An adequacy approach for deciding the number of clusters for OTRIMLE robust Gaussian mixture based clustering

Christian Hennig* Pietro Coretto†

3rd September 2020

Abstract

We introduce a new approach to deciding the number of clusters. The approach is applied to Optimally Tuned Robust Improper Maximum Likelihood Estimation (OTRIMLE; Coretto and Hennig (2016)) of a Gaussian mixture model allowing for observations to be classified as “noise”, but it can be applied to other clustering methods as well. The quality of a clustering is assessed by a statistic $Q$ that measures how close the within-cluster distributions are to elliptical unimodal distributions that have the only mode in the mean. This nonparametric measure allows for non-Gaussian clusters as long as they have a good quality according to $Q$. The simplicity of a model is assessed by a measure $S$ that prefers a smaller number of clusters unless additional clusters can reduce the estimated noise proportion substantially. The simplest model is then chosen that is adequate for the data in the sense that its observed value of $Q$ is not significantly larger than what is expected for data truly generated from the fitted model, as can be assessed by parametric bootstrap. The approach is compared with model-based clustering using the Bayesian Information Criterion (BIC) in a simulation study and on two datasets of scientific interest.

Keywords: parametric bootstrap; noise component; unimodality; model-based clustering

*christian.hennig@unibo.it, Dipartimento di Scienze Statistiche, Universita di Bologna, Italy
†Department of Economics and Statistics University of Salerno, Italy
1 Introduction

We introduce an approach for finding a suitable number of clusters for use with Optimally Tuned Robust Improper Maximum Likelihood (OTRIMLE) clustering (Coretto and Hennig, 2016, 2017), which attempts to find approximately Gaussian distributed clusters allowing for some observations to be classified as noise or outliers. The approach in its general form is very flexible and can be adapted to other clustering methods and other types of clusters, but we focus on its use with OTRIMLE here.

Here is a key issue with choosing the number of clusters. In reality, model assumptions never hold precisely, so it is important that statistical methods produce reasonable results even if the model assumptions are violated. The problem with this is that it is usually defined in terms of the nominal (assumed) model what the method tries to estimate, and if the model does not hold, it is not always clear what a “reasonable” result would be. If clusters are supposed to be (approximately) Gaussian, using a Gaussian mixture model for clustering (Banfield and Raftery, 1993) looks attractive. Estimation of the number of clusters for this is often done using the Bayesian Information Criterion (BIC), e.g., in the R package mclust (Scrucca et al., 2016). The BIC has been proven to be consistent for estimating the number of mixture components (Keribin, 2000) under some rather restrictive assumptions, and is believed to be more generally consistent. This looks like good news, but is in fact a problem. In reality data do not stem precisely from a Gaussian mixture model, but a Gaussian mixture model with a sufficiently large number of mixture components can approximate more or less any distribution arbitrarily well (for a recent precise version of this statement and a discussion of some older versions see Nguyen et al. (2020)). This means that if the number of observations becomes larger, a consistent method for estimating the number of mixture components can be expected to add mixture components in order to fit the real distribution better and better, and ultimately several components will fit an approximately but not precisely Gaussian subset of the data that intuitively would qualify as a single cluster, in turn overestimating the number of clusters. This has also been observed in practice for the BIC (Hennig, 2010). The estimation of the number of clusters is therefore affected by violations of the model assumptions in a more critical way than most standard statistical estimation problems.

Allowing some observations to be classified as noise as is done in OTRIMLE and other robust clustering methods in order to treat outliers appropriately adds another issue. There is an ambiguity between noise and clusters in two respects. Firstly, it is not clear how large a group of outliers has to be in order to be interpreted as a cluster on its own, and secondly, there may be
very widely spread observations that can be well approximated by a Gaussian
distribution with a very low density everywhere but may more appropriately
be interpreted as noise than as a cluster, depending on the subject matter
and the meaning of the data. Not allowing for noise classification does not
really avoid these issues, because observations of such a kind will still either
be declared a cluster on their own, or integrated in other clusters affecting
their estimation, potentially leading to an inappropriate clustering.

A further issue is that to some extent more mixture components can be
traded off against more flexible covariance matrices. Too flexible covariance
matrices are already an issue for a fixed number of mixture components
because of potential degeneracy or near-degeneracy of the likelihood.

The consequence is that finding an appropriate number of clusters should
not be seen as a well-defined estimation problem in a statistical model. Rather it essentially requires decisions by the user: how much better ap-
proximation of the data, how much simpler covariance matrix structure that
is less prone to degeneracy, and what decrease of the noise proportion, would
justify adding another mixture component? A method that does not require
any user input such as the BIC should not be trusted naïvly. These issues
are acknowledged for example by the authors of the R-package tclust for
robust trimmed clustering (Fritz et al., 2012), who do not offer an auto-
matic method for choosing the number of clusters, but rather some graphical
displays that allow the used to track the different aspects to be traded off
against each other.

On the other hand, in many situations users do not have sufficient back-
ground knowledge to make all the required decisions in a well founded man-
er, and also an automatic approach that does not require manual adaptation
to every data set is required to systematically evaluate the quality of an ap-
proach. For this reason we offer an approach that allows the user to make
the required tuning decisions but we also suggest some default choices to give
the user a starting point and to enable evaluation by simulation.

The approach is based on the concept of “adequacy” introduced by Davies
(1995). According to this concept, a model (Davies’ use of the term “model”
includes specific distributions with given parameter values) is adequate for
a dataset with respect to a statistic \( Q \) if the value of \( Q \) on the dataset is
“typical” for datasets generated by the model. This does not mean anything
else than that a significance test based on \( Q \) does not reject the model. \( Q \)
is chosen to reflect the sense in which the model needs to “fit” the data
in a given application rather than following optimality considerations such
as those by Neyman-Pearson; more than one test statistic can be chosen
and can be combined using Bonferroni’s correction. Unless the distribution
of \( Q \) on the model can be handled analytically, parametric bootstrap can
be applied to approximate this distribution. The selection of the number of clusters is a model selection problem, and Davies recommends to select the simplest model that is adequate for the data (Davies and Kovac 2001), which could be the model with the lowest number of mixture components, but see Section 4.4. Note that whenever a mixture with a low number of mixture components fits the data adequately, the data could also be fit by a model with more mixture components (one could just add low probability low spread components around single observations), which means that the data actually cannot distinguish between a model with a small number of well fitting mixture components and a model with a larger number of components, despite the fact that automatic rules such as the BIC suggest that this were possible. Choosing the simplest model that fits is just a pragmatic choice.

The OTRIMLE method is introduced in Section 2. Section 3 gives an outline of the approach for deciding the number of clusters. This approach requires a number of decisions by the user. Section 4 contains proposals for these decisions. Particularly, a statistic $Q$ is proposed that measures to what extent the found clusters in a dataset for a given number of clusters qualify as “adequate”. In Section 5 we compare the method with the BIC for Gaussian mixtures, Gaussian mixtures with noise, and a mixture of skew t-distributions. Section 6 concludes the paper.

2 The OTRIMLE approach to robust clustering

When using mixture models for cluster analysis, usually mixtures of families of distributions are considered that formalise the idea of a homogeneous cluster. Every mixture component is then interpreted as modelling a cluster, and the number of mixture components corresponds to the number of clusters (there are exceptions to this, see Hennig (2010)).

The most popular choice for continuous data is the family of Gaussian distributions. A standard Gaussian mixture model assumes data $x_1, \ldots , x_n$ to be generated independently identically distributed from a distribution with density

$$f(x; \theta) = \sum_{g=1}^{G} \pi_g \phi_p(x; \mu_g, \Sigma_g),$$

(1)

where $\phi_p(\cdot; \mu, \Sigma)$ is the $p$-variate Gaussian density with mean $\mu$ and covariance matrix $\Sigma$, $\pi_g \in [0, 1]$ for $j = 1, 2, \ldots , G$, $\sum_{i=1}^{G} \pi_g = 1$, and $\theta$ is the parameter vector collecting all $\pi_g, \mu_g, \Sigma_g, j = 1, 2, \ldots , G$. For given $G$, the parameters $\theta$ can be estimated by maximum likelihood. More precisely,
a global optimum is often not available, and algorithms such as the EM-algorithm are used that find a local optimum of the likelihood. Given estimators (here denoted $\hat{\theta}, \hat{\pi}_g, \hat{\mu}_g, \hat{\Sigma}_g, j = 1, 2, \ldots, G$), probabilities that observations $x_i, i = 1, 2, \ldots, n$, were generated by mixture component $g$ can be estimated as

$$p_{ig} = \frac{\hat{\pi}_g \phi_p(x_i; \hat{\mu}_g, \hat{\Sigma}_g)}{f(x_i; \hat{\theta})}, \quad (2)$$

and observation $i$ can be assigned to the mixture component $g$ that maximises $p_{ig}$. This is implemented in the R-package mclust (Banfield and Raftery, 1993; Scrucca et al., 2016), along with a number of models defined by various constraints on the within-component covariance matrices. The mclust-approach for deciding the number of mixture components $G$ and the covariance matrix model is to minimise the Bayesian Information Criterion (BIC),

$$\text{BIC} = k \ln(n) - 2 \ln(\hat{L}_n), \quad (3)$$

where $k$ is the number of free parameters ($k = (G - 1) + pG + p(p + 1)G/2$ for a model with fully free covariance matrices), and $\hat{L}_n$ is the maximised likelihood for the model under investigation.

It is well known that statistical methods based on a Gaussian distributional assumption can be strongly affected by outliers, and this not different in cluster analysis. For fixed $G$, outliers have to be included in a cluster, in turn affecting their mean and covariance matrix estimators and often the classification of many further observations. In order to deal with this, Banfield and Raftery (1993) proposed to add a so-called “noise component” to the mixture in order to collect outliers and to prevent them from affecting the Gaussian clusters. The density then becomes

$$f(x; \theta) = \pi_0 \delta + \sum_{g=1}^{G} \pi_g \phi_p(x; \mu_g, \Sigma_g), \quad (4)$$

$\delta \geq 0$, $\pi_0 \in [0, 1]$, and now $\sum_{i=0}^{G} \pi_g = 1$. They proposed to estimate the $\delta$ as $1/M$, where $M$ is the hypervolume of the smallest hyperrectangle to cover all data, assuming that $\delta = 0$ outside that hyperrectangle. The number of clusters is still estimated by the BIC, adding the $\pi_0$-parameter to the parameter count. Although this method often works reasonably well, it is actually not the maximum likelihood estimator for $\delta$ (Coretto and Hennig, 2011), and neither is it breakdown robust, because a single extreme outlier can make $M$ arbitrarily large, preventing any other outlier from being classified as noise (Hennig, 2004). The same holds for another mixture approach that
is meant to be more robust than plain Gaussian mixtures, namely mixtures of t-distributions (Peel and McLachlan, 2000). Hennig (2004) noted that a method with a better breakdown point can be defined by fixing $\delta$ in (1). Allowing $\delta$ to be positive on the whole Euclidean space makes $f$ an improper density, although a proper density can be defined that constrains the noise component to occur in an unspecified set of Lebesgue measure $1/\delta$ that is assumed to cover all actually observed data. In this way, all other parameters can still be estimated using the EM algorithm, enjoy improved robustness properties, and observations can still be clustered using (2). For multivariate Gaussian mixtures this has in detail been explored by Coretto and Hennig (2016, 2017) under the name “Robust Improper Maximum Likelihood Estimator” (RIMLE). Coretto and Hennig (2016) propose to choose $\delta$ as

$$\arg \min_{\delta} (D(\delta) + \beta \hat{\pi}_0(\delta)),$$

where $D(\delta)$ is a measure of the Kolmogorov-type difference between the distribution function of within-cluster Mahalanobis distances weighted by (2) between the observations and the cluster centre, and the $\chi^2$-distribution function, which should be observed for perfectly Gaussian distributed observations. The weighting assigns all observations to the clusters according to the estimated probability of being generated by that cluster, which particularly means that observations that have a high estimated probability of being “noise” will be downweighted. Minimising $D(\delta)$ means that $\delta$ is chosen so that the estimated clusters will look optimally Gaussian. This happens if $\beta = 0$ is chosen. $\beta$ is a tuning constant that allows for tolerating more non-normality within clusters if in turn the estimated noise probability $\hat{\pi}_0(\delta)$ is decreased. Coretto and Hennig (2016) suggest $\beta = 1/3$ as alternative to $\beta = 0$. This is particularly useful for estimating the number of clusters with clusters that are not necessarily required to be normal, see Section 5.1.

$D(\delta)$ can degenerate and becomes meaningless if $\delta$ is so large that all or most observations are classified as noise. Therefore, using (5) requires that the average posterior pseudo probability of observations to have been generated by the noise component is limited, and Coretto and Hennig (2017) propose an upper bound of 0.5.

Like other methods based on Gaussian mixtures, OTRIMLE needs to address the issue of a potentially degenerating likelihood due to covariance matrices with very small or zero eigenvalues. This is done imposing the constraint

$$\frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma < +\infty,$$

(6)
where $\lambda_{\text{max}}(\theta)$ and $\lambda_{\text{min}}(\theta)$ are the maximum and minimum of the eigenvalues of the covariance matrices of the different Gaussian mixture components parameterised in $\theta$, and $\gamma \geq 1$ is a constant to be chosen by the user. Based on experiments in Coretto and Hennig (2017), $\gamma = 20$ seems to be a sensible choice for standardised data (if the measurements of different variables in the dataset have different orders of magnitude, there is hardly any reasonable way to specify $\gamma$), although occasionally a user may look for either more spherical clusters (which requires smaller $\gamma$) or for even more flexibility of the covariance matrices (which requires larger $\gamma$). See García-Escudero et al. (2018) for a comprehensive discussion of covariance matrix constraints in Gaussian mixture modelling. Cerioli et al. (2018) argue that the choice of $\gamma$ has impact on the number of clusters, and explore this for the case of a plain Gaussian mixture model.

The resulting method is called “Optimally Tuned RIMLE” (OTRIMLE), and implemented in the R-package otrimle (Coretto and Hennig 2019). Theory including consistency for the canonical functional, a breakdown point, and detailed information about computation is given in Coretto and Hennig (2017). A simulation study comparing OTRIMLE with plain Gaussian mixtures and alternative robust methods is in Coretto and Hennig (2016).

3 An adequacy approach to decide the number of clusters

We have argued in the Introduction that the problem of finding a suitable number of clusters is essentially different from the problem of estimating the number of mixture components. Even if a Gaussian mixture model is precisely fulfilled, a “submixture” of several poorly separated Gaussian components taken together can still be unimodal and even look fairly close to a single Gaussian distribution. In most applications this would qualify as a single cluster, and the number of meaningful real clusters in such a case would be smaller than the number of Gaussian mixture components.

The problem of estimating the number of Gaussian mixture components is ill-posed in the sense that any dataset generated from a Gaussian mixture with a certain number of components can be arbitrarily well approximated by a mixture with more components; in any suitably defined neighbourhood of a Gaussian mixture there are Gaussian mixtures with arbitrarily many components. This particularly means that if the Gaussian mixture model assumption is not precisely fulfilled (as is always the case in reality), with enough observations a mixture with arbitrarily many components will fit the
data better than a mixture with few components, even if the latter may look like an excellent representation of the intuitive clusters in the data. This is illustrated in Figure 1 which shows data generated by a mixture of three multivariate $t_3$-distributions (generated by the setup “T Gauss.3” in [Coretto and Hennig (2016)]). The left side shows a clustering from a plain Gaussian mixture produced by mclust with default settings. Although there are three elliptical clusters clearly visible, the BIC estimates the number of Gaussian mixture components as 6, because the intuitive clusters have not been generated exactly by a Gaussian distribution. Adding a uniform noise component (right side of Figure 1) classifies some outliers appropriately as “noise”, but does not help with the estimation of the number of clusters, as the BIC still estimates 6 Gaussian components. A mixture of t-distributions will fit these data well with three mixture components, however if the underlying distributions are not exactly t-distributions, it runs into similar problems, see Section 5.1. The “estimation” of the number of mixture components is rather a model selection than an estimation problem, and a consistent method such as the BIC has more use for picking a mixture that fits the empirical density well than for interpreting the resulting components as clusters.

![Figure 1: Data generated from a mixture of three multivariate $t_3$-distributions with clustering by Gaussian mixture fitting (left side) and Gaussian mixture fitting with noise component (right side); the number of mixture components was estimated by the BIC.](image)

This implies that the problem of deciding the number of clusters is not a well defined statistical estimation problem. It does not only rely on parameters of an assumed underlying distribution, but also on user decisions. Even
assuming that the Gaussian distribution is used as a “cluster prototype”, i.e., a cluster should look Gaussian or similar, the user has to decide

1. what is required of a data subset to be interpreted as cluster,
2. how far from a Gaussian distribution a within-cluster distribution is tolerated to be,
3. in case that some observations can be classified as outliers/noise, how small and homogeneous an outlying data subset is required to be in order to be interpreted as cluster rather than a group of outliers.

These decisions cannot be made from the data alone, and therefore user tuning is essential for estimating the number of clusters. We believe that this is quite generally the case in cluster analysis, and that the vast majority of the literature ignores this, probably because most users expect a solution without having to make decisions, and a solution that depends crucially on user tuning may not be accepted as “objective”; see Gelman and Hennig (2017) for a discussion of this issue.

We now introduce a general scheme for deciding the number of clusters that can be applied to general model-based clustering methods, and that can be tuned by the user addressing the issues above.

The scheme is based on a general approach to model selection proposed first in Davies (1995) and more explicitly (in the context of nonparametric regression) in Davies and Kovac (2001). The idea is that one can choose the simplest model that is adequate for the data in the sense that it produces data that cannot be distinguished from typical data generated by the model. Obviously, more complex models can be adequate as well, as is the case in mixture modelling, but a more complex model will not be chosen if a simpler one exists that is already adequate. Entry points for user tuning are

1. the target model, i.e., the model for which adequacy of the data is evaluated (in cluster analysis this will often be a mixture model; here a Gaussian mixture model, as we assume that the Gaussian distribution serves as “cluster prototype”),
2. the statistic or potentially more than one statistics that are used to distinguish the data from what is expected under the model (in cluster analysis a statistic $Q$ is required that measures whether what is interpreted as clusters behave as clusters should behave in the application at hand),
3. how atypical data has to look like in order to decide against the model (standard significance levels such as 0.01 or 0.05 may be used),
4. the formal definition of simplicity $S$ (in cluster analysis the standard choice would be the number of clusters, but we will penalise this with the estimated noise proportion in order to stop the method from declaring too many observations “noise”).

We will work with a statistic $Q$ that does not allow for simple analytic derivation of its distribution for data generated by a mixture, and therefore its distribution will be approximated by parametric bootstrap.

Let $X = (x_1, \ldots, x_n)$, $x_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$, $i = 1, \ldots, n$ be the dataset and $C_G(X)$ be the output of the clustering method $C$ with $G$ clusters on $X$. Here is the general scheme:

1. Choose a target model, a clustering method that fits the target model, a statistic $Q$ that measures clustering quality, and a statistic $S$ measuring the simplicity of a fit. In practice also a maximum number $G_{\text{max}}$ of clusters and a number of bootstrap resamples $B$ are required.

2. For $G = 1, \ldots, G_{\text{max}}$, compute a fit (clustering) of $X$ with $G$ clusters.

3. For $G = 1, \ldots, G_{\text{max}}$, generate $B$ datasets $D_{G,b}$, $b = 1, \ldots, B$ from the fitted model.

4. For given $G$, the clustering is adequate for the data if $Q(C_G(X))$ is consistent with the empirical distribution of $Q(C_G(D_{G,b}))$, see Section 4.3.

5. The final number of clusters is chosen as $\arg \min_{G \text{ adequate}} S(G)$. In the simplest case $S(G) = G$, and the scheme can be stopped once an adequate $G$ is found.

A possible outcome of the scheme is that no clustering is adequate. This is informative for the user in its own right, and means that the data are not compatible with the target model, at least not for $G \leq G_{\text{max}}$. There are various options to enforce a clustering if it is required anyway. One could try a larger $G_{\text{max}}$, choose the best found clustering according to $C(G)$, or $\frac{C(X) - m_QG}{s_QG}$ (see Section 4.3 for the definition), or try a non-model based clustering method.

## 4 Key decisions and tuning

The clustering method of interest here is OTRIMLE. The number of bootstrap replications $B$ and the maximum number of clusters $G_{\text{max}}$ should optimally be as large as possible, but the method is computationally intensive,
so they need to be limited for pragmatic reasons. The choice of \( G_{\max} \) should also depend on potential background information about a realistic or required number of clusters. \( B \) should be at least around 20 to give the method some stability, but \( B = 100 \) and higher would be better. The further choices are less straightforward.

4.1 Data generation from the target model

The target model should be a Gaussian mixture with noise, similar to (4), but (4) in the given form is not a proper probability model without constraining the set where noise (i.e., observations from mixture component zero) can occur.

With all parameters estimated by OTRIMLE and assuming the noise to be constrained to an unspecified set of Lebesgue measure \( 1/\delta \), the estimated posterior probability of observation \( x_i, i = 1, \ldots, n \), to be noise is

\[
\hat{p}_{i0} = \frac{\hat{\pi}_0 \delta}{f(x_i; \theta)}.
\]

For data generation from the target model for the parametric bootstrap, an observation is assigned to the noise with probability \( \hat{\pi}_0 \), and given that it is assigned to the noise, we propose to resample it from the existing dataset with the noise distribution defined by

\[
\hat{P}_0\{x_i\} = \frac{\hat{p}_{i0}}{\sum_{h=1}^n \hat{p}_{h0}},
\]

so that the probability of every observation to be drawn as noise is proportional to its estimated probability to be noise in the dataset. Non-noise data are generated in a standard way from the estimated Gaussian mixture.

4.2 The clustering quality statistic

The clustering quality statistic \( Q \) is meant to formalise what a “good” clustering is. We do not insist on a precisely Gaussian shape, but we assume that the clusters of interest here should be elliptical and unimodal with density decreasing from the mean symmetrically in all directions. In such a case the use of the Gaussian distribution as a cluster prototype and the Gaussian mixture approach seem justified.

The \( Q \) proposed here measures in a nonparametric way to what extent the clusters have such a shape. We start from a one-dimensional measure for a single cluster. The values of this measure are then aggregated over all
principal components and over all clusters to compute the overall $Q$. The definition is not motivated by any model-based optimality theory, but rather custom-made in order to express exactly what is required. It is based on a test for unimodality by (Pons [2013], p. 103).

Assuming one-dimensional data standardised to have mean zero and variance one in cluster $g = 1, \ldots, G$, we use the following definition:

1. Choose a kernel density estimator and $q$ points $z_1 < z_2 < \ldots < z_q$ symmetrically around the mean. Our software uses the default of the R-function `density`, $q = 100$, and the 100 points are chosen as $p$-quantiles of the standard Gaussian distribution with $p$ ranging from 0.005 to 0.995 in equidistant manner.

2. Compute kernel density estimators at the quantiles $\hat{f}(z_1), \ldots, \hat{f}(z_q)$ based on a weighted sample in which $x_{ij}$ has a weight according to (2).

3. Let $\hat{f}^{(1)} \geq \hat{f}^{(2)} \geq \ldots \geq \hat{f}^{(q)}$ be the sorted version of $\hat{f}(z_1), \ldots, \hat{f}(z_q)$.

4. For $h = 1, \ldots, q/2$, let $\hat{f}^{*h} = \frac{\hat{f}^{2h-1} + \hat{f}^{2h}}{2}$. This implies that $f^{*1}, f^{*2}, \ldots, f^{*(q/2)}, f^{*(q/2)}, \ldots, f^{*1}$ is a symmetric version of the original $\hat{f}(z_1), \ldots, \hat{f}(z_q)$.

5. Compare the symmetrised kernel density with the mean ($q_l$ and $q_r$ refer to the left and right side of the mean, respectively):

$$q_l = \frac{1}{q} \sum_{i=1}^{q/2} (\hat{f}(z_{q/2+1-i}) - \hat{f}^{*i})^2, \quad q_r = \frac{1}{q} \sum_{i=1}^{q/2} (\hat{f}(z_{q/2+i}) - \hat{f}^{*i})^2.$$  

Aggregating: $\tilde{Q}_g = \sqrt{\frac{1}{q} (q_l + q_r)}$.

The process is illustrated in Figure 2. In case that the estimated density in fact decreases monotonically and symmetrically from the mean, $\tilde{Q}_g = 0$, which is the best possible value.

For aggregating $\tilde{Q}_g$-values over different clusters, it is important to take the size of the estimated clusters, i.e., $\hat{\pi}_g$, $g = 1, \ldots, G$, into account in order to avoid that the overall measure is dominated by a highly unreliable value from a small clusters. The rationale is not to give bigger clusters more weight, because this is about estimating the number of clusters, so small clusters that are bad should not be tolerated. However, $\tilde{Q}_g$ can also be expected to be more variable for even valid small clusters, and this needs to be accounted for. Therefore we use

$$Q_g^* = \frac{\tilde{Q}_g - E_{n\hat{\pi}_g} \tilde{Q}_g}{\sqrt{\text{var}_{n\hat{\pi}_g}(\tilde{Q}_g)}}.$$  

12
Figure 2: Illustration of the one-dimensional measurement of cluster quality. Left side: Suppose this is the kernel-estimated density for the weighted dataset within the first estimated cluster at $z_1, \ldots, z_q$, obviously not looking unimodal. Middle: Density values at $z_1, \ldots, z_q$ are ordered from the largest to the smallest. Pairs of density values (the two largest ones, then the third and fourth largest and so on) are averaged, and the resulting density values are shown on the right side of the mean at $z_{q/2+1}, \ldots, z_q$ from largest to smallest. Right side: The same values are also put on the left side of the mean in descending order from the mean to the outskirts, producing a density symmetric about the mean. $Q_q$ is the root of the averaged squared difference between these.
where the expectation $E_m$ and variance $\text{var}_m$ are computed assuming $m$ i.i.d. observations from an $\mathcal{N}(0, 1)$-distribution in the corresponding cluster. These values can be simulated to very high precision and interpolated to allow for non-integer $m$.

For $p$-dimensional clusters with $p > 1$, within cluster principal components (PCs) are computed first, based on the weighted within cluster data with weights according to (2) again. For $j = 1, \ldots, p$, let $Q_{jg}$ be $Q_g^*$ computed on the $j$th standardised within-cluster PC of cluster $g$. Aggregating information from the PCs,

$$Q_g = \frac{1}{p} \sum_{j=1}^{p} (Q_{jg}^2) 1(Q_{jg} > 0),$$

where $1(\cdot)$ denotes the indicator function. The rationale here is that (a) if $Q_{jg} \leq 0$ it means that on the $j$th PC, the symmetric unimodality statistic behaves as expected under a Gaussian distribution or even better, so there is no indication whatsoever against this being a cluster, and (b) squaring positive $Q_{jg}$ will emphasise problematic issues in certain PCs.

Finally, for the same reason squares are applied when aggregating over the clusters in order to make $q$ sensitive against substantial issues in any cluster:

$$Q(G) = \sqrt{\sum_{g=1}^{G} (Q_g^2)}.$$

4.3 Bootstrap adequacy

Because the method is computer intensive and precise quantiles may require a too large $B$, $G$ will be defined to be adequate if

$$\frac{Q(G) - m_{QG}}{s_{QG}} \leq c,$$

where $m_{QG}$ and $s_{QG}$ are location and scatter statistics of the empirical distribution of $Q(G)$ for data generated from the fitted model. We have observed that with OTRIMLE (as potentially with other clustering methods) $Q(G)$ may produce outlying values. Certain fitted distributions may generate datasets that are quite ambiguous regarding the optimal clustering and the number of clusters. Such outlying values normally indicate a very bad clustering, and $Q(G)$ on the original dataset should not be assessed as adequate just because certain $Q(G)$ on bootstrapped data are even worse. For
this reason, \( m_{QG} \) and \( s_{QG} \) should be chosen robustly. We suggest the robust \( \tau \)-estimator for location and scale (Maronna and Zamar, 2002). With appropriate consistency factor, this is consistent if the parametric bootstrap distribution of \( Q(G) \) is Gaussian, allowing for a standard interpretation of the constant \( c \).

\( Q \) is assumed to be defined so that lower values imply a better clustering quality, and adequacy will only be rejected if \( Q(G) \) is too large. Choosing, e.g., \( c = 2 \) then means that if \( Q(G) \) on bootstrapped data follows a Gaussian distribution, the probability that adequacy is rejected is about 0.977.

4.4 The simplicity measure

The simplest choice for the simplicity measure \( S \) is \( S(G) = G \); a model is seen as simpler if it has fewer clusters. This is appropriate for standard non-robust clustering, but it is problematic if it is allowed to classify a number of observations as “noise”. With OTRIMLE, as well as with trimmed clustering and the noise component in \texttt{mclust}, it would be possible to declare all observations “noise” that make clustering ambiguous or belong to small clusters, in which case a high quality clustering with small \( G \) for the remaining observations could be found easily. For this reason, and because it is generally ambiguous whether observations that belong to small groups in some distance from the bigger clusters should be declared noise or clusters on their own, too much noise should be penalised. We propose

\[
S(G) = G + \frac{\hat{\pi}_0}{p_0},
\]

where \( p_0 \) is a constant chosen by the user. It specifies the smallest percentage of additional noise that the user is willing to trade in for adding another cluster, i.e., if \( p_0 = 0.05 \) (which we use as a default), it means, say, that a clustering with \( G = 6 \) and \( \hat{\pi}_0 = 0.04 \) is assessed as “simpler” as a clustering with \( G = 5 \) and \( \hat{\pi}_0 = 0.1 \). The former clustering will then be preferred by our method if both clusterings are adequate. Particularly this will normally imply that clusters with \( \hat{\pi}_0 < p_0 \) are not found, because they could simply be declared noise and the resulting clustering would be “simpler” and as adequate, although there may be exceptions in case that the smallest cluster has a high quality \( Q_g \) compared to the other clusters.

5 Experiments

The adequacy approach to choose the number of clusters with OTRIMLE (called “adotrimle” in the following) is compared to different BIC/mixture
model-based methods in a simulation study and on two datasets of scientific interest, one with and the other one without given true $G$. There is always a tension between stating that a method requires user tuning dependent on the specific situation, and running it in a default fashion on artificial datasets, but we think that both of these have their justification. Where user decisions can be used with convincing justification to adapt the method to what is required in a given application, this is certainly recommended. However in many situations the user does not have a clear idea how to make some or all of these choices, and therefore defaults are often useful. They are also required in order to compare the method in a “neutral” fashion with others. In the following we choose $p_0 = 0.05$ in (8), i.e., we prefer a solution with one cluster more if that reduces the estimated noise by 0.05 or more. We did some experiments with $p_0 = 0.02$ (not shown), but results were rarely different. We choose $c = 2$ in (7) as maximum value of the standardised clustering quality for the model to still count as “adequate”. The maximum eigenvalue ratio for covariance matrices was chosen as $\gamma = 20$. Variables in the simulation study were standardised before clustering in order to allow for a scale-independent interpretation of $\gamma$; the datasets in Sections 5.2 and 5.3 were not standardised, because their variables are compatibly scaled by definition.

We looked at both $\beta = 0$ and $\beta = 1/3$ in (5), the latter meaning that for fixed $G$ more non-Gaussianity within clusters is tolerated if that reduced noise. Results were occasionally different. Note that $\beta$ is a tool to trade non-Gaussianity against noise, whereas $c$ tunes trading non-Gaussianity against non-adequacy of the non-noise, usually leading to more clusters (if anything changes at all, which it often does not).

We chose $G_{\text{max}} = 10$ in the simulations, but smaller (7 or 8, respectively) for the smaller datasets in Sections 5.2 and 5.3 where methods would have difficulties fitting parameters with too small clusters. This choice does not matter, however, as long as the finally chosen $G$ has a value of $S(G) < G_{\text{max}} + 1$ in (8), because then it will be chosen regardless of results for higher $G$. As far we have seen, for all datasets, larger $G_{\text{max}}$ could not have changed results for this reason; for the BIC this can never be known, which is an advantage of our approach.

The number of bootstrap replicates is chosen as $B = 30$ in the computer intensive simulations, but $B = 100$ in Sections 5.2 and 5.3.

**Declaration of selection bias.** As this paper introduces a new method, as a proof of concept we need to show some situations in which it works well. We looked at some other datasets and data generating mechanisms (although usually with a very small number of test runs). In many cases there was no big difference between the different methods, and sometimes mclust with or
without noise, or a mixture of t-distributions or skew t-distributions worked better, though never all of them. Sometimes nothing worked well. So we do not claim that adotrimle is universally the best, just where we show it is. DGP 3 was the first DGP we tried, and we show it despite not being a clear win for the new methods.

5.1 Simulation study

In this study we compare two versions of adotrimle (with \( \beta = 0 \) and \( \beta = 1/3 \), see above; the latter called “adotrimlebeta”) with some mixture model-based clustering methods that estimate the number of clusters using the BIC. More precisely, we use the R-package mclust for fitting a Gaussian mixture with or without noise component (“gmixbic”, “gmixnoisebic”; default settings, the noise component is initialised by the R-function NNClean in package prabclus with parameter \( \text{nnk}=5 \), Byers and Raftery (1998)). We use the R-package EMNMIXskew for fitting mixtures of t- and skew t-distributions (Wang et al. (2009); Lee and McLachlan (2013); “tmixbix”, “skewtmixbic”). We use fully flexible covariance matrices and degrees of freedom if possible, but sometimes EMNMIXskew does not deliver a solution with the default settings, in which case we try out more constrained covariance matrix models as offered by EMNMIXskew until a valid solution is found, which in the simulations ultimately always was the case. 100 datasets have been generated from each DGP.

![Figure 3: First two dimensions of data from simulated DGP 1, generated from a mixture of three multivariate Gaussian distributions.](image)

We simulated data from three data generating processes (DGPs). The
first one (DGP 1) is supposedly easy in the sense that the Gaussian mixture model assumption is fulfilled except of one added outlier. This causes surprising difficulties for all methods, though. There are \( n = 2000 \) observations in \( p = 20 \) dimensions. There is a Gaussian mixture with three components in the first two dimensions, see Figure 3, therefore \( G = 3 \). The other 18 dimensions are just standard Gaussian, except that one observation in the third variable is replaced by the value 1000. Without the artificial outlier, this has been used as “noiseless.3h” in Coretto and Hennig (2016); a precise definition is given in the supplement of that paper. Note that we had EMMIXskew fit mixtures of t-distributions here, whereas mixtures of skew t-distributions were used for DGP 2 and DGP 3. From limited experiments, the respective other choice would not have improved results, and would have performed mostly similarly.

Figure 4: Left side: Distribution of numbers of clusters by method for DGP 1 (true \( G = 3 \)) over 100 simulation runs. Right side: Corresponding distribution of adjusted Rand index values; connecting lines indicate results on the same simulated dataset.

The results for DGP 1 are shown in Figure 4. We consider the chosen number of clusters and the adjusted Rand index comparing the resulting clustering (including the noise component in case of the OTRIMLE-methods and gmixnoisebic) with the true clustering (ARI; Hubert and Arabie (1985)). This becomes 1 for perfect correspondence, and 0 is its expected value for comparing two random clusterings. tmixbic does a perfect job regarding the number of clusters, always estimating \( G = 3 \). However, it is important to understand that this is quite useless if the resulting clustering is not good.

18
In fact, tmixbic mostly joins two of the true clusters and uses the third component as a widely spread component to unite the outlier with a handful of other slightly atypical observations. This results in the lowest average ARI-values, see Figure 1 and Table 1. adotrimle and adotrimlebeta perform similarly to each other. They find \( G = 3 \) clusters in 68 of 100 cases and achieve an ARI larger than 0.9 in all of these cases, which the other methods almost never reach. gmixnoisebic could have been expected to perform better, given that its model assumptions are pretty precisely fulfilled; the one outlier should be fitted by the noise component. The problem seems to be, as well as with gmixbic, that 18 standard normal variables dominate the two variables with cluster patterns when it comes to automatically deciding the covariance matrix model, as \texttt{mclust} does. The true structure is only fitted (out of the \texttt{mclust}-options) by fully flexible covariance matrices, but the 18 standard normal variables apparently make the method guess a more constrained and therefore wrong model. In turn, the number of clusters is overestimated. 6 datasets are fitted by adotrimle and adotrimlebeta with only with a single cluster, leading to an ARI of 0.

DGP 2 was designed to deviate from the model assumptions in a way that does not make the clusters look strikingly different from Gaussian ones, but with some heavier tails. Again \( n = 2000, p = 20 \); see the supplement of \cite{CorettoHennig2016} for full details. Again the clustering structure is present only in the first two variables, but these are now \( t_3 \)-distributed; variable 3-20 are again standard Gaussian; outliers as occasionally generated by \( t_3 \)-distributions are now in the same variables that also have the clustering structure, as opposed to DGP 1. Figure 1 shows the first two variables generated by this DGP. Results are shown in Figure 5 and Table 1. Somewhat surprisingly, these results are almost identical to those of DGP 1. The OTRIMLE-based methods estimate \( G = 3 \) correctly for 69 datasets, and get the clustering almost completely right in these cases, which does not hold for any of the other methods. tmixbic was replaced here by skewtmixbic, but the result is the same as before, also compared to the other methods. skewtmixbic estimates \( G = 3 \) mostly, but its fit is in most cases quite a bit worse than that of the other methods, so that the good estimation of \( G \) is rather a coincidence than an achievement. gmixbic and gmixnoisebic choose slightly more clusters than before; on top of the covariance constraint problem, more Gaussians are often needed to fit a non-Gaussian distribution such as the \( t_3 \). This is a bit less of a problem for gmixnoisebic, because it can assign observations in the fringes of the \( t_3 \) to noise before fitting it with one or more Gaussians.

DGP 3 with \( n = 660, p = 6 \) brings together different shapes of distributions in the same dataset, as is the case in some real applications. Cluster
Table 1: Average adjusted Rand index values over 100 simulation runs. The last line gives the values for DGP 3 excluding the observations that were classified as noise.

| Method          | adotrimle | adotrimlebeta | gmixnoisebic | gmixbic | tmixbic |
|-----------------|-----------|---------------|--------------|---------|---------|
| DGP 1           | 0.823     | 0.824         | 0.637        | 0.638   | 0.564   |
| DGP 2           | 0.833     | 0.833         | 0.646        | 0.654   | 0.563   |
| DGP 3           | 0.709     | 0.830         | 0.700        | 0.673   | 0.835   |
| DGP 3 w/o noise | 0.897     | 0.856         | 0.716        | 0.673   | 0.835   |

Figure 5: Left side: Distribution of numbers of clusters by method for DGP 2 (true $G = 3$) over 100 simulation runs. Right side: Corresponding distribution of adjusted Rand index values; connecting lines indicate results on the same simulated dataset.
structure occurs on the first four variables, the fifth variable is standard Gaussian, the sixth is $t_2$, generating some outliers. There are two Gaussian clusters with sizes 250 and 150, an independent product of exponential variables with 70 observations, a shifted multivariate $t_2$-distribution with 70 observations, and a tight uniform with 100 observations, therefore $G = 5$. There are 20 “true” noise points, 10 of which are generated by a wide uniform distribution and 10 by a wider spread $t_2$, see Figure 6. This was taken from \cite{Hennig2007}, where details are given. Only the uniform cluster was added, centered at $(2, 0, 4, 4)$ with range 0.4 on the first four variables.

Figure 6: Data simulated from DGP 3 with true clustering. “N” denotes noise, half of which was generated by a uniform and half by a $t_3$, see \cite{Hennig2007}.

For the results see Table 1 and Figure 8. The skew t-mixture does the best job regarding choosing $\hat{G}$ and also regarding the plain ARI. adotrimele and adotrimelebeta have a tendency to underestimate the number of clusters. This can mainly be explained by the fact that the strongly asymmetric exponential cluster is not well represented by a mode at the mean, and therefore the $Q$-criterion will prefer solutions that classify this as noise. This is not a proper cluster in the sense defined by $Q$ (asymmetric versions of $Q$ are conceivable) and should arguably not be counted when operating with a symmetric prototype idea of a cluster. We also give ARI-results not involving the observations classified as noise in Table 1 and Figure 8, and regarding
Figure 7: Distribution of numbers of clusters by method for DGP 3 (true $G = 5$) over 100 simulation runs.

Figure 8: Distribution of adjusted Rand index values for DGP 3 over 100 simulation runs. Left side: Estimated noise is included. Right side: Estimated noise is not included. Connecting lines indicate results on the same simulated dataset.
these adotrimle and adotrimlebeta beat the skew t-mixture; one can argue that classifying observations as noise expresses an uncertainty about these observations, and the ARI leaving out these observations expresses whether observations that are classified are actually well classified. Regarding the number of clusters and raw ARI, adotrimlebeta with $\beta = 1/3$ is clearly better than adotrimle. The latter is better when estimated noise is discounted, but this is largely due to the larger estimated noise proportion. gmixbic and gmixnoisebic try to fit non-Gaussian clusters with more than one Gaussian component, and overestimate $G$ in this way. They are much worse than the other methods. In none of the DGPs, gmixnoisebic is clearly better than gmixbic, despite the presence of outliers. This is different from the simulations with fixed $G$ in Coretto and Hennig (2016).

5.2 Single cell RNA sequencing data

The first real dataset is from Biase et al. (2014) and represents single cell RNA sequencing data. Clustering of such data can be valuable for definition or discovery of new cell types, and for information reduction in further analyses. The data here regard early embryonic development. The data have 49 observations and 25737 genes, which are normalised and reduced to principal components here, following Lun et al. (2016). 3 PCs are used, representing more than 70% of the overall variance. The last big drop of the contribution of a single PC to the overall variation (4%) occurs between 3 and 4 PCS. Figure 9 shows a pairs plot of the data.

Figure 9: Pairs plot of single cell RNA sequencing data with three given true cell types.
Figure 10: adotrimle results for single cell RNA sequencing data. Left side: Density-based clustering quality criterion $Q(G)$ for the different numbers of clusters. The connected lines refer to the clustering of the original dataset, the circles to the clustering of bootstrapped datasets. The red “X” denotes the cutoff point for a clustering to be adequate. The graph has been truncated at $Q = 40$, but in fact bootstrapped results go up to about 100 for $G = 1$ and $G = 2$. Right side: Noise proportions, and ordering of numbers of clusters according to $S(G)$.

Figure 10 shows plots that illustrate the result. On the left side, $Q(G) = 0$ for $G = 2, 3, 4,$ and 6 for the original data, and also for many bootstrapped datasets. In fact, although this is hard to see, only the solutions for $G = 5$ and 7 are not adequate. On the left side the estimated noise proportion is shown, and derived from it the order of values $G$ according to $S(G)$. At $G = 3$, $\hat{\pi}_0 = 0$, but $\hat{\pi}_0$ is very large for $G = 1, 2$, so that $S(G)$ is minimised at $G = 3$. This solution is also adequate, and therefore chosen as the number of clusters (adotrimlebeta yields the same result). The resulting clustering is identical to the true one, the ARI is 1.

This clustering problems is not as easy as it seems. gmixbic estimates $G = 4$ (one of the true clusters does not look Gaussian enough); gmixnoisebic chooses $G = 3$ correctly, but classifies 7 observations as noise. EmSkew runs into computational trouble, probably from trying to fit a cluster that is too small for the required number of parameters, and only delivers results for $G = 1,$ and 2, both for multivariate t- and skew t-mixtures.
5.3 Simulation results data

We decided to use the results data from our own simulation study shown as a parallel coordinates plot on the right side of Figure 3 in order to illustrate the application of the methods to a dataset of real research interest without a given number of clusters. The plot shows that there seems to be clustering structure, but also outliers. The observations here are the 100 generated datasets for the simulation runs, and the variables are the ARI-values for the five clustering methods. The interest in such a clustering could be motivated by trying to pin down characteristics of different datasets that can lead to substantially different results. We focus here on the clustering itself.

Figure 11 illustrates the adotrimle results. On the left side it can be seen that all numbers of clusters except $G = 3$ are adequate. The right side shows that $G = 1, 2,$ and $3$ have rather large estimated noise proportions, so that $G = 4$ minimises $S(G)$. Figure 12 shows the two-dimensional scatterplots that distinguish the clusters. Cluster 4 is most characteristic, collecting the datasets for which adotrimle and adotrimlebeta showed a mediocre performance with ARI-values around or even smaller than 0.6. Cluster 1 has datasets on which adotrimle and adotrimlebeta both performed flawlessly.
This is also very homogeneous regarding gmixbic and gmixnoisebic; all of these datasets have an ARI around 0.75 for those two methods. There are more datasets for which adotrimle and adotrimlebeta performed well, and these form clusters 2 and 3. Cluster 3 contains datasets for which gmixnoisebic performed better (ARI around 0.75) than gmixbic (ARI around 0.5). Cluster 2 mops up the remaining datasets for which both adotrimle and adotrimlebeta performed well, leading to a somewhat larger within-cluster variation. There are 8 “noise” datasets, among those all 6 for which adotrimle and adotrimlebeta ended up with ARI= 0. This is a well interpretable solution that looks convincing on the data.

Figure 12: Some scatterplots of simulation results data with clustering by adotrimle (“N” denotes observations classified as noise).

gmixbic produces another solution with $G = 4$, which inappropriately joins adotrimle’s noise and cluster 4, and some additional points from the data as a single cluster, which seems inappropriate. gmixnoisebic prefers a solution with $G = 7$ and several noise points, which seems too complex for the potentially useful information in the data as can be assessed from plots. tmixbic picks a solution indicated as $G = 3$, which has one empty cluster, so it is in effect a $G = 2$-solution, and it puts everything together in the same cluster except the datasets with high ARI by adotrimle and adotrimlebeta. skewtmixbic even estimates $G = 1$.

In order to explore the stability of the bootstrap scheme to assess the adequacy of the clusterings, adotrimle analyses on both real data sets were repeated three times each with $B = 100$ and $B = 30$. The optimal solutions were always the same. Looking at the adequacy assessments for every single number of clusters, the only difference was that for the simulation results data the presented solution has for $G = 3$: $\frac{Q^{(G)}-m_{QG}}{\overline{QG}} = 2.36$, whereas in the repeated analyses this was smaller than 2 in all but one case. This does not affect the final solution, because the $G = 3$-clustering has too much noise
anyway. However it shows that standardised values not substantially larger than 2 deserve some attention.

6 Conclusion

The problem of choosing the number of clusters is very difficult, particularly in applications in which observations occur that do not belong to any cluster. It is often treated as an estimation problem regarding the true number of mixture components in a parametric mixture distribution, e.g., a Gaussian mixture, but this is inappropriate, because as estimation problem it is ill-posed. Even slight violations of the model assumptions will for large enough $n$ lead to an estimate of the number of clusters that is larger than the number of data subsets that could reasonably be interpreted as clusters, because not precisely Gaussian subsets will eventually be fitted by more than one Gaussian mixture component.

An appropriate decision rule for the number of clusters in a Gaussian mixture context involves a decision about what kind of non-Gaussian data subset still qualifies as a cluster. This is formalised by our clustering quality statistic $Q$. The observed value of $Q$ is compared to what is expected if data are indeed generated by a Gaussian mixture with the estimated parameter values. If an underlying distribution of a cluster has a tendency to produce better clusters than a Gaussian according to $Q$ (which is the case for distributions such as the t-distribution, for which the density goes down faster from the mean than for the Gaussian), the procedure will accept such clusters. Some users may be willing to accept certain potentially unimodal clusters even though they look somewhat worse than what is expected from the Gaussian. This could be achieved by changing the cutoff value $c$ for adequacy to something larger, say from 2 to 3 or 4. However this would allow for clusters that look less unimodal. Another possible modification is to re-define $Q$ in order to allow for asymmetric clusters; density values could be re-ordered as decreasing from the observed mode rather than the mean without imposing symmetry by arranging a pair of the same density values to the left and to the right. In that case one may however wonder whether it makes sense to start with a Gaussian mixture in the first place. It is ultimately up to the user to decide what kind of clusters are required in a given application. Without such decisions, the data on their own do not provide sufficient information about the clustering structure required to fit them; there are severe identifiability problems when choosing a mixture model.

The general adequacy approach presented here can be used for choosing
the number of clusters for other clustering methods, as long as a model is
given that formalises a prototype clustering structure of interest to which
parametric bootstrap can be applied. Other concepts of admissible clusters
can be formalised by design of the clustering quality statistic $Q$. This will
be future work.

The approach as presented here along with the accompanying plots shown
in Sections 5.2 and 5.3 will be implemented in the R-package otrimle by the
time of the appearance of the accepted version of this paper in the journal.

References

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

Biase, F. H., X. Cao, and S. Zhong (2014). Cell fate inclination within 2-cell
and 4-cell mouse embryos revealed by single-cell rna sequencing. *Genome
research* 24, 1787–1796.

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, 577–584.

Cerioli, A., L. A. Garca-Escudero, A. Mayo-Iscar, and M. Riani (2018). Finding
the number of normal groups in model-based clustering via constrained
likelihoods. *Journal of Computational and Graphical Statistics* 27(2), 404–
416.

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

Coretto, P. and C. Hennig (2016). Robust improper maximum likelihood:
tuning, computation, and a comparison with other methods for robust
Gaussian clustering. *Journal of the American Statistical Association* 111,
1648–1659.

Coretto, P. and C. Hennig (2017). Consistency, breakdown robustness, and
algorithms for robust improper maximum likelihood clustering. *Journal of
Machine Learning Research* 18, 1–39.

Coretto, P. and C. Hennig (2019). otrimle: Robust model-based
clustering. R package version 1.3. Available at: https://CRAN.R-
project.org/package=otrimle.
Davies, P. L. (1995). Data features. *Statistica Neerlandica* 49(2), 185–245.

Davies, P. L. and A. Kovac (2001). Local extremes, runs, strings and multi-resolution. *Ann. Statist.* 29(1), 1–65.

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

Garca-Escudero, L. A., A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar (2018). Eigenvalues and constraints in mixture modeling: geometric and computational issues. *Advances in Data Analysis and Classification* 12, 203–233.

Gelman, A. and C. Hennig (2017). Beyond objective and subjective in statistics (with discussion). *Journal of the Royal Statistical Society, Series A* 180, 967–1033.

Hennig, C. (2004). Breakdown points for maximum likelihood estimators of location-scale mixtures. *Annals of Statistics* 32, 1313–1340.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis* 52, 258–271.

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

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification* 2(2), 193–218.

Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhya: The Indian Journal of Statistics, Series A* (1961-2002) 62(1), 49–66.

Lee, S. X. and G. J. McLachlan (2013). On mixtures of skew normal and skew t-distributions. *Advances in Data Analysis and Classification* 7, 241–266.

Lun, A. T., D. J. McCarthy, and J. C. Marioni (2016). A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Research* 5:2021.

Maronna, R. A. and R. H. Zamar (2002). Robust estimates of location and dispersion of high-dimensional datasets. *Technometrics* 44, 307–317.
Nguyen, T. T., H. D. Nguyen, F. Chamroukhi, and G. J. McLachlan (2020). Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics* 7.

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

Pons, O. (2013). *Statistical Tests of Nonparametric Hypotheses: Asymptotic Theory*. World Scientific, Singapore.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal* 8(1), 289–317.

Wang, K., S. Ng, and G. J. McLachlan (2009). Multivariate skew $t$ mixture models: Applications to fluorescence-activated cell sorting data. In 2009 *Digital Image Computing: Techniques and Applications*, pp. 526–531.