Robust Constrained Clustering in Presence of Entry-wise Outliers

Alessio FARCOMENI

Department of Public Health and Infectious Diseases
Sapienza - University of Rome
Rome, Italy
(*alessio.farcomeni@uniroma1.it*)

We propose a robust heteroscedastic model-based clustering method based on snipping. An observation is snipped when some of its dimensions are discarded, but the remaining are used for estimation. An expectation-maximization algorithm augmented with a stochastic optimization step is used to derive inference, and its convergence properties are studied. We show global robustness of our resulting `sclust` procedure also when outliers arise entry-wise. The method is robust to contamination, even when most or even all of the observations contain outliers. Simulations and two real data applications illustrate and compare the approach with existing methods.

KEY WORDS: Component-wise contamination; Model-based clustering; Snipping; Trimming.

1. INTRODUCTION

The definition of robust approaches for moderate- to high-dimensional data is challenging as departure from model assumptions can be much more complex than in small-dimensional settings. In small dimensions, there may be no practical difference between an observation that is entirely contaminated, arising therefore from a multivariate distribution far from the bulk of data (Tukey 1962; Huber 1964), and an observation containing few (maybe one or two) contaminated entries. To this end, Alqallaf et al. (2009) proposed a contaminating scheme, the *component-wise contamination model* (CWC), in which each dimension is (dependently or independently) contaminated with a certain probability. Under CWC and in high dimensions, most (if not all) of the observations may be contaminated, often only in one or two entries. Component-wise outliers can be thought of measurement rather than sampling errors, and it is not difficult to think about many situations in which CWC may happen. The focus of this article is on robustness in model-based clustering (Fraley and Raftery 2002). Some approaches to this are based on using flexible components in a finite mixture, like Banfield and Raftery (1993), Fraley and Raftery (1998), McLachlan and Peel (2000), and Frühwirth-Schnatter and Pyne (2010). Other approaches are based on robustifying the estimation procedure, like Cambpell (1984) and Markatou (2000). This article is in the spirit of a third class of approaches, which is based on trimming (Gallegos and Ritter 2005, Neykov et al. 2007, García-Escudero et al. 2008, Gallegos and Ritter 2009a, 2010), that is, simultaneous clustering and zero weighting of contaminated observations. For general reviews of robust statistics, we refer the reader to the books by Huber and Ronchetti (2009), Heritier et al. (2009), and to the articles by Hubert, Rousseeuw, and Van Aelst (2008) and Farcomeni and Ventura (2012). In principle, all these methods can be applied to the case of CWC. In practice, there are at least four reasons why they may not be appropriate. First, a single contaminated entry may be enough to misclassify an otherwise clean observation, which may either be downweighted, trimmed, or assigned to an inappropriate mixture component according to the chosen approach. Second, treating clean entries in the same row of a contaminated one as contaminated may be inefficient and even lead to bias. Third, in many applications it may be desirable to cluster all observations, or at least as many as possible. Finally, when the proportion of observations with at least one contaminated entry is large, trimming procedures may not be feasible. A recently proposed approach to tackle isolated contaminated entries in cluster analysis is that of *snipping* (Farcomeni 2013). An observation is snipped when one or more of its dimensions are excluded from the analysis. Snipped observations for which one or more dimensions are retained can be assigned to clusters. Snipping is a simple and effective way of tackling the aforementioned limitations: contaminated entries are discarded but, based on the remaining clean entries, the observation is assigned to a cluster. Potentially all observations can be snipped. Note that snipping is more flexible than trimming as removal of all entries of an observation, which is possible, corresponds to trimming. A snipped k-means procedure is proposed in Farcomeni (2013). In this article, we define and solve a snipped model-based clustering problem. Estimates in our framework cannot be obtained via a direct optimization as in Farcomeni (2013). We set up an expectation-maximization (EM) type algorithm (Dempster, Laird, and Rubin 1977), which incorporates a constrained stochastic optimization step and a constrained maximization step. The former is used to obtain the optimal feasible snipping configuration, while the latter to obtain locally optimal parameter estimates avoiding spurious solutions.

The rest of the article is as follows: in the next section, we set up the appropriate spurious outliers model and outline assumptions and constraints to make the problem well defined. In Section 3, we obtain the likelihood and describe a novel EM

TECHNOMETRICS, FEBRUARY 2014, VOL. 56, NO. 1
DOI: 10.1080/00401706.2013.826148

algorithm with stochastic optimization to maximize it. Convergence of the algorithm and robustness properties of the resulting estimates are studied in Section 4. In Section 5, we illustrate and compare our proposal, and give some conclusions in Section 6.

2. A MIXTURE MODEL WITH ISOLATED OUTLIERS

We assume we have a sample of n observations $X_1, \ldots, X_n$, with $X_i \in \mathcal{R}^d$, which are divided in k groups indexed in $\mathcal{J}_1, \ldots, \mathcal{J}_k$, where $\mathcal{J} = \cup_{c=1}^k \mathcal{J}_c \subseteq \{1, \ldots, n\}$ and $\mathcal{J}_j \cap \mathcal{J}_h = \emptyset$. We assume that $nd\varepsilon$ entries of the data matrix X are contaminated according to the model we specify below, for some $\varepsilon \geq 0$. We also assume the cardinality of $\mathcal{J}$, denoted $|\mathcal{J}|$ to be such that $n(1-\varepsilon) \leq |\mathcal{J}| \leq n$. Note that when $|\mathcal{J}| = \lfloor n(1-\varepsilon) \rfloor$, we have that all contamination arises from structural outliers, while when $|\mathcal{J}| = n$ we only have component-wise contamination arranged so that no observation is completely spoiled. Let now V be an $n \times d$ binary matrix, with a number of nonzero elements equal to the closest integer to $nd(1-\varepsilon)$. We have that V_{ij} is an indicator of the ijth entry of the data matrix being free of contamination, while when $V_{ij} = 0$ the jth column of the ith observation has been contaminated. Let furthermore w denote an nxk binary matrix, with $w_{ic} = 1$ being an indicator of $i \in \mathcal{J}_c$ and $\sum_c w_{ic} \leq 1$. When $\sum_j V_{ij} = 0$, $i \notin \mathcal{J}$ and $\sum_c w_{ic} = 0$. Our general contamination model is

$$\begin{cases} X_i = V_i Y_i + (1 - V_i) Z_i & i \in \mathcal{J} \\ X_i = Z_i & i \notin \mathcal{J} \end{cases} \tag{1}$$

for $i = 1, \ldots, n$, where V_i denotes the ith row of matrix V, $Z_i \sim g_i(\cdot)$, with $g_i(\cdot)$ being a density in $\mathcal{R}^d$, and the products are component-wise. The clustering assumption is that Y_i is generated from one of k multivariate normal distributions, with first moment μ_c and covariance Σ_c, for $c = 1, \ldots, k$. We assume that the first moments are not equal, that is, $||\mu_c - \mu_d|| > 0$ when $c \neq d$. As common in the model-based clustering literature, the clustering indicator matrix w is treated as a latent variable, with $\Pr(w_{ic} = 1 | i \in \mathcal{J}) = \pi_c$, $\pi_c > 0$ for $c = 1, \ldots, k$; $\sum_{c=1}^k \pi_c = 1$. We also assume that V is a parameter, as common in the robust model-based clustering literature (e.g., Fritz, García-Escudero, and Mayo-Iscar 2013).

We have that (1) generalizes the spurious outliers model of Gallegos and Ritter (2005), which is obtained with the additional constraint that $(\sum_j V_{ij}) \sum_j (1 - V_{ij}) = 0$ for all $i = 1, \ldots, n$. The likelihood corresponding to model (1) can be computed as

$$\prod_i \sum_{c=1}^k \pi_c \phi_{\sum V_i}(x_{i(V_i)}, \mu_{c(V_i)}, \Sigma_{c(V_i)}) g_{i(1-V_i)}(x_{i(1-V_i)}). \tag{2}$$

In (2), multivariate densities of varying dimensionality are involved. We use the notation $\phi_{\sum V_i}$ to denote the density of a multivariate normal of dimensionality $\sum_j V_{ij}$. When $\sum_j V_{ij} = 1$, ϕ_1 denotes the density of a univariate Gaussian distribution; and we adopt the convention that $\phi_0(\cdot) = 1$. Similarly, with $g_{i(1-V_i)}(\cdot)$, we denote a multivariate density restricted to the zero dimensions of V_i; and $g_{i0}(\cdot) = 1$. Further, with $\mu_{c(V_i)}$ we denote the entries of μ_c that correspond to a nonzero element in the vector V_i, the same applies to $\Sigma_{c(V_i)}$. This approach corresponds to a situation in which elements corresponding to a zero entry in V are missing at random, and are therefore ignored for estimation of μ and Σ. The problem is well defined as long as two conditions are satisfied. First, we need at least one clean observation for each dimension and cluster. This reduces to the constraint

$$\sum_i V_{ij} w_{ic} > 0 \quad \forall j = 1, \ldots, d; c = 1, \ldots, k. \tag{3}$$

In case, (3) is violated for a certain $\bar{j}$ and $\bar{c}$, the likelihood (2) is constant with respect to $\mu_{\bar{c}\bar{j}} \in \mathcal{R}$. A similar condition shall be used for Σ_c as follows:

$$\sum_i V_{ij} V_{ih} w_{ic} > 0 \quad \forall j, h = 1, \ldots, d; c = 1, \ldots, k. \tag{4}$$

This second condition guarantees that for each cluster there exists at least one observation with clean entries in each pair of dimensions. Note that (4) implies (3), so the latter can be ignored in practice. A final assumption regards the contaminating distributions $g_i(\cdot)$. To be able to ignore the nonregular entries in (2), we assume that contaminated entries are generated far enough from the clean model and are affine independent. Similar assumptions are used in Gallegos and Ritter (2005) and García-Escudero et al. (2008). In our case, the classical separation condition on $g_i(\cdot)$ becomes

$$\sup_V \sup_{\mu,\Sigma,\pi} \prod_i \sum_{c=1}^k \pi_c \phi_{\sum V_i}(x_{i(V_i)}, \mu_{c(V_i)}, \Sigma_{c(V_i)}) \geq \sup_V \prod_i g_{i(1-V_i)}(x_{i(1-V_i)}). \tag{5}$$

In words, identification of clean entries by maximization of the part of the likelihood corresponding to the Gaussian mixture identifies the same entries as would identification of contaminated entries by maximizing the part of the likelihood corresponding to the noise g_i. As a consequence, once clean entries are identified by maximizing the left-hand side of (5), the contaminated entries are consequently identified optimally. Given that it is completely arbitrary, the estimator of $g_i(1 - V_i)$ is nothing but the Dirac delta in $x_{i(1-V_i)}$. As a consequence, at the maximum likelihood estimator (MLE) all the likelihood contributions of the contaminated entries are equal to the unity in (2).

Finally, a known caveat is that the resulting finite mixture model shall not be estimated without constraints, as spurious solutions may otherwise be often obtained. One possible solution, as proposed by García-Escudero et al. (2008) is to fix λ such that

$$\frac{\max_j \lambda_1(\Sigma_j)}{\min_j \lambda_d(\Sigma_j)} \leq \lambda, \tag{6}$$

for some $\lambda \geq 1$, where $\lambda_1(A)$ and $\lambda_d(A)$ denote the largest and smallest eigenvalue of A. A default choice is $\lambda = 12$. There are few other possible constraints, see, for instance, Gallegos and Ritter (2009a, b, 2010). These are based on either constraining the determinant of the covariance matrices, or on constraining the cardinality of clusters. In our implementation we will use (6), but any other constraint can be readily incorporated. Choice of the most suitable constraint is beyond the scope of this work.

3. THE sclust ALGORITHM FOR MODEL ESTIMATION

Given the assumptions on contamination, the likelihood will be maximized by the MLE for μ, Σ, π, and V; and by point masses for g_i for what concerns contaminated entries. In this sense, we can ignore the terms in which g_i is involved in (2), and obtain a profile likelihood for the remaining parameters as

$$l(\theta) = \prod_i \sum_{c=1}^{k} \pi_c \phi_{\sum V_i}(x_{i(V_i)}, \mu_{c(V_i)}, \Sigma_{c(V_i)}).$$

Maximization of the expression above is anyway cumbersome, and as it is common in finite mixture models we set up an EM type algorithm. The EM algorithm is based on the complete log-likelihood

$$l_c(\theta) = \sum_{c=1}^{k} \sum_{i=1}^{n} \log(\pi_c \phi_{\sum V_i}(x_{i(V_i)}, \mu_{c(V_i)}, \Sigma_{c(V_i)})) w_{ic}, \quad (7)$$

where θ is a short-hand notation for the unknowns involved. Maximization must be carried out under the restrictions that $\sum_{ij} V_{ij} \geq nd(1-\varepsilon)$ and $\sum_i w_{ic} = 0$ when $\sum_j V_{ij} = 0$. Additionally, we shall impose (6) for a prefixed $\lambda \geq 1$. A simple approach to maximization of (7) would basically be based on alternating three steps: an E step, in which the indicators w_{ic} are substituted with their conditional expected values $\tilde{w}_{ic}$; an S (snipping) step in which we maximize the conditional expected value of (7), evaluated at the current value of μ, Σ, and π, with respect to V. Finally, a constrained M step used to maximize the conditional expected value of (7), evaluated at the current V, with respect to π, μ, and Σ. We will propose a better solution below, but to fix the ideas we start describing an algorithm along these lines.

As far as the E step is concerned, it is straightforward to see that for fixed V, π, μ, and Σ, the conditional expected value of w_{ic} is given by

$$\begin{aligned} &\Pr(w_{ic} = 1 | X, V, \mu, \Sigma, \pi) \\ &\quad = \tilde{w}_{ic} \propto \pi_c \phi_{\sum V_i}(x_{i(V_i)}, \mu_{c(V_i)}, \Sigma_{c(V_i)}), \end{aligned} \quad (8)$$

for all i such that $\sum_j V_{ij} > 0$; while $\tilde{w}_{ic} = 0$ otherwise. While we give a detailed description of the snipping step below, we can here say that at the M step we update the remaining parameters as follows. The cluster weights are updated as $\pi_c \propto \sum_i \tilde{w}_{ic}$, while for $c = 1, \ldots, k$ and $j = 1, \ldots, d$ we have that

$$\mu_{cj} = \frac{\sum_i x_{ij} V_{ij} \tilde{w}_{ic}}{\sum_i V_{ij} \tilde{w}_{ic}}. \quad (9)$$

An initial estimate of Σ_c is obtained by fixing

$$\Sigma_c = \frac{\sum_i \tilde{w}_{ic} (V_i(X_i - \mu_c))(V_i(X_i - \mu_c))'}{\sum_i \tilde{w}_{ic} V_i V_i'}. \quad (10)$$

The initial estimate of Σ_c is then modified, if needed, to satisfy the constraint (6). To this end, $2kd + 1$ function evaluations suffice to impose the eigenvalue constraint, exactly as in the tclust algorithm of Fritz, García-Escudero, and Mayo-Iscar (2013).

We can now describe that the S step that in our final formulation of the resulting sclust algorithm used to maximize (7) is embedded within the E step. Consequently, $\tilde{w}$ and V are simultaneously updated. It is of course possible to perform the E and the S steps separately, but we have found that the resulting algorithm, albeit convergent, is more easily trapped into local optima. The properties of the proposed algorithm will formally be studied in the next section. The ES step proceeds with a simple stochastic optimization algorithm (Chakraborty and Chaudhury 2008; Farcomeni 2013), see also Farcomeni and Viviani (2011): a sequence of random proposals $(\tilde{w}_c(t), V_c(t))$, $t = 1, \ldots, T$ is built by switching a 0 and a 1 entry uniformly at random in $V_c(t-1)$. Obviously, $V_c(0) = V$, the current optimal solution. For each $V_c(t)$, $\tilde{w}_c(t)$ is computed through (8), as outlined above. The random proposal $V_c(t)$ is accepted with probability

$$p_t = \min\left(e^{-\frac{\log(t+1)}{D}(l(V_c(t-1)) - l(V_c(t)))}, 1\right). \quad (11)$$

If $V_c(t)$ is not accepted, we let $V_c(t) = V_c(t-1)$. After T steps we set the new $\tilde{w}$ and V equal to the $V_c(t)$ and $\tilde{w}_c(t)$ corresponding to the largest likelihood obtained. This finishes the ES step. At the M step, we estimate π_c, μ_c, and Σ_c, $c = 1, \ldots, k$ as outlined above. There are two tuning parameters for our sclust algorithm, T and D. The latter is used in (11) to control the acceptance probability. A large D will lead to a greedy stochastic optimizer, while a small D will allow some candidates leading to smaller likelihood to be accepted, so to escape local optima for V. It is generally recommended (Chakraborty and Chaudhury 2008; Farcomeni 2013) that D is set approximately equal to the maximal change in the objective function that can be seen when switching two clean entries. Few pilot runs can be used to estimate the latter quantity and fine tune D. The other tuning parameter, T, shall be set large enough so that the stochastic optimizer is allowed to improve the current V. We have actually found that often even as much as $T = 10$ is large enough, which is not surprising given that the stochastic optimizer is repeated at each iteration. We have also found that when the starting solution is not close to the optimum, it may be better to break the optimizer as soon as an accepted proposal $V_c(t)$ gives a larger likelihood. For the actual tuning of T used in this article, see below.

It can be noted that the computational complexity of each iteration is linear, that is, each iteration has a complexity of order $kndT$. Nevertheless, the number of iterations needed before convergence is dependent on the initial solution, the separation of the clusters within the data, and other issues.

After convergence of the algorithm, observations may be assigned to clusters based for instance on a maximum-a-posteriori criterion applied to $\tilde{w}_{ic}$ as obtained in the final E step.

A final issue regards practical choice of the snipping level ε and number of groups k. While a careful choice of k is often very important, in our experience, tuning of ε is not as crucial. A slightly large ε will have negligible consequences on the estimates and on quality of clustering. Snipping some clean entries will not even have deleterious effects on clustering: units completely free of contamination will not be trimmed entirely and therefore will be assigned to the closest cluster even after snipping. On the other hand, a slightly small ε will not in most cases affect the quality of the estimates as gross influential outliers are usually the first to be flagged.

To choose k and ε, one can proceed in practice by comparing stability of the results. Formal suggestions along these lines are the G statistic of Farcomeni (2009) and the trimmed curves of García-Escudero et al. (2011). The G statistic can be used to select ε. It is a distance function between the parameters estimated using a certain ε and those estimated using $\varepsilon = 0$. Formally, we propose to use

$$G(\varepsilon) = \sum_{jc} (\mu_{jc}(\varepsilon) - \mu_{jc}(0))^2, \tag{12}$$

where $\mu_{jc}(\varepsilon)$ denotes the centroid estimated using a snipping level of ε. Note that the clusters shall be identified to compute (12). To do this, we identify the clusters based on increasing values of the estimate for μ_{1c}. The idea behind (12) is that for fixed k, a substantial G statistic is expected only when contaminated entries are flagged. If there is no contamination, $G(\varepsilon)$ will be small for any $\varepsilon > 0$. A heuristic approach for choosing ε is given by comparing $G(\varepsilon)$ for a grid of values, and selecting the smallest ε for which $G(\varepsilon)$ is approximately constant. The trimmed likelihood curves of García-Escudero et al. (2011) can instead be used to choose k. These curves are based on plotting the likelihood obtained at convergence against ε, for different choices of k. The number of groups shall be set equal to the minimum k for which there is no substantial increase in the maximum of the likelihood when adding one extra group. Note that the trimmed likelihood curves can also be used for simultaneous selection of k and ε, if desired.

3.1 Additional Constraints on V

Different snipped clustering procedures can be obtained with simple modifications of the proposed EM. First, one can require that $|\mathcal{J}| = n(1-\varepsilon)$. In this case snipping reduces to trimming and therefore the proposed EM corresponds to a `tclust` algorithm with stochastic search for the trimmed observations (as opposed to the more commonly used concentration steps, see Fritz, García-Escudero, and Mayo-Iscar 2013). This is obtained by starting the algorithm with an appropriate V matrix, and building candidates by switching rows uniformly at random. At the opposite extreme, one can require that $\sum_j V_{ij} > 0$ for all i, so that $|\mathcal{J}| = n$ and no observation is entirely trimmed. This may be useful in applications where clustering *all* of the available observations is important, and is easily obtained by discarding any random proposal V_c that does not satisfy the requirement. Trimming and snipping can also be combined if desired, to guarantee that at least ε_1 observations are trimmed (while remaining ε_2 entries are snipped). The latter statement corresponds to a requirement that $|\mathcal{J}| \geq n(1-\varepsilon_1)$, while $\sum_{ij} V_{ij} = nd(1-\varepsilon)$ for some $\varepsilon > \varepsilon_1$. To do so, one can simply start with a V for which $n\varepsilon_1$ rows sum to zero, and build a candidate V_c based on switching two rows uniformly at random, and switching two entries uniformly at random only if the entries do not belong to a row summing to zero.

4. PROPERTIES

In this section, we study the theoretical properties of our `sclust` procedure. First, we evaluate convergence of the algorithm, then robustness of the estimates.

4.1 Existence and Consistency

First of all, it can be shown that the proposed `sclust` algorithm is convergent regardless of T. More formally

Theorem 1. Fix $T \geq 1$ and $D > 0$, and denote with θ_j the estimate of θ obtained at the jth step of the EM algorithm. We have that

$$l(\theta_{j_1}) \leq l(\theta_{j_2}) \quad \forall\ j_1 \leq j_2. \tag{13}$$

Furthermore, there exists $J \in \mathcal{N}$ such that $l(\theta_{j_1}) = l(\theta_{j_2})$ $\forall J \leq j_1 \leq j_2$, and that $\theta_{j_1} = \theta_{j_2} \quad \forall J \leq j_1 \leq j_2$.

Proof. We first show that the likelihood is nondecreasing in the sequence of solutions. The current parameter vector can be decomposed as follows: $\theta_j = (\pi^{(j)}, \mu^{(j)}, \Sigma^{(j)}, V^{(j)}, \tilde{w}^{(j)})$. The stochastic search is designed so that

$$l\big(\pi^{(j)}, \mu^{(j)}, \Sigma^{(j)}, V^{(j+1)}, \tilde{w}^{(j)}\big) \geq l(\theta_j).$$

For a given V, we maximize the conditional expected likelihood using (8), which by construction maximizes $l(\pi^{(j)}, \mu^{(j)}, \Sigma^{(j)}, V^{(j+1)}, w)$ with respect to w. Hence,

$$l\big(\pi^{(j)}, \mu^{(j)}, \Sigma^{(j)}, V^{(j+1)}, \tilde{w}^{(j+1)}\big) \geq l(\theta_j).$$

At this point of the iteration we use a classical M step that directly maximizes the conditional expected likelihood with respect to π, μ, and Σ. We now only need to show that the likelihood is bounded. The latter holds given that, under (6), the MLE is well defined conditionally on V, and regardless of V. □

Theorem 1, combined with results from Dempster, Laird, and Rubin (1977), shows that the proposed algorithm leads to a local optimum for the likelihood. We have to stress anyway that this local optimum is, due to the discrete nature of the parameter, conditional on V. That is to say, the final estimates will correspond to an improved solution, and the estimates for all parameters will be a local maximizer of the likelihood conditionally on the final V, which will be at least as good as that of the starting solution. Additionally, using results from Chakraborty and Chaudhury (2008), the following can be shown:

Theorem 2. Suppose $T \to \infty$ and $D > 0$. We have that `sclust` converges to a local optimum of the likelihood, and that the final V is the global optimizer of the likelihood conditionally on the estimates of the other parameters.

To obtain the same result of Theorem 2 in practice, it suffices to let T be large enough so that the conditional global optimum can be reached. In our implementation, we set it equal to a small value (say $T = 10$) for the first few iterations to explore the parameter space, and then use a large T (say $T = 1000, 10{,}000$, or even larger) after convergence with $T = 10$. As usual with EM type algorithms, to increase the likelihood of ending up in the global optimum we use a multistart approach. We build an initial deterministic starting solution integrating multivariate outlier identification methods with the `tclust` procedure; and then other starting solutions are obtained by randomly perturbing the result obtained at convergence from the deterministic starting solution.

4.2 Robustness

The breakdown point of an estimator is the largest fraction of the data that can be moved arbitrarily without affecting the estimator: thus the higher the breakdown point, the more robust the estimator against extreme outliers. For early ideas in global robustness, see Hodges (1967), Hampel (1971), and Donoho and Huber (1983). In the context of cluster analysis, Gallegos and Ritter (2005) defined the *individual breakdown point* as the minimal fraction of rows that must be replaced to be able to break out a parameter of interest θ. More formally, let X_R denote the data matrix in which R rows are replaced by arbitrary vectors in $\mathcal{R}^d$. If for instance the parameter of interest θ is the location vector μ, the individual breakdown point is defined as

$$\varepsilon^{(i)} = \frac{1}{n} \min \left\{ R : \sup_{X_R} ||\mu(X) - \mu(X_R)|| = \infty \right\}, \quad (14)$$

where $\mu(X)$ is the centroid estimate obtained from data X. Another definition that is more useful in our context is that of *cell breakdown point*, proposed by Farcomeni (2009), which is defined as the minimal fraction of entries of the data matrix that must be replaced before it is possible to spoil a parameter. Formally, let X_C denote the data matrix in which C elements are replaced by arbitrary values in $\mathcal{R}$. If the parameter of interest θ is the location vector μ, the cell breakdown point can be defined as

$$\varepsilon^{(c)} = \frac{1}{nd} \min \left\{ C : \sup_{X_C} ||\mu(X) - \mu(X_C)|| = \infty \right\}. \quad (15)$$

From (14) and (15), it is clear that estimates for π cannot break down. Appropriate definitions of breakdown points for bounded parameters can be found in Genton and Lucas (2003), but a study of breakdown of π is material for further work. Here we will focus on estimates of μ and Σ. To obtain meaningful results for μ, we restrict to a class of "well-clustered" clean datasets. This will provide the *restricted* breakdown points (Gallegos and Ritter 2005; Ruwet et al. 2012). It is fairly easy to show our main results if we adopt the same restrictions of Ruwet et al. (2012). This will also allow a comparison with trimmed clustering obtained with the `tclust` algorithm. The class of well-clustered datasets is defined in Ruwet et al. (2012) as those datasets with a natural cluster structure, that is, that can be separated in k groups. The class includes all datasets for which, after trimming ε subjects, there exists a partition into k groups whose centers have a Euclidean distance that is larger than or equal to a certain threshold. For more details and an account on how to check the conditions on the class of well-clustered datasets, we refer the reader to Ruwet et al. (2012). We start dealing with our procedure:

Theorem 3. Suppose $\hat{\mu}$ and $\hat{\Sigma}$ are obtained with the `sclust` procedure. Assume X is arbitrary and the snipping level is such that $\varepsilon \leq 0.5 - k(d+1)/2n$. The (universal) breakdown points for $\hat{\Sigma}$ are such that $\varepsilon^{(i)} = \varepsilon + k/n$ and $\varepsilon^{(c)} = \varepsilon + k/nd$.

Suppose now X is restricted to the class of well-clustered datasets defined in Ruwet et al. (2012), and the snipping level ε is such that

$$\varepsilon \leq \min \left(0.5 - \frac{k(k-1)d}{2n} + \frac{1}{2n}, \frac{1}{k} - \frac{1}{n} \right).$$

The (restricted) breakdown points for $\hat{\mu}$ are such that $\varepsilon^{(i)} \geq \varepsilon + 1/n$ and $\varepsilon^{(c)} \geq \varepsilon + 1/nd$.

Proof. Given that trimmed robust clustering is a special case of snipped robust clustering, using results from Ruwet et al. (2012), Propositions 1 and 3, we can directly obtain the results on the individual breakdown point. For what concerns the cell breakdown points, with component-wise contamination, we can have a minimum of $n\varepsilon$ and a maximum of $nd\varepsilon$ contaminated rows, when one entry per row is contaminated. The proof follows then from the same propositions in Ruwet et al. (2012), where we set the number of clean observations r to the smallest feasible value such that $r \geq n(1 - d\varepsilon)$. □

We now focus on other clustering procedures. For what concerns classical model-based clustering, it can be shown that even the restricted breakdown points are infinitesimal. For what concerns `tclust`, Ruwet et al. (2012) showed under the same conditions of Theorem 3 that the same bounds hold for the individual breakdown points as that obtained for the `sclust` procedure. In the following proposition, we study the cell breakdown points of `tclust`:

Theorem 4. Assume the same conditions of Theorem 3 hold. Then, for `tclust` at level ε we have for the location estimator $\varepsilon^{(c)} \geq \frac{\varepsilon}{d} + 1/nd$. For the scatter estimator we have that $\varepsilon^{(c)} = \frac{\varepsilon}{d} + 1/nd$.

Proof. Note that contamination of C entries corresponds to contamination of $C/d \leq R \leq C$ rows. Therefore, with a contamination proportion of ε entries we can spoil up to $d\varepsilon$ of the rows. The statement therefore immediately follows as in Ruwet et al. (2012), with a contamination $d\varepsilon$ ($r = n(1 - d\varepsilon)$ in the notation of Ruwet et al. 2012). □

In summary, if there is (only) structural contamination, `sclust` is as robust as `tclust`, which is based on trimming. If there is (also) component-wise contamination, `sclust` is more and more robust than `tclust` as the dimensionality d grows.

5. ILLUSTRATION OF THE METHOD

In this section, we illustrate the `sclust` procedure on simulated and real data examples.

5.1 Simulation: *M5* Type Data

In this section, we outline a brief simulation study with *M5* type data (García-Escudero et al. 2008), which are very challenging in the context of robust heterogeneous clustering. The set up is as follows: we have $k = 3$ groups, with $\mu_1 = (0, 8, \ldots, 0)'$, $\mu_2 = (8, 0, \ldots, 0)$, $\mu_3 = (-8, -8, 0, \ldots, 0)$, $\Sigma_1 = \text{diag}(1)$, $\Sigma_2 = \text{diag}(45, 30, 1, \ldots, 1)$, $\text{diag}(\Sigma_3) = (15, 15, 1, \ldots, 1)$, $\Sigma_{312} = \Sigma_{321} = -10$, and $\pi = (0.2, 0.4, 0.4)$. The groups have different scales and two of them overlap. We contaminate component-wise and generate outliers from independent uniforms in the interval $[-100, 100]$. We evaluate clustering procedures based on their ability to estimate the population parameters π, μ, and Σ. The ability to reconstruct the true underlying clustering will be evaluated in confirmatory cluster analysis on real data in the next sections. For each

Table 1. SSE in estimation of μ, Σ, and π with $M5$ type data, for the classical Gaussian mixture (GM) model, `tclust` and `sclust` procedures. For the last two, in parentheses we report the trimming or snipping level, respectively. The results are averaged over $B = 1000$ iterations

	$n = 2000$			$n = 200$		
	SSE_μ	SSE_Σ	SSE_π	SSE_μ	SSE_Σ	SSE_π
$d = 2, \varepsilon = 0.05$						
Classical GM	4168.61	982,868.51	0.43	4463.36	775,487.42	0.40
`tclust`(ε)	1414.43	4515.16	0.02	1529.53	3243.10	0.03
`tclust`($d\varepsilon$)	0.76	492.75	0.00	2.74	568.49	0.01
`sclust`(ε)	1.62	279.15	0.00	0.47	261.40	0.00
$d = 2, \varepsilon = 0.1$						
Classical GM	3982.63	1,473,266.81	0.34	4534.46	1,325,257.78	0.33
`tclust`(ε)	3749.72	16,400.89	0.05	2757.12	9024.05	0.05
`tclust`($d\varepsilon$)	0.60	620.90	0.00	3.08	761.73	0.01
`sclust`(ε)	2.56	312.73	0.01	1.00	282.34	0.01
$d = 6, \varepsilon = 0.05$						
Classical GM	1155.09	424,557.17	0.25	1891.16	320,958.86	0.26
`tclust`(ε)	2939.96	111,356.29	0.32	3785.59	80,987.07	0.33
`tclust`($d\varepsilon$)	5.03	1643.95	0.01	0.68	1378.27	0.01
`sclust`(ε)	2.32	1229.42	0.00	0.77	1137.26	0.00
$d = 6, \varepsilon = 0.10$						
Classical GM	270.47	1,749,127.81	0.08	771.48	1,491,404.77	0.10
`tclust`(ε)	503.88	438,421.91	0.16	2186.23	324,018.74	0.19
`sclust`(ε)	2.85	1269.11	0.00	1.04	1153.24	0.00
$d = 10, \varepsilon = 0.05$						
Classical GM	286.88	613,580.59	0.14	644.86	485,734.03	0.15
`tclust`(ε)	331.54	243,470.35	0.19	790.25	210,901.31	0.18
`sclust`(ε)	2.88	1580.69	0.00	0.93	1331.41	0.00
$d = 10, \varepsilon = 0.10$						
Classical GM	266.00	2,488,250.35	0.03	513.98	2,128,864.83	0.04
`tclust`(ε)	273.91	995,208.55	0.04	539.06	801,389.99	0.05
`sclust`(ε)	3.61	1537.58	0.00	1.25	1346.04	0.00

setting, we report the Sum of Squared Errors (SSE) in estimating each vector/array of parameters for the classical Gaussian mixture model, the `tclust` procedure with trimming level ε, and `sclust` procedure with snipping level ε. When feasible, we report also the results of the `tclust` procedure with trimming level $d\varepsilon$, which according to the results in Section 4 should have the same robustness properties of the `sclust` procedure. We also compare the true and predicted group labels using the modified Rand index (Hubert and Arabie 1985), which is a measure of agreement between the true and reconstructed clustering; and report the number of false positives (FP) and false negatives (FN) when classifying outlying entries. The Rand index gives a measure of the performance of the algorithms in estimation of w, and FP and FN measure performance of the algorithms in estimation of V. For all procedures, including the classical Gaussian mixture model, we use constraint (6) with $\lambda = 12$. In Tables 1 and 2, we report the results for $n = 200, 2000$, $d = 2, 6, 10$, and $\varepsilon = 5\%, 10\%$.

From the table, it can be seen that classical Gaussian mixture models and `tclust` with trimming level equal to ε lead to break down of the estimates in all settings. When feasible, a trimming level of $d\varepsilon$ leads to satisfactory SSE, but at the price of having to drop some solutions due for instance to empty clusters. Finally, `sclust` with snipping level ε is seen to work well in all settings, with good resistance to the level of contamination and dimensionality (the SSE is approximately constant with ε and d). The difference in performance between `tclust` and `sclust` with same trimming/snipping level is only due to the outlier generating model. If there only are structural (i.e., row-wise) outliers, the two procedures give approximately the same results. We have performed some simulations in this direction, which are not shown for reasons of space.

5.2 Metallic Oxide Data

The data we use for our first real data example were recorded by Bennet (1954) and are related to an investigation of the effects of processing on properties of metallic oxide. The metal content minus 80% by weight was recorded for two types of metallic oxide raw material, in, respectively, 18 and 13 lots. Two samples were randomly chosen from each lot, and each measurement was performed twice by each of two chemists. Hence, in this application $k = 2$, $n = 31$, and $d = 8$. The same data were analyzed using mixed models in Fellner (1986), Zewotir and Galpin (2007), and with robust double clustering in Farcomeni (2009). There is structural contamination as some lots have extra low metal content, as noted also by Farcomeni (2009). There are two sources of component-wise contamination: on one hand, there are measurement errors introduced by the chemists; on the other hand, some structural outliers are sampled *within* a lot. This is a *confirmatory* cluster analysis as the true clusters are known in advance (and of course labels are not used for clustering). We

Table 2. Rand index, average number of false positives (FP) and false negatives (FN) with $M5$ type data, for the classical Gaussian mixture model (GM), `tclust` and `sclust` procedures. For the last two, in parentheses we report the trimming or snipping level, respectively. The results are averaged over $B = 1000$ iterations

	$n = 2000$			$n = 200$		
	Rand	FN	FP	Rand	FN	FP
$d = 2, \varepsilon = 0.05$						
Classical GM	0.08	20	0	0.00	200	0
`tclust`(ε)	0.77	8.62	10.62	0.76	94.94	96.93
`tclust`($d\varepsilon$)	0.87	2.79	24.79	0.89	28.63	230.63
`sclust`(ε)	0.89	4.75	4.75	0.90	48.75	48.75
$d = 2, \varepsilon = 0.1$						
Classical GM	0.00	40	0	0.00	400	0
`tclust`(ε)	0.65	17.33	19.33	0.62	182.68	184.69
`tclust`($d\varepsilon$)	0.86	620.90	0.00	0.87	47.34	449.34
`sclust`(ε)	0.87	7.35	7.35	0.88	100.95	100.95
$d = 6, \varepsilon = 0.05$						
Classical GM	0.00	60	0	0.00	600	0
`tclust`(ε)	0.00	45.41	51.41	0.00	461.69	467.69
`tclust`($d\varepsilon$)	0.82	2.96	308.96	0.87	30.22	3036.22
`sclust`(ε)	0.90	12.83	12.83	0.90	141.52	141.52
$d = 6, \varepsilon = 0.10$						
Classical GM	0.00	120	0	0.00	1200	0
`tclust`(ε)	0.00	84.44	90.44	0.00	840.86	846.86
`sclust`(ε)	0.82	20.20	20.20	0.84	219.82	219.82
$d = 10, \varepsilon = 0.05$						
Classical GM	0.00	100	0	0.00	1000	0
`tclust`(ε)	0.00	80.40	90.40	0.00	800.23	810.23
`sclust`(ε)	0.86	16.75	16.75	0.87	138.30	138.30
$d = 10, \varepsilon = 0.10$						
Classical GM	0.00	200	0	0.00	2000	0
`tclust`(ε)	0.00	147.71	157.71	0.00	1481.46	1391.46
`sclust`(ε)	0.75	25.91	25.91	0.82	238.97	238.97

compare the true and predicted group labels using the modified Rand index. To illustrate the advantage of using model-based clustering over k-means, we also compare with the classical k-means procedure, trimmed k-means (Cuesta-Albertos, Gordaliza, and Matrán 1997), and snipped k-means (Farcomeni 2013). The classical k-means procedure with $k = 2$ leads to a very bad solution, in which lots 6 and 7 of Type 2 belong to one group and all the other rows belong to the other group. The same result is given by partitioning around medoids (PAMs) and Gaussian mixture models. The resulting Rand index is only 5.8%. If we increase the number of groups to $k = 3$, there still is a badly behaved solution: one group is made of two lots (6 and 7 of Type 2). The Rand index for the other rows is about 17% in all cases. We now turn to robust procedures. We use different trimming and snipping levels, whose results are shown in Table 3.

When $\varepsilon = 5\%$, the results of `tclust` and `sclust` coincide. There are two rows that must be trimmed, being structural outliers with extra low metal content. When $\varepsilon > 5\%$, snipping procedures are able to remove isolated entries in addition to the two structural outliers, better unveiling the true labels. It is important to emphasize that to have a fair comparison across procedures, in Table 3 we computed the Rand index only based on observations that were not trimmed by any of them for fixed ε. Consequently, even if it could seem like there is some sensitivity to the choice

Table 3. Comparison of the modified Rand index for the metallic oxide example for different trimming/snipping levels, and G statistic associated with `sclust`. A comparison of the Rand index across different trimming levels is not reliable since the Rand index is computed on a different number of elements

ε	`tkmeans`	`skmeans`	`tclust`	`sclust`	G
5%	17.2%	17.2%	24.0%	24.0%	60.39
7.5%	22.1%	26.0%	22.1%	26.2%	63.64
10%	20.0%	26.0%	27.9%	29.8%	63.20
12.5%	17.6%	19.1%	19.7%	40.9%	62.76
15%	17.6%	26.0%	25.9%	49.0%	64.15
17.5%	14.7%	26.0%	24.0%	38.9%	64.88

of ε, the results should not be compared row-wise. Sensitivity to the choice of ε can instead, at least in part, be evaluated in the last column, where we report the G statistic as proposed in (12). An independent check of sensitivity to the choice of ε was performed by checking the distance between parameter estimates at consecutive values of ε, and was confirmed by checking that the estimates are very close for $\varepsilon \geq 0.075$. A comparison of the G statistics leads to select $\varepsilon = 0.075$, as a clear elbow is seen in correspondence of this value. The large values of the G statistic confirm the presence of outliers, and the slight increase from $\varepsilon = 0.05$ to $\varepsilon = 0.075$ indicates that the proportion of outlying entries should be at least 7.5%. On the other hand, the results are fairly stable at larger values of ε. This leads us to conclude that (i) it is not needed that $\varepsilon > 0.075$ and that (ii) results are fairly stable (in terms of centroid estimate, and consequently of clustering of observations) for $\varepsilon \geq 0.075$.

5.3 Water Treatment Plant Data

Our second example is a genuine cluster analysis problem regarding daily measures of sensors in a urban waste water treatment plant. Data have been collected by the UCI Machine Learning Repository (Bache and Lichman 2013), and contain $n = 380$ daily measurements for $d = 38$ variables. These variables include measurements of levels of Zinc, pH, chemical and biological demand of oxygen, suspended and volatile solids, and sediments. These measurements are repeated at four different locations (input, first settler, second settler, output). It is easy to presume that there may be structural outliers, that is, unusual days for what concerns the measurements; but also that there may be entry-wise outliers, that is, occasional measurement errors only for few dimensions or situations at single locations for one or more features.

We begin by using the trimmed likelihood curve methodology of García-Escudero et al. (2011) to select the number of clusters. From Figure 1, it can be seen that the optimum of the objective function is rather close for $k = 3, 4, 5$ when ε is above a threshold. On the other hand, there is a clear advantage in passing from $k = 1$ to $k = 2$ and from $k = 2$ to $k = 3$. On the basis of these considerations, we fix $k = 3$.

We now choose ε. To do so, we compute the G statistic as in (12) for $k = 3$ and different values of ε. A plot of the G statistics versus ε is given in Figure 2. The plot suggests that the estimates are fairly stable for $\varepsilon \geq 0.05$, so that we end up setting $\varepsilon = 0.05$. Note that this suggestion is confirmed by

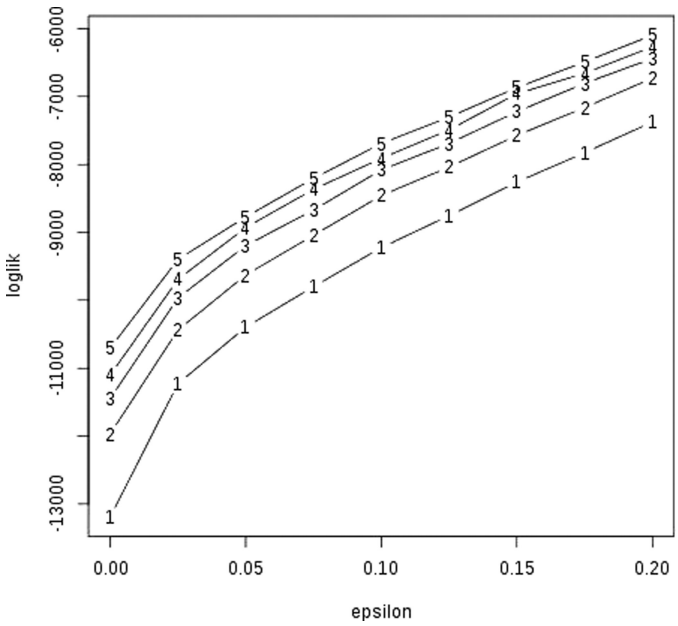

Figure 1. Maximum of the likelihood for different k and ε for the water treatment plant data.

the likelihood curve in Figure 1, as for $\varepsilon \geq 0.05$ the curves for $k = 3, 4, 5$ are rather close. Figure 2 is also a substantial evidence that there are outliers in the data, so that classical methods may be biased.

We therefore use `sclust` with $k = 3$ and $\varepsilon = 0.05$. For reasons of space we do not report the parameter estimates, but only a summary of them. These are obtained by standardizing the clean entries in the data matrix and averaging groups of measurements by location. Note that we could not do this operation on the original data matrix due to the possibility of outliers (which are indeed detected as a 5% of the entries), while the operation is safe when restricted to entries not flagged by the `sclust` algorithm. We report only the average for the standardized measurements at the input, first and second settler, and output in the first panel of Table 4. In the second panel, we compare with the results of classical Gaussian mixture models.

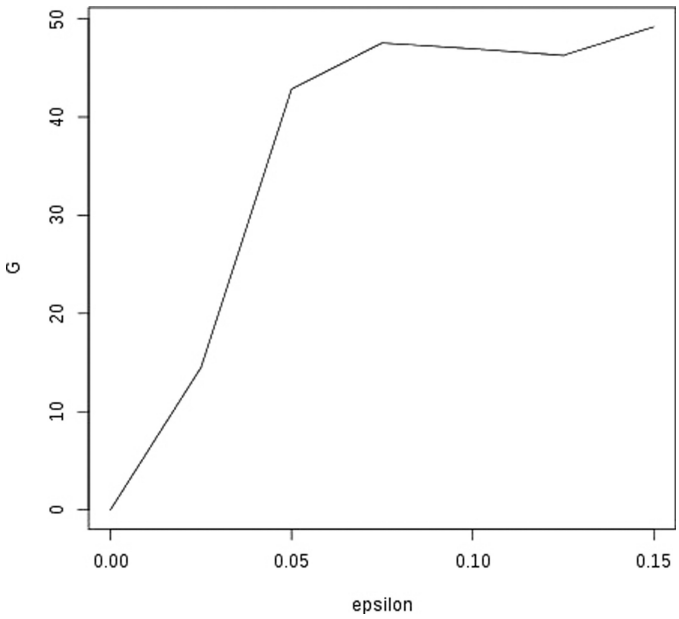

Figure 2. G statistics for the water treatment plant data when $k = 3$.

Table 4. Average of standardized clean measurements by location and cluster as identified with `sclust` (upper panel) and with classical Gaussian mixture models (lower panel) for the water treatment plant data

	Cluster 1	Cluster 2	Cluster 3
Summary of robust centroid estimates			
Input	−0.062	−0.385	0.327
First settler	−0.024	−0.537	0.393
Second settler	0.089	−0.640	0.388
Output	0.034	−0.333	0.154
Summary of classical centroid estimates			
Input	−0.022	−0.123	0.268
First settler	0.014	−0.208	0.214
Second settler	0.054	−0.563	0.499
Output	−0.039	−0.218	0.476

Given that measurements are averaged within clusters after standardization, the numbers in Table 4 can be interpreted as the average difference between the amounts at a specific location and cluster and the average marginal amount (i.e., regardless of cluster and location). The latter is set to zero by column standardization. For example, −0.062 indicates that for days in cluster 1, at input, the load is only slightly smaller than the average load regardless of cluster and location.

We can safely conclude that `sclust` has identified groups of days with about average (group 1, about 40% of the observations), low (group 2, 25%), and heavy (group 3, the remaining 35%) loads to the plant. The groups are well separated as testified by the F statistic, which can be computed as 22.3. On the other hand, classical Gaussian mixture models seem to produce groups that are on the one hand slightly less separated (especially at input), where the centroids of the second group are generally closer to zero than the robustly estimated centroid; and on the other hand seems to grossly overestimate the output in the third cluster. Furthermore, only 14.2% of the observations are assigned to cluster 3, which is associated with heavy load; and 66% are assigned to cluster 1, which is associated with about average load for classical Gaussian mixture.

The clustering results are visualized in Figure 3, where we show the first two principal components obtained from clean observations as identified by `sclust` in the first panel, and the first two principal components obtained from the entire dataset in the second.

Figure 3 clearly confirms that there is contamination in this dataset, which can make results less interpretable and meaningful. The first biplot, on the other hand, clearly identifies the third group in the upper right portion of the plot and the second in the lower left portion. As often happens, the separation of the clusters is of course underestimated in the biplot, given that a 38-dimensional space is projected onto a two-dimensional space.

Another important outcome of this analysis is the distribution of the number of consecutive days in cluster 3, which identifies days of heavy duty. This information shall be used for planning of plant use and prevention of faults. With our algorithm, we can

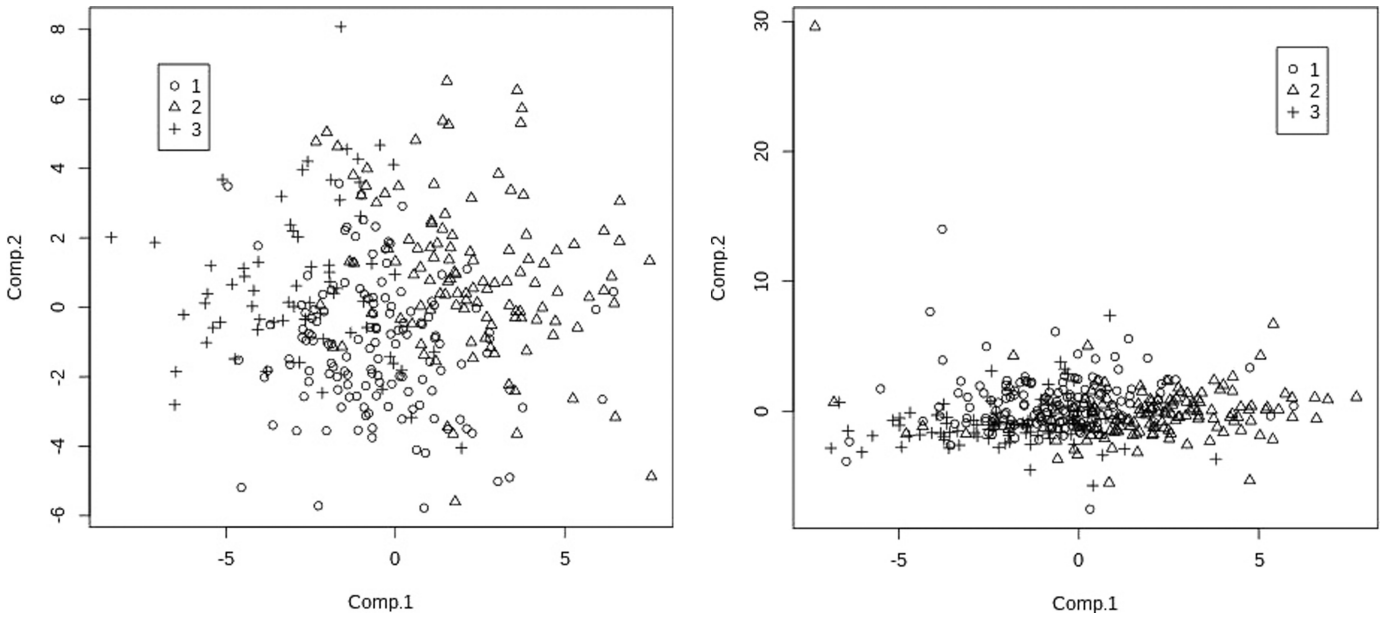

Figure 3. Biplot based on clean (first panel) and all observations (second panel), with clustering evaluated by `sclust`.

estimate an average of 1.6 consecutive days in state 3, standard deviation 0.97, with a probability of persistence of two or more consecutive days in state 3 of 37%. On the other hand, if we use classical Gaussian mixture models, we obtain an average consecutive number of days in state 3 of 1.47, standard deviation 0.88, with a probability of two or more days in state 3 being only 25%.

A conclusion is that Gaussian mixture models that ignore outlying entries may lead to underestimate the activity and load of the plant.

As suggested by the associate editor, we also perform a stability analysis as follows: after the three clusters are identified, we remove observations assigned to one cluster and repeat the clustering operation on the remaining observations. With both `sclust` and Gaussian mixture models, we can recover the two remaining clusters when we remove the average and low load clusters. On the other hand, when we remove the heavy load cluster only with `sclust`, we can recover the two remaining clusters. This is reported in Table 5, where it can be seen that for classical Gaussian mixture models only at the second settler there is a marked separation between the two clusters.

Finally, with this dataset we report on the computational burden of the snipping procedure: under all settings it takes less than 2 min to obtain parameter estimates, using nonoptimized R code, when $k = 3$.

Table 5. Average of standardized clean measurements by location and cluster as identified with `sclust` (upper panel) and with classical Gaussian mixture models (lower panel) for the water treatment plant data, after removing the originally identified high load cluster

	Cluster 1	Cluster 2
Summary of robust centroid estimates		
Input	−0.007	−0.290
First settler	0.044	−0.408
Second settler	0.165	−0.518
Output	0.066	−0.278
Summary of classical centroid estimates		
Input	−0.032	−0.081
First settler	−0.009	−0.117
Secondsettler	0.045	−0.475
Output	−0.043	−0.188

6. CONCLUSIONS

We have proposed a robust model-based clustering procedure under an entry-wise spurious outliers model. A first important comment regards the fact that our spurious outlier model is *not* scale invariant. Any linear combination may propagate the outliers and contaminate some clean entries. See also Alqallaf et al. (2009) on this point. Further, constraint (6) makes estimators also not scale invariant even under scale invariant contamination models.

Our estimation procedure is based on an EM algorithm augmented with a stochastic optimization, in place of commonly used concentration steps, for snipping. It has been noted that the computational complexity of the iteration of the proposed EM algorithm is linear in all parameters. It shall be added that the number of iterations needed to convergence is likely to increase with the dimensionality and number of clusters, and therefore, as with many other model-based clustering algorithms, `sclust` may be too time consuming for d in the hundreds or larger.

We assumed that V is a parameter, as common in the robust model-based clustering literature. As suggested by a referee, we could also treat V as a latent variable and flag outliers based on their posterior probability. This approach to the best of our knowledge has not been pursued previously, and is an interesting possibility for further extension of robust model-based clustering. Treating V as random would undoubtedly connect our approach with weighted likelihood methods for dealing with robustness (e.g., Markatou 2000); but would also make the procedure less general as more, possibly restrictive, assumptions would be needed on the contaminating mechanism. A similar

reasoning applies to the use of additional assumptions g_i, which could lead to more efficient clustering algorithms. It is important to underline that while we may have some control on the data-generating mechanism, we usually have no control on the contaminating one, so that it is difficult to justify and check assumptions on it.

ACKNOWLEDGMENTS

The author is grateful to an associate editor and two referees for kind suggestions and comments.

[Received January 2013. Revised July 2013.]

REFERENCES

Alqallaf, F., Van Aelst, S., Yohai, V. J., and Zamar, R. H. (2009), "Propagation of Outliers in Multivariate Data," *The Annals of Statistics*, 37, 311–331. [1,9]

Bache, K., and Lichman, M. (2013), UCI Machine Learning Repository, *http://archive.ics.uci.edu/ml*, Irvine, CA: University of California, School of Information and Computer Science. [7]

Banfield, J., and Raftery, A. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821. [1]

Bennet, C. A. (1954), "Effect of Measurement Error on Chemical Process Control," *Industrial Quality Control*, 11, 17–20. [6]

Cambpell, N. A. (1984), "Mixture Models and Atypical Values," *Mathematical Geology*, 16, 465–477. [1]

Chakraborty, B., and Chaudhury, P. (2008), "On an Optimization Problem in Robust Statistics," *Journal of Computational and Graphical Statistics*, 17, 683–702. [3,4]

Cuesta-Albertos, J., Gordaliza, A., and Matrán, C. (1997), "Trimmed k-means: An Attempt to Robustify Quantizers," *The Annals of Statistics*, 25, 553–576. [7]

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood From Incomplete Data Via the EM Algorithm" (with discussion), *Journal of the Royal Statistical Society,* Series B, 39, 1–38. [1,4]

Donoho, D. L., and Huber, P. J. (1983), "The Notion of Breakdown Point," in *A Festschirift for Erich L. Lehmann*, eds. P. Bickel, K. Doksum, and J. L. Jr. Hodges, Belmont, CA: Wadsworth, pp. 157–184. [5]

Farcomeni, A. (2009), "Robust Double Clustering: A Method Based on Alternating Concentration Steps," *Journal of Classification*, 26, 77–101. [4,5,6]

——— (2013), "Snipping for Robust k-means Clustering Under Component-wise Contamination," *Statistics and Computing*, forthcoming. [1,3,7]

Farcomeni, A., and Ventura, L. (2012), "An Overview of Robust Methods in Medical Research," *Statistical Methods in Medical Research*, 21, 111–133. [1]

Farcomeni, A., and Viviani, S. (2011), "Robust Estimation for the Cox Regression Model Based on Trimming," *Biometrical Journal*, 53, 956–973. [3]

Fellner, W. H. (1986), "Robust Estimation of Variance Components," *Technometrics*, 28, 51–60. [6]

Fraley, C., and Raftery, A. E. (1998), "How Many Clusters? Which Clustering Method? Answers Via Model-based Cluster Analysis," *The Computer Journal*, 41, 578–588. [1]

——— (2002), "Model Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, 97, 611–631. [1]

Fritz, H., García-Escudero, L. A., and Mayo-Iscar, A. (2013), "A Fast Algorithm for Robust Constrained Clustering," *Computational Statistics and Data Analysis*, 61, 124–136. [2,3,4]

Frühwirth-Schnatter, S., and Pyne, S. (2010), "Bayesian Inference for Finite Mixtures of Univariate Skew-normal and Skew-t Distributions," *Biostatistics*, 11, 317–336. [1]

Gallegos, M. T., and Ritter, G. (2005), "A Robust Method for Cluster Analysis," *The Annals of Statistics*, 33, 347–380. [1,2,5]

——— (2009a), "Trimmed ML Estimation of Contaminated Mixtures," *Sankhyā*, 71, 164–220. [1,2]

——— (2009b), "Trimming Algorithms for Clustering Contaminated Grouped Data and their Robustness," *Advances in Data Analysis and Classification*, 3, 135–167. [2]

——— (2010), "Using Combinatorial Optimization in Model-based Trimmed Clustering With Cardinality Constraints," *Computational Statistics and Data Analysis*, 54, 637–654. [1,2]

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008), "A General Trimming Approach to Robust Cluster Analysis," *The Annals of Statistics*, 36, 1324–1345. [1,2,5]

——— (2011), "Exploring the Number of Groups in Robust Model-based Clustering," *Statistics and Computing*, 21, 585–599. [4,7]

Genton, M. G., and Lucas, A. (2003), "Comprehensive Definitions of Breakdown Points for Independent and Dependent Observations," *Journal of the Royal Statistical Society,* Series B, 65, 81–94. [5]

Hampel, F. R. (1971), "A General Qualitative Definition of Robustness," *Annals of Mathematical Statistics*, 42, 1887–1896. [5]

Heritier, S., Cantoni, E., Copt, S., and Victoria-Feser, M.-P. (2009), *Robust Methods in Biostatistics*, Chichester: Wiley. [1]

Hodges, J. L., Jr. (1967), "Efficiency in Normal Samples and Tolerance of Extreme Values for Some Estimates of Location," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1), Berkeley, CA: Univ. California Press, pp. 163–186. [5]

Huber, P. J. (1964), "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, 35, 73–101. [1]

Huber, P. J., and Ronchetti, E. M. (2009), *Robust Statistics*, New York: Wiley. [1]

Hubert, L., and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193–218. [6]

Hubert, M., Rousseeuw, P. J., and Van Aelst, S. (2008), "High-breakdown Robust Multivariate Methods," *Statistical Science*, 23, 92–119. [1]

Markatou, M. (2000), "Mixture Models, Robustness, and the Weighted Likelihood Methodology," *Biometrics*, 56, 483–486. [1,9]

McLachlan, G., and Peel, D. (2000), *Finite Mixture Models*, New York: Wiley. [1]

Neykov, N., Filsmoser, P., Dimova, R., and Neytchev, P. (2007), "Robust Fitting of Mixtures Using the Trimmed Likelihood Estimator," *Computational Statistics and Data Analysis*, 52, 299–308. [1]

Ruwet, C., Garcia-Escudero, L. A., Gordaliza, A., and Mayo-Iscar, A. (2013), "On the Breakdown Behavior of Robust Constrained Clustering Procedures," *TEST*, 22, 466–487. [5]

Tukey, J. W. (1962), "The Future of Data Analysis," *Annals of Mathematical Statistics*, 33, 1–67. [1]

Zewotir, T., and Galpin, J. S. (2007), "A Unified Approach on Residuals, Leverages and Outliers in the Linear Mixed Model," *TEST*, 16, 58–75. [6]