The Sparse MinMax k-means Algorithm for High-Dimensional Data Clustering

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

Master of Technology
in
Computer Science

by

Sayak Dey
[Roll No: CS-1617]

under the guidance of

Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit

Indian Statistical Institute
Kolkata-700108, India

July 2018

To my family and friends

CERTIFICATE

This is to certify that the dissertation entitled **"The Sparse MinMax *k*-means Algorithm for High-Dimensional Data Clustering"** submitted by **Sayak Dey** 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 this institute and, in my opinion, has reached the standard needed for submission.

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

Acknowledgments

I would like to show my highest gratitude to my advisor, *Dr. Swagatam Das*, Associate Professor, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. 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 which added an important dimension to my research work.

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. I thank all those, whom I have missed out from the above list.

Sayak Dey
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

We consider the problem of clustering observations $X_i \in R^p, i = 1, 2, ...n$ into K possible clusters. We are mainly interested in the modern regime of $p >> n$, i.e., when we have a potentially large set of features compared to the number of observations, and where classical clustering methods face challenges.

In the framework of *sparse clustering* that uses lasso-type penalty to adaptively select the best features for clustering, we propose the method of Sparse MinMax k-Means Clustering. We use the MinMax k-Means Clustering algorithm, that assigns weights to the clusters relative to their variance and optimizes a weighted version of the k-Means objective. The Influential Features PCA (IF-PCA) method selects features based on largest Kolmogorov-Smirnov(KS) scores and has been able to obtain good results on high-dimensional gene microarray data sets. The method suggested by us has out performed this IF-PCA method and also the general Sparse k-Means method.

Keywords: *High-Dimensional Data Clustering, k-Means Clustering, Feature Selection, Sparsity, IF-PCA, MinMax k-Means.*

Contents

List of Tables

Chapter 1

Introduction

1.1 Introduction

High-Dimensional Data Clustering problems, i.e., where the objects to be clustered have a very large feature set, are still very challenging [5, 12, 14, 26]. In most of the real-world scenarios, only a small portion of the features is assumed to be relevant for clustering [5, 12, 26]. For example, only a tiny portion of genes (relevant features) are responsible for a certain biological activity, while the others are irrelevant or noisy features [12]. The goal of a good clustering approach is to be able to identify the relevant features and avoid the negative influences of the noisy features [12, 26]. Intuitively, if we do a thresholding by assigning positive weights to the relevant features and exact zero weights to the noisy features, the negative influences of the noisy features could be avoided.

One of the most well-studied clustering algorithms is k-Means [16], which minimizes the sum of the intra-cluster variances. Its simplicity and efficiency have established it as a popular means for performing clustering across different disciplines. Even an extension to kernel space has been developed [6, 20] to enable the identification of non-linearly separable groups. Despite its wide acceptance, k-Means suffers from a serious limitation. Its solution heavily depends on the initial positions of the cluster centers, thus after a bad initialization it easily gets trapped in poor local minima [4, 19].

To deal with this problem, several methods have been proposed. k-Means with multiple random restarts is often employed in practice. Another alternative approach is the Global k-means algorithm [29]. It is an incremental approach that starts from one cluster and at each step a new cluster is deterministically added to the solution according to an appropriate criterion. Based on the algorithm, Bagirov et al. proposed some modifications [2, 3]. A new method to tackle this problem is the MinMax k-means clustering algorithm [23], which starts from a randomly picked set of cluster centers and tries to minimize the maximum intra-cluster variance instead of the sum. This method assigns weights to clusters relative to their variance. The proposed weighting scheme limits the emergence of large variance clusters and allows high quality solutions to be systematically uncovered, irrespective of the initialization.

The MinMax k-Means Clustering Algorithm does not perform any weighting on the the attributes. Hence to extend its clustering performance in high-dimensional data clustering, sparse regularizations may be imposed. We first justify that MinMax k-Means can be reformulated into Witten and Tibshirani's [26] *sparse clustering framework*. Witten and Tibshirani's [26] *sparse clustering framework* offered a specific attribute-weighting method, which optimizes a weighted cost objective function using the ℓ_1-norm regularization technique, thus is able to assign exact zero weights to noisy features [26]. We propose the Sparse MinMax k-Means algorithm which maximizes a new weighted between cluster sum of squares (BCSS) with ℓ_1-norm regularization.

1.2 Our Contribution

Our contributions are summarized as follows.

- We have proved that the MinMax k-Means algorithm can be reformulated into Witten and Tibshirani's [26] *sparse clustering framework* and hence developed the Sparse MinMax k-Means algorithm.
- We have proposed the new weighted Between Cluster Sum of Squares (BCSS) measure for our Sparse MinMax k-means model.
- We have also provided the performance evaluation of our scheme. We have compared our method with the Influential Features PCA (IF-PCA) method [12] and also the general Sparse k-Means method [26] on real world as well as synthetic data sets. We have mainly evaluated our scheme on the high-dimensional gene micro-array data sets.
- We have also done a complexity analysis of our method.

1.3 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we briefly discuss about the preliminaries and the MinMax k-Means algorithm. 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 $\mathbf{X} = (x_{ij}) \in \mathcal{R}^{n\times p}$ to be our data set in matrix format where x_i represents the i$^{\text{th}}$ observation (row) and X_j represents the j$^{\text{th}}$ feature (column). Here n represents the number of observations and p represents the number of features and we will mostly deal with the scenario of $n >> p$. 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 represents 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 k and 0 otherwise. ε_{sum} and ε_{max} denote the sum of intra-cluster variances and the maximum intra-cluster variance respectively. ε_w represents the weighted formulation of the sum of intra-cluster variances and w_k denotes the weight assigned to cluster k in the MinMax k-Means algorithm.

2.2 The *k*-Means Algorithm

To partition a data set X into K disjoint clusters, k-Means [16] minimizes the sum of intra-cluster variances (2.1).

$$\begin{gathered}\varepsilon_{sum} = \sum_{k=1}^{K} V_k = \sum_{k=1}^{K}\sum_{i=1}^{n} \delta_{ik}\|x_i - C_k\|^2 \\ \text{where} C_k = \frac{\sum_{i=1}^{n}\delta_{ik}x_i}{\sum_{i=1}^{n}\delta_{ik}}\end{gathered} \tag{2.1}$$

2.3 The MinMax *k*-Means Algorithm

In this section we are going to discuss the MinMax k-Means Algorithm proposed by G. Tzortzis and A. Likas [23] in detail. In the Section 1.1 we have already talked about the sensitivity of k-Means to initialization and stated that the MinMax k-Means methodology allows k-Means to produce high quality partitionings more systematically, while restarted from random initial centers. It seeks to minimize the maximum intra-cluster variance instead of the sum.

2.3.1 The maximum variance objective

The MinMax k-Means algorithm minimizes the maximum intra-cluster variance (2.2).

$$\varepsilon_{max} = \max_{1\leq k\leq K} V_k = \max_{1\leq k\leq K} \sum_{i=1}^{n} \delta_{ik}\|x_i - C_k\|^2 \tag{2.2}$$

The summation over all clusters in k-Means would give similar ε_{sum} values, for both, when there are a few clusters with large variance that are counterbalanced by others with small variance, or when there is a moderate variance for all clusters. Hence it does not take into account the relative variances among clusters. The variance of a cluster is a measure of its quality. While minimizing ε_{max} in MinMax k-Means [23], large variance clusters are avoided and the solution space is restricted towards clusters that exhibit more similar variances. Thus with the above objective the MinMax k-Means algorithm [23] is less likely to converge to a local minima owing to bad initialization (as shown in Figure 2.1) and produces balanced partitionings based on the variance of clusters, which is highly desirable in most cases.

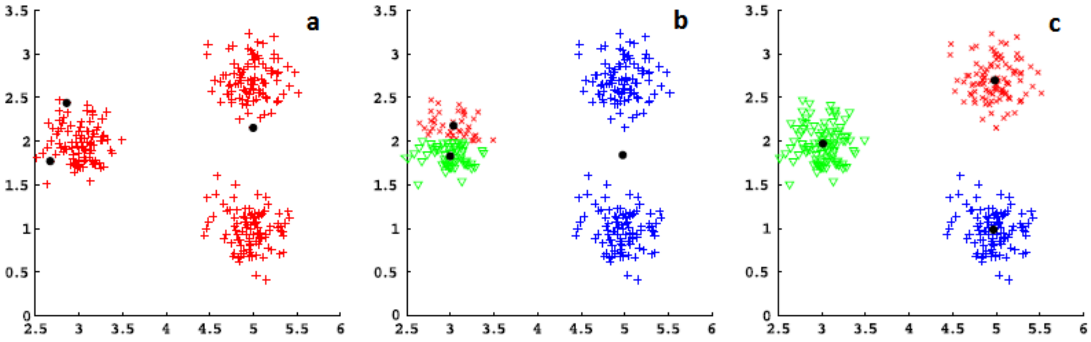


Figure 2.1: Example of (a) a bad initialization that leads to (b) a poor k-Means solution but (c) a good solution is obtained by MinMax k-Means using the same initialization

2.3.2 The relaxed maximum variance objective

Minimizing ε_{max} is a non-trivial optimization problem so a relaxed maximum variance objective was proposed [23]. A weighted formulation ε_w of the sum of the intra-cluster variances was thus constructed (2.3), where a higher weight w_k was placed on clusters with large variance, to mimic the behavior of the maximum variance criterion.

$$\begin{gathered}\varepsilon_w = \sum_{k=1}^{K} w_k^{\alpha} V_k = \sum_{k=1}^{K} w_k^{\alpha} \sum_{i=1}^{n} \delta_{ik} \|x_i - C_k\|^2, \\ w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1\end{gathered} \tag{2.3}$$

The exponent α is a user defined constant. We would discuss the role of α in Section 2.3.4. To penalize large clusters, a higher variance should lead to a higher weight, which can be realized by maximizing ε_w with respect to the weights. Thus the min-max problem can be written as:

$$\begin{gathered}\min_{\{c_k\}_{k=1}^{K}} \max_{\{w_k\}_{k=1}^{K}} \varepsilon_w \\ \text{s.t.} \quad w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1\end{gathered} \tag{2.4}$$

2.3.3 The Algorithm

The MinMax k-Means algorithm iteratively alternates between the c_k and w_k optimization steps which are the minimization and maximization steps that are explained below. The algorithm stops when the stopping criteria is met (like cluster assignments do not change in successive iterations or maximum iterations have been reached).

2.3.3.1 Minimization step

In this step new cluster assignments are made keeping the w_k values fixed. The assignments are made as shown below:

$$\delta_{ik} = \begin{cases} 1 & k = argmin_{1 \leq k' \leq K} \quad w_{k'}^{\alpha} \sum_{i=1}^{n} \delta_{ik} \|x_i - C_{k'}\|^2 \\ 0 & otherwise \end{cases} \tag{2.5}$$

The new cluster centers C_k are calculated using the new cluster assignments.

$$C_k = \frac{\sum_{i=1}^{n} \delta_{ik} x_i}{\sum_{i=1}^{n} \delta_{ik}} \tag{2.6}$$

2.3.3.2 Maximization step

For updating weights, the new cluster assignments and centers, the weight constraints (2.4) are incorporated into the objective via a Lagrange multiplier and the derivatives with respect to w_k are set to zero. Since $\alpha \in [0, 1)$ so it is a concave objective function. Weights are updated as:

$$w_k = V_k^{1/1\,-\,\alpha} / \sum_{k'=1}^{K} V_{k'}^{1/1\,-\,\alpha}$$
$$\text{where} V_k = \sum_{i=1}^{n} \delta_{ik} \|x_i - C_k\|^2 \tag{2.7}$$

As proposed by the authors in [23], a memory effect could be added to the weights enhance the stability of the MinMax k-Means algorithm.

$$w_k^{(t)} = \beta w_k^{(t-1)} + (1 - \beta) \left(V_k^{1/1\,-\,\alpha} / \sum_{k'=1}^{K} V_{k'}^{1/1\,-\,\alpha} \right) \tag{2.8}$$

After updating the w_k values, the stopping criteria is checked, if it is not met, we go back to the Minimization step.

2.3.4 The role of the exponent α

The greater (smaller) the α value is, the less (more) similar the weight values become, as relative differences of the variances among the clusters are enhanced (suppressed). Therefore for a high value of α, large variance clusters accumulate considerably higher w_k and w_k^α values compared to low variance clusters, resulting in an objective that severely penalizes clusters with high variance. So higher α values restricts high variance clusters and lower α values allows high variance clusters. In practice, a moderate value of α is usually suitable. In [23], the authors give a practical framework that extends the MinMax k-Means to automatically adapt the exponent α to the data set. It begins with a small α (α_{init}) that is increased by α_{step} after each iteration, until a maximum value α (α_{max}) is attained. For the method, we should first decide the values of parameters α_{init}, α_{max} and α_{step}.

Chapter 3

Related Work

3.1 Past Work on High-Dimensional Data Clustering

Different approaches have been suggested for clustering data in a high-dimensional setting. Here we do a brief review of the the previous proposals of dimensionality reduction and feature selection in clustering which are generally used in the high-dimensional setting.

Ghosh and Chinnaiyan [10] and Liu et al. [15] proposed to perform principal component analysis (PCA) to reduce the dimensionality of the data matrix X from $n \times p$ to matrix A of dimensions $n \times q$ where $q << p$ and then the n rows were clustered. Similarly, Tamayo et al. [21] suggested to use Non-negative Matrix Factorization (Lee and Seung [13]) to decompose X to obtain the A matrix. But these methods have multiple drawbacks; like A is a function of the full set of p features and there was no guarantee that A would contain the signal that one is interested in detecting via clustering.

The model based clustering frameworks have also been popular in recent years. The basic idea of model based clustering approaches is as follows. One can model the rows of X as independent multivariate observations drawn from a mixture model with K components; usually a mixture of Gaussians is used. The Gaussian density can be parametrized by its mean μ_k and covariance matrix Σ_k. The EM algorithm (to maximize the log likelihood) can be used to fit the model but problem arises in the $p >> n$ case as the $p \times p$ covariance matrix cannot be estimated from only n observations. Proposals for overcoming this has been proposed by Mclachlan, Peel, and Bean in [17,18], which assumes observations lie in a low-dimensional latent factor space. This leads to dimensionality reduction. Rather than choosing μ_k and Σ_k that maximize the log likelihood, instead one can maximize the log likelihood subject to a penalty that is chosen to yield sparsity in the features. This approach is taken in many papers [24, 27, 28] and they lend to feature selection by yielding sparsity.

Friedman and Meulman [9] proposed *Clustering Objects on Subsets of Attributes* (COSA), but unfortunately it did not truly result in sparse clustering as all variables had non-zero weights. Witten and Tibshirani [26] (2010) proposed a framework for feature selection in clustering and used it to design the Sparse k-Means clustering method and the Sparse Hierarchical clustering method. J. Jiashun and W. Wang [12] (2016) propose Influential Features PCA (IF-PCA) as a new clustering procedure. In IF-PCA features were selected based on the largest Kolmogorov-Smirnov (KS) scores and then PCA was applied on the post-selection normalized matrix. Finally it was clustered using the classical k-means. This approach seems to outperform most of the well known clustering algorithms [30]. We would broadly discuss the IF-PCA [12] and the Sparse k-Means [26] methods in the coming sections (3.2, 3.3), with which we mainly compare our proposal.

3.2 Sparse *k*-Means Clustering Method

The k-means algorithm minimizes the Within-Cluster Sum of Squares (WCSS) which can be written as follows:

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in c_k} \sum_{j=1}^{p} d_{i,i',j} \tag{3.1}$$

where $n_k = |c_k|$ and $d_{i,i',j} = (x_{ij} - x_{i'j})^2$. It is same as maximizing the Between-Cluster Sum of Squares (BCSS) which can be written as:

$$\begin{gathered} \sum_{j=1}^{p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in c_k} d_{i,i',j} \right\} \\ \text{we write} \quad a_j \triangleq \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in c_k} d_{i,i',j} \end{gathered} \tag{3.2}$$

Witten and Tibshirani [26] defined a *sparse clustering framework* and modelled the K-means algorithm as

$$\begin{aligned} \max_{\boldsymbol{\omega}, \Theta(C) \in D} \quad \sum_{j=1}^{p} \omega_j f_j(\boldsymbol{X_j}, \Theta(C)) = \sum_{j=1}^{p} \Bigg\{ & \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} (x_{ij} - x_{i'j})^2 \\ & - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in c_k} (x_{ij} - x_{i'j})^2 \Bigg\} \\ \text{s.t.} \qquad \|\boldsymbol{\omega}\|_2 \le 1, \quad & \|\boldsymbol{\omega}\|_1 \le s, \quad \omega_j \ge 0, \forall j \end{aligned} \tag{3.3}$$

where $f_j(\boldsymbol{X_j}, \Theta(C))$ is a function that involves only the j^{th} feature of the data, $\Theta(C)$ is a parameter restricted to a set D, s is a tuning parameter and $1 \le s \le \sqrt{p}$, $\|.\|_2$ is the Euclidean norm, $\|.\|_1$ is the ℓ_1 norm, $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_p)^{\mathrm{T}}$ is the feature weight vector, and the rest of the notations are same as in Section 2.1.

k-means algorithm can be fit to the above framework where $\Theta = (c_1, c_2, \ldots, c_K)$ (a partitioning into K clusters), $f_j(\boldsymbol{X_j}, \Theta(C)) = a_j$, and D denotes the set of partitions of all possible observations into K clusters. Witten and Tibshirani [26] optimized (3.3) using an iterative algorithm: holding $\boldsymbol{\omega}$ fixed, (3.3) is optimized with respect to Θ, and holding Θ fixed, (3.3) is optimized with respect to $\boldsymbol{\omega}$.

3.2.1 Algorithm for Sparse *k*-means Clustering

The Algorithm as described in [26] is as follows:

1. Initialize $\boldsymbol{\omega}$ as $\omega_1 = \ldots = \omega_p = \frac{1}{\sqrt{p}}$.
2. Iterate until convergence:
   (a) Holding $\boldsymbol{\omega}$ fixed, optimize (3.3) with respect to $c_1, c_2, \ldots, c_K$. Hence maximize (3.2) which is same as minimizing (3.1). That is,

$$\underset{c_1, c_2, \ldots, c_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in c_k} \sum_{j=1}^{p} d_{i,i',j} \right\} \tag{3.4}$$

   by applying the standard k-means algorithm to the $n \times n$ dissimilarity matrix with (i, i') element $= \sum_j \omega_j d_{i,i',j}$.

   (b) Holding $c_1, c_2, \ldots, c_K$ fixed, optimize (3.3) with respect to $\boldsymbol{\omega}$ by applying the Proposition: $\boldsymbol{\omega} = \frac{S(\boldsymbol{a}_+, \triangle)}{\|S(\boldsymbol{a}_+, \triangle)\|_2}$ where a_j is as defined in (3.2) and $\triangle = 0$ if that results in $\|\boldsymbol{\omega}\|_1 < s$; otherwise, $\triangle > 0$ is chosen so that $\|\boldsymbol{\omega}\|_1 = s$.
3. The clusters are given by $c_1, c_2, \ldots, c_K$, and the feature weights corresponding to this clustering are given by $\omega_1, \omega_2, \ldots, \omega_p$.

Here d is squared Euclidean distance, Step 2(a) can be optimized by performing K-means on the data after scaling each feature j by $\sqrt{\omega_j}$. Iterate Step 2 until the stopping criterion

$$\frac{\sum_{j=1}^{p} \left|\omega_j^r - \omega_j^{r-1}\right|}{\sum_{j=1}^{p} \left|\omega_j^{r-1}\right|} < 10^{-4}$$

is satisfied.

3.3 Influential Features PCA (IF-PCA) Method

$$X = [x_1, x_2, \dots, x_n]^{\mathrm{T}}$$

The Influential Features PCA (IF-PCA) is a new spectral clustering method by Jin and Wang [12]. The IF-PCA contains an IF part and a PCA part. In the IF part, features are selected by exploiting the sparsity of the contrast mean vectors, where many columns of X are removed, leaving only those which are influential for clustering. In the PCA part, classical PCA is applied to the post-selection data matrix.

Each column of X is normalized and the resultant matrix is denoted by W:

$$W(i,j) = [x_{ij} - \bar{X_j}]/\sigma\hat{(j)}, \quad 1 \le i \le n, 1 \le j \le p, \tag{3.5}$$

where $\bar{X_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ and $\sigma\hat{(j)} = [\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{X_j})^2]^{1/2}$ are the empirical mean and standard deviation associated with feature j, respectively.

$$W = [W_1, W_2, \dots, W_n]^{\mathrm{T}}$$

For any $1 \le j \le p$, the empirical CDF associated with feature j is denoted by

$$F_{n,j}(t) = \frac{1}{n}\sum_{i=1}^{n} 1\{W_i(j) \le t\}. \tag{3.6}$$

3.3.1 The IF-PCA Algorithm

The IF-PCA algorithm contains two ‘IF’ steps and two ‘PCA’ steps as described in [12] and is as follows:

Input: Data matrix X, number of classes K, and parameter t.

Output: Predicted $n \times 1$ label vector $\hat{y}_t^{IF} = (\hat{y}_{t,1}^{IF}, \hat{y}_{t,2}^{IF}, \dots, \hat{y}_{t,n}^{IF})$.

IF-1: For each $1 \le j \le p$, compute a Kolmogorov-Smirnov (KS) statistic by

$$\psi_{n,j} = \sqrt{n}. \sup_{-\infty<t<\infty} |F_{n,j}(t) - \Phi(t)|, \quad (\Phi : \text{CDF of } N(0,1)). \tag{3.7}$$

IF-2: Following the suggestions by Effron [7], renormalize by

$$\psi_{n,j}^* = [\psi_{n,j} - \text{mean of all } p \text{ KS-scores}]/\text{SD of all } p \text{ KS-scores}. \tag{3.8}$$

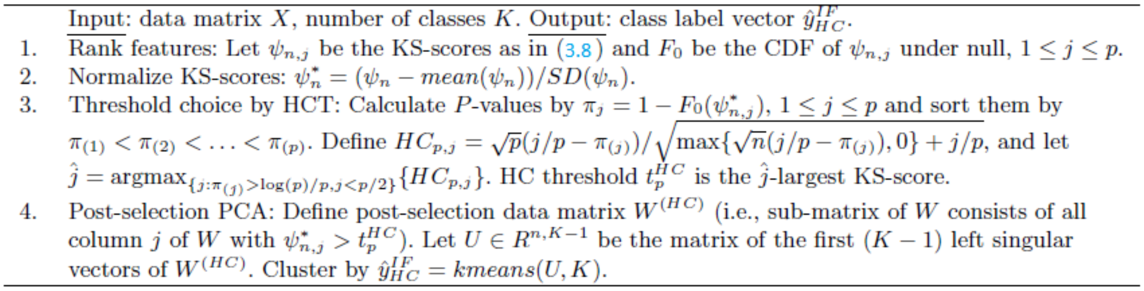
PCA-1: Fix a threshold $t > 0$. Let $W^{(t)}$ be the matrix formed by restricting the columns of W to the set of retained indices $\hat{S}_p(t)$, where

$$\hat{S}_p(t) = \{1 \le j \le p : \psi_{n,j}^* \ge t\} \tag{3.9}$$

Let $\hat{U}^{(t)} \in R^{n,K-1}$ be the matrix consisting $K-1$ (unit-norm) left singular vectors of $W^{(t)}$.

PCA-2: Cluster by applying the classical k-means to $\hat{U}^{(t)}$ assuming K classes. Let $\hat{y}_t^{IF}$ be the predicted label vector.

Here the threshold t is the only tuning parameter. A data-driven threshold choice by Higher Criticism (HC) has been suggested in [12]. The pseudocode of the algorithm including threshold choice by HC is given below.



```
    Input: data matrix $X$, number of classes $K$. Output: class label vector $\hat{y}_{HC}^{IF}$.
1.  Rank features: Let $\psi_{n,j}$ be the KS-scores as in (3.8) and $F_0$ be the CDF of $\psi_{n,j}$ under null, $1 \leq j \leq p$.
2.  Normalize KS-scores: $\psi_n^* = (\psi_n - mean(\psi_n))/SD(\psi_n)$.
3.  Threshold choice by HCT: Calculate $P$-values by $\pi_j = 1 - F_0(\psi_{n,j}^*)$, $1 \leq j \leq p$ and sort them by $\pi_{(1)} < \pi_{(2)} < \ldots < \pi_{(p)}$. Define $HC_{p,j} = \sqrt{p}(j/p - \pi_{(j)})/\sqrt{\max\{\sqrt{n}(j/p - \pi_{(j)}), 0\} + j/p}$, and let $\hat{j} = \text{argmax}_{\{j:\pi_{(j)} > \log(p)/p, j < p/2\}}\{HC_{p,j}\}$. HC threshold $t_p^{HC}$ is the $\hat{j}$-largest KS-score.
4.  Post-selection PCA: Define post-selection data matrix $W^{(HC)}$ (i.e., sub-matrix of $W$ consists of all column $j$ of $W$ with $\psi_{n,j}^* > t_p^{HC}$). Let $U \in R^{n,K-1}$ be the matrix of the first $(K-1)$ left singular vectors of $W^{(HC)}$. Cluster by $\hat{y}_{HC}^{IF} = kmeans(U, K)$.
```

Figure 3.1: *Pseudocode for IF-HCT-PCA (threshold set by Higher Criticism)*

Chapter 4

The Proposed Sparse MinMax *k*-Means Algorithm

4.1 Notations

The previous notations defined in Sections 2.1 and 3.2 for the MinMax k-Means and Sparse k-Means algorithms, remains the same for our proposal also.

Table 4.1: Notations used in our method

Notations	Description
n	The number of observations
p	The number of features
$\mathbf{X} = (x_{ij}) \in \mathcal{R}^{n\times p}$	Data set in matrix form
$x_i \in \mathcal{R}^p$ and $X_j \in \mathcal{R}^n$	The ith row and the jth column of $\mathbf{X}$
$\mathbf{C} = (C_{kj}) \in \mathcal{R}^{K\times p}$	Cluster Centers
C_k and $c_k, k = 1, \dots, K$	The k^{th} cluster center and the k^{th} cluster
n_k	Number of observations in cluster k
$V_k, k = 1, \dots, K$	Variance of the k^{th} cluster
δ_{ik}	Cluster indicator variable
ε_w	Weighted formulation of the intra-cluster sum of squares
$w_k, k = 1, \dots, K$	Weight assigned to cluster k
α	The exponent for cluster weights
$\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_p)^{\mathrm{T}}$	Feature weights
$a_j = BCSS(\mathcal{G})_j$	The new BCSS for feature j
s	The tuning parameter

4.2 Sparse MinMax *k*-Means

Witten and Tibshirani [26] showed that the classical k-means and hierarchical clustering models can be reformulated using their framework which is as follows:

$$\max_{\Theta(\mathcal{G})} \quad \sum_{j=1}^{p} f_j(\boldsymbol{X_j}, \Theta(\mathcal{G})) \tag{4.1}$$

where $f_j(\boldsymbol{X_j}, \Theta(\mathcal{G}))$ is a function related only to the jth feature of the data, and $\Theta(\mathcal{G})$ is a model parameter. They further defined a *sparse clustering framework*

$$\begin{aligned} &\max_{\boldsymbol{\omega}, \Theta(\mathcal{G})} \quad \sum_{j=1}^{p} \omega_j f_j(\boldsymbol{X_j}, \Theta(\mathcal{G})) \\ \text{s.t.} \quad &\|\boldsymbol{\omega}\|_2 \leq 1, \quad \|\boldsymbol{\omega}\|_1 \leq s, \quad \omega_j \geq 0, \forall j \end{aligned} \tag{4.2}$$

where s is a tuning parameter and $\|\boldsymbol{\omega}\|_1 = \sum_{j=1}^{p} |\omega_j|$ is the ℓ_1-norm of $\boldsymbol{\omega}$. Here, ω_j can be interpreted as the contribution of the jth feature to the objective function (4.2) and the tuning parameter s controls the number of features relevant for clustering. The ℓ_1-norm has been proved to be able to generate sparse solutions in various applications [5, 14, 26].

4.2.1 The Formulation

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

$$\begin{gathered} z_{ik} = \begin{cases} w_k & if \quad i \in c_k \\ 0 & otherwise \end{cases} \\ \text{and } z_{ik}^{\alpha} = w_k^{\alpha} \quad \text{if } i \in c_k. \end{gathered} \tag{4.3}$$

Here w_k are the cluster weights as defined in the MinMax k-Means algorithm [23] in Section 2.3. Thus the ε_w defined in (2.3) can be re-written as follows:

$$\varepsilon_w = \sum_{k=1}^{K} w_k^{\alpha} \sum_{i=1}^{n} \delta_{ik} \|x_i - C_k\|^2 = \sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik}^{\alpha} \|x_i - C_k\|^2 \tag{4.4}$$

Next, we provide Lemma 1 which justifies that the MinMax k-Means clustering model is also a special case of the framework (4.1). We use a method similar to the one used by Chang et al. [5] .

Lemma 1 *Suppose* $\boldsymbol{X}$ *is the data matrix, then*

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

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 as defined in (2.6).*

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

Proof:

The right hand side can be written as

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

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

$$\begin{aligned}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha} z_{i'k}^{\alpha}\|x_i - C_k\|^2 &= \sum_{i'=1}^{n} z_{i'k}^{\alpha}\sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - C_k\|^2 \\ &= n'_k\sum_{i=1}^{n} z_{ik}^{\alpha}\|x_i - C_k\|^2.\end{aligned}$$

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

$$\begin{aligned}\sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha} z_{i'k}^{\alpha}(x_i - C_k)^{\mathrm{T}}(x_{i'} - C_k) &= \Big[\sum_{i=1}^{n} z_{ik}^{\alpha}(x_i - C_k)\Big]\Big[\sum_{i'=1}^{n} z_{i'k}^{\alpha}(x_{i'} - C_k)\Big] \\ &= 0.\end{aligned}$$

So

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

□

The left hand side of Lemma 1 is the objective function (2.3) that is optimized by the MinMax k-Means Algorithm [23], while the right hand side evaluates the dissimilarity within 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

$$\text{If } i, i' \in c_k$$

$$z_{ik}^{\alpha} = z_{i'k}^{\alpha} = w_k^{\alpha}$$

$$\begin{aligned}\sum_{k=1}^{K} \frac{1}{2n'_k} \sum_{i=1}^{n}\sum_{i'=1}^{n} z_{ik}^{\alpha} z_{i'k}^{\alpha} \|x_i - x_{i'}\|^2 &= \sum_{k=1}^{K} \frac{1}{2n_k . w_k^{\alpha}} \sum_{i,i' \in c_k} w_k^{2\alpha} \|x_i - x_{i'}\|^2 \\ &= \sum_{k=1}^{K} \frac{w_k^{\alpha}}{2n_k} \sum_{i,i' \in c_k} \|x_i - x_{i'}\|^2\end{aligned} \tag{4.5}$$

Note that the product $z_{ik}^{\alpha}.z_{i'k}^{\alpha}$ is non-zero only when both i and $i' \in c_k$ and is zero otherwise.

From the objective function (2.3) of Tzortzis and Likas's [23] MinMax k-Means algorithm, we have $\sum_{k=1}^{K} w_k = 1$ and $0 \leq \alpha < 1$. Hence w_k^{α} is always less than 1. So we can write

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

That is the WCSS of MinMax k-Means will always be less than the WCSS defined for classical k-Means. Therefore, we can model the *between cluster sum of squares* (BCSS) of MinMax k-Means (4.6) as

$$BCSS(\mathcal{G})_j = \frac{1}{n} \sum_{i=1}^{n}\sum_{i'=1}^{n} (x_{ij} - x_{i'j})^2 - \sum_{k=1}^{K} \frac{w_k^{\alpha}}{n_k} \sum_{i,i' \in c_k} (x_{ij} - x_{i'j})^2 \tag{4.6}$$

where $BCSS(\mathcal{G}) = (BCSS(\mathcal{G})_1, \ldots, BCSS(\mathcal{G})_p)^{\mathrm{T}}$ with respect to the partition $\mathcal{G}$. Thus, the WCSS and the BCSS measures for our Sparse MinMax k-Means model can be written as

$$WCSS(\mathcal{G})_j = \sum_{k=1}^{K} \frac{w_k^{\alpha}}{n_k} \sum_{i,i' \in c_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{4.7}$$

$$BCSS(\mathcal{G})_j = \sum_{j=1}^{p} \left\{ \frac{1}{n} \sum_{i=1}^{n}\sum_{i'=1}^{n} (x_{ij} - x_{i'j})^2 - \sum_{k=1}^{K} \frac{w_k^{\alpha}}{n_k} \sum_{i,i' \in c_k} (x_{ij} - x_{i'j})^2 \right\} \tag{4.8}$$

Now we can rewrite the MinMax k-Means problem as

$$\begin{aligned} \max_{c,\mathcal{G}} \quad \min_{\boldsymbol{w}} \quad & \sum_{j=1}^{p} BCSS(\mathcal{G})_j \\ \text{s.t.} \quad z_{ik} = & \begin{cases} w_k & if \quad i \in c_k \\ 0 & otherwise \end{cases} \\ \text{with } z_{ik}^{\alpha} = w_k^{\alpha} \quad & \text{if } i \in c_k, \quad i = 1, \dots, n \\ w_k \geq 0, \qquad & \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1 \end{aligned} \tag{4.9}$$

Note that in MinMax k-Means, we maximize ε_w with respect to the cluster weights (w_k-s), so we must minimize $\sum_{j=1}^{p} BCSS(\mathcal{G})_j$ with respect to cluster weights.

$BCSS(\mathcal{G})_j, j = 1, \dots, p$ is a function that is only related to the jth feature. Thus we can conclude that MinMax k-Means satisfies the framework (4.1). According to Witten and Tibshirani's [26] sparse clustering framework (4.2), the MinMax k-Means can be generalized to the following model:

$$\begin{aligned} \max_{\mathcal{G},c,\boldsymbol{\omega}} \quad \min_{\boldsymbol{w}} \quad & F(\boldsymbol{w}, c, \boldsymbol{\omega}) = \boldsymbol{\omega}^{\mathrm{T}} BCSS(\mathcal{G}) \\ \text{s.t.} \qquad & \|\boldsymbol{\omega}\|_2 \leq 1, \quad \|\boldsymbol{\omega}\|_1 \leq s, \quad \omega_j \geq 0, \forall j \\ & z_{ik} = \begin{cases} w_k & if \quad i \in c_k \\ 0 & otherwise \end{cases} \\ \text{with } z_{ik}^{\alpha} = w_k^{\alpha} \quad & \text{if } i \in c_k, \quad i = 1, \dots, n \\ w_k \geq 0, \qquad & \sum_{k=1}^{K} w_k = 1, \qquad 0 \leq \alpha < 1 \end{aligned} \tag{4.10}$$

We will call (4.10) as the Sparse MinMax k-Means model.

We denote $aj = BCSS(G)j$, $\mathbf{a} = (a_1, \dots, a_p)^{\mathrm{T}}$ and the objective function of (4.10) as $F(\boldsymbol{w}, c, \boldsymbol{\omega}) = \sum_{j=1}^{p} \omega_j a_j$. We apply the alternative iteration technique (similar to the one used in [26]) to construct an algorithm to solve the problem (4.10). We first fix $\boldsymbol{w}$ and $\boldsymbol{\omega}$ and maximize $F(c)$ with respect to c, and then we fix $\boldsymbol{\omega}$ and c and minimize $F(\boldsymbol{w})$ with respect to $\boldsymbol{w}$. Finally, we fix $\boldsymbol{w}$ and c and maximize $F(\boldsymbol{\omega})$ with respect to $\boldsymbol{\omega}$. The first two steps have been explained in detail in the Section 4.2.2. The optimization problem that arises in the final step can be written as

$$\begin{aligned} \underset{\boldsymbol{\omega}}{\text{maximize}} \quad & \boldsymbol{\omega}^{\mathrm{T}} a \\ \text{s.t.} \quad \|\boldsymbol{\omega}\|_2 \leq 1, \quad & \|\boldsymbol{\omega}\|_1 \leq s, \quad \omega_j \geq 0, \forall j \end{aligned} \tag{4.11}$$

As per the Proposition stated in [26], the solution to the convex problem (4.11) is $\boldsymbol{\omega} = \frac{S(\boldsymbol{a}_+,\triangle)}{\|S(\boldsymbol{a}_+,\triangle)\|_2}$, where x_+ denotes the positive part of x and where $\triangle = 0$ if that results in $\|\boldsymbol{\omega}\|_1 < s$; otherwise, $\triangle > 0$ is chosen to yield $\|\boldsymbol{\omega}\|_1 = s$. S is the soft-thresholding operator, defined as $S(x, c) = sign(x)(|x| - c)_+$. The assumptions are same as in [26], i.e., there is a unique maximal element of $\boldsymbol{a}$, and $1 \leq s \leq \sqrt{p}$.

4.2.2 The Algorithm

From our definitions of WCSS and BCSS in Section 4.2.1, we see that maximizing the BCSS is equivalent to minimizing the WCSS and vice-versa. This property can be used to simplify our algorithm.

Input: Data matrix X, number of classes K, and 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. Iterate until convergence:
   (a) Holding $\boldsymbol{w}$ and $\boldsymbol{\omega}$ fixed, optimize (4.10) with respect to $c_1, c_2, \ldots, c_K$. Hence maximize (4.8) which is same as minimizing (4.7). That is,

   $$\underset{c_1,c_2,\ldots,c_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{w_k^\alpha}{n_k} \sum_{i,i' \in c_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

   (b) Holding $\boldsymbol{\omega}$ and c fixed, we optimize (4.10) with respect $\boldsymbol{w}$. Hence minimize (4.8) which is same as maximizing (4.7) with respect to $\boldsymbol{w}$. That is,

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

   Both steps (a) and (b) can be done by applying the standard MinMax k-Means clustering algorithm to the updated data matrix with (i, j) element $= \sqrt{\omega_j}.x_{ij}$. The cluster assignments are made as stated in (2.5) and cluster weights are assigned as in (2.8).

   (c) Holding $c_1, c_2, \ldots, c_K$ and $w_1, w_2, \ldots, w_K$ fixed, optimize (4.10) with respect to $\boldsymbol{\omega}$ by applying the Proposition: $\boldsymbol{\omega} = \frac{S(\boldsymbol{a}_+,\triangle)}{\|S(\boldsymbol{a}_+,\triangle)\|_2}$ where $a_j = BCSS(\mathcal{G})_j$ and $\triangle = 0$ if that results in $\|\boldsymbol{\omega}\|_1 < s$; otherwise, $\triangle > 0$ is chosen so that $\|\boldsymbol{\omega}\|_1 = s$.
3. The clusters are given by $c_1, c_2, \ldots, c_K$, the cluster weights by $w_1, w2, \ldots,$ w_k and the feature weights corresponding to this clustering are given by $\omega_1, \omega_2, \ldots, \omega_p$.

Steps 2(a) and 2(b) can be optimized together by performing MinMax k-means on the data after scaling each feature j by $\sqrt{\omega_j}$. Iterate Step 2 until the stopping criterion

$$\frac{\sum_{j=1}^{p}\left|\omega_j^{new}-\omega_j^{old}\right|}{\sum_{j=1}^{p}\left|\omega_j^{old}\right|}<10^{-4} \tag{4.12}$$

is satisfied. We should mention that although the iterative technique used in the Algorithm (Section 4.2.2) is not guaranteed to converge to the global optimum, the objective function will increase monotonically and achieve the local maximal value.

4.2.3 Selection of tuning parameters

Following a procedure similar to [26], we apply a permutation technique and calculate the gap statistic [22] to select s, and α is chosen by using the data driven approach of MinMax k-Means as mentioned in Section 2.3.4 [23]. The value α_{init} is taken as 0. The values of α_{step} and α_{max} are taken as 0.01 and 0.5 respectively, as in [23]. The method to select value of s for Sparse MinMax k-Means is discussed below.

Algorithm to select tuning parameter s by gap statistics

1. Obtain permuted datasets $X_1, \ldots, X_B$ by independently permuting the observations within each feature.

2. For each candidate tuning parameter value s:

   (a) Compute $O(s)=\sum_j \omega_j a_j$, the objective obtained by performing Sparse MinMax k-Means with tuning parameter value s on the data X.

   (b) For $b=1,2,\ldots,B$, compute $O_b(s)$, the objective obtained by performing Sparse MinMax k-Means with tuning parameter value s on the data X_b.

   (c) Calculate $Gap(s)=\log(O(s))-\frac{1}{B}\sum_{b=1}^{B}\log(O_b(s))$.

3. Choose s^* corresponding to the largest value of $Gap(s)$.

Chapter 5

Performance Analysis of Sparse MinMax $\boldsymbol{k}$-Means

5.1 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 updation step which can be done in $O(nK)$, hence the algorithm also has a complexity of $O(npKi)$. To update $\boldsymbol{\omega}$ in step 2(c), we have to solve (4.11) by the Dichotomy searching scheme numerically. The main computational complexity of solving (4.11) is $O(p \log 1/\epsilon)$ where ϵ is the required error for searching. Therefore, the computational complexity of each iteration of the proposed sparse MinMax k-Means algorithm is $O(npKi) + O(p \log 1/\epsilon)$. We iterate until the convergence criteria (4.12) is met.

5.2 Experimental Study

In this section, we evaluate and compare the performance of Sparse MinMax k-Means mainly with the Sparse k-Means [26] and the IF-PCA method [12]. We use the Clustering Error Rate (CER) measure for comparisons in our study, which is defined as $CER \triangleq \sum_{i>i'} \left| \mathbf{1}_{\hat{\mathcal{G}}(i,i')} - \mathbf{1}_{\mathcal{G}(i,i')} \right| / \binom{n}{2}$, where $\mathbf{1}_{\mathcal{G}(i,j)}$ is an indicator function to record whether the ith and jth observations are in the same cluster with respect to the partition $\mathcal{G}$.

We first evaluate the performance of our scheme on synthetic 2-D shape data sets [8], so that the results can be visualized. We then compare the performance of our scheme on 4 UCI real world data sets [1]. Finally we do a detailed comparative study on the 10 high-dimensional gene microarray data sets [11]. Each of the simulations is repeated 30 times.

5.2.1 Description of data sets used in our study

Table 5.1: Synthetic 2-D shape data sets

Dataset Name	K	n	p
Aggregation	7	788	2
Compound	6	399	2
Flame	2	240	2
Jain	2	373	2
Pathbased	3	300	2
Spiral	3	312	2

Table 5.2: UCI real world data sets

Dataset Name	K	n	p
Iris	5	150	4
Wine	2	178	13
Libras Movement	15	360	90
Gesture Phase Segmentation	7	1747	50

Table 5.3: Gene microarray data sets (p: number of genes, n: number of subjects)

#	Dataset Name	K	n	p
1	Brain	5	42	5597
2	Breast Cancer	2	276	22215
3	Colon Cancer	2	62	2000
4	Leukemia	2	72	3571
5	Lung Cancer(1)	2	181	12533
6	Lung Cancer(2)	2	203	12600
7	Lymphoma	3	62	4026
8	Prostate Cancer	2	102	6033
9	SRBCT	4	63	2308
10	SuCancer	2	174	7909

5.2.2 Evaluation on Synthetic 2-D shape data sets

We use 6 synthetic 2-D shape data sets [8], the details of which are given in Table 5.1. We apply the proposed Sparse MinMax k-Means algorithm to these data sets and plot the results. The results obtained on two of the data sets *Aggregation* and *Flame* 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 applying Sparse MinMax k-Means.

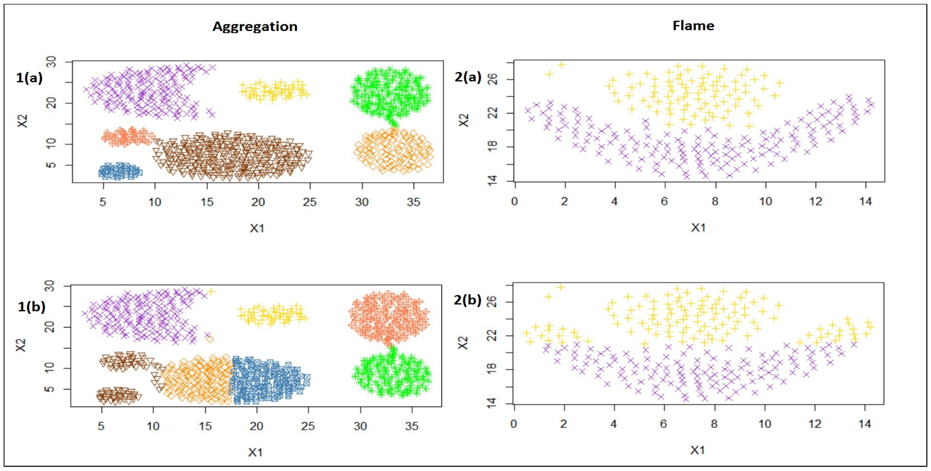


Figure 5.1: Sparse MinMax k-Means results on synthetic 2-D shape data sets

We apply Sparse MinMax k-Means algorithm with $\alpha_{max} = 0.5$, $\alpha_{step} = 0.01$ and $\beta = 0$ following the suggestions in [23]. The value of the tuning parameter s, obtained by using Gap statistics as mentioned in Section 4.2.3, is not very important for these sets as both the features are important for clustering and non-zero weights are assigned for both features in all the cases. We also apply the Sparse k-Means and the IF-HCT-PCA algorithms on these sets. The Clustering Error Rates (CERs) obtained for all the three methods is given in the Table 5.4. From the table we can see that Sparse MinMax k-Means produces lower error rates than both the other methods for all the 6 data sets. In few of the cases, other methods also produce the same lowest error rate as our scheme.

5.2.3 Evaluation on UCI real world data sets

We use the 4 real world data sets mentioned in Table 5.2 obtained from the UCI machine learning repository [1]. The *Iris* and the *Wine* data sets have relatively less number of features as compared to the *Libras Movement* and the *Gesture Phase Segmentation* data sets.

Here we apply our Sparse MinMax k-Means algorithm with $\alpha_{max} = 0.5$ and $\alpha_{step} = 0.01$, and we use β values of $0, 0.1, 0.3$. The best of the three results is noted. The value of the tuning parameter s is estimated using the Gap statistics. The comparisons with Sparse k-Means and the IF-HCT-PCA algorithm for these data sets is given in Table 5.5. In three out of the four cases, our scheme out performs the other two methods. For the *Libras Movement* data set, the lowest clustering error rate is obtained by the Sparse k-Means algorithm which is only 0.031 less than that obtained by our scheme.

Table 5.4: Comparison of Clustering Error Rates (CERs) obtained by the Sparse k-Means, IF-HCT-PCA, and Sparse MinMax k-Means methods for the 2-D Shape data sets introduced in Table 5.1

Dataset Name	Sparse k-Means	IF-HCT-PCA	Sparse MinMax k-Means
Aggregation	0.193	0.457-0.534	**0.183**
Compound	**0.341**	0.506-0.521	**0.341**
Flame	**0.150**	0.196	**0.150**
Jain	0.345	0.137	**0.086**
Pathbased	0.333	**0.240**	**0.240**
Spiral	**0.577**	0.657	**0.577**

Table 5.5: Comparison of CERs obtained by the Sparse k-Means, IF-HCT-PCA, and Sparse MinMax k-Means methods for the UCI data sets introduced in Table 5.2

Dataset Name	Sparse k-Means	IF-HCT-PCA	Sparse MinMax k-Means
Iris	0.067	0.187	**0.053**
Wine	0.297	0.140	**0.050**
Libras Movement	**0.075**	0.094	0.106
Gesture Phase Segmentation	0.574	**0.376**	**0.376**

Table 5.6: Comparison of CERs obtained by different methods for the 10 gene microarray data sets introduced in Table 5.3. Column 3: numbers in the brackets are the standard deviations (SD); SD for all other methods are negligible so are not reported

Dataset Name	kmeans	kmeans++	Hier	SpecGem	IF-HCT-PCA	Sparse k-Means	Sparse MinMax k-Means
Brain	0.286	0.427(.09)	0.524	**0.143**	0.262	0.214	0.238
Breast Cancer	0.442	0.430(.05)	0.500	0.438	**0.406**	0.449	0.417
Colon Cancer	0.443	0.460(.07)	0.387	0.484	0.403	0.306	**0.145**
Leukemia	0.278	0.257(.09)	0.278	0.292	0.069	0.139	**0.028**
Lung Cancer(1)	0.116	0.196(.09)	0.177	0.122	**0.033**	0.122	0.122
Lung Cancer(2)	0.436	0.439(.00)	0.301	0.434	**0.217**	0.315	**0.217**
Lymphoma	0.387	0.317(.13)	0.468	0.226	0.065	**0.032**	0.274
Prostate Cancer	0.422	0.432(.01)	0.480	0.422	0.382	0.392	**0.372**
SRBCT	0.556	0.524(.06)	0.540	0.508	0.444	0.349	**0.333**
SuCancer	0.477	0.459(.05)	0.448	0.489	0.333	**0.328**	**0.328**

5.2.4 Evaluation on gene microarray data sets

Our study is mainly related to the 10 high-dimensional gene microarray data sets [11] given in Table 5.3. The data sets include patients from several classes (normal, diseased), and for each patient, we have measurements (gene expression levels) on the same set of genes. The classes are predicted by a clustering algorithm. These data sets are the ones that have been used by Jiashun and Wang [12] in their study. These data sets belong to the modern regime of $p >> n$ and the value of K is also small. In [12], a comparison study was done of IF-HCT-PCA with other well-known clustering algorithms on these 10 data sets. Using their code, we obtain the clustering error rates for IF-HCT-PCA and also the other algorithms. We now add two more columns to the comparison table, which are the error rates obtained by Sparse k-Means and our Sparse MinMax k-Means.

Similar to Section 5.2.3, here also we apply our algorithm with $\alpha_{max} = 0.5$ and $\alpha_{step} = 0.01$, and we use β values of $0, 0.1, 0.3$. The best of the three results is noted for each data set. The comparison table of our scheme with IF-HCT-PCA, Sparse k-Means and other well-known clustering algorithms including the classical k-means, k-means++, Hierarchical clustering and the Spectral GEM algorithm is given in the Table 5.6. On 6 out of the 10 data sets, our scheme obtains the lowest CER among all the other algorithms. Sparse MinMax k-Means obtains lower or same CER as that obtained by IF-HCT-PCA in 7 out of 10 cases and in 8 out of 10 cases for Sparse k-Means.

Table 5.7: CER, Number of Non-zero weights out the p weights (denoting number of useful features), s values obtained from the Gap statistics method for our Sparse MinMax k-Means Model for the gene microarray data sets

Dataset Name	CER	# Non-zero weights/p	s value from Gap stats
Brain	0.238	1810/5597	27.51
Breast Cancer	0.417	79/22215	6.26
Colon Cancer	0.145	76/2000	5.69
Leukemia	0.028	148/3571	7.27
Lung Cancer(1)	0.122	16/12533	2.84
Lung Cancer(2)	0.217	5/12600	1.61
Lymphoma	0.274	717/4026	14.22
Prostate Cancer	0.372	5650/6033	41.41
SRBCT	0.333	1019/2308	6.04
SuCancer	0.328	1370/7909	14.97

We apply the Gap statistics method explained in Section 4.2.3 to obtain the value of the tuning parameter s. For data sets with n values less than 100, we use number of permutations $B = 10$, and for others we use $B = 20$, for estimating the value of s. The number of non-zero feature weights ω_j out of the p weights would signify the number of features useful for clustering as determined by our algorithm. These values, along with the values of the tuning parameter s, as obtained from the Gap statistics method for the gene microarray data sets, is listed in the Table 5.7. In few of the cases, we see that a lower s value renders the same clustering as that given by s^* obtained from Gap statistics (see Section 4.2.3). Hence for them, a lower number of features can be considered to be effective for clustering than the number of non-zero weights obtained by using the value of s from Gap statistics.

Table 5.8: 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 data sets

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**

For analyzing the performance of our algorithm, we evaluate it's performance based on two more criteria and compare the results with IF-HCT-PCA and Sparse k-Means in Table 5.8. Dunn Index (DI) is the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance. 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. Retained Features (RF) is the number of retained features in each of the algorithms or basically the number of non-zero feature weights (for sparse algorithms). 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 are however obtained in only 4 cases by our method but it has the best CER values in most cases, which is our main criteria.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Clustering data in the high-dimensional setting is challenging due to the existence of abundant noisy features. In our study, which is inspired by the literature of sparse clustering, we proposed a novel MinMax k-Means with sparse regularization. Based on the framework, we developed an efficient algorithm to solve the model and named it as the Sparse MinMax k-Means algorithm. Our algorithm is based on Witten and Tibshirani's [26] *sparse clustering framework*.

The experimental results obtained in Section 5.2 confirmed the outperformance of our approach over other approaches like the Sparse k-Means [26] and the Influential Features PCA (IF-PCA) [12] in most of the cases. We have mainly worked with the 10 high-dimensional gene microarray data sets [11]. Our algorithm, outperforms both of them and all other common clustering methods in most of the cases. For the UCI and the synthetic 2-D shape data sets also, our scheme has performed better or equal in most of the cases. In the cases where the other two algorithms have produced better clustering error rates, our scheme is only marginally behind.

For our comparisons, we have used the original codes by Jiashun and Wang [12] for IF-HCT-PCA and the R-package *sparcl* by Witten and Tibshirani [26] for the Sparse k-Means. The code for our implementation of the Sparse MinMax k-Means algorithm is available at *https://github.com/sayak94/Sparse-MinMax-k-Means*.

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_{max}, α_{step} and β included in the MinMax k-Means part, are to be entered by the user and not completely auto tuned. We have used the empirical values for them suggested in [23], but the results often vary with the change in those parameters. We can address this issue by a method suggested by Wang et al. [25] of using PSO (Particle Swarm Optimization) to obtain the α and β values. Secondly, we can also explore and search for more consistent approaches rather than using the Gap statistics to estimate the value of the tuning parameter s. Finally, we can look to improve the time complexity of our algorithm by using an approach similar to the one suggested in [5] for sparse FCM, where rather than going up to convergence in the MinMax k-Means part in each iteration, we can use only a single run of the algorithm in each iteration. This would significantly reduce the complexity of our algorithm.

Bibliography

[1] Bache, K., Lichman, M.: Uci machine learning repository (2013), http://archive.ics.uci.edu/ml

[2] Bagirov, A.M.: Modified global k-means algorithm for minimum sum-of-squares clustering problems. Pattern Recognition 41(10), 3192–3199 (2008)

[3] Bagirov, A.M., Ugon, J., Webb, D.: Fast modified global k-means algorithm for incremental cluster construction. Pattern Recognition 44(4), 866–876 (2011)

[4] Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications 40(1), 200–210 (2013)

[5] Chang, X., Wang, Q., Liu, Y., Wang, Y.: Sparse regularization in fuzzy c -means for high-dimensional data clustering. IEEE Transactions on Cybernetics 47(9), 2616–2627 (2017)

[6] Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means. Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 04 (2004)

[7] Efron, B.: Large-scale simultaneous hypothesis testing. Journal of the American Statistical Association 99(465), 96–104 (2004)

[8] Franti, P.: Clustering basic benchmark (2015), http://cs.uef.fi/sipu/datasets/

[9] Friedman, J.H., Meulman, J.J.: Clustering objects on subsets of attributes (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology) 66(4), 815–849 (2004)

[10] Ghosh, D., Chinnaiyan, A.M.: Mixture modelling of gene expression data from microarray experiments. Bioinformatics 18(2), 275–286 (Jan 2002)

[11] Jin, J., Wang, W.: Gene microarray data sets (2014), https://www.dropbox.com/sh/l5ul024bx48reqj/AACx3-MW9Pm4bKkmxTbxZK0sa/Data?dl=0

[12] Jin, J., Wang, W.: Influential features pca for high dimensional clustering. The Annals of Statistics 44(6), 2323–2359 (2016)

[13] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (Oct 1999)

[14] Li, R., Chang, X., Wang, Y., Xu, Z.: Sparse k-means with ℓ_∞/ℓ_0 penalty for high-dimensional data clustering. Statistica Sinica (2018)

[15] Liu, J. S., Z.J.L.P.M.J., Lawrence, C.E.: Bayesian clustering with variable and transformation selections. Bayesian Statistics 18(7), 249–275

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

[17] Mclachlan, G.J., Bean, R.W., Peel, D.: A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 18(3), 413–422 (Jan 2002)

[18] Mclachlan, G., Peel, D., Bean, R.: Modelling high-dimensional data by mixtures of factor analyzers. Computational Statistics Data Analysis 41(3-4), 379–388 (2003)

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

[20] Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), 1299–1319 (1998)

[21] Tamayo, P., Scanfeld, D., Ebert, B.L., Gillette, M.A., Roberts, C.W.M., Mesirov, J.P.: Metagene projection for cross-platform, cross-species characterization of global transcriptional states. Proceedings of the National Academy of Sciences 104(14), 5959–5964 (2007)

[22] Tibshirani, R., Walther, G., Hastie, T.: 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)

[23] Tzortzis, G., Likas, A.: The minmax k-means clustering algorithm. Pattern Recognition 47(7), 2505–2516 (2014)

[24] Wang, S., Zhu, J.: Variable selection for model-based high-dimensional clustering and its application to microarray data. Biometrics 64(2), 440–448 (2008)

[25] Wang, X., Bai, Y.: A modified minmaxk-means algorithm based on pso. Computational Intelligence and Neuroscience 2016, 1–13 (2016)

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

[27] Xie, B., Pan, W., Shen, X.: Variable selection in penalized model-based clustering via regularization on grouped parameters. Biometrics 64(3), 921–930 (2007)

[28] Xie, B., Pan, W., Shen, X.: Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. Electronic Journal of Statistics 2, 168–212 (2008)

[29] Xie, J., Jiang, S.: A simple and fast algorithm for global k-means clustering. 2010 Second International Workshop on Education Technology and Computer Science (2010)

[30] Xu, R., Wunschii, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)