Computational Statistics and Data Analysis 167 (2022) 107370

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

www.elsevier.com/locate/csda

K-bMOM: A robust Lloyd-type clustering algorithm based on bootstrap median-of-means

Camille Brunet-Saumard [a], Edouard Genetay [b,c], Adrien Saumard [b,*]

[a] *twice.ai, Rennes, France*
[b] *Université de Rennes, Ensai, CREST-UMR 9194, Rennes F-35000, France*
[c] *LumenAI, Tours, France*

ARTICLE INFO

Article history:
Received 10 December 2020
Received in revised form 3 September 2021
Accepted 8 October 2021
Available online 15 October 2021

Keywords:
Bootstrap
Breakdown point
Color quantization
Adversarial contamination
Median-of-means
Robust clustering

ABSTRACT

The median-of-means is an estimator of the mean of a random variable that has emerged as an efficient and flexible tool to design robust learning algorithms with optimal theoretical guarantees. However, its use for the clustering task suggests dividing the dataset into blocks, which may provoke the disappearance of some clusters in some blocks and lead to bad performances. To overcome this difficulty, a procedure termed "bootstrap median-of-means" is proposed, where the blocks are generated with a replacement in the dataset. Considering the estimation of the mean of a random variable, the bootstrap median-of-means has a better breakdown point than the median-of-means if enough blocks are generated. A clustering algorithm called K-bMOM is designed, by performing Lloyd-type iterations together with the use of the bootstrap median-of-means strategy. Good performances are obtained on simulated and real-world datasets for color quantization and an emphasis is put on the benefits of our robust intialization procedure. On the theoretical side, K-bMOM is also proven to have a non-trivial probabilistic breakdown point in well-clusterizable situations.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Massive and complex datasets are often corrupted by outliers. Classical data mining procedures such as K-means or more general EM algorithms for instance, are however, sensitive to the presence of outliers, which can induce time consuming data pre-processing.

In this context, robust versions of data mining procedures are particularly relevant and we investigate a way to produce a Lloyd-type algorithm for hard clustering that is robust with respect to the presence of outliers. We propose more precisely using a variant of the median-of-means (MOM) strategy, that we call "bootstrap median-of-means" (bMOM). The MOM principle has been the object of recent intensive research in mean estimation, regression, high-dimensional framework and supervised classification and machine learning (Lerasle and Oliveira, 2011; Devroye et al., 2016; Lecué and Lerasle, 2019, 2020; Lugosi and Mendelson, 2019a,b, 2020; Minsker, 2018). Other approaches to robustness for K-means also exist in the literature, such as for instance, K-median or trimmed K-means (García-Escudero et al., 2010; Brécheteau, 2018) to name but a few. The design of robust estimators with a control of the algorithmic complexity has also been investigated (Diakonikolas and Kane, 2019).

* Corresponding author.
E-mail addresses: csaumard@twice.ai (C. Brunet-Saumard), edouard.genetay@ensai.fr (E. Genetay), adrien.saumard@ensai.fr (A. Saumard).

https://doi.org/10.1016/j.csda.2021.107370
0167-9473/© 2021 Elsevier B.V. All rights reserved.

Given a dataset, bMOM consists of first generating a (large) bootstrap sample and then performing a classical median-of-means on this bootstrap sample. This can be seen also as a so-called subragging procedure - for "sub-sample robust aggregating" - in the terminology of Bühlmann (2003). We prove in Section 3.1 that if enough blocks are generated from the bootstrap sampling, then for a fixed block size, bMOM has a higher breakdown point than MOM. In other words, bMOM is more robust to contamination than the classical MOM. Note that one strength of bMOM, that will be very useful in the context of clustering, is that sampling is done *with replacement* when constructing the blocks. Hence, the number of blocks for a fixed length is not limited by the amount of initial data, unlike MOM or its variant by sampling without replacement (Laforgue et al., 2019).

We propose a robust-to-outliers version of K-means, that we call K-bMOM, and that performs Lloyd-type iterations through the use of bMOM estimates for the K-means risk, as further explained in Section 2. In that section, a robust variant to the traditional K-means++ initialization strategy by applying the bMOM strategy is also presented.

Theoretical results are summarized in Section 3. We prove in particular in Section 3.2, that the K-bMOM algorithm is robust in a sense of a probabilistic version of a breakdown point if the initial data is in a well-clusterizable situation. This is very much in line with the results on the trimmed K-means for example (García-Escudero and Gordaliza, 1999). Further theoretical results, concerning an idealized version of our algorithm, can be found in a Supplementary Material (Brunet-Saumard et al., 2022).

In Section 4, the scope of application of K-bMOM is illustrated and practical considerations and guidelines are provided for choosing the number and size of the blocks. In Section 5, the proposed initialization procedure and the K-bMOM approach are tested in several simulation settings of outliers. It is also compared to existing robust K-means based clustering approaches in Section 6. And finally, this algorithm is applied to the well-known problem of color quantization in the image processing field.

Our framework is close to the recent work (Klochkov et al., 2020) that investigates the use of median-of-means statistics to produce a robust K-means type clustering. However, the latter work is theoretical only and the authors study probabilistic performance bounds for the minimizer of the median-of-means of the K-means distortion loss under a finite second moment assumption. In particular the authors do not discuss the use of median-of-means through Lloyd-type iterations nor a practical way to compute the estimator. Neither do they discuss the possibility of generating blocks with replacements in the dataset.

2. K-bMOM procedure

We recall first in Section 2.1, the Median-of-Means procedure and introduce a variant, called bootstrap Median-of-Means (bMOM), for the estimation of the mean in dimension one. We then use the latter methodology in a robust iterative clustering algorithm presented in Section 2.2. Our algorithm applies to multi-dimensional data, by estimating centroids according to a bMOM strategy applied to the K-means risk, that is real valued. Moreover, since most clustering approaches crucially depend on the choice of the starting centers, we propose in Section 2.3, a robust initialization procedure based on the bMOM principle.

2.1. Median-of-means and bootstrap median-of-means

Consider a real-valued sample $u_1^n = (u_1, ..., u_n)$ and a partition $I_1^B = \{I_b : b \in \{1, ..., B\}\}$ of the set of indices $\{1, ..., n\}$. The block of index $b \in \{1, ..., B\}$ corresponds to the dataset $(u_i)_{i \in I_b}$. There are thus B disjoint blocks, that form a partition of the sample. By a slight abuse of language, we sometimes refer to the "block b" instead of the "block of index b". The median-of-means (MOM) estimator of the mean in dimension one (Alon et al., 1999; Jerrum et al., 1986; Nemirovsky and Yudin, 1983), subject to the partition of indices I_1^B, consists of taking a median of the arithmetic means computed on the collection of blocks. The lengths of the blocks are generally taken to be equal, possibly up to one data. We thus write the MOM estimator as follows,

$$\mathrm{MOM}(u_1^n, I_1^B) = \mathrm{med}\left\{ \frac{1}{|I_b|} \sum_{i \in I_b} x_i : b \in \{1, ..., B\} \right\},$$

where $|I_b|$ denotes the cardinal of I_b and med is a median, that is $|\left\{b \in \{1, ..., B\} : a_b \leq \mathrm{med}\left\{a_1^B\right\}\right\}| \geq B/2$ and $|\left\{b \in \{1, ..., B\} : a_b \geq \mathrm{med}\left\{a_1^B\right\}\right\}| \geq B/2$ for a median of a collection of real numbers $a_1^B = (a_1, ..., a_B)$. In the following, when the set of possible medians is not a singleton, we always consider its middle point as being our choice of median, that is thus uniquely defined.

We may consider that the blocks are generated according to a random sampling process, that proceeds without replacements (disjoint blocks) and according to the uniform distribution over the remaining data at each step. This formulation naturally leads to consider more general random block generating processes.

For any positive integers n_B and B, denote $q = Bn_B$ and generate a bootstrap sample $v_1^q = (v_1, ..., v_q)$ from the dataset u_1^n. More precisely, each v_l, $l \in \{1, ..., q\}$, is taken uniformly at random from the values $(u_1, ..., u_n)$ and independently from $(v_{l'})_{l' \neq l}$. The boostrap median-of-means (bMOM) of the dataset u_1^n with parameters n_B and B is then the (classical) MOM estimator based on the boostrap sample v_1^q with a partition of indices I_1^B given by $I_b = \{(b-1)n_B + 1, \dots, bn_B\}$ for any $b \in \{1, \dots, B\}$,

$$\mathrm{bMOM}(u_1^n, n_B, B) = \mathrm{MOM}(v_1^q, I_1^B).$$

Note that the bMOM is a randomized estimator, due to the sampling of the bootstrap tuple $(v_1, \dots, v_q)$. Also, for any fixed sample size n, we can choose any block size n_B and number of blocks B to define a bMOM estimator, unlike the classical MOM, where the product of the block size with the number of blocks is equal to the sample size. This will turn out to be extremely useful in the clustering context, where we do not want too small block sizes in order to avoid the disappearance of some clusters in the blocks.

2.2. A robust Lloyd-type algorithm

In this section we propose a hard clustering algorithm based on the bMOM strategy. Unlike section 2.1 above, we consider multi-dimensional data, but the bMOM strategy will still be applied to some one-dimensional statistics.

Let us first introduce some notations. Let $x_1^n = (x_1, \dots, x_n) \in \mathbb{R}^{p \times n}$ denote a dataset of n observations belonging to the Euclidean space $\mathbb{R}^p$, that we want to cluster into K homogeneous groups. We choose two positive integers B and n_B, with $n_B > K$. For $b \in \{1, \dots, B\}$, we denote by $(y_1^{(b)}, \dots, y_{n_B}^{(b)})$ the block of size n_B and of index b generated according to the bootstrap sampling process, that selects at each step, independently from the other steps, an observation according to the uniform distribution over the sample x_1^n. The collection $(y_1^{(1)}, \dots, y_{n_B}^{(1)}, \dots, y_1^{(B)}, \dots, y_{n_B}^{(B)})$, thus forms a bootstrap sample of size $q = Bn_B$ generated from the dataset x_1^n. Again, by a slight abuse of language, we sometimes refer to the "block b" instead of the "block of index b". We define the empirical risk - also called distortion - of the block b as:

$$R^{(b)} = \frac{1}{n_B} \sum_{k=1}^{K} \sum_{l=1}^{n_B} \left\| y_l^{(b)} - c_k^{(b)} \right\|^2 \mathbf{1}\{y_l^{(b)} \in \mathcal{C}_k^{(b)}\},$$

where $y_l^{(b)}$ stands for the l-th datapoint contained in the b-th block, $\mathcal{C}_k^{(b)}$ stands for the set of datapoints belonging to the cluster k in the block b, $\|.\|$ is the Euclidean norm in $\mathbb{R}^p$ and $\mathbf{1}\{E\}$ is the indicator of the event E, that equals 1 when E is true and 0 otherwise. Furthermore, $c_k^{(b)}$ stands for the mean vector of the cluster k in the block b. Finally, we denote by $\mathcal{P}(\mathbf{c}^{(b)})$, the Voronoï partition obtained from the set of centroids $\mathbf{c}^{(b)} = (c_1^{(b)}, \dots, c_K^{(b)})$.

The K-bMOM algorithm

The algorithm that we propose alternates three main steps iteratively. At iteration t, B blocks each containing n_B data are built by sampling uniformly with replacement, and independently from the other iterations, from the original dataset x_1^n. A partition per block is then computed by assigning each data point to its closest centroid given by the previous iteration. The centroids of each block of index b at iteration t, noted as $\mathbf{c}_t^{(b)} = (c_{1,t}^{(b)}, \dots, c_{K,t}^{(b)})$, are then updated according to their block partition and the empirical risk $R_t^{(b)}$ is calculated. The block with the median empirical risk, denoted by *bmed*, is selected and the centers of this median block become the current ones. The bMOM strategy is thus used here since we consider the median of the risks, that are real-valued empirical means of the K-means loss computed from the data in each block. These steps are repeated several times.

In practice, the algorithm is run through a given number of maximum iterations ($t_{max} = 25$ by default). In order to obtain a more precise estimation of the centroids, instead of retrieving the centroids of the median block computed in the last iteration, the centroids of the last 10 iterations are aggregated. The algorithm thus returns $\bar{\mathbf{c}}^{(bmed)} = (\bar{c}_1^{(bmed)}, \dots, \bar{c}_K^{(bmed)})$ such that $\bar{c}_k^{(bmed)} = 1/10 \sum_{t=t_{max}-10}^{t_{max}} \widehat{c}_{k,t}^{(bmed)}$ where $\widehat{c}_{k,t}^{(bmed)}$ stands for the centroid of the cluster k of the median block at iteration t. The final partition over the whole dataset is obtained by assigning each data point to its nearest centroid in $\bar{\mathbf{c}}$. A pseudo algorithm of this procedure is detailed in Algorithm 1.

Our algorithm shares some similarity with the techniques of so-called consensus/ensemble clustering (Nguyen and Caruana, 2007), since it amounts at each step, to producing a robust clustering, given by a codebook, from a collection of candidates computed on bootstrap sub-samples. However, there are also essential differences, since we select one of the candidates by a simple median criterion for dimension one statistics, whereas consensus clusterings aggregate the candidates in a more complicated fashion, using some similarity measures between clusterings. Interestingly, so-called bagged clustering (Leisch, 1999; Dolnicar and Leisch, 2003; D'Urso et al., 2013) proposes performing clusterings on bootstrap samples and aggregating them using a hierarchical clustering on the collection of obtained centroids. Moreover, the size of the bootstrap samples is equal to the original sample size, whereas in our approach the sub-sampling is crucial and directly related to the allowed proportion of outliers (see Section 3.2).

A robust trimmed clustering approach for probabilities in Wasserstein space is developed in del Barrio et al. (2019) and used to advantage to robustly aggregate model-based clusterings on multivariate data - each clustering being seen as a probability - that are previously learned on sub-samples of the original data. This approach, however, concerns the robust aggregation of model-based clusterings, whereas our focus is on robust hard clustering in the context of the K-means problem.

Algorithm 1 Iteration phase structure.

```
procedure K-bMOM($x_1^n, K, B, n_B$)                                   ▷ ($n_B > K$)
    Let $(c_{1,0}, \dots, c_{K,0})$ be the $K$ initial centroids and called reference centroids.
    Set $t = 1$.
    while $t \leq t_{max}$ do
        Create $B$ blocks $(y_{1,t}^{(b)}, ..., y_{n_B,t}^{(b)})$ for $b \in \{1, ..., B\}$, according to a random sampling process that at each step selects an observation uniformly over the data $x_1^n$ and independently from the other steps.
        for all $b \in \{1, \dots, B\}$ do
            Assign each data point in the block of index $b$ to its closest reference centroid.
            Set $n_{k,t}^{(b)}$ the number of data points in the block $b$ belonging to the cluster $k$.
            if $n_{k,t}^{(b)} > 1, \forall k \in \{1, \dots, K\}$ then
                for all $k \in \{1, \dots, K\}$ do
                    $c_{k,t}^{(b)} \leftarrow 1/n_{k,t}^{(b)} \sum_{l=1}^{n_B} y_{l,t}^{(b)} \mathbf{1}\{y_{l,t}^{(b)} \in \mathcal{C}_{k,t}^{(b)}\}$.
                    $R_t^{(b)} \leftarrow \frac{1}{n_B} \sum_{k=1}^{K} \sum_{l=1}^{n_B} \left\| y_{l,t}^{(b)} - c_{k,t}^{(b)} \right\|^2 \mathbf{1}\{y_{l,t}^{(b)} \in \mathcal{C}_{k,t}^{(b)}\}$.
                end for
            else
                Skip the block.
            end if
        end for
        Get the median block bmed such that $R_t^{(bmed)} = \text{med}\left\{R_t^{(b)} : b \in \{1, ..., B\}\right\}$ and $\left(\widehat{c}_{1,t}^{(bmed)}, \dots, \widehat{c}_{K,t}^{(bmed)}\right)$ the centroids assigned to the median block bmed at iteration $t$ becoming the reference centroids.
        $t \leftarrow t + 1$.
    end while
    return $\bar{\mathbf{c}}^{(bmed)} = \left(\bar{c}_1^{(bmed)}, \dots, \bar{c}_K^{(bmed)}\right)$ such that $\bar{c}_k^{(bmed)} = \frac{1}{10} \sum_{t=t_{max}-10}^{t_{max}} \widehat{c}_{k,t}^{(bmed)}$ for all $k \in \{1, ..., K\}$ and $\mathcal{P}(\bar{\mathbf{c}}^{(bmed)})$.
end procedure
```

Model selection

In model-based clustering, it is frequent to consider several models in order to find the most appropriate one for the considered data. In particular, for most clustering algorithms, the model is specified by its number of clusters K. There are lots of ad-hoc approaches in the literature to select the number of components K and we can therefore think of the Gap statistics from Tibshirani et al. (2001), the Silhouette criterion and so one.

Furthermore, since the K-means algorithm can be seen as a hard version of an EM-like algorithm which tries to estimate a mixture of K Gaussians with isotropic covariance matrices, we can therefore try to adapt classical tools for model selection including BIC, ICL criteria and the slope heuristics (Baudry et al., 2012) for example. All these model selection criteria are based on a so-called penalty that is added to the empirical risk. However, it is not reasonable to use the empirical risk in our context, due to the presence of outliers. Instead, for each of K, we could think of using the centroids that are returned by the K-bMOM algorithm, generate blocks according to the bootstrap sampling process and use the median empirical risk over the blocks as a robust estimate of the empirical risk. The idea would then be to penalize these median empirical risks for various values of K by a classical penalty, such as the one used in BIC for instance. Such robust model selection procedure would require investigations that represent an interesting direction of research for future work.

2.3. A robust initialization

It is well-known that since the clustering problem is non-convex, the initialization step is a keystone for the resulting partition. We propose therefore, a robust variant of a classical and efficient initialization procedure by applying the bMOM strategy.

More precisely, the idea is to build B blocks of n_B data points, with $n_B > K$, by sampling uniformly and with replacement over the dataset x_1^n. The K-means++ initialization (Arthur and Vassilvitskii, 2007) is then operated in each block. Recall that the latter approach proceeds in an iterative way: it starts with a centroid picked at random among the data points. Iteratively and until the number of groups K is reached, a new centroid is then chosen from the data points with a probability which increases exponentially with the squared Euclidean distance to the already chosen closest centers.

In each block, the empirical risk $R_{++}^{(b)}$ is therefore computed and the centers linked to the median empirical risk, called the median block, are selected as the initial centers. This algorithm is summarized in Algorithm 2. The robustness of this initialization scheme is evaluated in Section 5.2.

3. A breakdown point analysis

We prove in Section 3.1 below that the bMOM estimator of the mean enables us to perform a more robust mean estimation than MOM, if enough blocks are generated, in the sense that the breakdown point of bMOM is higher. We believe that this result is of independent interest, as it shows the advantage of taking a large bootstrap sample before applying the MOM principle. This result still has some limitations however in the perspective of clustering, since K-bMOM does not correspond to the bMOM estimator for $K = 1$. In Section 3.2 therefore, we study a notion of breakdown point for the K-bMOM algorithm itself.

Algorithm 2 Initialization strategy.

```
procedure K-bMOM-KM++($x_1^n, K, B, n_B$)
    Create $B$ blocks $(y_1^{(b)}, ..., y_{n_B}^{(b)})$ for $b \in \{1, ..., B\}$, according to a random sampling process that at each step selects an observation uniformly over the data $x_1^n$ and independently from the other steps.
    for all $b \in \{1, \dots, B\}$ do
        Proceed to a $K$-means++ initialization based on the sample $(y_1^{(b)}, ..., y_{n_B}^{(b)})$. This gives the centroids $(c_{1,++}^{(b)}, ..., c_{K,++}^{(b)})$.
        Compute the empirical risk $R_{++}^{(b)}$ of the block $b$:
        $R_{++}^{(b)} \leftarrow \frac{1}{n_B} \sum_{k=1}^{K} \sum_{l=1}^{n_B} \left\| y_l^{(b)} - c_{k,++}^{(b)} \right\|^2 \mathbf{1}\{y_l^{(b)} \in C_k^{(b)}\}$.
    end for
    Select the block $bmed$ having the median empirical risk.
    return $\left(\widehat{c}_{1,++}^{(bmed)}, \dots, \widehat{c}_{K,++}^{(bmed)}\right)$ the centroids of the median block $bmed$.
end procedure
```

3.1. Breakdown points for mean estimation

The breakdown point is a classical concept of robust statistics (Hampel et al., 1986; Huber and Ronchetti, 2009; Maronna et al., 2019), that gives the maximal proportion of outliers that is allowed so that the deviations of the estimator stay bounded compared to the no-corruption setting.

Assume that we are given a sample $u_1^n = (u_1, ..., u_n)$ of real valued random variables.

Definition 1 *(Deterministic breakdown point).* The (deterministic) breakdown point $\delta_n(\widehat{T}, u_1^n)$ of a real-valued estimator $\widehat{T}$ given the sample u_1^n, is the maximal proportion of outliers that leaves the value of the estimator bounded.

$$\delta_n\left(\widehat{T}, u_1^n\right) = \frac{1}{n} \max \left\{ m : \max_{i_1, ..., i_m} \sup_{e_1, ..., e_m} \left|\widehat{T}\left(s_1, ..., s_n\right)\right| < \infty \right\},$$

where the sample $(s_1, ..., s_n)$ is obtained by replacing the m data points $u_{i_1}, ..., u_{i_m}$ of the sample u_1^n by arbitrary values $e_1, ..., e_m$.

Definition 1 corresponds to a worst case analysis, the outliers potentially appearing at the worst places for the estimator $\widehat{T}$. If the estimator $\widehat{T}$ is randomized - rather denoted $\widehat{T}^{\omega}$ in this case -, then its breakdown point is a random variable.

For a median $\text{med}\left\{u_1^n\right\}$, it holds that $\delta_n\left(\text{med}\left\{u_1^n\right\}, u_1^n\right)$=$\lfloor (n-1)/2 \rfloor /n$ and for the empirical mean, $\bar{u}_n = 1/n \sum_{i=1}^n u_i$, $\delta_n\left(\bar{u}_n, u_1^n\right) = 0$.

Proposition 1. *The breakdown point of the median-of-means estimator of the mean in dimension one is*

$$\delta_n\left(\text{MOM}(u_1^n, I_1^B), u_1^n\right) = \frac{\left\lfloor \frac{B-1}{2} \right\rfloor}{n}.$$

The proof of Proposition 1 is direct since MOM diverges if and only if there is at least one outlier in a majority of blocks. Note that the same breakdown point is achieved for a more general estimator of a multi-dimensional mean called the median-of-means tournament (Rodriguez and Valdora, 2019).

Let us now consider the use of replacements while constructing the blocks and study the breakdown point of bMOM.

Proposition 2. *Assume first that the sample size satisfies $n = Bn_B$, for positive integers B and n_B. We then have*

$$\delta_n\left(\text{bMOM}(u_1^n, n_B, B), u_1^n\right) \leq \delta_n\left(\text{MOM}(u_1^n, I_1^B), u_1^n\right) \ a.s.$$

Secondly, fix the block size n_B and the sample size n and let the number of blocks taken in the bMOM, tend to infinity. It holds that

$$\lim_{B \to \infty} \delta_n\left(\text{bMOM}(u_1^n, n_B, B), u_1^n\right) = \frac{\left\lfloor n\left(1 - \frac{1}{2^{1/n_B}}\right)\right\rfloor}{n} \ a.s.$$

Note that $1 - \frac{1}{2^{1/n_B}} \sim_{n_B \to \infty} \frac{\log 2}{n_B} \simeq \frac{0.69}{n_B}$.

The proof of Proposition 2 can be found in the Supplementary Material (Brunet-Saumard et al., 2022).

On the one hand, the first display in Proposition 2 states that when the number of blocks in bMOM is equal to the number of blocks in MOM, bMOM has a breakdown point that is less than or equal to the breakdown point of MOM. On the other hand, the second display in Proposition 2 states that for a fixed block size, when the number of blocks in bMOM tends to infinity, its breakdown point tends to a value that is strictly greater than the breakdown point of MOM

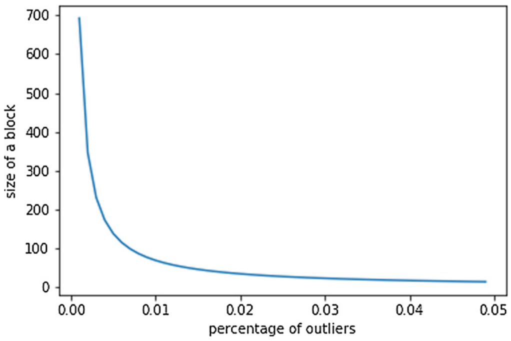

Fig. 1. Maximum admissible block size n_B for bMOM as a function of the proportion of outliers $p = m/n$.

taken with the same block size, at least for n sufficiently large. Indeed, by assuming $n = Bn_B$, Proposition 1 gives that the breakdown point of MOM is smaller than $1/(2n_B)$, while for n sufficiently large, the second display of Proposition 2 ensures that the breakdown point of bMOM is strictly greater than $1/(2n_B)$. This is of importance for practice, since it implies that one should consider if possible, building a bootstrap sample that is larger than the original one, and then take the MOM statistics on this bootstrap sample rather than on the original dataset.

Considering that the contaminated sample is given (fixed), it is interesting to evaluate the probability that a randomized estimator does not diverge when the outliers go to infinity. It can indeed happen that the indices of the outliers are not the worst with respect to the block sampling process. This leads to the following definition.

Definition 2 *(Probabilistic breakdown point).* The probabilistic breakdown point of a randomized estimator $\widehat{T}^{\omega}$ given the sample u_1^n is

$$p_n\left(\widehat{T}^{\omega}, u_1^n, (i_1, ..., i_m)\right) = \mathbb{P}\left(\left\{\omega : \sup_{e_1,...,e_m} \left|\widehat{T}^{\omega}(s_1, ..., s_n)\right| < \infty\right\}\right), \tag{1}$$

where the sample $(s_1, ..., s_n)$ is obtained by replacing the m data points $u_{i_1}, ..., u_{i_m}$, for some fixed indices $(i_1, ..., i_m)$, by the arbitrary values $e_1, ..., e_m$.

Note that in the probability at the right-hand side of Identity (1), the non-corrupted dataset u_1^n is fixed, the indices $(i_1, ..., i_m)$ where the outliers replace the non-corrupted data $(x_{i_1}, ..., x_{i_m})$ are fixed and the outliers $(e_1, ..., e_m)$ are deterministic. The only randomness that is taken into account is the randomness induced by the randomized estimator. This may be indeed relevant in practice, where the corrupted dataset is fixed, as it allows us to discuss if a randomized estimator has a high probability, depending on the randomness of its generating process, of being robust to the presence of outliers.

As $p_n\left(\text{bMOM}(u_1^n, n_B, B), u_1^n, (i_1, ..., i_m)\right)$ only depends on m, but not on the values of $(i_1, ..., i_m)$, we will rather denote it $p_n\left(\text{bMOM}(u_1^n, n_B, B), m\right)$. We have the following bound.

Proposition 3. *Assume that the block length n_B in bMOM and the proportion of outliers m/n are such that $(1 - m/n)^{n_B} > 1/2$. Then it holds that*

$$p_n\left(\text{bMOM}(u_1^n, n_B, B), m\right) \geq 1 - \exp\left(-2B\left((1 - m/n)^{n_B} - 1/2\right)^2\right).$$

The proof of Proposition 3 is based on an application of Hoeffding's concentration inequality and can be found in the Supplementary Material (Brunet-Saumard et al., 2022).

According to Proposition 3, if the number of outliers m and the sample size n are fixed, then the block length n_B should be chosen such that $(1 - m/n)^{n_B} > 1/2$, i.e. $n_B < \log(2)/\log(1/(1 - m/n))$ (Fig. 1). One can indeed notice that the quantity $(1 - m/n)^{n_B}$ corresponds to the probability, according to the bootstrap sampling, that a block is not corrupted. Hence, in case of a large proportion of outliers m/n, the block length should not be taken too large. Furthermore, by denoting $D = (1 - m/n)^{n_B} - 1/2 > 0$, we have that $p_n\left(\text{bMOM}(x_1^n, n_B, B), m\right) \geq 1 - R$ if $B > \log(1/R)/\left(2D^2\right)$. Consequently, if the block size n_B is chosen correctly (not too large according to the proportion of outliers, so that $D > 0$), then the probability that the bMOM remains stable under contamination, tends to one when the number of blocks B tends to infinity.

3.2. Probabilistic breakdown point for the K-bMOM algorithm

Let us turn now to the study of the breakdown point of the K-bMOM algorithm presented in Section 2.2. It produces a randomized estimator, consisting of a set of K centroids, due to the sampling of the blocks at each step of the algorithm.

We thus denote it $\bar{\mathbf{c}}^\omega$ rather than $\bar{\mathbf{c}}$ as in Section 2.2, the notation ω accounting for randomization induced by the sampling of the blocks. We discuss the following notion of probabilistic breakdown point.

Definition 3 *(Probabilistic breakdown point for K-bMOM).* The probabilistic breakdown point $p_n\left(\bar{\mathbf{c}}^\omega, x_1^n, (i_1, ..., i_m)\right)$ of the randomized estimator $\bar{\mathbf{c}}^\omega = \bar{\mathbf{c}}$, defined in Section 2.2 as the output of the algorithm K-bMOM, is given by

$$p_n\left(\bar{\mathbf{c}}^\omega, x_1^n, (i_1, ..., i_m)\right) = \mathbb{P}\left(\left\{\omega : \sup_{y_1,...,y_m} \max_{c \in \bar{\mathbf{c}}^\omega(z_1,...,z_n)} \|c\| < \infty\right\}\right) , \tag{2}$$

where the sample $(z_1, ..., z_n)$ is obtained by replacing the m data points $x_{i_1}, ..., x_{i_m}$, for some fixed indices $(i_1, ..., i_m)$, by the arbitrary values $y_1, ..., y_m$ and $\bar{\mathbf{c}}^\omega(z_1, ..., z_n)$, consisting in a set of K centroids, is the output of the K-bMOM algorithm when taking as input the dataset $(z_1, ..., z_n)$.

The probabilistic breakdown point defined in Identity (2) of Definition 3 corresponds to the probability that the K centroids output by the K-bMOM algorithm, stay bounded when the input dataset is corrupted by outliers that can take any possible values. In this probability, the non-corrupted dataset x_1^n is fixed, the indices $(i_1, ..., i_m)$ where the outliers replace the non-corrupted data $(x_{i_1}, ..., x_{i_m})$ are fixed and the outliers $(y_1, ..., y_m)$ are deterministic. The only randomness that is taken into account is the randomness induced by the sampling of the blocks at each step of the algorithm. In practice indeed, the corrupted dataset is given to the statistician and it is important to know if the randomized estimator produced by the K-bMOM algorithm has a high probability, through the sampling process, of being robust to the presence of outliers.

The K-bMOM algorithm will be proven to be robust in terms of its probabilistic breakdown point in the case of a "well-clusterizable" clustering configuration, that is a classical assumption for obtaining robustness in clustering (García-Escudero et al., 2010; Ritter, 2015). Roughly speaking, a well-clusterizable configuration is made of "compact" clusters that are well "separated". We give the following formal definition, suitable for our needs.

Definition 4. A dataset x_1^n is said to be in a well-clusterizable configuration, with compactness parameter r and separation parameter R satisfying $R > 2r > 0$, if the points x_1^n lie in a union of K disjoint balls $B(a_k, r)$, $k = 1, ..., K$, of radius r with centers a_k separated from each other by at least a distance R: $\min_{k \neq k'} ||a_k - a_{k'}|| \geq R$. Moreover, each ball $B(a_k, r)$ is assumed to contain exactly one cluster.

Theorem 1. *Let $\bar{\mathbf{c}}^\omega$ be the K-bMOM output, computed iteratively by using at each step B blocks of size n_B. Assume that the block length n_B and the proportion of outliers m/n are such that $(1 - m/n)^{n_B} > 1/2$. Assume furthermore that the regular data points x_1^n are in a well-clusterizable situation, with compactness and separation parameters denoted respectively r and R, satisfying $R^2 > 16 n_B r^2$. Finally, assume that at the beginning of the last 10 iterations, the algorithm has identified the correct partition of the regular data, meaning that one cluster is associated with one centroid. It holds then that $p_n(\bar{\mathbf{c}}^\omega, x_1^n, (i_1, ..., i_m)) \geq \max\{p_1 - p_2, 0\}$ with*

$$p_1 = \left(1 - \sum_{k=1}^{K} \left(1 - \frac{n_k^r}{n}\right)^{n_B}\right)^{10B} \tag{3}$$

and

$$p_2 = 10 \exp\left(-2B\left(\left(1 - \frac{m}{n}\right)^{n_B} - \frac{1}{2}\right)^2\right) , \tag{4}$$

where the quantity n_k^r in display (3) stands for the number of regular data belonging to cluster k in the sample $(z_1, ..., z_n)$ defined in (2).

The proof of Theorem 1 can be found in the Supplementary Material (Brunet-Saumard et al., 2022).

We assume in Theorem 1 that the K-bMOM algorithm is not too far from the solution, by postulating that it has found the right partition at the beginning of the last 10 iterations. The output of the algorithm will indeed be built from the centroids obtained during these last 10 steps. This assumption seems legitimate, since analyzing the behavior of clustering algorithms in a neighborhood of the optimal solutions, by assuming a "warm start" for instance, is very classical. We could also assume a warm start by requiring that the initialization procedure has found the right partition, at the price of considering the total number of iterations of K-bMOM instead of the last 10 iterations.

A natural and important question is to ask whether this warm start assumption is realistic in a context where outliers are present in the dataset. Even if we have no theoretical evidence of this fact, we show in our experiments that it is indeed possible to propose an initialization that is robust to the presence of some outliers, in the sense that it produces a high classification accuracy in our different experiment settings. See Section 5.2 for details.

Furthermore, Theorem 1 provides some guarantee for the use of the K-bMOM algorithm, when the latter is performed using an accurate, robust initialization. This rationale is confirmed by our experiments, that show the good behavior of K-bMOM when initialized by the bMOM strategy coupled with the algorithm K-means++ (see Section 5.3 below).

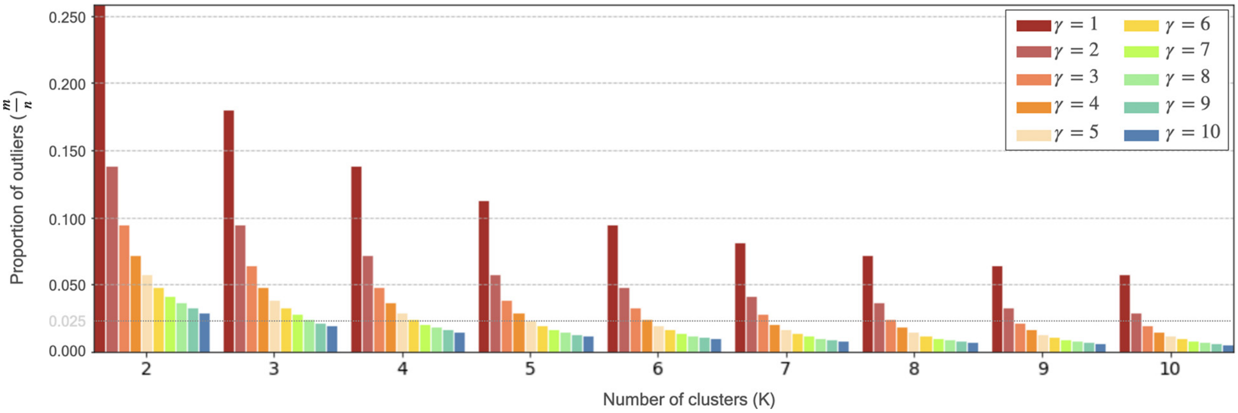

Fig. 2. Evolution of the maximum proportion of outliers allowed for the probability that a block is not contaminated by outliers to be greater than 0.5, according to the number of clusters and the coefficient γ. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

We also assume that the data is in a well-clusterizable configuration with compactness and separation parameters r and R that satisfy $R^2 > 16 n_B r^2$. This condition is surely pessimistic, but it allows us to deduce a non-trivial lower bound for the probabilistic breakdown point with a reasoning that is kept rather simple. The rationale to keep in mind for practice is that if r and R can be well defined, then when the ratio R/r increases, the algorithm is more likely to be robust with respect to the presence of outliers.

We see from Theorem 1 that the K-bMOM algorithm has a high probability of being robust if the quantity p_1 defined in (3) is close to 1 and the quantity p_2 given by (4) is close to 0. To analyze p_1, note that if the clusters are well-balanced and if the proportion of outliers is not too large, then the quantities n_k^r/n can be approximated by $1/K$. In this case, p_1 simplifies to $p_1 \simeq (1 - K(1 - 1/K)^{n_B})^{10B}$. We see from this approximation that p_1 can indeed be close to 1, even if our theoretical computations may be too pessimistic for being accurate in practice. For instance, if $K = 3$, $n_B = 10K = 30$ and $B = 100$, the value of p_1 is greater than 0.98.

Finally, for the quantity p_2 to be close to 0, we need the number of blocks B generated in each step to be sufficiently large. This seems to corroborate our empirical conclusions related to the choice of the number of blocks, see Section 4.2 below.

4. Scope of K-bMOM and practical considerations

4.1. Block length, number of clusters and proportion of outliers

The purpose of this section is to clarify the scope of the K-bMOM algorithm according to the proportion of outliers m/n, the number of clusters K and the block size n_B.

The reader is reminded that Theorem 1 on the probabilistic breakdown point of K-bMOM holds if the probability that a block is not contaminated by outliers, is strictly greater than 0.5. This probability takes the value $(1 - m/n)^{n_B}$. The block length n_B is therefore dependent of the proportion of outliers m/n. Taking a block size $n_B = \gamma K$ with $\gamma \in \mathbb{N}^*$ (it can be seen roughly as the number of data points per cluster), the maximum proportion of outliers for which our proposed approach is robust can be evaluated for a given probability that a block is not contaminated by outliers.

To do so, let us take this probability equal to 0.55, such that $(1 - m/n)^{n_B} = 0.55 > 1/2$. According to the previous condition, we get: $m/n = 1 - (0.55)^{1/(\gamma K)}$. Fig. 2 stands for the maximum level of contamination according to the number of clusters for different values of $\gamma \in [1, 10]$. As can be observed, the case $\gamma = 1$ enables us to deal with a contamination level up to 10% for a number of clusters $K < 7$. However, if by chance each cluster is represented in a block, the estimation of centroids is roughly based on one data point only which may lead to inaccurate estimations. On the other hand, by taking $\gamma = 5$ (around 5 data points per cluster), the level of outliers for which the K-bMOM algorithm is robust, has to be low (under 5% if $K > 2$) but we can expect having an accurate estimation of centroids. Therefore, there is a trade-off in practice between the number of clusters, the proportion of outliers and the accuracy of the centroid estimation.

The second illustration, proposed in Table 1, evaluates the maximum block size n_B according to a range of probabilities that a block is healthy for different proportions of outliers. It can be noted that the block size n_B dramatically decreases when the percentage of outliers increases. A proportion of outliers up to 0.04 leads to a majority of block sizes containing less than 10 data points. When the number of clusters remains small ($K \in \{2, 3\}$) with a uniform presence in the block ($\gamma \in \{3, 4, 5\}$), the K-bMOM algorithm should behave correctly insuring that the probability that a block is healthy is greater than 1/2 for a proportion of outliers up to 0.4 (see bold values located in the upper left diagonal in Table 1). However, if the number of groups becomes quite high, e.g. $K = 10$ with at least 5 datapoints per cluster ($\gamma = 5$), the scope of K-bMOM narrows to a proportion of outliers of 0.01 and below, as illustrated by the blue bold values in Table 1.

Table 1
Lookup table of the maximum block size n_B evaluated according to a range of proportion of outliers m/n and a range of probabilities that a block is healthy. In bold, the ranges of block sizes for $K = 3$ with $\gamma = 5$. In bold blue, the possible ranges of n_B for $K = 10$ with $\gamma = 5$.

		Probability that a block is healthy									
		0.51	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
Proportion of outliers m/n	0.001	**673**	**597**	**510**	**430**	**356**	**287**	**223**	**162**	**105**	**51**
	0.005	**134**	**119**	**101**	**85**	**71**	**57**	**44**	**32**	**21**	10
	0.01	**66**	**59**	**50**	**42**	**35**	**28**	**22**	**16**	10	5
	0.02	**33**	**29**	**25**	**21**	**17**	14	11	8	5	2
	0.03	**22**	**19**	**16**	14	11	9	7	5	3	1
	0.04	**16**	14	12	10	8	7	5	3	2	1
	0.05	13	11	9	8	6	5	4	3	2	1
	0.1	6	5	4	4	3	2	2	1	1	0

In conclusion, when the number of clusters is small ($K \leq 5$) the K-bMOM algorithm should be robust with respect to a proportion of outliers up to $m/n = 0.03$ with a limited block size ($n_B \simeq 25$ and $\gamma \geq 5$). For a higher number of groups, the K-bMOM algorithm should remain accurate but for a smaller percentage of outliers (below 1%). In practice, this situation should not be too restrictive. Indeed, since an outlier is a data point that differs considerably from all or most other data in a dataset, we do not expect having a large proportion of them in a dataset (in contrast to noisy data). Section 5.3 evaluates the performance of the K-bMOM algorithm in different simulation contexts.

4.2. Influence of the number of blocks

In the previous section, we showed the strong influence of the block size on the maximum proportion of outliers allowed to guarantee a percentage of healthy blocks. In particular, the smaller the size, the more robust the algorithm is to a large proportion of outliers. In this section, we focus on the influence of the second hyper-parameter of the K-bMOM algorithm which is the number of blocks. To do so, we consider a 2-dimensional isotropic Gaussian mixture model of $K = 3$ components with equal size $n_1 = n_2 = n_3 = 300$. The mean vectors are set to $\mu_1 = [3, 12], \mu_2 = [6, 3]$ and $\mu_3 = [-6, 9]$ and the scaling parameter is set to $\sigma^2 = 0.6$. Twenty outliers are randomly selected from the data and their coordinates are multiplied by 10. The block size is set to $5K = 15$ to be robust to any level of outliers (see Table 1) with a sufficient number of elements per group. The number of blocks varies between 1 block up to 1000 blocks and the process is iterated 100 times. In order to evaluate the influence of the number of blocks on the robustness of the algorithm, three criteria are computed in each context:

- the accuracy computed between the partition obtained by the nearest fitted centers and the labels of regular data.
- the empirical distortion $\hat{R}$ obtained at the end of the studied process and computed over the $(1 - m/n)n$ regular data, such as:

$$\hat{R} = \frac{1}{(1 - m/n)n} \sum_{k=1}^{K} \sum_{i \in \mathcal{I}} \left\| x_i - \bar{c}_k^{(bmed)} \right\|^2 \mathbf{1}\{x_i \in \mathcal{C}_k\}. \tag{5}$$

We recall that $\bar{\mathbf{c}}^{(bmed)} = (\bar{c}_1^{(bmed)}, \ldots, \bar{c}_K^{(bmed)})$ is the output of the K-bMOM algorithm (see Algorithm 1) and $\mathcal{C}_k$ is the corresponding cluster k.

The violin plots of these three criteria are illustrated in Fig. 3. The median is shown by a bold black dash. As it can be observed, as of a number of blocks $B \geq 50$, the performance of the algorithm is ensured and remains stable.

These results confirm the choice of a small size of a block (see Section 4.1 and Section 3.1) and a high number of blocks ($B > 50$) to have a procedure that is likely to be robust. In practice, the values $n_B = 5K$ and $B = 500$ are the default parameters applied in the K-bMOM algorithm when the proportion of outliers is unknown.

5. Experimental simulations

This section aims at evaluating the scope of performances of the K-bMOM as an initialization strategy and as a clustering algorithm according to a taxonomy of different types of outliers on one hand and according to several specifications such as sample size, dimension and number of groups on the other hand.

5.1. Experimental contexts and practical considerations

The same experimental context will be addressed to evaluate the proposed robust initialization and the K-bMOM algorithm. In particular, the different situations considered will depend on the 6 different aspects listed below:

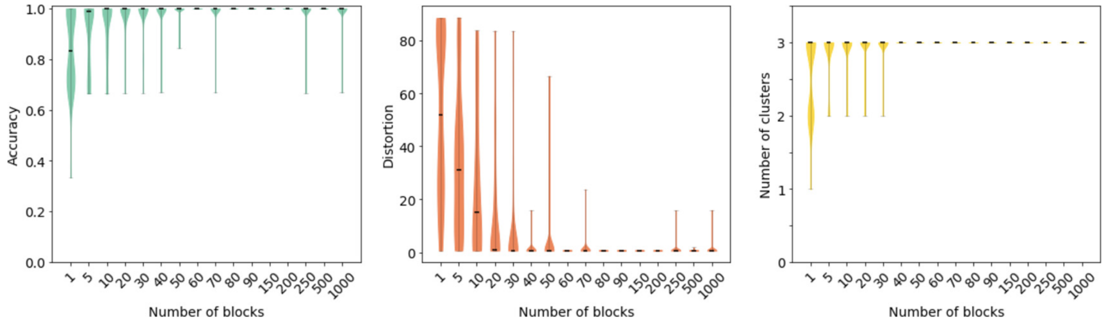

Fig. 3. Violin plots of accuracy (left), distortion (middle), number of clusters (right) computed on the partition of regular data obtained by the K-bMOM algorithm as a function of the number of blocks.

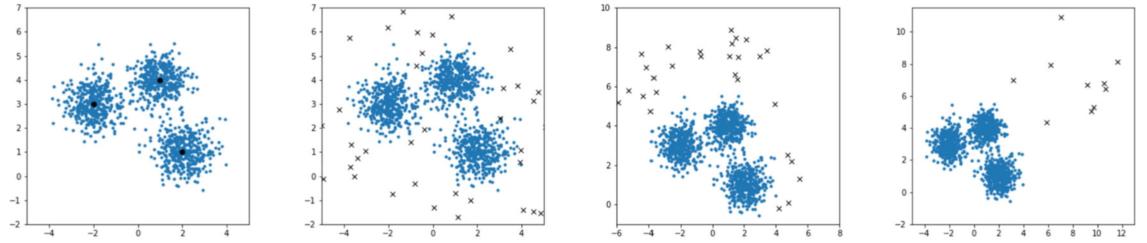
(a) Regular data (b) Punctual outliers (T1) (c) Punctual oriented outliers (T2) (d) Cluster of outliers (T3)

Fig. 4. Illustrations of simulated regular data (blue points) generated according to a Gaussian Mixture Model with isotropic variance and different types of outliers (black crosses).

1. the outliers typology. Three types of outliers are considered: isolated multi-directional outliers, isolated oriented outliers and a cluster of outliers.
2. the proportion m/n of outliers contamination.
3. the sample size n of data.
4. the dimension p of data. Let us note that our strategy is not designed to deal with high-dimensional data. Therefore, the number of dimensions tested will remain small and discriminant since we do not deal with the problem of noisy dimensions.
5. the separability of clusters, by varying the level of the scaling parameter σ^2 of the identity covariance matrix.
6. the number K of clusters.

Regular data and outliers generation procedures

We generate n data from K multivariate Gaussian distributions of dimension p with equal size n/K, isotropic variance $\Sigma = \sigma^2 \mathbf{I}_p$ (with $\sigma^2 > 0$ and $\mathbf{I}_p$ the p-dimensional identity matrix) and average vectors μ_k with $k \in \{1, \dots, K\}$. Fig. 4.a illustrates one realization of the simulated context in the case of $K = 3$ groups. Three typologies of outliers are considered and are generated as expressed below:

1. isolated outliers: they are generated uniformly in a parallelogram defined by the coordinatewise ranges of the regular data points. Using an acceptance–rejection algorithm as in García-Escudero et al. (2008), only data points having squared Mahalanobis distances from the centers greater than the quantile $\chi^2_{p,0.975}$ are retained and only m of them are going to replace the same number of randomly selected regular data. This case is illustrated in Fig. 4.b.
2. isolated oriented outliers: m regular data points are randomly selected as potential outliers and their coordinates are multiplied by a constant term β which quantifies how far these outliers are from their own distribution. Fig. 4.c illustrates such a type of outliers.
3. Cluster of outliers: m regular data points are randomly replaced by a cluster of outliers of the same size generated according to a 2-dimensional Gaussian distribution with average $\mu_{outlier} = \beta[1, 1]$ and $\sigma^2_{outlier}$ as a scaling parameter of the covariance matrix. This situation is depicted in Fig. 4.d.

Experimental values

The experimental values taken for each of these scenarios are detailed below:

Aspects	Detailed case	Values
proportion of contamination		$m/n \in \{0, 0.001, 0.005, 0.01, ..., 0.04\}$
outliers parameters		$\beta \in \{9, 27\}$, $\sigma^2_{outlier} = 2$
dimension		$p \in \{2, 5\}$
sample size		$n \in \{120, 1200, 12000\}$
separability of clusters	(high, medium, low)	$\sigma^2 \in \{0.4, 0.6, 0.8\}$
number of clusters		$K \in \{3, 5, 10\}$

The average vectors according to each configuration of the considered Gaussian mixture model, are defined as it follows:

- $(K = 3, p = 2)$: $\mu_1 = [1, 4], \mu_2 = [2, 1], \mu_3 = [-2, 3]$,
- $(K = 5, p = 2)$: μ_1, μ_2, μ_3 and $\mu_4 = [0, -1], \mu_5 = [1, -3]$,
- $(K = 10, p = 2)$: $\{\mu_k\}_{k\in[1,5]}$ and $\mu_6 = [0, 7], \mu_7 = [3, 6], \mu_8 = [5, 1], \mu_9 = [7, 0], \mu_{10} = [8, 4]$,
- $(K = 3, p = 5)$: $\mu_1 = [1, 4, b, a, a], \mu_2 = [2, 1, a, b, a], \mu_3 = [-2, 3, a, a, b]$,
- $(K = 5, p = 5)$: $\{\mu_k\}_{k\in[1,3]}$ and $\mu_4 = [0, -1, b, b, b], \mu_5 = [1, -3, a, a, a]$,
- $(K = 10, p = 5)$: $\{\mu_k\}_{k\in[1,5]}$ and $\mu_6 = [0, 7, a, b, b], \mu_7 = [3, 6, b, a, a]$, $\mu_8 = [5, 1, a, b, a], \mu_9 = [7, 0, b, a, a]$, and $\mu_{10} = [8, 4, a, b, b]$,

with $a = 0$ and $b = -1$. Let us note that for all the experiments, the number of clusters (and the proportion of outliers) is supposed to be known and fixed to its true value K (respectively m/n).

Performance criteria

In order to compare the different starting strategies in terms of performance, we compute three criteria: (1) the accuracy computed between the partition obtained by the nearest fitted centers and computed on the regular data, (2) the empirical distortion computed on the regular data and defined in Equation (5) and (3) the Root Mean Square Error (RMSE) in order to evaluate the robustness of fitted centers. This criterion is calculated between the centers fitted by the studied process and those used to simulate the data:

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{K} \left\| \bar{c}_k^{(bmed)} - \mu_k \right\|^2}{K}}, \tag{6}$$

where $\bar{c}_k^{(bmed)}$ stands for the fitted center the most probable for the class k averaged on the last 10 iterations and μ_k the average parameter of the k-th component.

For each simulation context, the experiment is repeated 1000 times. These criteria are averaged and standard deviations have been computed for each initialization or clustering approach.

5.2. Comparing initialization strategies for the clustering task

The experimental context of this section aims at evaluating the scope of performances of the robust initialization process that we propose. In particular, we apply the MOM principle to the most widely used initialization methods amongst which K-means++ and K-medians++ as expressed in Algorithm 2 in Section 2.2. We consider the following three traditional initialization strategies: Random initialization, K-means++ (Arthur and Vassilvitskii, 2007), K-medians++ and also a robust initialization strategy developed by Al Hasan et al. (2009) named ROBIN. The implementations that we used in this study for the above approaches come from SCIKIT-LEARN library which is a free software machine learning library for the Python programming language and is publicly available in Pedregosa et al. (2011).

Global comments and results

First of all, it is important to notice that the behavior of the six initialization procedures is really stable and comparable in terms of accuracy when the data are not polluted by any kind of outliers as illustrated in Table 2 (top). As expected, the accuracy decreases when the cluster separability becomes weaker (i.e. when the scaling parameter σ^2 increases). Moreover, an interesting point is the level of accuracy obtained by the different approaches to estimate the centers of each cluster. It appears that, on regular data, this is the K-bMOM-km++ and K-bMOM-kmed which on average, fits the centers of clusters better than the other methods. See Table 2 (bottom).

Table 3 highlights aggregated performances on the different situations (dimension, separability, number of clusters, etc) of six initialization strategies according to the typology of outliers considered. Median and standard deviations (in brackets) of three metrics are represented. They are computed on all simulated contexts with outliers. First of all, it can be noted that in the case of isolated outliers (T1), all methods (except for the random case) perform quite well on average: their average accuracy remains above 0.83. Secondly, it is worth noting that the traditional approaches are being impacted by oriented isolated outliers (T2) and clustered outliers (T3). In particular, in the case T2, the average accuracy of these approaches are 10% (as K-medians++) to 40% (K-means++) lower than the MOM-based approaches. Moreover, the K-bMOM-km++ RMSE remains smaller compared to the rest of the approaches. Similar results are observed in the case T3.

Table 2
Averages of accuracy (top) and RMSE (bottom) and their standard deviations (in brackets) of initialization procedures on regular data as a function of the separability of clusters.

Accuracies:						
σ^2	Random	K-medians++	K-means++	ROBIN	K-bMOM-km++	K-bMOM-kmed
0.4	0.588 (0.028)	**0.995 (0.009)**	**0.995 (0.009)**	0.979 (0.028)	**0.997 (0.015)**	0.970 (0.027)
0.6	0.531 (0.033)	0.883 (0.033)	**0.895 (0.035)**	**0.898 (0.039)**	**0.902 (0.042)**	0.886 (0.048)
0.8	0.476 (0.047)	0.712 (0.047)	0.711 (0.050)	0.735 (0.053)	0.716 (0.053)	0.702 (0.056)
RMSE:						
0.4	1.333 (0.205)	0.612 (0.164)	0.614 (0.179)	0.822 (0.175)	**0.398 (0.093)**	**0.422 (0.147)**
0.6	1.528 (0.242)	0.955 (0.265)	0.943 (0.254)	1.160 (0.259)	**0.737 (0.185)**	**0.739 (0.186)**
0.8	1.860 (0.275)	1.267 (0.351)	1.258 (0.350)	1.521 (0.362)	**1.096 (0.242)**	**1.105 (0.251)**

Table 3
Aggregated performances according to the typology of outliers for different strategies of initialization.

Type of outlier		Initialization	RMSE	Distortion	Accuracy
T1	isolated	random	1.643 (0.370)	4.307 (1.700)	0.538 (0.069)
		K-medians++	0.934 (0.389)	1.887 (1.656)	0.833 (0.137)
		K-means++	0.979 (0.405)	1.752 (1.668)	0.857 (0.137)
		ROBIN	1.351 (1.122)	2.674 (2.273)	0.847 (0.196)
		K-bMOM-km++	**0.702 (0.534)**	**1.421 (1.363)**	**0.894 (0.136)**
		K-bMOM-kmed	**0.727 (0.412)**	**1.491 (1.355)**	**0.871 (0.134)**
T2	oriented & isolated	random	**4.155 (5.653)**	4.652 (1.699)	0.708 (0.046)
		K-medians++	39.53 (39.09)	7.936 (5.574)	0.412 (0.189)
		K-means++	23.38 (33.49)	3.458 (2.339)	0.770 (0.146)
		ROBIN	15.95 (50.41)	7.646 (89.79)	0.635 (0.346)
		K-bMOM-km++	6.552 (9.142)	**1.828 (1.491)**	**0.874 (0.085)**
		K-bMOM-kmed	7.420 (8.819)	**1.972 (1.505)**	**0.849 (0.081)**
T3	cluster of outliers	random	1.505 (0.360)	4.157 (1.597)	0.544 (0.066)
		K-medians++	0.842 (0.358)	1.872 (1.667)	0.810 (0.152)
		K-means++	0.880 (0.360)	2.472 (1.755)	0.756 (0.158)
		ROBIN	1.256 (0.817)	3.847 (4.067)	0.694 (0.330)
		K-bMOM-km++	**0.637 (0.429)**	**1.630 (1.523)**	**0.851 (0.153)**
		K-bMOM-kmed	0.697 (0.421)	1.718 (1.522)	0.800 (0.152)

We are going to focus on the case $\sigma^2 = 0.4$ in order to highlight the benefits and limitations of the proposed initialization approaches, depending on the number of clusters, the percentage and the degree of outliers. Moreover, since the case of isolated outliers (T1) impacts none of the initialization approaches, the specific comments will focus on types of outliers T2 and T3.

Specific comments: sensibility to the type of outliers

The three barplots shown in Fig. 5 indicate the average accuracies of the six initialization approaches for different proportions of type T3 outliers. They have been drawn for different number of groups: $K = 3$ (left side), $K = 5$ (middle) and $K = 10$ (right side). In the easiest configuration ($K = 3$), it is worth noting that K-bMOM-km++ and K-bMOM-kmed++ are on average, the most robust approaches to outliers whatever their proportion. Let us note that K-means++ and ROBIN remain accurate until a proportion of outliers equals 0.01 before dropping out. The same robust behavior of K-bMOM-km++ can be observed in the case T2 in the Supplementary Material.

5.3. Evaluation of the behavior of K-bMOM algorithm

The aim of this section is to evaluate how K-bMOM behaves according to the different scenarios detailed in Section 5.1. Table 4 show the mean and the standard deviation (in parentheses) of RMSE, distortion and accuracy for each type of outlier (T1, T2, T3) with several proportions of outliers. These metrics have been averaged over the different combinations linked to the number of clusters, the dimension of data and the level of cluster separability.

First of all, it can be observed that isolated outliers have almost no influence on the clustering results. On average, the performances remain really close to that obtained on regular data irrespective of the percentage of outliers. This is mainly explained by the distribution of outliers which is well-dispersed around regular data from the isotropic Gaussian mixture model and therefore, it does not affect the clustering task. Secondly, concerning the T2 and T3 type outliers, it can be noted that the accuracy rate remains over 0.90 for a proportion of outliers lower than 3%. This behavior is explained by the limitations seen in Section 4.1. Moreover, the K-bMOM algorithm seems to be more robust in the configuration T3 of a cluster of outliers than in the configuration T2 of isolated and oriented outliers, as can be seen on the decrease of accuracies in Table 4.

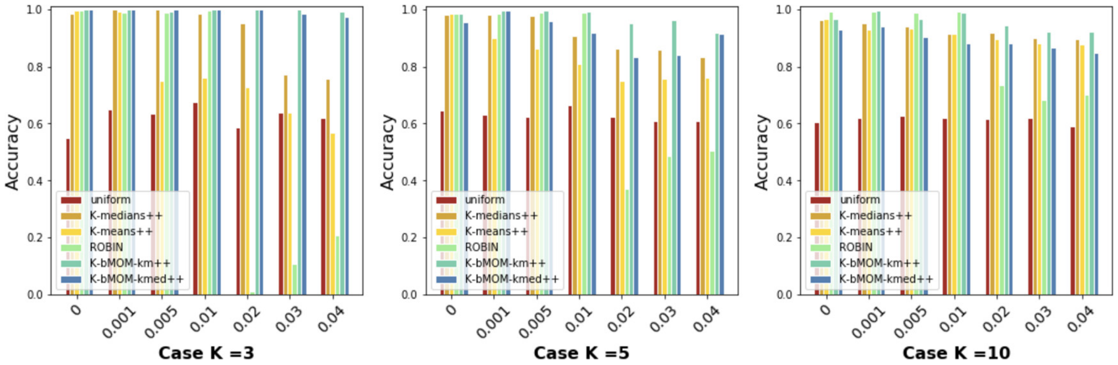

Fig. 5. Case: cluster of outliers (T3). Comparison of the average accuracies per initialization approach depending on the proportion of outliers and the number of clusters ($K = 3$, $K = 5$ and $K = 10$) from the left to the right.

Table 4
Aggregated performance of K-bMOM as a function of the percentage of outliers for type T1 (isolated), type T2 (oriented and isolated) and type T3 (clustered) of outliers.

Type of outlier		m/n	RMSE	Distortion	Accuracy
	regular data	0	0.181 (0.086)	0.913 (0.554)	**0.993 (0.011)**
		0.001	0.186 (0.086)	0.914 (0.556)	**0.993 (0.010)**
		0.005	0.190 (0.117)	0.911 (0.554)	**0.992 (0.015)**
T1	isolated	0.01	0.199 (0.099)	0.904 (0.548)	**0.992 (0.015)**
		0.02	0.205 (0.090)	0.912 (0.552)	**0.993 (0.011)**
		0.03	0.210 (0.095)	0.922 (0.557)	**0.991 (0.016)**
		0.04	0.215 (0.099)	0.930 (0.560)	**0.990 (0.020)**
		0.001	0.293 (0.366)	1.297 (2.759)	**0.981 (0.065)**
		0.005	0.299 (0.362)	1.299 (2.782)	**0.980 (0.063)**
T2	oriented & isolated	0.01	0.393 (0.515)	1.552 (2.860)	**0.937 (0.107)**
		0.02	0.465 (0.575)	1.808 (2.921)	**0.911 (0.131)**
		0.03	0.674 (0.709)	2.309 (3.283)	0.864 (0.146)
		0.04	0.885 (0.713)	2.518 (2.798)	0.814 (0.146)
		0.001	0.186 (0.086)	0.914 (0.556)	**0.993 (0.010)**
		0.005	0.213 (0.148)	0.978 (0.571)	**0.984 (0.036)**
T3	cluster of outliers	0.01	0.215 (0.151)	0.959 (0.561)	**0.976 (0.039)**
		0.02	0.260 (0.177)	1.016 (0.609)	**0.966 (0.055)**
		0.03	0.416 (0.221)	1.247 (0.694)	0.898 (0.091)
		0.04	0.480 (0.247)	1.330 (0.732)	0.882 (0.089)

The following remarks focus on the cases $\sigma^2 = 0.4$ to visualize the evolution of performances of the K-bMOM algorithm according to the number of clusters, the proportion and the type of outliers.

We consider on the one hand, the case of isolated oriented outliers (T2) that is displayed in Figs. 6, 7 and 8. They represent violin plots of accuracies, RMSE and number of clusters fitted by the K-bMOM algorithm for a number of clusters equal to $K = 3$, $K = 5$ and $K = 10$ respectively and for several proportions of outliers. As can be observed, for a low number of groups ($K \in \{3, 5\}$), the K-bMOM algorithm is really robust to a proportion of outliers up to 3%. As expected (see Section 4.1), when the number of clusters is high ($K = 10$), the procedure remains robust for a lower proportion of outliers ($m/n \leq 0.005$).

On the other hand, the case of a cluster of outliers (T3) is displayed in Figs. 9, 10 and 11. Again, for $K = 3$ and $K = 5$, we can observe that the procedure is really robust and stable for a proportion of outliers lower than 4%. In the case $K = 10$, the K-bMOM algorithm resists very well to the increasing proportion of outliers and appears to be more robust in this typology of outliers than for type T2.

6. Benchmark K-means type robust clustering algorithm

The objective of this section is to compare the performance of the K-bMOM strategy in its scope of application with the robust clustering algorithms based on K-means approaches on a specific framework with isolated oriented outliers.

Benchmark algorithms

We consider six different algorithms: our proposed robust clustering algorithm named K-bMOM with a time complexity $\mathcal{O}(Kn_B Bp)$ at each iteration, the traditional K-means for comparison and also four well-known robust versions of the K-means described below:

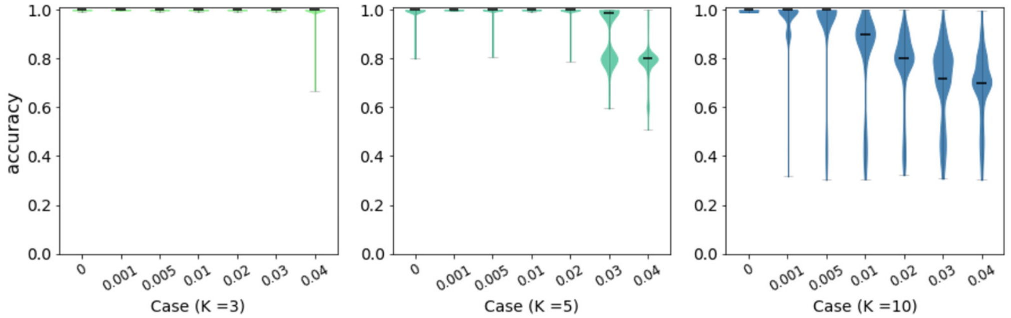

Fig. 6. Violin plots of K-bMOM accuracies computed on the fitted partition among the regular data, for different proportions of T2 outliers, for different sample sizes, outlier degrees and dimensions, for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$.

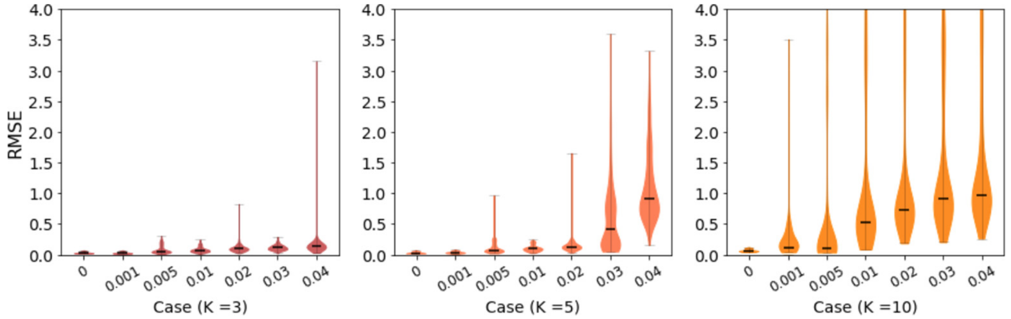

Fig. 7. Violin plots of K-bMOM RMSE computed on the fitted partition among the regular data, for different proportions of T2 outliers, for different sample sizes, outlier degrees and dimensions, for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$.

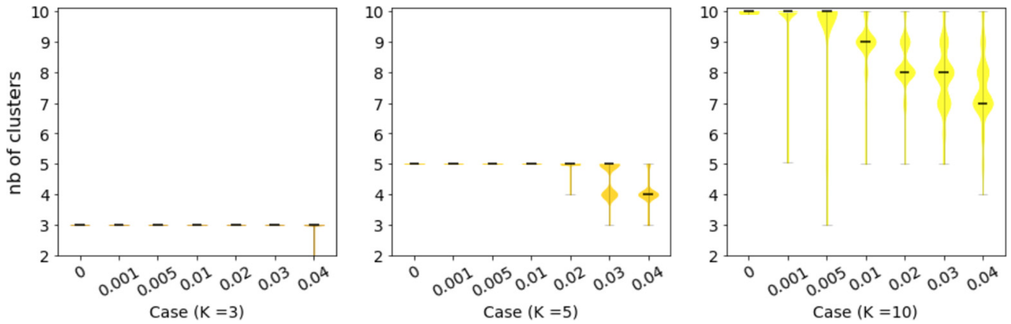

Fig. 8. Violin plots of the number of clusters computed on the fitted partition among the regular data for different proportions of T2 outliers for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$, all scenarios included.

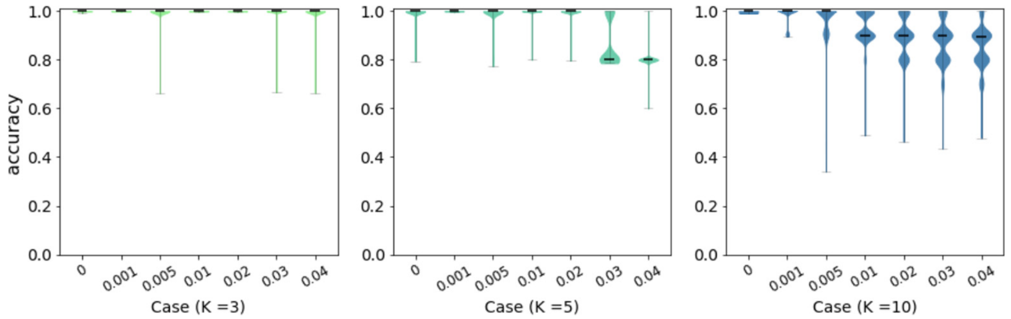

Fig. 9. Violin plots of K-bMOM accuracies obtained on the fitted partition among the regular data, for different proportion of T3 outliers, for different sample sizes, outlier degrees and dimensions for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$.

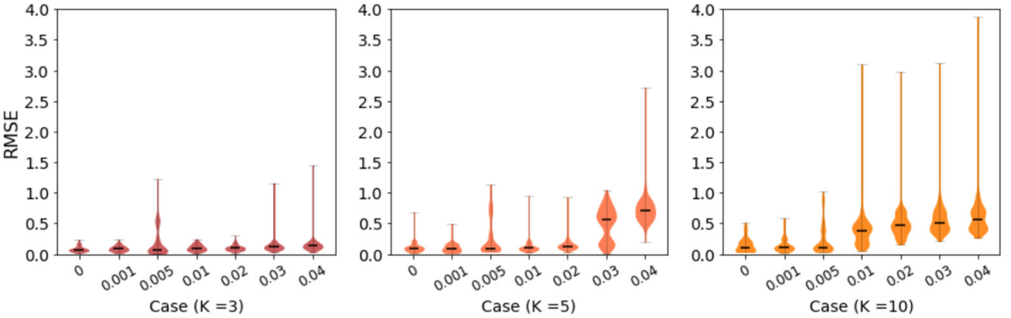

Fig. 10. Violin plots of K-bMOM RMSE obtained on the fitted partition among the regular data, for different proportions of T3 outliers, for different sample sizes, outlier degrees and dimensions for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$.

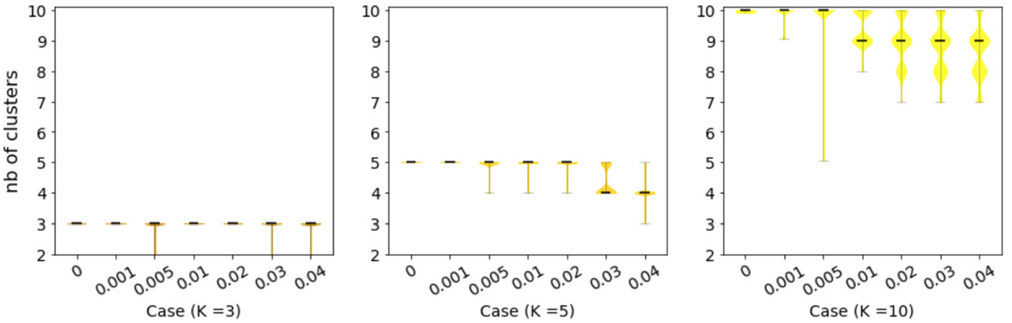

Fig. 11. Violin plots of the number of clusters obtained on the fitted partition for different proportion of T3 outliers, among different sample sizes, outlier degrees and dimensions for $K = 3$ (left), $K = 5$ (middle) and $K = 10$ (right) with cluster separability $\sigma^2 = 0.4$.

- K-medoids which aims at finding K data points as centers such that the within inertia is minimized. This optimization process is done according to the Partition Around Medoids algorithm named PAM (Kaufman and Rousseeuw, 1987). It has a complexity dominated by $\mathcal{O}\left(K(n-K)^2 p\right)$ per iteration. Faster versions have been proposed in Schubert and Rousseeuw (2019).
- K-medians which is a robust variant of the K-means algorithm (Jain and Dubes, 1988): in the aggregation step, instead of computing the barycenter of each group as in the K-means procedure, K-medians computes in each single dimension, the median based on the Manhattan-distance formulation. The complexity of such a procedure is dominated by $\mathcal{O}(nKp)$ per iteration as for Loyd's algorithm (Hartigan and Wong, 1979).
- trimmed-K-means which is an EM-like algorithm introduced by Cuesta-Albertos et al. (1997) in the late 90s. It benefits robustness properties from the trimming action during the maximisation step where only a proportion $1 - m/n$ of the closest data point from their assigned centroid, is taken into account. Since the trimming needs to sort the data points according to their distance to centroid, it leads therefore to an overall complexity of $\mathcal{O}(Knp + n\log n)$ at each iteration.
- K-PDTM which is a robust quantization algorithm introduced by Brécheteau (2018), aims to infer the manifold from which the data points are drawn. This inference is done by means of K centroids that should be on the manifold if the algorithm runs well. It is based also on a Lloyd-type algorithm where in the updating step, the centroid is computed as the barycenter of the s nearest neighbors of the barycenter of the cluster. In the assignment step, the data point is assigned according to a Bregman divergence. This algorithm has two hyper-parameters: s, the number of neighbors used to compute the centroid and the number of clusters K. This leads to the following complexity for one iteration: $\mathcal{O}(Ksp + n\log n)$.

The implementations used for the clustering approaches to compare the MOM-based ones in this experiment are publicly available. Table 5 details the programming languages and associated libraries used as well as selected hyper-parameters. Let us note that the trimming approach from the TRIMCLUSTER (Hennig, 2021) R package has been implemented in Python language in order to be able to play easily with the initial conditions.

Simulation context

We dispose of $N = 1500$ points of dimension $p = 3$ which are generated according to a mixture of $K \in \{3, 4, 5\}$ multivariate Gaussian density functions with an isotropic covariance matrix. The average vectors for the K components are respectively $\mu_1 = [0, 1, 4]$, $\mu_2 = [2, 1, 0]$, $\mu_3 = [0, -2, 3]$, for $K = 3$, $\mu_4 = [0, 5, -5]$ is added in the case $K = 4$, then $\mu_5 = [-1, -2, 0]$ for $K = 5$. Isolated outliers have been generated by randomly taken 30 datapoints from the regular dataset

Table 5
Implementations and hyper-parameters.

Algorithm	Language	hyper-parameters
K-means	Python (Pedregosa et al., 2011)	K, $init$ =given*,$n\_init = 1$
K-medoids	Python (Novikov, 2019)	$initial\_index\_medoids$ = given*
K-medians	Python (Novikov, 2019)	$initial\_centers$ =given*
trimK-means	R (Cuesta-Albertos et al., 1997; Hennig, 2021)	K, $trim = m/n$, $runs = n$, $points$ =given*,$maxit = 300$
K-pdtm	Python (Brécheteau, 2018; Brecheteau, 2020)	K, $query\_pts$ =given*,$s = K$, $k = K$, $sig = N - m$, $iter\_max = 300$, $nstart = 1$,$leaf\_size = 30$
K-bMOM	Python (Brunet-Saumard and Genetay, 2021)	K, $n_B = 5K$, $B = 500$, $iter\_max = 25$, $initial\_centers$ =given*

*given: same centers obtained either with a random initialization or K-bMOM-km++

Table 6
Average performances and standard deviations (in brackets) of K-means and the five robust clustering approaches on regular data with the same random initializations, among $K \in \{3, 4, 5\}$.

	Average performances on regular data (m/n = 0)					
	K-means	K-pdtm	trimK-means	K-medians	K-medoids	K-bMOM
RMSE	0.071 (0.367)	0.068 (0.384)	0.071 (0.366)	0.726 (0.379)	0.173 (0.579)	0.200 (0.03)
distortion	1.096 (0.671)	1.093 (0.911)	1.144 (0.821)	1.700 (1.241)	1.120 (1.505)	1.111 (0.024)
accuracy	0.995 (0.111)	0.995 (0.127)	0.995 (0.111)	0.989 (0.088)	0.995 (0.121)	0.996 (0.001)

Table 7
Average accuracies and standard deviations of clustering approaches depending on the number of clusters and the initialization scheme with type (1) random initialization, type (2) K-means++ and type (3) K-bMOM-km++.

K	init	K-means	K-pdtm	trimK-means	K-medians	K-medoids	K-bMOM
3	1	0.819 (0.162)	0.938 (0.151)	0.938 (0.156)	0.915 (0.082)	0.864 (0.161)	0.943 (0.123)
	2	0.613 (0.124)	0.931 (0.131)	0.833 (0.124)	0.612 (0.123)	0.613 (0.124)	0.810 (0.163)
	3	**0.994 (0.006)**	0.942 (0.123)	**0.997 (0.001)**	0.980 (0.059)	**0.995 (0.003)**	**0.997 (0.002)**
4	1	0.869 (0.124)	0.889 (0.142)	0.869 (0.124)	0.903 (0.116)	0.839 (0.120)	0.879 (0.124)
	2	0.502 (0.001)	0.916 (0.116)	0.915 (0.056)	0.502 (0.001)	0.502 (0.001)	0.958 (0.001)
	3	**0.988 (0.049)**	**0.989 (0.021)**	**0.988 (0.049)**	**0.984 (0.0498)**	**0.988 (0.049)**	**0.988 (0.049)**
5	1	0.820 (0.110)	0.860 (0.155)	0.900 (0.114)	0.896 (0.104)	0.854 (0.113)	**0.961 (0.075)**
	2	0.534 (0.093)	0.894 (0.141)	0.777 (0.134)	0.520 (0.097)	0.534 (0.093)	**0.996 (0.002)**
	3	0.965 (0.069)	0.898 (0.132)	**0.996 (0.002)**	0.971 (0.058)	0.984 (0.042)	**0.996 (0.002)**

and their coordinates have been multiplied by a factor of +/-10. All the algorithms have been initialized with the exact same conditions: a random initialization, a K-means++ strategy and a K-bMOM-km++ strategy presented in Section 5.2. These conditions have been repeated 1000 times and in order to compare the performances of these algorithms, the RMSE, the distortion and the accuracy have been computed based on the true parameters of data distribution and their label membership amongst the regular data. Moreover, the average number of clusters found amongst the regular data have also been computed.

Results and analysis

First of all, let us note that the clustering algorithms have been executed on regular data with the same random initializations. As can be seen in Table 6, all the methods perform equally well in average and whatever the number of clusters, when there is no outlier.

The average accuracies calculated for the different number of groups are summarized in Table 7. As can be seen, the initialization scheme affects most of the clustering approaches and the more sensitive ones are the K-means, K-medians and K-medoids procedures with a K-means++ initialization. The average accuracies of K-bMOM and trimK-means decrease slightly when initialized by a K-means++ procedure. On the other hand, K-pdtm seems to be quite insensitive to the tested initialization schemes.

The proposed K-bMOM-km++ strategy appears to be a very good initialization strategy and K-bMOM is also a sensible strategy when only purely random initializations are considered. These two advantages are reflected in the detailed analysis of the case $K = 5$ where the average RMSE, distortions and accuracies are summarized in Table 8. Let us note that the illustrations of distributions of 1000 repetitions for each metric and tested algorithm according to violin plots, are available in the Supplementary Material (Brunet-Saumard et al., 2022). Again, we can observe the strong stability of K-bMOM performances in this configuration whatever the initialization process. The average accuracy is about 0.99 and the average RMSE and distortions are the lowest amongst the tested procedure. Let us note that the detailed contents of performances for the cases $K = 3$ and $K = 4$ are available in the Supplementary Material (Brunet-Saumard et al., 2022).

Table 8
Averages and standard deviations of the performances of the five robust clustering algorithms on polluted data with a random initialization (left), a K-means++ initialization (middle) and the proposed robust initialization (K-bMOM-km++) in the case $K = 5$.

Initialization	Methods	RMSE	Distortion	Accuracy	nb K
random initialization	K-means	1.076 (0.264)	2.560 (0.680)	0.820 (0.110)	4.1 (0.5)
	K-pdtm	0.508 (0.491)	1.993 (1.199)	0.860 (0.155)	**4.9 (0.3)**
	trimK-means	0.389 (0.385)	1.577 (0.662)	0.900 (0.114)	**4.8 (0.3)**
	K-medians	0.993 (0.559)	2.498 (1.569)	0.896 (0.104)	4.7 (0.4)
	K-medoids	1.068 (0.32)	2.535 (1.042)	0.854 (0.113)	4.7 (0.4)
	K-bMOM	**0.290 (0.236)**	**1.284 (0.375)**	**0.961 (0.075)**	**4.9 (0.2)**
K-means++ initialization	K-means	1.669 (0.271)	4.881 (1.09)	0.534 (0.093)	2.6 (0.5)
	K-pdtm	0.385 (0.393)	1.742 (0.977)	0.884 (0.141)	**5.0 (0.0)**
	trimK-means	0.624 (0.452)	1.836 (1.437)	0.857 (0.134)	4.9 (0.3)
	K-medians	1.828 (0.785)	7.650 (2.454)	0.520 (0.097)	2.6 (0.4)
	K-medoids	1.686 (0.449)	5.128 (1.203)	0.534 (0.093)	2.6 (0.4)
	K-bMOM	**0.186 (0.034)**	**1.105 (0.025)**	**0.996 (0.002)**	**5.0 (0.0)**
K-bMOM-km++ initialization	K-means	0.882 (0.122)	1.891 (0.286)	0.965 (0.069)	4.8 (0.3)
	K-pdtm	0.427 (0.429)	1.743 (0.871)	0.888 (0.132)	4.9 (0.2)
	trimK-means	**0.063 (0.011)**	**1.067 (0.019)**	**0.996 (0.002)**	**5.0 (0.0)**
	K-medians	0.738 (0.228)	1.638 (0.365)	0.971 (0.058)	4.9 (0.3)
	K-medoids	0.819 (0.154)	1.721 (0.205)	0.984 (0.042)	**5.0 (0.0)**
	K-bMOM	0.185 (0.025)	**1.096 (0.022)**	**0.996 (0.002)**	**5.0 (0.0)**

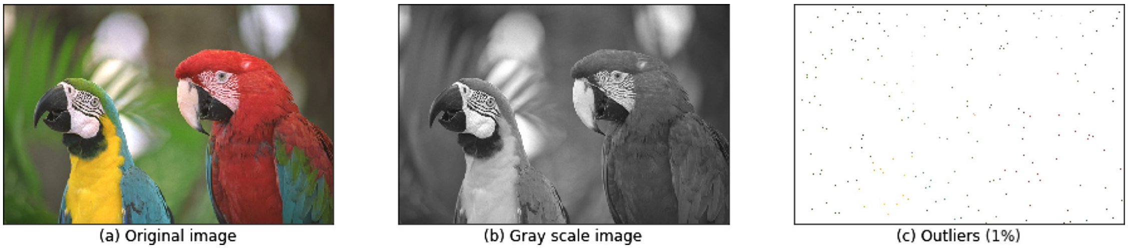
(a) Original image
(b) Gray scale image
(c) Outliers (1%)

Fig. 12. Original Image (left), grayscale image (middle) and a filter of outliers applied on the grayscale image (right).

7. Color quantization in image processing

We have seen in Section 5.2 that using a robust initialization even in the context of few outliers, is the best strategy in terms of resulting partition stability and accuracy. Given that spirit, in this last experimental section, the K-bMOM procedure is applied to the problem of color quantization addressed in image processing and computer graphics.

Color quantization (CQ) is a procedure commonly used for color analysis, image compression, segmentation, non-photorealistic rendering, etc. It is a process which aims to reduce the number of colors used in an image with the goal of keeping the same quality of visualisation as the original. CQ is a challenging problem since most real-world images contain tens of thousands of colors. CQ can be viewed as a 3-dimensional clustering problem according to the Red, Green, Blue channels of pixels of an image. A wide literature is devoted to this problem and it appears that the K-means algorithm is not used so often mainly because of its sensitivity to the initialization. We propose therefore to use the K-bMOM procedure as a robust CQ process providing confident and high-quality quantization on a noisy image.

7.1. Images and experimental setup

The K-bMOM algorithm has been used on a popular 24-bit test image, the Parrots, (768 × 512) coming from the Kodak Lossless True Color Image Suite database and illustrated in Fig. 12(a).

The K-bMOM procedure has been used on the grayscale image of parrots as shown in Fig. 12(b) from which 1% of randomized pixels have retained their color from the original image. In our experiment, we consider these colored pixels, illustrated in Fig. 12(c), as outliers of the grayscale image. The objective is therefore to extract the main shades of gray from the image. The image of size $768 \times 512 \times 3$ has been shaped into a 3-dimensional matrix. The K-bMOM algorithm has been repeated 50 times for a number K of gray levels (or clusters) equals to 8, 16 and 24 respectively. For these three segmentations, the number of blocks has been set to $B = 1000$ and the size of each block set to $n_B = 5 * K$.

Table 9
Median and standard deviation (in parentheses) of the distortion and the number of gray levels obtained by the K-bMOM procedure for $K = \{8, 16, 24\}$ groups.

		$K = 8$	$K = 16$	$K = 24$
K-bMOM	distortion	171.0 (1.69)	**56.40** (27.01)	**28.70** (3.42)
	number of gray levels	8.00 (0.00)	**16.00** (0.58)	**24.00** (0.00)
K-means	distortion	171.5 (1.53)	67.20 (17.77)	42.50 (2.36)
	number of gray levels	8.00 (0.00)	15.00 (0.55)	22.00 (0.52)

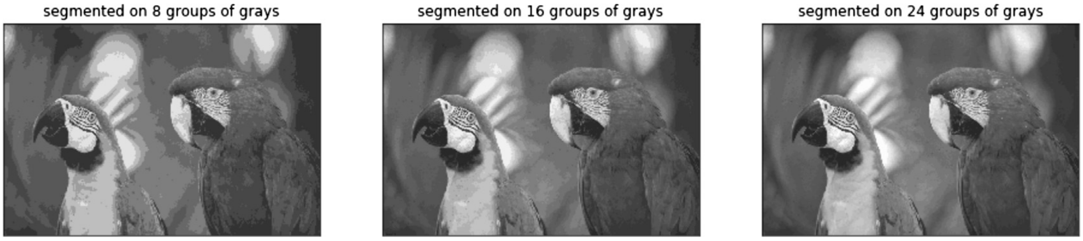
segmented on 8 groups of grays
segmented on 16 groups of grays
segmented on 24 groups of grays

Fig. 13. Sample quantization results for $K = 8$, 16 and 24 respectively from left to right on grayscale noisy Parrot image.

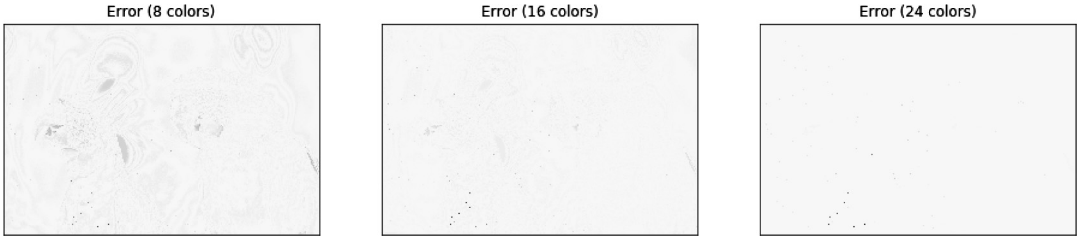

Fig. 14. Full scale error images for $K = 8$, 16 and 24 respectively from left to right grayscale Parrot image with K-bMOM procedure.

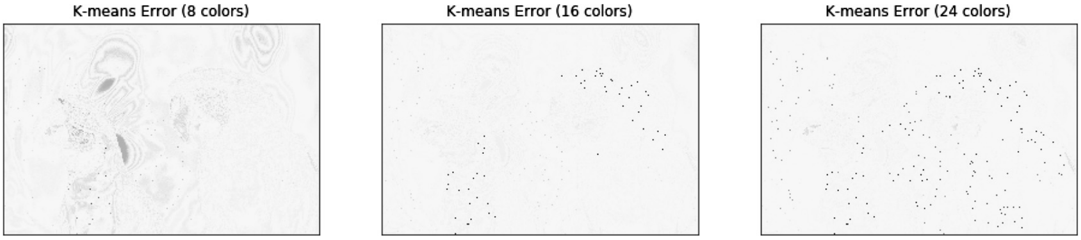

Fig. 15. Full scale error images for $K = 8$, 16 and 24 respectively from left to right grayscale Parrot image with a K-means procedure.

7.2. Experimental results

In order to evaluate the quality of the quantization, the empirical distortion has been computed between the pixels $x_i \in \mathbb{R}^3,\ i \in 1, \dots, n$, of the original grayscale image and their segmented version $\bar{c}_k^{(bmed)}$ returned by the K-bMOM procedure, i.e. their nearest color. It has been averaged amongst 50 repetitions and the standard deviation has also been computed. Moreover, in order to evaluate the robustness property of the K-bMOM algorithm, the number of gray values amongst the centroids is displayed. It is expected to have K levels of gray (i.e. no color amongst the centroids). In order to benchmark the K-bMOM algorithm, the traditional K-means algorithm has been executed under the same experimental conditions.

The results are summarized in Table 9. First of all, it can be noted that color quantization processed by the K-bMOM approach seems to be robust. Indeed, the centroids are in the shades of gray: the number of gray levels equals the number of clusters.

Fig. 13 illustrates the quantization process on a grayscale Parrot image with outliers for $K = 8$, $K = 16$ and $K = 24$ respectively. Figs. 14 and 15 show the error per pixel in a reverse grayscale mapping for the K-bMOM and K-means procedures respectively. The higher the error, the darker the pixel. It can be seen that the K-bMOM approach performs well in allocating K-representative gray levels to the different image regions while the K-means procedure fails, by selecting pixels of outliers as centers.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2021.107370.

References

Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.J., 2009. Robust partitional clustering by outlier and density insensitive seeding. Pattern Recognit. Lett. 30 (11), 994–1002.

Alon, N., Matias, Y., Szegedy, M., 1999. The space complexity of approximating the frequency moments. In: Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, PA, 1996. J. Comput. Syst. Sci. 58, 137–147. https://doi.org/10.1006/jcss.1997.1545.

Arthur, D., Vassilvitskii, S., 2007. K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07, pp. 1027–1035.

Baudry, J.-P., Maugis, C., Michel, B., 2012. Slope heuristics: overview and implementation. Stat. Comput. 22 (2), 455–470.

Brécheteau, C., 2018. Robust shape inference from a sparse approximation of the Gaussian trimmed loglikelihood. Preprint.

Brecheteau, C., 2020. https://www.math.sciences.univ-nantes.fr/~brecheteau/notebooks/. Notebook_kPDTM_kPLM.html.

Brunet-Saumard, C., Genetay, E., 2021. https://github.com/csaumard/kbmom.

Brunet-Saumard, C., Genetay, E., Saumard, A. Supplement to: "K-bMOM: a robust Lloyd-type clustering algorithm based on bootstrap Median-of-Means".

Bühlmann, P., 2003. Bagging, subagging and bragging for improving some prediction algorithms. In: Recent Advances and Trends in Nonparametric Statistics. Elsevier B.V., Amsterdam, pp. 19–34.

Cuesta-Albertos, J.A., Gordaliza, A., Matrán, C., 1997. Trimmed k-means: an attempt to robustify quantizers. Ann. Stat. 25 (2), 553–576.

del Barrio, E., Cuesta-Albertos, J.A., Matrán, C., Mayo-Íscar, A., 2019. Robust clustering tools based on optimal transportation. Stat. Comput. 29 (1), 139–160.

Devroye, L., Lerasle, M., Lugosi, G., Oliveira, R.I., 2016. Sub-Gaussian mean estimators. Ann. Stat. 44 (6), 2695–2725.

Diakonikolas, I., Kane, D.M., 2019. Recent advances in algorithmic high-dimensional robust statistics. arXiv:1911.05911.

Dolnicar, S., Leisch, F., 2003. Winter tourist segments in Austria: identifying stable vacation styles using bagged clustering techniques. J. Travel Res. 41 (3), 281–292.

D'Urso, P., De Giovanni, L., Disegna, M., Massari, R., 2013. Bagged clustering and its application to tourism market segmentation. Expert Syst. Appl. 40 (12), 4944–4956.

García-Escudero, L.A., Gordaliza, A., 1999. Robustness properties of k means and trimmed k means. J. Am. Stat. Assoc. 94 (447), 956–969.

García-Escudero, L.A., Gordaliza, A., Matrán, C., Mayo-Iscar, A., et al., 2008. A general trimming approach to robust cluster analysis. Ann. Stat. 36 (3), 1324–1345.

García-Escudero, L.A., Gordaliza, A., Matrán, C., Mayo-Iscar, A., 2010. A review of robust clustering methods. Adv. Data Anal. Classif. 4 (2–3), 89–109.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A., 1986. Robust Statistics: The Approach Based on Influence Functions. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York.

Hartigan, J., Wong, M., 1979. Algorithm AS 136: a K-means clustering algorithm. Appl. Stat., 100–108.

Hennig, C., 2021. trimcluster: cluster analysis with trimming. R package version 0.1-5.

Huber, P.J., Ronchetti, E.M., 2009. Robust Statistics, 2nd edition. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ.

Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs.

Jerrum, M.R., Valiant, L.G., Vazirani, V.V., 1986. Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. 43 (2–3), 169–188. https://doi.org/10.1016/0304-3975(86)90174-X.

Kaufman, L., Rousseeuw, P.J., 1987. Clustering by means of medoids. In: Statistical Data Analysis Based on the L1 Norm and Related Methods. North-Holland, Amsterdam, pp. 405–416.

Klochkov, Y., Kroshnin, A., Zhivotovskiy, N., 2020. Robust k-means clustering for distributions with two moments. arXiv preprint. arXiv:2002.02339v1.

Laforgue, P., Clémençon, S., Bertail, P., 2019. On medians of (randomized) pairwise means. In: 36th International Conference on Machine Learning, vol. 97.

Lecué, G., Lerasle, M., 2019. Learning from MOM's principles: Le Cam's approach. Stoch. Process. Appl. 129 (11), 4385–4410.

Lecué, G., Lerasle, M., 2020. Robust machine learning by median-of-means: theory and practice. Ann. Stat. 48 (2), 906–931.

Leisch, F., 1999. Bagged clustering. Working Paper 51, SFB "Adaptive Information Systems and Modeling in Economics and Management Science". http://epub.wu.ac.at/1272/1/document.pdf.

Lerasle, M., Oliveira, R.I., 2011. Robust empirical mean estimators. arXiv preprint. arXiv:1112.3914.

Lugosi, G., Mendelson, S., 2019a. Sub-Gaussian estimators of the mean of a random vector. Ann. Stat. 47 (2), 783–794.

Lugosi, G., Mendelson, S., 2019b. Mean estimation and regression under heavy-tailed distributions: a survey. Found. Comput. Math. 19 (5), 1145–1190.

Lugosi, G., Mendelson, S., 2020. Risk minimization by median-of-means tournaments. J. Eur. Math. Soc. 22 (3), 925–965.

Maronna, R.A., Martin, R.D., Yohai, V.J., Salibián-Barrera, M., 2019. Robust Statistics. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ.

Minsker, S., 2018. Uniform bounds for robust mean estimators. arXiv preprint. arXiv:1812.03523.

Nemirovsky, A.S., Yudin, D.B., 1983. Problem Complexity and Method Efficiency in Optimization. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Inc., New York. Translated from the Russian and with a preface by E.R. Dawson.

Nguyen, N., Caruana, R., 2007. Consensus clusterings. In: Seventh IEEE International Conference on Data Mining. ICDM 2007. IEEE, pp. 607–612.

Novikov, A., 2019. PyClustering: data mining library. J. Open Sour. Softw. 4 (36), 1230.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Ritter, G., 2015. Robust Cluster Analysis and Variable Selection. Monographs on Statistics and Applied Probability, vol. 137. CRC Press, Boca Raton, FL.

Rodriguez, D., Valdora, M., 2019. The breakdown point of the median of means tournament. Stat. Probab. Lett. 153, 108–112.

Schubert, E., Rousseeuw, P.J., 2019. Faster k-medoids clustering: improving the PAM, CLARA, and CLARANS algorithms. arXiv preprint. arXiv:1810.05691.

Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc., Ser. B, Stat. Methodol. 63 (2), 411–423.