



**HAL**  
open science

# A penalized criterion for selecting the number of clusters for K-medians

Antoine Godichon-Baggioni, Sobihan Surendran

► **To cite this version:**

Antoine Godichon-Baggioni, Sobihan Surendran. A penalized criterion for selecting the number of clusters for K-medians. 2024. hal-03771959v3

**HAL Id: hal-03771959**

**<https://hal.science/hal-03771959v3>**

Preprint submitted on 26 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A penalized criterion for selecting the number of clusters for K-medians

Antoine Godichon-Baggioni and Sobihan Surendran,  
Laboratoire de Probabilités, Statistique et Modélisation  
Sorbonne-Université, 75005 Paris, France  
antoine.godichon\_baggioni@upmc.fr, sobihan.surendran@sorbonne-universite.fr

## Abstract

Clustering is a usual unsupervised machine learning technique for grouping the data points into groups based upon similar features. We focus here on unsupervised clustering for contaminated data, i.e in the case where K-medians algorithm should be preferred to K-means because of its robustness. More precisely, we concentrate on a common question in clustering: how to chose the number of clusters? The answer proposed here is to consider the choice of the optimal number of clusters as the minimization of a penalized criterion. In this paper, we obtain a suitable penalty shape for our criterion and derive an associated oracle-type inequality. Finally, the performance of this approach with different types of K-medians algorithms is compared on a simulation study with other popular techniques. All studied algorithms are available in the R package `Kmedians` on CRAN.

**Keywords:** Clustering, K-medians, Robust statistics

## 1 Introduction

Clustering is an unsupervised machine learning technique that involves grouping data points into a collection of groups based on similar features. Clustering is commonly used for data compression in image processing, which is also known as vector quantization (Gersho and Gray, 2012). There is a vast literature on clustering techniques, and general references regarding clustering can be found in Spath (1980); Jain and Dubes (1988); Mirkin (1996); Jain et al. (1999); Berkhin (2006); Kaufman and Rousseeuw (2009). Classification methods can be categorized as hard clustering also referred as crisp clustering (including K-means, K-medians, and hierarchical clustering) and soft clustering (such as Fuzzy K-means (Dunn, 1973; Bezdek, 2013) and Mixture Models). In hard clustering methods, each data point belongs to only one group, whereas in soft clustering, a probability or likelihood of a data point belonging to a cluster is assigned, allowing each data point to be a member of more than one group.

We focus here on hard clustering methods. The most popular partitioning clustering methods are the non sequential (Forgy, 1965) and the sequential (MacQueen, 1967) versions of the K-means algorithms. The aim of the K-means algorithm is to minimize the sum of squared distances between the data points and their respective cluster centroid. More precisely, considering  $X_1, \dots, X_n$  random vectors taking values in  $\mathbb{R}^d$ , the aim is to find  $k$  centroids  $\{c_1, \dots, c_k\}$  minimizing the empirical distortion

$$\frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, k} \|X_i - c_j\|^2. \quad (1)$$

Nevertheless, in many real-world applications, data collected may be contaminated with outliers of large magnitude, which can make traditional clustering methods such as K-means sensitive to their presence. As a result, it is necessary to use more robust clustering algorithms that produce reliable outcomes. One such algorithm is K-medians clustering, which was introduced by MacQueen (1967) and further developed by Kaufman and Rousseeuw (2009). Instead of using the mean to determine the centroid of each cluster,

K-medians clustering uses the geometric median. It consists in considering criteria based on least norms instead of least squared norms. More precisely, considering the same sequence of i.i.d copies  $X_1, \dots, X_n$ , the objective of K-medians clustering is to minimize the empirical  $L^1$ -distortion :

$$\frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, k} \|X_i - c_j\|.$$

In practical applications, the number of clusters  $k$  is often unknown. In this paper, we will focus on the choice of optimal number of clusters for robust clustering. Several methods for determining the optimal number of clusters have been studied for K-means algorithms and can be easily adapted for K-medians. One commonly used method for determining the optimal number of clusters is the elbow method. Other methods often used are the Silhouette (Kaufman and Rousseeuw, 2009) and the Gap Statistic (Tibshirani et al., 2001). The silhouette coefficient of a sample is defined as the difference between the within-cluster distance between the sample and other data points in the same cluster and the inter-cluster distance between the sample and the nearest cluster. The Silhouette method suggests selecting the value of  $k$  that maximizes the average silhouette coefficient of all data points. The silhouette score is typically calculated using Euclidean or Manhattan distance. Regarding the Gap Statistic, the idea is to compare the within-cluster dispersion to its expected value under an appropriate null reference distribution. The reference data set is generated via Monte Carlo simulations of the sampling process.

In Fischer (2011), the objective is to minimize the empirical distortion, which is defined in (1), as a function of  $k$  in order to determine the optimal number of clusters. However, if all data points are placed in a single cluster, the empirical distortion will be minimized. To prevent choosing too large a value for  $k$ , a penalty function is introduced. It was shown that the penalty shape is  $\sqrt{\frac{k}{n}}$  in the case of K-means clustering and by finding the constant of the penalty with the data-based calibration method, one can obtain better results than by using usual other methods. The data-driven calibration algorithm is a method proposed by Birgé and Massart (2007) and developed by Arlot and Massart (2009), to find the constant of penalty function. Theoretical properties on this data-based penalization procedures have been studied by Birgé and Massart (2007); Arlot and Massart (2009); Baudry et al. (2012). The aim of this paper is to adapt these methods for K-medians algorithms. We first provide the shape of the penalty function and then use the slope heuristic method to calibrate the constant and construct a penalized criterion for selecting the number of clusters for K-medians algorithms.

The paper is organized as follows. We provide a recap of two different methods for estimating the geometric median, followed by the introduction of three K-median algorithms (“Online”, “Semi-Online”, and “Offline”). In section 3, we propose a penalty shape for the proposed penalized criterion and give an upper bound for the expectation of the distortion at empirically optimal codebook with size of optimal number of clusters which ensure our penalty function. We illustrate the proposed approach with some simulations and compare it with several methods in section 4. Finally, the proofs are gathered in section 5. All the proposed algorithms are available in the R package `Kmedians` on CRAN <https://cran.r-project.org/package=Kmedians>.

## 2 Framework

### 2.1 Geometric Median

In what follows, we consider a random variable  $X$  that takes values in  $\mathbb{R}^d$  for some  $d \geq 1$ . It is well-known that the standard mean of  $X$  is not robust to corruptions. Hence, the median is preferred to the mean in robust statistics. The geometric median  $m$ , also called  $L^1$ -median or spatial median, of a random variable  $X \in \mathbb{R}^d$  is defined by Haldane (1948) as follows:

$$m = \arg \min_{u \in \mathbb{R}^d} \mathbb{E} [\|X - u\|].$$

For the 1-dimensional case, the geometric median coincides with the usual median in  $\mathbb{R}$ . As Euclidean space  $\mathbb{R}^d$  is strictly convex, the geometric median  $m$  exists and is unique if the points are not concentrated around a straight line (Kemperman, 1987). The geometric median is known to be robust and has a breakdown point of 0.5.

Let us now consider a sequence of i.i.d copies  $X_1, \dots, X_n$  of  $X$ . In this paper, we focus on two methods to determine the geometric median. The first one is iterative and consists in considering the fix point estimates (Weiszfeld, 1937; Vardi and Zhang, 2000)

$$\hat{m}_{t+1} = \frac{\sum_{i \in \mathcal{X}_t} \frac{X_i}{\|X_i - \hat{m}_t\|}}{\sum_{i \in \mathcal{X}_t} \frac{1}{\|X_i - \hat{m}_t\|}}$$

with a initial point  $\hat{m}_0 \in \mathbb{R}^d$  chosen arbitrarily such that it does not coincide with any of the  $X_i$  and  $\mathcal{X}_t = \{i, X_i \neq \hat{m}_t\}$ . This Weiszfeld algorithm can be a flexible technique, but there are many implementation difficulties for massive data in high-dimensional spaces.

An alternative and simple estimation algorithm which can be seen as a stochastic gradient algorithm (Robbins and Monro, 1951; Ruppert, 1985; Duflo, 1997; Cardot et al., 2013) and is defined as follows

$$m_{j+1} = m_j + \gamma_j \frac{X_{j+1} - m_j}{\|X_{j+1} - m_j\|}$$

where  $m_0$  is an arbitrarily chosen starting point and  $\gamma_j$  is a step size such that  $\forall j \geq 1, \gamma_j > 0, \sum_{j \geq 1} \gamma_j = \infty$  and  $\sum_{j \geq 1} \gamma_j^2 < \infty$ . Its averaged version (ASG), which is effective for large samples of high dimension data, introduced by Polyak and Juditsky (1992) and adapted by Cardot et al. (2013), is defined by

$$\bar{m}_{j+1} = \bar{m}_j + \frac{1}{j+1} (m_{j+1} - \bar{m}_j).$$

One can speak about averaging since  $\bar{m}_j = \frac{1}{j} \sum_{i=1}^j m_i$ . We note that, under suitable assumptions, both  $\hat{m}_t$  and  $\bar{m}_t$  are asymptotically efficient (Vardi and Zhang, 2000; Cardot et al., 2013).

## 2.2 K-medians

For a positive integer  $k$ , a vector quantizer  $Q$  of dimension  $d$  and codebook size  $k$  is a (measurable) mapping of the  $d$ -dimensional Euclidean  $\mathbb{R}^d$  into a finite set of points  $\{c_1, \dots, c_k\}$  (Linder, 2000). More precisely, the points  $c_i \in \mathbb{R}^d, i = 1, \dots, k$  are called the codepoints and the vector composed of the code points  $\{c_1, \dots, c_k\}$  is called codebook, denoted by  $c$ . Given a  $d$ -dimensional random vector  $X$  admitting a finite first order moment, the  $L^1$ -distortion of a vector quantizer  $Q$  with codebook  $c = \{c_1, \dots, c_k\}$  is defined by

$$W(c) := \mathbb{E} \left[ \min_{j=1, \dots, k} \|X - c_j\| \right]. \quad (2)$$

Let us now consider  $X_1, \dots, X_n$  random vectors  $\in \mathbb{R}^d$  i.i.d with the same law as  $X$ . Then, one can define the empirical  $L^1$ -distortion as :

$$W_n(c) := \frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, k} \|X_i - c_j\|. \quad (3)$$

In this paper, we consider two types of K-medians algorithms : sequential and non sequential algorithm. The non sequential algorithm uses Lloyd-style iteration which alternates between an expectation (E) and maximization (M) step and is precisely described in Algorithm 1:

**Inputs :**  $D = \{x_1, \dots, x_n\}$  datapoints,  $k$  number of clusters

**Output:** A set of  $k$  clusters :  $C_1, \dots, C_k$

Randomly choose  $k$  centroids :  $m_1, \dots, m_k$ .

```

while the clusters change do
  for  $1 \leq i \leq n$  do
     $r = \arg \min_{1 \leq j \leq k} \|x_i - m_j\|$ 
     $C_r \leftarrow x_i$ 
  end
  for  $1 \leq j \leq k$  do
     $m_j = \arg \min_m \sum_{i, x_i \in C_j} \|x_i - m\|$ 
  end
end

```

**Algorithm 1:** Non Sequential K-medians Algorithm .

For  $1 \leq j \leq k$ ,  $m_j$  is nothing but the geometric median of the points in the cluster  $C_j$ . As  $m_j$  is not explicit, we will use Weiszfeld (indicated by ‘‘Offline’’) or ASG (indicated by ‘‘Semi-Online’’) to estimate it. The Online K-median algorithm proposed by [Cardot et al. \(2012\)](#) based on an averaged Robbins-Monro procedure ([Robbins and Monro, 1951](#); [Polyak and Juditsky, 1992](#)) is described in [Algorithm 2](#):

**Inputs :**  $D = \{x_1, \dots, x_n\}$  datapoints,  $k$  number of clusters,  $c_\gamma > 0$  and  $\alpha \in (1/2, 1)$

**Output:** A set of  $k$  clusters :  $C_1, \dots, C_k$

Randomly choose  $k$  centroids :  $m_1, \dots, m_k$ .

$\bar{m}_j = m_j \forall 1 \leq j \leq k$

$n_j = 1 \forall 1 \leq j \leq k$

```

for  $1 \leq i \leq n$  do
   $r = \arg \min_{1 \leq j \leq k} \|x_i - \bar{m}_j\|$ 
   $C_r \leftarrow x_i$ 
   $m_r \leftarrow m_r + \frac{c_\gamma}{(n_r + 1)^\alpha} \frac{x_i - m_r}{\|x_i - m_r\|}$ 
   $\bar{m}_r \leftarrow \frac{n_r \bar{m}_r + m_r}{n_r + 1}$ 
   $n_r \leftarrow n_r + 1$ 
end

```

**Algorithm 2:** Online K-medians Algorithm .

The non-sequential algorithms are effective but the computational time is huge compared to the sequential (‘‘Online’’) algorithm, which is very fast and only requires  $O(knd)$  operations, where  $n$  is the sample size,  $k$  is the number of clusters and  $d$  is dimension. Furthermore, in case of large samples, Online algorithm is expected to estimate the centers of the clusters as well as the non-sequential algorithm [Cardot et al. \(2012\)](#). Then, in case of large sample size, Online algorithm should be preferred and vice versa.

### 3 The choice of $k$

In this section, we adapt the results that have been shown for K-means in [Fischer \(2011\)](#) to K-medians clustering. In this aim, let  $X_1, \dots, X_n$  random vectors with the same law as  $X$ , and we assume that  $\|X\| \leq R$  almost surely for some  $R > 0$ . Let  $S_k$  denote the countable set of all  $\{c_1, \dots, c_k\} \in \mathbf{Q}^k$ , where  $\mathbf{Q}$  is some grid over  $\mathbb{R}^d$ . It is important to note that  $\mathbf{Q}$  represents the search space for the centers. Since  $\|X\|$  is assumed to be bounded by  $R$ , we consider a grid  $\mathbf{Q} \subset \bar{\mathcal{B}}(0, R)$  (where  $\bar{\mathcal{B}}(0, R)$  denotes the closed ball centered at 0 with radius  $R$ ). A codebook  $\hat{c}_k$  is said empirically optimal codebook if we have  $W_n(\hat{c}_k) = \min_{c \in S_k} W_n(c)$ . Let  $\hat{c}_k$  be a minimizer of the criterion  $W_n(c)$  over  $S_k$ . Our aim is to determine  $\hat{k}$  minimizing a criterion of the type

$$\text{crit}(k) = W_n(\hat{c}_k) + \text{pen}(k)$$

where  $\text{pen} : \{1, \dots, n\} \rightarrow \mathbb{R}_+$  is a penalty function described later. The purpose of this penalty method is to prevent choosing too large a value for  $k$  by introducing a penalty into the objective function.

In this section, we will give an upper bound for the expectation of the distortion at empirically optimal codebook with size of optimal number of clusters which is based on a general non asymptotic upper bound for

$$\mathbb{E} \left[ \sup_{c \in S_k} \{W(c) - W_n(c)\} \right].$$

**Theorem 3.1.** *Let  $X_1, \dots, X_n$  be random vectors taking values in  $\mathbb{R}^d$  with the same law as  $X$ , and we assume that  $\|X\| \leq R$  almost surely for some  $R > 0$ . Define  $W$  and  $W_n$  as in (2) and (3), respectively. Then for all  $1 \leq k \leq n$ ,*

$$\mathbb{E} \left[ \sup_{c \in S_k} \{W(c) - W_n(c)\} \right] \leq 48R \sqrt{\frac{kd}{n}}.$$

This theorem shows that the maximum difference of the distortion and the expected empirical distortion of any vector quantizer is of order  $n^{-1/2}$ . Selecting the search space for the centers is crucial because a larger search space results in a higher upper bound.

**Theorem 3.2.** *Let  $X$  be a random vector taking values in  $\mathbb{R}^d$  and we assume that  $\|X\| \leq R$  almost surely for some  $R > 0$ . Consider nonnegative weights  $\{x_k\}_{1 \leq k \leq n}$  such that  $\sum_{k=1}^n e^{-x_k} = \Sigma$ . Define  $W$  as in (2) and suppose that for all  $1 \leq k \leq n$*

$$\text{pen}(k) \geq R \left( 48 \sqrt{\frac{kd}{n}} + 2 \sqrt{\frac{x_k}{2n}} \right).$$

Then:

$$\mathbb{E} [W(\tilde{c})] \leq \inf_{1 \leq k \leq n} \left\{ \inf_{c \in S_k} W(c) + \text{pen}(k) \right\} + \Sigma R \sqrt{\frac{\pi}{2n}},$$

where  $\tilde{c} = \hat{c}_{\tilde{k}}$  minimizer of the penalized criterion.

We remark the presence of the weights  $\{x_k\}_{1 \leq k \leq n}$  in penalty function and  $\Sigma$  which depends on the weights in upper bound for the expectation of the distortion at  $\tilde{c}$ . The larger the weights  $\{x_k\}_{1 \leq k \leq n}$ , the smaller the value of  $\Sigma$ . So, we have to make a compromise between these two terms. Let us indeed consider the simple situation where one can take  $\{x_k\}_{1 \leq k \leq n}$  such that  $x_k = Lk$  for some positive constant  $L$  and  $\Sigma = \sum_{k=1}^n e^{-x_k} \leq 1$ . If we take

$$\text{pen}(k) = R \left( 48 \sqrt{\frac{kd}{n}} + 2 \sqrt{\frac{Lk}{2n}} \right) = R \sqrt{\frac{k}{n}} \left( 48 \sqrt{d} + 2 \sqrt{\frac{L}{2}} \right),$$

we deduce that the penalty shape is  $a \sqrt{\frac{k}{n}}$  where  $a$  is a constant.

**Proposition 3.1.** *Let  $X$  be a  $d$ -dimensional random vector such that  $\|X\| \leq R$  almost surely. Then for all  $1 \leq k \leq n$ ,*

$$\inf_{c \in S_k} W(c) \leq 4Rk^{-1/d},$$

where  $W$  is defined in (2).

Assume that for every  $1 \leq k \leq n$

$$\text{pen}(k) = aR \sqrt{\frac{k}{n}},$$

where  $a$  is a positive constant that satisfies  $a \geq \left(48\sqrt{d} + 2\sqrt{\frac{L}{2}}\right)$  to verify the hypothesis of Theorem 3.2. Using Theorem 3.2 and Proposition 3.1, we obtain:

$$\mathbb{E} [W(\tilde{c})] \leq R \left( \inf_{1 \leq k \leq n} \left\{ 4k^{-1/d} + a\sqrt{\frac{k}{n}} \right\} + \Sigma\sqrt{\frac{\pi}{2n}} \right).$$

Minimizing the term on the right hand side of previous inequality leads to  $k$  of the order  $n^{\frac{d}{d+2}}$  and

$$\mathbb{E} [W(\tilde{c})] = \mathcal{O}(n^{-\frac{1}{d+2}}).$$

We conclude that our penalty shape is  $a\sqrt{\frac{k}{n}}$  where  $a$  is a constant. In Birgé and Massart (2007), a data-driven method has been introduced to calibrate such criteria whose penalties are known up to a multiplicative factor: the “slope heuristics”. This method consists of estimating the constant of penalty function by the slope of the expected linear relation of  $-W_n(\hat{c}_k)$  with respect to the penalty shape values  $\text{pen}_{\text{shape}}(k) = \sqrt{\frac{k}{n}}$ .

**Estimation of constant  $a$ :** Let denote  $c^* = \arg \min_{c \in S} W(c)$  and  $c_k = \arg \min_{c \in S_k} W(c)$ , where  $S$  any linear subspace of  $\mathbb{R}^d$  and  $S_k$  set of predictors (called a model). It was shown in Birgé and Massart (2007); Arlot and Massart (2009); Baudry et al. (2012) that under conditions, the optimal penalty verifies for large  $n$ :

$$\text{pen}_{\text{opt}}(k) := a_{\text{opt}} \text{pen}_{\text{shape}}(k) \approx 2(W_n(c^*) - W_n(\hat{c}_k)).$$

This gives

$$\frac{a_{\text{opt}}}{2} \text{pen}_{\text{shape}}(k) - W_n(c^*) \approx -W_n(\hat{c}_k).$$

The term  $-W_n(\hat{c}_k)$  with respect to the penalty shape behaves like a linear function for a large  $k$ . The slope  $\hat{S}$  of the linear regression of  $-W_n(\hat{c}_k)$  with respect to  $\text{pen}_{\text{shape}}(k)$  is computed to estimate  $\frac{a_{\text{opt}}}{2}$ . Finally, we obtain

$$\text{pen}(k) := a_{\text{opt}} \text{pen}_{\text{shape}}(k) = 2\hat{S} \text{pen}_{\text{shape}}(k).$$

Of course, since this method is based on asymptotic results, it can encounter some practical problems when the dimension  $d$  is larger than the sample size  $n$ .

## 4 Simulations

This whole method is implemented in **R** and all these studied algorithms are available in the **R** package **Kmedians** <https://cran.r-project.org/package=Kmedians>. In what follows, the centers initialization are generated from robust hierarchical clustering algorithm with **genieclust** package (Gagolewski et al., 2016).

### 4.1 Visualization of results with the package **Kmedians**

In Section 3, we proved that the penalty shape is  $a\sqrt{\frac{k}{n}}$  where  $a$  is a constant to calibrate. To find the constant  $a$ , we will use the data-based calibration algorithm for penalization procedures that is explained at the end of section 3. This data-driven slope estimation method is implemented in CAPUSHE (CALibrating Penalty Using Slope HEuristics) (Brault et al., 2011) which is available in the **R** package **capushe** <https://cran.r-project.org/package=capushe>. This proposed slope estimation method is made to be robust in order to preserve the eventual undesirable variations of criteria. More precisely, for a certain number of clusters  $k$ , the algorithm may be trapped by a local minima, which could create a “bad point” for the slope heuristic. The slope heuristic has therefore been designed to be robust to the presence of such points.

In what follows, we consider a random variable  $X$  following a Gaussian Mixture Model with  $k = 6$  classes where the mixture density function is defined as

$$p(x) = \sum_{j=1}^k \pi_j \mathcal{N}(x|\mu_j, \mathbb{I}_d)$$

with,  $\pi_j = \frac{1}{k} \quad \forall 1 \leq j \leq k$ ,  $\mu_j \sim \mathcal{U}_{10}$  where  $\mathcal{U}_{10}$  is the uniform law on the sphere of radius 10 and,

$$\mathcal{N}(x|\mu, \mathbb{I}_d) = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2}\|x - \mu\|^2\right).$$

In what follows, we consider  $n = 3000$  i.i.d realizations of  $X$  and  $d = 5$ . We first focus on some visualization of our slope method.

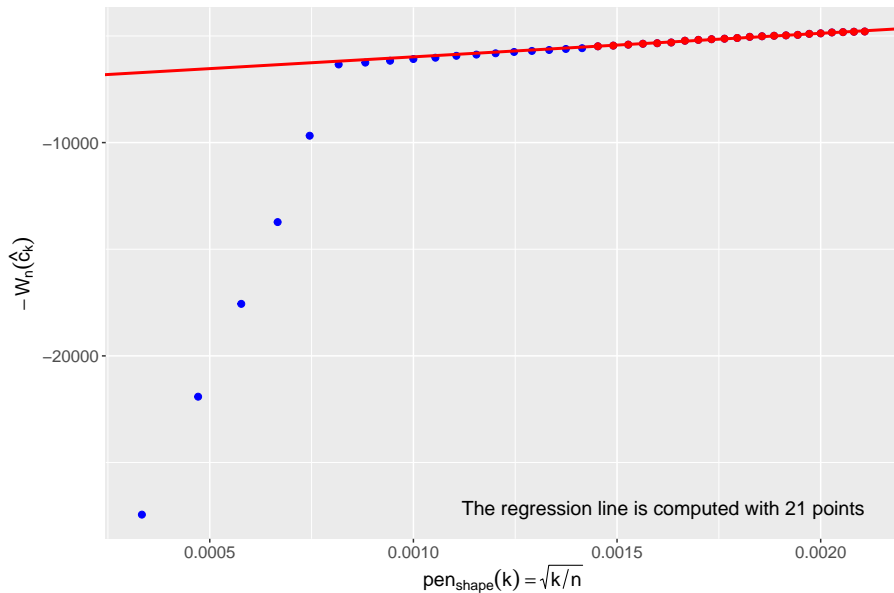


Figure 1: Evolution of  $-W_n(\hat{c}_k)$  with respect to penalty shape:  $\sqrt{k/n}$ .

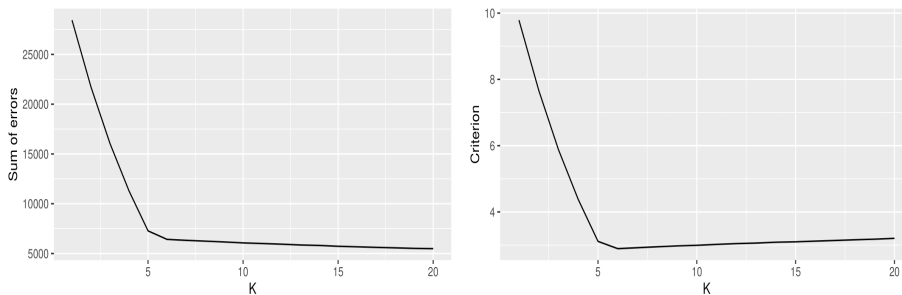


Figure 2: Evolution of  $W_n(\hat{c}_k)$  (on the left) and  $\text{crit}(k)$  (on the right) with respect to  $k$ .

To estimate  $a \approx 2\hat{S}$  in the penalty function, it is sufficient to estimate  $\hat{S}$ , which is the slope of the red curve in Figure 1. As shown in Figure 1, the regression slope is estimated using the last 21 points, as it behaves like an affine function when  $k$  is large. In Figure 2 (left), two possible elbows are observed in the curve. Consequently, the elbow method suggests considering either 5 or 6 as the number of clusters.



We would prefer to choose 5 since the elbow point at 5 is more pronounced compared to the one at 6. Therefore, this method is not ideal in this case.

Figures 3 to 5 represent the data as curves, which we call “profiles” (the x-label corresponds to the coordinates, and the y-label to the values of the coordinates), gathered by cluster and with the centers of the groups represented in red. We also show the first two principal components of the data using robust principal component analysis components (RPCA) (Cardot and Godichon-Baggioni, 2017). In Figure 3, we focus on the clustering obtained with the K-medians algorithm (“Offline” version) for non contaminated data. In each cluster, the curves are close to each other and also close to the median, and the profiles differ from one cluster to another, meaning that our method separated well the 6 groups. In order to visualize the robustness of the proposed method, we considered contaminated data with the law  $Z = (Z_1, \dots, Z_5)$  where  $Z_i$  are i.i.d, with  $Z_i \sim \mathcal{T}_1$  where  $\mathcal{T}_1$  is a Student law with one degree of freedom. Applying our method for selecting the number of clusters for K-medians algorithms, we selected the corrected number of clusters. Furthermore, the obtained groups, despite the presence of some outliers in each cluster, are coherent. Nevertheless, in the case of K-means clustering, the method found non homogeneous clusters, i.e. the method assimilates some far outliers as single clusters (see Figure 5). It’s important to note that, in the case of contaminated data (Figures 4 and 5), we only represented 95% of the data to better visualize them. Then, in Figure, 5, Clusters 5, 7, 8, 11 and 12 are not visible since they are “far” outliers.

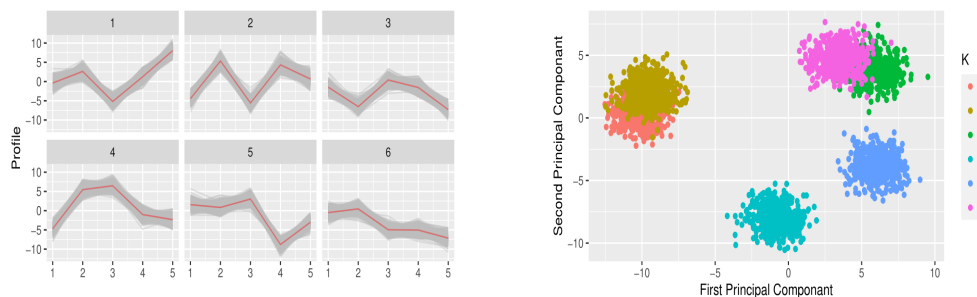


Figure 3: Profiles (on the left) and clustering via K-medians represented on the first two principal components (on the right) without contaminated data.

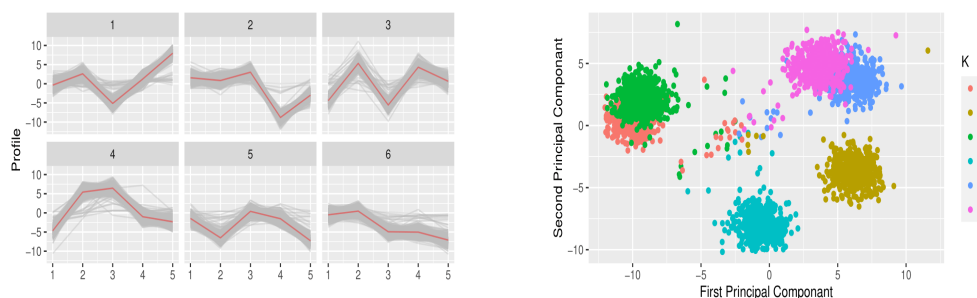


Figure 4: Profiles (on the left) and clustering via K-medians algorithm represented on the first two principal components (on the right) with 5% of contaminated data.

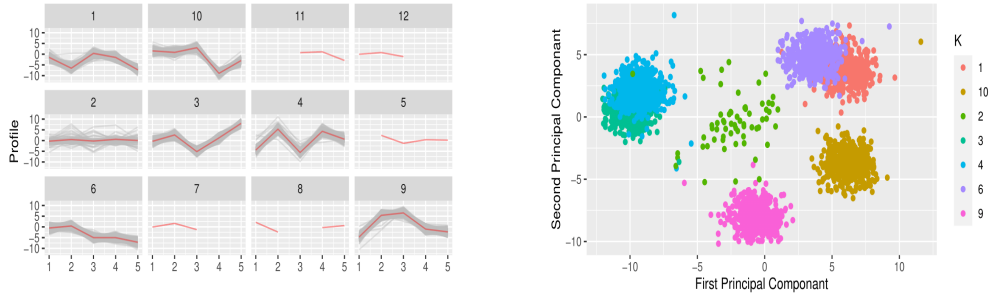


Figure 5: Profiles (on the left) and clustering via K-means algorithm represented on the first two principal components (on the right) with 5% of contaminated data.

## 4.2 Comparison with Gap Statistic and Silhouette

In what follows, we focus on the choice of the number of clusters and compare our results with different methods. For this, we generated some basic data sets in three different scenarios (see Fischer (2011)) :

**(S1) 4 clusters in dimension 3 :** The data are generated by Gaussian mixture centered at  $(0, 0, 0)$ ,  $(0, 2, 3)$ ,  $(3, 0, -1)$ , and  $(-3, -1, 0)$  with variance equal to the identity matrix. Each cluster contains 500 data points.

**(S2) 5 clusters in dimension 4 :** The data are generated by Gaussian mixture centered at  $(0, 0, 0, 0)$ ,  $(3, 5, -1, 0)$ ,  $(-5, 0, 0, 0)$ ,  $(1, 1, 6, -2)$  and  $(1, -3, -2, 5)$  with variance equal to the identity matrix. Each cluster contains 500 data points.

**(S3) 3 clusters in dimension 2 :** The data are generated by a Student mixture centered at  $(0, 0)$ ,  $(0, 6)$  and  $(5, 3)$  with 2 degree of freedom. Each cluster contains 500 data points.

We applied three different methods for determining the number of clusters : the proposed slope method, Gap Statistic and Silhouette method. For each method, we use four clustering algorithms : K-medians (“Online”, “Semi-Online”, “Offline”) and K-means. For each scenario, we contaminated our data with the law  $Z = (Z_1, \dots, Z_d)$  where  $Z_i$  are i.i.d, with  $Z_i \sim \mathcal{T}_1$  where  $\mathcal{T}_1$  is a Student law with 1 degree of freedom. Then, we evaluate our method for the different methods and scenarios by considering:

- $N$  : The number of times we get the correct value of cluster in 50 repeated trials without contaminated data.
- $\bar{k}$  : The average of number of clusters obtained over 50 trials without contaminated data.
- $N_{0.1}$  : The number of times we get the correct value of cluster in 50 repeated trials with 10% of contaminated data.
- $\bar{k}_{0.1}$  : The average of number of clusters obtained over 50 trials with 10% of contaminated data.

In case of well separated clusters as in the scenario (S2), the gap statistics method and silhouette method give competitive results. Nevertheless, for closer clusters, the slope method works much better than gap statistics and silhouette method as in the scenario (S1). The gap statistics method only works in scenario 2 and is ineffective in the presence of contamination. In closer cluster scenarios, it often predicts 1 as the number of clusters. The silhouette method performs moderately well in scenario 2 and very well in scenario 3, but it is globally not as competitive as the slope method, especially in cases of contaminated data. In scenarios 1 and 2 with slope method, Offline, Semi-Online, Online and K-means give better results but in cases of contamination, K-means crashes completely while the other three methods seem to be not too much sensitive. Furthermore, on non-Gaussian data (scenario 3), the K-means method does not work at all. In such cases, K-median clustering is often preferred over K-means clustering.

Overall, in every scenario, Offline, Semi-Online, Online K-medians with the slope method give very competitive results and in the case where the data are contaminated, they clearly outperform other methods, especially the Offline method.

Simulations		S1				S2				S3			
	Algorithms	$N$	$\bar{k}$	$N_{0.1}$	$\bar{k}_{0.1}$	$N$	$\bar{k}$	$N_{0.1}$	$\bar{k}_{0.1}$	$N$	$\bar{k}$	$N_{0.1}$	$\bar{k}_{0.1}$
Slope	Offline	<b>50</b>	<b>4</b>	<b>50</b>	<b>4</b>	<b>50</b>	<b>5</b>	<b>50</b>	<b>5</b>	<b>50</b>	<b>3</b>	49	3.04
	Semi-Online	50	4	49	4.02	<b>50</b>	<b>5</b>	46	5.1	<b>50</b>	<b>3</b>	49	3.04
	Online	48	4	42	4.1	<b>50</b>	<b>5</b>	40	5.2	<b>50</b>	<b>3</b>	49	3.04
	K-means	<b>50</b>	<b>4</b>	1	7.9	<b>50</b>	<b>5</b>	2	6.7	3	5.3	0	7.2
Gap	Offline	6	1.7	0	1	47	4.8	2	1.2	0	1	0	1
	Semi-Online	7	1.7	0	1	47	4.8	2	1.2	0	1	0	1
	Online	8	2.4	0	1	47	4.8	2	1.2	0	1	0	1
	K-means	0	1.2	0	1.2	12	2	0	1.3	0	1	0	1
Silhouette	Offline	0	3	0	2.9	27	4.4	1	3.5	<b>50</b>	<b>3</b>	44	3.1
	Semi-Online	0	3	0	2.9	24	4.4	1	3.5	<b>50</b>	<b>3</b>	43	3.1
	Online	0	3	2	3.2	22	4.5	2	4.5	49	3.02	29	3.5
	K-means	0	3	7	3.2	20	4.5	0	6.7	3	5.9	2	7.2

Table 1: Comparison of the number of times we get the right value of clusters and the averaged selected number of clusters obtained with the different methods without contaminated data and with 10% of contaminated data.

### 4.3 Contaminated Data in Higher Dimensions

We now focus on the impact of contaminated data on the selection of the number of clusters in K-medians clustering, particularly in higher dimensions. We compare our method with Gap Statistic and SigClust (Liu et al., 2008; Huang et al., 2015; Bello et al., 2023) in the Offline setting, as it yields competitive results, as noted in the previous section. Concerning SigClust, it is a method which enables to test whether a sample comes from a single Gaussian or several in high dimension. Then, starting from  $k = k_0$ , we test for all possible pairs of clusters whether the fusion of the two clusters comes from a single Gaussian or not. If the test rejected the hypothesis that the combined cluster is a single Gaussian for all fittings, the same procedure is repeated for  $k + 1$ . If there is a fitting for which the test is not rejected, it is considered that these two clusters should be merged, and the procedure is stopped. The optimal number of clusters is then determined as  $k_{\text{opt}} = k - 1$ . It is important to note that we did not compare with Gap Statistics, as it is computationally expensive, especially in high dimensions.

In this aim, we generate data using a Gaussian mixture model with 10 classes in 100 and 200 dimensions, where the centers of the classes are randomly generated on a sphere with radius 10, and each class contains 100 data points. The data is contaminated with the law  $Z = (Z_1, \dots, Z_d)$ , where  $d$  is the dimension (100 or 200 for each scenario),  $Z_i$  are i.i.d, with two possible scenarios:

1.  $Z_i \sim \mathcal{T}_1$ ,
2.  $Z_i \sim \mathcal{T}_2$ .

Here,  $\mathcal{T}_m$  is the Student law with  $m$  degrees of freedom. In what follows, let us denote by  $\rho$  the proportion of contaminated data. In order to compare the different clustering results, we focus on the Adjusted Rand Index (ARI) (Rand, 1971; Hubert and Arabie, 1985) which is a measure of similarity between two clusterings and which relies on taking into account the right number of correctly classified pairs. We evaluate, for each scenario, the average number of clusters obtained over 50 trials and the average ARI evaluated only on uncontaminated data.

We observe that in the case of non-contamination, we obtain similar results across all methods. However, in the presence of contamination, our method consistently performs well, while others struggle to identify an appropriate number of clusters. With a Student distribution contamination of 1 degree of freedom, our method excels in terms of both the number of clusters and the ARI. The results with a Student distribution contamination of 2 degrees of freedom are comparable to those obtained using the Silhouette method.

		$\rho$	0	0.01	0.02	0.03	0.05	0.1
$d = 100$	$Z_i \sim \mathcal{T}_1$	Our Method	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10.2</b>	<b>7.6</b>
		Silhouette	<b>10</b>	8.7	5.9	4.2	3.6	2.6
		SigClust	<b>10</b>	2.7	2.9	3.1	4.3	5.2
	$Z_i \sim \mathcal{T}_1$	Our Method	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.94</b>	<b>0.91</b>	<b>0.53</b>
		Silhouette	<b>1</b>	0.75	0.51	0.29	0.18	0.15
		SigClust	<b>1</b>	0.22	0.28	0.29	0.31	0.46
$d = 100$	$Z_i \sim \mathcal{T}_2$	Our Method	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	10.9
		Silhouette	<b>10</b>	<b>10</b>	<b>10</b>	10.1	9.8	<b>10.4</b>
		SigClust	<b>10</b>	5.3	4.5	4.2	4.4	4.2
	$Z_i \sim \mathcal{T}_2$	Our Method	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.99</b>	<b>0.99</b>
		Silhouette	<b>1</b>	<b>1</b>	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	0.98
		SigClust	<b>1</b>	0.52	0.36	0.35	0.34	0.32
$d = 200$	$Z_i \sim \mathcal{T}_1$	Our Method	<b>10</b>	<b>10</b>	<b>10</b>	<b>9.6</b>	<b>9.6</b>	<b>7.3</b>
		Silhouette	<b>10</b>	6	2.9	2.9	2.5	3.1
		SigClust	9.2	2.7	3.4	4.3	5	6.5
	$Z_i \sim \mathcal{T}_1$	Our Method	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.89</b>	<b>0.76</b>	<b>0.53</b>
		Silhouette	<b>1</b>	0.39	0.18	0.17	0.16	0.21
		SigClust	0.97	0.22	0.28	0.34	0.42	0.48
$d = 200$	$Z_i \sim \mathcal{T}_2$	Our Method	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>
		Silhouette	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	10.1	10.2
		SigClust	9.2	4.6	4.3	3.9	3.8	2.7
	$Z_i \sim \mathcal{T}_2$	Our Method	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
		Silhouette	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	0.99	0.99
		SigClust	0.97	0.37	0.35	0.32	0.32	0.22

Table 2: Comparison of the selected number of clusters and the averaged ARI obtained using different methods with respect to the proportion of contaminated data for  $Z_i \sim \mathcal{T}_1$  and  $Z_i \sim \mathcal{T}_2$ .

In summary, our method demonstrates remarkable robustness in the face of contaminated data, making it a strong choice for clustering in higher dimensions. The comparison with the Silhouette, Gap Statistic, and SigClust in the offline setting reaffirms the effectiveness of our approach, especially when computational efficiency is a critical factor in high-dimensional data.

#### 4.4 An illustration on real data

We will first briefly discuss the data we used for clustering, which was provided by Califrais, a company specializing in developing environmentally responsible technology to optimize logistics flows on a large scale. Our goal is to build a Recommender System that is designed to suggest items individually for each user based on their historical data or preferences. In this scenario, the clustering algorithms can be employed to identify groups of similar customers where each group consists of customers share similar features or properties. It is crucial to perform robust clustering in order to develop an effective Recommender System.

The dataset includes information on 508 customers, including nine features that represent the total number of products purchased in each of the following categories: Fruits, Vegetables, Dairy products, Seafood, Butcher, Deli, Catering, Grocery, and Accessories and equipment. Therefore, we have a sample size of  $n = 508$  and a dimensionality of  $d = 9$ . To apply clustering, we will determine the appropriate number of clusters using the proposed method. Before applying our method, we normalize our data using RobustScaler. This removes the median and scales the data according to the Interquartile Range, which is the range between the 1st quartile and the 3rd quartile.

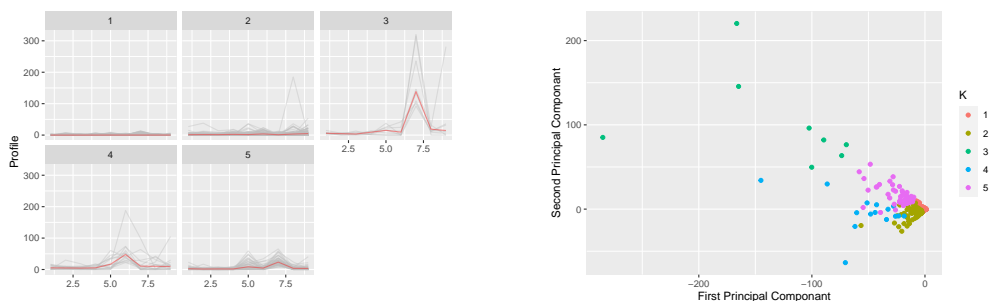


Figure 6: Califrais data: Profiles (on the left) and clustering with Slope method represented on the first two principal components (on the right).

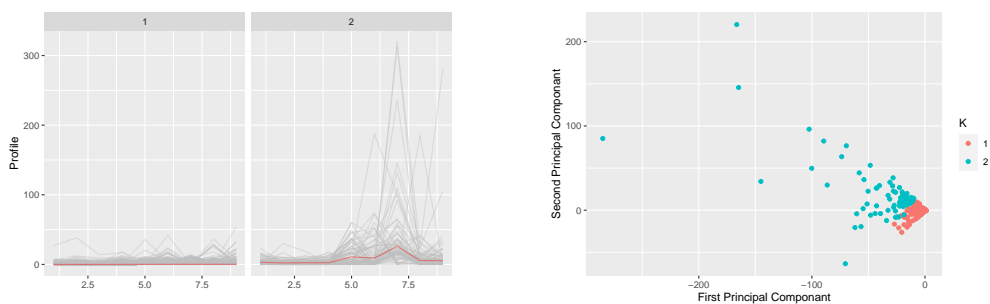


Figure 7: Califrais data: Profiles (on the left) and clustering with Silhouette method represented on the first two principal components (on the right).

We plotted the profiles of the clusters obtained using our Slope method, Silhouette and Gap Statistic in Figures 6, 7, and 8. We observe that our method indicates 5 clusters, while the Gap Statistic suggests 3 clusters, and Silhouette suggests 2 clusters. Regarding the Silhouette method, the second cluster

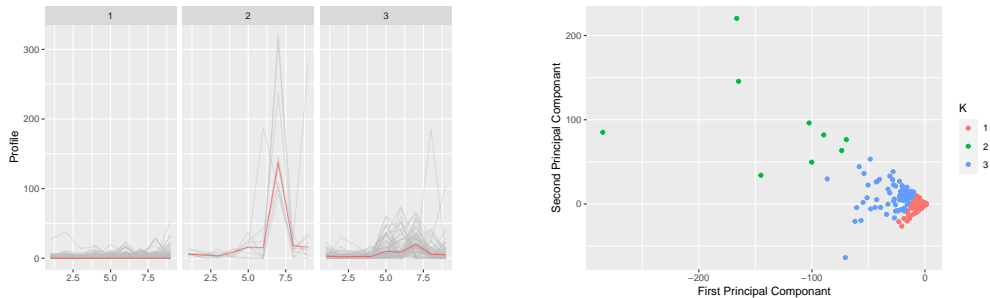


Figure 8: Califrains data: Profiles (on the left) and clustering with Gap Statistics method represented on the first two principal components (on the right).

obtained is not homogeneous, as seen in Figure 7. We obtain 3 clusters with the Gap Statistic method. The important thing to note is that the Gap Statistic method separates the second cluster obtained by Silhouette into two clusters (Cluster 2 and Cluster 3). However, in the third cluster of Gap Statistic (Figure 8), homogeneity is still not achieved. In Figure 6, it can be seen that the clusters generated by our slope method are more or less homogeneous. To establish a connection with the simulations conducted in Section 4.2, for example, in scenario (S1), we observed that Silhouette and Gap Statistics failed to find the correct number of clusters when the clusters are closer. This is reflected here, as the behavior of clients does not change significantly, resulting in close clusters. To provide an overview of our clusters, the first cluster represents customers who regularly consume products from all categories. The third cluster consists of customers who frequently engage with catering products. Clusters 2, 4, and 5 correspond to customers who consume significant amounts of Butcher, Deli, and Catering products at different levels, as depicted in the figure 6.

## 4.5 Conclusion

The proposed penalized criterion, calibrated with the help of the slope heuristic method, consistently gives competitive results for selecting the number of clusters in K-medians, even in the presence of outliers, outperforming other methods such as Gap Statistics, Silhouette, and SigClust. Notably, our method demonstrates excellent performance even in high dimensions. Among the three K-medians algorithms, Offline, Semi-Online, and Online, their performances are generally analogous, with Offline being slightly better. However, for large sample sizes, one may prefer the Online K-medians algorithm in terms of computation time. As discussed in Section 2, it is recommended to use the Offline algorithm for moderate sample sizes, the Semi-Online algorithm for medium sample sizes, and the Online algorithm for large sample sizes. In our real-life data illustration, our proposed method consistently produces more robust clusters and a more suitable number of clusters compared to other methods.

In conclusion, our paper presents a robust and efficient approach for selecting clusters in K-medians, demonstrating superior performance even in challenging scenarios. The findings provide practical recommendations for algorithm selection based on sample size, reinforcing the applicability of our proposed method in real-world clustering scenarios.

## Acknowledgement

The authors wish to thank Califrains for providing the real-life data and Raphaël Carpintero Perez for the data preprocessing work.

## 5 Proofs

### 5.1 Some definitions and lemma

First, we provide some definitions and lemmas that are useful to prove Theorems 3.1 and 3.2.

**Definitions :**

- Let  $(S, p)$  be a totally bounded metric space. For any  $F \subset S$  and  $\epsilon > 0$  the  $\epsilon$ -covering number  $N_p(F, \epsilon)$  of  $F$  is defined as the minimum number of closed balls with radius  $\epsilon$  whose union covers  $F$ .
- Let  $(S, p)$  be a totally bounded metric space. For any  $F \subset S$ ,  $\text{diam}(F) = \sup \{p(x, y) : x, y \in F\}$ .
- A Family  $\{T_s : s \in S\}$  of zero-mean random variables indexed by the metric space  $(S, p)$  is called subgaussian in the metric  $p$  if for any  $\lambda > 0$  and  $s, s' \in S$  we have

$$\mathbb{E} \left[ e^{\lambda(T_s - T_{s'})} \right] \leq e^{\frac{\lambda^2 p(s, s')^2}{2}}.$$

- The Family  $\{T_s : s \in S\}$  is called sample continuous if for any sequence  $s_1, s_2, \dots \in S$  such that  $s_j \rightarrow s \in S$  we have  $T_{s_j} \rightarrow T_s$  with probability one.

**Lemma 5.1 (Hoeffding (1994)).** *Let  $Y_1, \dots, Y_n$  are independent zero-mean random variables such that  $a \leq Y_i \leq b, i = 1, \dots, n$ , then for all  $\lambda > 0$ ,*

$$\mathbb{E} \left[ e^{\lambda(\sum_{i=1}^n Y_i)} \right] \leq e^{\frac{\lambda^2 n(b-a)^2}{8}}.$$

**Lemma 5.2 (Cesa-Bianchi and Lugosi (1999), Proposition 3).** *If  $\{T_s : s \in S\}$  is subgaussian and sample continuous in the metric  $p$ , then*

$$\mathbb{E} \left[ \sup_{s \in S} T_s \right] \leq 12 \int_0^{\text{diam}(S)/2} \sqrt{\ln N_p(S, \epsilon)} d\epsilon.$$

**Lemma 5.3 (Bartlett et al. (1998), Lemma 1).** *Let  $S(0, R)$  denote the closed  $d$ -dimensional sphere of radius  $R$  centered at 0. Let  $\epsilon > 0$  and  $N(\epsilon)$  denote the cardinality of the minimum  $\epsilon$  covering of  $S(0, R)$ , that is,  $N(\epsilon)$  is the smallest integer  $N$  such that there exist points  $\{y_1, \dots, y_N\} \subset S(0, R)$  with the property*

$$\sup_{x \in S(0, R)} \min_{1 \leq i \leq N} \|x - y_i\| \leq \epsilon.$$

*Then, for all  $\epsilon \leq 2R$  we have*

$$N(\epsilon) \leq \left( \frac{4R}{\epsilon} \right)^d.$$

**Lemma 5.4.** *For any  $0 < \epsilon \leq 2R$  and  $k \geq 1$ , the covering number of  $S_k$  in the metric*

$$p(c, c') = \sup_{\|x\| \leq R} \left\{ \left| \min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \right| \right\}$$

*is bounded as*

$$N_p(S_k, \epsilon) \leq \left( \frac{4R}{\epsilon} \right)^{kd}.$$

**Proof of the Lemma 5.4 :** . Let  $0 < \epsilon \leq 2R$ , by Lemma 5.3 there exists a  $\epsilon$ -covering set of points  $\{y_1, \dots, y_N\} \subset S(0, R)$  with  $N \leq \left(\frac{4R}{\epsilon}\right)^d$ . Since, we have  $N^k$  ways to choose  $k$  codepoints from a set of  $N$  points  $\{y_1, \dots, y_N\}$ , that implies

$$N_p(S_k, \epsilon) \leq \left(\frac{4R}{\epsilon}\right)^{kd}.$$

For any codepoints  $\{c_1, \dots, c_k\}$  which are contained in  $S(0, R)$ , there exists a set of codepoints such that  $\|c_j - c'_j\| \leq \epsilon$  for all  $j$ .

Let us first show

$$\min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \leq \epsilon.$$

In this aim, let us consider  $q \in \arg \min_{j=1, \dots, k} \|x - c_j\|$ , then

$$\min_{j=1, \dots, k} \|x - c'_j\| - \min_{j=1, \dots, k} \|x - c_j\| \leq \|x - c'_q\| - \|x - c_q\| \leq \|c_q - c'_q\| \leq \epsilon.$$

In the same way, considering  $q' \in \arg \min_{j=1, \dots, k} \|x - c'_j\|$ , we show

$$\min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \leq \|x - c_{q'}\| - \|x - c'_{q'}\| \leq \|c_{q'} - c'_{q'}\| \leq \epsilon.$$

So,

$$\left| \min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \right| \leq \epsilon$$

for any codepoints  $\{c_1, \dots, c_k\}$  which are contained in  $S(0, R)$ , there exists a set of codepoints  $\{c'_1, \dots, c'_k\}$  such that

$$\left| \min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \right| \leq \epsilon.$$

□

**Lemma 5.5** (McDiarmid et al. (1989), Massart (2007) : Theorem 5.3). *If  $X_1, \dots, X_n$  are independent random variables and  $\mathcal{F}$  is a finite or countable class of real-valued functions such that  $a \leq f \leq b$  for all  $f \in \mathcal{F}$ , then if  $Z = \sup_{f \in \mathcal{F}} \sum_{i=1}^n (f(X_i) - \mathbb{E}[f(X_i)])$ , we have, for every  $\epsilon > 0$ ,*

$$\mathbb{P}[Z - \mathbb{E}[Z] \geq \epsilon] \leq \exp\left(-\frac{2\epsilon^2}{n(b-a)^2}\right).$$

## 5.2 Proof of Theorem 3.1

The proof of the Theorem 3.1 is inspired by the proof of Theorem 3 in Linder (2000).

*Proof.* For any  $c \in S_k$ , let  $T_n^{(c)} = \frac{n}{2}(W(c) - W_n(c)) = \frac{1}{2} \sum_{i=1}^n (\mathbb{E}[\min_{j=1, \dots, k} \|X_i - c_j\|] - \min_{j=1, \dots, k} \|X_i - c_j\|)$ . So

$$\mathbb{E}\left[\sup_{c \in S_k} (W(c) - W_n(c))\right] = \frac{2}{n} \mathbb{E}\left[\sup_{c \in S_k} T_n^{(c)}\right].$$

Let us first demonstrate that the family of random variables  $\{T_n^{(c)} : c \in S_k\}$  is subgaussian and sample continuous in a suitable metric. For any  $c, c' \in S_k$  define

$$p(c, c') = \sup_{\|x\| \leq R} \left\{ \left| \min_{j=1, \dots, k} \|x - c_j\| - \min_{j=1, \dots, k} \|x - c'_j\| \right| \right\},$$



and  $p_n(c, c') = \sqrt{n}p(c, c')$ ,  $p_n$  is a metric on  $S_k$ . Since we have:

$$\begin{aligned} |T_n^{(c)} - T_n^{(c')}| &= \frac{n}{2} |W(c) - W(c') + W_n(c') - W_n(c)| \\ &\leq \frac{n}{2} (|W(c) - W(c')| + |W_n(c') - W_n(c)|) \\ &\leq np(c, c') = \sqrt{n}p_n(c, c'), \end{aligned}$$

the family  $\{T_n^{(c)} : c \in S_k\}$  is then sample continuous in the metric  $p_n$ . To show that  $\{T_n^{(c)} : c \in S_k\}$  is subgaussian in  $p_n$ , let

$$Y_i = \frac{1}{2} \left( W(c) - \min_{j=1, \dots, k} \|x - c_j\| \right) - \frac{1}{2} \left( W(c') - \min_{j=1, \dots, k} \|x - c'_j\| \right).$$

Then

$$T_n^{(c)} - T_n^{(c')} = \sum_{i=1}^n Y_i,$$

where  $Y_i$  are independent, have zero mean, and

$$|Y_i| \leq \frac{1}{\sqrt{n}} p_n(c, c').$$

By Lemma 5.1, we obtain

$$\mathbb{E} \left[ e^{\lambda(T_n^{(c)} - T_n^{(c')})} \right] \leq e^{\frac{\lambda^2 p_n(c, c')^2}{2}}.$$

So,  $\{T_n^{(c)} : c \in S_k\}$  is subgaussian in  $p_n$ . As the family  $\{T_n^{(c)} : c \in S_k\}$  is subgaussian and sample continuous in  $p_n$ , Lemma 5.2 gives

$$\mathbb{E} \left[ \sup_{c \in S_k} T_n^{(c)} \right] \leq 12 \int_0^{\text{diam}(S_k)/2} \sqrt{\ln N_{p_n}(S_k, \epsilon)} d\epsilon.$$

Since  $p_n(c, c') = \sqrt{n}p(c, c')$ , by Lemma 5.4 with the metric  $p_n$ , for all  $\epsilon \leq 2R\sqrt{n}$  we obtain

$$N_{p_n}(S_k, \epsilon) \leq \left( \frac{4R\sqrt{n}}{\epsilon} \right)^{kd},$$

and as  $\text{diam}(S_k) := \sup \{p_n(c, c') : c, c' \in S_k\} = \sqrt{n} \sup \{p(c, c') : c, c' \in S_k\} \leq \sqrt{n}2R$ ,

$$\begin{aligned} \mathbb{E} \left[ \sup_{c \in S_k} T_n^{(c)} \right] &\leq \frac{24}{n} \int_0^{\sqrt{n}R} \sqrt{\ln \left( \left( \frac{4R\sqrt{n}}{\epsilon} \right)^{kd} \right)} d\epsilon \\ &= \frac{24\sqrt{kd}}{n} \int_0^{\sqrt{n}R} \sqrt{\ln \left( \frac{4R\sqrt{n}}{\epsilon} \right)} d\epsilon. \end{aligned}$$

Considering  $x = \frac{\epsilon}{4R\sqrt{n}}$ , we obtain,

$$\mathbb{E} \left[ \sup_{c \in S_k} T_n^{(c)} \right] \leq \frac{24\sqrt{kd}}{n} \int_0^{\frac{1}{4}} 4R\sqrt{n} \sqrt{\ln \left( \frac{1}{x} \right)} dx.$$

Applying Jensen's inequality to the concave function  $f(x) = \sqrt{x}$  :

$$\begin{aligned} \mathbb{E} \left[ \sup_{c \in S_k} T_n^{(c)} \right] &\leq 24R \sqrt{\frac{kd}{n}} \sqrt{\int_0^{\frac{1}{4}} 4 \ln \left( \frac{1}{x} \right) dx} \\ &= 24R \sqrt{\frac{kd}{n}} \sqrt{1 + \ln 4} \\ &\leq 48R \sqrt{\frac{kd}{n}}, \end{aligned}$$

where we used that  $\int \ln x = x \ln x - x$  and  $\ln 4 \leq 3$ .

Thus,

$$\mathbb{E} \left[ \sup_{c \in S_k} \{W(c) - W_n(c)\} \right] \leq 48R \sqrt{\frac{kd}{n}}.$$

□

### 5.3 Proof of Theorem 3.2

Theorem 3.2 is an adaptation of Theorem 8.1 in Massart (2007) and Theorem 2.1 in Fischer (2011).

*Proof.* By definition of  $\tilde{c}$ , for all  $k, 1 \leq k \leq n$  and  $c_k \in S_k$ , we have:

$$\begin{aligned} W_n(\tilde{c}) + \text{pen}(\hat{k}) &\leq W_n(c_k) + \text{pen}(k) \\ W(\tilde{c}) &\leq W_n(c_k) + W(\tilde{c}) - W_n(\tilde{c}) + \text{pen}(k) - \text{pen}(\hat{k}). \end{aligned} \quad (4)$$

Consider nonnegative weights  $\{x_l\}_{1 \leq l \leq n}$  such that  $\sum_{l=1}^n e^{-x_l} = \Sigma$  and let  $z > 0$ .

Applying Lemma 5.5 with  $f(x) = \frac{1}{n} \min_{j=1, \dots, l} \|x - c_j\|$ ,  $a = 0$  and  $b = \frac{2R}{n}$  for all  $l, 1 \leq l \leq n$  and all  $\epsilon_l > 0$

$$\mathbb{P} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) - \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right] \geq \epsilon_l \right] \leq \exp \left( -\frac{n\epsilon_l^2}{2R^2} \right).$$

It follows that for all  $l$ , taking  $\epsilon_l = 2R \sqrt{\frac{x_l + z}{2n}}$

$$\mathbb{P} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \geq \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right] + 2R \sqrt{\frac{x_l + z}{2n}} \right] \leq e^{-x_l - z}.$$

Thus, we have

$$\begin{aligned} \mathbb{P} \left[ \bigcap_{l=1}^n \sup_{c \in S_l} (W(c) - W_n(c)) \leq \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right] + 2R \sqrt{\frac{x_l + z}{2n}} \right] \\ = 1 - \mathbb{P} \left[ \bigcup_{l=1}^n \sup_{c \in S_l} (W(c) - W_n(c)) \geq \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right] + 2R \sqrt{\frac{x_l + z}{2n}} \right] \geq 1 - \Sigma e^{-z}. \end{aligned}$$

Considering  $Z_l = \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right]$ , let us show if we have for all  $1 \leq l \leq n$ ,

$$\sup_{c \in S_l} (W(c) - W_n(c)) \leq Z_l + 2R \sqrt{\frac{x_l + z}{2n}}$$

then,

$$W(\tilde{c}) \leq W_n(c_k) + 2R \sqrt{\frac{z}{2n}} + \text{pen}(k).$$

We suppose that we have

$$\sup_{c \in S_l} (W(c) - W_n(c)) \leq Z_l + 2R\sqrt{\frac{x_l + z}{2n}} \quad \forall 1 \leq l \leq n. \quad (5)$$

Particularly it's true for  $l = \hat{k}$ , we have also  $W(\tilde{c}) - W_n(\tilde{c}) \leq \sup_{c \in S_{\hat{k}}} (W(c) - W_n(c))$  and  $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b} \quad \forall a, b \geq 0$ . By combining this result with (2) and (3), we get

$$\begin{aligned} W(\tilde{c}) &\leq W_n(c_k) + \sup_{c \in S_{\hat{k}}} (W(c) - W_n(c)) + \text{pen}(k) - \text{pen}(\hat{k}) \\ &\leq W_n(c_k) + Z_{\hat{k}} + 2R\sqrt{\frac{x_{\hat{k}}}{2n}} + 2R\sqrt{\frac{z}{2n}} + \text{pen}(k) - \text{pen}(\hat{k}). \end{aligned}$$

With the help of Theorem 3.2, we have  $Z_k \leq 48R\sqrt{\frac{kd}{n}}$  for all  $k, 1 \leq k \leq n$  and if we have  $\text{pen}(k) \geq R \left( 48\sqrt{\frac{kd}{n}} + 2\sqrt{\frac{x_k}{2n}} \right)$

$$\begin{aligned} W(\tilde{c}) &\leq W_n(c_k) + 48R\sqrt{\frac{\hat{k}d}{n}} + 2R\sqrt{\frac{x_{\hat{k}}}{2n}} + 2R\sqrt{\frac{z}{2n}} + \text{pen}(k) - R \left( 48\sqrt{\frac{\hat{k}d}{n}} + 2\sqrt{\frac{x_{\hat{k}}}{2n}} \right) \\ &= W_n(c_k) + 2R\sqrt{\frac{z}{2n}} + \text{pen}(k), \end{aligned}$$

which shows that

$$W(\tilde{c}) \leq W_n(c_k) + 2R\sqrt{\frac{z}{2n}} + \text{pen}(k).$$

Thus

$$\begin{aligned} &\mathbb{P} \left[ W(\tilde{c}) \leq W_n(c_k) + 2R\sqrt{\frac{z}{2n}} + \text{pen}(k) \right] \\ &\geq \mathbb{P} \left[ \bigcap_{l=1}^n \sup_{c \in S_l} (W(\mu, c) - W(\mu_n, c)) \leq \mathbb{E} \left[ \sup_{c \in S_l} (W(c) - W_n(c)) \right] + 2R\sqrt{\frac{x_l + z}{2n}} \right] \geq 1 - \Sigma e^{-z}. \end{aligned}$$

We get

$$\begin{aligned} &\mathbb{P} \left[ W(\tilde{c}) - W_n(c_k) - \text{pen}(k) \geq 2R\sqrt{\frac{z}{2n}} \right] \leq \Sigma e^{-z} \\ &\mathbb{P} \left[ \frac{\sqrt{2n}}{2R} (W(\tilde{c}) - W_n(c_k) - \text{pen}(k)) \geq \sqrt{z} \right] \leq \Sigma e^{-z} \end{aligned}$$

or, setting  $z = u^2$ ,

$$\mathbb{P} \left[ \frac{\sqrt{2n}}{2R} (W(\tilde{c}) - W_n(c_k) - \text{pen}(k)) \geq u \right] \leq \Sigma e^{-u^2}$$

$$\begin{aligned} \mathbb{E} \left[ \frac{\sqrt{2n}}{2R} (W(\tilde{c}) - W_n(c_k) - \text{pen}(k))_+ \right] &= \int_0^\infty \mathbb{P} \left[ \frac{\sqrt{2n}}{2R} (W(\tilde{c}) - W_n(c_k) - \text{pen}(k))_+ \geq u \right] du \\ &\leq \int_0^\infty \mathbb{P} \left[ \frac{\sqrt{2n}}{2R} (W(\tilde{c}) - W_n(c_k) - \text{pen}(k)) \geq u \right] du \\ &\leq \Sigma \int_0^\infty e^{-u^2} du = \Sigma \frac{\sqrt{\pi}}{2}. \end{aligned}$$

We get

$$\mathbb{E} [(W(\tilde{c}) - W_n(c_k) - \text{pen}(k))_+] \leq \Sigma R \sqrt{\frac{\pi}{2n}}.$$

Since  $\mathbb{E} [W_n(c_k)] = W(c_k)$ , we have :

$$\mathbb{E} [W(\tilde{c})] \leq W(c_k) + \text{pen}(k) + \Sigma R \sqrt{\frac{\pi}{2n}}.$$

$$\mathbb{E} [W(\tilde{c})] \leq \inf_{1 \leq k \leq n, c_k \in S_k} \{W(c_k) + \text{pen}(k)\} + \Sigma R \sqrt{\frac{\pi}{2n}}.$$

□

## 5.4 Proof of Proposition 3.1

*Proof.* If  $k \leq 2^d$ , we have  $4Rk^{-1/d} \geq 4R2^{-1} = 2R$ . Thus,  $W(c) \leq 2\sqrt{d} \leq 4\sqrt{d}k^{-1/d}$  for any vector quantizer with codebook  $c$ .

Otherwise, let  $\epsilon = 4Rk^{-1/d}$ . Then  $\epsilon \leq 2R$  and by Lemma 5.3 there exists a set of points  $\{y_1, \dots, y_k\} \subset S(0, R)$  that  $\epsilon$ -covers  $S(0, R)$ . A quantizer with the codebook  $c = \{y_1, \dots, y_k\}$  verifies :

$$W(c) \leq \epsilon \leq 4Rk^{-1/d}.$$

That concludes

$$\inf_{c \in S_k} W(c) \leq 4Rk^{-1/d}.$$

□

## References

- Arlot, S. and Massart, P. (2009). Data-driven calibration of penalties for least-squares regression. *Journal of Machine learning research*, 10(2).
- Bartlett, P. L., Linder, T., and Lugosi, G. (1998). The minimax distortion redundancy in empirical quantizer design. *IEEE Transactions on Information theory*, 44(5):1802–1813.
- Baudry, J.-P., Maugis, C., and Michel, B. (2012). Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470.
- Bello, D. Z., Valk, M., and Cybis, G. B. (2023). Towards u-statistics clustering inference for multiple groups. *Journal of Statistical Computation and Simulation*, pages 1–19.
- Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.
- Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.
- Birgé, L. and Massart, P. (2007). Minimal penalties for gaussian model selection. *Probability theory and related fields*, 138(1):33–73.
- Brault, V., Baudry, J.-P., Maugis, C., Michel, B., and Brault, M. V. (2011). Package ‘capushe’.
- Cardot, H., Cénac, P., and Monnez, J.-M. (2012). A fast and recursive algorithm for clustering large datasets with k-medians. *Computational Statistics & Data Analysis*, 56(6):1434–1449.

- Cardot, H., Cénac, P., and Zitt, P.-A. (2013). Efficient and fast estimation of the geometric median in hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19(1):18–43.
- Cardot, H. and Godichon-Baggioni, A. (2017). Fast estimation of the median covariation matrix with application to online robust principal components analysis. *Test*, 26(3):461–480.
- Cesa-Bianchi, N. and Lugosi, G. (1999). Minimax regret under log loss for general classes of experts. In *Proceedings of the Twelfth annual conference on computational learning theory*, pages 12–18.
- Duflo, M. (1997). Random iterative models, stochastic modelling and applied probability, vol. 34.
- Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.
- Fischer, A. (2011). On the number of groups in clustering. *Statistics & Probability Letters*, 81(12):1771–1781.
- Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769.
- Gagolewski, M., Bartoszek, M., and Cena, A. (2016). Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Information Sciences*, 363:8–23.
- Gersho, A. and Gray, R. M. (2012). *Vector quantization and signal compression*, volume 159. Springer Science & Business Media.
- Haldane, J. (1948). Note on the median of a multivariate distribution. *Biometrika*, 35(3-4):414–417.
- Hoeffding, W. (1994). Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer.
- Huang, H., Liu, Y., Yuan, M., and Marron, J. (2015). Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24(4):975–993.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.
- Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.
- Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.
- Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.
- Kemperman, J. (1987). The median of a finite measure on a banach space. *Statistical data analysis based on the L1-norm and related methods (Neuchâtel, 1987)*, pages 217–230.
- Linder, T. (2000). On the training distortion of vector quantizers. *IEEE Transactions on Information Theory*, 46(4):1617–1623.
- Liu, Y., Hayes, D. N., Nobel, A., and Marron, J. S. (2008). Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293.
- MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.
- Massart, P. (2007). *Concentration inequalities and model selection: Ecole d’Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer.
- McDiarmid, C. et al. (1989). On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188.

- Mirkin, B. (1996). *Mathematical classification and clustering*, volume 11. Springer Science & Business Media.
- Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.
- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.
- Ruppert, D. (1985). A newton-raphson version of the multivariate robbins-monro procedure. *The Annals of Statistics*, 13(1):236–245.
- Spath, H. (1980). *Cluster analysis algorithms for data reduction and classification of objects*. Ellis Horwood Chichester.
- Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.
- Vardi, Y. and Zhang, C.-H. (2000). The multivariate l 1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426.
- Weiszfeld, E. (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386.