Convex Clustering Redefined: Robust Learning with the Median of Means Estimator

Koustav Chowdhury[1*], Bibhabasu Mandal[1*], Sourav De[1*], Sagar Ghosh[2], Swagatam Das[3], Debolina Paul[4], Saptarshi Chakraborty[5]

[1]Indian Statistical Institute, Kolkata
[2]Department of Statistics and Data Science, University of Texas at Austin
[3]Electronics and Communications Sciences Unit, Indian Statistical Institute, Kolkata
[4]Department of Statistics, University of Oxford
[5]Department of Statistics, University of Michigan
koustavchowdhury2003@gmail.com, bibhabasumandal04@gmail.com, desourav02@gmail.com, sagarghosh1729@utexas.edu, swagatamdas19@yahoo.co.in, debolina.paul@stats.ox.ac.uk, saptarsc@umich.edu

Abstract

Clustering approaches that utilize convex loss functions have recently attracted growing interest in the formation of compact data clusters. Although classical methods like $k-$means and its wide family of variants are still widely used, all of them require the number of clusters (k) to be supplied as input and many are notably sensitive to initialization. Convex clustering provides a more stable alternative by formulating the clustering task as a convex optimization problem, ensuring a unique global solution. However, it faces challenges in handling high-dimensional data, especially in the presence of noise and outliers. Additionally, strong fusion regularization, controlled by the tuning parameter, can hinder effective cluster formation within a convex clustering framework. To overcome these challenges, we introduce a robust approach that integrates convex clustering with the Median of Means (MoM) estimator, thus developing an outlier-resistant and efficient clustering framework that does not necessitate a prior knowledge of the number of clusters. By leveraging the robustness of MoM alongside the stability of convex clustering, our method enhances both performance and efficiency, especially on large-scale datasets. Theoretical analysis demonstrates weak consistency under specific conditions, while experiments on synthetic and real-world datasets validate the method's superior performance compared to existing approaches.

Code and Supplementary Material — https://tinyurl.com/2v3dx75x
Benchmark Dataset (CC BY-NC-ND 4.0) — https://tinyurl.com/2zatwkf3
Keel Datasets (GPLv3) — https://tinyurl.com/bdvmd86m
ASU Datasets (GPLv2) — https://tinyurl.com/49n36ume
Micro-array Datasets — https://tinyurl.com/2f2pjz7j
Brain Dataset — https://tinyurl.com/4ntav7b9
Wisconsin Dataset — https://tinyurl.com/58wxjha5
Extended Version — https://tinyurl.com/2v3dx75x

*These authors contributed equally.

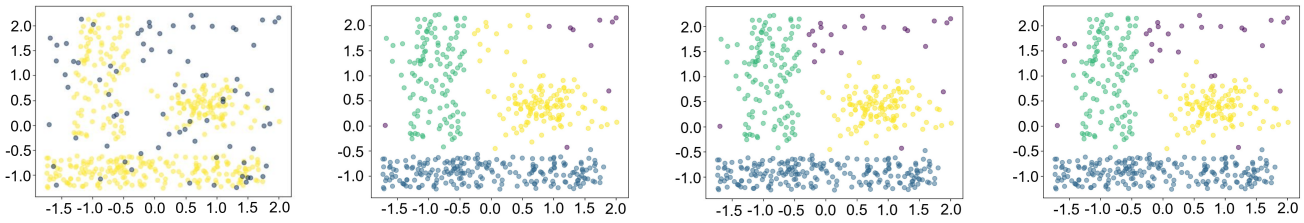
Introduction

Clustering is a fundamental task in unsupervised learning, aiming to organize unlabeled data into coherent groups for better interpretation and downstream applications. It plays a critical role in diverse areas such as customer segmentation (Kansal et al. 2018), image analysis (Münz, Li, and Carle 2007), and anomaly detection (Coleman and Andrews 1979). Traditional algorithms, such as k-means, approach clustering as a non-convex optimization problem (Lu and Zhou 2016), typically solved using greedy heuristics. Although computationally efficient and widely used, these methods suffer from several well-known limitations (Jain 2010): they require pre-specifying the number of clusters (Tibshirani, Walther, and Hastie 2001; Hamerly and Elkan 2004), are sensitive to initialization (Ostrovsky et al. 2013; Xu and Lange 2019), and degrade in performance in high-dimensional spaces or when the data contains noise and outliers (Witten and Tibshirani 2010; De Amorim 2016; Chakraborty and Das 2022).

To overcome these challenges, convex relaxations of non-convex clustering problems have gained significant attention (Tropp 2006). A prominent example is *convex clustering* (or sum-of-norms clustering), which enjoys strong theoretical guarantees such as global optimality and convergence, while remaining broadly applicable in practice (Pelckmans et al. 2005a; Hocking et al. 2011a; Lindsten, Ohlsson, and Ljung 2011a). Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, where each row represents a data point in d-dimensional Euclidean space, convex clustering solves the following objective:

$$\min_{\mathbf{u}} \frac{1}{2}\left[\|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \gamma \sum_{i,j} w_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|_p^2\right], \quad (1)$$

where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$ and denotes the cluster center attached to point $\mathbf{x}_i$, w_{ij} are edge weights, and $\|\cdot\|_p$ is the ℓ^p norm. The first term encourages each point to remain close to its centroid, while the second term (controlled by tuning parameter $\gamma > 0$) promotes fusion across centroids, effectively determining the number of clusters (Chi and Steinerberger 2019). Although convex clustering is effective even



(a) Dataset (b) $\mu = 0.4$ (c) $\mu = 0.2$ (d) $\mu = 0.12$

Figure 1: Figure 1a shows the original dataset in yellow, with 20% added noise represented by blue dots. As μ decreases, our method progressively identifies more noise points as outliers, which are marked by purple dots in Figures 1b, 1c, and 1d respectively.

in large-sample settings (Radchenko and Mukherjee 2017), strong regularization can lead to undesirable merging of outliers with genuine clusters, especially in high-dimensional data (Feng, Chen, and Liu 2023).

This paper focuses on addressing the **robustness challenges** of clustering in the presence of noise and outliers. Robust methods can mitigate these effects by either discarding outlier features (Wang et al. 2016) or directly controlling the influence of anomalous data points. One powerful approach is the *Median-of-Means (MoM)* estimator, which provides strong robustness and concentration guarantees under mild assumptions (Lugosi and Mendelson 2017; Lerasle 2019; Lecué and Lerasle 2020; Laforgue, Clemencon, and Bertail 2019; Bartlett, Boucheron, and Lugosi 2002). Related work by (Paul et al. 2021) further unifies robust center-based clustering under general dissimilarity measures.

Consider n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ to be grouped into k clusters. Each cluster is represented by a centroid $\boldsymbol{\theta}_j \in \mathbb{R}^d$, and the set of centroids forms a matrix $\boldsymbol{\Theta} \in \mathbb{R}^{k \times d}$. Using a Bregman divergence $d_\phi(\cdot, \cdot)$ as the dissimilarity measure, where $\phi : \mathbb{R}^d \to \mathbb{R}$ is a differentiable convex function, clustering can be formulated as

$$f_{\boldsymbol{\Theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{1}{n} \sum_{i=1}^{n} \Psi_\alpha(d_\phi(\mathbf{x}_i, \boldsymbol{\theta}_1), \ldots, d_\phi(\mathbf{x}_i, \boldsymbol{\theta}_k)), \tag{2}$$

where $\Psi_\alpha : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a non-decreasing function, $\Psi_\alpha(0) = 0$, and α is a hyperparameter. Different choices of ϕ and Ψ_α recover well-known clustering algorithms such as k-means, power k-means, and k-harmonic-means.

Instead of directly minimizing (2), MoM partitions the data into L disjoint subsets $B_1, \ldots, B_L$, each containing b samples, and optimizes a robust median-based objective:

$$MoM_L^n(\boldsymbol{\Theta}) = Median\left(\left\{ \frac{1}{b} \sum_{i \in B_j} f_{\boldsymbol{\Theta}}(\mathbf{x}_i) \right\}_{j=1}^{l} \right). \tag{3}$$

Because outliers typically contaminate only a fraction of the partitions, the median effectively suppresses their influence, ensuring robustness even in adversarial settings. Formal breakdown-point analyses of MoM estimators support this intuition (Rodriguez and Valdora 2019; Guillaume and Matthieu 2017).

To demonstrate our method, Figure 1 illustrates results on a benchmark dataset with $k = 5$, $N = 1000$, $\gamma = 5000$, 20% noise, and varying μ. Outliers near the boundary are successfully isolated. When any pair μ_i, μ_j exceeds the separation threshold μ, the connecting edge is dropped, preventing spurious merging. While most outliers are identified, some deeply embedded points remain undetected.

Contributions

In this paper, we present a novel clustering framework that extends convex clustering with enhanced robustness using the Median-of-Means (MoM) estimator. Our main contributions are summarized below:

- **Robust Convex Clustering Framework:** We propose a new clustering method that integrates the MoM estimator into the convex clustering paradigm, effectively mitigating the adverse impact of outliers and noisy data. We also develop a dedicated algorithm for the proposed framework, ensuring practical applicability and computational efficiency.
- **Theoretical Guarantees:** We establish uniform deviation bounds and concentration inequalities under standard regularity assumptions, providing strong theoretical reliability for our method.
- **Empirical Validation:** Extensive simulation studies demonstrate that our method consistently outperforms conventional clustering approaches, particularly in terms of robustness and efficiency under data contamination.

Related Works

Convex Clustering and Semi-definite Programming: Since the introduction of Convex Clustering by (Pelckmans et al. 2005b), various extensions and perspectives have been explored (Lindsten, Ohlsson, and Ljung 2011b), (Hocking et al. 2011b), (Zhu et al. 2014). Pelckmans and De Moor introduced a shrinkage term to induce sparsity between centroids, enabling hierarchical clustering by tuning the trade-off parameter. (Hocking et al. 2011a) propose a convex relaxation-based clustering algorithm that efficiently traces a regularization path, achieves state-of-the-art performance on non-convex clusters, and simultaneously infers a hierarchical tree structure from the data. In (Chen, Zhou, and

Ye 2011), two optimization approaches — ADMM and a variant of AMA were introduced to solve convex clustering problems for practical applications. Additionally, convex relaxations of the $k-$Means problem via Semi-Definite Programming (SDP) have been developed (Peng and Wei 2007; Awasthi et al. 2015; Mixon, Villar, and Ward 2017), replacing the $k-$means objective with a trace-based formulation. (Mixon, Villar, and Ward 2017) further showed that this SDP relaxation achieves perfect recovery with high probability under the stochastic unit-ball model in $\mathbb{R}^d$, given mild regularity conditions.

Robustness and Feature Selection: Robustness to outliers is essential for ensuring learning algorithms remain stable under adversarial or noisy conditions. To address this, (Gong, Ye, and Zhang 2012) proposed a Robust Multi-Task Feature Learning model that not only identifies shared features across tasks but also detects outlier tasks. Similarly, (Chen, Zhou, and Ye 2011) introduced a robust multi-task learning framework combining a low-rank structure for related tasks and a sparse group structure to isolate outlier tasks.

Median of Means based Clustering: The Median of Means (MoM) estimator provides a robust and efficient framework for mean estimation with strong theoretical guarantees. (Brunet-Saumard, Genetay, and Saumard 2022) introduced a bootstrap-based MoM method, forming blocks with replacement, which improves the breakdown point over standard MoM when enough blocks are used. In the context of interpretable clustering, (Moshkovitz et al. 2020) proposed a method using small decision trees to partition data, enabling clear cluster characterization. They further analyzed whether such tree-induced clusterings can match the cost of optimal unconstrained clustering and how to compute them efficiently.

Proposed Method

In this section, we outline our proposed clustering technique based on the median of means estimate in sufficient detail. We also include an Adam-based gradient descent method to optimize our non-convex objective function effectively.

Let $\mathbf{X}_{n\times d} \in \mathbb{R}^{n\times d}$ be the data matrix, where each row $\{\mathbf{x}_i\}_{i=1}^n$ is a data point, and $\mathbf{x}_i \in \mathbb{R}^d$ for each $i \in \{1, 2, ..., n\}$. Let $\mathbf{u}_i$ be the agent corresponding to point $\mathbf{x}_i$ $\forall i \in \{1, 2, \cdots, n\}$, and we define $\mathbf{U}_{n\times d} \in \mathbb{R}^{n\times d}$ as the agent matrix.

One of the most challenging problems in convex clustering is assigning weights to every pair of neighbours based on certain similarity measures. This enforces a restriction on its performance in high dimensions, which depends heavily on the choice of the pairwise similarities, although most people follow a $k-$nearest neighbour-based approach (Chi and Lange 2015a), coupled with a Gaussian similarity measure:

$$w_{ij} = \mathbb{1}_{ij,k} e^{-\phi||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}, \tag{4}$$

where $\mathbb{1}_{ij,k} = 1$ if x_i is one of the k-nearest neighbours of x_j with respect to the $\|\cdot\|_2$ and 0 otherwise. Here, ϕ represents the bandwidth of the Gaussian kernel, and smaller values of ϕ indicate greater similarities between the two nodes. Arbitrary choices of ϕ can lead to poor cluster generation, formation of arbitrary clusters, or even collapse of all cluster centroids to a global centroid (Hocking et al. 2011a), hindering the overall effectiveness of the $k-$nearest neighbour-based heuristics.

Next, we introduce a Random Binning strategy to partition the dataset before minimizing a non-convex objective function. This class of Random Binning (RB) techniques was originally proposed in (Rahimi and Recht 2007) and subsequently revisited in (Wu et al. 2016), where it was shown to yield faster convergence than other Random Features methods when scaling large-scale kernel machines. Although these previous approaches typically employ a parametrized feature map (Wu et al. 2018), incorporating both bin widths and offsets, our method adopts a simplified variant of this strategy - designed specifically to randomly partition the dataset into $\mathcal{O}(n)$ number of bins, each containing a fixed number of samples drawn from the observables. Formally, we partition the index set $1, 2, \cdots, n$ into $l = \mathcal{O}(n)$ subsets, denoted by $B = \{B_i\}_{i=1}^l$, where each B_i contains exactly $b(= \lfloor \frac{n}{l} \rfloor)$ elements. If n is not divisible by l, a small number of elements are discarded to maintain uniform bin sizes across all partitions.

Henceforth, we define the "contribution" of point $\mathbf{x}_r$ in a "convex" type cost function as

$$f_U(\mathbf{x}_r) = \frac{1}{2}||\mathbf{x}_r - \mathbf{u}_r||_2^2 + \frac{\gamma}{2}\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2. \tag{5}$$

By our aforementioned MoM framework, instead of directly minimizing $\frac{1}{n}\sum_{r=1}^{n} f_{\mathbf{U}}(\mathbf{x}_r)$, we aim to minimize an objective function of the form

$$C(\mathbf{U}) = Median\left(\left\{\frac{1}{b}\sum_{r\in B_j} f_{\mathbf{U}}(\mathbf{x}_r)\right\}_{j=1}^{l}\right). \tag{6}$$

Noting that the second term in (5) is independent of r, we define $l_t \in \{1, 2, \cdots, l\}$ such that

$$\begin{aligned} MoM_B(\mathbf{U}) &:= Median\left(\left\{\frac{1}{2b}\sum_{i\in B_j}||\mathbf{x}_i - \mathbf{u}_i||_2^2\right\}_{j=1}^{l}\right) \\ &= \frac{1}{2b}\sum_{i\in B_{l_t}}||\mathbf{x}_i - \mathbf{u}_i||_2^2. \end{aligned} \tag{7}$$

Next, we rewrite the cost function as

$$C(\mathbf{U}) = MoM_B(\mathbf{U}) + \frac{\gamma}{2}\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2. \tag{8}$$

Before initiating the optimization procedure of the cost function in (8), we will involve another robustness criterion to make our objective function more stable from outliers: we use $\sum_{i,j} w_{ij}\min\left(\mu, ||\mathbf{u}_i - \mathbf{u}_j||_2^2\right)$ instead of $\sum_{i,j} w_{ij}||\mathbf{u}_i -$

$\mathbf{u}_j||_2^2$. By clipping the maximum pairwise distances by another hyperparameter μ, we can significantly remove the effect of such outliers or other distant clusters.

Now, we are in a position to write down the final cost function, which is

$$C(\mathbf{U}) = MoM_B(\mathbf{U}) + \frac{\gamma}{2}\sum_{i,j} w_{ij} \min\left\{\mu, ||\mathbf{u}_i - \mathbf{u}_j||_2^2\right\}. \tag{9}$$

Due to the non-convex nature of this objective function, we use the ADAM gradient descent algorithm (Kingma and Ba 2014) to minimize it. The gradient of $C(\mathbf{U})$ with respect to $\mathbf{u}_i$ is

$$\begin{aligned} g_i := \frac{\partial C(\mathbf{U})}{\partial \mathbf{u}_i} =& \frac{1}{b}(\mathbf{u}_i - \mathbf{x}_i)\mathbb{1}(i \in B_{l_t}) \\ &+ \gamma \sum_j w_{ij}(\mathbf{u}_i - \mathbf{u}_j)\mathbb{1}(||\mathbf{u}_i - \mathbf{u}_j||_2^2 < \mu). \end{aligned} \tag{10}$$

After N iterations, we construct a graph with $\{\mathbf{u}_i\}_{i=1}^n$ as vertices and where $\mathbf{u}_i$ and $\mathbf{u}_j$ are adjacent if $||\mathbf{u}_i - \mathbf{u}_j|| < \eta_1$. The tuning parameter $\eta_1 \in (0.001, 0.1)$ adapts to data and desired cluster count, ensuring robustness. We assign each connected component of this graph as a cluster and combine all clusters with less than half the average cluster size into a single cluster, marking this combined cluster as noise.

Theoretical Properties

This section establishes the theoretical properties of the (global) optimal solutions of the proposed objective function. We also analyse computational complexity and discuss the convergence properties of our method.

Finite Sample Error Bounds and Weak Consistency

We begin our statistical analysis of COMET by providing finite sample error bounds on the prediction error, using the widely used Hanson-Wright inequalities, especially the recent uniform versions by (Bousquet, Klochkov, and Zhivotovskiy 2020). These bounds provide sufficient conditions for the consistency of the centroid and weight estimates.

Recall the objective function (8),

$$\min_U \left\{ \frac{1}{2b}\sum_{i\in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i||_2^2 + \frac{\gamma}{2}\sum_{i,j} w_{ij}||\mathbf{u}_i - \mathbf{u}_j||_2^2 \right\},$$

where l_t is such that $\frac{1}{2b}\sum_{i\in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i||_2^2 = Median\left(\left\{\frac{1}{2b}\sum_{i\in B_j} ||\mathbf{x}_i - \mathbf{u}_i||_2^2\right\}_{j=1}^{l}\right), l_t \in \{1,\ldots,l\}$.

Let $\boldsymbol{x} = vec(X)$ and $\boldsymbol{u} = vec(U)$, where $vec(\cdot)$ means to vectorize a matrix by appending its columns together. So, $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^{nd}$ and $\boldsymbol{x}_{d(i-1)+j} = X_{ij}$, $\boldsymbol{u}_{d(i-1)+j} = U_{ij}$.
Consider $\boldsymbol{I}_{B_{l_t}}$ to be an $nd \times nd$ diagonal matrix with i-th diagonal element = 1 if $bd \le i < (b+1)d$ where $b \in B_{l_t}$ and all other elements 0. So, we can write

$$\sum_{i\in B_{l_t}} ||\mathbf{x_i} - \mathbf{u_i}||_2^2 = (\boldsymbol{x} - \boldsymbol{u})^\top I_{B_{l_t}}(\boldsymbol{x} - \boldsymbol{u}).$$

Algorithm 1: COMET : Convex Clustering with Median of Mean Estimator and Adam Optimization

Input: Data $\{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$
Hyperparameters: $N, k, \phi, \gamma, \mu, \eta_1$
Output: Cluster assignment $\{Z_i\}_{i=1}^n$ where $Z_i \in \mathbb{N}$

1: Construct a $k-$NN graph on $\{\mathbf{x}_i\}_{i=1}^n$ and assign $w_{ij} = e^{-\phi||\mathbf{x}_i - \mathbf{x}_j||_2^2}$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are adjacent, $w_{ij} = 0$ otherwise
2: Initialize $\mathbf{m}_i^{(0)} = 0$, $\mathbf{v}_i^{(0)} = 0$ and $\mathbf{u}_i^{(0)} = \mathbf{x}_i$
3: **for** $t = 0$ to $N-1$ **do**
4: Construct a partition, $B = \{B_i\}_{i=1}^l$, of $\{1, 2, \cdots, n\}$ into l bins each of size b
5: Find $B_{l_t} \in B$ such that $MoM_B(\mathbf{U}^{(t)}) = \frac{1}{2b}\sum_{i\in B_{l_t}} ||\mathbf{x}_i - \mathbf{u}_i^{(t)}||_2^2$
6: $\mathbf{g}_i^{(t)} = \frac{1}{b}(\mathbf{u}_i^{(t)} - \mathbf{x}_i)\mathbb{1}(i \in B_{l_t}) + \gamma\sum_j w_{ij}(\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)})\mathbb{1}(||\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}||_2^2 < \mu)$
7: $\mathbf{m}_i^{(t)} = \boldsymbol{\beta}_1\mathbf{m}_i^{(t-1)} + (1-\boldsymbol{\beta}_1)\mathbf{g}_i^{(t)}$
8: $\mathbf{v}_i^{(t)} = \boldsymbol{\beta}_2\mathbf{v}_i^{(t-1)} + (1-\boldsymbol{\beta}_2)\left(\mathbf{g}_i^{(t)} \odot \mathbf{g}_i^{(t)}\right)$
9: Calculate $\mathbf{u}_i^{(t+1)}$ from $\mathbf{u}_i^{(t)}$ with the help of $\hat{\mathbf{m}}_i^{(t)}$ and $\hat{\mathbf{v}}_i^{(t)}$ using the ADAM update rule. Refer to the supplementary material (A.1) for the exact update rule.
10: **end for**
11: Construct a graph on $\left\{\mathbf{u}_i^{(N)}\right\}_{i=1}^n$ where $\mathbf{u}_i$ and $\mathbf{u}_j$ are adjacent if $||\mathbf{u}_i - \mathbf{u}_j|| < \eta_1$
12: Assign each connected component of this graph as a cluster
13: Combine all clusters with less than half the average cluster size into a single cluster and mark this combined cluster as noise

For notational simplicity, we write $\|\boldsymbol{y}\|_{\boldsymbol{A}}^2 = \boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{y}$, for any positive semidefinite matrix $\boldsymbol{A}$.
Also note that w_{ij}'s remain fixed in each iteration of the algorithm. Since w_{ij}'s are either 0 or < 1, we work with an upper bound of the cost function, where each w_{ij} is replaced by $w'_{ij} = \mathbb{1}(w_{ij} > 0)$. Let $D^{n(n-1)d\times nd}$ be such that $\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u} = \mathbf{u}_i - \mathbf{u}_j$, where $\mathcal{C}(i,j)$ is an index set: then the objective function can be written as

$$\min\left\{\frac{1}{2b}\|\boldsymbol{x} - \boldsymbol{u}\|_{I_{B_{l_t}}}^2 + \frac{\gamma}{2}\sum_{(i,j)\in\mathcal{E}} \|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\|_2^2\right\}, \tag{11}$$

where $\mathcal{E} \subseteq \{(i,j) : i,j \in \{1,2,\ldots,n\}\}$ is an index set. We will assume the model $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{\epsilon}$, where $\epsilon \in \mathbb{R}^{nd}$ is a vector of independent noise variables and $\mathbb{E}(\epsilon) = 0$. This model is fairly standard for analysing the large-sample behaviour of convex clustering methods (Tan and Witten 2015);(Wang et al. 2018). For all practical purposes, one may assume that the error terms are almost surely bounded, that is, for some $M > 0$, $|\epsilon_i| \le M$ for all $i = 1,\ldots,nd$.

The goal of this analysis is to find probabilistic bounds on $\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}}$, where $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{I}}_{B_{l_t}}$ are obtained by minimising the objective function in (11).

Theorem 1. *Suppose the model behaves as $\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^{nd}$ is a vector of independent bounded random variables, with mean 0, covariance matrix $\sigma^2 \boldsymbol{I}_{nd\times nd}$ and $|\epsilon_i| \leq M$, for all $i = 1, \ldots, nd$. Further assume that $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{I}}_{B_{l_t}}$ are obtained from minimizing* (11)*, then if $\gamma' \geq \frac{M}{ndb\sqrt{n}}$ the following holds with probability at least 1 - δ,*

$$\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \leq M^2\left(\frac{\sqrt{db}+d\sqrt{n}}{n\sqrt{db}}\right)$$
$$+M^2\left(\frac{c}{b\sqrt{nd}}\sqrt{\log\left(\frac{1}{\delta}\right)} + \frac{c\log\left(\frac{1}{\delta}\right)}{ndb}\right) + \gamma'\frac{|\mathcal{E}|}{4} \quad (12)$$
$$+\gamma'\left[\sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 + \sum_{(i,j)\in\mathcal{E}}\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2^2\right].$$

The proof of this theorem is deferred to the supplemntary material (A.2). Bounded noise coomonly arises in practice, e.g., in qunatized or range-limited measurements, normalized or compact feature spaces with $|x| \leq 1$. noise is inherently bounded and is standard in robust optimization. From this theorem, we also arrive at the following two corollaries: Corollary 1 addresses the convergence of the centroid estimates under a minimum constraint on the hyperparameter, for the number of features being small enough with respect to the number of sample points, while Corollary 2 elaborates on the rate of convergence of those estimates under the constraint on the hyperparameter only.

Corollary 1. *Suppose $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, for all $1 \leq i, j \leq n$, for some constant C, $|\mathcal{E}| \leq kn$ and $\gamma' \geq \frac{M}{ndb\sqrt{n}}$. If $d = o(n)$, then $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} \xrightarrow{p} 0$ as $n, d \to \infty$.*

Corollary 2. *Suppose $\left\|\boldsymbol{D}_{\mathcal{C}(i,j)}\boldsymbol{u}\right\|_2 \leq C$, for all $1 \leq i, j \leq n$, for some constant C, $|\mathcal{E}| \leq kn$ and $\gamma' \geq \frac{M}{ndb\sqrt{n}}$. Then $\frac{1}{2ndb}\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\hat{\boldsymbol{I}}_{B_{l_t}}} = O\left(\frac{1}{\sqrt{n}}\right)$.*

We lay out the complete proofs of Corollary 1 and Corollary 2 in the supplementary material (A.3) and (A.4), respectively.

Computational Complexity and Other Competing Methods

We compare the efficiency of our algorithm with other popular algorithms such as Convex Clustering ((Chi and Lange 2015b)), Robust Continuous Clustering ((Shah and Koltun 2017)), and Robust Convex Clustering ((Wang et al. 2016)). For a detailed explanation, refer to supplementary material (A.5).

From the above table, it is clear that COMET is better or at least on par in terms of computational cost with recent and most widely used robust clustering algorithms.

Algorithm	Complexity
COMET	$\mathcal{O}(Nnkd)$
Convex-Clustering	$\mathcal{O}(N(n^2d + d\epsilon))$
Robust Continuous Clustering	$\mathcal{O}(N(n^2d + nkd)$
Robust Convex Clustering	$\mathcal{O}(Nnkd)$

Table 1: Comparison of Runtime Complexity with other SOTA methods

Experiments and Results

In this section, we demonstrate the superiority of our proposed algorithm, COMET, over different variants of existing clustering algorithms, including both real and simulated datasets. The description of the datasets is given in the supplementary material (A.6). For simulated datasets, the generation procedure is described later.

Algorithms under consideration

We consider the following well-known clustering algorithms to assess the effectiveness of COMET: $k-$means (KM)(Hartigan and Wong 1979), Convex Clustering (CC) (Chi and Lange 2015b), MoM $k-$means (MKM) (Paul et al. 2021), Robust Convex Clustering (RConv) (Wang et al. 2016), Robust Continuous Clustering (RCC) (Shah and Koltun 2017) and Robust Bregman $k-$means (RBKM) (Brécheteau, Fischer, and Levrard 2021).

Performance Measures

For evaluating our proposed COMET algorithm against competing methods, we adopt the following metrics and resources:

- **Evaluation Metrics:** Since ground-truth cluster labels are available for all real and simulated datasets, we evaluate clustering performance using
  - **Adjusted Rand Index (ARI)**
  - **Adjusted Mutual Information (AMI)**

  Both metrics provide robust comparisons across algorithms.
- **Estimated Number of Clusters:** We also report the average number of clusters estimated by each algorithm to further assess performance.

Experimental Set-up

We apply all the selected algorithms on the datasets listed in the supplementary material (A.6). Our main goal is to make a proper comparison of robustness of these algorithms to the presence of noise and outliers in the data. We artificially add different levels of noise and outliers to the datasets under study and record the performances of the algorithms.

To add noise of level $p\%$ to a dataset, we first consider the smallest axis-parallel hypercube containing the whole original dataset. Then, we simulate $\lfloor \frac{np}{100} \rfloor$ points uniformly from the hypercube and add them to the original dataset, labeled as "noise". All the algorithms are run on this modified dataset, and the ARI/AMI is calculated based on the obtained cluster labels of the original data points only. We vary

Dataset	Index	KM	MKM	CC	RCC	RConv	RBKM	COMET
Newthyroid (k = 3)	k^*	3.08±1.28	2.94±1.38	14.14±1.23	212.13±3.36	3.79±0.58	2.00±0.00	4.14±0.36
	ARI	0.34±0.21†	0.40±0.26†	0.69±0.04†	0.00±0.00†	0.81±0.21†	0.11±0.03†	**0.97±0.01**
	AMI	0.34±0.19†	0.39±0.25†	0.52±0.03†	0.003±0.004†	0.77±0.16†	0.08±0.03†	**0.90±0.02**
Wisconsin (k = 2)	k^*	2.25±0.63	2.00±0.82	15±1.86	477±9.06	2.00±1.04	2.00±0.00	3.00±0.00
	ARI	0.52±0.35†	0.47±0.39†	0.81±0.01†	0.01±0.00†	0.85±0.03$^\sim$	0.15±0.06†	**0.87±0.01**
	AMI	0.48±0.31†	0.41±0.34†	0.67±0.01†	0.07±0.003†	0.75±0.03†	0.19±0.05†	**0.76±0.01**
Wine (k = 3)	k^*	3.14±1.18	3.17±1.35	25.29±2.16	178±0.00	2.43±0.51	2.00±0.00	4.64±0.84
	ARI	0.66±0.31$^\sim$	0.59±0.29†	0.59±0.15†	0.0±0.0†	0.22±0.28†	0.01±0.02†	**0.79±0.15**
	AMI	0.67±0.29$^\sim$	0.61±0.28†	0.59±0.10†	0.00±0.00†	0.32±0.31†	0.04±0.03†	**0.80±0.09**
Dermatology (k = 6)	k^*	5.16±1.93	4.77±2.09	4.00±0.00	358±0.00	5.00±0.00	2.00±0.00	5.85±0.53
	ARI	0.61±0.17†	0.56±0.17†	0.21±0.00†	0.00±0.00†	0.66±0.01†	0.004±0.02†	**0.81±0.06**
	AMI	0.78±0.10†	0.73±0.13†	0.44±0.00†	0.00±0.00†	0.79±0.01†	0.04±0.04†	**0.86±0.04**
Lung-Discrete (k = 7)	k^*	5.91±1.57	5.23±1.39	2.36±0.63	14.9±16.8	9.79±0.43	2±0.00	9.21±0.80
	ARI	0.44±0.09†	0.50±0.10†	0.07±0.03†	0.41±0.12†	0.39±0.05†	0.01±0.01†	**0.71±0.02**
	AMI	0.53±0.07†	0.58±0.08†	0.20±0.07†	0.51±0.15†	0.51±0.04†	0.07±0.04†	**0.69±0.01**
ORLRaws10p (k = 10)	k^*	4.85±1.83	4.89±1.72	42±0.00	100±0.00	16±0.00	2±0.00	14±0.00
	ARI	0.33±0.11†	0.33±0.10†	0.53±0.00†	0.00±0.00†	0.54±0.002†	0.02±0.01†	**0.73±0.00**
	AMI	0.58±0.11†	0.61±0.09†	0.61±0.00†	0.00±0.00†	0.69±0.001†	0.11±0.03†	**0.81±0.00**

† : significantly different from the best performing algorithm, $^\sim$: statistically similar to the best performing algorithm .

Table 2: Results for Real-life Datasets

p to observe the change in performance of the algorithms with the introduction of noise. $k-$means, MoM $k-$means and Robust Bregman $k-$means require the exact number of clusters to be given as input, but that gives these algorithms an unfair advantage considering Convex Clustering, Robust Convex Clustering, Robust Continuous Clustering as well as COMET determine the number clusters automatically. Hence, to ensure a fair comparison, we used *Gapstat* ((Tibshirani, Walther, and Hastie 2001)) in those three algorithms to get an estimate of the number of clusters from the data itself and used that value for the clustering. We run all algorithms according to the recommended specification of hyperparameters or tune them to achieve maximum ARI. $k-$means, MoM $k-$means, and RBKM are run till there is no further update in the cluster assignment matrix. Convex Clustering and Robust Convex Clustering were run on each dataset after tuning its hyperparameters for 150 epochs. RCC is run according to the hyperparameter recommendations and termination condition specified in (Shah and Koltun 2017). $k-$means, MoM $k-$means and Robust Bregman $k-$means are dependent on the choice of initial centroids, each of them were run 25 times for every noise level and the mean performance is reported with their standard deviation. The random noise was added to the data using numpy.random.default_rng(0) in numpy library of python 3 (ipykernel).

We perform the experiments for both generated and real-life datasets. In the next section we will focus on the results for real-life datasets. **For the detailed study on generated datasets refer to the supplementary material (A.7)**

Real-Life Datasets

Here we show the clustering results of our algorithm COMET and other selected algorithms on some of the real-life datasets with 10% noise. For the results on other datasets refer to the supplementary material (A.8). Here, k^* refers to an estimated number of clusters. The actual number of clusters is indicated as k. Here the standard deviation is that of the performance measure, not of the mean statistic.

Discussion Table 2 shows that COMET outperforms other algorithms, achieving nearly accurate cluster numbers with low standard deviation across most datasets. However, in datasets like ORLRaws10P (Table 2), Brain, and Wisconsin (present in the supplementary material (A.8)), the detected cluster size slightly deviates from the actual value, likely due to limitations in the $k-$NN graph structure. This issue is more pronounced in other algorithms. Despite this, COMET still provides better clustering patterns, with higher ARI and AMI than the others.

Significance of Our Results: Wilcoxon-Rank Sum Test

For various datasets, we want to test if the ARI and AMI produced by our algorithm are "significantly higher" than other selected clustering algorithms. We use Wilcoxon-Rank Sum test for this purpose. Refer to the supplementary material (A.11) for detailed discussion related to this.

Case Study on Brain dataset

We evaluate our algorithm's performance on the Brain dataset, a Microarray dataset with 42 instances (brain tumor patients) and 5597 features. The dataset includes 5 categories: 10 medulloblastomas, 10 malignant gliomas, 10 AT/RT, 4 normal cerebellums, and 8 supratentorial PNETs, as described in (Pomeroy et al. 2002).

We compare the algorithms under varying noise levels (0%, 5%, 10%, 15%, 20%) using the procedure outlined earlier, without applying any feature reduction methods like PCA. The results, shown in Table 3 and Figure 2, show

Index	Noise(%)	KM	MKM	CC	RCC	RConv	RBKM	COMET
ARI	0	0.28±0.10†	0.23±0.11†	0.64±0.00†	0.00±0.00†	0.56±0.00†	0.01±0.01†	**0.65±0.00**
	5	0.31±0.13†	0.31±0.13†	0.64±0.00†	0.00±0.00†	0.56±0.01†	0.01±0.01†	**0.66±0.00**
	10	0.26±0.10†	0.26±0.10†	0.64±0.02$^{\sim}$	0.00±0.00†	0.56±0.06†	0.016±0.02†	**0.66±0.03**
	15	0.22±0.09†	0.10±0.08†	0.63±0.02$^{\sim}$	0.00±0.00†	0.55±0.06†	0.02±0.02†	**0.66±0.03**
	20	0.19±0.11†	0.08±0.07†	0.63±0.04†	0.00±0.00†	0.63±0.03†	0.02±0.02†	**0.65±0.02**
AMI	0	0.35±0.10†	0.32±0.10†	0.62±0.00†	0.00±0.00†	0.62±0.00†	0.017±0.01†	**0.67±0.00**
	5	0.38±0.13†	0.28±0.14†	0.62±0.00†	0.00±0.00†	0.62±0.01†	0.02±0.02†	**0.72±0.00**
	10	0.33±0.10†	0.27±0.11†	0.62±0.03†	0.00±0.00†	0.62±0.05†	0.03±0.04†	**0.72±0.03**
	15	0.29±0.11†	0.18±0.12†	0.62±0.01†	0.00±0.00†	0.62±0.04†	0.03±0.04†	**0.72±0.02**
	20	0.27±0.13†	0.15±0.12†	0.62±0.01†	0.00±0.00†	0.62±0.05†	0.03±0.05†	**0.72±0.03**
k^*	0	5±1.92	5±1.50	18±0.00	42±0.00	5±0.00	2±0.00	**5±0.00**
	5	5±1.85	5±1.46	18±0.00	42±0.00	5±0.00	2±0.00	**4±0.00**
	10	5±1.92	5±1.49	18±0.00	42±0.00	5±0.36	2±0.50	**4±0.00**
	15	4±1.90	4±1.41	18±0.00	42±0.00	5.1±0.22	2±0.50	**4±0.00**
	20	3±1.45	3±1.42	18±0.00	42±0.00	18±0.52	2±0.5	**4±0.00**

† : significantly different from the best performing algorithm, $^{\sim}$: statistically similar to the best performing algorithm .

Table 3: Performance of Different Algorithms on **Brain** on Different Noise Levels

that COMET consistently outperforms all other algorithms, maintaining an ARI above 0.6 across all noise levels. Here, the standard deviation is that of the performance measure, not of the mean statistic. Convex clustering and Robust Convex clustering perform well but are outpaced by COMET. $k-$means and MoM $k-$means show poor results, worsening with increased noise. RCC and RBKM perform very poorly, as reflected in the results. The *t*-SNE plots for the clustering results for various algorithms on this dataset are provided in the supplementary material (A.10).

Also, refer to the supplementary material (A.9) for a case study on the Wisconsin dataset.

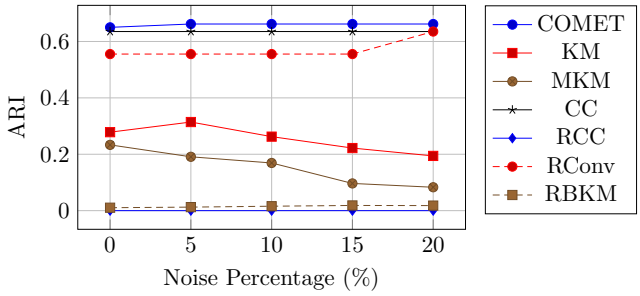


Figure 2: Line plot for performance of different algorithms on **Brain** dataset

Ablation Study

A sensitivity analysis of the performance of our algorithm for the hyperparameter γ, μ and k are illustrated in the supplementary material (A.12). For a detailed illustration of the tuning process of our hyperparameters on the **Wisconsin Breast Cancer** dataset, refer to the supplementary material (A.12). Another ablation study on the **NewThyroid** Dataset is discussed in the supplementary material (A.12).

Limitations

While we tested our algorithm on specific distributions of noise for clustering synthetic data, a systematic method to choose an appropriate cost function needs to be developed. However, it is still obscure to us how the cost function can be modified in case the noise follows some definite pattern and is not randomly distributed. Also, in our proof of theoretical consistency 1, we assumed $d = o(n)$. However, modifications to the clustering procedure need to be done for higher-dimensional datasets. Overall, given the flexibility of our clustering framework, other possibilities can be explored by incorporating different methods for different steps. One may further explore to relax assumptions on the errors 1 for consistency in a more general settings.

Conclusion

In this study, we introduce a robust and interpretable clustering framework designed for multivariate datasets affected by noise. Our method reformulates the underlying cost function to reduce the adverse effects of random noise that often undermine conventional convex clustering techniques. We provide rigorous theoretical guarantees by establishing both the consistency and the convergence rate of the proposed estimators. Furthermore, extensive empirical evaluations—including experiments on diverse real-world datasets, detailed case studies, and ablation studies—demonstrate the practical effectiveness and reliability of our approach.

References

Awasthi, P.; Bandeira, A. S.; Charikar, M.; Krishnaswamy, R.; Villar, S.; and Ward, R. 2015. Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, 191–200.

Bartlett, P.; Boucheron, S.; and Lugosi, G. 2002. Model Selection and Error Estimation. *Machine Learning*, 48: 85–113.

Bousquet, O.; Klochkov, Y.; and Zhivotovskiy, N. 2020. Sharper Bounds for Uniformly Stable Algorithms. In Abernethy, J.; and Agarwal, S., eds., *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, 610–626. PMLR.

Brunet-Saumard, C.; Genetay, E.; and Saumard, A. 2022. K-bMOM: A robust Lloyd-type clustering algorithm based on bootstrap median-of-means. *Computational Statistics & Data Analysis*, 167: 107370.

Brécheteau, C.; Fischer, A.; and Levrard, C. 2021. Robust Bregman clustering. *The Annals of Statistics*, 49(3).

Chakraborty, S.; and Das, S. 2022. Detecting Meaningful Clusters From High-Dimensional Data: A Strongly Consistent Sparse Center-Based Clustering Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 2894–2908.

Chen, J.; Zhou, J.; and Ye, J. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 42–50.

Chi, E. C.; and Lange, K. 2015a. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4): 994–1013.

Chi, E. C.; and Lange, K. 2015b. Splitting Methods for Convex Clustering. *Journal of Computational and Graphical Statistics*, 24(4): 994–1013.

Chi, E. C.; and Steinerberger, S. 2019. Recovering trees with convex clustering. *SIAM Journal on Mathematics of Data Science*, 1(3): 383–407.

Coleman, G. B.; and Andrews, H. C. 1979. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5): 773–785.

De Amorim, R. C. 2016. A Survey on Feature Weighting Based K-Means Algorithms. *Journal of Classification*, 33(2): 210–242.

Feng, Q.; Chen, C. P.; and Liu, L. 2023. A review of convex clustering from multiple perspectives: models, optimizations, statistical properties, applications, and connections. *IEEE Transactions on Neural Networks and Learning Systems*.

Gong, P.; Ye, J.; and Zhang, C. 2012. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 895–903.

Guillaume, L.; and Matthieu, L. 2017. Learning from MOM's principles: Le Cam's approach. arXiv:1701.01961.

Hamerly, G.; and Elkan, C. 2004. Learning the K in K-Means. *Advances in Neural Information Processing Systems*, 17.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28(1): 100.

Hocking, T.; Vert, J.-P.; Bach, F.; and Joulin, A. 2011a. Clusterpath An Algorithm for Clustering using Convex Fusion Penalties. In *Clustering Algorithm for Convex Fusion Penalties*, 745–752.

Hocking, T. D.; Joulin, A.; Bach, F.; and Vert, J.-P. 2011b. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, 1.

Jain, A. K. 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8): 651–666.

Kansal, T.; Bahuguna, S.; Singh, V.; and Choudhury, T. 2018. Customer segmentation using K-means clustering. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, 135–139. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laforgue, P.; Clemencon, S.; and Bertail, P. 2019. On Medians of (Randomized) Pairwise Means. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 1272–1281. PMLR.

Lecué, G.; and Lerasle, M. 2020. Robust machine learning by median-of-means: Theory and practice. *The Annals of Statistics*, 48(2): 906 – 931.

Lerasle, M. 2019. Lecture notes: Selected topics on robust statistical learning theory. *arXiv preprint arXiv:1908.10761*.

Lindsten, F.; Ohlsson, H.; and Ljung, L. 2011a. Clustering Using Sum-Of-Norms Regularization; with Application to Particle Filter Output Computation. In *IEEE Workshop on Statistical Signal Processing Proceedings*.

Lindsten, F.; Ohlsson, H.; and Ljung, L. 2011b. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, 201–204. IEEE.

Lu, Y.; and Zhou, H. H. 2016. Statistical and computational guarantees of lloyd's algorithm and its variants. *arXiv preprint arXiv:1612.02099*.

Lugosi, G.; and Mendelson, S. 2017. Regularization, sparse recovery, and median-of-means tournaments. *Bernoulli*, 25.

Mixon, D. G.; Villar, S.; and Ward, R. 2017. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4): 389–415.

Moshkovitz, M.; Dasgupta, S.; Rashtchian, C.; and Frost, N. 2020. Explainable k-means and k-medians clustering. In *International conference on machine learning*, 7055–7065. PMLR.

Münz, G.; Li, S.; and Carle, G. 2007. Traffic anomaly detection using k-means clustering. In *Gi/itg workshop mmbnet*, volume 7.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2013. The effectiveness of lloyd-type methods for the k-means problem. *J. ACM*, 59(6).

Paul, D.; Chakraborty, S.; Das, S.; and Xu, J. 2021. Uniform Concentration Bounds toward a Unified Framework for Robust Clustering. ArXiv:2110.14148 [cs, math, stat].

Pelckmans, K.; Brabanter, J. D.; Moor, B. D.; and Suykens, J. A. K. 2005a. Convex Clustering Shrinkage. In *Convex Clustering Shrinkage*.

Pelckmans, K.; De Brabanter, J.; Suykens, J. A.; and De Moor, B. 2005b. Convex clustering shrinkage. In *PASCAL workshop on statistics and optimization of clustering workshop*, volume 1524.

Peng, J.; and Wei, Y. 2007. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1): 186–205.

Pomeroy, S. L.; Tamayo, P.; Gaasenbeek, M.; Sturla, L. M.; Angelo, M.; McLaughlin, M. E.; Kim, J. Y. H.; Goumnerova, L. C.; Black, P. M.; Lau, C.; Allen, J. C.; Zagzag, D.; Olson, J. M.; Curran, T.; Wetmore, C.; Biegel, J. A.; Poggio, T.; Mukherjee, S.; Rifkin, R.; Califano, A.; Stolovitzky, G.; Louis, D. N.; Mesirov, J. P.; Lander, E. S.; and Golub, T. R. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870): 436–442.

Radchenko, P.; and Mukherjee, G. 2017. Convex clustering via l 1 fusion penalization. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(5): 1527–1546.

Rahimi, A.; and Recht, B. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

Rodriguez, D.; and Valdora, M. 2019. The breakdown point of the median of means tournament. *Statistics & Probability Letters*, 153: 108–112.

Shah, S. A.; and Koltun, V. 2017. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37): 9814–9819.

Tan, K. M.; and Witten, D. 2015. Statistical Properties of Convex Clustering.

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.

Tropp, J. 2006. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3): 1030–1051.

Wang, B.; Zhang, Y.; Sun, W. W.; and Fang, Y. 2018. Sparse Convex Clustering. *Journal of Computational and Graphical Statistics*, 27(2): 393–403.

Wang, Q.; Gong, P.; Chang, S.; Huang, T. S.; and Zhou, J. 2016. Robust Convex Clustering Analysis. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1263–1268. Barcelona, Spain: IEEE. ISBN 9781509054732.

Witten, D. M.; and Tibshirani, R. 2010. A Framework for Feature Selection in Clustering. *Journal of the American Statistical Association*, 105(490): 713–726.

Wu, L.; Chen, P.-Y.; Yen, I. E.-H.; Xu, F.; Xia, Y.; and Aggarwal, C. 2018. Scalable spectral clustering using random binning features. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2506–2515.

Wu, L.; Yen, I. E.; Chen, J.; and Yan, R. 2016. Revisiting random binning features: Fast convergence and strong parallelizability. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1265–1274.

Xu, J.; and Lange, K. 2019. Power k-Means Clustering. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6921–6931. PMLR.

Zhu, C.; Xu, H.; Leng, C.; and Yan, S. 2014. Convex optimization procedure for clustering: Theoretical revisit. *Advances in Neural Information Processing Systems*, 27.