framework for robust crisp cluster analysis based on Gaussian distributions (more recent theoretical results are given in Gallegos and Ritter (2013)). García-Escudero et al. (2008) proposed a modification allowing different clusters to have different weights. This amounts to maximizing the weighted likelihood function

\[
\left\{ \prod_{j=1}^{G} \prod_{i \in R_j} \pi_j \phi(x_i; \mu_j, \Sigma_j) \right\} \left\{ \prod_{i \not\in R} g_i(x_i) \right\},
\]

(2.6)

where \( R = \bigcup_{j=1}^{G} R_j \) is the set of indexes of clustered observations with \( \# \{ R \} = [n(1-\alpha)] \), and \( g_i(\cdot) \) are the densities modelling outlying observations, \( \pi_j \) is the \( j \)th cluster’s weight with \( \pi_j \in [0,1] \) and \( \sum_{j=1}^{G} \pi_j = 1 \). Let \( \theta \) be the unknown parameter vector including all triplets \( \pi_j, \mu_j, \Sigma_j \) for all \( j = 1, 2, \ldots, G \) (in the original spurious outlier model, \( \pi_j = 1 \) \( \forall j \), defining a “fixed partition model” with parameters for the cluster membership of every point). \( \alpha > 0 \) is the trimming rate, i.e., the proportion of points interpreted as outliers/noise. The authors show that for maximization of (2.6) it suffices to maximize of the left-hand side term of (2.6) and to “trim” the outliers, so that the problem boils down to

\[
\theta_{\text{tclust}} := \arg \max_{\theta \in \Theta, \# \{ R \} = [n(1-\alpha)]} \sum_{j=1}^{G} \sum_{i \in R_j} (\log \pi_j + \log \phi(x_i; \mu_j, \Sigma_j)).
\]

(2.7)

As for the previously introduced methods, constraints will be discussed in detail in Section 3. Under mild regularity conditions, García-Escudero et al. (2008) propose an algorithm for computing \( \theta_{n, \text{tclust}} \) and provide consistency theory. The TCLUST methodology is implemented in R’s TCLUST package by Fritz et al. (2012). Partition methods with trimming started with the trimmed \( k \)-means proposal of Cuesta-Albertos et al. (1997).

**Dortmund data.** Substantial disagreement may exist even between different robust clustering methods. Applying the methods implemented in tclust and discussed in García-Escudero et al. (2011), \( \alpha = 0.2 \) was found to be a good trimming rate for TCLUST with \( G = 4 \) fixed for the Dortmund dataset. The resulting clustering is compared with OTRIMLE’s in Figure 2. The major difference is that most of what was assigned to OTRIMLE’s cluster no. 4 is classified as noise by TCLUST, which is mainly due to the higher trimming rate. Again somewhat different clusterings with \( G = 4 \) are produced by MCLUST with noise component and a mixture of \( t \)-distributions and can be seen in Coretto and Hennig (2014).

**Further existing work.** More approaches to robust model-based clustering can be found in the literature. Neykov et al. (2007) proposed and implemented a trimmed likelihood method. Qin and Priebe (2013) introduce an EM-algorithm adapted to maximum \( L_q \)-likelihood estimation and study its behaviour under a gross error model. References to other approaches to robust clustering are given in García-Escudero et al. (2010).
3 Optimally tuned robust improper maximum likelihood

3.1 Robust improper maximum likelihood

The robust improper maximum likelihood estimator (RIMLE) is based on the “noise component”-idea for robustification of the Gaussian mixture MLE (2.3). The main idea is to use a pseudo-model where the noise is represented by an improper constant density over the whole Euclidean space:

\[
\psi_\delta(x, \theta) = \pi_0 \delta + \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \Sigma_j),
\]

(3.1)

with \(\pi_0, \pi_j \in [0, 1]\) for \(j = 1, 2, \ldots, G\), \(\pi_0 + \sum_{i=1}^{G} \pi_j = 1\). \(\delta > 0\) is the improper constant density (icd). The parameter vector \(\theta\) contains all Gaussian parameters plus all proportion parameters including \(\pi_0\), i.e. \(\theta = (\pi_0, \pi_1, \ldots, \pi_G, \mu_1, \ldots, \mu_G, \Sigma_1, \ldots, \Sigma_G)\). Given the sample improper pseudo-log-likelihood function

\[
l_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \psi_\delta(x_i, \theta),
\]

(3.2)

the RIMLE is defined as

\[
\theta_n(\delta) = \arg \max_{\theta \in \Theta} l_n(\theta),
\]

(3.3)

where \(\Theta\) is an appropriate constrained parameter space discussed below. \(\theta_n(\delta)\) is then used to cluster points as for model-based clustering methods. Define pseudo posterior probabilities in analogy with (2.2):

\[
\tau_j(x_i, \theta) := \begin{cases} 
\frac{\pi_0 \delta}{\psi_\delta(x_i, \theta)} & \text{if } j = 0 \\
\frac{\pi_j \phi(x_i; \mu_j, \Sigma_j)}{\psi_\delta(x_i, \theta)} & \text{if } j = 1, 2, \ldots, G;
\end{cases}
\]

for \(i = 1, 2, \ldots, n\),

and assign the points based on the following rule

\[
J(x_i, \theta) := \arg \max_{j \in \{0, 1, 2, \ldots, G\}} \tau_j(x_i, \theta).
\]

(3.4)

Fixing \(\delta\), (3.1) does not define a proper probability model, but (3.3) yields a useful procedure for data modelled as a proportion of \((1 - \pi_0)\) of a mixture of Gaussian distributions plus a proportion of \(\pi_0\) points not assigned to any meaningful cluster. Regions of high density are rather associated with clusters than with noise, so the noise regions should be those with the lowest density. This kind of distinction can be achieved by using the uniform density as in gmix.u, but for this the presence of gross outliers the dependence of the uniform distribution on the convex hull of the data still causes a robustness problem as shown in Hennig (2004).

The optimization problem in (3.3) requires that \(\Theta\) is suitably defined, otherwise \(\theta_n(\delta)\) may not exist. As discovered by Day (1969), the likelihood of a Gaussian mixtures easily degenerates and this problem extends to (3.1) as well. Let \(\lambda_{k,j}\) an eigenvalue of \(\Sigma_j\) for some \(k = 1, 2, \ldots, p\) and \(j = 1, 2, \ldots, G\). Take a sequence \((\theta_m)_{m \in \mathbb{N}}\) such that \(\lambda_{k,j,m} \searrow 0\) and \(\mu_{j,m} = x_1\), then \(l_n(\theta_m) \to +\infty\). There are various ways to prevent the unboundedness of
\(\ln(\cdot)\). Let \(\lambda_{\max}(\theta)\) and \(\lambda_{\min}(\theta)\) be respectively the maximum and the minimum eigenvalues of the covariance matrices in \(\theta\), Coretto and Hennig (2013) adopt the “eigenratio constraint”

\[
\frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma < +\infty
\]  

(3.5)

with fixed \(\gamma \geq 1\). \(\gamma = 1\) constrains all component covariance matrices to be spherical and equal, as in \(k\)-means clustering, while \(\gamma > 1\) restricts the relative scatter discrepancy among clusters. This type of constraint has been proposed by Dennis (1981) and studied by Hathaway (1985) for one dimensional Gaussian mixtures. EM-algorithms for computing the ML of multivariate Gaussian mixtures under (3.5) have been studied by Ingrassia (2004) and Ingrassia and Rocci (2007), although asymptotic properties of the corresponding MLE have not been proved. The same constraints are used for TCLUST by García-escudero et al. (2008). There are a number of alternative constraints, see Ingrassia and Rocci (2011); Gallegos and Ritter (2009). It may be seen as a disadvantage of (3.5) that the resulting estimator will not be affine equivariant (this would require allowing \(\lambda_{\max}(\theta)/\lambda_{\min}(\theta) \to \infty\) within any component), although it is rotation equivariant and therefore standardisation of variables will enforce affine equivariance.

Although (3.5) prevents the unboundness of the likelihood in standard mixture models and TCLUST, for RIMLE this is not enough. Points not fitted by any of the Gaussian components can still be fitted by the improper uniform component. Therefore, Coretto and Hennig (2013) proposes an additional “noise proportion constraint”:

\[
\frac{1}{n} \sum_{i=1}^{n} \tau_{0}(x_{i}, \theta) \leq \pi_{\max},
\]

(3.6)

for fixed \(0 < \pi_{\max} < 1\). The quantity \(n^{-1} \sum_{i=1}^{n} \tau_{0}(x_{i}, \theta)\) can be interpreted as the estimated proportion of noise points. Setting \(\pi_{\max} = 0.5\), just implements a familiar condition in robust statistics that at most half of the data should be classified as “outliers/noise”. The resulting restricted parameter space for RIMLE is then

\[
\Theta := \left\{ \theta : \pi_{j} \geq 0 \ \forall j \geq 1; \ \pi_{0} + \sum_{j=1}^{G} \pi_{j} = 1; \ \frac{1}{n} \sum_{i=1}^{n} \tau_{0}(x_{i}, \theta) \leq \pi_{\max}; \ \frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma \right\}
\]

(3.7)

Coretto and Hennig (2013) show that \(\theta_{n}(\delta)\) exists for any \(\delta \geq 0\) if \(\#(x_{n}) > G + \lceil n\pi_{\max} \rceil\). They also show \(\theta_{n}(0)\) exists under the milder condition that \(\#(x_{n}) > G\). For \(\delta = 0\), the RIMLE reduces to ML for plain Gaussian mixtures. Let \(E_{P}f(x)\) be the expectation of \(f(x)\) under \(P\), where \(P\) is the underlying probability measure that generates \(x_{n}\). The RIMLE functional is defined as

\[
\theta^{\ast}(\delta) = \arg \max_{\theta \in \Theta_{G}} E_{P} \log \psi_{\delta}(x; \theta).
\]

(3.8)

Existence of (3.8) and consistency of \(\theta_{n}(\delta)\) on the quotient space topology identifying all loglikelihood maxima is shown in Coretto and Hennig (2013) along with establishing its breakdown point. For \(\delta = 0\) the previous result extends the consistency result for the MLE of univariate Gaussian mixtures of Redner (1981). Note that asymptotic analysis with respect to the original parameter space is not possible here because of the lack of uniqueness of (3.8).
3.2 Optimal improper density level

Occasionally, subject matter knowledge may be available aiding the choice of $\delta$, but such situations may be rather exceptional. Here we suggest a data dependent choice of $\delta$. Note that $\delta$ is not treated as a model quantity to be estimated here, but rather as a tuning device to enable a good robust clustering. The aim of the RIMLE is to approximate the density of clustered regions of points when these regions look like those produced by a Gaussian distribution. We define the “optimal” $\delta$ value as minimizer of a criterion function which measures the discrepancy of the found clusters from the Gaussian prototype. Here is some more notation. Given $\theta_n(\delta)$, define the clusterwise squared Mahalanobis distances to clusters’ centres as

$$d_{i,j,n} = (x_i - \mu_{j,n})'(\Sigma_{j,n}^{-1}(x_i - \mu_{j,n}),$$

and the clusterwise weighted empirical distribution of $d_{i,j,n}$,

$$M_j(t; \delta) = \frac{1}{\sum_{i=1}^{n} \tau_j(x_i, \theta_n(\delta))} \sum_{i=1}^{n} \tau_j(x_i, \theta_n(\delta)) 1\{d_{i,j,n} \leq t\}, \quad j = 1, 2, \ldots, G. \quad (3.9)$$

In $M_j$, the $i$th point’s distance is weighted according to the pseudo posterior probability that the $i$th observation has been generated by the $j$th mixture component. If the $j$th cluster is approximately Gaussian and $\mu_{j,n}$ and $\Sigma_{j,n}$ are good approximation of its location and scatter, we expect that squared Mahalanobis distances to $\mu_{j,n}$ of the points indeed belonging to mixture component no. $j$ (for which $\tau_j(\cdot)$ indicates the estimated probability) will approximate a $\chi^2_p$ distribution. With $\chi^2_p(a)$ being the value of the cdf of the $\chi^2_p$ distribution at $a$, define the Kolmogorov-type distance for the $j$th cluster

$$K_j(\delta) = \max_{i=1,\ldots,n} |M_j(d_{i,j,n}; \delta) - \chi^2_p(d_{i,j,n})|.$$ \quad (3.10)

The quality of the overall Gaussian approximation is then evaluated by weighting $K_j(\cdot)$ with the estimated component proportion $\pi_{j,n}$:

$$D(\delta) = \frac{1}{\sum_{j=1}^{G} \pi_{j,n}} \sum_{j=1}^{G} \pi_{j,n} K_j(\delta). \quad (3.11)$$

For a given constant $\beta \geq 0$, define the optimal icd level as

$$\delta_n = \arg\min_{\delta \in [0, \delta_{\text{max}}]} D(\delta) + \beta \pi_{0,n}.$$ \quad (3.12)

The corresponding optimally tuned RIMLE (OTRIMLE) will be denoted $\theta_n(\delta_n)$. The straightforward choice for $\beta$, formalizing the idea explained above, is $\beta = 0$. However, often in practice it is not so important that the clusters are of as precise Gaussian shape as possible. $\beta > 0$ (but normally smaller than 1) formalizes that less Gaussian shapes of clusters are tolerated if this brings the estimated noise proportion down. As can be seen in the simulation results below, choosing $\beta = \frac{1}{3}$ leads to improvements if the true clusters are $t$-distributed.

(3.12) also depends on the maximum icd level $\delta_{\text{max}}$. For too large values of $\delta$, (3.6) will enforce RIMLE solutions on the border of the parameter space (unrestricted optima would violate (3.6)) that are usually far from optimal regarding (3.12). Therefore, the choice of
\( \delta_{\text{max}} \) is not of much theoretical importance but choosing it too large will waste computing time and may create numerical problems. See Section 3.3 for a recommendation. Existence and uniqueness of \( \delta_n \) are not trivial. Section 3.3 discusses how to find a good solution.

### 3.3 Computation

For a fixed \( \delta \) the RIMLE can be appropriately computed using an Expectation–Maximization (EM)-algorithm. See Coretto and Hennig (2013) for details. The outcome of the EM-algorithm depends on the initialization. One possibility to initialize the EM-algorithm is to initialize the algorithm by assigning observations randomly to the clusters and the noise. This can be done many times and the result with the best pseudo-likelihood value can then be picked. For use in a comprehensive simulation study, running the EM-algorithm from many random initializations is too time consuming, and therefore we used an initialization method inspired by the MCLUST software (actually, even if computing time is not too big an issue, the proposed initialization method may often outperform random partitioning, so that it is recommended to let it compete with the random partitions for the best pseudo-loglikelihood): in order to avoid spurious clusters, we consider as valid initial partitions only those containing at least \( \min.pr \times n \) observations in each cluster (\( \min.pr = 0.005 \), say).

As a first attempt to find such a valid partition, nearest neighbor based clutter/noise detection proposed by Byers and Raftery (1998) is applied to identify an initial noise guess. Agglomerative hierarchical clustering based on ML criteria for Gaussian mixture models proposed by Banfield and Raftery (1993) is then used for finding initial Gaussian clusters among the non-noise. If the found partition is not “valid” in the sense above, we first perform a number of adjustments such as assigning too small clusters to the noise, and assign non-noise points randomly to clusters, in order to find a valid partition. For details see Coretto and Hennig (2014).

The OTRIMLE can be found by computing RIMLEs on a grid of \( \delta \) values ranging from zero to some large enough \( \delta_{\text{max}} \). In practice, we solve the program (3.12) by the “golden section search” of Kiefer (1953) over the candidate set \( \delta \in [0, \delta_{\text{max}}] \). In most numerical experiments we found that no more than 30 RIMLE evaluations are required. \( \delta_{\text{max}} \) can be chosen as highest density value occurring within an initialized cluster. Experience shows that within computation of the OTRIMLE at least if no random initializations are used, it is advantageous to discard \( \delta \)-values for which the RIMLE-solution ends up at the border of the parameter space (unless this happens for all \( \delta \), enforcing (3.5), (3.6) or zero component proportions, because \( D(\delta) \) may behave irregularly and the golden search may be led astray. This particularly rules out too large values of \( \delta \), which will trigger (3.6). Experience shows that the set of values of discarded \( \delta \)-values is usually a single interval, often containing the largest values in the candidate set. Discarded solutions can be brought back if they are needed in later stages of the golden section search, which is not normally necessary. Further details can be found in Coretto and Hennig (2014).

### 4 Definition of “true” clusters and misclassification

In most simulation studies in cluster analysis, in which data are generated from mixture (or fixed partition) models, it is assumed that the “true” clusters are identified with the mixture components, and methods can then be evaluated by misclassification rates. But this can
be problematic. Consider the comparison of ML-estimators for Gaussian mixtures (with or without “noise component”) and for mixtures of t-distributions. In most applications, both approaches would be considered as potentially appropriate for doing the same thing, namely finding clusters that are unimodal and elliptical in a way that is unaffected by outliers far away from such “cluster cores”. In applications in which the clustering is of main interest (as opposed to parameter estimation), researchers would not mind much whether the density around their cluster cores rather looks like a Gaussian or a t-distribution. But implications for which points are considered as “true outliers” vs. “truly belonging to a cluster” would be different, because some points generated by a t-distribution with low degrees of freedom are indeed outlying with respect to the core of the t-distribution from which they are generated.

More generally, the identification of clusters and mixture components cannot be taken for granted. Hennig (2010) illustrates that the interpretation of Gaussian mixture components as clusters depends on whether the components are separated enough or at least generate clearly different “patterns” in the data. And in robust clustering, one would often interpret a group of a few points with low density as “noise” even if they were generated by a Gaussian distribution.

We now define a “reference truth” for the mixture models that are used in our simulation study, from which misclassification rates then can be computed. For motivation, consider Figure 3, which shows an artificial dataset where 950 points have been drawn from two Gaussian distributions with distinct mean vectors and spherical covariance matrices. The remaining 50 points are drawn from a uniform distribution on the square $[-10,10] \times [-5,15]$. Figure 3(a) shows the unlabeled dataset, the starting point in a cluster analysis. Figure 3(b) shows the points labelled by the mixture components that generated them. Observe that there are three red stars (generated from the uniform noise) in the middle of the region where the two Gaussians have most of their mass. Furthermore, there are green points from the left Gaussian component with lower density that fall into the dense blue region. In fact, no method can be expected to reconstruct all the cluster memberships in such overlapping regions. Figure 3(c) illustrates the classification that we define as the “reference truth”, which is based on connected cluster/noise regions defined by the underlying mixture, which we would expect to be identified by the clustering methods.

The idea is that we choose probability measures $P_1, P_2, \ldots, P_G$ to correspond to the $G$ “true clusters” (which implies that they are chosen in such a way that they “cluster”, i.e., generate clearly distinguishable, although not necessarily non-overlapping, data patterns). For each of these, we define a region of points that can be considered as non-outliers based on a mean and covariance matrix functional, which are Fisher-consistent at the Gaussian distribution, but which are robust and exist for other distributions, too, which defines a region of non-outliers of Gaussian shape. We consider all points “noise” that are outliers according to this definition for all “cluster components”, and we apply quadratic discriminant analysis to assign points to clusters that are non-outliers with respect to more than one “cluster component”. This means that points are assigned to clusters by optimal classification boundaries under the Gaussian assumption, even if the components are in fact not Gaussian, which formalizes using “Gaussian cluster prototypes” but not necessarily assuming that clusters really have to be Gaussian.

To this end, let $m_j$ and $S_j$ be the Minimum Covariance Determinant (MCD) center and scatter functional at $P_j$ (Rousseeuw, 1985). Cator and Lopuhaä (2012) proved existence of
Figure 3: An artificial dataset consisting of 950 points drawn from two 2-dimensional Gaussian distributions and 50 points from a uniform distribution. (a) shows unlabeled points, (b) shows points colored according to the mixture component that generated them, (c) colors represent the GR \( \alpha \) concept (with \( \alpha = 10^{-4} \)), while the red stars are points belonging to NR \( \alpha \).

The MCD functional for a wide class of probability measures. The \( S_j \) can be corrected for achieving consistency at \( P_j \) equal to the Gaussian distribution, (Croux and Haesbroeck, 1999; Pison et al., 2002), so that when \( P_j \) is Gaussian \( m_j \) and \( S_j \) are the corresponding mean vector and covariance matrix. Let \( \pi_j \) be the expected proportion of points generated from \( P_j \). We allow \( \sum_{j=1}^{G} p_j \leq 1 \), so that points could be generated by (noise-)distributions other than \( P_1, P_2, \ldots, P_G \). Define the quadratic discriminant score for assigning the point \( y \in \mathbb{R}^p \) to the \( j \)th cluster by maximizing

\[
qs(y; \pi_j, m_j, S_j) := \log(\pi_j) - \frac{1}{2} \log(\det(S_j)) - \frac{1}{2} (y - m_j)' S_j^{-1} (y - m_j),
\]

for \( j = 1, 2, \ldots, G \), which is equivalent to (3.4), if clusters are generated by Gaussian distributions. Consider the ellipsoid

\[
E_\alpha(m_j, S_j) := \{ y : (y - m_j)' S_j^{-1} (y - m_j) \leq \chi^2_p(1-\alpha) \},
\]

where \( \chi^2_p(1-\alpha) \) is the \( 1-\alpha \) quantile of the \( \chi^2_p \) distribution. For a fixed choice of \( \alpha \), the set \( E_\alpha(m_j, S_j) \) defines the subset of \( \mathbb{R}^p \) that hosts the \( j \)th cluster. The size of this ellipsoid is defined in terms of \( \chi^2_p(1-\alpha) \), because for Gaussian \( P_j \), \( P_j(\mathbb{R}^p \setminus E_\alpha(m_j, S_j)) = \alpha \). For a fixed level \( \alpha \), the \( \alpha \)-Gaussian Region is defined as the union of these ellipsoids:

\[
\text{GR}_\alpha := \bigcup_{j=1}^{G} E_\alpha(m_j, S_j),
\]

while the noise region is given by \( \text{NR}_\alpha := \mathbb{R}^p \setminus \text{GR}_\alpha \).

**Definition 1** (\( \alpha \)-Gaussian cluster memberships). Given \( \alpha \in [0, 1] \), a data generating process (DGP) with cluster parameters \( \theta_C := \{(\pi_j, m_j, S_j), \; j = 1, 2, \ldots, G\} \), and a dataset \( x_n := \{x_1, x_2, \ldots, x_n\} \); the \( \alpha \)-Gaussian cluster memberships are given by the following
assignment rule

\[
\text{AGR}_\alpha(x_i; \theta_C) := 1\{x_i \in \text{GR}_\alpha\} \times \arg \max_{j=1,2,...,G} \{j = \arg \max_{g=1,...,G} q_s(y; \pi_g, m_g, S_g)\}. 
\]

(4.1)

\(\text{AGR}_\alpha(x_i, \theta_C) = 0\) means that \(x_i \in \text{NR}_\alpha\).

This definition is inspired by the definition of outliers with respect to a reference model as in Davies and Gather (1993), Becker and Gather (1999). A difference is that here the parameter \(\alpha\) does not directly control the probability of the noise region. Once \(\alpha\) and the triples \((\pi_j, m_j, S_j)\) are fixed for all \(j = 1,2,...,G\), the size of the noise region will depend on the degree of overlap and Gaussianity of the ellipsoids in \(\text{GR}_\alpha\). \(\alpha\) needs to be small because the idea of an outlier implies that under the Gaussian distributions outliers are very rare. We choose \(\alpha = 10^{-4}\), which implies that the probability that there is at least one outlier in \(n = 500\) i.i.d. Gaussian observations is 0.0488. For a given dataset “true” clusters are defined according to the assignment rule introduced next.

The different robust clustering methods have different implicit ways of classifying points as “noise” (noise component, trimming, outlier identification in t-distributions). In order to make them comparable, we use (4.1) to unify the point assignment of the methods by computing \(\text{AGR}_\alpha(\cdot)\) based on the parameters estimated by the methods, from which we assume that estimators of the triples \((\pi_j, m_j, S_j)\) (cluster proportion, center, and covariance matrix) can be computed (see Section 5.2). Let the estimated cluster parameters be \(\hat{\theta}_{C,n}\). Let \(\sigma\{\cdot\}\) be a permutation over a subset of the integers \(\{0,1,...,G\}\). We compute the misclassification rate as

\[
\text{mcr}(\theta_C, \hat{\theta}_{C,n}) = \arg \min_{\sigma} \frac{1}{n} \sum_{i=1}^{n} 1\{\text{AGR}_\alpha(x_i; \theta_C) = \sigma\{\text{AGR}_\alpha(x_i; \hat{\theta}_{C,n})\}\},
\]

(4.2)

so that membership relabeling does not affect the performance measure.

The vector of \(\theta_C\) with cluster parameters is based on the MCD location and scatter computed at each \(P_j\) in the DGP. When \(P_j\) is the Gaussian distribution \(m_j\) and \(S_j\) coincide with the Gaussian expectation and covariance matrix. However, in the following simulation study deviations from Gaussianity are considered by exploring \(P_j\) that are: (i) non-central multivariate Student-t distributed, (ii) non-central multivariate Student-t on some marginals and Gaussian on other marginals. In these cases, appropriate MCD parameters have to be derived to obtain the reference \(\theta_C\). Although the required MCD parameters exist for the cases we will consider here (Cator and Lopuhaä, 2012), it is not easy to compute them in closed form. For non-Gaussian elements of the DGP we computed the MCD parameters by Monte Carlo integration based on \(10^6\) random draws, assuming zero mean and unit scale matrix and using affine equivariance of the MCD for transformation, by the \texttt{cov.rob} function of the R package MASS (Venables and Ripley, 2002).

5 A comparative simulation study

Here we present a comprehensive simulation study comparing the OTRIMLE with the methods introduced in Section 2.
5.1 Data generating processes

Clustering methods are compared on a total of 24 DGPs with 1000 Monte Carlo-replicates from each DGP. Half of the DGPs produce 2-dimensional datasets. The remaining twelve DGPs are 20-dimensional versions that are constructed adding independent 18-dimensional uncorrelated zero-means unit-variance Gaussian and/or Student-t marginals (note that different from some other literature, we do not use the term “noise” for these, but rather, as before, for observations not belonging to any cluster). Therefore clusters are always only defined on the first two marginals. Note that the aim of the simulation study is not variable selection; we designed the DGPs so that clustering information is only in the first two dimensions in order to be able to visualize and control the clustering patterns. Still we compare clustering methods that use all variables. Of course it could be of interest to try variable selection methods with these DGPs, but we do not think that variable selection or dimension reduction is mandatory in clustering, because the meaning of the clusters is determined by the involved variables and there may be subject matter reasons to include all variables, see the discussion rejoinder in Hennig and Liao (2013).

For a design based on $G$ clusters, a DGP produces $n$ points by sampling from distributions $P_1, P_2, \ldots, P_G$ so that an expected number of points $n\pi_j$ is sampled under $P_j$ for $j = 1, 2, \ldots, G$. For those designs noise is generated by adding a further component that is denoted by $P_0$. The expected proportion of points under $P_0$ is denoted by $\pi_0$. In each Monte Carlo replicate sampling is performed as follows: the vector $(n_0, n_1, \ldots, n_G)'$ is sampled from a Multinomial$(n, \pi_0, \pi_1, \ldots, \pi_G)$ distribution, than $n_j$ points are independently sampled from $P_j$ for all $j = 1, 2, \ldots, G$. Therefore, in each Monte Carlo sample is an iid sample from the mixture distribution $\pi_0 P_0 + \sum_{j=1}^G \pi_j P_j$. We choose $n = 1000$ for 2-dimensional designs and $n = 2000$ for the 20-dimensional versions. DGPs have been designed in order to test a variety of “noise patterns”, numbers of clusters $G$, and patterns of separation/overlap between different clusters.

We consider two main classes of DGP, namely DGPs with a uniform noise component on the first two marginals, and DGPs that do not have a noise component. The first group includes the following setups, all of which have clusters generated from Gaussian distributions, and for $p = 20$ the 18 uninformative variables are Gaussian: (i) for “WideNoise” DGPs, the uniform noise component produces points that are widespread but overlap with the clustered regions entirely; (ii) for “SideNoise” DGPs the uniform noise component spreads points on a wide region that overlaps slightly with some of the clusters; (iii) in “SunSpot” DGPs there is a uniform component that produces few extremely outlying points. On the other hand we consider DGPs that do not include a noise component (i.e. $\pi_0 = 0$). This second group can be divided into three further subgroups: (i)/(ii) in “GaussT” and “TGauss” DGPs, multivariate Student-t distributions with three degrees of freedom are used. In “GaussT” these are used as uninformative distributions for $p = 20$, whereas the first two clustered dimensions use Gaussians; in “TGauss” the clusters are generated by noncentral multivariate $t_3$-distributions and for $p = 20$ the 18 uninformative variables are Gaussian; (iii) in “Noiseless” DGPs, all points are drawn from Gaussian distributions.

For each of the six setups, there are variants with a lower and a higher number of clusters $G$, $p = 2$ (denoted by “l”) and $p = 20$ (as explained above; denoted by “h”), adding up to 24 DGPs. The nomenclature used in the following puts these at the end of the setup name, i.e., “TGauss.5h” refers to the “TGauss”-setup with higher $G = 5$ and higher $P = 20.$
For “WideNoise”, “SideNoise”, and “GaussT”, the lower $G$ was 2 and the higher $G$ was 3. For “SunSpot”, “TGAuss” and “Noiseless”, the lower $G$ was 3 and the higher $G$ was 5. The overlap between clusters as well as the combinations of cluster shapes varied between DGP’s. Full details of the definition of the DGP’s are given in Coretto and Hennig (2014) together with exemplary scatterplots of the first two dimensions of a dataset from every setup.

### 5.2 Implementation of methods

Table 1 summarizes settings for the compared methods. TCLUST and RIMLE/OTRIMLE are based on eigenratio constraints, but this is not the case for the MCLUST software (Fraley and Raftery, 2006) and the available implementation of mixtures of t-distributions (McLachlan and Peel, 2000b). In order to have full comparability of the solutions, the eigenratio constraints have been implemented for all methods including gmix, tmix and gmix.u, where the latter is computed by use of the same routine that is used for RIMLE/OTRIMLE. TCLUST based methods have been performed by the TCLUST R package, which has its own implementation of the constraints. Therefore, eigenratio constraints are implemented as for the RIMLE (see Section 3.3) except for TCLUST methods. For all methods the eigenratio constraint has been set equal to 20 for each of the 24 DGP’s. The latter choice is motivated by the fact that 20 is larger than the maximum true eigenratio across the designs involved in the comparison, and it is in general a value that often enables rather smooth optimization (obviously in reality the true eigenvalue ratios are not known, but for variables with comparable scales and value ranges, 20 gives the covariance matrices enough flexibility for most applications). OTRIMLE has been tested with and without the penalty term $\beta = \frac{1}{3}$ in (3.12), denoted by ot.rimle ($\beta = 0$) and ot.rimle.p respectively. Other values for $\beta$, ranging from 0.1 to 0.5, have been tried, and results did not change much. A difficulty with TCLUST is that an automatic data driven choice of the trimming level is currently not available. In tclust.f we set the trimming level to 10%. This choice is motivated by the fact the DGP’s produced an average proportion of points belonging to the NR set in the range [0%, 23%] (see Table 1 in Coretto and Hennig, 2014). Furthermore, since the trimming level plays a role similar to $\delta$ in RIMLE/OTRIMLE, there are two versions of TCLUST for which the same idea for automatic decisions the trimming level is used as proposed here for OTRIMLE, see Section 3. In ot.tclust and ot.tclust.p the trimming level has been selected using (3.12) with the trimming level playing the role of $\delta$, and the weights $\tau_j(\cdot)$ in $M_j(\cdot)$ replaced by the 0-1 crisp weights of TCLUST, again using $\beta = 0$ or $\beta = \frac{1}{3}$. For t-mixtures, in the original proposal by McLachlan and Peel (2000b), the degrees of freedom are assumed to be equal across the mixture components and are estimated from the data and covariance matrix constraints are not considered. In tmix we fix the degrees of freedom to 3 and we incorporate eigenratio constraints, as for the other methods. This is motivated by the following arguments: (i) for some of the DGP (particularly the SunSpot designs) constraints were needed in order to avoid spurious solutions; (ii) for some of the sampling designs not based on Student-t distributions, estimation of the degrees of freedom of the t-distribution produced an extremely large variability in the misclassification rates; (iii) since a number of designs are based on Student-t with 3 degrees of freedom, the decision to fix this parameter to 3 can be seen as giving the t-mixture a slight advantage, which seems fair given that the majority of setups rather seem to favour noise component/trimming.
Table 1: Summary of the main settings for the methods under comparison.

| Method    | Setup                                                                 |
|-----------|------------------------------------------------------------------------|
| gmix.u    | RIMLE with $\delta = 1/V_n$ where $V_n$ is volume of the smallest hyperrectangle that contains the data. |
| rimle.o   | OTRIMLE without penalty ($\beta = 0$).                                    |
| rimle.op  | OTRIMLE with penalty term $\beta = 1/3$.                                 |
| tclust.f  | TCLUST with fixed trimming level set at 0.1.                             |
| tclust.o  | TCLUST with trimming level selected by the OTRIMLE criterion without penalty ($\beta = 0$). |
| tclust.op | TCLUST with trimming level selected by the OTRIMLE criterion with penalty term $\beta = 1/3$. |
| tmix      | ML for the Student-t mixture model (2.1) with $v = 3$ for all components plus eigenratio constraints. |
| gmix      | ML for the Gaussian mixture model (2.4) plus eigenratio constraints.     |

For some of the DGPs the experience suggested that solutions may depend strongly on the initialization. In order to reduce the bias introduced by different initializations, all methods are initialized from the same partition. Notice that the TCLUST package does not allow to fix an initial partition, instead it allows to set the number of random initializations. In order to compare all methods from the same initial partition an additional set of R functions has been provided by TCLUST’s authors. The definition of the initial partition is inspired by MCLUST here. In order to avoid spurious clusters, we consider as valid initial partitions only those containing at least $\text{min.pr} \times n$ observations in each cluster ($\text{min.pr} = 0.005$, say). As a first attempt to find such a valid partition, nearest neighbor based clutter/noise detection proposed by Byers and Raftery (1998) is applied to identify an initial noise guess. Agglomerative hierarchical clustering based on ML criteria for Gaussian mixture models proposed by Banfield and Raftery (1993) is then used for finding initial Gaussian clusters among the non-noise. If the found partition is not “valid” in the sense above, we first perform a number of adjustments such as assigning too small clusters to the noise, and assign non-noise points randomly to clusters, in order to find a valid partition. Notice that rimle.o, rimle.op, tclust.o and tclust.op require to compute the RIMLE solution and the TCLUST solution for different values of $\delta$ and trimming level respectively. For a given dataset all RIMLE and TCLUST solutions are computed from the same starting partition as defined previously. Details about the initial partition are given in Coretto and Hennig (2014).

5.3 Results

The methods are compared using misclassification rates as defined in (4.2). These are more relevant in clustering tasks than parameter estimates. The results are graphically summarized in Figure 4, while average misclassification rates with standard errors are given in Table 2. Each square in the plot is a color-coded representation of the misclassification rate averaged over the 1000 Monte Carlo replicates for a given method-DGP pair. The gray-scaling of the plot is not linear. It is constructed in order to emphasize differences in the range between 0%-20% (differences between very high values are not important because if a method gets the clustering seriously wrong, precise differences are
Figure 4: Level plot representing the sample mean of the Monte Carlo distribution of misclassification rates (percentage scale) for each DGP-method pair. Each square of the plot represents the average misclassification according to the bottom grey color scale.

not very informative). Further details about average misclassification rates are given in the online supplent. Coretto and Hennig (2014) also contains boxplots of the misclassification rates for all method-DGP pairs. Figure 4 shows clear evidence that using robust methods is important. The gmix method only performs well for the Noiseless and some DGPs with Gaussians and t-distributions, but most other methods work well (although slightly worse at times) for these DGPs, too. tmix works well for most DGPs involving t-distributions, but for the other DGPs with noise/outliers, it is often seriously worse than gmix.u, OTRIMLE and TCLUS.

gmix.u performs relatively well, although for a number of DGPs it suffers strongly from high dimensionality. For the 2-dimensional versions of WideNoise, gmix.u in fact is a proper ML-estimator, so the method should be advantageous for these DGPs, and gmix.u is indeed best for these DGPs. However, for the 20-dimensional WideNoise DGPs, $\delta = 1/V_n$ on the smallest hyperrectangle containing the data is no longer the ML estimate for the noise-generating mixture component, and its performance deteriorates strongly. Misclassification rates are generally higher for the 20-dimensional DGPs, also for the other methods.

Regarding the TCLUS methods, even though the automatic trimming level of ot.tclust and ot.tclust.p did not always improve the results, it demonstrated to provide a reasonable choice for the trimming level. In fact, for all situations where the true average noise proportion is about equal to the trimming level of tclust, performances of tclust, ot.tclust and ot.tclust.p are very similar, meaning that the OTRIMLE criterion is a good starting point for fixing the trimming rate. Compared to the RIMLE-type methods, the performance of TCLUS suffers in situations where there is a considerable degree of overlap between clusters. For DGPs with overlap (such as WideNoise.2, SunSpot.5, GaussT.2 and
Table 2: Monte Carlo average misclassification rates (%) with their standard errors in brackets. Misclassification rates are computed as in (4.1). Notice that both averages (and standard errors) are reported in percentage scale.

| DGP          | gmix.u | ot.rimle | ot.rimle.p | tclust | ot.tclust | ot.tclust.p | tmix | gmix |
|--------------|--------|----------|------------|--------|-----------|-------------|------|------|
| WideNoise.2l | 1.33(0.02) | 5.00(0.29) | 1.35(0.02) | 12.70(0.33) | 13.11(0.32) | 10.20(0.32) | 17.48(0.10) | 18.40(0.04) |
| WideNoise.2h | 17.36(0.04) | 5.28(0.05) | 5.40(0.06) | 22.94(0.13) | 26.20(0.32) | 25.85(0.25) | 18.90(0.03) | 17.43(0.04) |
| WideNoise.3l | 0.34(0.01) | 0.42(0.01) | 0.40(0.01) | 0.44(0.01) | 0.38(0.01) | 0.39(0.01) | 3.10(0.02) | 7.79(0.13) |
| WideNoise.3h | 7.59(0.18) | 3.16(0.09) | 4.27(0.09) | 0.72(0.01) | 1.31(0.02) | 3.04(0.06) | 3.91(0.04) | 8.56(0.15) |
| SideNoise.2l | 0.01(0.00) | 0.03(0.01) | 0.06(0.02) | 0.18(0.01) | 0.02(0.00) | 0.01(0.00) | 0.01(0.00) | 0.01(0.00) |
| SideNoise.2h | 0.03(0.01) | 0.09(0.02) | 0.15(0.03) | 0.20(0.01) | 0.20(0.01) | 0.06(0.02) | 0.06(0.02) | 0.06(0.02) |
| SideNoise.3l | 0.13(0.00) | 0.19(0.03) | 0.24(0.04) | 0.25(0.01) | 0.19(0.01) | 0.07(0.00) | 0.06(0.00) | 18.89(0.47) |
| SideNoise.3h | 0.01(0.00) | 0.03(0.01) | 0.06(0.01) | 0.11(0.03) | 0.11(0.03) | 0.11(0.03) | 0.11(0.03) | 0.11(0.03) |
| Sunspot.3l   | 0.11(0.00) | 0.12(0.00) | 0.11(0.00) | 0.21(0.00) | 0.13(0.01) | 2.17(0.36) | 34.65(0.11) | 35.38(0.19) |
| Sunspot.3h   | 0.24(0.00) | 0.24(0.00) | 0.24(0.00) | 21.38(0.95) | 27.19(1.00) | 36.83(1.01) | 34.75(0.06) | 40.00(0.37) |
| Sunspot.5l   | 11.72(0.10) | 11.63(0.10) | 11.65(0.10) | 11.88(0.10) | 11.63(0.10) | 11.61(0.10) | 13.84(0.12) | 13.30(0.13) |
| Sunspot.5h   | 1.84(0.02) | 4.56(0.09) | 0.90(0.01) | 0.81(0.01) | 1.63(0.03) | 0.95(0.01) | 3.26(0.02) | 3.32(0.06) |
| TGauss.3l    | 0.09(0.01) | 0.53(0.01) | 0.51(0.01) | 0.77(0.01) | 0.51(0.01) | 0.49(0.01) | 0.83(0.01) | 0.85(0.02) |
| TGauss.3h    | 1.77(0.02) | 4.65(0.07) | 1.20(0.02) | 1.21(0.01) | 2.55(0.04) | 1.32(0.02) | 3.37(0.02) | 3.59(0.05) |
| TGauss.5l    | 0.92(0.02) | 0.57(0.01) | 0.09(0.01) | 1.39(0.01) | 0.96(0.01) | 0.89(0.01) | 1.43(0.02) | 1.65(0.04) |
| GaussT.2l    | 0.57(0.01) | 0.57(0.01) | 0.56(0.01) | 28.77(0.33) | 21.50(0.48) | 16.32(0.48) | 8.83(0.02) | 0.57(0.01) |
| GaussT.2h    | 32.45(0.05) | 33.60(0.14) | 44.91(0.10) | 42.42(0.13) | 41.72(0.15) | 43.13(0.13) | 31.32(0.16) | 32.17(0.03) |
| GaussT.3l    | 1.11(0.00) | 1.11(0.00) | 1.11(0.00) | 0.27(0.01) | 0.13(0.00) | 0.12(0.00) | 0.17(0.01) | 0.11(0.00) |
| GaussT.3h    | 12.32(0.12) | 3.69(0.03) | 3.44(0.03) | 9.87(0.03) | 12.22(0.04) | 6.85(0.08) | 18.34(0.03) | 36.23(0.49) |
| Noiseless.3l | 0.12(0.00) | 0.12(0.00) | 0.12(0.00) | 0.28(0.01) | 0.15(0.01) | 0.13(0.00) | 0.18(0.01) | 0.12(0.00) |
| Noiseless.3h | 0.25(0.00) | 0.25(0.00) | 0.25(0.00) | 0.38(0.00) | 0.28(0.00) | 0.28(0.00) | 0.29(0.00) | 0.25(0.00) |
| Noiseless.5l | 4.53(0.08) | 4.47(0.08) | 4.43(0.08) | 14.43(0.13) | 13.33(0.13) | 13.14(0.14) | 6.33(0.14) | 4.58(0.09) |
| Noiseless.5h | 12.39(0.10) | 12.63(0.10) | 12.62(0.10) | 16.69(0.12) | 16.13(0.12) | 16.10(0.12) | 11.92(0.08) | 12.37(0.10) |

Noiseless.5) the misclassification rate of TCLUST is completely dominated by misclassifications between clusters (to see this consider Table 2 and Table 4 in Coretto and Hennig (2014)). This is due to the fact the TCLUST parameters are based on a classification-type likelihood, which depends too much on the separation between clusters. The TCLUST also seems to not tolerate the large number of Student-t marginals of GaussT.2h and GaussT.3h. TCLUST performs well for a number of DGPs and is clearly best in WideNoise.2h (particularly tclust and ot.tclust, although ot.tclust.p beats ot.tclust for the DGFs with Gaussian and t-marginals).

The OTRIMLE methods show a very good overall performance. They produce high misclassification rates only for some 20-dimensional DGPs for which all methods are in trouble (performances for GaussT.2h are generally bad with even tmix, the best method there, producing an average misclassification rate of more than 30%), and they are best for a number of DGPs, particularly 20-dimensional WideNoise and some TGAuss-DGPS. The comparison between ot.rimle and ot.rimle.p is mixed (as between ot.tclust and ot.tclust.p), with $\beta = \frac{1}{3}$ improving matters clearly for TGAuss.2l and TGAuss.3l (the shape of the t-distribution encourages ot.rimle to assign too many points to the noise; see Tables 2 and 3 in Coretto and Hennig (2014)), but being significantly worse for WhiteNoise.3h.
6 Concluding Remarks

This paper aims at establishing the OTRIMLE as a valuable robust alternative to what is already in the literature. To our knowledge, we ran the most comprehensive simulation study to date comparing robust model-based clustering methods for “Gaussian-shaped” clusters, and on the way we touched a number of critical issues such as the definition of “true” clusters and making methods comparable that formally estimate different parameters although they will be used in practice in mostly the same kind of situation.

Despite our effort to make the simulation study fair, we would like to remind readers modestly that one should generally not expect a perfectly fair comparison from authors who want to introduce a specific method and compare it with competitors from the literature. Every method was best for certain DGPs in the simulation study, and therefore, by focussing on just the DGPs that a certain method does best, simulation studies could be designed that make any method “win”. Readers need to make their own mind up about to what extent our study covered situations that are important to them. One of our major aims was to confront all methods with DGPs that do not exactly match their (implicit or explicit) model assumptions, but for which the methods nevertheless could be legitimately used. Also it is important that our simulation study did not use the available original software for competing methods, but used all methods in a more unified implementation that allowed better comparison. We do not think that this gave OTRIMLE an unduly advantage. In fact, we incorporated crucial ideas from both MCLUST (initialization) and TCLUST (eigenvalue ratio constraints), and the combination of these ideas used here could actually be beneficial for all methods.

References

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

Becker, C. and U. Gather (1999). The masking breakdown point of multivariate outlier identification rules. Journal of the American Statistical Association 94(447), 947–955.

Byers, S. and A. E. Raftery (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. Journal of the American Statistical Association 93(442), 577–584.

Cator, E. A. and H. P. Lopuhaä (2012). Central limit theorem and influence function for the MCD estimators at general multivariate distributions. Bernoulli 18(2), 520–551.

Coretto, P. and C. Hennig (2010). A simulation study to compare robust clustering methods based on mixtures. Advances in Data Analysis and Classification 4(2), 111–135.

Coretto, P. and C. Hennig (2011). Maximum likelihood estimation of heterogeneous mixtures of gaussian and uniform distributions. Journal of Statistical Planning and Inference 141(1), 462–473.

Coretto, P. and C. Hennig (2013). Finding approximately Gaussian clusters via robust improper maximum likelihood. arXiv:1309.6895 preprint available at http://arxiv.org/pdf/1309.6895.
Coretto, P. and C. Hennig (2014). Supplement to: “Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust Gaussian clustering”. Available at http://arxiv.org/src/1406.0808/anc/supplement.pdf.

Croux, C. and G. Haesbroeck (1999). Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis* 71(2), 161–190.

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

Davies, L. and U. Gather (1993). The identification of multiple outliers. *Journal of the American Statistical Association* 88(423), 782–792.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika* 56, 463–474.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)* 39(1), 1–47.

Dennis, J. E. J. (Ed.) (1981). *Algorithms for nonlinear fitting*, Cambridge, England. NATO advanced Research Symposium: Cambridge University Press.

Fraley, C. and A. E. Raftery (2006, September). MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics.

Fritz, H., L. A. García-Escudero, and A. Mayo-Iscar (2012). tclust: An R package for a trimming approach to cluster analysis. *Journal of Statistical Software* 47(12), 1–26.

Gallegos, M. T. (2002). Maximum likelihood clustering with outliers. In *Classification, Clustering, and Data Analysis*, pp. 247–255. Springer.

Gallegos, M. T. and G. Ritter (2005). A robust method for cluster analysis. *Annals of Statistics* 33(5), 347–380.

Gallegos, M. T. and G. Ritter (2009). Trimmed ML estimation of contaminated mixtures. *Sankhya (Ser. A)* 71, 164–220.

Gallegos, M. T. and G. Ritter (2013). Strong consistency of k-parameters clustering. *Journal of Multivariate Analysis* 117, 14–31.

García-Escudero, L. A., A. Gordaliza, Matrán, and A. Mayo-Iscar (2011). Exploring the number of groups in robust model-based clustering. *Statistics and Computing* 21, 585–599.

García-Escudero, L. A., A. Gordaliza, C. Matrán, and A. Mayo-Iscar (2008). A general trimming approach to robust cluster analysis. *Annals of Statistics* 38(3), 1324–1345.

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

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics* 13, 795–800.
Hennig, C. (2004). Breakdown points for maximum likelihood estimators of location-scale mixtures. *The Annals of Statistics* 32(4), 1313–1340.

Hennig, C. (2010). Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification* 4, 3–34.

Hennig, C. and T. F. Liao (2013). How to find an appropriate clustering for mixed type variables with application to socioeconomic stratification (with discussion). *Journal of the Royal Statistical Science, Series C (Applied Statistics)* 62, 309–369.

Ingrassia, S. (2004). A likelihood-based constrained algorithm for multivariate normal mixture models. *Statistical Methods & Applications* 13(2), 151–166.

Ingrassia, S. and R. Rocci (2007). Constrained monotone EM algorithms for finite mixture of multivariate gaussians. *Computational Statistics and Data Analysis* 51, 5339–5351.

Ingrassia, S. and R. Rocci (2011). Degeneracy of the EM algorithm for the MLE of multivariate gaussian mixtures and dynamic constraints. *Computational statistics & data analysis* 55(4), 1715–1725.

Kiefer, J. (1953). Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society* 4(3), 502–506.

McLachlan, G. J. and D. Peel (2000a). *Finite Mixture Models*. New York: Wiley.

McLachlan, G. J. and D. Peel (2000b). Robust mixture modelling using the t–distribution. *Statistics and Computing* 10(4), 339–348.

Neykov, N., P. Filzmoser, R. Dimova, and P. Neytchev (2007). Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics and Data Analysis* 17(3), 299–308.

Pison, G., S. V. Aelst, and G. Willems (2002). Small sample corrections for LTS and MCD. *Metrika* 55(1-2), 111–123.

Qin, Y. and C. E. Priebe (2013). Maximum lq-likelihood estimation via the expectation-maximization algorithm: A robust estimation of mixture models. *Journal of the American Statistical Association* 108(503), 914–928.

Redner, R. (1981). Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions. *The Annals of Statistics* 9, 225–228.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. *Mathematical statistics and applications* 8, 283–297.

Sommerer, E.-O. and C. Weihs (2005). Introduction to the contest “social milieus in dortmund”. In *Classification - the Ubiquitous Challenge*, Berlin, pp. 667–673. Springer.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.