The Sparse MinMax k-Means Algorithm for High-Dimensional Clustering

Sayak Dey[1] , **Swagatam Das**[2] * and **Rammohan Mallipeddi**[3]
[1]Samsung Research Institute, Bangalore, 560037, India
[2]Indian Statistical Institute, Kolkata, 700108, India
[3]Kyungpook National University, Daegu, 41566, Republic of Korea
sayak.d@samsung.com, swagatam.das@isical.ac.in, mallipeddi.ram@gmail.com

Abstract

Classical clustering methods usually face tough challenges when we have a larger set of features compared to the number of items to be partitioned. We propose a Sparse MinMax k-Means Clustering approach by reformulating the objective of the MinMax k-Means algorithm (a variation of classical k-Means that minimizes the *maximum* intra-cluster variance instead of the *sum* of intra-cluster variances), into a new weighted *between-cluster sum of squares* ($BCSS$) form. We impose sparse regularization on these weights to make it suitable for high-dimensional clustering. We seek to use the advantages of the MinMax k-Means algorithm in the high-dimensional space to generate good quality clusters. The efficacy of the proposal is showcased through comparison against a few representative clustering methods over several real world datasets.

1 Introduction

Traditional clustering algorithms including k-means, k-medoids, and the hierarchical ones very often lose their effectiveness when the dataset contains significantly fewer data items compared to the dimensionality of the associated feature space [Witten and Tibshirani, 2010; Li *et al.*, 2018; Chang *et al.*, 2017; Jin and Wang, 2016]. Such *curse of dimensionality* for clustering, manifests in the following different forms [Pandove *et al.*, 2018]: difficulty in global optimization of the clustering objective, distance concentration on L_p norms, the effect of irrelevant or noisy features, and extremely sparse data volume. In most scenarios, only a small proportion of the features can be assumed to be relevant for clustering [Witten and Tibshirani, 2010; Chang *et al.*, 2017; Jin and Wang, 2016]. The goal of a clustering algorithm is to identify these features, avoid the negative influences of the noisy features, and thus, more accurately identify the underlying cluster structure [Witten and Tibshirani, 2010; Jin and Wang, 2016]. Conventional ways of handling the challenge of high-dimensional clustering include subspace clustering of various forms (CLIQUE, SUBCLUE, DBSCAN, DUSC etc.) [Kriegel *et al.*, 2009], correlation clustering [Kriegel *et al.*, 2008], bi-clustering [Pontes *et al.*, 2015], and several dimensionality-reduction based approaches including Principal Component Analysis (PCA), independent component analysis, non-linear matrix factorization, isometric feature mapping, sparse feature weighting, and so on [Kriegel *et al.*, 2009]. Among the competitive and recent approaches in this direction, the Influential Features PCA (IF-PCA) [Jin and Wang, 2016] is a spectral clustering method that selects features with a higher degree of relevance to the clustering task by using the largest Kolmogorov-Smirnov(KS) scores and then applies classical PCA to the post-selection data matrix. Sarkar and Ghosh [2019] recently suggested an approach to tackle high-dimensional clustering with a new data-driven measure of dissimilarity, referred by the authors as MADD (Mean of Absolute Differences of pairwise Distances) specifically tailored for the high-dimensional feature spaces.

The widely popular k-Means [Lloyd, 1982] algorithm suffers from a strong sensitivity to initialization. Its clustering solution is not at all robust against the initial seed points (candidate cluster centers) and thus, it often gets trapped in poor local minima [Peña *et al.*, 1999; Celebi *et al.*, 2013]. To tackle this problem, Tzortzis and Likas [2014] suggested the MinMax k-means clustering algorithm, which begins with a random set of candidate cluster centers and attempts to minimize the *maximum* intra-cluster variance instead of the traditional *sum*. Minimizing the *sum* does not consider the relative differences among the cluster variances, but by minimizing the *maximum* intra-cluster variance, large variance clusters are avoided and high-quality solutions are produced.

The relaxed *maximum* variance objective of the MinMax k-Means clustering algorithm assigns weights to each cluster but it does not perform any weighting on the features. Sparse regularization can, therefore, be imposed to extend its clustering efficacy in the high-dimensional space. We first justify that MinMax k-Means objective can be reformulated into Witten and Tibshirani's [2010] *sparse clustering framework*. This framework offers a specific feature-weighting method, which optimizes the weighted cost objective function using a *lasso*-type penalty (ℓ_1-norm regularization), hence assigning exact zero weights to noisy features [Witten and Tibshirani, 2010]. In this work, we thus propose the novel Sparse MinMax k-Means algorithm. Our main contributions can be summarized in the following way:

- We justify that MinMax k-Means objective can be re-

*Contact Author

formulated into Witten and Tibshirani's [2010] *sparse clustering framework*, which offers a specific feature-weighting method and optimizes a weighted cost objective function by using a *lasso* type penalty.

- We propose the Sparse MinMax *k*-Means algorithm which maximizes a new weighted *between-cluster sum of squares* ($BCSS$) with the ℓ_1-norm regularization to impose exact zero weights on the noisy features. We thus extend the advantages of the MinMax *k*-Means algorithm in the high-dimensional space.
- We compare the performance of our approach with other well-known high-dimensional clustering algorithms through extensive experiments on several real word datasets (especially, the high-dimensional gene microarray datasets).

2 Preliminaries

In this section, we briefly discuss the MinMax *k*-Means and Sparse *k*-Means objectives along with the notations leading to the formulation of our proposed method.

2.1 Notations

We consider $\mathbf{X} = (x_{ij}) \in \mathcal{R}^{n\times p}$ to be our data set in matrix format where x_{ij} represents the j^{th} feature (column) of the i^{th} observation (row). Here n and p denote the number of observations and the number of features respectively. We consider *K* clusters and the set of cluster centers $\mathbf{c} = (c_{kj}) \in \mathcal{R}^{K\times p}$. c_k represents the k^{th} cluster center and C_k denotes the k^{th} cluster. V_k denotes variance of the cluster *k* where the cluster variance is defined as the sum, and not the average, of the squared distances from the observations belonging to the cluster to its center. δ_{ik} is a cluster indicator variable with $\delta_{ik} = 1$ if x_i belongs to cluster C_k and 0 otherwise. ε_{max} denotes maximum intra-cluster variance. ε_w represents the weighted formulation of sum of the intra-cluster variances and w_k denotes the weight assigned to cluster C_k in the MinMax *k*-Means algorithm [Tzortzis and Likas, 2014].

2.2 The MinMax *k*-Means Objective

The MinMax *k*-Means algorithm minimizes the *maximum* intra-cluster variance ε_{max} (1):

$$\varepsilon_{max} = \max_{1\le k\le K} V_k = \max_{1\le k\le K} \sum_{i=1}^{n} \delta_{ik}\|x_i - c_k\|^2 . \quad (1)$$

Minimizing ε_{max} is a non-trivial optimization problem. Thus, Tzortzis and Likas [2014] came up with a relaxed maximum variance objective. A weighted formulation ε_w of the *sum* of intra-cluster variances was thus constructed as in (2), where a higher weight w_k was given to clusters with high variance to follow the behavior induced by the maximum variance criterion.

$$\begin{aligned} \varepsilon_w &= \sum_{k=1}^{K} w_k^{\alpha} V_k = \sum_{k=1}^{K} w_k^{\alpha} \sum_{i=1}^{n} \delta_{ik}\|x_i - c_k\|^2 , \\ w_k &\ge 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \le \alpha < 1. \end{aligned} \quad (2)$$

The exponent α is a user defined constant.[1] To compensate for the formation of large clusters, a higher weight should be induced for a higher variance. Maximizing ε_w with respect to the weights provide a way to realize this. Thus, the min-max problem can be written as:

$$\begin{aligned} &\min_{\{C_k\}_{k=1}^K} \max_{\{w_k\}_{k=1}^K} \varepsilon_w, \\ &\text{s.t.} \quad w_k \ge 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \le \alpha < 1. \end{aligned} \quad (3)$$

2.3 The Sparse Clustering Framework

Witten and Tibshirani [2010] reformulated *k*-Means and hierarchical clustering as an optimization problem in the following way:

$$\max_{\boldsymbol{\Theta}\in\tau} \quad \sum_{j=1}^{p} f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}), \quad (4)$$

where $f_j(\boldsymbol{X_j}, \boldsymbol{\Theta})$ is a function that involves only the jth feature of the data, and $\boldsymbol{\Theta}$ is a model parameter that belongs to a set τ. They further defined a *sparse clustering framework* as a solution to the above problem which is as follows:

$$\begin{aligned} &\max_{\boldsymbol{\omega},\boldsymbol{\Theta}\in\tau} \quad \sum_{j=1}^{p} \omega_j f_j(\boldsymbol{X_j}, \boldsymbol{\Theta}), \\ &\text{s.t.} \qquad \|\boldsymbol{\omega}\|_2 \le 1, \quad \|\boldsymbol{\omega}\|_1 \le s, \quad \omega_j \ge 0 \quad \forall j, \end{aligned} \quad (5)$$

where s is a tuning parameter that determines the number of retained features for clustering and $\|\boldsymbol{\omega}\|_1, \|\boldsymbol{\omega}\|_2$ are respectively the ℓ_1 and ℓ_2-norms of the weight vector $\boldsymbol{\omega}$. ω_j determines the contribution of the jth feature to the objective function (5). The ℓ_1-norm or *Lasso* penalty results in sparsity in different applications [Witten and Tibshirani, 2010; Li *et al.*, 2018; Chang *et al.*, 2017].

3 The Sparse MinMax *k*-Means Algorithm

3.1 Deriving the Formulation

We introduce a new variable z_{ik}, which is similar to the cluster indicator variable δ_{ik} and can be defined as:

$$\begin{aligned} z_{ik} &= \begin{cases} w_k & \text{if } x_i \in C_k, \\ 0 & \text{otherwise,} \end{cases} \\ \text{and } z_{ik}^{\alpha} &= w_k^{\alpha}, \quad \text{if } x_i \in C_k. \end{aligned} \quad (6)$$

Here w_k's are the cluster weights as defined in the MinMax *k*-Means algorithm [Tzortzis and Likas, 2014] in Section 2.2. Thus, the ε_w defined in (2) can be re-written as follows:

$$\varepsilon_w = \sum_{k=1}^{K} w_k^{\alpha} \sum_{i=1}^{n} \delta_{ik}\|x_i - c_k\|^2 = \sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - c_k\|^2 . \quad (7)$$

We now provide Lemma 1 to prove that the MinMax *k*-Means clustering model can also be reformulated in the framework (4). We use a method similar to the one used by Chang et al. [2017] in context to sparse fuzzy clustering.

[1] The role of α is discussed in Section 3.3

Lemma 1. *For the data matrix* $\boldsymbol{X}$*,*

$$\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - c_k\|^2 = \sum_{k=1}^{K}\frac{1}{2n'_k}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_i - x_{i'}\|^2,$$

where $n'_k = \sum_{i=1}^{n} z_{ik}^{\alpha} = n_k w_k^{\alpha}$*,* n_k *is the number of observations in cluster k, and* c_k *is the* k^{th} *cluster center such that:*

$$c_k = \frac{\sum_{i=1}^{n} z_{ik}^{\alpha}x_i}{n'_k} = \frac{w_k^{\alpha}\sum_{i=1}^{n}\delta_{ik}x_i}{w_k^{\alpha}n_k} = \frac{\sum_{i=1}^{n}\delta_{ik}x_i}{\sum_{i=1}^{n}\delta_{ik}}.$$

Proof. The right hand side can be written as

$$\sum_{k=1}^{K}\frac{1}{2n'_k}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_i - x_{i'}\|^2$$
$$= \sum_{k=1}^{K}\frac{1}{2n'_k}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\Big\{\|x_i - c_k\|^2 + \|x_{i'} - c_k\|^2 + 2(x_i - c_k)^{\mathrm{T}}(x_{i'} - c_k)\Big\}.$$

Since $n'_k = \sum_{i'=1}^{n} z_{i'k}^{\alpha}$, we have

$$\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_i - c_k\|^2 = n'_k\sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - c_k\|^2.$$

Similarly, we have

$$\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_{i'} - c_k\|^2 = n'_k\sum_{i'=1}^{n} z_{i'k}^{\alpha}\|x_{i'} - c_k\|^2.$$

Now, $\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}(x_i - c_k)^{\mathrm{T}}(x_{i'} - c_k)$

$$= \Big[\sum_{i=1}^{n} z_{ik}^{\alpha}(x_i - c_k)\Big]\Big[\sum_{i'=1}^{n} z_{i'k}^{\alpha}(x_{i'} - c_k)\Big] = 0.$$

Thus,

$$\sum_{k=1}^{K}\frac{1}{2n'_k}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_i - x_{i'}\|^2 = \sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - c_k\|^2.$$

□

The left hand side of Lemma 1 is the objective function (2) that is optimized by the MinMax *k*-Means Algorithm [Tzortzis and Likas, 2014], while the right hand side evaluates the dissimilarity within a cluster, which can be referred to as the *within-cluster sum of squares* ($WCSS$) of MinMax *k*-Means. The right hand side of Lemma 1 can be further simplified as:

$$\text{If } x_i, x_{i'} \in C_k \text{ then } z_{ik}^{\alpha} = z_{i'k}^{\alpha} = w_k^{\alpha}.$$

$$\text{In addition, } \sum_{k=1}^{K}\frac{1}{2n'_k}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha}z_{i'k}^{\alpha}\|x_i - x_{i'}\|^2$$
$$= \sum_{k=1}^{K}\frac{1}{2n_k.w_k^{\alpha}}\sum_{x_i,x_{i'}\in C_k} w_k^{2\alpha}\|x_i - x_{i'}\|^2$$
$$= \sum_{k=1}^{K}\frac{w_k^{\alpha}}{2n_k}\sum_{x_i,x_{i'}\in C_k}\|x_i - x_{i'}\|^2. \tag{8}$$

Note that the product $z_{ik}^{\alpha}z_{i'k}^{\alpha}$ is non-zero only when both x_i and $x_{i'} \in C_k$ and is zero otherwise. The $WCSS$ measure for our Sparse MinMax *k*-Means model can be written as:

$$WCSS = \sum_{k=1}^{K}\frac{w_k^{\alpha}}{n_k}\sum_{x_i,x_{i'}\in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2. \tag{9}$$

From the constraints of the objective function (2) of the MinMax *k*-Means algorithm [2014], we can deduce that $w_k^{\alpha} \leq 1$. Hence, we can write:

$$\sum_{k=1}^{K} w_k^{\alpha}\sum_{i=1}^{n}\delta_{ik}\|x_i - c_k\|^2 \leq \sum_{k=1}^{K}\sum_{i=1}^{n}\delta_{ik}\|x_i - c_k\|^2.$$

This implies that the $WCSS$ of MinMax *k*-Means will always be less than or equal to the $WCSS$ defined for classical *k*-Means. Therefore, we can model the new *between-cluster sum of squares* ($BCSS$) of MinMax *k*-Means as:

$$BCSS_j = \frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n}(x_{ij} - x_{i'j})^2 - \sum_{k=1}^{K}\frac{w_k^{\alpha}}{n_k}\sum_{x_i,x_{i'}\in C_k}(x_{ij} - x_{i'j})^2, \tag{10}$$

where $BCSS(\boldsymbol{\Theta}) = (BCSS_1, \ldots, BCSS_p)^{\top}$ with $\boldsymbol{\Theta} = \boldsymbol{C} = (C_1, \ldots, C_K)$ and τ is a set of all possible partitions of the observations into *K* clusters.

Now we can rewrite the MinMax *k*-Means problem by using the Witten and Tibshirani's [2010] *sparse clustering framework* as shown below:

$$\max_{\boldsymbol{\Theta}} \quad \min_{\boldsymbol{w}} \quad \sum_{j=1}^{p} BCSS_j,$$
$$\text{s.t.} \quad z_{ik} = \begin{cases} w_k, & \text{if } x_i \in C_k, \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$
$$\text{with } z_{ik}^{\alpha} = w_k^{\alpha} \quad \text{if } x_i \in C_k, \quad i = 1, \ldots, n,$$
$$w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1.$$

Note that in MinMax *k*-Means, we maximize ε_w with respect to the cluster weights (w_k's), and hence we must minimize $\sum_{j=1}^{p} BCSS_j$ with respect to cluster weights.

$BCSS_j, j = 1, \ldots, p$ is a function that only involves the jth feature. Thus, we may conclude that MinMax *k*-Means fits into the framework (4). According to Witten and Tibshirani's [2010] *sparse clustering framework* (5), the MinMax *k*-Means can be generalized to the following model:

$$\max_{\boldsymbol{\Theta},\boldsymbol{\omega}} \quad \min_{\boldsymbol{w}} \quad \boldsymbol{\omega}^{\top} BCSS(\boldsymbol{\Theta}),$$
$$\text{s.t.} \quad \|\boldsymbol{\omega}\|_2 \leq 1, \quad \|\boldsymbol{\omega}\|_1 \leq s, \quad \omega_j \geq 0, \forall j$$
$$z_{ik} = \begin{cases} w_k, & \text{if } x_i \in C_k, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$
$$\text{with } z_{ik}^{\alpha} = w_k^{\alpha} \quad \text{if } x_i \in C_k, \quad i = 1, \ldots, n,$$
$$w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1.$$

We will call (12) as the Sparse MinMax k-Means model.

Denoting $a_j = BCSS_j, \quad \boldsymbol{a} = (a_1, \ldots, a_p)^\top$, the objective function of (12) can be rewritten as $\sum_{j=1}^{p} \omega_j a_j$. We use the technique similar to the one used in [Witten and Tibshirani, 2010] to build an algorithm to solve the problem (12). We alternatively fix two of $\boldsymbol{w}$, c, and $\boldsymbol{\omega}$ and maximize the objective with respect to the other. The optimization problem that arises in the final step while maximizing the objective with respect to ω can be written as:

$$\begin{aligned} &\underset{\boldsymbol{\omega}}{\text{maximize}} \quad \boldsymbol{\omega}^\top a, \\ &\text{s.t.} \quad \|\boldsymbol{\omega}\|_2 \leq 1, \quad \|\boldsymbol{\omega}\|_1 \leq s, \quad \omega_j \geq 0, \forall j. \end{aligned} \tag{13}$$

Following the Proposition stated in page 715 of [Witten and Tibshirani, 2010]), the convex problem bears a solution of the form (13) is $\boldsymbol{\omega} = S(\boldsymbol{a}_+, \triangle)/\left\|S(\boldsymbol{a}_+, \triangle)\right\|_2$, where x_+ indicates the positive part of x and $\triangle = 0$, if that results in $\|\boldsymbol{\omega}\|_1 < s$; else, $\triangle > 0$ is taken to get $\|\boldsymbol{\omega}\|_1 = s$. S is the soft-thresholding operator, defined as $S(x, c) = sign(x)(|x| - c)_+$. The assumptions are the same as in [Witten and Tibshirani, 2010], i.e., there exists a unique maximal element of $\boldsymbol{a}$, and $1 \leq s \leq \sqrt{p}$.

3.2 The Algorithm

From our definitions of $WCSS$ and $BCSS$ in Section 3.1, we see that maximizing the $BCSS$ is equivalent to minimizing the $WCSS$ and vice-versa. This property can be used to simplify our algorithm (1). Steps 3 and 4 can be optimized together by performing MinMax k-means clustering on the data after scaling each feature j by $\sqrt{\omega_j}$, which is same as updating the data matrix with each (i, j) element = $\sqrt{\omega_j} x_{ij}$. In Step 3, the new cluster centres can be calculated in the following way.

$$c_k = \frac{\sum_{i=1}^{n} \delta_{ik} \hat{x_i}}{\sum_{i=1}^{n} \delta_{ik}}, \quad \text{where} \quad \hat{x_i} = (\sqrt{\omega_j} x_{i1}, \ldots, \sqrt{\omega_j} x_{ip}),$$

$$\delta_{ik} = \begin{cases} 1, & k = \text{argmin}_{1 \leq k' \leq K} \, w_{k'}^{\alpha} \sum_{i=1}^{n} \delta_{ik} \|\hat{x_i} - c_{k'}\|^2, \\ 0, & otherwise. \end{cases} \tag{14}$$

In Step 4, the cluster weight constraints in (12) and the new cluster centres obtained in Step 3 are incorporated in the objective via a Lagrange multiplier and the derivatives with respect to w_k are set to zero. Thus, the cluster weights are updated as:

$$w_k = V_k^{1/1-\alpha} / \sum_{k'=1}^{K} V_{k'}^{1/1-\alpha} \text{ where } V_k = \sum_{i=1}^{n} \delta_{ik} \|\hat{x_i} - c_k\|^2. \tag{15}$$

Tzortzis and Likas [2014] observed that addition of a memory effect to the weights could be beneficial in terms of increasing the stability. Thus, for each iteration t,

$$w_k^{(t)} = \beta w_k^{(t-1)} + (1-\beta) \left(V_k^{1/1-\alpha} / \sum_{k'=1}^{K} V_{k'}^{1/1-\alpha} \right), 0 \leq \beta \leq 1. \tag{16}$$

The algorithm iterates through steps 3-5 until the stopping criterion

$$\frac{\sum_{j=1}^{p} \left| \omega_j^{new} - \omega_j^{old} \right|}{\sum_{j=1}^{p} \left| \omega_j^{old} \right|} < \epsilon \tag{17}$$

Algorithm 1 Sparse MinMax k-Means Algorithm

Input: Data matrix X and number of clusters k
Parameter: Tuning parameter s
Output: Clusters $C_1, C_2, \ldots, C_k$

1: Initialize $\boldsymbol{\omega}$ as $\omega_1 = \ldots = \omega_p = \frac{1}{\sqrt{p}}$.
2: **while** stopping criteria (17) is not satisfied **do**
3: Optimize (12) with respect to $C_1, C_2, \ldots, C_K$, keeping $\boldsymbol{w}$ and $\boldsymbol{\omega}$ fixed. That is,

$$\underset{C_1, C_2, \ldots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{w_k^{\alpha}}{n_k} \sum_{x_i, x_{i'} \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

Maximizing (10) is same as minimizing (9).
4: Optimize (12) with respect to $\boldsymbol{w}$, keeping $C_1, C_2, \ldots, C_K$ abd $\boldsymbol{\omega}$ fixed. That is,

$$\underset{w_1, w_2, \ldots, w_K}{\text{maximize}} \left\{ \sum_{k=1}^{K} \frac{w_k^{\alpha}}{n_k} \sum_{x_i, x_{i'} \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

Minimizing (10) is same as maximizing (9).
5: Optimize (12) with respect to $\boldsymbol{\omega}$, keeping $C_1, C_2, \ldots, C_K$ and $w_1, w_2, \ldots, w_K$ fixed, which results in the optimization problem stated in (13) and can be solved using the Proposition stated in page 715 of [Witten and Tibshirani, 2010] to get $\boldsymbol{\omega}^{new}$.
6: **end while**
7: **return** the clusters given by $C_1, C_2, \ldots, C_K$, the cluster weights by $w_1, w_2, \ldots, w_k$ and the feature weights corresponding to this clustering given by $\omega_1, \omega_2, \ldots, \omega_p$.

is satisfied, where the precision level ϵ was chosen as 10^{-4} as suggested in [Witten and Tibshirani, 2010].

3.3 Selection of the Tuning Parameters

With the increase (decrease) of the exponent α, the similarity among the weight values also reduces (enhances) pertaining to the enhancement (suppression) of the relative difference of the variances among the clusters. α is chosen by using a data driven approach to automatically adapt to the data set as in [Tzortzis and Likas, 2014]. It initiates with a small α (α_{init}), which is increased by α_{step} after each iteration, till the attainment of a maximum value α (α_{max}). Following a procedure similar to [Witten and Tibshirani, 2010], we employ a permutation technique and determine the gap statistic [Tibshirani *et al.*, 2001] to select the value of s.

3.4 On Convergence of the Proposed Algorithm

An exact theoretical proof of the local convergence of the minmax objective function in our proposed algorithm is quite difficult and outside the scope of this paper. Nevertheless, we provide some empirical demonstrations of the convergence behavior by recording the variation of the weighted BCSS objective with iterations of the alternating optimization procedure. In all our experiments, the value of the weighted $BCSS$ objective (12) is seen to converge within 10 iterations. Figure 1 empirically demonstrates the gradual convergence of the proposed algorithm (1) to a local stationary point on 5 selected

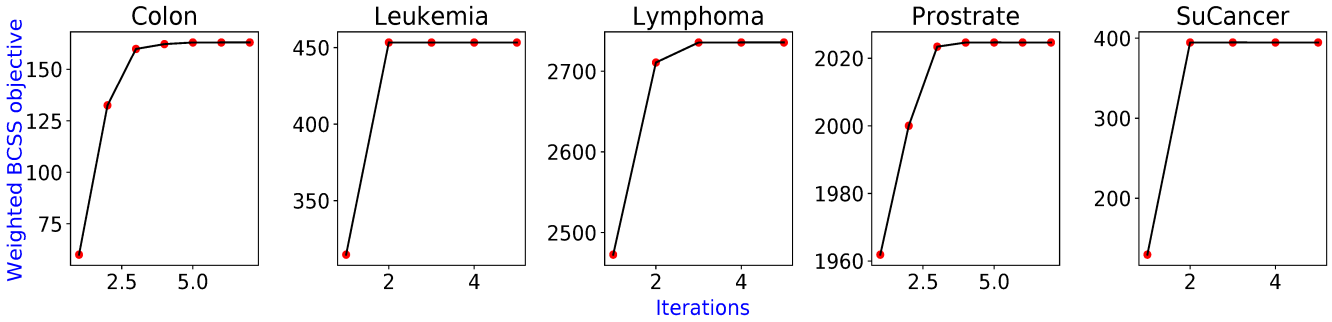


Figure 1: Demonstration of analytical convergence of the weighted $BCSS$ objective value on 5 selected gene microarray datasets

gene microarray datasets [Jin and Wang, 2014]. For other datasets (not shown here), the convergence characteristics exhibit a similar trend.

3.5 Complexity Analysis

The computational complexity of the classical k-Means is $O(npKi)$ where i is the number of iterations. The MinMax k-Means algorithm has an additional weight update step which can be done in $O(nK)$, hence the algorithm also has a complexity of $O(npKi)$. In the $\boldsymbol{\omega}$ update step, we solve (13) by the binary searching scheme numerically. The main computational complexity of Step 5 is thus $O(p \log 1/e)$ where e is the error threshold for searching as determined in [Chang *et al.*, 2017]. Hence, the asymptotic complexity of each iteration of the proposed Sparse MinMax k-Means algorithm is $O(npKi + p \log 1/e)$. In most real world cases, the first term will dominate. Thus, it has an overall complexity of the same order as that of classical k-Means. We iterate until the convergence criteria (17) is met.

4 Experimental Evaluation

In this section, we present a comparative analysis of the performance of Sparse MinMax k-Means mainly against the Sparse k-Means [Witten and Tibshirani, 2010], the IF-PCA method [Jin and Wang, 2016], and the MADD versions of k-Means [Sarkar and Ghosh, 2019], which already exhibited an edge over several other high-dimensional clustering methods in their respective papers. We use the Clustering Error Rate (CER) as a performance indicator. It is defined as $CER \triangleq \sum_{i>i'} \left| \mathbf{1}_{\hat{\mathcal{D}}(i,i')} - \mathbf{1}_{\mathcal{D}(i,i')} \right| / \binom{n}{2}$, where $\mathbf{1}_{\mathcal{D}(i,j)}$ is an indicator function to record whether the i^{th} and j^{th} observations are in the same cluster with respect to partition $\mathcal{D} \in \tau$.

As a warm-up exercise, We first evaluate the performance of our algorithm on 3 UCI real world datasets [Bache and Lichman, 2013] along with the *Yale* facial image data set [Belhumeur *et al.*, 1997]. We further evaluate our method on the benchmark datasets used by Sarkar and Gosh [2019] for an effective comparison with their MADD versions of k-Means which are specifically tailored to handle clustering in high-dimensional spaces. Finally, we compare the algorithms using 10 high-dimensional gene microarray datasets [Jin and Wang, 2014]. In our experiments, we apply Sparse MinMax k-Means with $\alpha_{max} = 0.5$ and $\alpha_{step} = 0.01$. Since results obtained with β values of $0, 0.1$, and 0.3 following [Tzortzis and Likas, 2014] did not differ statistically, we report all results with $\beta = 0$. As mentioned earlier, we use the gap statistics for finding the value of the tuning parameter s for both our method and Sparse k-Means. Each simulation is run 20 times and unless otherwise stated, the best result obtained by each algorithm is reported. The best results are marked in boldface across all the result tables. The R code for our implementation of the Sparse MinMax k-Means algorithm is available at *https://github.com/sayak94/Sparse-MinMax-k-Means*. All codes were run on an HP Omen Laptop with intel Core i7-9750H 6-core 2.6 Ghz processor, Windows 10 operating system with 16 GB RAM.

Dataset Name	K	$n \times p$	MADD k-Means	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Glass	6	214×9	0.467(3)	0.532(4)	0.439(2)	**0.374**(1)
Breast	2	699×9	0.050(3)	0.072(4)	**0.047**(1)	**0.047**(1)
Vehicle	4	846×18	0.572(3)	0.611(4)	**0.548**(1)	**0.548**(1)
Yale	15	165×1024	0.558(3)	0.697(4)	0.485(2)	**0.442**(1)
Avg. Rank			3	4	1.5	**1**

Table 1: Comparison of CERs along with the relative ranks (in brackets) obtained by 4 peer algorithms for the real world datasets.

4.1 Evaluation on Selected Real World Datasets

We use the 3 UCI real world datasets (*Glass*, *Breast*, *Vehicle*) [Bache and Lichman, 2013] and the *Yale* face data set which consists of 165 grey scale images of 15 subjects, as used in [Wang *et al.*, 2019]. Each 32×32 image in the latter is flattened into 1024 -dimensional vector [Cai *et al.*, 2007]. The comparisons of our algorithm with MADD k-Means, IF-HCT-PCA, and Sparse k-Means methods for these datasets is shown in Table 1. Here all the MADD versions of k-Means produce similar results, hence it is shown in a single column. In all the 4 cases, our scheme obtains the lowest error rates(pertaining to the improved detection of cluster structures) and hence obtains the best *average rank* among the other algorithms.

Among the 3 benchmark datasets used by Sarkar and Ghosh [2019], *Lymphoma* is studied as one of the gene microarray datasets. The peer algorithms are compared on the other two datasets in this section. *Trace* data has 50 observations from each of the 4 classes and 275 features per observation. *Control Chart* data has 6 classes, a total of 600 observations and 60

Dataset Name	K	$n \times p$	kmeans	kmeans++	Hier	SpecGem	kM_0	kM_1	kM_2	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Brain	5	42×5597	0.286(5)	0.427(9)	0.524(10)	**0.143**(1)	0.309(6)	0.309(6)	0.309(6)	0.262(4)	0.214(2)	0.238(3)
Breast Cancer	2	276×22215	0.442(9)	0.430(6)	0.500(10)	0.438(7)	**0.337**(1)	**0.337**(1)	**0.337**(1)	0.406(4)	0.449(8)	0.417(5)
Colon Cancer	2	62×2000	0.443(8)	0.460(9)	0.387(6)	0.484(10)	0.355(3)	0.355(3)	0.355(3)	0.403(7)	0.306(2)	**0.145**(1)
Leukemia	2	72×3571	0.278(8)	0.257(7)	0.278(8)	0.292(10)	0.042(2)	0.111(5)	0.194(6)	0.069(3)	0.069(3)	**0.028**(1)
Lung Cancer(1)	2	181×12533	0.116(2)	0.196(10)	0.177(9)	0.122(3)	0.127(7)	0.122(3)	0.127(7)	**0.033**(1)	0.122(3)	0.122(3)
Lung Cancer(2)	2	203×12600	0.436(8)	0.439(10)	0.301(5)	0.434(7)	**0.217**(1)	0.227(4)	0.438(9)	**0.217**(1)	0.315(6)	**0.217**(1)
Lymphoma	3	62×4026	0.387(9)	0.317(8)	0.468(10)	0.226(7)	**0.032**(1)	**0.032**(1)	0.048(5)	0.065(6)	**0.032**(1)	**0.032**(1)
Prostate Cancer	2	102×6033	0.422(7)	0.432(9)	0.480(10)	0.422(7)	0.382(2)	0.392(4)	0.412(6)	0.382(2)	0.392(4)	**0.372**(1)
SRBCT	4	63×2308	0.556(10)	0.524(8)	0.540(9)	0.508(5)	0.476(4)	0.508(5)	0.508(5)	0.444(3)	0.349(2)	**0.333**(1)
SuCancer	2	174×7909	0.477(6)	0.459(5)	0.448(4)	0.489(10)	0.477(6)	0.477(6)	0.477(6)	0.333(3)	**0.328**(1)	**0.328**(1)
Avg. Rank			7.2	8.1	8.1	6.7	3.3	3.8	5.4	3.4	3.2	**1.8**

Table 2: Comparison of CERs along with the relative ranks (in brackets) obtained by different clustering methods for the 10 gene microarray datasets. Last row gives the average rank of each method over all 10 datasets.

Dataset Name	kM_0	kM_1	kM_2	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Trace	0.475(3)	**0.445**(1)	0.485(5)	0.485(5)	0.475(3)	**0.445**(1)
Control Chart	0.342(4)	0.365(5)	0.317(2)	0.517(6)	0.335(3)	**0.200**(1)
Avg. Rank	3.5	3	3.5	5.5	3	**1**

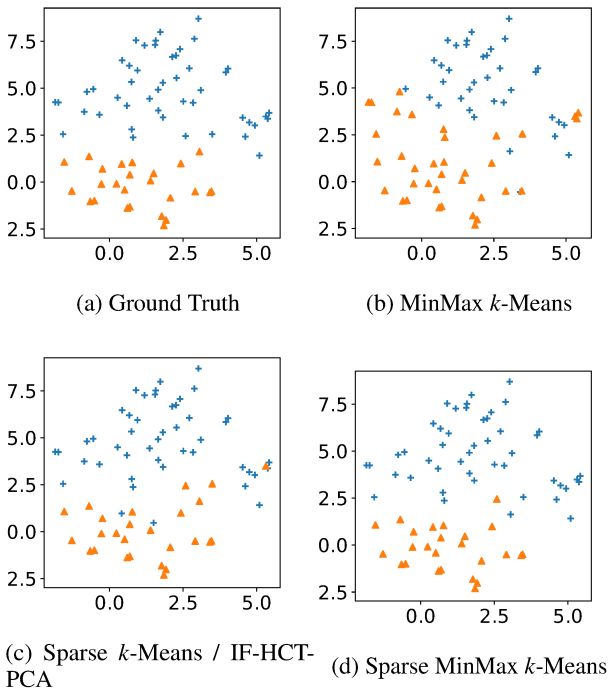
Table 3: Comparison of CERs along with the relative ranks (in brackets) obtained by MADD versions of k-Means and the other three methods for benchmark datasets.

features. Our method obtains better or similar results for both the datasets as seen in Table 3.

4.2 Evaluation on Gene Microarray Datasets

We now present the comparative results on 10 high-dimensional gene microarray datasets [Jin and Wang, 2014]. The datasets stem from patients from several classes (normal, diseased). For each patient, gene expression levels on the same set of genes are available. These datasets are the ones that have been used by Jiashun and Wang [2016] for testing their IF-HCT-PCA clustering. These datasets have the property of $p \gg n$. In [Jin and Wang, 2016], the IF-HCT-PCA clustering algorithm was compared with other well-known clustering methods by using these 10 datasets. Using their code, we obtain the CERs for IF-HCT-PCA and also the other algorithms. We now add more columns to the comparison table, which are the error rates obtained by MADD versions of k-Means, Sparse k-Means, and Sparse MinMax k-Means. Additionally, we show 2D t-SNE plots [van der Maaten and Hinton, 2008] of the results obtained by our method, MinMax k-Means, Sparse k-Means, and IF-HCT-PCA clustering on the *Leukemia* dataset in Figure 2.

Table 2 summarises the comparison results between our algorithm and the MADD versions of k-Means, IF-HCT-PCA, Sparse k-Means, along with 4 other well-known clustering algorithms as baselines: classical k-means, k-means++ [Arthur and Vassilvitskii, 2007], classical Hierarchical clustering, and the Spectral GEM algorithm [Lee *et al.*, 2010]. On 7 out of the 10 datasets, our method obtains the lowest CER among all the other algorithms. It attains lower or similar CER compared to that obtained by the MADD versions of k-Means in 9 out of 10 cases, in 7 out of 10 cases for IF-HCT-PCA and in 9 out of 10 cases for Sparse k-Means.



(a) Ground Truth (b) MinMax k-Means

(c) Sparse k-Means / IF-HCT-PCA (d) Sparse MinMax k-Means

Figure 2: A t-SNE visualization of the clustering results obtained on the Leukemia data set in 2D.

We also compare our algorithm with MADD k-Means (kM_0), IF-HCT-PCA, Sparse k-Means, and Sparse MinMax k-Means methods in terms of the mean CERs obtained over 20 independent runs. The results are shown in Table 4 along with the standard deviations obtained in each case. If the value of a standard deviation is less than 10^{-4}, it is recorded as ± 0.000. Wilcoxon's rank sum test at 5% significance level is used to validate the pairwise comparison against the best method in each row. The p-values obtained are indicated within the pair of second brackets. If in an entry, $p \leq 0.05$, then the corresponding difference of the result with the best algorithm in the

Dataset Name	MADD k-Means	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Brain	0.358± 0.029(4) $\{2.13e-03\}$	0.355± 0.065(3) $\{1.47e-03\}$	**0.214± 0.000**(1)	0.278± 0.078(2) $\{7.58e-02\}$
Breast Cancer	**0.337± 0.0012**(1)	0.406± 0.002(2) $\{3.29e-06\}$	0.449± 0.000(4) $\{2.63e-07\}$	0.419± 0.004(3) $\{4.39e-05\}$
Colon Cancer	0.354± 0.003(3) $\{5.08e-04\}$	0.403± 0.000(4) $\{3.21e-05\}$	0.306± 0.000(2) $\{7.93e-03\}$	**0.206± 0.078(1)**
Leukemia	0.042± 0.000(2) $\{1.64e-03\}$	0.069± 0.000(3) $\{6.38e-05\}$	0.069± 0.000(3) $\{6.29e-05\}$	**0.038± 0.013(1)**
Lung Cancer(1)	0.127± 0.000(3) $\{9.03e-06\}$	**0.033± 0.000**(1)	0.122± 0.000(2) $\{4.26e-06\}$	0.171± 0.056(4) $\{3.72e-07\}$
Lung Cancer(2)	**0.217± 0.000**(1)	**0.217± 0.000**(1)	0.315± 0.000(2) $\{5.38e-05\}$	**0.217± 0.000**(1)
Lymphoma	0.079± 0.067(2) $\{4.72e-05\}$	0.115± 0.074(3) $\{8.14e-08\}$	**0.032± 0.000**(1)	**0.032± 0.000**(1)
Prostate Cancer	0.387± 0.005(3) $\{4.46e-02\}$	0.382± 0.000(2) $\{7.32e-02\}$	0.392± 0.000(4) $\{8.61e-04\}$	**0.381± 0.011**(1)
SRBCT	0.515± 0.059(4) $\{3.89e-04\}$	0.416± 0.058(3) $\{4.76e-03\}$	0.349± 0.000(2) $\{4.02e-02\}$	**0.335± 0.005**(1)
SuCancer	0.477± 0.000(3) $\{5.37e-06\}$	0.333± 0.000(2) $\{3.17e-03\}$	**0.328± 0.000**(1)	**0.328± 0.000**(1)
Avg. Rank	2.6	2.4	2.2	**1.6**

Table 4: Comparison of mean CERs along with their standard deviations obtained by MADD k-Means (kM_0), IF-HCT-PCA, Sparse k-Means, and Sparse MinMax k-Means methods for the gene microarray datasets. The p-values corresponding to Wilcoxon's rank sum test (at 5% significance level) for the pairwise comparison against the best method in each row is indicated within the pair of second brackets.

row can be treated as statistically significant. Using the mean CERs we obtain a similar performance as the one obtained in Table 2.

To further analyze the performance of our algorithm, we use two more criteria (Retained Features (RF) and Dunn Index (DI)). DI [Dunn, 1974] denotes the ratio of the minimum distance between observations from two different clusters to the maximum intra-cluster dissimilarity. It can be mathematically defined as $DI \triangleq \min_{1\leq k\leq l\leq K} \delta(G_k, G_l)/\max_{1\leq m\leq K} \triangle_m$, where $\delta(G_k, G_l)$ is the inter-cluster distance between clusters G_k and G_l, and $\triangle_m$ calculates the maximum distance between all items within cluster G_m. RF is the number of retained features in each of the algorithms or basically the number of non-zero feature weights (for sparse algorithms). We show the comparative results against IF-HCT-PCA and Sparse k-Means in Table 5. Here we do not compare with MADD as it is not one of the feature selection oriented clustering methods. Higher DI values would indicate better clustering and our method obtains the highest DI values among the three methods in 8 out of the 10 cases. The lowest RF values were, however, obtained in only 4 cases by our method although it has the best CER values in most of the cases (indicating a better recovery of the ground truth cluster structure).

In addition, we provide the average run-time (in seconds) over 20 independent runs for 4 main algorithms used in this study on the gene microarray datasets in Table 6. The IF-HCT-PCA method is seen to have the highest run-time, while Sparse k-Means and our algorithm have a run-time of similar order (though, experimentally, the later takes a slightly higher execution time due to the extra cluster-weight update step). The MADD k-Means approach has the least run-time.

Dataset Name	IF-HCT-PCA		Sparse k-Means		Sparse MinMax k-Means	
	RF	DI	RF	DI	RF	DI
Brain	453	0.634	**123**	0.589	1810	**0.647**
Breast Cancer	728	0.182	22215	0.189	**79**	**0.197**
Colon Cancer	**25**	0.427	1237	0.377	76	**0.435**
Leukemia	213	0.556	3571	0.620	**148**	**0.621**
Lung Cancer(1)	251	0.128	260	0.245	**16**	**0.245**
Lung Cancer(2)	418	0.548	12600	0.244	**5**	**0.548**
Lymphoma	**44**	0.509	4026	**0.651**	717	0.616
Prostate Cancer	**1551**	**0.509**	6033	0.399	5650	0.393
SRBCT	**52**	0.433	742	0.443	1019	**0.544**
SuCancer	**805**	0.486	7909	0.505	1370	**0.505**

Table 5: RF (Retained Features) and DI (Dunn Index) obtained by the Sparse k-Means, IF-HCT-PCA, and Sparse MinMax k-Means methods for the gene microarray datasets.

Dataset Name	MADD k-Means	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Brain	0.063	4.338	0.936	1.386
Breast Cancer	8.953	48.294	8.501	7.712
Colon Cancer	0.127	1.413	0.289	0.341
Leukemia	0.126	2.636	0.317	0.784
Lung Cancer(1)	1.881	17.158	2.841	5.305
Lung Cancer(2)	2.635	18.87	3.614	5.137
Lymphoma	0.098	7.814	0.552	1.857
Prostate Cancer	0.269	5.16	1.453	4.293
SRBCT	0.079	1.619	0.398	1.126
SuCancer	1.137	8.469	1.703	2.918

Table 6: Average run-time in seconds obtained by MADD k-Means (kM_0), IF-HCT-PCA, Sparse k-Means, and Sparse MinMax k-Means methods for the gene microarray datasets.

5 Conclusion

We proposed a Sparse MinMax k-Means approach to detect meaningful clusters in higher dimensional feature spaces. Our approach attempts to extend the reach of traditional k-Means clustering to the high-dimension and low sample-size (i.e. $p \gg n$) situations. The experimental results obtained in Section 4 supports the improved performance of our approach over other approaches in general. Our algorithm obtained the best average rank in all the experiments among the other algorithms. In the cases where any other algorithm has produced a better CER, our scheme remains only marginally behind.

The proposed clustering scheme can be readily extended to sparse multi-view settings by using multiple kernels for handling more complicated vision data. It may also be useful to employ other alternative dissimilarity measures, especially the divergence-based ones (see for example [Chakraborty and Das, 2017]) in the proposed clustering framework.

Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

References

[Arthur and Vassilvitskii, 2007] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[Bache and Lichman, 2013] K. Bache and M. Lichman. Uci machine learning repository, 2013.

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang. Learning a spatially smooth subspace for face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007.

[Celebi *et al.*, 2013] M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

[Chakraborty and Das, 2017] Saptarshi Chakraborty and Swagatam Das. k-means clustering with a new divergence-based distance metric: Convergence and performance analysis. *Pattern Recognition Letters*, 100:67–73, 2017.

[Chang *et al.*, 2017] Xiangyu Chang, Qingnan Wang, Yuewen Liu, and Yu Wang. Sparse regularization in fuzzy c -means for high-dimensional data clustering. *IEEE Transactions on Cybernetics*, 47(9):2616–2627, 2017.

[Dunn, 1974] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[Jin and Wang, 2014] Jiashun Jin and Wanjie Wang. Gene microarray data sets, 2014.

[Jin and Wang, 2016] Jiashun Jin and Wanjie Wang. Influential features pca for high dimensional clustering. *The Annals of Statistics*, 44(6):2323–2359, 2016.

[Kriegel *et al.*, 2008] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Detecting clusters in moderate-to-high dimensional data: subspace clustering, pattern-based clustering, and correlation clustering. *Proceedings of the VLDB Endowment*, 1(2):1528–1529, 2008.

[Kriegel *et al.*, 2009] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[Lee *et al.*, 2010] Ann B. Lee, Diana Luca, and Kathryn Roeder. A spectral graph approach to discovering genetic ancestry. *Ann. Appl. Stat.*, 4(1):179–202, 03 2010.

[Li *et al.*, 2018] Rongjian Li, Xiangyu Chang, Yu Wang, and Zongben Xu. Sparse k-means with ℓ_∞/ℓ_0 penalty for high-dimensional data clustering. *Statistica Sinica*, 2018.

[Lloyd, 1982] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[Pandove *et al.*, 2018] Divya Pandove, Shivan Goel, and Rinkl Rani. Systematic review of clustering high-dimensional and large datasets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):16, 2018.

[Peña *et al.*, 1999] J.m Peña, J.a Lozano, and P Larrañaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027–1040, 1999.

[Pontes *et al.*, 2015] Beatriz Pontes, Raúl Giráldez, and Jesús S. Aguilar-Ruiz. Biclustering on expression data: A review. *Journal of Biomedical Informatics*, 57:163 – 180, 2015.

[Sarkar and Ghosh, 2019] Soham Sarkar and Anil K. Ghosh. On perfect clustering of high dimension, low sample size data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[Tibshirani *et al.*, 2001] Robert Tibshirani, Guenther Walther, and Trevor Hastie. 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, 2001.

[Tzortzis and Likas, 2014] Grigorios Tzortzis and Aristidis Likas. The minmax k-means clustering algorithm. *Pattern Recognition*, 47(7):2505–2516, 2014.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[Wang *et al.*, 2019] Xiao-Dong Wang, Rung-Ching Chen, Fei Yan, Zhi-Qiang Zeng, and Chao-Qun Hong. Fast adaptive k-means subspace clustering for high-dimensional data. *IEEE Access*, 7:42639–42651, March 2019.

[Witten and Tibshirani, 2010] Daniela M. Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.