Center-based Robust Clustering

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

M.Tech in Computer Science

by

Pranta Das

[Roll No: CS-1923]

under the guidance of

Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit

Indian Statistical Institute
Kolkata-700108, India
July 2021

Certificate

This is to certify that the dissertation entitled "**Center-based Robust Clustering**" submitted by **Pranta Das** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of the institute and, in my opinion, has reached the standard needed for submission.

Swagatam Das

Dr. Swagatam Das
Associate Professor,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Swagatam Das, Associate Professor, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions. Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support.

pranta das

Pranta Das
Indian Statistical Institute
Kolkata - 700108, India

Abstract

We consider the problem of clustering observations $x_i \in \mathbb{R}^d, i = 1, ..., n$ into k possible clusters. We are mainly interested in clustering in the presence of outliers, where classical clustering algorithms face challenges.

In the framework of center-based clustering that uses seeding method to initialize centroid and update the centroid in each iterations, we proposed the method of Modified k-Means clustering. In Modified k-Means method, we introduce a new sampling method for initialize the centroids where the Robust k-Means++ method [1] has been tweaked in a straightforward and understandable way and a new centroid update strategy for avoiding the effect of outlier during centroid update stage. Now use this Modified k-Means algorithm as building blocks we proposed Robust center-based clustering algorithm that provides outlier detection and data clustering simultaneously. The proposed algorithm consists of two stages. The first stage consists of Modified k-Means process, while the second stage iteratively remove the points which are far away from their cluster center. The experimental results suggest that our method has out performed this Robust k-Means++ [1] and also TMK++ [2] and local search (LSO) [3] on real world and synthetic data.

Keywords : Robust center-based clustering, k-Means clustering, Outliers, Robust k-Means++, TMK++, LSO.

Contents

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

Among the center-based clustering method, k-Means algorithm is a widely used tool in data analysis and an important method in statistics and unsupervised learning. The objective of the k-Means clustering is to find k disjoint partitions such that the Sum of Squared Error (SSE) is minimized. More formally, given a set $X \subseteq \mathbb{R}^d$ of n points and number of cluster k, the k-Means objective is to find a set $C = \{c_1, ..., c_k\} \subseteq \mathbb{R}^d$ of k centers and respective k disjoint partitions that minimizes $\sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - c_j||_2^2$. The variation of the k-Means algorithm for kernel methods and Bregman divergence are widely used methods in pattern recognition. The most popular algorithm that try to give approximate solution for the NP-hard k-means objective is Lloyd's method [4] which is originally a vector quantization technique in signal processing. Lloyd's method [4] is a simple, fast heuristic that starts with any random solution and iteratively converges to a local minima. Understanding the importance of the k-means algorithm, Lloyd's method as one of the top ten algorithms used in data mining [5]. However, this method does not provide any theorytical guarantee on the quality of the solution and sometimes it takes exponenital number of iteration for converge to a local minima.

But the k-Means algorithm suffers from two major problems – (a) the objective function itself is highly delicate to outliers and, (b) the k-Means method does not have a good seeding strategy that is not picked outliers as a center of one cluster.

The mean is not a robust statistic that means even a single outlier can change the mean arbitrarily. In that sense, the k-Means objective function is highly delicate to outliers. Although median is robust but the compution of geometric median in high dimension is a non-trivial computational problem [6].

To deal with these problems, several methods have been proposed. Trimmed k-Means method [7] which optimize k-means objective but in a different way, it minimizes the objective on a specific subset of the data points. Random sampling is a popular seeding but sometimes it ended up non desireable clusters. To takle this problem k-Means++ [8] is proposed. Based on this algorithm Bahman Bahmani and Olivier Bachem proposed its faster, scalable versions [9, 10] respectively which are provide good initialization. The k-means++ initialization which is based on D^2 sampling, has a high probability of selecting outliers as centers. A new method to tackle this problem is the Robust k-Means++ algorithm [1], which use convex combination of D^2 and uniform sampling.

Therefore with this noise sensitive objective function and noise sensitive intialization, k-Means algorithm gives a poor quality clustering on noisy data. Our work address this issue positively by improving the k-Means algorithm to make it robust to outlier by introducing a centroid initialization where the Robust k-Means++ method [1] has been tweaked in a straightforward and understandable way and update the centroids in a robust manner. We also proposed Robust center-based clustering algorithm which cluster the data using this Modified k-Means and simultaneously removes points far from the currently estimated centroids.

1.2 Our Contribution

The following is a list of our contributions.

- We have proposed Modified k-Means algorithm which is the improved version of the k-Means algorithm to make it robust to outlier by by introducing a centroid initialization strategy and update the centroids in robust fashion.
- We also proposed Robust center-based clustering algorithm which cluster the data using this Modified k-Means and simultaneously removes points far from the currently estimated centroids.
- We have also provided the performance evaluation of our scheme. We have compared our method with the Robust k-Means++ method [1] , TMK++ [2] and LSO [3] on real world as well as synthetic data sets.
- We have also provide a detailed complexity analysis of our method.

1.3 Thesis Outline

The remainder of the thesis is coordinated as follows. In Chapter 2, we briefly discuss about the preliminaries and clustering with outliers. In Chapter 3, we discuss about the background related to our work. Chapter 4, describes the detailed construction of our scheme. In Chapter 5, we give a detailed performance analysis of our scheme. In Chapter 6, we summarize the work done and discuss about the future directions related to our work.

Chapter 2

Preliminaries

2.1 Notations

We consider $X = (x_{ij}) \in \mathbb{R}^{n \times d}$ to be our data set in matrix format where x_i represents the ith observation (row) and X_j represents the jth feature (column). Here n is the number of observations and d is the number of features. We consider k clusters and the set of cluster centers $\{c_1, ..., c_k\}$, where $c_i \in \mathbb{R}^d$. The kth cluster is represented by C_k and kth cluster center is represented by c_k. V_k denotes variance of the cluster k where the cluster variance is defined as the sum not the average of the squared distances between cluster members and center. δ_{ik} is a cluster indicator variable with $\delta_{ik} = 1$ if x_i belongs to C_k and 0 otherwise.

2.2 The k-Means Algorithm

To partition a data set X into k disjoint clusters, k-Means [11] minimizes the sum of intra-cluster variances (2.1).

$$\sum_{j=1}^{k} V_j = \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_{ij} ||x_i - c_j||_2^2 \tag{2.1}$$

where

$$c_k = \frac{\sum_{i=1}^{n} \delta_{ik} x_i}{\sum_{i=1}^{n} \delta_{ik}} \tag{2.2}$$

2.3 Clustering with the presence of outliers

Even though k-means problem is very much examined, but the algorithms which solve it approximately can perform poorly on real-world data. The reason for it that the k-means objective expects that all of the points can be normally partitioned into k disjoint groups, which is frequently an unreasonable presumption practically speaking. Real-world data comes with a lot of outliers, and the k-means method is highly delicate to it. Outliers can definately change the quality of clustering and consider this into account when designing algorithms for the k-means objective.

To handle the data with outliers, the problem of kmeans with outliers is proposed. In this form of the problem, the clustering objective stays same as before, however the algorithm is furthermore permitted to remove a small set of points from the input data. These discarded points are marked as noise and are overlooked in the objective and in this way allowing the clustering algorithm to focus on correctly clustering the data which is noise-free. Applying the k-means method and listing the top L points that are the farthest distant from their nearest cluster centres as outliers is a basic technique. There is, however, a minor remark to be made: the k-means method is very delicate to anomalies, and such anomalies may affect the final cluster design. This can lead to numerous false negatives, in which data points that should be classified as outliers are suppressed by clustering, as well as false positives, in which data points are mistakenly classified as outliers. As a result, a more robust version of the k-means method is needed to handle the data with outliers.

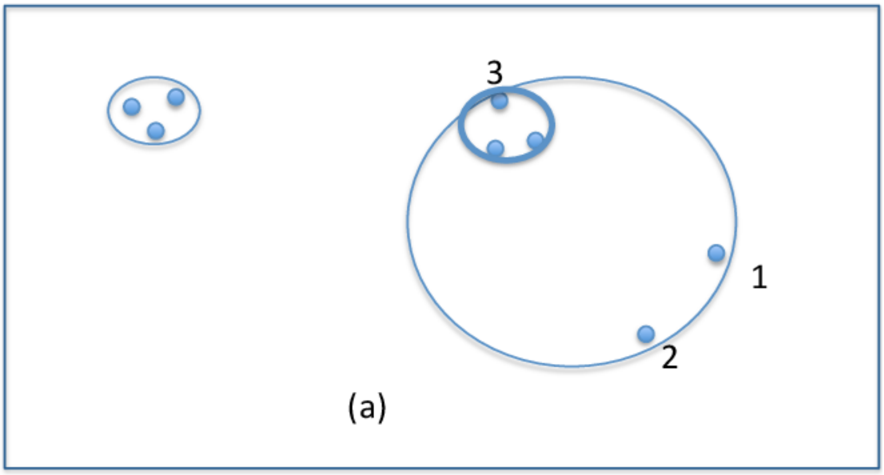


Figure 2.1: Clustering with outlier

Figure 2.1 shows a speculative situation where the k-Means method could possibly be seriously influenced by the presence of outliers. In the event that k = 2, the k-Means will group the five data points directly into one group. Then again, if the k-Means algorithm is intended to at the same time structure groups and track outliers, a more regular result is likewise displayed in this figure, where the large group has become more smaller (striking circle) and the two points (1) and (2) are considered as outliers.

2.4 Classical Multivariate Outlier Detection Review

Anomaly or outlier detection is a profoundly investigated issue in both statistics and machine learning with alternate points of view. In patter recognition, Knorr and Ng [12] proposed a meaning of distance-based anomaly, which is liberated from any distributional suppositions and is generalizable to multidimensional datasets. Instinctively, anomalies are data points that are far away from their closest neighbors.

Several variants and methods for detecting distance-based outliers have been presented in the wake of Knorr and Ng.In any case, the anomalies identified by these strategies are with respect to the entire data set. Breunig et al. [13] have contended that in certain circumstances nearby outliers are a higher priority than outliers with respect to the entire data and can't be effortlessly recognized by standard distance-based strategies. They presented the idea of local outlier factor (LOF), which catches how segregated an item is concerning its encompassing area. The notion of local outliers has now been expanded in a number of ways.

Distance based outlier detection method is easily effected by outliers.Therefore a robust method is necessary for detecting outliers. The most famous robust method in statistics literature for outlier detection is Minimum Covariance Determinant (MCD) [14]. The MCD method tries to minimize the determinant of the covariance matrix Σ over the subset of the original data set and for optimal subset for which $|\Sigma|$ is minimized, considered as inlier points. This method depending on the Mahalanobis distance and a famous theorem around it.

Chapter 3

Related Work

3.1 Past Work on center-based Robust Clustering

Different approaches have been suggested for clustering data in the presence of outliers. Here we do a brief review of the previous proposals for center-based robust clustering.

M. Charikar [15] and K. Chen [16] proposed constant-factor polynomial time approximation algorithm for the k-Means problem with outlier. These methods involve highly complex mathematical optimization and never used in practical scenario.

The development of efficient, practical methods for clustering with presence of outliers is still a hot topic in academia. For the k-center objective, few constant-factor approximation algorithms are known [15, 17, 18] . Nonetheless, the more generally utilized k-means objective stays unattainable. [19] extended Lloyd's technique to the scenario when there are outliers, however there are no assurances about the quality of the solution provided by the algorithm.

A. Deshpande, P. Kacham and R. Pratap [1] proposed Robust k-Means++ method which offer constant-factor approximation to the k-Means objective with outlier and discarding slightly more points than optimal situation. Luis Angel Garcia-Escudero and Alfonso Gordaliza [7] proposed Trimmed k-Means method that also optimize k-Means objective but in slightly different way, it minimizes the objective on all possible subset of particular size of input data set and pick the best one and rest of the points are considered as outliers. Another line of work revisits this idea and proposed an alternate variantion [20]. Shalmoli Gupta developed a local-search algorithm [3] that offers constant factor approximation to the trimmed k-Means objective and simultaneously remove outliers,but the number of outlier remove by this method is excessive. We would broadly discuss trimmed k-Means [7] , LSO [3] and Robust k-Means++ [1] methods in the coming sections, with which we mainly compare our proposal.

3.2 Trimmed k-Means Method

The k-Means algorithm minimizes the Sum of Squares Error (SSE) which can be written as follows :

$$\phi_X(C) = \sum_{x \in X} min_{c \in C} ||x - c||_2^2 \tag{3.1}$$

The trimmed k-Means clustering objective is same as k-Means objective but the method minimize the objective on all possible subset of the data set ranther than with respect to the entire data set. More formally, given the data set $X \subseteq \mathbb{R}^d$,the number of cluster k, and a parameter $\beta \in (0, 1)$ denotes the fraction of outlier, the trimmed k-Means method is optimize the following objective function :

$$\rho_X(C) = min_{Y \subseteq X, |Y|=(1-\beta)n} \sum_{x \in Y} min_{c \in C} ||x - c||_2^2 \tag{3.2}$$

Let $\phi_Q(C)$ denotes the contribution of the points in k-Means objective for the subset $Q \subseteq X$. For the trimmed k-Means problem, let C_{OPT} be the set of optimal k centers and Y_{OPT} be the optimal set of inliers, then $\rho(C_{OPT}) = \phi_{Y_{OPT}}(C_{OPT})$, since the error is only measured over inliers. Now each point of Y_{OPT} is assigned a label according to its closest center in C_{OPT}. Therefore this gives k partitions of Y_{OPT} as $A_1 \cup A_2 \cup ... \cup A_k$ into disjoint subsets with means $\mu_1, \mu_2, ..., \mu_k$ respectively, while $X \setminus Y_{OPT}$ are the outliers. Therefore,

$$\rho(C_{OPT}) = \phi_{Y_{OPT}}(C_{OPT}) = \sum_{j=1}^{k} \phi_{A_j}(\{\mu_j\}) \tag{3.3}$$

3.3 Local search method for k-Means with outlier

For solving the NP-hard k-Means objective approximately, vanila local search algorithm [21, 22] was introduced. It starts with a random k centers, lets call this set C and swap each point $c \in C$ with each point $x \in X$, if this swap decrease the cost then keep this combination otherwise check with another point. In this way the algorithm converges to a local minima and we get the best combination after all the swap. The condition for the termination of the algorithm is $New\ Cost > (1 - \frac{\epsilon}{k}) Old\ cost$.

The extended version of this local search algorithm is also proposed which particularily develop for handle outliers, named as LS-Outlier (LSO). This technique extends the vanila local search algorithm by enabling outliers to be eliminated. It keeps track of a set of outliers and set of centers (outliers are identified as farthest point from these centers) and try to converge to a local minima using the below three steps :

(1) Run the vanila local search algorithm with random set of k centers and noise free data set F, where outliers are considered as farthest point from these k centers.

(2) Now again remove the farthest points from these locally converges center and check if cost is decrease or not. If decreased then stop the iteration and report the best centers,partitions and set of outliers and do not go to step (3), otherwise perform step (3).

(3) Swap a data point with a center, if this swap and remove farthest point from centers decrease the cost, do this until termination condition is satisfied and finally report the optimal centers, partitions of the data point and the set of outliers.

3.4 Robust k-Means++ Method

The robust k-Means++ technique is a simple outlier-resistant modification of the k-means++ method. It uses convex combination of D^2 and uniform sampling for picking initial centroids.

This method produces $O(k)$ clusters while eliminating outliers and providing constant-factor approximation for the trimmed k-Means method. This algorithm is able to produce exactly k partitions if we increase the number of iterations.

Algorithm 1: Robust k-Means++

```
Input: $X$: a data set of $n$ points, $k$ : the number of clusters and
       $\beta \in [0,1]$,$\delta \in (0,1]$ : the parameters.
Output: a set initial centers $Y \subseteq X$
$Y_0 = \emptyset$
for $i \longleftarrow 1, ..., O(k)$ do
    for $j \longleftarrow 1, ...., O(1/\delta)$ do
        $Pr(picking\ x_j) = (1-\alpha)\frac{\phi_{\{x_j\}}(S_{i-1})}{\phi_X(S_{i-1})} + \alpha\frac{1}{n}$
    end
    $Y_i \longleftarrow Y_{i-1} \cup \{x_1, ..., x_{O(1/\delta)}\}$
    $i \longleftarrow i+1$
end
```

At ith iteration of the outer loop of this algorithm gives an inequality, $\sum_{j:\ C_j \in BAD} |C_j| \leq \delta n$, that means the total number of points in the bad clusters are at most δn and in this step the set of initial centers Y_{i-1} provides costant-factor approximation to the trimmed k-Means method. Formally, the algorithomic procedure of this method is follows :

- Select initial centers using algorithm 1 and remove those δn outliers given by the algorithm.
- Calculate the weight of each cluster and find the top k cluster with respect to the weight. Now these top k cluster are treated as the main cluster and the points of the remaining clusters either assigned to a near by main cluster or removed if it is far away from these cluster centers.

Chapter 4

The Proposed Robust Center-based Clustering Algorithm

4.1 Notations

The notations use to express the algorithm are given in the following Table 4.1

Notations	Descriptions
n	Number of observations
d	Number of features
$\mathbf{X} = (x_{ij}) \in \mathrm{R}^{n \times d}$	Data set in matrix form
$\mathbf{C} = (c_{kj}) \in \mathrm{R}^{k \times d}$	Cluster Centers
C_k and $\mu^k, k = 1, ..., K$	The kth cluster and kth cluster center
n_k	Number of observations in cluster k
$[N]$	$\{1, .., N\}$
$\|\|.\|\|_{sp}$	Spectral norm

Table 4.1: Notations used in our method

4.2 Building Blocks of Robust Center-based Clustering

The classical center-based clustering using a very simple and elegant framework which is as follows :

$$min \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - c_j||_2^2 \tag{4.1}$$

where $||x_i - c_j||_2^2$ is the squared distances of the points from its cluster center. But this framework has two main problems -

a) Noise sensitive objective function and

b) Lack of robust center initialization method.

Therfore we try to address these issue and come up with an algorithm which is robust to outliers.

4.2.1 The Formulation

The classical center-based clustering algorithm rely on centroid initialization and this can done by random initialization or k-Means++ method.But both these methods of initialization are prone to outliers.

The k-Means++ method select initial center with probability proportional to the distance from its nearest center. That means it initialize centers that are well separated from each other, but higher the distance means higher the probabilty of picking that point as center allowing this method delicate to outliers. On the other hand, random sampling often times initialize centers in such a way that it ignores the small clusters and break the big clusters, therefore very often natural partitions are not possible.

Most of the center-based clustering algorithm updates the centroid in each iteration using mean. But the mean statistic is highly delicate to outliers, that means even a single outlier can the change the mean arbitarily. Although median is robust measure for central tendency but its computational complexity is exponential in dimension, therefore we need a different approach to update the centroid in our work.

Centroid Initialization : Our work addresses the above issue about centroid initialization by proposing a initialization strategy where Robust k-Means++ [1] method has been tweaked in a straightforward and understandable way. We use convex combination of D^2 and uniform sampling to initialize k centers,which is as follows :

$$Pr(x \in C) = (1-\alpha)D^2 + \alpha U \tag{4.2}$$

Where $\alpha \in [0,1]$ and the D^2 sampling and uniform sampling is respectively as follows :

$$Pr(x \in C) = \frac{\phi_x(C)}{\phi(C)} \tag{4.3}$$

$$Pr(x \in C) = \frac{1}{n} \tag{4.4}$$

where

$$\phi(C) = \sum_{i=1}^{n} min\{\sum_{j=1}^{k} ||x_i - c_j||_2^2\} \tag{4.5}$$

$$\phi_x(C) = min \sum_{j=1}^{k} ||x - c_j||_2^2 \tag{4.6}$$

Centroid Update : Mean statistics is used for centroid update in classical center-based clustering framework despite its non-robustness. Although median is robust statistic, but computing it in high dimension is way more expensive. Therefore, to address this issue we come up with an very simple approach which enables the stability of the centroid even in the presence of outliers.

Before going to forward, we make an assumption about the distribution of the input data, that our data is normaly distributed. Now from the property 68-95-99.7 of normal distribution , we can say that in the range of $\mu - \sigma$ to $\mu + \sigma$ there are 68% data,$\mu - 2\sigma$ to $\mu + 2\sigma$ there are 95% data and $\mu - 3\sigma$ to $\mu + 3\sigma$ there are 99.7%

data situated. Therefore,we can say that if a point x such that $||x - \mu||_2 > 3\sigma$,we consider it as an outlier,where μ is the mean of the distribution. But the threshold 3σ can be misleading when the data contains outlier as σ is not a robust measure of variability.Therefore,instead of σ we can use Median Absolute Deviation(MAD) as a measure of variability which is also robust to outliers.

Lemma : For normal distribution $\hat{\sigma} = 1.4826MAD$ is an unbiased estimator of σ

Proof : Let $\mathbf{X} = \{x_1, ..., x_n\}$ be a set of n points.Then Meadian Absolute Deviation(MAD) of $\mathbf{X}$ is

$$MAD(\mathbf{X}) = median\{|x_1 - \tilde{\mathbf{X}}|, ..., |x_n - \tilde{\mathbf{X}}|\} \tag{4.7}$$

where $\tilde{\mathbf{X}}$ = median of $\mathbf{X}$.
To estimate σ using MAD one must takes $\hat{\sigma} = k.MAD$, where k is determined by the data distribution.
For normal distribution, the range $\mu - MAD$ to $\mu + MAD$ covers 50% of the tha data,where μ is the mean of the distribution. Therefore,

$$Pr(|X - \mu| < MAD) = \frac{1}{2}$$

$$\implies Pr(|\frac{X - \mu}{\sigma}| < \frac{MAD}{\sigma}) = \frac{1}{2}$$

$$\implies Pr(|Z| < \frac{MAD}{\sigma}) = \frac{1}{2}$$

where $Z \sim \mathcal{N}(0, 1)$.
Now from the above equation we get,

$$\phi(\frac{MAD}{\sigma}) - \phi(-\frac{MAD}{\sigma}) = \frac{1}{2} \tag{4.8}$$

Also

$$\phi(-\frac{MAD}{\sigma}) = 1 - \phi(\frac{MAD}{\sigma}) \tag{4.9}$$

where ϕ is a cumulative distribution function of $\mathcal{N}(0, 1)$.
From equation 4.8 & 4.9 we get,

$$\phi(\frac{MAD}{\sigma}) = \frac{3}{4}$$

$$\implies \frac{MAD}{\sigma} = \phi^{-1}(\frac{3}{4})$$

From this we obtain, $k = \frac{1}{\phi^{-1}(\frac{3}{4})} = 1.4826$
Therefore

$$\hat{\sigma} = 1.4826MAD$$

From the above lemma we can set the threshold for detecting outlier as follows :

$$T = 3\sigma = 4.4478MAD$$

Now we make sure that the centroid of the clusters are stable,that means they are not arbitarily far away after update in the presence of otlier. For that we calculate the Median Absolute Deviation(MAD) of each cluster and then obtain the threshold of the respective cluster.Now for each cluster, calculate $||x - c||_2$ for each point x where c is the centroid of the respective cluster. If $||x - c||_2 > T$ then this point x detected as an outlier and cannot be used for centroid update for that cluster. More formally,

Let C^t_{in} and C^t_{out} be the set of inlier and outlier of a cluster respectively in tth iteration.
Therefore, the centroid update for that cluster takes place as follows :

$$\mu^{t+1} = \frac{1}{|C^t_{in}|} \sum_{x \in C^t_{in}} x$$

In that way we can avoid the influence of outlier for centroid update which takes place using mean statistics.

4.2.2 The Proposed Modified k-Means Algorithm

We'd like to keep the naturally pleasant characteristics of the classical center-based clustering objective, even though it is prone to noise, for the clustering in the presence of outliers as we succesfully tackle the influence of outlier.Therefore with this new centroid initialization and centroid update strategy,our modified version of the classical k-Means algorithm as follows:

```
Algorithm 2: Modiffied k-Means
Input: The number of cluster $k$ and $\alpha$ .
Output: The partitions $C_1, C_2, ...., C_k$.
t = 0
while $||\mu_i^{t+1} - \mu_i^t||_2 > \epsilon \ \forall i$ do
    Initialize k centroid using mixture sampling, i.e,
      $Pr(x \in C) = (1 - \alpha)D^2 + \alpha U$
    Each point assigned to a cluster using k-Means objective.
    Calculate the threshold for each cluster $T_1, T_2, ..., T_k$ respectively,where
      $T_i = 4.4478MAD_i$
    Update the cluster centroid as $\mu_i^{t+1} = \frac{1}{|C^t_{in}|} \sum_{x \in C^t_{in}} x$
    $t = t + 1$
end
```

Note: For $t = 0$,the condition $||\mu_i^{t+1} - \mu_i^t||_2 > \epsilon \ \forall i$ is always true.

4.2.3 The Alternate Proposal of Modified k-Means Algorithm

Algorithm 1 depicts the Modified k-Means algorithm for clustering which is heavily dependent on the assumption that our data is normaly distributed.But in real life that is always not the case,the data came from any distribution.Therefore to avoid this assumption about the distribution of the data,we come up with an alternate proposal of Modified k-Means algorithm where we update the centroid by robust mean estimation.

We utilize the structure for the mean estimation in a robust manner, acquainted in [23]. The goal is to give each data point a non-negative weight (inlier data points gets more weight than outliers) such that the weighted mean is near to the real mean. More formally,given data points $x_1, ..., x_n \in \mathbb{R}^d$ with corresponding data matrix $X \in \mathbb{R}^{d \times n}$,the goal is to find a weight vector $w \in \mathbb{R}^n$ such that the weighted mean μ_w is as close as possible to the real mean μ^*. The only constraint on w is that it belongs to the set

$$\Psi_{n,\epsilon} = \{w \in \mathbb{R}^n : \sum |w_j| = 1 \ and \ 0 \leq w_j \leq \frac{1}{(1-\epsilon)n} \forall j\}$$

where $\Psi_{n,\epsilon}$ is a convex hull.

The lemma proved by Diankonikolas in [24] says if a weight vector w minimizes the spectral norm of the weighted covariance matrix, $\Sigma_w = \sum_{i=1}^{n} (x_i - \mu_w)(x_i - \mu_w)^T$ then the weighted mean provided by this w is close to the real mean. Therefore the optimization problem that gives us best w looks like :

$$min \ ||\Sigma_w||_{sp} \ \ such \ that \ w \in \Psi_{n,2\epsilon} \tag{4.10}$$

Now the problem is that the spectrul norm is non-convex and therefore the avobe optimization problem is difficult to solve. But using the subgradient of the spectrul norm, the projected sub-gradient descent algorithm on (4.10) outputs a approximate stationary point w after $O(n^2 d^4)$ iterations which serves our purpose .

Algorithm 3: Mean estimation in a robust manner

```
Input: a set of n samples $\{x_i\}_{i=1}^{n}$ on $\mathbb{R}^d$ which contains outliers
Output: $w \in \mathbb{R}^n$ and optimal weighted mean
Let $F(w,u) = u^T \Sigma_w u$ , so $||\Sigma_w||_{sp} = Max_{||u||_2=1} \ F(w,u)$
Let $T = O(n^2 d^4)$
Start with any $w_0 \in \mathcal{K} = \Delta_{n,2\epsilon}$
for $t = 0 \ to \ T-1$ do
    Let $v \in \ argmax_{||v||_2=1} \ v^T \Sigma_w v$
    $w_{t+1} \longleftarrow \mathcal{P}_{\mathcal{K}}(w_t - \eta \frac{\partial(v^t \Sigma_w v)}{\partial w})$
end
```

Using the above robust mean estimation method, the new version of the Modified k-Means algorithm is as follows :

```
Algorithm 4: New Version of Modified k-Means
Input: The number of cluster $k$ and parameter $\alpha$
Output: The partitions $C_1, ..., C_k$
$t = 0$
while $||\mu_i^{t+1} - \mu_i^t||_2 > \epsilon \forall i$ do
    Initialize $k$ centroid using mixture sampling , $i.e$,
      $Pr(x \in C) = (1 - \alpha)D^2 + \alpha U$
    Each point assigned to a cluster using k-Means objective.
    For each cluster update the centroid as $\mu_i^{t+1} \longleftarrow RobustMean(C_i^t)$
    $t = t + 1$
end
```

Note: For $t = 0$,the condition $||\mu_i^{t+1} - \mu_i^t||_2 > \epsilon \ \forall i$ is always true.

4.2.4 Performance Analysis of Modified k-Means

The Modified k-Means algorithm now applies to synthetic as well as real world data set and compare the results with k-Means[11],k-Means++[8], k-Median algorithm.The data set descriptions are given bellow :

Dataset Name	n	K	Outlier
Synthetic	1100	20	100
Shuttle	43500	3	180

Table 4.2: Dataset used for Modified k-Means

The below Table 4.3 , 4.4 shows the result of the Modified k-Means algorithm on synthetic as well as real world data set and comparison with other algorithms based on **Sum of Squared Error(SSE)** , **Silhouette Score(SC)**.

Algorithm	α	SSE	SC
Modified k-Means	0.1	**17843.4**	**0.77**
k-Means++		23971.6	0.73
k-Means		36868.9	0.63
k-Median		23842.5	0.72

Table 4.3: Modified k-Means on Synthetic Data set

Algorithm	α	SSE	SC
Modified k-Means	0.1	**1066348689**	**0.91**
k-Means++		1142392287.74	0.83
k-Means		1544776568.87	0.52
k-Median		1771149274	0.29

Table 4.4: Modified k-Means on Shuttle data set

4.2.5 Selection of α in Modified k-Means

The α used in mixture sampling for Modified k-Means is lies in the range of $0 \leq \alpha \leq 1$.Therefore we search the value of α in this range and pick the value for which Modified k-Means converges with lowest cost.

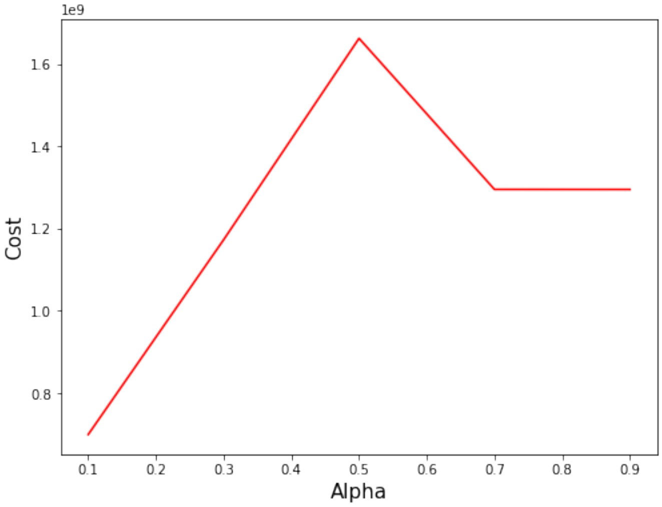


Figure 4.1: Changing of cost with respect to α

From the above Figure 4.1 we clearly see that for $\alpha = 0.1$ Modified k-Means converges with lowest cost than the other values of α. Therefore in this scenario, we pick $\alpha = 0.1$ for our Modified k-Means algorithm.

4.3 Robust Center-based Clustering Algorithm

Robust k-Means algorithm doing clustering while also removing possible outliers simultaneously. It uses Modified k-Means clustering algorithm as a back bone. Since Modified k-Means algorithm able to do good quality clustering even in the presence of outlier, we use this algorithm to do the clustering part and simultaneously we remove the outlier by means of normalized threshold.

Algorithm 5: Robust center-based clustering

```
Input: The data matrix $X$ and parameters $\alpha$ , $I$ , $T$
Output: The partitions $C_1, ..., C_k$ without outliers
$C \longleftarrow$ Modified k-Means$(X, \alpha)$
for $j \longleftarrow 1, ..., I$ do
    $d_{max} = max_i\{||x_i - c_k^i||_2\}$ , $c_k^i$ : Cluster center of $x_i$
    for $i \longleftarrow 1, ..., n$ do
        $U_i = ||x_i - c_k^i||_2/d_{max}$
        if $U_i > T$ then
            $X \longleftarrow X \setminus \{x_i\}$
        else
            $X \longleftarrow X$
        end
    end
    $(C, P) \longleftarrow$ Modified k-Means $(X, \alpha, C)$
end
```

Note : $P \longrightarrow$ Set of partitions.

4.3.1 Selection of tuning parameters

The parameters used in Robust center-based clustering algorithm are α, I, T. The selection of α is same as described in Section 4.2.5, that is apply the Modified k-Means algorithm on the data and select the α with lowest cost. The value of I is taken as vk, where v is the number of outliers and k is the number of cluster.It is a completely heuristic method of seleting I,there is no systemic procedure for that. By setting the threshold to $T < 1$, atleast one point removed as an outlier because we used normalized distance for detection of outlier. Thus, increasing the number of iterations and decreasing the threshold will in effect remove more number of points, possibly more than the number of outlier. Therefore, there is a tradeoff between the parameters I and T.

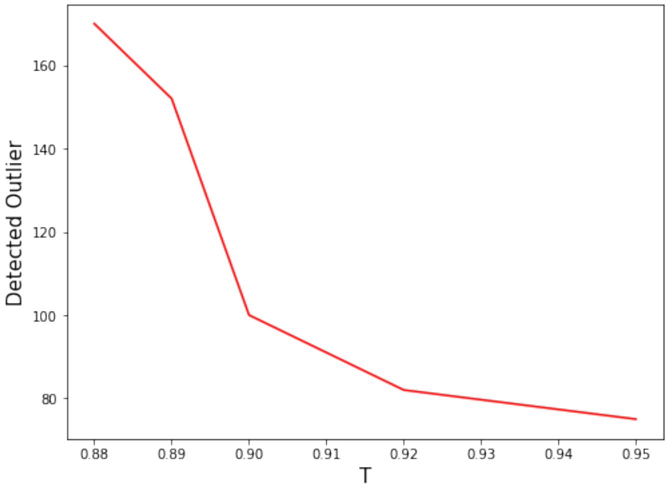


Figure 4.2: Number of detected outlier changes w.r.t threshold T

To resolve the tradeoff between I and T, fix the value of I as described avove and vary the threshold to choose optimal value of T for which optimal number of outlier is detected if the number of outlier present in the data we know beforehand. Figure 4.2 shows a similar illustration in a dataset with 100 outlier, therefore from the figure we clearly see that 0.9 is the optimal value of T. But if the number of outlier is not known beforehand then we set a tight threshold T and $T = 0.95$ is the rule of thumb.

4.3.2 Convergence analysis of the algorithm

Robust center-based clustering algorithm use Modified k-Means algorithm in its backbone and Modified k-Means algorithm inherently use k-Means framework. Therefore, Robust center-based clustering algorithm optimize the same old k-Means objective. Hence, the convergence of Robust center-based clustering algorithm follows directly from convergence of k-Means [11] algorithm. But the Robust center-based clustering algorithm converges with much lower cost than the k-Means algorithm as the algorithm constantly detect and remove outlier in each iteration.

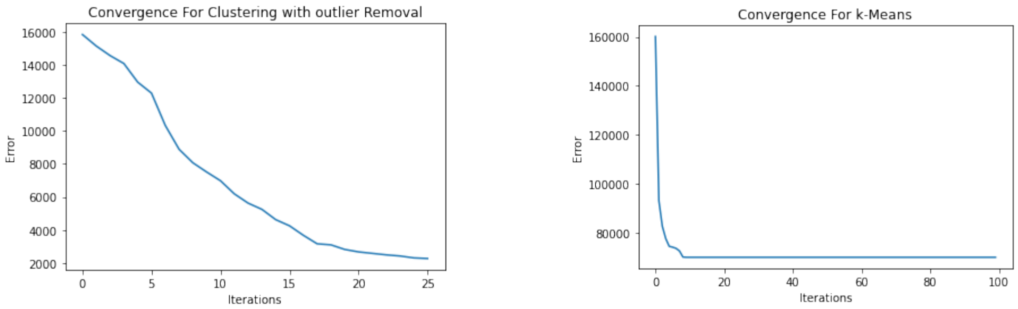


Figure 4.3: **Left :** cost changes with iterations in Robust center-based clustering algorithm, **right :** cost changes with iterations in k-Means algorithm

From the Figure 4.3 we can see that Robust center-based clustering algorithm converges with cost 2000, where k-Means[11] algorithm converges with cost nearly 7000 when applying both the algorithm on same dataset.

Chapter 5

Performance Analysis of Robust Center-based Clustering

5.1 Complexity Analysis

The time complexity of the classical k-Means is $O(nkdi)$, where i is the number of iterations,n is the number of samples and d is the dimensionality of the data. The Modified k-Means algorithm's centroid update step takes $O(k\lfloor\frac{n}{k}\rfloor\log\lfloor\frac{n}{k}\rfloor)$ time for each iteration, hence the Modified k-Means algorithm has a time complexity of $O((nkd+k\lfloor\frac{n}{k}\rfloor\log\lfloor\frac{n}{k}\rfloor)i)$. Now in each iteration of the Robust center-based clustering, we have to find the distance of the farthest point from its center and it takes $O(nd+n\log n)$ time,aso we detect outlier based on the normalized distance of each point and compare it with the threshold and finally among inlier apply Modified k-Means. Hence, the time complexity for each iteration of the proposed Robust center-based clustering algorithm is $O(n(d+\log n)+(nkd+k\lfloor\frac{n}{k}\rfloor\log\lfloor\frac{n}{k}\rfloor)i)$ and we iterate untill convergence.

5.2 Experimental Study

We analyse and compare the performance of our Robust center-based clustering algorithm primarily with Robust k-Means++, LSO, TKM++ etc in this section mainly based on Precision and Recall measure, which is defined as

$$Precision = \frac{U \cap U^*}{U}, Recall = \frac{U \cap U^*}{U^*} \tag{5.1}$$

Where U^* is the optimal number of outlier and U is the number of outlier detected by the algorithm.

We first evaluate the performance of our scheme on synthetic 2-D shape data sets, so that the results can be visualized. We then compare the performance of our scheme on UCI real world data sets. Each of the simulation is repeated 10 times.

5.2.1 Description of data sets used in our study

Dataset Name	K	n	Outlier
Synthetic-1	20	1025	25
Synthetic-2	20	1050	50
Synthetic-3	20	1100	100
Synthetic-4	7	804	20
Synthetic-5	5	1040	40

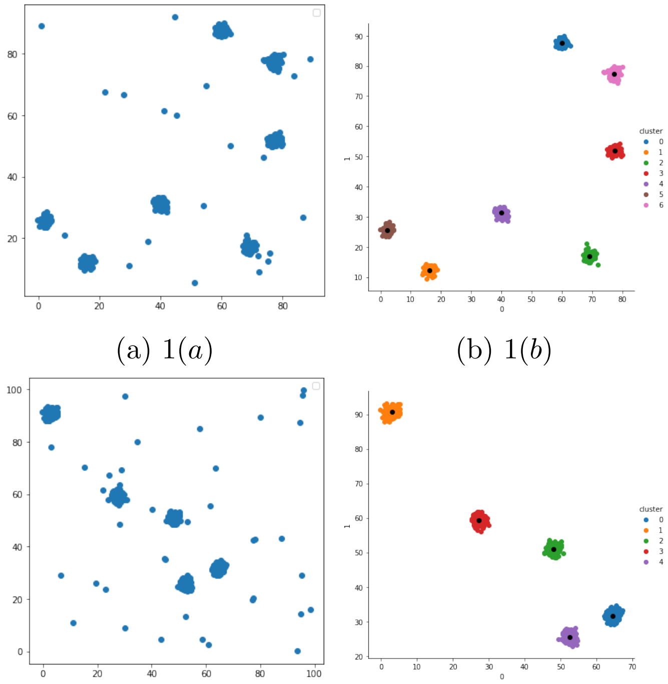
Table 5.1: Synthetic 2D shape data sets

Dataset Name	K	n	Outlier
Shuttle-1	5	43500	21
Shuttle-2	10	43500	34
Shuttle-3	15	43500	51

Table 5.2: UCI real world data sets

5.2.2 Evaluation on Synthetic 2-D shape data sets

We use 5 synthetic 2-D shape data sets, the details of which are given in Table 5.1. We apply the proposed Robust center-based clustering algorithm to these data sets and plot the results. The results obtain on two of the data sets synthetic-4 and synthetic-5 is shown in the Figure 5.1. $1(a)$ and $2(a)$ show the original cluster distributions, and $1(b)$ and $2(b)$ show the respective results obtained by Robust center-based clustering algorithm.



(a) $1(a)$

(b) $1(b)$

(c) $2(a)$

(d) $2(b)$

Figure 5.1: Robust center-based clustering results on synthetic 2-D shape data sets

We apply Robust center-based clustering algorithm with parameter α , T and number of iteration I . The value of these tuning parameter is obtained by the method discussed in Section. The detailed results obtained on these data sets is shown in the Table 5.3.

Dataset Name	α	T	I	SSE	SC	Precision	Recall
Synthetic-1	0.1	0.9	16	2705.7	0.85	0.97	0.78
Synthetic-2	0.1	0.9	20	3652.5	0.8	1	0.69
Synthetic-3	0.1	0.9	25	3369.4	0.84	0.99	0.7
Synthetic-4	0.1	0.9	15	1599.5	0.86	1	1
Synthetic-5	0.1	0.9	19	1942.3	0.9	1	1

Table 5.3: Results of Robust center-based clustering on synthetic 2-D data sets

We also apply Robust k-Means++,TKM++[2] etc on the synthetic data set 1,2 and 3. We compare the results and summarize it in the below Tables 5.4, 5.5, 5.6 .

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	16	**0.97**	0.78
Robust k-Means++	1			0.9	0.9
TMK++				0.65	0.65
k-Means++				0.51	0.51
RAND				0.07	0.07
LSO				0.94	**0.94**

Table 5.4: Robust center-based clustering on synthetic-1 dataset

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	20	**1**	0.69
Robust k-Means++	1			0.9	0.9
TMK++				0.86	0.86
k-Means++				0.5	0.5
RAND				0.07	0.07
LSO				0.91	**0.91**

Table 5.5: Robust center-based clustering on synthetic-2 dataset

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	25	**0.99**	0.7
Robust k-Means++	1			0.79	**0.95**
TMK++				0.49	0.59
k-Means++				0.37	0.44
RAND				0.21	0.26
LSO				0.72	0.91

Table 5.6: Robust center-based clustering on synthetic-3 dataset

From the table we can see that Robust center-based clustering achieve higher Precision than the other methods for all the 3 data sets.

5.2.3 Evaluation on UCI real world data sets

We use the real world data sets mentioned in Table 5.2 , which are obtained from UCI machine learning repository.

Here we apply our Robust center-based clustering algorithm with $\alpha = 0.1$ and $T = 0.9$ and we use I values 10,13,15. We tuned these parameter as discussed in section. The comparison of our algorithm with Robust k-Means++, TMK++ etc algorithm for these data sets are given in Table 5.7 ,5.8, 5.9. In all of the cases our scheme out performs the other methods.

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	10	**0.35**	**0.35**
Robust k-Means++	1			0.19	0.23
TMK++				0.17	0.21
k-Means++				0.16	0.20
RAND				0.19	0.23
LSO				0.17	0.21

Table 5.7: Robust center-based clustering on shuttle-1 dataset

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	13	**0.4**	**0.35**
Robust k-Means++	0.25			0.17	0.35
TMK++				0.14	0.29
k-Means++				0.15	0.31
RAND				0.14	0.29
LSO				0.17	0.35

Table 5.8: Robust center-based clustering on shuttle-2 dataset

Algorithm	α	T	I	Precision	Recall
Our Work	0.1	0.9	15	**0.35**	**0.7**
Robust k-Means++	0.75			0.22	0.67
TMK++				0.17	0.52
k-Means++				0.17	0.52
RAND				0.13	0.41
LSO				0.18	0.55

Table 5.9: Robust center-based clustering on shuttle-3 dataset

5.3 Statistical test for significance

We mainly compare our results with other algorithm by taking the average measure of the repeated simulation. But in this way we are not always confidently say that our algorithm is better than the others. For that we need to perform a statistical test namely **Wilcoxon signed-rank test** [25], which is a non-parametric hypothesis test. This test takes repeated outcomes of our algorithm and other algorithm and tell that the population mean of repeated outcomes of the two algorithm are different or not. The null hypothesis represents population mean are same and alternate hypothesis reoresents population mean are different.

We apply Wilcoxon signed-rank test [25] on our algorithm repeated outcomes against other algorithms and obtain the p-values for the test. The p-values are given in the Table 5.10 .

Data set	Robust k-Means++	TMK++	k-Means++	RAND	LSO
Synthetic-1	0.02	0.01	0.01	0.01	0.01
Synthetic-2	0.02	0.01	0.01	0.01	0.01
Synthetic-3	0.01	0.01	0.01	0.01	0.01
Synthetic-4	0.02	0.01	0.01	0.01	0.01
Synthetic-5	0.03	0.02	0.01	0.01	0.01
Shuttle-1	0.01	0.01	0.01	0.01	0.01
Shuttle-2	0.01	0.01	0.01	0.01	0.01
Shuttle-3	0.02	0.01	0.01	0.01	0.01

Table 5.10: p-values for the hypothesis test

From the above table we can see that all the $p\ values\ < 0.05$, that means we can say that our results are significantly better than other algorithm with 95% confidence.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Clustering data in the presence of outliers is challenging. In our study , which is inspired by the literature of Center-based clustering, we proposed a novel Modified k-Means algorithm with new centroid initialization method and modified centroid update strategy. Based on this Modified k-Means algorithm, we proposed a clustering algorithm which is removeing outliers by means of normalized threshold while doing clustering and we named it as the Robust center-based clustering algorithm.

The experimental results obtain in Section 4.2.4 confirmed outperformance of our approach Modified k-Means over other approaches like k-Means [11], k-Means++ [8], k-Median etc in all the cases when evaluated on synthetic 2D shape as well as real world data sets. The experimental results obtained in Section 5.2 confirmed outperformance of our approach Robust center-based clustering over other approaches like Robust k-Means++ [1], LSO [3], TMK++ [2] etc in all the cases. For the UCI and the sythetic 2D shape data sets, our scheme has performed better or equal in most of the cases.In the cases where other algorithm have produced better recall,our scheme is only marginally behind. But in the metrics like precision, SSE and Silhouette Score, our scheme consistantly better than all the other methods.

For our comparisons, we have used original codes by Amit Deshpande and Rameshwar Pratap [1] for Robust k-Means++ and quote the results for TMK++ [2] and LSO [3] from Robust k-Means++ paper. The code for our implementation of the Robust center-based clustering algorithm is available at this repository https://github.com/pranta123456/Center-based-Robust-Clustering .

6.2 Scope for Future Work

Despite its good performance, this method has a few limitations and there is scope of improvement. Firstly, the parameters α, I and T included in the Robust center-based clustering algorithm, are to be entered by the user and not completely auto tuned. We select the parameter I as heuristic and based on that we select T for which optimal number of outlier detected if we know the number of outlier in advance, otherwise we set $T = 0.95$ as a rule of thumb. We can address this issue in future by tuned these parameter in data driven approach. Secondly, we can also explore and extend our algorithm to overlapping clustering for more meaningfull cluster by using an approach similar to the one suggested in [26], where rather than optimizing a non-constrained objective we optimize constrained objective, where the constrained on the total sum of the number of points in each cluster and this total sum should be $> n$ .Also we can explore how this algorithm works when impose divengence measure by using an approach similar to the one suggested in [27]. Finally, we can look to improve the time complexity of our algorithm.

Bibliography

[1] A. Deshpande, P. Kacham, and R. Pratap, “Robust k-means++,” in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)* (J. Peters and D. Sontag, eds.), vol. 124 of *Proceedings of Machine Learning Research*, pp. 799–808, PMLR, 03–06 Aug 2020.

[2] A. Bhaskara, S. Vadgama, and H. Xu, “Greedy sampling for approximate clustering in the presence of outliers,” in *NeurIPS*, 2019.

[3] S. Gupta, R. Kumar, K. Lu, B. Moseley, and S. Vassilvitskii, “Local search methods for k-means with outliers,” *Proc. VLDB Endow.*, vol. 10, no. 7, pp. 757–768, 2017.

[4] S. Lloyd, “Least squares quantization in pcm,” *IEEE Trans. Inf. Theor.*, vol. 28, p. 129–137, Sept. 2006.

[5] X. Wu, V. Kumar, Q. Ross, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg, “Top 10 algorithms in data mining,” *Knowledge and Information Systems*, vol. 14, pp. 1–37, Jan. 2008.

[6] M. B. Cohen, Y. T. Lee, G. L. Miller, J. W. Pachocki, and A. Sidford, “Geometric median in nearly linear time,” *CoRR*, vol. abs/1606.05225, 2016.

[7] J. Cuesta-Albertos, A. Gordaliza, and C. Matrán, “Trimmed k-means: An attempt to robustify quantizers,” *The Annals of Statistics*, vol. 25, pp. 553–576, 04 1997.

[8] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” in *SODA ’07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[9] O. Bachem, M. Lucic, S. H. Hassani, and A. Krause, “Fast and provably good seedings for k-means,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 55–63, 2016.

[10] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, “Scalable k-means++,” *CoRR*, vol. abs/1203.6402, 2012.

[11] J. Hartigan and M. Wong, “A k-means clustering algorithm,” vol. 28, pp. 100–108, 1979.

[12] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: Algorithms and applications,” *The VLDB Journal*, vol. 8, p. 237–253, Feb. 2000.

[13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers,” *SIGMOD Rec.*, vol. 29, p. 93–104, May 2000.

[14] P. Rousseeuw and K. V. Driessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol. 41, pp. 212–223, 1999.

[15] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan, “Algorithms for facility location problems with outliers,” in *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 642–651, Dec. 2001. 2001 Operating Section Proceedings, American Gas Association ; Conference date: 30-04-2001 Through 01-05-2001.

[16] K. Chen, “A constant factor approximation algorithm for ¡i¿k¡/i¿-median clustering with outliers,” in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’08, (USA), p. 826–835, Society for Industrial and Applied Mathematics, 2008.

[17] G. Malkomes, M. J. Kusner, W. Chen, K. Q. Weinberger, and B. Moseley, “Fast distributed ¡i¿k¡/i¿-center clustering with outliers on massive data,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, (Cambridge, MA, USA), p. 1063–1071, MIT Press, 2015.

[18] R. McCutchen and S. Khuller, “Streaming algorithms for k-center clustering with outliers and with anonymity,” in *Approximation, Randomization and Combinatorial Optimization*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 165–178, Sept. 2008. 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2008 and 12th International Workshop on Randomization and Computation, RANDOM 2008 ; Conference date: 25-08-2008 Through 27-08-2008.

[19] S. Chawla and A. Gionis, “k-means–: A unified approach to clustering and outlier detection,” in *13th SIAM International Conference on Data Mining, Austin, Texas, 2013*, pp. 189–197, 2013. VK: hiit.

[20] A. Georgogiannis, “Robust k-means: a theoretical revisit,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2883–2891, 2016.

[21] A. Gupta and K. Tangwongsan, “Simpler analyses of local search algorithms for facility location,” *CoRR*, vol. abs/0809.2554, 2008.

[22] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “A local search approximation algorithm for k-means clustering,”

in *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, SCG '02, (New York, NY, USA), p. 10–18, Association for Computing Machinery, 2002.

[23] Y. Cheng, I. Diakonikolas, R. Ge, and M. Soltanolkotabi, "High-dimensional robust mean estimation via gradient descent," *CoRR*, vol. abs/2005.01378, 2020.

[24] I. Diakonikolas, G. Kamath, D. M. Kane, J. Z. Li, A. Moitra, and A. Stewart, "Robust estimators in high dimensions without the computational intractability," *CoRR*, vol. abs/1604.06443, 2016.

[25] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[26] J. J. Whang, Y. Hou, D. F. Gleich, and I. S. Dhillon, "Non-exhaustive, overlapping clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2644–2659, 2019.

[27] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. 58, pp. 1705–1749, 2005.