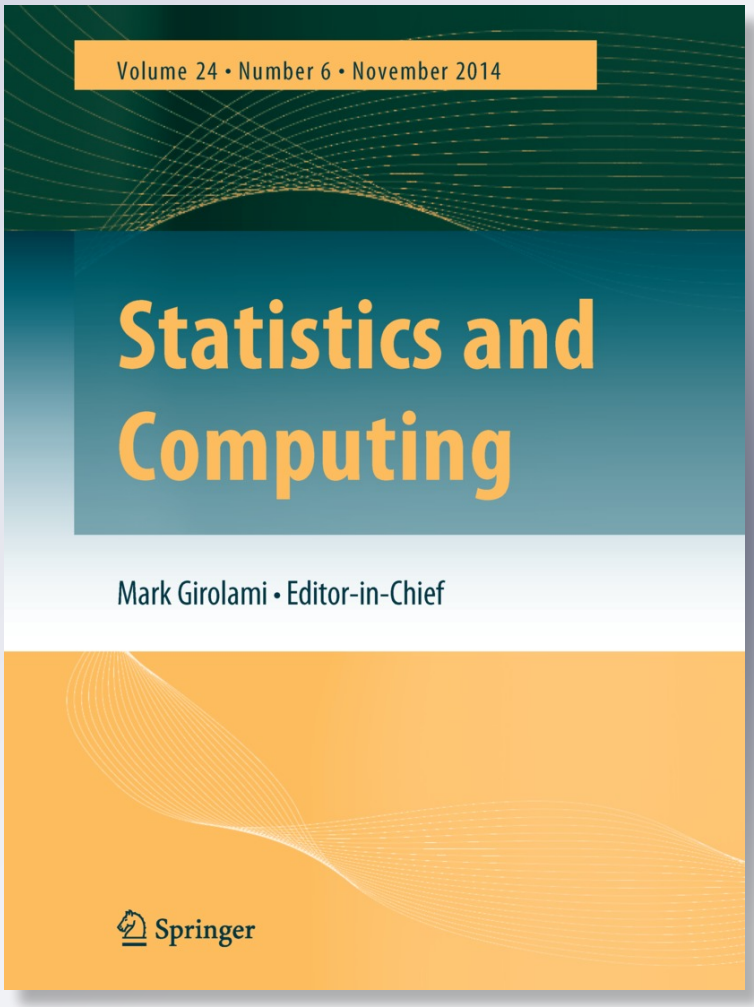
# *Snipping for robust k-means clustering under component-wise contamination*

**Alessio Farcomeni**

**Statistics and Computing**

ISSN 0960-3174
Volume 24
Number 6

Stat Comput (2014) 24:907-919
DOI 10.1007/s11222-013-9410-8




Springer

Stat Comput (2014) 24:907–919
DOI 10.1007/s11222-013-9410-8

# Snipping for robust $k$-means clustering under component-wise contamination

**Alessio Farcomeni**

Received: 26 November 2012 / Accepted: 5 June 2013 / Published online: 16 July 2013


**Abstract** We introduce the concept of snipping, complementing that of trimming, in robust cluster analysis. An observation is snipped when some of its dimensions are discarded, but the remaining are used for clustering and estimation. Snipped $k$-means is performed through a probabilistic optimization algorithm which is guaranteed to converge to the global optimum. We show global robustness properties of our snipped $k$-means procedure. Simulations and a real data application to optical recognition of handwritten digits are used to illustrate and compare the approach.

**Keywords** Clustering · $k$-Means · Outliers · Robustness · Snipping · Trimming

## 1 Introduction

Many procedures in robust statistics implicitely or explicitely adopt a Tukey–Huber contamination (THC) model (Tukey 1962; Huber 1964), according to which a small fraction of observations are structural outliers, while the remaining are not contaminated. Under the THC, an outlier is an observation arising from its own population, different from that of the bulk of the data. A different setting is given by component-wise contamination (CWC), in which each dimension of an observation can be independently contaminated with a certain probability $\varepsilon$ (Alqallaf et al. 2009). Under CWC, contamination arises during the measurement rather than the sampling phase. Observations are sampled from the specified model, but each measurement can be independently contaminated. To fix the ideas, suppose we are measuring gene expression levels repeatedly on independent slides. We have structural outliers when one or more genes arise from a different pathway than the one under study, while we have component-wise contamination due to contamination within each slide. For reviews of the basic concepts in robust statistics we refer the reader to the books by Huber and Ronchetti (2009), Heritier et al. (2009), and to the papers by Hubert et al. (2008) and Farcomeni and Ventura (2012). The focus of this paper is on unsupervised classification methods. The first approaches to robust clustering can be found in Kaufman and Rousseeuw (1990). There are now extremely effective clustering procedures able to deal with Tukey–Huber contaminated data. We can identify two main approaches: one is based on flexible mixture models (e.g., Banfield and Raftery 1993, Frühwirth-Schnatter and Pyne 2010), and another is based on the idea of impartial trimming (Gordaliza 1991). A different approach, which also is based on a THC model, is proposed in Forero et al. (2012). For a detailed review of robust clustering methods based on trimming see García-Escudero et al. (2010). The basis of this work is that all of these methods, including impartial trimming approaches, may break down under CWC. This is due to the fact that the number of observations with at least one contaminated entry can be very large under our general contamination model (even 100 %, indeed), and that trimming approaches remove contaminated observations entirely. The number of observations with at least one contaminated entry may be then too large for trimming methods to handle. To see this, consider Table 1 where we report on the expected proportion of entirely uncontaminated observations under CWC, for different values of $\varepsilon$ and dimensionality $d$. Note that in Table 1 we assume independent contamination (and homogeneity of the probability of contamination over

A. Farcomeni (✉)
Department of Public Health and Infectious Diseases, Sapienza—University of Rome, P.zzale Aldo Moro, 5, 00185 Roma, Italy
e-mail: alessio.farcomeni@uniroma1.it

**Table 1** Proportion of clean observations under independent CWC, for different values of component-wise contamination $\varepsilon$ and dimensionality $d$

| $d$ | 5 | 10 | 20 | 50 | 80 | 100 |
|---|---|---|---|---|---|---|
| $\varepsilon = 0.005$ | 0.98 | 0.95 | 0.90 | 0.78 | 0.67 | 0.61 |
| $\varepsilon = 0.01$ | 0.95 | 0.90 | 0.82 | 0.61 | 0.45 | 0.37 |
| $\varepsilon = 0.05$ | 0.77 | 0.60 | 0.36 | 0.08 | 0.02 | 0.01 |
| $\varepsilon = 0.1$ | 0.59 | 0.35 | 0.12 | 0.01 | 0.00 | 0.00 |

columns). If there is positive dependence, the proportion of clean observations may be a larger; while under negative dependence this proportion may be smaller.

In order to tackle robust clustering under CWC, we introduce the concept of *snipping*. 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, and potentially all observations can be snipped. Snipping is more flexible than trimming as removal of all entries of an observation, which is possible, corresponds to trimming. There are clear advantages of snipping under the CWC: in the setting of Table 1, when $d = 20$ and $\varepsilon = 0.1$ about 85 % of the contaminated observations will have only one or two contaminated entries. These clean entries will have to be discarded under trimming. With snipping, 18 or 19 values for each observation can be retained and used for clustering, with a very mild information loss. Even if the idea of snipping can be directly included in more general methodologies, and beyond the area of robust clustering, we focus in this paper on robust $k$-means (Cuesta-Albertos et al. 1997). Under a general contamination model we propose an efficient algorithm which simoultaneously performs $k$-means type clustering and snipping. Our algorithm, which is guaranteed to converge to the global optimum under mild conditions, automatically chooses which observations shall be trimmed, which shall be snipped and which and how many entries shall be removed for those. The rest of the paper is as follows: in the next section we introduce snipping and the corresponding contamination model. In Sect. 2.1 we illustrate an efficient algorithm for solving the snipped $k$-means problem and discuss its global convergence properties. In Sect. 2.2 we investigate robustness of our procedure. In Sect. 3 we compare clustering procedures with a brief simulation study, and illustrate with a real data set in Sect. 4. A discussion is given in Sect. 5. A sample C-within-R code for our approach can be found at http://www.afarcome.altervista.org/snip.zip.

## 2 Snipped $k$-means

We assume we have a sample of (possibly standardized) $n$ observations $Y_1, \ldots, Y_n$, with $Y_i \in \mathcal{R}^d$, which are divided in $k$ groups indicized in $\mathcal{J}_1, \ldots, \mathcal{J}_k$; where $\mathcal{J} = \cup_{c=1}^{k} \mathcal{J}_c \subseteq \{1, \ldots, n\}$ and $\mathcal{J}_j \cap \mathcal{J}_h = \emptyset$. The groups are separated in that the population mean in the $c$-th group is $\mu_c$, with $\mu_c \neq \mu_{c'}$ when $c \neq c'$. We assume $nd\varepsilon$ entries of the data matrix $Y$ 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 so that no observation is completely spoiled.

Our general contamination model is as follows: let $b_{ij}$, $i = 1, \ldots, n$ be such that $b_{ij} \in \{0, 1\}$ and $\sum_{ij} b_{ij}$ is equal to the integer closest to $nd(1-\varepsilon)$. We have that $b_{ij}$ is an indicator of the $ij$-th entry of the data matrix being free of contamination, while when $b_{ij} = 0$ the $j$-th column of the $i$-th observation has been contaminated. We assume

$$Y_i = b_i X_i + (1 - b_i) Z_i, \tag{1}$$

for $i = 1, \ldots, n$; where $Z_i \sim g_i(\cdot)$, with $g_i(\cdot)$ being a density in $\mathcal{R}^d$. Model (1) brings about component-wise contamination as entries of $X_i$ corresponding to a zero $b_i$ will be replaced by the elements of a spurious outlier in $\mathcal{R}^d$. We assume $X_i \sim MVN(\mu_c, \sigma^2 I_d)$ when $i \in \mathcal{J}_c$, $c = 1, \ldots, k$. The observations for which $b_{ij} = 0$ for all $j = 1, \ldots, d$ are outside $\mathcal{J}$; and are structural outliers, i.e., $Y_i = Z_i \sim g_i(\cdot)$ when $i \notin \mathcal{J}$. We adopt a crisp classification approach, therefore we will obtain a $k$-means type procedure given the assumption on the covariance matrices (i.e., given the restriction that the clusters are spherical). In our contamination model we assume there is a partition of the data matrix into contaminated and clean entries, in parallel with the fixed clustering partition, and that the number of contaminated entries is $nd\varepsilon$. When $k = 1$ we have a partially spoiled contamination model in the definition of Alqallaf et al. (2009). When $k > 1$ we have a partially spoiled contaminated (spherical) clusters model. Our non-probabilistic setting naturally includes inhomogeneity (that is, different contamination fractions in different columns), being based only on the resulting total number of contaminated entries. With the notation $Z_i \sim g_i(Z_i)$ we would like to underline that we sample each outlier from its own distribution. Along the lines of Gallegos and Ritter (2005) we make assumptions that outliers are affine independent and that non-regular observations can be ignored when estimating $\mu$. The latter is the consequence of the formal assumption that in any optimal partition with $nd\varepsilon$ snipped entries, the non-regular observations could be also obtained by maximizing the likelihood associated with $g_i(\cdot)$. For more details refer to Gallegos and Ritter (2005), García-Escudero et al. (2008) and also Ruwet et al. (2012).

Given the assumptions, the *snipped k-means* estimates of the centroid $\mu$ can be obtained by solving

$$\min_{V\in\mathcal{V}} \inf_{\{\mu_1,\ldots,\mu_k\}\in\mathcal{R}^d} \sum_{i=1}^{n} \min_{1\le c\le k} \sum_{j=1}^{d} v_{ij}(Y_{ij}-\mu_{cj})^2, \tag{2}$$

where $V$ is a binary matrix and $\mathcal{V}$ denotes the set of all subsets of $\mathcal{M}_{\{0,1\}}(n,d)$, the set of binary matrices with $n$ rows and $d$ columns, with exactly $\lceil nd\varepsilon\rceil$ entries equal to zero. A zero in $V$ identifies a discarded entry, no matter if it is discarded because of snipping or because of trimming of its corresponding row. An entire row of zeros identifies a trimmed entry. Note that with trimmed $k$-means at level $\varepsilon$ we remove exactly the same number of entries as snipped $k$-means at level $\varepsilon$. The only difference is that with trimming the entries are all aligned in rows, while with snipping they can be located anywhere within the data matrix.

### 2.1 Minimization of the loss function

The set of $\varepsilon$-snipped subsets of the data matrix $Y$, indicized by $V\in\mathcal{V}$, has an impressive cardinality. Furthermore, the problem of minimization of the loss (2) is substantially that of enumeration of the subsets of $\mathcal{V}$, which is impossible to perform in a reasonable computing time. Trimming approaches usually proceed by iteration of least square estimation of centroids and concentration steps to flag outliers with respect to the current estimates (e.g., Rousseeuw and Van Driessen 1999, Gallegos and Ritter 2005, Farcomeni 2009). We have found that in high dimensions this approach is easily trapped in local optima. We here develop a novel acceptance-rejection algorithm, along the lines of a general proposal by Chakraborty and Chaudhury (2008). The algorithm proceeds by iteratively updating an initial solution, identified by a $V(0)\in\mathcal{V}$. Let $V(t)$ denote the update obtained at the $t$-th iteration. Let $\mu(V(t))=(\mu_1(V(t)),\ldots,\mu_k(V(t)))$ be the corresponding centroid, estimated as usual but using only the dimensions corresponding to a positive entry in $V(t)$. Finally, let $l(V(t),\mu(V(T)))=\sum_{i=1}^{n}\min_{1\le c\le k}\sum_{j=1}^{d}v_{ij}(Y_{ij}-\mu_{cj})^2$ be the loss associated with $V(t)$.

Each iteration of our proposed algorithm proceeds in two steps: first, a candidate new solution $V_c$ is built; secondly, it is rejected or accepted with a certain probability. If it is accepted, $V(t)$ is updated to $V_c$, otherwise $V(t+1)\equiv V(t)$. Formally, $V_c$ is built by switching the positions of a zero and a positive entry of $V(t)$. These two entries are selected uniformly at random. Let now

$$p_t=\min\left(e^{-\frac{\log(t+1)}{D}(l(V_c,\mu(V_c))-l(V(t),\mu(V(t))))},1\right). \tag{3}$$

The candidate solution $V_c$ is accepted with probability $p_t$. This guarantees that a candidate solution leading to smaller loss is always accepted, while a candidate solution leading to slightly larger loss may also be accepted but with probability being lower and lower as $t$ grows. The algorithm can then escape local optima and explore the solution space during the first few runs. The only tuning parameter is $D$, which shall be set approximately equal to the maximal expected change in loss when switching two non-contaminated entries.

We now show that the proposed algorithm converges to the global optimum as $t$ grows:

**Theorem 1** *Let $H$ denote the superset of all sets in $\mathcal{V}$ corresponding to a global optimum satisfying (2). For any $V(0)\in\mathcal{V}$ we have that* $\lim_t \Pr(V(t)\in H)=1$.

*Proof* Note that if $\sup_{(V,Z)\in\mathcal{V}\times\mathcal{V}}|l(V,\mu(V))-l(Z,\mu(Z))|=0$, the thesis trivially holds. Now suppose $\sup_{(V,Z)\in\mathcal{V}\times\mathcal{V}}|l(V,\mu(V))-l(Z,\mu(Z))|>0$. Define $\Delta_t=\sup_{(V,Z)}\frac{\log(t+1)}{D}(l(V,\mu(V))-l(Z,\mu(Z)))$, where $V\in\mathcal{V}$, and $Z\in\mathcal{V}$ is obtained by switching two entries or two rows of $V$. To see that the proposed algorithm satisfies conditions in Chakraborty and Chaudhury (2008, Pag. 686), we only need to show that $\sum_t e^{-\Delta_t}=\infty$, which is straightforward since $\sum_t e^{-\Delta_t}\ge\sum_t\frac{1}{t+1}=\infty$. The thesis follows. □

We shall notice that reaching the global optimum is guaranteed only when $t$ grows indefinitely. Since the algorithm is run in practice for a fixed, even if large, number of iterations, we now briefly discuss the rate of converge of the algorithm and propose an alternative accelerated convergence strategy.

Using results from Chakraborty and Chaudhury (2008, Theorem 2), we can derive an approximate rate of convergence. It is shown in Chakraborty and Chaudhury (2008, Theorem 2) that the rate of convergence is of the order $t^{-d_{\min}/(d_{\min}+D)}$ eventually, where $d_{\min}$ is the change in loss obtained when switching an entry from the optimal solution. As a consequence, the maximal rate of convergence that can be obtained is $\sqrt{t}$, and the actual rate depends on the difference between the smallest loss and the next smallest value that can be obtained. It can also be shown (Chakraborty and Chaudhury 2008, Theorem 3) that a rate of $t^{-d_{\min}/D}$ can be obtained if we let the algorithm run for a fixed (say, $N/2$) number of iterations with an acceptance probability of

$$p_t=\min\left(e^{-\frac{\log(N/2)}{D}\frac{2t}{N}(l(V_c,\mu(V_c))-l(V(t),\mu(V(t))))},1\right), \tag{4}$$

where $N$ denotes the pre-specified total number of iterations of the algorithm. In practice, acceptance probability (4) can be used to quickly explore the solution space, and then after the first half iterations we can switch to (3) in order to increase the likelihood of convergence to the global optimum. In our implementation we use the accelerated convergence acceptance probability (4) for $10^5$ iterations, and then let the algorithm run for another $10^5$ iterations with

the global convergence acceptance probability (3). Given that the computational complexity of each iteration is minimal (it only involves random sampling and evaluation of the loss, in practice), the C-within-R code is relatively quick. More precisely, it can be seen that the computational complexity of each iteration is of order $knd$. To summarize, we give the pseudo-code for the algorithm based on $2N$ iterations:

**for** $t = 1, \ldots, 2N$ **do**
  Let $V(t) := V(t-1)$, $V_c := V(t-1)$.
  Sample uniformly at random a zero and one entry of $V(t-1)$, with coordinates $(r_0, c_0)$ and $(r_1, c_1)$, respectively.
  Set $V_c[r_0, c_0] = 1$, $V_c[r_1, c_1] = 0$
  **if** $t <= N$ **then**
    Set $p_t$ as in (4)
  **else**
    Set $p_t$ as in (3)
  **end if**
  Let $u \sim U[0, 1]$.
  **if** $u < p_t$ **then**
    let $V(t) = V_c$
  **end if**
**end for**

Even with the accelerated probability (4), global convergence is guaranteed only when the number of iterations diverge. Therefore, as with many other iterative algorithms, it is recommended to repeatedly run the algorithm, with different starting solutions $V(0)$. In this paper we build an initial deterministic starting solution by integrating multivariate outlier identification methods with the trimmed $k$-means procedure. Other starting solutions are obtained by randomly perturbing the result obtained at convergence from the deterministic starting solution.

A final comment regards practical choice of the snipping level $\varepsilon$. In our experience, a careful choice of $\varepsilon$ is not crucial as long as it is larger than or equal to the actual contamination level. A slightly large $\varepsilon$ will have negligible consequences on the estimates and on quality of clustering. Snipping will not even have deleterious effects on clustering of units, as they will be assigned anyway to the closest cluster. One can proceed in practice by comparing stability of the results for different snipping levels. A formal suggestion along these lines is the $G$ statistic of Farcomeni (2009). See also García-Escudero et al. (2011).

### 2.2 Robustness properties

There are many available methods to evaluate the robustness properties of a procedure (Huber and Ronchetti 2009; Hampel et al. 1986), some of which have been developed especially for clustering (e.g., Hennig 2008). We will focus here on global measures, given by *breakdown values*. Hodges (1967) and Donoho and Huber (1983) define a finite sample breakdown value as the smallest fraction of outliers that can break down the estimate. The asymptotic breakdown value (Hampel 1971) is the breakdown value of a procedure as the sample size grows. Clustering procedures can be formally declared to be robust when the asymptotic breakdown value is not infinitesimal.

We here must distinguish between two kinds of corruption: one, that is brought about by structural outliers, replaces entire rows of the data matrix by arbitrary values. This kind of corruption lead (Gallegos and Ritter 2005) to the definition of *individual breakdown point* in the context of cluster analysis. More formally, let $Y_R$ denote the data matrix in which $R$ rows are replaced by arbitrary vectors in $\mathcal{R}^d$. The individual breakdown point is defined as

$$\varepsilon^{(i)} = \frac{1}{n} \min\left\{ R : \sup_{Y_R} \left\| \mu(Y) - \mu(Y_R) \right\| = \infty \right\}, \tag{5}$$

where $\mu(Y)$ is the centroid estimate obtained from data $Y$.

Another kind of corruption, that is brought about by component-wise contamination, is that obtained by replacing isolated entries. A definition of breakdown point specifically tailored to this situation is that of *cell breakdown point* given in Farcomeni (2009), who derived it in a different but related context. Formally, let $Y_C$ denote the data matrix in which $C$ elements are replaced by arbitrary values in $\mathcal{R}$. The cell breakdown point is defined as

$$\varepsilon^{(c)} = \frac{1}{nd} \min\left\{ C : \sup_{Y_C} \left\| \mu(Y) - \mu(Y_C) \right\| = \infty \right\}. \tag{6}$$

Before stating our results in terms of the definitions above, we note that the global robustness properties of trimmed $k$-means shall be discussed for ideally well-behaved data matrices, in which the clean observations form well-separated groups. This gives the *restricted* breakdown points (Gallegos and Ritter 2005). It can be shown that the unrestricted (*universal*) breakdown points for any robust or classical $k$-means type procedure are infinitesimal.

In order to allow a comparison with trimmed $k$-means, we adopt the same restrictions of Ruwet et al. (2012). In the following, let $W_{\mathcal{J}}$ denote the total sum of squares and products for a partition $\mathcal{J}$.

**Definition 1** Let $\rho > 0$. Let $\mathcal{J}$ be any partition $\mathcal{J}$ of $Y_n$ into $J_1, \ldots, J_k$, each of size at least $u$. Let $T_n \subseteq Y_n$ be any subset of $Y_n$ of size smaller than $n(1-\epsilon)$ and larger than or equal to $q_{u,\rho} = \max(n(1-2\epsilon), k(k-1)+1, (n-u)/(1-\rho))$.

A $d$-dimensional data set $Y_n$ belongs to the set of well clustered data sets $\mathcal{K}_{u,\rho}$ if for any partition $\mathcal{T}$ of $T_n$ into $k-1$ groups $J_{T,1}, \ldots, J_{T,k-1}$ it happens that

$$1 + \frac{\eta_\rho |\mathcal{J} \cap \mathcal{T}|}{\operatorname{tr} W_{\mathcal{J} \cap \mathcal{T}}}$$

$$\times \min_{h,l\neq c, J_{T,h}\cap J_l\neq\emptyset, J_{T,h}\cap J_c\neq\emptyset} \|\hat{\mu}_{J_{T,h}\cap J_l} - \hat{\mu}_{J_{T,h}\cap J_c}\|$$

$$\geq \frac{k^{2/d}\max_{R_n:T_n\subseteq R_n}\operatorname{tr} W_{\mathcal{J}\cap R_n}}{\operatorname{tr} W_{\mathcal{J}\cap\mathcal{T}}},$$

where $\eta_\rho = \rho(1-\rho)$ if $k=2$ and $\eta_\rho = \rho/2$ if $k>2$, and $R_n$ is such that we added observations to $T_n$ so to reach $n(1-\varepsilon)$ observations.

For more details and an account on how to check the conditions we refer the reader to Ruwet et al. (2012). We start dealing with snipped $k$-means:

**Theorem 2** *If $Y_n \in \mathcal{K}_{u,\rho}$, $k(k-1)+1 < n(1-\varepsilon)$, $u \in \mathcal{N}$ and $\rho > 0$ are such that $n-(1-\rho)(n(1-\varepsilon)-1) \leq u \leq n/k$, then for snipped $k$-means at level $\varepsilon$ we have that $\varepsilon^{(i)} \geq \varepsilon + 1/n$ and $\varepsilon^{(c)} \geq \varepsilon + 1/nd$.*

*Proof* As far as the individual breakdown point is concerned, a $\varepsilon$-trimmed procedure is a special case of a $\varepsilon$-snipped procedure. Given the assumptions, we can use the results in Ruwet et al. (2012) to show that $\varepsilon^{(i)} \geq \varepsilon + 1/n$. Regarding the cell breakdown point, note that the set of well clustered data sets is actually defined by addition of arbitrary rows. 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 Lemma 4 and Proposition 5 in Ruwet et al. (2012), where we set $r$ equal to the smallest feasible value such that $r \geq n(1-d\varepsilon)$. See also Gallegos and Ritter (2009a, 2009b). □

Well clustered data sets as in Definition 1 are focused on structural contamination. We speculate, and this is a direction for further work, that the same bound on cell breakdown of snipped $k$-means can be obtained under a more general definition of well clustered data sets. We now focus on other $k$-means type procedures. For what concerns classical $k$-means, it can be shown that even the restricted breakdown points are infinitesimal. Even with well clustered data sets, a single contaminated entry suffices to break down the estimates of the centroids. For what concerns trimmed $k$-means, Ruwet et al. (2012) showed under the same conditions of Theorem 2 that the individual breakdown point of trimmed $k$-means is larger than or equal to $\varepsilon + 1/n$. The robustness properties of trimmed $k$-means were also studied in García-Escudero and Gordaliza (1999). In the following proposition we give an additional contribution by dealing with the cell breakdown point of trimmed $k$-means:

**Theorem 3** *Assume the same conditions of Theorem 2 hold. For trimmed $k$-means at level $\varepsilon$ we have that $\varepsilon^{(c)} \geq \frac{\varepsilon}{d} + 1/nd$.*

*Proof* Note that contamination of $C$ entries corresponds to contamination of $C/d \leq R \leq C$ rows. Therefore, with a contamination of $\varepsilon$ entries we can spoil up to $d\varepsilon$ 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)). □

Theorem 3 says that trimmed $k$-means can achieve a cell breakdown point equivalent to snipped $k$-means only when the trimming level is set equal to $d\varepsilon$. This may be too large to be feasible, and is a consequence of the fact that trimming is not tailored to taking care of outliers in isolated entries.

## 3 Simulations

We outline here a brief simulation study to illustrate the performance and robustness of our approach. We fix $k=3$, $\sigma^2 = 1$ and we replicate data sampling, contamination and clustering $B = 1000$ times each time. We let $\mu_1 = (0,\ldots,0)'$, $\mu_2 = \mu_1 + c$ and $\mu_3 = \mu_2 + 2c$, for $c = 0.75, 1.5, 10$; corresponding to a small, moderate and large separation of the clusters. We contaminate from a mixture of a fully dependent and an independent component-wise contamination model. As a result, $\varepsilon_1 = 5$ % of the rows will be fully contaminated (structural outliers) and, for each column, a proportion $\varepsilon_2 = 5, 15$ % of entries will be independently selected for contamination. As a result, we have $\varepsilon = 10, 20$ %. We take $g_i(\cdot)$ as a product of uniform densities with support in $[-10c, 10c]$. In this way we include gross outliers, extreme values, bridge points, and inliers. Note that if the data matrix is standardized it may be unlikely in practice to have values of outliers as large as those corresponding to certain values of $c$. To complement the simulation, we also generate concentrated contamination, with $g_i(\cdot)$ as a product of Gaussian densities centered in $-c$ and with standard deviation $\sigma/2$. This generates a small cluster of outliers, rather than simple background noise. For this case we only use a total contamination of $\varepsilon = 10$ %, with $\varepsilon_1 = 5$ % as before. We select data matrices with sizes $n = 200, 1000$ and $d = 5, 50$. Each cluster label is sampled uniformly at random in $\{1,\ldots,k\}$, so that the size of each cluster is approximately $k/n$.

We evaluate clustering through the average sum of squares error (SSE) in estimation of the centroid. Furthermore, we 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 predicted cluster labels. We compute the modified Rand index only using samples that are not trimmed by any of the procedures involved, for a fair comparison. Finally, for trimmed and snipped $k$-means we evaluate misclassification of contaminated entries with the average *swamping* and *masking*. The

**Table 2** SSE in estimation of the centroid on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. The proportion of structural outliers is fixed at $\varepsilon_1 = 0.05$ and *conc* denotes concentrated contamination with $\varepsilon = 0.1$. The results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | $\varepsilon$ | kmeans | tkmeans($\varepsilon$) | tkmeans($\varepsilon_1 + d\varepsilon_2$) | snip($\varepsilon$) |
|---|---|---|---|---|---|---|---|
| 200 | 5 | 0.75 | 0.10 | 27.64 | 18.03 | 4.31 | 2.09 |
| 200 | 50 | 0.75 | 0.10 | 301.48 | 19.56 | – | 2.70 |
| 1000 | 5 | 0.75 | 0.10 | 19.52 | 13.65 | 2.09 | 0.60 |
| 1000 | 50 | 0.75 | 0.10 | 41.37 | 2.74 | – | 0.53 |
| 200 | 5 | 0.75 | 0.20 | 32.31 | 29.68 | – | 2.57 |
| 200 | 50 | 0.75 | 0.20 | 172.24 | 56.00 | – | 3.24 |
| 1000 | 5 | 0.75 | 0.20 | 24.05 | 29.28 | – | 0.77 |
| 1000 | 50 | 0.75 | 0.20 | 37.16 | 20.33 | – | 0.61 |
| 200 | 5 | 0.75 | *conc* | 3.31 | 3.45 | 3.52 | 2.81 |
| 200 | 50 | 0.75 | *conc* | 4.43 | 4.93 | – | 4.66 |
| 1000 | 5 | 0.75 | *conc* | 1.73 | 1.35 | 1.49 | 1.19 |
| 1000 | 50 | 0.75 | *conc* | 2.39 | 2.52 | | 0.87 |
| 200 | 5 | 1.50 | 0.10 | 174.44 | 112.99 | 0.51 | 0.34 |
| 200 | 50 | 1.50 | 0.10 | 1646.17 | 13.01 | – | 2.53 |
| 1000 | 5 | 1.50 | 0.10 | 121.90 | 79.55 | 0.10 | 0.07 |
| 1000 | 50 | 1.50 | 0.10 | 532.84 | 3.08 | – | 0.50 |
| 200 | 5 | 1.50 | 0.20 | 160.74 | 132.60 | – | 0.38 |
| 200 | 50 | 1.50 | 0.20 | 1268.21 | 105.84 | – | 2.81 |
| 1000 | 5 | 1.50 | 0.20 | 137.03 | 127.95 | – | 0.08 |
| 1000 | 50 | 1.50 | 0.20 | 172.23 | 17.12 | – | 0.56 |
| 200 | 5 | 1.50 | *conc* | 1.36 | 1.32 | 0.90 | 1.39 |
| 200 | 50 | 1.50 | *conc* | 49.53 | 9.51 | – | 5.20 |
| 1000 | 5 | 1.50 | *conc* | 0.80 | 0.87 | 0.25 | 0.67 |
| 1000 | 50 | 1.50 | *conc* | 44.27 | 6.65 | – | 0.75 |
| 200 | 5 | 10 | 0.10 | 6826.56 | 4107.40 | 0.36 | 0.24 |
| 200 | 50 | 10 | 0.10 | 82058.94 | 440.21 | – | 2.53 |
| 1000 | 5 | 10 | 0.10 | 4977.93 | 1062.62 | 0.07 | 0.05 |
| 1000 | 50 | 10 | 0.10 | 26746.52 | 116.25 | – | 0.50 |
| 200 | 5 | 10 | 0.20 | 7503.50 | 6035.05 | – | 0.28 |
| 200 | 50 | 10 | 0.20 | 65264.91 | 3914.52 | – | 2.81 |
| 1000 | 5 | 10 | 0.20 | 6678.39 | 5704.37 | – | 0.06 |
| 1000 | 50 | 10 | 0.20 | 7289.96 | 706.86 | – | 0.56 |
| 200 | 5 | 10 | *conc* | 457.30 | 18.77 | 0.36 | 0.25 |
| 200 | 50 | 10 | *conc* | 3550.96 | 178.92 | – | 2.54 |
| 1000 | 5 | 10 | *conc* | 400.53 | 18.53 | 0.07 | 0.05 |
| 1000 | 50 | 10 | *conc* | 2356.06 | 146.20 | – | 0.51 |

first is the proportion of clean entries that are included in the set of trimmed/snipped values, while the second is the proportion of contaminated entries that are not discarded and used for estimation. Note that with this contamination model, the two coincide for snipped $k$-means but not for trimmed $k$-means. We compare our procedure with snipping level $\varepsilon$ with classical $k$-means, trimmed $k$-means with trimming levels $\varepsilon$ and $\varepsilon_1 + d\varepsilon_2$ when feasible. We should actually use a trimming level of $d\varepsilon$, but that is never feasible for our choices of $d$ and $\varepsilon$. Therefore, we use prior information on the proportion of structural outliers to select a trimming level. We report the results on SSE in Table 2, the results on the modified Rand index in Table 3 and the results on misclassification of contaminated entries in Table 4.

**Table 3** Modified Rand index on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. The proportion of structural outliers is fixed at $\varepsilon_1 = 0.05$ and *conc* denotes concentrated contamination with $\varepsilon = 0.1$. The results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | $\varepsilon$ | kmeans | tkmeans($\varepsilon$) | tkmeans($\varepsilon_1 + d\varepsilon_2$) | snip($\varepsilon$) |
|---|---|---|---|---|---|---|---|
| 200 | 5 | 0.75 | 0.10 | 0.18 | 0.19 | 0.26 | 0.31 |
| 200 | 50 | 0.75 | 0.10 | 0.45 | 0.52 | – | 0.98 |
| 1000 | 5 | 0.75 | 0.10 | 0.18 | 0.19 | 0.29 | 0.33 |
| 1000 | 50 | 0.75 | 0.10 | 0.61 | 0.62 | – | 0.98 |
| 200 | 5 | 0.75 | 0.20 | 0.08 | 0.11 | – | 0.28 |
| 200 | 50 | 0.75 | 0.20 | 0.33 | 0.23 | – | 0.97 |
| 1000 | 5 | 0.75 | 0.20 | 0.10 | 0.12 | – | 0.30 |
| 1000 | 50 | 0.75 | 0.20 | 0.35 | 0.30 | – | 0.96 |
| 200 | 5 | 0.75 | *conc* | 0.25 | 0.21 | 0.14 | 0.22 |
| 200 | 50 | 0.75 | *conc* | 0.86 | 0.73 | – | 0.87 |
| 1000 | 5 | 0.75 | *conc* | 0.27 | 0.23 | 0.14 | 0.24 |
| 1000 | 50 | 0.75 | *conc* | 0.87 | 0.74 | – | 0.88 |
| 200 | 5 | 1.50 | 0.10 | 0.34 | 0.40 | 0.46 | 0.79 |
| 200 | 50 | 1.50 | 0.10 | 0.53 | 0.87 | – | 1.00 |
| 1000 | 5 | 1.50 | 0.10 | 0.35 | 0.43 | 0.47 | 0.80 |
| 1000 | 50 | 1.50 | 0.10 | 0.53 | 0.88 | – | 1.00 |
| 200 | 5 | 1.50 | 0.20 | 0.15 | 0.21 | – | 0.75 |
| 200 | 50 | 1.50 | 0.20 | 0.42 | 0.41 | – | 1.00 |
| 1000 | 5 | 1.50 | 0.20 | 0.17 | 0.20 | – | 0.76 |
| 1000 | 50 | 1.50 | 0.20 | 0.53 | 0.52 | – | 1.00 |
| 200 | 5 | 1.50 | *conc* | 0.58 | 0.55 | 0.61 | 0.62 |
| 200 | 50 | 1.50 | *conc* | 0.85 | 0.79 | – | 0.91 |
| 1000 | 5 | 1.50 | *conc* | 0.59 | 0.57 | 0.62 | 0.65 |
| 1000 | 50 | 1.50 | *conc* | 0.85 | 0.79 | – | 0.90 |
| 200 | 5 | 10 | 0.10 | 0.43 | 0.53 | 0.57 | 0.99 |
| 200 | 50 | 10 | 0.10 | 0.48 | 0.89 | – | 1.00 |
| 1000 | 5 | 10 | 0.10 | 0.43 | 0.53 | 0.57 | 0.99 |
| 1000 | 50 | 10 | 0.10 | 0.58 | 0.90 | – | 1.00 |
| 200 | 5 | 10 | 0.20 | 0.15 | 0.24 | – | 0.99 |
| 200 | 50 | 10 | 0.20 | 0.43 | 0.45 | – | 1.00 |
| 1000 | 5 | 10 | 0.20 | 0.16 | 0.24 | – | 0.99 |
| 1000 | 50 | 10 | 0.20 | 0.55 | 0.54 | – | 1.00 |
| 200 | 5 | 10 | *conc* | 0.63 | 0.78 | 0.82 | 0.91 |
| 200 | 50 | 10 | *conc* | 0.83 | 0.93 | – | 1.00 |
| 1000 | 5 | 10 | *conc* | 0.62 | 0.77 | 0.86 | 0.90 |
| 1000 | 50 | 10 | *conc* | 0.86 | 0.93 | – | 1.00 |

The results in Table 2 clearly show that insufficient (or no) trimming may lead to break down of the estimates. On the other hand, when $\varepsilon$ or $d$ is large, sufficient trimming is not feasible. Furthermore, when a trimming level equal to $\varepsilon_1 + d\varepsilon_2$ is feasible, trimmed $k$-means results in a slightly larger SSE than snipped $k$-means. This is due to the fact that a significant fraction of clean entries is discarded in order to get rid of the contaminated ones. This is an illustration that trimmed $k$-means may be able to deal with component-wise contamination, but at the price of a much larger trimming level. More precisely, when $d = 5$ and $\varepsilon_2 = 5$ %, a snipping level of 10 % is compared to a trimming level of 30 %. Under a completely general component-wise contamination model we should actually use a trimming level of 50 %.

**Table 4** Swamping probability (and masking probability in parentheses, when not equal to the former) on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. The proportion of structural outliers is fixed at $\varepsilon_1 = 0.05$ and *conc* denotes concentrated contamination with $\varepsilon = 0.1$. The results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | $\varepsilon$ | `tkmeans`($\varepsilon$) | `tkmeans`($\varepsilon_1 + d\varepsilon_2$) | `snip`($\varepsilon$) |
|---|---|---|---|---|---|---|
| 200 | 5 | 0.75 | 0.10 | 0.0811 (0.0749) | 0.4238 (0.0139) | 0.0091 |
| 200 | 50 | 0.75 | 0.10 | 0.0895 (0.0819) | – | 0.0009 |
| 1000 | 5 | 0.75 | 0.10 | 0.0796 (0.0736) | 0.4196 (0.0136) | 0.0008 |
| 1000 | 50 | 0.75 | 0.10 | 0.0890 (0.0801) | – | 0.0000 |
| 200 | 5 | 0.75 | 0.20 | 0.1319 (0.1241) | – | 0.0102 |
| 200 | 50 | 0.75 | 0.20 | 0.1496 (0.1396) | – | 0.0015 |
| 1000 | 5 | 0.75 | 0.20 | 0.1294 (0.1195) | | 0.0016 |
| 1000 | 50 | 0.75 | 0.20 | 0.1494 (0.1394) | – | 0.0001 |
| 200 | 5 | 0.75 | *conc* | 0.0933 (0.0901) | 0.2819 (0.0744) | 0.0779 |
| 200 | 50 | 0.75 | *conc* | 0.0931 (0.0893) | – | 0.0777 |
| 1000 | 5 | 0.75 | *conc* | 0.0911 (0.0899) | 0.2831 (0.0736) | 0.0764 |
| 1000 | 50 | 0.75 | *conc* | 0.0909 (0.0903) | – | 0.0489 |
| 200 | 5 | 1.50 | 0.10 | 0.0442 (0.0366) | 0.2088 (0.0064) | 0.0036 |
| 200 | 50 | 1.50 | 0.10 | 0.0440 (0.0365) | – | 0.0004 |
| 1000 | 5 | 1.50 | 0.10 | 0.0438 (0.0371) | 0.2087 (0.0063) | 0.0004 |
| 1000 | 50 | 1.50 | 0.10 | 0.0459 (0.0425) | – | 0.0000 |
| 200 | 5 | 1.50 | 0.20 | 0.1065 (0.0940) | – | 0.0036 |
| 200 | 50 | 1.50 | 0.20 | 0.1233 (0.1108) | – | 0.0002 |
| 1000 | 5 | 1.50 | 0.20 | 0.1039 (0.0954) | – | 0.0005 |
| 1000 | 50 | 1.50 | 0.20 | 0.1201 (0.1116) | – | 0.0000 |
| 200 | 5 | 1.50 | *conc* | 0.0838 (0.0733) | 0.2384 (0.0346) | 0.0579 |
| 200 | 50 | 1.50 | *conc* | 0.0941 (0.0844) | – | 0.0366 |
| 1000 | 5 | 1.50 | *conc* | 0.0836 (0.0811) | 0.2381 (0.0290) | 0.0576 |
| 1000 | 50 | 1.50 | *conc* | 0.0933 (0.0901) | – | 0.0287 |
| 200 | 5 | 10 | 0.10 | 0.0439 (0.0366) | 0.2046 (0.0010) | 0.0016 |
| 200 | 50 | 10 | 0.10 | 0.0436 (0.0362) | – | 0.0001 |
| 1000 | 5 | 10 | 0.10 | 0.0407 (0.0372) | 0.2036 (0.0010) | 0.0002 |
| 1000 | 50 | 10 | 0.10 | 0.0459 (0.0425) | – | 0.0000 |
| 200 | 5 | 10 | 0.20 | 0.1062 (0.0937) | – | 0.0014 |
| 200 | 50 | 10 | 0.20 | 0.1233 (0.1107) | – | 0.0001 |
| 1000 | 5 | 10 | 0.20 | 0.1036 (0.0951) | – | 0.0002 |
| 1000 | 50 | 10 | 0.20 | 0.1201 (0.1115) | – | 0.0000 |
| 200 | 5 | 10 | *conc* | 0.0821 (0.0699) | 0.2038 (0.0008) | 0.0019 |
| 200 | 50 | 10 | *conc* | 0.0536 (0.0441) | – | 0.0001 |
| 1000 | 5 | 10 | *conc* | 0.0803 (0.0746) | 0.2041 (0.0007) | 0.0001 |
| 1000 | 50 | 10 | *conc* | 0.0511 (0.0379) | – | 0.0000 |

For snipped $k$-means the SSE for $\varepsilon = 20$ % is only slightly larger than the SSE for $\varepsilon = 10$ %, indicating that the larger contamination is only contributing to a loss of information due to a lower number of clean entries overall. From Table 3 we can see that the modified Rand index of snipped $k$-means is always acceptable, being 99 % or 100 % with $c = 10$ (large group separation), and when $d = 50$ also with $c = 1.5$ and $c = 0.75$. When $d = 5$, all other procedures have a Rand index which is 30 % to 70 % smaller. Only when $\varepsilon = 10$ % and $d = 50$ trimmed $k$-means is able to correctly predict the true clustering, with a Rand index of approximately 90 %. Finally, we comment on misclassification of contaminated and clean entries. For what concerns trimmed $k$-means at level $\varepsilon$, we have that misclassification is approx-

**Table 5** SSE and Rand index on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. No structural outliers are generated and the proportion of component-wise outliers, generated from independent Gaussians centered on $-c$ and with standard deviation $\sigma/2$, is $\varepsilon_2 = 0.1$. Results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | SSE | | |
|---|---|---|---|---|---|
| | | | `kmeans` | `tkmeans` (10 %) | `snip` (10 %) |
| 200 | 5 | 0.75 | 4.61 | 4.87 | 3.87 |
| 200 | 50 | 0.75 | 6.43 | 6.95 | 4.50 |
| 1000 | 5 | 0.75 | 3.72 | 3.70 | 1.90 |
| 1000 | 50 | 0.75 | 4.23 | 4.22 | 0.71 |
| 200 | 5 | 1.50 | 3.15 | 1.24 | 0.66 |
| 200 | 50 | 1.50 | 27.52 | 16.84 | 3.57 |
| 1000 | 5 | 1.50 | 0.56 | 0.42 | 0.23 |
| 1000 | 50 | 1.50 | 17.03 | 13.15 | 0.60 |
| 200 | 5 | 10 | 541.42 | 26.20 | 0.25 |
| 200 | 50 | 10 | 2105.00 | 614.53 | 2.53 |
| 1000 | 5 | 10 | 480.86 | 19.55 | 0.05 |
| 1000 | 50 | 10 | 1600.67 | 547.68 | 0.50 |

| $n$ | $d$ | $c$ | Rand index | | |
|---|---|---|---|---|---|
| | | | `kmeans` | `tkmeans` (10 %) | `snip` (10 %) |
| 200 | 5 | 0.75 | 0.24 | 0.21 | 0.22 |
| 200 | 50 | 0.75 | 0.92 | 0.81 | 0.96 |
| 1000 | 5 | 0.75 | 0.25 | 0.22 | 0.25 |
| 1000 | 50 | 0.75 | 0.94 | 0.83 | 0.97 |
| 200 | 5 | 1.50 | 0.52 | 0.55 | 0.69 |
| 200 | 50 | 1.50 | 0.95 | 0.89 | 1.00 |
| 1000 | 5 | 1.50 | 0.56 | 0.57 | 0.72 |
| 1000 | 50 | 1.50 | 0.99 | 0.89 | 1.00 |
| 200 | 5 | 10 | 0.56 | 0.77 | 1.00 |
| 200 | 50 | 10 | 0.85 | 0.89 | 1.00 |
| 1000 | 5 | 10 | 0.58 | 0.76 | 1.00 |
| 1000 | 50 | 10 | 0.87 | 0.89 | 1.00 |

imately 5 % when $\varepsilon = 10$ % and 10 % when $\varepsilon = 20$ %. Most of the structural outliers are trimmed, but only a small fraction of component-wise outliers are discarded with trimming. When the trimming level is $\varepsilon_1 + d\varepsilon_2$, masking is approximately 0.6 % when $c = 1.5$ and 0.1 % when $c = 10$. Consequently, almost all outliers are discarded, but at the price of a large swamping, which is around 20 %. Finally, we can see snipping allows to obtain a very small probability of swamping and masking in all scenarios; and note that the challenging concentrated contamination setting gives rise to similar considerations as the background noise setting with respect to SSE, Rand index and misclassification.

It can be noted that the concentrated contamination setting with $\varepsilon_1 > 0$ is adding a small cluster of outliers, which could be easy to identify. In order to illustrate the performance of procedures when this cluster is not present, we show in Tables 5 and 6 results on concentrated contamination with $\varepsilon_1 = 0$ and $\varepsilon_2 > 0$, that is, with CWC only. The procedures behave similarly to the previous settings.

We conclude this section by comparing the clustering procedures in an outlier free setting. We generate data as before, but without any contamination whatsoever. We fix the trimming and snipping levels at 10 %. The results of this simulation study are important to evaluate the loss of efficiency when robust procedures are used on clean datasets. The results are summarized in Table 7.

From Table 7 it can be seen that only a mild loss of efficiency, in terms of SSE, is expected with trimming and snipping. The SSE of snipped $k$-means is often slightly larger than that of trimmed $k$-means. On the other hand, for the Rand index this comparison is often reversed.

**Table 6** Swamping probability (and masking probability in parentheses, when not equal to the former) on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. No structural outliers are generated and the proportion of component-wise outliers, generated from independent Gaussians centered on $-c$ and with standard deviation $\sigma/2$, is $\varepsilon_2 = 0.1$. Results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | `tkmeans` (10 %) | `snip` (10 %) |
|---|---|---|---|---|
| 200.00 | 5.00 | 0.75 | 0.0900 (0.0850) | 0.0750 |
| 200.00 | 50.00 | 0.75 | 0.0920 (0.0870) | 0.0580 |
| 1000.00 | 5.00 | 0.75 | 0.0864 (0.0855) | 0.0702 |
| 1000.00 | 50.00 | 0.75 | 0.0883 (0.0873) | 0.0302 |
| 200.00 | 5.00 | 1.5 | 0.0778 (0.0728) | 0.0377 |
| 200.00 | 50.00 | 1.5 | 0.0898 (0.0848) | 0.0244 |
| 1000.00 | 5.00 | 1.5 | 0.0746 (0.0736) | 0.0331 |
| 1000.00 | 50.00 | 1.5 | 0.0862 (0.0852) | 0.0096 |
| 200.00 | 5.00 | 10 | 0.0780 (0.0729) | 0.0000 |
| 200.00 | 50.00 | 10 | 0.0895 (0.0845) | 0.0000 |
| 1000.00 | 5.00 | 10 | 0.0749 (0.0739) | 0.0000 |
| 1000.00 | 50.00 | 10 | 0.0859 (0.0849) | 0.0000 |

**Table 7** SSE and Rand index on simulated data for different clustering procedures. `kmeans` denotes the classical $k$-means procedure, `tkmeans`($\eta$) trimmed $k$-means with trimming level $\eta$, and `snip` snipped $k$-means. Results are based on $B = 1000$ replications

| $n$ | $d$ | $c$ | $\varepsilon$ | SSE | | |
|---|---|---|---|---|---|---|
| | | | | `kmeans` | `tkmeans` (10 %) | `snip` (10 %) |
| 200 | 5 | 0.75 | 0.00 | 2.94 | 3.02 | 2.63 |
| 200 | 50 | 0.75 | 0.00 | 2.31 | 2.71 | 3.30 |
| 1000 | 5 | 0.75 | 0.00 | 1.21 | 1.32 | 0.91 |
| 1000 | 50 | 0.75 | 0.00 | 0.46 | 0.53 | 0.56 |
| 200 | 5 | 1.50 | 0.00 | 0.31 | 0.38 | 0.47 |
| 200 | 50 | 1.50 | 0.00 | 2.28 | 2.66 | 3.23 |
| 1000 | 5 | 1.50 | 0.00 | 0.06 | 0.07 | 0.09 |
| 1000 | 50 | 1.50 | 0.00 | 0.44 | 0.52 | 0.55 |
| 200 | 5 | 10 | 0.00 | 0.23 | 0.30 | 0.39 |
| 200 | 50 | 10 | 0.00 | 2.28 | 2.66 | 3.22 |
| 1000 | 5 | 10 | 0.00 | 0.05 | 0.06 | 0.07 |
| 1000 | 50 | 10 | 0.00 | 0.46 | 0.53 | 0.56 |

| $n$ | $d$ | $c$ | $\varepsilon$ | Rand index | | |
|---|---|---|---|---|---|---|
| | | | | `kmeans` | `tkmeans` (10 %) | `snip` (10 %) |
| 200 | 5 | 0.75 | 0.00 | 0.32 | 0.26 | 0.26 |
| 200 | 50 | 0.75 | 0.00 | 0.98 | 0.92 | 0.96 |
| 1000 | 5 | 0.75 | 0.00 | 0.34 | 0.29 | 0.28 |
| 1000 | 50 | 0.75 | 0.00 | 0.98 | 0.93 | 0.97 |
| 200 | 5 | 1.50 | 0.00 | 0.81 | 0.68 | 0.69 |
| 200 | 50 | 1.50 | 0.00 | 1.00 | 1.00 | 1.00 |
| 1000 | 5 | 1.50 | 0.00 | 0.82 | 0.70 | 0.72 |
| 1000 | 50 | 1.50 | 0.00 | 1.00 | 1.00 | 1.00 |
| 200 | 5 | 10 | 0.00 | 1.00 | 1.00 | 1.00 |
| 200 | 50 | 10 | 0.00 | 1.00 | 1.00 | 1.00 |
| 1000 | 5 | 10 | 0.00 | 1.00 | 1.00 | 1.00 |
| 1000 | 50 | 10 | 0.00 | 1.00 | 1.00 | 1.00 |

**Table 8** Recognition of handwritten digits data: modified Rand index and conditional probability of correct classification $\Pr(j|j)$, $j = 0, \ldots, 9$, for different clustering procedures

| Method | Rand | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$-means | 67 % | 0.37 | 0.76 | 0.88 | 0.71 | 0.43 | 0.97 | 0.95 | 0.82 | 0.44 | 1.00 |
| PAM | 64 % | 0.57 | 0.89 | 0.89 | 0.78 | 0.72 | 0.98 | 0.95 | 0.44 | 0.75 | 0.99 |
| `tkmeans` (5 %) | 71 % | 0.64 | 0.90 | 0.88 | 0.89 | 0.80 | 0.99 | 0.97 | 0.86 | 0.67 | 0.99 |
| `snip` (5 %) | 83 % | 0.87 | 0.91 | 0.91 | 0.95 | 0.76 | 0.99 | 0.99 | 0.90 | 0.85 | 0.99 |
| `tkmeans` (10 %) | 75 % | 0.61 | 0.93 | 0.92 | 0.92 | 0.83 | 0.99 | 0.98 | 0.84 | 0.79 | 1.00 |
| `snip` (10 %) | 85 % | 0.93 | 0.93 | 0.92 | 0.95 | 0.80 | 0.99 | 0.99 | 0.90 | 0.82 | 0.99 |

## 4 Optical recognition of handwritten digits

We describe here an example regarding optical recognition of handwritten digits. Data are freely available from the UCI Matchine Learning Repository (Frank and Asuncion 2010) website http://archive.ics.uci.edu/ml/ and describe $n = 3823$ digits, from `0` to `9` (hence, $k = 10$), handwritten by 30 subjects. The resulting bitmaps are divided into nonoverlapping blocks and the number of pixels are counted in each block. This generates $d = 64$ variables, recording the normalized count of pixels in each block. We ignore the known labeling, and therefore perform a *confirmatory* cluster analysis. In this application it is easy to realize that there may be a mix of structural and component-wise outliers, with possibly an heavy presence of component-wise outliers. Structural outliers may be given by the orthographics habits of certain subjects, who can draw slightly different digits with respect to the rest of the population. On the other hand, component-wise outliers are given by even a only slightly different draw with respect to the usual, for instance due to nervous handwriting.

We begin reporting the computation time with such a large data matrix. Regardless of the snipping level, with our non-optimized code it takes about 2.5 hours to perform $2e10^5$ iterations of the algorithm on a laptop with a 2.40 GHz CPU and 512 Mb RAM. To evaluate convergence, we repeatedly used the algorithm from 100 different initial solutions. The same smallest loss was obtained for 37 of those. We underline that we expect a much lower dependence on the initial solution for smaller data matrices. In light of these results we can state anyway that 7 runs from different initial solutions are enough in order to achieve approximately 95 % probability of attaining the global optimum, and 10 for a 99 % probability.

In Table 8 we report the modified Rand index and, for each digit, the conditional probability of correct classification for different procedures. The latter is obtained dividing the diagonal of the classification matrix by the frequency table of true labels. The operation is intended entry-wise. Cluster labels are permuted in order to maximize the sum of the elements in the diagonal of the classification matrix. As with the simulated data, these values are computed by using samples that are not trimmed by any of the procedures involved.

From Table 8 we can see that with the same number of discarded entries snipping can achieve a Rand index which is substantially larger than trimmed $k$-means, due to the additional flexibility in arranging them. There is a substantial improvement also for what concerns the misclassification error. There are patterns of misclassification, which snipping is able to partially overcome. The most important example is given by the `0`, which seems to be much better classified with snipping. This is probably because minimal scratches of the handwriting can make it look like a `6`, a `9` or even an `8`. Trimming on the other hand seems to be particularly useful for the digit `4`, which is the only that is correctly classified less often with snipping than with trimming. This probably happens since by removing some entries few handwritten `4` can look like a `1`. As noted by a referee, some kind of curve registration may be used in connection with this data set, and a reason behind the better performance of snipping in comparison with trimmed $k$-means or PAM is that snipping may be automatically doing part of this registration task.

## 5 Conclusions

It is now clear in the robust clustering literature that trimming and clustering must be performed simoultaneously. It can be argued along the same lines that component-wise outliers can not be tackled with univariate outlier identification methods, screening variables one at a time. An outlier is in fact an extreme point *with respect to its own centroid*. Identification of bridge points between two clusters may not be doable with univariate screening.

We give three key contributions to the robust cluttering literature in this paper. First of all, we introduce the idea of snipping, which can be used to tackle component-wise contamination. Isolated aberrant values can be discarded and the corresponding subject can still be assigned to a cluster and contribute, at least in part, to centroid estimation. Snipping

is completely flexible, so that any configuration of contaminated entries can be taken care and potentially all observations can be snipped. If for instance we have a structural *column* outlier, snipping can discard it. On the other hand, the entire data set would have to be discarded if using trimming. This situation can also be taken care with the robust double clustering approach of Farcomeni (2009), which is anyway much less flexible since in Farcomeni (2009) the same dimensions must be snipped in all rows. Secondly, along the lines of Chakraborty and Chaudhury (2008), we developed a loss minimization algorithm with guaranteed convergence to the global optimum. This algorithm can be directly employed also with trimming methods, with minor adjustments (e.g., that the candidate solution $V_c$ is obtained by switching two *rows* of the current solution $V(t)$). A detailed comparison of the two possible algorithms (the one based on concentration steps and the one based on stochastic optimization) in terms of performance and computation time is also grounds for further work. Third, in parallel with Alqallaf et al. (2009) we introduce a crisp general contamination model which includes structural and component-wise outliers simoultaneously. We have assumed, in parallel with the clustering partition, that entries of the data matrix can be partitioned into clean and contaminated entries. This general contamination model is particularly challenging and includes many situations of interest (e.g., structural column outliers, structural row outliers and any fixed pattern of contamination) as special cases.

It must be here acknowledged that the proposed contamination and the snipped $k$-means procedure are *not* rotation invariant; while on the other hand trimming procedures (and the Tukey–Huber contamination model) do have this property. An interesting discussion in Alqallaf et al. (2009) gives more insights on this point.

There are many possibilities for further work. Methods for heterogeneous robust clustering (e.g., García-Escudero et al. 2008; Gallegos and Ritter 2005, 2009b, 2010) will be discussed in a companion paper (Farcomeni 2013). Another issue regards the formal robustness properties. We have shown that the restricted individual and cell breakdown points of snipped $k$-means are bounded from below by the snipping level essentially adapting results and definitions of well-clustered data sets from Ruwet et al. (2012). We believe the same results hold, at least regarding the cell breakdown point, under a more general class of well-clustered data sets. It would be also interesting to study the dissolution points and isolation robustness of snipping as well, as proposed in Hennig (2008). The concepts in Hennig (2008) could be used to better underline the inherent differences between CWC and THC in cluster analysis.

We finally note that the idea of snipping is applicable in settings also outside the robust clustering area.

**Acknowledgements** The author is grateful to an AE and two anonymous referees for very kind suggestions.

## References

Alqallaf, F., Van Aelst, S., Yohai, V.J., Zamar, R.H.: Propagation of outliers in multivariate data. Ann. Stat. **37**, 311–331 (2009)

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

Chakraborty, B., Chaudhury, P.: On an optimization problem in robust statistics. J. Comput. Graph. Stat. **17**, 683–702 (2008)

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

Donoho, D., Huber, P.: The notion of breakdown point. In: Bickel, P., Doksum, K., Hodges, J. (eds.) A Festschirift for Erich L. Lehmann, Wadsworth, Belmont, CA, pp. 157–184 (1983)

Farcomeni, A.: Robust double clustering: a method based on alternating concentration steps. J. Classif. **26**, 77–101 (2009)

Farcomeni, A.: Robust constrained clustering in presence of entry-wise outliers. Technometrics (2013, to appear)

Farcomeni, A., Ventura, L.: An overview of robust methods in medical research. Stat. Methods Med. Res. **21**, 111–133 (2012)

Forero, P.A., Kekatos, V., Giannakis, G.B.: Robust clustering using outlier-sparsity regularization. IEEE Trans. Signal Process. **60**, 4163–4177 (2012)

Frank, A., Asuncion, A.: UCI machine learning repository (2010). http://archive.ics.uci.edu/ml

Frühwirth-Schnatter, S., Pyne, S.: Bayesian inference for finite mixtures of univariate skew-normal and skew-t distributions. Biostatistics **11**, 317–336 (2010)

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

Gallegos, M., Ritter, G.: Trimmed ML estimation of contaminated mixtures. Sankhya **71**, 164–220 (2009a)

Gallegos, M., Ritter, G.: Trimming algorithms for clustering contaminated grouped data and their robustness. Adv. Data Anal. Classif. **3**, 135–167 (2009b)

Gallegos, M., Ritter, G.: Using combinatorial optimization in model-based trimmed clustering with cardinality constraints. Comput. Stat. Data Anal. **54**, 637–654 (2010)

García-Escudero, L., Gordaliza, A.: Robustness properties of $k$ means and trimmed $k$ means. J. Am. Stat. Assoc. **94**, 956–969 (1999)

García-Escudero, L.A., Gordaliza, A., Matrán, C., Mayo-Iscar, A.: A general trimming approach to robust cluster analysis. Ann. Stat. **36**, 1324–1345 (2008)

García-Escudero, L.A., Gordaliza, A., Matrán, C., Mayo-Iscar, A.: A review of robust clustering methods. Adv. Data Anal. Classif. **4**, 89–109 (2010)

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

Gordaliza, A.: Best approximations to random variables based on trimming procedures. J. Approx. Theory **64**, 162–180 (1991)

Hampel, F.: A general qualitative definition of robustness. Ann. Math. Stat. **42**, 1887–1896 (1971)

Hampel, F., Rousseeuw, P., Ronchetti, E., Stahel, W.: Robust Statistics: the Approach Based on the Influence Function. Wiley, New York (1986)

Hennig, C.: Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. J. Multivar. Anal. **99**, 11541176 (2008)

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

Hodges, J.: Efficiency in normal samples and tolerance of extreme values for some estimates of location. In: Proc. Fifth Berkeley Symp. Math. Statist. Probab., vol. 1, pp. 163–186. University of California Press, Berkeley (1967)

Huber, P., Ronchetti, E.: Robust Statistics. Wiley, New York (2009)

Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. **35**, 73–101 (1964)

Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)

Hubert, M., Rousseeuw, P., Van Aelst, S.: High-breakdown robust multivariate methods. Stat. Sci. **23**, 92–119 (2008)

Kaufman, L., Rousseeuw, P.: Finding Groups in Data. Wiley, New York (1990)

Rousseeuw, P., Van Driessen, K.: A fast algorithm for the minimum covariance determinant estimator. Technometrics **41**, 212–223 (1999)

Ruwet, C., Garcia-Escudero, L., Gordaliza, A., Mayo-Iscar, A.: On the breakdown behavior of robust constrained clustering procedures. TEST (2012, to appear)

Tukey, J.W.: The future of data analysis. Ann. Math. Stat. **33**, 167 (1962)