Adv Data Anal Classif (2010) 4:111–135
DOI 10.1007/s11634-010-0065-4

REGULAR ARTICLE

A simulation study to compare robust clustering methods based on mixtures

Pietro Coretto · Christian Hennig

Received: 30 November 2009 / Revised: 4 June 2010 / Accepted: 8 June 2010 /
Published online: 26 June 2010

Abstract The following mixture model-based clustering methods are compared in a simulation study with one-dimensional data, fixed number of clusters and a focus on outliers and uniform "noise": an ML-estimator (MLE) for Gaussian mixtures, an MLE for a mixture of Gaussians and a uniform distribution (interpreted as "noise component" to catch outliers), an MLE for a mixture of Gaussian distributions where a uniform distribution over the range of the data is fixed (Fraley and Raftery in Comput J 41:578–588, 1998), a pseudo-MLE for a Gaussian mixture with improper fixed constant over the real line to catch "noise" (RIMLE; Hennig in Ann Stat 32(4): 1313–1340, 2004), and MLEs for mixtures of t-distributions with and without estimation of the degrees of freedom (McLachlan and Peel in Stat Comput 10(4):339–348, 2000). The RIMLE (using a method to choose the fixed constant first proposed in Coretto, The noise component in model-based clustering. Ph.D thesis, Department of Statistical Science, University College London, 2008) is the best method in some, and acceptable in all, simulation setups, and can therefore be recommended.

Keywords Model-based clustering · Gaussian mixture · Mixture of t-distributions · Noise component

Mathematics Subject Classification (2000) 62H30 · 62F35 · 62F10 · 62F12

P. Coretto (✉)
Dipartimento di Scienze Economiche e Statistiche,
Università degli Studi di Salerno, Fisciano, Italy
e-mail: pcoretto@unisa.it

C. Hennig
Department of Statistical Sciences, University College London, London, UK
e-mail: chrish@stats.ucl.ac.uk

1 Introduction

This paper compares several methods for robust clustering based on mixture models. The term “model-based cluster analysis” was coined by Banfield and Raftery (1993) for clustering based on finite mixtures of Gaussian distributions and related methods. The standard Gaussian mixture model is to assume that data $(X_1, \ldots, X_n)$ are modelled as drawn i.i.d. from a distribution with density

$$f(x; \theta) = \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \sigma_j^2), \tag{1}$$

where $\phi(\cdot, \mu, \sigma^2)$ is the density of a Gaussian distribution with mean μ and variance σ^2, π_j is the proportion of the jth mixture component, $\sum_{j=1}^{G} \pi_j = 1$. The parameter vector θ contains all proportions, means and variances. Analogous notations will be used later for other models as well, the proportions sometimes including a π_0.

The maximum likelihood estimator (MLE) of the parameter θ from a dataset $\underline{x}_n = \{x_1, x_2, \ldots, x_n\}$ is obtained by

$$\hat{\theta}_n = \arg \max_{\theta \in \Theta} \sum_{i=1}^{n} \log f(x_i; \theta), \tag{2}$$

where $\Theta = \{\theta \mid \sigma_j^2 \geq s,\ j = 1, 2, \ldots, G,\ \sum_{j=1}^{G} \pi_j = 1\}$ for some choice of $s > 0$. It is necessary to bound variances from below in order to avoid degeneracies of the log-likelihood function. The MLE is usually computed by the EM algorithm, which means that only local maxima of the likelihood function are available. For more background and details on the EM algorithm, see Redner and Walker (1984), McLachlan and Krishnan (1997), and McLachlan and Peel (2000).

Based on $\hat{\theta}_n$ (the included parameters are denoted by giving them hats, too), observations x_i can be classified to component

$$k_i = \arg \max_{p=1,\ldots,G} \hat{\tau}_{ip}, \tag{3}$$

where the quantity $\hat{\tau}_{ip}$ is the estimated posterior probability that x_i was generated by the pth mixture component:

$$\hat{\tau}_{ip} = \frac{\hat{\pi}_p \phi(x_i; \hat{\mu}_p, \hat{\sigma}_p^2)}{f(x_i; \hat{\theta})}. \tag{4}$$

A sensible philosophy for model-based clustering is that (1) is not necessarily assumed to be “true”, but rather that the Gaussian distribution is treated as a cluster shape prototype, given that many distributions can be approximated closely by a Gaussian mixture. However, for a wide class of finite mixtures, including Gaussians, MLEs are not robust (Hennig 2004). This implies that deviations from the nominal model such as a small proportion of outliers in the data can lead to poor estimates and clustering. Note, however, that the concept of "high breakdown" is generally more problematic in cluster analysis than in traditional robust statistics, because it cannot be taken for granted that, adding some outliers, it is not desired to use a mixture component to fit them, which would yield breakdown of parameters.

Here we present a simulation study that compares the ML-estimator for Gaussian mixtures with several alternatives that have been proposed in order to deal better with outliers. Considerations are confined to one-dimensional data and a fixed number of clusters G. Apart from the practical relevance of fitting one-dimensional data, our aim is to contribute to the deeper understanding of a simple situation in order to also contribute to the understanding of more complex setups. In robust cluster analysis the comparison of the quality of different methods depends strongly on features such as separation of clusters and number and location of outliers, and even for one-dimensional data and fixed G one could imagine many more setups of interest apart from those that we consider in the present study.

A way to deal with the outlier problem is to add a "noise component" to the mixture, i.e., an additional mixture component to capture points like outliers that are not consistent with the Gaussian mixture. This was originally proposed by Banfield and Raftery (1993), who propose to add a uniform mixture component on the convex hull of the data (the range in one-dimensional situations). Variations of this idea have been proposed by Coretto (2008), who added uniform mixture components estimated by ML to the mixture, and the proposal of Hennig (2004), where the noise component is represented by a fixed constant on the real line. He showed that the resulting estimates are robust to even extreme outliers, as opposed to the noise component method of Banfield and Raftery (1993). Coretto (2008) studied the asymptotics of this approach and its computation based on the EM algorithm [note that further journal publications of the material in Coretto (2008) are in preparation by the authors of the present paper, and that an overview of the basic ideas was already given in Hennig and Coretto (2008)].

As an alternative approach, fitting mixtures of t-distributions (McLachlan and Peel 2000) is included in our simulation study.

In Sect. 2, the compared methods are defined. Section 3 explains in more details how they are computed in the simulation study. Section 4 defines the setups from which data are generated. Section 5 presents and discusses the results of the simulation study. Section 6 concludes the paper.

Alternative methods for robust clustering exist, particularly high breakdown methods based on a fixed partition model and trimming of observations (Cuesta-Albertos et al. 1997; García-Escudero et al. 2008; Gallegos and Ritter 2005). See Neykov et al. (2007) for a recent approach based on mixtures.

2 Methods and estimators

In this section we describe the estimators and methods under comparison. For computational details of the estimators proposed in this section, see Sect. 3.

2.1 MLE for Gaussian mixtures

In order to assess whether things can be improved by the robust alternatives, the ML estimator for Gaussian mixtures as in (1), computed by the EM-algorithm, is included in the simulation study.

2.2 Gaussian mixtures with uniform noise on the data-range

The method suggested in Banfield and Raftery (1993) and Fraley and Raftery (1998) consists of modelling the population by the following density function (notation as above):

$$g(x;\zeta)=\frac{\pi_0\mathbf{1}\{x\in[\min(\underline{x}_n),\max(\underline{x}_n)]\}}{\max(\underline{x}_n)-\min(\underline{x}_n)}+\sum_{j=1}^{G}\pi_j\phi(x;\mu_j,\sigma_j^2). \tag{5}$$

The vector of all parameters is called $\zeta\in\Psi$, $\Psi=\{\zeta\mid\sigma_j^2\geq s>0,\ \ j=1,2,\ldots,G,$ $\sum_{j=0}^{G}\pi_j=1\}$ for some fixed s. $\hat{\zeta}_n$ is defined as maximizer of the log-likelihood function associated to g. Note that this does not define a proper maximum likelihood estimator (MLE) because the model specification is data dependent. Moreover, $n\to\infty\Rightarrow[\max(\underline{x}_n)-\min(\underline{x}_n)]^{-1}\to 0$ a.s. Banfield and Raftery (1993) and Fraley and Raftery (1998) suggested to compute $\hat{\zeta}_n$ by applying the EM algorithm (the range held fixed).

The estimated posterior probability $\hat{\tau}_{ip}$ that the observation x_i was generated by component p now looks like this:

$$\hat{\tau}_{ip}=\begin{cases}\frac{\hat{\pi}_p\phi(x_i;\hat{\mu}_p,\hat{\sigma}_p^2)}{g(x_i;\hat{\zeta})} & \text{if } p=1,2,\ldots,G\\ \frac{\hat{\pi}_0}{[\max(\underline{x}_n)-\min(\underline{x}_n)]g(x_i;\hat{\zeta})} & \text{if } p=0.\end{cases}. \tag{6}$$

Hennig (2004) argued that this method is not robust against very extreme outliers. As opposed to low breakdown methods in traditional setups such as location estimation, it may however remain unaffected by less extreme outliers, and it may have good satisfactory behavior in many practical situations. We call this the range- or R-method.

2.3 MLE for Gaussian mixtures with uniform noise

In Coretto (2008), an ML-estimator for the following model is considered:

$$u(x;\gamma)=\frac{\pi_0\ \mathbf{1}\{x\in[a,b]\}}{b-a}+\sum_{j=1}^{G}\pi_j\phi(x;\mu_j,\sigma_j^2). \tag{7}$$

where $\mathbf{1}(A)$ is the indicator function of the set A. The parameter vector is called γ,

$$\hat{\gamma}_n = \arg\max_{\gamma \in \Gamma} \sum_{i=1}^{n} \log u(x_i; \gamma), \tag{8}$$

with $\Gamma = \{\gamma \mid \min_{m,p} v_m/v_p \geq h > 0;\ m, p = 0, 1, 2, \ldots G;\ \sum_{j=0}^{G} \pi_j = 1\}$, where $v_0 = (b - a)/\sqrt{12}$ and $v_p = \sigma_p$ for $p > 0$. The restricted set Γ allows to obtain an MLE that exists and is scale-equivariant (see Hathaway 1985), though the maximization problem in (8) is rather difficult: the log-likelihood function has infinitely many points of discontinuity, the restricted parameter set is not compact, moreover Γ cannot be described as a set of smooth inequalities, hence standard optimization theory does not apply. Coretto (2008) showed that the MLE exists and is strongly consistent for the set of maximizers of the expected log-likelihood function, and he also developed the EM algorithm for computing (8) and showed convergence. In the simulation study, for computational reasons, we stick to the simpler optimization on $\Gamma_0 = \{\gamma \mid v_j^2 \geq s,\ \ j = 0, 1, 2, \ldots G; \}$ for fixed $s > 0$.

The estimated posterior probability $\hat{\tau}_{ip}$ to be maximized for classification of x_i becomes

$$\hat{\tau}_{ip} = \begin{cases} \frac{\hat{\pi}_0 \mathbf{1}(x_i \in [\hat{a}, \hat{b}])}{(\hat{b} - \hat{a}) u(x_i; \hat{\gamma}_n)} & \text{if } p = 0 \\ \frac{\hat{\pi}_p \phi(x_i; \hat{\mu}_p, \hat{\sigma}_p^2)}{u(x_i; \hat{\gamma}_n)} & \text{if } p = 1, 2, \ldots, G \end{cases}. \tag{9}$$

Coretto (2008) showed that the EM algorithm generates local maxima of the log-likelihood for (a, b) chosen as any pair of data points as long as the restrictions above are fulfilled, so that the EM-algorithm is not very informative about the parameters of the uniform. One possible solution is to run the EM algorithm several times, and each time the uniform parameters are initialized by a suitable pair of distinct data-points. Among all solutions the MLE will be chosen so that the likelihood value is the largest. We confine the search to the best local maximum over a selected grid of distinct data-points. By this the quality of approximation of the MLE is affected but the computational complexity can be controlled. We call this the grid- or G-method. Note that the breakdown point for this approach, according to the definition of Hennig (2004), can at best be $\frac{2}{n+2}$ (compared to $\frac{1}{n+1}$ for the R-method), because two added outliers on both sides converging to infinity in absolute value emulate the robustness problem that the R-method has with a single point, namely that the log-likelihood can only be prevented from converging to $-\infty$ by fitting one of the outliers by a Gaussian component.

2.4 Robust improper maximum likelihood estimator

The idea of the robust improper maximum likelihood estimator (RIMLE) is to choose a fixed constant over the whole real line for the noise, instead of modelling it by a proper uniform distribution. Hennig (2004) showed that this has a better breakdown behavior than the two previous approaches. The approach is based on the following improper density:

$$\lambda_c(x;\eta) = \pi_0 c + \sum_{j=1}^{G} \pi_j \phi(x;\mu_j,\sigma_j^2). \tag{10}$$

$c \geq 0$ is a constant that (in order to apply the theory in Hennig 2004) needs to be specified in advance. The idea is that points are classified as noise if they arise from areas where the Gaussian components account for density values smaller than c. The RIMLE is defined as:

$$\hat{\eta}_n(c) = \arg\max_{\eta\in\Lambda} \sum_{i=1}^{n} \log(\lambda_c(x_i;\eta)). \tag{11}$$

where $\Lambda = \{\eta|\sigma_j \geq s > 0,\ \ j = 1,2,\ldots,G;\ \sum_{j=0}^{G}\pi_j = 1\}$. The constrained set Λ ensures existence but does not guarantee scale-equivariance. The EM algorithm can be applied to compute the RIMLE because for a fixed dataset (10) can be written down as a proper density with value c on a set of Lebesgue-measure $\frac{1}{c}$ containing the observed data (the RIMLE is not a proper ML estimator for such a model, though). Asymptotic analysis of the method is given in Coretto (2008). Points can be classified by maximizing

$$\hat{\tau}_{ip} = \frac{\hat{\pi}_0 c}{\lambda_c(x_i;\hat{\eta}_n(c))} \quad \text{for} \quad p=0 \quad \text{and}$$
$$\hat{\tau}_{ip} = \frac{\hat{\pi}_p \phi(x_i;\mu_p,\sigma_p^2)}{\lambda_c(x_i;\hat{\eta}_n(c))} \quad \text{for} \quad p=1,2,\ldots,G, \tag{12}$$

where $\hat{\tau}_{i0}$ is the "improper" posterior probability that the observation x_i belongs to the noise component based on the proper, but data dependent model mentioned below (11).

The key issue here is how to fix c. Hennig (2005) gives some orientation about a data-independent (subject matter based) choice of c. However Coretto (2008) provided empirical evidence that in most situations a bad choice of c can seriously affect the performance of the RIMLE. The author suggested a data dependent choice of c called "filtering method":

Step 1 Start from a set of values $c' \in [0,\bar{c}]$ and for each of them compute the RIMLE $\hat{\eta}_{c'}$.

Step 2 For each $\hat{\eta}_{c'}$ remove those observations classified as noise and thus obtain the "filtered dataset".

Step 3 Use the filtered dataset to compute the MLE of a distribution function of a mixture (1).

Step 4 Compute the empirical distribution function over the filtered dataset and use the Kolmogorov distance to compare it to the distribution function of the Gaussian mixture estimated in the previous step.

Step 5 Choose the value of c minimizing the Kolmogorov distance.

The method depends on the choice of $\bar{c}$, but this can be easily fixed. For large values of c the RIMLE will end up classifying all points as noise, which is not desired. Hence the value $\bar{c}$ can be fixed in terms of the maximum proportion of data points we are willing to accept as noise points. We chose 50% here.

2.5 MLE for *t*-mixtures

McLachlan and Peel (2000) argue that for elliptically shaped clusters with heavier than Gaussian tails or atypical observations, the use of Gaussian components may affect the fit of the data. The strategy proposed by the authors is as follows: if the population is assumed to contain G Gaussian clusters plus noise, we model it as arising from a finite mixture of t-distributions with G components. Noise is identified with the set of data-points that are far away from the cluster centers. The meaning of "far away" will be explained later in this section. Consider a finite mixture of univariate t-distributions with G components having density

$$t(x;\xi) = \sum_{j=1}^{G} \pi_j \psi(x;\mu_j,\sigma_j,u_j), \tag{13}$$

where $\psi(x;\mu_i,\sigma_i,u_i)$ is the density of a non-central t-distribution with location μ_i, scale σ_i and degrees of freedom u_i. Note that the scale parameter is related to the variance v_j^2 of the distribution by

$$v_j^2 = \frac{u_j}{u_j - 2}\sigma_j^2. \tag{14}$$

The parameter vector is denoted by ξ (the degrees of freedom could be included or fixed). McLachlan and Peel (2000) proposed to estimate ξ by ML. The estimator can be obtained by the EM algorithm. Classification can be carried out maximizing

$$\hat{\tau}_{ip} = \frac{\hat{\pi}_p \psi(x_i;\hat{\mu}_p,\hat{\sigma}_p,\hat{u}_p)}{t(x_i;\hat{\xi}_n)}, \qquad j = 1,2,\ldots,G \tag{15}$$

over $\Xi = \{\xi \mid v_j^2 \geq s,\ \ j = 1,2,\ldots,G,\ \sum_{j=1}^{G}\pi_j = 0\}$ for fixed $s > 0$. Define the squared Mahalanobis distance between the point x and μ

$$\delta(x;\mu,\sigma) = \frac{(x-\mu)^2}{\sigma^2}. \tag{16}$$

We also define the sequence:

$$C_i = \sum_{j=1}^{G} \mathbf{1}\left\{ \underset{j=1,2,\ldots,G}{\arg\max}\ \hat{\tau}_{ij} = j \right\} \delta(x_i;\hat{\mu}_j,\hat{\sigma}_j); \qquad i = 1,2,\ldots,n. \tag{17}$$

McLachlan and Peel (2000) argued (without proof) that if an i.i.d. sample is generated from a Gaussian mixtures with G components, then $\{C_i;\ i = 1, 2, \ldots, n\}$ follows (at least approximately) a χ_1^2 distribution. They consider the observation x_i as "noise" if C_i exceeds the 0.95-quantiles of χ_1^2. Therefore, x_i is assigned to the k_ith cluster (the Gaussian cluster prototype is mimicked by the set of non-noise points of a t-distribution here), if $k_i = \arg\max_{j=1,2,\ldots,G} \hat{\tau}_{ij} \mathbf{1}\{C_i < 3.841459\}$. $k_i = 0$ (assignment to no component) means that the points is assigned to the noise.

In terms of the breakdown behavior, this approach is very similar to the R-method, see Hennig (2004).

3 Computational details

In the simulation study we considered sample sizes of $n = 50, 200, 500$. For each data generating process and for each sample size we drew 500 samples, and for each replica we applied the estimation methods described before. For each estimator we performed clustering and computed summary statistics to evaluate relative performance (see Sect. 5.1). Here we describe some computational aspects of these methodologies.

The EM algorithm depends on the initialization of the parameter values. For all methods involving a π_0, its initial value was fixed at 0.05, following the idea that if data are partitioned into several clusters, the remaining noise proportion will normally be quite small (this is not necessarily the case in practice, but many practitioners would like to have as many points as possible assigned to clusters). For all methods, the proportions of the other components are always initialized at equal value. The means and variances of Gaussian components are initialized by trimming the 10% of observations in both the tails of the data and then applying the k-means algorithm with G components with randomly chosen initial values. This is related, but not identical to the trimmed-k-means method (Cuesta-Albertos et al. 1997), which is usually computed by a more computer-intensive algorithm. In general, good initialization of the EM-algorithm for a Gaussian mixture alone is a complicated issue, and alternatives to our approach exist, see for example Karlis and Xekalaki (2003) and the hierarchical approach in the software package MCLUST (Fraley and Raftery 2006). For the t-mixtures we do the same but we use both the variance from k-means after trimming and degrees of freedom to get the initial values for scales, see (14). Each EM run stops either when the likelihood value has not improved by more than 10^{-6} or when the number of iterations exceeds 600. The latter did not happen more than once (out of 500 replicas) in any simulation setup, in which case the corresponding result was discarded.

The lower variance bound was chosen as $s = 0.1$ for all methods. Note that as a side-effect of the described initialization method variances below 0.1 never occurred in any EM run (though of course in practice the order of magnitude of the observations would need to be taken into account choosing s).

Gaussian mixtures with uniform noise (*R, G-method*) In Sect. 2 we introduced two methods where the noise is represented by uniform densities. The R-method was initialised as previously explained.

The ML estimate for (7) was approximated using the EM algorithm with the uniform parameter (a, b) initialized over a selected grid of data points (G-method). The proportion parameters, means and variances were initialized as before. Given the sample size, we defined a preliminary grid of equi-spaced points on the range of the data. For computational reasons, the size of the preliminary grid decreases as the sample size increases. For $n = 50$ the grid consists of 20 points, for $n = 200$ the grid consists of 15 points, and for $n = 500$ it consists of 10 points. We then defined the initialization grid by using the nearest data point for each point in the preliminary grid. Out of two points in the initialization grid with distance of less than 1% of the interquartile range, one was removed (in order to prevent problems with the variance restrictions in Sect. 2.3).

RIMLE (*If, I-method*) The RIMLE defined in Sect. 2 was computed by selecting the improper constant density via the filtering method. This approach is referred to as "If-method".

In order to illustrate how good the method could be with near optimal selection of c (and how close the quality of the If-method is to that), we added a method called "I-method" that makes use of (unrealistic) knowledge of the data generation process. In the I-method the value of c was fixed as the value that we found for a given data-generating process (over all replicas and all sample sizes) that achieved the lowest average misclassification percentage. All other initializations were as previously described.

t-mixture (*Te, Tf-method*) We considered the methodology based on t-mixtures both in the case where degrees of freedom are estimated ("Te-method"), and the case when the degrees of freedom are fixed ("Tf-method"). The computation of the MLE for the Te-method was done using the EM/ECM algorithm studied in Liu (1997). During our experiments we noted that the EM/ECM algorithm does not move too far from the starting values with respect to degrees of freedom when these are estimated. This is possibly due to the fact that the log-likelihood surface has many local maxima or has flat regions with respect to the degrees of freedom. We could not find any research on this topic. A good practice (though not followed in the previous study for computational reasons) would be to run the ECM several times, with many possible combinations of initial values for the degrees of freedom, and then select the solution that corresponds to the highest log-likelihood value.

When the degrees of freedom were estimated, they were initialized to be equal to 15 for each component (half way between Gaussian tails and heavier tails, given that from 30 degrees of freedom upwards the t-distribution can be considered to be approximately Gaussian). For the t-mixture with fixed degrees of freedom we fixed them at 3 for all components (allowing for heavy tails but with existing variances).

See Sect. 4.3 for the proportions, means and variances estimated by the Te- and Tf-method.

Furthermore, we computed the *MLE for plain Gaussian mixtures* (*N-method*) as explained already.

4 Data generating processes

We considered six different data generating processes. Throughout the rest of this chapter, $\mathrm{N}(\mu, v)$ is the Gaussian distribution with mean μ and variance v; $U(a, b)$ is the uniform distribution with support on the interval $[a, b]$. For $g > 2$, $T_g(\mu, v)$ is the non-central t-distribution with g degrees of freedom, location parameter μ and variance v. Note that we here parameterized the t-distribution in terms of variance, assuming that $g > 2$, see (14). We tried to cover a range of essentially different archetypical situations that are not obviously unrealistic. Note particularly that, whereas all setups fulfil the model assumptions of at least one method (and all methods apart from the RIMLE have their model assumptions fulfilled in at least one setup), most methods

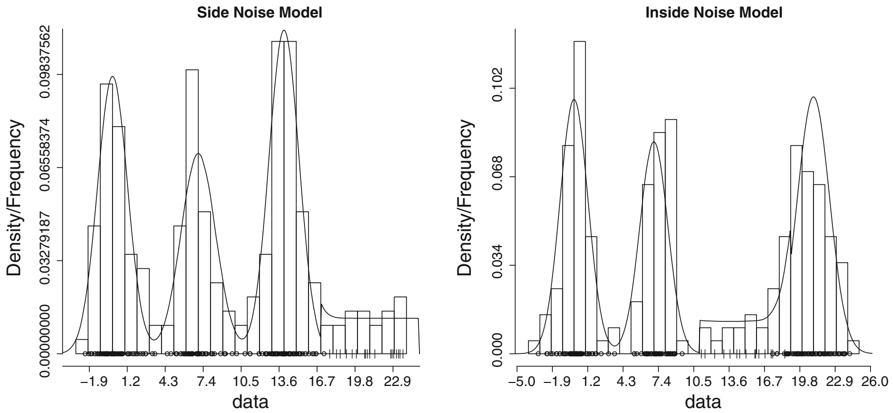

Fig. 1 *Right side* histogram for a sample of 200 points drawn from the side-noise model (18) with density function. *Circles* on the bottom represent the non-noise points in data set, *strokes* represent noise points. *Left side* same for a sample of 200 points drawn from the inside-noise model (19)

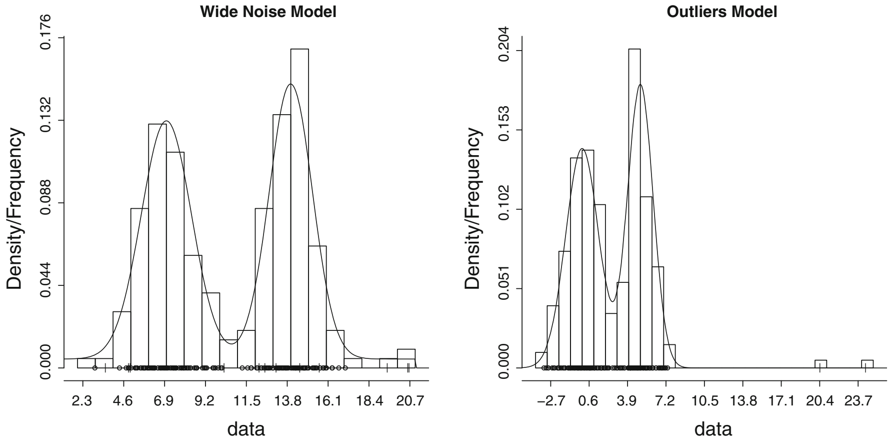

Fig. 2 *Left side* sample of 200 points from wide-noise model (20). *Right side* same for outlier process, 198 points from (21) and 2 outliers

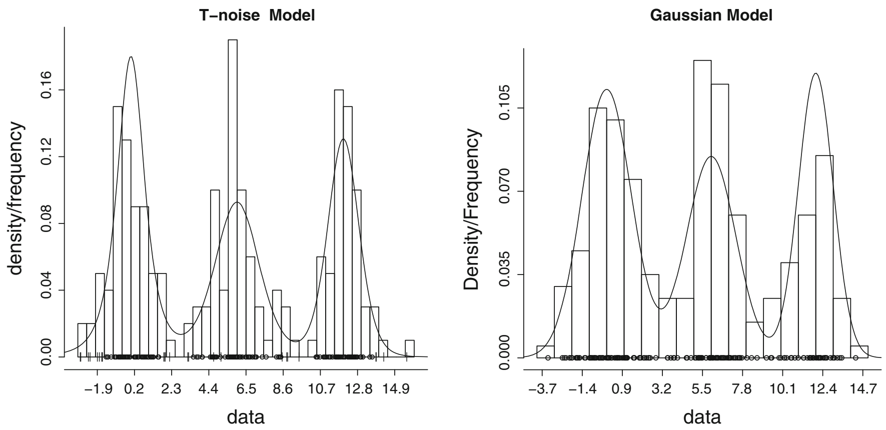

Fig. 3 *Left side* sample of 200 points from t-noise model (22). Again, *circles* on the bottom represent the non-noise points in data set, *strokes* represent noise points. *Right side* sample of 200 points from Gaussian mixture model (23)

are in most setups confronted with data generating processes that violate their nominal assumptions. This is realistic and it also reflects the philosophy that we are interested in the method to construct reasonable Gaussian shaped clusters regardless of whether model assumptions are fulfilled.

4.1 Side, inside and wide uniform noise

Noise here is defined as points drawn from a uniform mixture component. We consider three alternatives, differing in terms of the position of the uniform support relative to the means of the Gaussians. We refer to these models as side-noise, inside-noise, and wide-noise.

Side-noise:

$$0.1\mathrm{U}(17, 25) + 0.30\mathrm{N}(0, 1.5) + 0.25\mathrm{N}(7, 2) + 0.35\mathrm{N}(14, 1.5). \tag{18}$$

We chose a noise proportion of 10% here in order to still have clusters that are clearly visible. Though this may not always be the case in reality, it is a minimum benchmark requirement whether cluster analysis methods can find the clusters in a situation with at least fairly clearly separated clusters. The noise produced in this model is located on the right of the mean of the largest Gaussian (see Fig. 1) Note that the Gaussian components are reasonably separated, they have relatively small variances, and their proportions do not deviate much from being equal. Hosmer (1978) showed that when the number of Gaussian components is larger than two and the separation between components is weak, the solution provided by the EM algorithm can be a poor approximation for the maximum likelihood estimate. In some experiments we noted also that this happens particularly when the variances in the underlying Gaussians are relatively large and the proportions deviate considerably from equality. This latter effect

Table 1 Average misclassification percentages for "side noise"

n	Method	Global	Component			
			Noise	2	3	4
50	G	20.72 (1.81)	13.12	0.60	4.58	2.41
	R	19.90 (1.79)	14.75	0.56	4.16	0.43
	I	13.43 (1.52)	3.31	0.85	5.90	3.37
	If	13.67 (1.54)	3.40	1.18	6.04	3.04
	Te	17.16 (1.69)	3.27	1.64	5.67	6.59
	Tf	17.57 (1.70)	6.70	1.31	5.12	4.44
	N	19.73 (1.78)	0.00	1.70	6.35	11.68
200	G	5.29 (1.00)	4.42	0.11	0.47	0.29
	R	11.00 (1.40)	10.17	0.01	0.78	0.04
	I	4.24 (0.90)	2.47	0.07	0.89	0.81
	If	4.30 (0.91)	2.50	0.07	0.90	0.82
	Te	8.78 (1.27)	3.25	0.03	0.70	4.81
	Tf	10.84 (1.39)	6.25	0.01	1.06	3.52
	N	13.89 (1.55)	0.00	0.17	0.79	12.93
500	G	2.51 (0.70)	2.08	0.11	0.19	0.12
	R	8.21 (1.23)	8.19	0.01	0.00	0.01
	I	2.39 (0.68)	0.89	0.08	0.09	1.33
	If	2.08 (0.64)	1.42	0.06	0.09	0.51
	Te	7.27 (1.16)	3.34	0.01	0.01	3.91
	Tf	9.66 (1.32)	6.32	0.01	0.03	3.30
	N	13.04 (1.51)	0.00	0.17	0.13	12.74

The components' average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

was documented by Karlis and Xekalaki (2003). However, these problems are not the focus of this research. Hence the choice of well separated Gaussian components with relatively small variances and not very dissimilar proportions.

Inside-noise:

$$0.1\mathrm{U}(11, 19) + 0.30\mathrm{N}(0, 1.5) + 0.25\mathrm{N}(7, 1.5) + 0.35\mathrm{N}(21, 2). \qquad (19)$$

The model is similar to the previous one with the exception that the uniform noise is now located in the region between the tails of two Gaussians (see Fig. 1).

Wide-noise:

$$0.1\mathrm{U}(0, 21) + 0.45\mathrm{N}(7, 2) + 0.45\mathrm{N}(14, 1.5). \qquad (20)$$

Here, the uniform noise spreads over the entire range of the data (see Fig. 2).

Table 2 Upper-trimmed means of distances for classes of parameters for “side noise”

n	Method	c	d_π	d_μ	d_v
50	G		0.36 (0.01)	2.22 (0.18)	2.54 (0.06)
	R		0.49 (0.01)	1.69 (0.13)	2.63 (0.05)
	I	0.020	0.26 (0.01)	1.68 (0.13)	3.20 (0.28)
	If	0.014 (0.001)	0.27 (0.01)	1.52 (0.15)	3.18 (0.26)
	Te		0.21 (0.01)	1.76 (0.18)	6.23 (0.34)
	Tf		0.20 (0.01)	1.59 (0.10)	5.42 (0.33)
	N		0.35 (0.01)	2.11 (0.18)	14.18 (0.37)
200	G		0.19 (0.01)	0.45 (0.01)	1.16 (0.03)
	R		0.41 (0.01)	0.46 (0.01)	1.76 (0.03)
	I	0.020	0.15 (0.00)	0.53 (0.02)	1.62 (0.16)
	If	0.020 (0.000)	0.18 (0.00)	0.45 (0.01)	1.17 (0.03)
	Te		0.12 (0.00)	0.65 (0.02)	3.12 (0.13)
	Tf		0.15 (0.00)	0.62 (0.01)	3.02 (0.09)
	N		0.32 (0.00)	1.16 (0.02)	14.43 (0.17)
500	G		0.10 (0.00)	0.27 (0.01)	0.68 (0.02)
	R		0.38 (0.00)	0.27 (0.01)	1.60 (0.01)
	I	0.020	0.12 (0.00)	0.30 (0.01)	0.84 (0.02)
	If	0.020 (0.000)	0.13 (0.00)	0.28 (0.01)	0.80 (0.02)
	Te		0.09 (0.00)	0.43 (0.01)	2.38 (0.07)
	Tf		0.14 (0.00)	0.43 (0.01)	2.59 (0.04)
	N		0.31 (0.00)	1.06 (0.01)	14.00 (0.10)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

4.2 Outlier process

This model consists of a two-Gaussian mixture plus two gross outliers drawn from a uniform distribution. In each replica of the simulation study for sample size equal to n we drew a sample of $n - 2$ points from the mixture model

$$0.5\mathrm{N}(0, 2) + 0.5\mathrm{N}(5, 1.2) \tag{21}$$

and then added two outliers from U(20, 25) (see Fig. 2). We call these points “outliers” because they are so far away from the Gaussian components that under a Gaussian mixture (21) such points are generated with a probability of virtually zero.

4.3 *t*-Noise

Here data were generated by a mixture of *t*-distributions, and points were declared to be “noise” if they are far in the tails of the *t*-distribution. This is necessary in order to make the distinction between “noise” and “cluster” compatible with the other setups, and it also reflects the idea that *t*-distributions are not used because they are thought

Table 3 Average misclassification percentages for "inside noise"

n	Method	Global	Component			
			Noise	2	3	4
50	G	21.04 (1.82)	15.92	1.04	2.14	1.94
	R	15.48 (1.62)	12.23	0.52	1.36	1.36
	I	10.81 (1.39)	3.27	0.74	3.40	3.40
	If	10.55 (1.37)	3.40	1.07	3.43	2.65
	Te	13.18 (1.51)	3.43	2.11	4.56	3.08
	Tf	14.07 (1.56)	6.81	1.85	3.05	2.36
	N	13.22 (1.51)	0.00	2.17	6.37	4.68
200	G	6.91 (1.13)	5.33	0.09	0.29	1.20
	R	9.05 (1.28)	7.66	0.00	0.03	1.36
	I	5.24 (1.00)	2.91	0.02	0.20	2.11
	If	5.65 (1.03)	3.14	0.02	0.21	2.27
	Te	8.41 (1.24)	3.20	0.04	2.18	2.99
	Tf	8.76 (1.26)	5.88	0.04	0.56	2.28
	N	10.96 (1.40)	0.00	0.06	5.43	5.47
500	G	5.27 (1.00)	3.87	0.06	0.14	1.20
	R	7.33 (1.17)	6.00	0.00	0.00	1.33
	I	4.62 (0.94)	1.08	0.02	0.06	3.47
	If	4.50 (0.93)	2.19	0.01	0.01	2.28
	Te	6.97 (1.14)	3.18	0.00	1.18	2.61
	Tf	7.78 (1.20)	5.53	0.00	0.10	2.15
	N	10.76 (1.39)	0.00	0.02	5.65	5.09

The components' average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

to model "true clusters", but rather because their ML-estimators can accommodate outliers. The setup is included in order to have "similar-to-Gaussian" but not precisely Gaussian clusters and "noise" that corresponds to the model assumptions of the Te-method (the fixed number of degrees of freedom assumed for the Tf-method is in this situation correct for one of the three mixture components):

$$0.4\mathrm{T}_3(0, 2) + 0.3\mathrm{T}_{10}(6, 2) + 0.3\mathrm{T}_{10}(12, 1). \tag{22}$$

"Noise" is defined by the method proposed by McLachlan and Peel (2000). x_{1,j_1}, $x_{2,j_2}, \ldots, x_{n,j_n}$ denotes the sample, where $j_i \in \{1, 2, \ldots, G\}$, and x_{i,j_i} is generated by the j_ith component. For each observation we compute $\delta_i = \delta(x_{i,j_i}; \mu_{j_i}, \sigma_{j_i})$. A point x_{i,j_i} will be defined as noise if $\delta_i \geq 3.841459$ (see Sect. 2.5, Fig. 3). In order to make the parameter estimators comparable over different methods, proportions were re-computed to add up to 1 (including a "noise component proportion" for the points classified as noise), and means and variances were re-computed for the non-noise components based on the non-noise points only. The same method was generally applied to the estimators from the Te- and Tf-method to calibrate it properly in order

Table 4 Upper-trimmed means of classes of parameters for "inside noise"

n	Method	c	d_π	d_μ	d_v
50	G		0.41 (0.01)	2.24 (0.14)	2.47 (0.07)
	R		0.42 (0.01)	1.44 (0.14)	2.35 (0.04)
	I	0.024	0.25 (0.01)	1.46 (0.10)	3.03 (0.20)
	If	0.016 (0.001)	0.25 (0.01)	1.30 (0.08)	3.05 (0.24)
	Te		0.20 (0.00)	1.56 (0.08)	5.24 (0.27)
	Tf		0.18 (0.00)	1.30 (0.06)	3.66 (0.20)
	N		0.29 (0.00)	1.80 (0.08)	8.11 (0.31)
200	G		0.22 (0.00)	0.48 (0.01)	1.15 (0.03)
	R		0.34 (0.00)	0.50 (0.01)	1.43 (0.02)
	I	0.024	0.15 (0.00)	0.72 (0.02)	2.05 (0.12)
	If	0.026 (0.000)	0.18 (0.00)	0.54 (0.01)	1.28 (0.03)
	Te		0.11 (0.00)	0.75 (0.02)	2.55 (0.14)
	Tf		0.12 (0.00)	0.57 (0.01)	1.55 (0.04)
	N		0.24 (0.00)	1.38 (0.02)	8.50 (0.20)
500	G		0.18 (0.00)	0.33 (0.01)	0.88 (0.02)
	R		0.31 (0.00)	0.33 (0.01)	1.20 (0.01)
	I	0.024	0.11 (0.00)	0.45 (0.01)	1.09 (0.03)
	If	0.026 (0.000)	0.14 (0.00)	0.39 (0.01)	0.99 (0.02)
	Te		0.07 (0.00)	0.49 (0.01)	1.53 (0.08)
	Tf		0.10 (0.00)	0.39 (0.01)	1.24 (0.02)
	N		0.24 (0.00)	1.30 (0.01)	9.10 (0.17)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

to be comparable with the other methods. This model produces noise with an average proportion of about 10.7%.

4.4 Gaussian mixture/noiseless data

We also added a process fulfilling the basic Gaussian mixture assumption (1) with moderately well separated clusters (see Fig. 3):

$$0.4\mathrm{N}(0, 2) + 0.3\mathrm{N}(6, 2) + 0.3\mathrm{N}(12, 1). \tag{23}$$

No points were treated as "true noise" here.

5 Evaluation of the simulation study

5.1 Measures of performance

As a measure of clustering performance we computed the misclassification percentages. Because in all setups the non-noise component means are well separated, we

Table 5 Average misclassification percentages for “wide noise”

n	Method	Global	Component		
			Noise	2	3
50	G	31.89 (2.08)	25.35	3.78	2.76
	R	15.04 (1.60)	10.47	2.42	2.15
	I	8.28 (1.23)	0.70	4.19	3.39
	If	9.54 (1.31)	2.34	3.98	3.21
	Te	9.70 (1.32)	3.77	3.33	2.61
	Tf	12.05 (1.46)	7.05	2.75	2.26
	N	10.44 (1.37)	0.00	5.41	5.02
200	G	8.96 (1.28)	2.54	3.44	2.98
	R	7.99 (1.21)	1.72	3.35	2.93
	I	7.70 (1.19)	0.44	3.88	3.38
	If	8.29 (1.23)	1.25	3.79	3.26
	Te	9.28 (1.30)	3.96	2.89	2.43
	Tf	11.02 (1.40)	6.29	2.54	2.19
	N	10.44 (1.37)	0.00	5.21	5.23
500	G	7.50 (1.18)	0.94	3.43	3.13
	R	7.49 (1.18)	0.94	3.43	3.13
	I	7.49 (1.18)	0.33	3.76	3.40
	If	7.74 (1.20)	0.65	3.72	3.37
	Te	9.12 (1.29)	3.98	2.75	2.39
	Tf	10.81 (1.39)	6.19	2.45	2.17
	N	10.35 (1.36)	0.00	5.14	5.21

The components’ average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

used lexicographical ordering of component means to match estimated with true components. Misclassification percentages were averaged over all 500 replicas and are given (component-wise and overall) in the Tables 1, 3, 5, 7, 9, and 11.

Recall that in most of the situations under consideration the data generating process does not coincide with the estimated model. Measurement of performance cannot fairly be made by comparing parameter estimates with the truth. What we do instead is to compare the moments and proportions of what was defined as “true” (non-noise) clusters with their estimated counterparts. We based our evaluation on the L_1 distance for vectors of classes of estimates. Let us assume that the data generating process consists of G components plus the noise component. Let $\pi_0^0, \pi_1^0, \ldots, \pi_G^0$ be the true proportion parameters, $\mu_1^0, \ldots, \mu_G^0$ the true means and $v_1^0, \ldots, v_G^0$ the true variances of the non-noise components. Suppose, for instance, that an estimation method A produces the following estimates $\pi_0^A, \pi_1^A, \ldots, \pi_G^A, \mu_1^A, \ldots, \mu_G^A$ and $v_1^A, \ldots, v_G^A$. For each replica we considered the L_1 distances of the three different classes of vectors: proportions, means and variances. That is, for each replica we computed

Table 6 Upper-trimmed means of classes of parameters for “wide noise”

n	Method	c	d_π	d_μ	d_v
50	G		0.49 (0.01)	1.41 (0.09)	1.97 (0.06)
	R		0.33 (0.01)	0.49 (0.01)	1.32 (0.03)
	I	0.043	0.17 (0.00)	0.45 (0.01)	1.06 (0.03)
	If	0.025 (0.001)	0.20 (0.00)	0.47 (0.01)	1.18 (0.03)
	Te		0.13 (0.00)	0.46 (0.01)	1.06 (0.03)
	Tf		0.13 (0.00)	0.48 (0.01)	1.22 (0.02)
	N		0.23 (0.00)	0.50 (0.01)	2.06 (0.07)
200	G		0.11 (0.00)	0.23 (0.01)	0.55 (0.02)
	R		0.10 (0.00)	0.22 (0.01)	0.51 (0.01)
	I	0.043	0.09 (0.00)	0.22 (0.01)	0.53 (0.01)
	If	0.035 (0.001)	0.12 (0.00)	0.22 (0.01)	0.65 (0.02)
	Te		0.07 (0.00)	0.23 (0.01)	0.69 (0.01)
	Tf		0.07 (0.00)	0.23 (0.01)	0.92 (0.02)
	N		0.20 (0.00)	0.34 (0.01)	2.04 (0.04)
500	G		0.06 (0.00)	0.14 (0.00)	0.32 (0.01)
	R		0.06 (0.00)	0.14 (0.00)	0.32 (0.01)
	I	0.043	0.06 (0.00)	0.14 (0.00)	0.33 (0.01)
	If	0.037 (0.001)	0.08 (0.00)	0.15 (0.00)	0.40 (0.01)
	Te		0.05 (0.00)	0.15 (0.00)	0.63 (0.01)
	Tf		0.05 (0.00)	0.15 (0.00)	0.92 (0.01)
	N		0.20 (0.00)	0.31 (0.01)	2.04 (0.02)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

$$d_\pi = \sum_{j=0}^{G} |\pi_j^0 - \pi_j^A|, \qquad d_\mu = \sum_{j=1}^{G} |\mu_j^0 - \mu_j^A|, \qquad d_v = \sum_{j=1}^{G} |v_j^0 - v_j^A|.$$

Note that d_μ and d_v do not contain means and variances for the noise component. For each setup and sample size we give 90%-upper-trimmed means (over the 500 replica) as quality measurements in the Tables 2, 4, 6, 8, 10, and 12. This is because for some sample the EM algorithm solution can be strongly dependent on the initial values, and in some situations this could cause anomalous values of the L_1 distances, which should not affect the quality measurement too much (following the idea that if a method gets an estimator grossly wrong, it is not important how wrong it exactly is). Standard errors for 90%-upper-trimmed means were estimated by nonparametric bootstrap.

Complementary to 90%-upper-trimmed mean, Table 13 gives an idea about how often a method produced grossly outlying parameter estimators, as suggested by a referee. It is subtle to define “gross outliers” in such a situation. What we did was to aggregate, for a given n and data generating process, all values of d_μ (distance from truth of mean estimators). All estimators in the upper 1% were interpreted as

Table 7 Average misclassification percentages for the "outlier process"

n	Method	Global	Component		
			Noise	2	3
50	G	31.10 (2.07)	24.50	4.06	2.53
	R	3.63 (0.84)	0.42	1.53	1.68
	I	3.33 (0.80)	0.08	1.62	1.62
	If	5.13 (0.99)	2.13	1.44	1.56
	Te	6.98 (1.14)	3.60	0.42	2.96
	Tf	9.50 (1.31)	5.44	1.73	2.34
	N	25.67 (1.95)	0.00	10.76	14.90
200	G	3.39 (0.81)	0.88	1.05	1.45
	R	2.61 (0.71)	0.10	1.03	1.47
	I	2.52 (0.70)	0.02	1.03	1.46
	If	3.22 (0.79)	0.73	1.02	1.47
	Te	7.23 (1.16)	5.63	0.64	0.97
	Tf	9.46 (1.31)	8.12	0.50	0.84
	N	22.05 (1.85)	0.00	0.10	21.95
500	G	2.43 (0.69)	0.11	1.03	1.29
	R	2.35 (0.68)	0.03	1.02	1.30
	I	2.33 (0.67)	0.01	1.03	1.30
	If	2.68 (0.72)	0.36	1.02	1.30
	Te	6.92 (1.14)	5.35	0.76	0.81
	Tf	9.88 (1.33)	8.83	0.41	0.64
	N	14.35 (1.57)	0.00	0.01	14.34

The components' average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

gross outliers, and Table 13 gives the percentage for this to happen for each method. The same principle was applied to variance estimators (results not shown; they were generally similar but looked a bit more favourable for the G- and R-method).

5.2 Results

Side noise (Tables 1, 2). The RIMLE method (I and If) clearly performed best, at least with respect to classification. Even the If-method performed better than the G-method, of which the model assumptions were fulfilled here. The G-method was bad for small n. The R-method assigned too many points to the noise component, but its location estimators were acceptable. As in many other setups, Te, Tf and N were clearly worse than the competitors.

Inside noise (Tables 3, 4). The results were generally similar to those for "side noise". However, the G- and R-method were better than If with respect to mean estimation with

Table 8 Upper-trimmed means of classes of parameters for the “outlier process”

n	Method	c	d_π	d_μ	d_v
50	G		0.38 (0.02)	6.95 (0.46)	1.24 (0.05)
	R		0.09 (0.00)	0.46 (0.01)	0.91 (0.02)
	I	0.015	0.06 (0.00)	0.46 (0.01)	0.88 (0.02)
	If	0.019 (0.002)	0.12 (0.01)	0.47 (0.01)	0.97 (0.02)
	Te		0.11 (0.00)	0.49 (0.01)	1.17 (0.02)
	Tf		0.16 (0.00)	0.53 (0.02)	1.38 (0.03)
	N		0.56 (0.01)	3.35 (0.41)	19.92 (0.41)
200	G		0.03 (0.00)	0.22 (0.01)	0.43 (0.01)
	R		0.04 (0.00)	0.22 (0.01)	0.43 (0.01)
	I	0.015	0.03 (0.00)	0.22 (0.01)	0.42 (0.01)
	If	0.025 (0.002)	0.06 (0.00)	0.22 (0.01)	0.48 (0.01)
	Te		0.11 (0.00)	0.24 (0.01)	0.87 (0.01)
	Tf		0.16 (0.00)	0.24 (0.01)	1.08 (0.01)
	N		0.53 (0.00)	1.71 (0.01)	10.95 (0.04)
500	G		0.02 (0.00)	0.14 (0.00)	0.27 (0.01)
	R		0.02 (0.00)	0.14 (0.00)	0.27 (0.01)
	I	0.015	0.02 (0.00)	0.14 (0.00)	0.27 (0.01)
	If	0.027 (0.002)	0.04 (0.00)	0.14 (0.00)	0.31 (0.01)
	Te		0.10 (0.00)	0.15 (0.00)	0.83 (0.01)
	Tf		0.17 (0.00)	0.16 (0.00)	1.14 (0.01)
	N		0.36 (0.00)	1.45 (0.02)	6.21 (0.05)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

not too small n (but worse with respect to classification), and the Te- and Tf-method were clearly better than the G- and R-method for small n.

Wide noise (Tables 5, 6). This setup did not only fulfil the model assumptions of the G-method, but also corresponded to the R-method in the sense that with high probability the two extreme points in the dataset were generated from the uniform distribution. Keeping this in mind, it is remarkable that the If-method was still better than the R-method for $n = 50$ and not much worse for larger n. The G-methods again required a high n to work well, and the Te-, Tf- and N-methods fell again clearly behind (though the t-mixture methods worked reasonably well to estimate the cluster locations and very well to estimate the proportion).

Outlier process (Tables 7, 8). The If-method was clearly worse than the optimal I-method here, so that this is the only setup in which the R-method worked consistently better than the If-method. The G-method was only good for $n = 500$ here. Comments as before apply for the Te-, Tf and N-method, though the Te-method was acceptable for parameter estimation. Both t-mixture methods classified far too many points as outliers, though.

Table 9 Average misclassification percentages for "*t*-noise"

n	Method	Global	Component			
			Noise	2	3	4
50	G	25.21 (1.94)	16.70	3.18	3.21	2.12
	R	12.81 (1.49)	8.50	1.67	1.89	0.75
	I	11.00 (1.40)	4.12	2.63	2.94	1.30
	If	10.55 (1.37)	1.36	3.83	3.31	2.05
	Te	8.52 (1.25)	1.91	2.51	2.68	1.43
	Tf	9.24 (1.30)	4.66	1.44	2.13	1.01
	N	12.24 (1.47)	0.00	5.48	3.90	2.85
200	G	10.95 (1.40)	4.96	2.70	1.71	1.58
	R	6.53 (1.10)	1.82	2.40	1.27	1.05
	I	5.93 (1.06)	2.15	2.03	0.94	0.82
	If	7.45 (1.17)	0.79	3.15	1.93	1.58
	Te	4.30 (0.91)	0.88	1.42	1.18	0.82
	Tf	4.71 (0.95)	3.30	0.54	0.65	0.23
	N	10.44 (1.37)	0.00	5.53	2.40	2.51
500	G	8.17 (1.22)	0.46	3.58	2.30	1.82
	R	7.74 (1.20)	0.14	3.65	2.27	1.67
	I	4.49 (0.93)	1.76	1.70	0.57	0.45
	If	6.82 (1.13)	0.24	3.21	1.90	1.47
	Te	3.18 (0.78)	0.43	1.20	1.00	0.54
	Tf	3.76 (0.85)	2.72	0.41	0.52	0.11
	N	10.58 (1.38)	0.00	5.64	2.39	2.55

The components' average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

t-noise (Tables 9, 10). As expected, the methods based on t-mixtures performed best in this setup. The If-method performed better than R in most respects (except with respect to classification for $n = 200$, and estimation of variances for $n = 50, 200$). It is interesting here that, whereas the I-method was overall better in terms of classification than If (as it should be), the If-method was better in terms of proportion estimation. The G-method was worse than the robust competitors, though still better than the N-method for $n = 500$.

Gaussian mixture (Tables 11, 12). As expected, the N-method was optimal here, but the almost flawless performance of the RIMLE (I and If) is remarkable. All other methods suffered from finding many outliers where they did not exist, though the Te-method was still relatively good.

General In most setups, the results with respect to classification were much more conclusive than those with respect to estimation. The G-method was generally bad for small n; apparently $n = 50$ is not enough to locate the uniform component correctly. For $n = 200$, its overall performance was still disappointing, given that four out

Table 10 Upper-trimmed means of classes of parameters for "*t*-noise"

n	Method	c	d_π	d_μ	d_v
50	G		0.35 (0.01)	1.01 (0.07)	1.29 (0.04)
	R		0.30 (0.01)	0.39 (0.02)	0.78 (0.03)
	I	0.056	0.21 (0.01)	0.37 (0.02)	0.88 (0.03)
	If	0.016 (0.001)	0.18 (0.01)	0.38 (0.01)	1.19 (0.05)
	Te		0.11 (0.01)	0.33 (0.01)	0.79 (0.04)
	Tf		0.11 (0.00)	0.34 (0.01)	0.65 (0.03)
	N		0.21 (0.00)	0.43 (0.01)	1.78 (0.07)
200	G		0.16 (0.00)	0.21 (0.01)	0.84 (0.03)
	R		0.14 (0.01)	0.16 (0.00)	0.52 (0.02)
	I	0.056	0.16 (0.00)	0.16 (0.00)	0.44 (0.01)
	If	0.041 (0.001)	0.11 (0.00)	0.17 (0.00)	0.79 (0.02)
	Te		0.06 (0.00)	0.14 (0.00)	0.35 (0.01)
	Tf		0.06 (0.00)	0.15 (0.00)	0.33 (0.01)
	N		0.20 (0.00)	0.21 (0.01)	1.64 (0.02)
500	G		0.10 (0.00)	0.12 (0.00)	0.91 (0.02)
	R		0.09 (0.00)	0.11 (0.00)	0.85 (0.02)
	I	0.056	0.18 (0.00)	0.09 (0.00)	0.30 (0.01)
	If	0.048 (0.000)	0.07 (0.00)	0.11 (0.00)	0.72 (0.02)
	Te		0.05 (0.00)	0.09 (0.00)	0.24 (0.01)
	Tf		0.04 (0.00)	0.10 (0.00)	0.26 (0.01)
	N		0.21 (0.00)	0.15 (0.00)	1.78 (0.02)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

of six setups fulfilled its model assumptions. The Te-, Tf- and N-method performed best in the setups for which their model assumptions were fulfilled, but fell behind in most other situations, though Te and Tf were often surprisingly good at estimation of mixture proportions, which apparently is still possible with suboptimal results with respect to the other criteria. The Te-method (estimating the degrees of freedom) was almost always better, and sometimes much better than Tf. The R-method was good for the outlier process, but in most situations it was worse than the If-method, which often was close to the I-method. This is a good result for the If-method, because the I-method was defined using (in practice unavailable) information about the true data generating process. The R-, Te- and Tf-method (and the G-method for small n) have a tendency to find too many outliers. As expected, the N-method is usually by far the worst one where its model assumptions are violated.

The variability of the results can be assessed by the standard errors for global misclassification rates and 90%-upper-trimmed distance means given in the tables. Note, however, that even though standard errors for some quantities may seem to be large, they give a too pessimistic impression of what can be said about the comparisons between methods, because all methods were computed on the same simulated data sets. In order to get an impression of which comparisons are statistically meaningful,

Table 11 Average misclassification percentages for “Gaussian mixture”

n	Method	Global	Component			
			Noise	2	3	4
50	G	28.05 (2.01)	24.88	0.54	1.54	1.09
	R	24.10 (1.91)	22.78	0.32	0.85	0.16
	I	3.06 (0.77)	0.00	0.76	1.80	0.50
	If	4.39 (0.92)	1.44	0.72	1.85	0.39
	Te	6.86 (1.13)	4.56	0.54	1.46	0.30
	Tf	11.88 (1.45)	10.15	0.42	1.15	0.15
	N	2.99 (0.76)	0.00	0.76	1.74	0.50
200	G	17.63 (1.70)	16.36	0.46	0.64	0.17
	R	10.82 (1.39)	10.09	0.32	0.35	0.05
	I	1.83 (0.60)	0.00	0.67	0.94	0.23
	If	2.92 (0.75)	1.21	0.62	0.90	0.19
	Te	5.65 (1.03)	4.55	0.39	0.64	0.07
	Tf	10.47 (1.37)	9.84	0.20	0.40	0.04
	N	1.83 (0.60)	0.00	0.67	0.93	0.23
500	G	6.87 (1.13)	5.73	0.48	0.51	0.15
	R	4.32 (0.91)	3.09	0.55	0.57	0.12
	I	1.65 (0.57)	0.00	0.67	0.76	0.22
	If	2.09 (0.64)	0.49	0.65	0.75	0.21
	Te	5.43 (1.01)	4.50	0.38	0.48	0.07
	Tf	10.22 (1.35)	9.77	0.16	0.26	0.03
	N	1.66 (0.57)	0.00	0.66	0.76	0.23

The components’ average misclassification percentage is the average percentage of points wrongly assigned to that component. Percentages are computed with respect to n. Standard errors are given in brackets

we also ran some paired two-sample Wilcoxon tests. Most of these were highly significant even in cases where there was considerable overlap between confidence intervals roughly calculated from standard errors. Just to give a single example for this, the comparison between the global misclassification rates of the I- and If-method for $n = 200$ in Table 1 was still borderline significant under the paired test (difference 0.06 between methods with both standard errors estimated to be about 0.9, but paired Wilcoxon $p = 0.043$).

Table 13 is consistent with the other results in the sense that in most cases the N- and G-method, which performed worst in other respects as well, produced the largest number of outlying estimators. Interestingly, the outlier percentages rarely behave monotonically over n, but of course the percentages are based on small numbers, so some variation can be expected.

6 Concluding remarks

Though different methods “win” different setups in the simulation study, the RIMLE method with c estimated by the filtering method (If) as explained in Sect. 2.4 can

Table 12 Upper-trimmed means of classes of parameters for "Gaussian mixture"

n	Method	c	d_π	d_μ	d_v
50	G		0.57 (0.01)	1.41 (0.05)	2.64 (0.04)
	R		0.65 (0.01)	1.00 (0.03)	2.74 (0.04)
	I	0.001	0.15 (0.00)	0.78 (0.02)	1.76 (0.05)
	If	0.006 (0.001)	0.18 (0.01)	0.81 (0.02)	1.90 (0.05)
	Te		0.19 (0.00)	0.83 (0.02)	1.97 (0.04)
	Tf		0.27 (0.00)	0.89 (0.02)	2.29 (0.04)
	N		0.15 (0.00)	0.78 (0.02)	1.76 (0.05)
200	G		0.39 (0.01)	0.66 (0.02)	1.60 (0.03)
	R		0.41 (0.01)	0.47 (0.01)	1.56 (0.03)
	I	0.001	0.08 (0.00)	0.42 (0.01)	0.83 (0.02)
	If	0.019 (0.001)	0.12 (0.00)	0.42 (0.01)	0.94 (0.02)
	Te		0.13 (0.00)	0.44 (0.01)	1.29 (0.02)
	Tf		0.21 (0.00)	0.46 (0.01)	1.81 (0.02)
	N		0.08 (0.00)	0.41 (0.01)	0.82 (0.02)
500	G		0.22 (0.00)	0.35 (0.01)	0.96 (0.02)
	R		0.22 (0.00)	0.27 (0.01)	0.89 (0.02)
	I	0.001	0.05 (0.00)	0.25 (0.01)	0.53 (0.01)
	If	0.020 (0.001)	0.08 (0.00)	0.26 (0.01)	0.60 (0.01)
	Te		0.10 (0.00)	0.27 (0.01)	1.14 (0.01)
	Tf		0.20 (0.00)	0.29 (0.01)	1.79 (0.02)
	N		0.05 (0.00)	0.25 (0.01)	0.53(0.01)

For the If-method and I-method we also report the average value for the improper density c. Standard errors are given in brackets

be recommended as optimal in some situations, and always acceptable. Particularly it does not suffer as many other supposedly robust methods from overestimating the number of outliers/noise points, and is therefore clearly better than those for data from a plain Gaussian mixture.

Note that we did not include in the simulations a setup with outliers in which the better breakdown point of the I- and If-method can be directly observed. This can be better explored by finding out how extreme an outlier has to be added to data from a "nice" mixture in order to drive a mixture component away from the non-outliers. This does not depend strongly on simulated variation. Examples given in Hennig (2004, 2005) show that a single outlier has to be very extreme for the R- and t-method to achieve "practical breakdown" (for example above 10^7 for the R-method in a situation with "good data" between -5 and 10; as opposed to estimators with robustness problems in other statistical setups, these methods remain almost unaffected as long as added outliers are not extreme enough), but two or three outliers put together on the same point do not have to be that extreme.

The authors currently work on a similar study for multidimensional data; the filtering method for the RIMLE needs to be slightly modified for this because the

Table 13 Percentage of estimation of mean parameters of a method in upper 1% of distances d_μ from true means aggregated over all methods for given n and model (these sum up to 7 for every n/model-combination because there are seven methods)

Model	n	Method						
		G	R	I	If	Te	Tf	N
Side noise	50	1.6	0.4	0.2	1.2	0.8	1.4	1.4
	200	0.8	2.0	1.6	1.0	0.8	0.2	0.6
	500	0.0	0.0	0.2	0.0	0.4	0.2	6.2
Inside noise	50	1.4	0.6	0.8	1.2	0.6	1.2	1.2
	200	3.4	1.8	0.8	0.4	0.2	0.2	0.2
	500	0.8	0.0	0.4	0.0	2.2	0.0	3.6
Wide noise	50	7.0	0.0	0.0	0.0	0.0	0.0	0.0
	200	2.0	0.2	0.2	0.2	0.4	0.4	3.6
	500	0.0	0.0	0.0	0.0	0.0	0.0	7.0
Outlier	50	5.2	0.0	0.0	0.0	0.0	0.0	1.8
	200	1.6	0.0	0.0	0.0	0.0	0.0	5.4
	500	0.2	0.0	0.0	0.0	0.0	0.0	6.8
t-Noise	50	3.0	0.6	0.4	0.4	0.4	0.6	1.6
	200	6.2	0.0	0.0	0.0	0.0	0.0	0.8
	500	1.8	1.0	0.0	0.6	0.0	0.0	3.6
Gaussian	50	4.0	1.2	0.4	0.4	0.2	0.4	0.4
	200	7.0	0.0	0.0	0.0	0.0	0.0	0.0
	500	4.2	0.4	0.4	0.4	0.4	1.0	0.2

multidimensional Kolmogorov distance is not suitable. The G-method is computationally cumbersome to generalise, and the current study suggests that it would not be worthwhile anyway (though mixtures of Gaussian and uniform distributions may be interesting for reasons other than robustness).

Designing simulations like the present one involves subtle decisions in order to make parameters and results from methods and data generating processes based on different models comparable. The present study does this in a particular way, which may be controversial, but at least highlighting the issue is an intended contribution of this paper.

Certainly it would be interesting to compare the methods discussed here with further robust clustering competitors such as high breakdown methods based on a fixed partition model, and to simulate situations where the number of clusters is estimated.

Acknowledgments We thank the referees for their careful reading and suggestions that led to strong improvements of the manuscript.

References

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

Coretto P (2008) The noise component in model-based clustering. PhD thesis, Department of Statistical Science, University College London. http://www.ontherubicon.com/pietro/docs/phdthesis.pdf

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

Fraley C, Raftery AE (1998) How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput J 41:578–588

Fraley C, Raftery AE (2006) Mclust version 3 for r: normal mixture modeling and model-based clustering. Technical report 504, Department of Statistics, University of Washington

Gallegos MT, Ritter G (2005) A robust method for cluster analysis. Ann Stat 33(5):347–380

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2008) A general trimming approach to robust cluster analysis. Ann Stat 38(3):1324–1345

Hathaway RJ (1985) A constrained formulation of maximum-likelihood estimation for normal mixture distributions. Ann Stat 13:795–800

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

Hennig C (2005) Robustness of ML estimators of location-scale mixtures. In: Baier D, Wernecke KD (eds) Innovations in classification. Data science, and information systems. Springer, Heidelberg, pp 128–137

Hennig C, Coretto P (2008) The noise component in model-based cluster analysis. In: Preisach C, Burkhardt H, Schmidt-Thieme L, Decker R (eds) Data analysis, machine learning and applications. Springer, Berlin, pp 127–138

Hosmer DW (1978) Comment on “Estimating mixtures of normal distributions and switching regressions” by R. Quandt and J.B. Ramsey. J Am Stat Assoc 73(364):730–752

Karlis D, Xekalaki E (2003) Choosing initial values for the EM algorithm for finite mixtures. Comput Stat Data Anal 41(3–4):577–590

Liu C (1997) ML estimation of the multivariate *t* distribution and the EM algorithms. J Multivar Anal 63:296–312

McLachlan G, Krishnan T (1997) The EM algorithm and extensions. Wiley, New York

McLachlan G, Peel D (2000) Robust mixture modelling using the *t*-distribution. Stat Comput 10(4):339–348

Neykov N, Filzmoser P, Dimova R, Neytchev P (2007) Robust fitting of mixtures using the trimmed likelihood estimator. Comput Stat Data Anal 17(3):299–308

Redner R, Walker HF (1984) Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev 26:195–239