Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrie en Operationele Research

Refining Clustering with Feature Selection in High-dimensional Mixed Datasets

Yuan Zhang (581010)

Supervisor:	Cavicchia, C
Second assessor:	
Date final version:	2nd July 2023

Abstract

In this thesis, we address the problem of feature selection in clustering by extending the framework proposed by Witten and Tibshirani (2010). This framework is particularly useful for datasets with a large amount of features, where the true underlying clusters differ only with respect to small fraction of the features. However, an area for improvement lies in the fact that their proposed methods only cover numerical datasets. Our contribution lies in adapting the framework to handle datasets containing categorical variables or a mix of categorical and numerical variables. Through simulation experiments and analysis of gene expression data, we show that our proposed methods can perform both clustering and feature selection, resulting in meaningful clusters with descriptive features, both numerical and categorical, for characterization.

1 Introduction

Clustering analysis is a widely used technique in data mining and pattern recognition, aimed at identifying natural groupings or clusters within a dataset. It tries to uncover patterns and relationships in the data, facilitating data exploration and decision-making processes. However, standard cluster analysis methods such as K-means or K-prototypes group observations using the full set of features. One might expect that the true underlying clusters differ with respect to a small fraction of the features. Faced with this problem, Witten and Tibshirani (2010) propose a framework for sparse clustering, where observations are clustered using a selectively determined subset of features. In light of the simplicity and effectiveness of this project, this thesis project build upon the work of Witten and Tibshirani (2010). One area for improvement in this paper is to consider addressing datasets that specifically consist of categorical data or mixed type data. Consequently, we investigate the following research question: "How can the feature selection framework in clustering proposed by Witten and Tibshirani (2010) be extended to handle datasets that contain either only categorical variables or a mix of numerical and categorical data?".

The framework can be summarized as an iterative process that combines a standard clustering method with feature shrinkage until the feature weights stabilize. As such, we can use a standard clustering method that can handle numerical and categorical data. For datasets with mixed-type data, we incorporate the K-prototype algorithm introduced by Huang (1998). Alternatively, for datasets that only consists of categorical variables, we employ the K-modes algorithm by Huang (1997b). Additionally, we explore different dissimilarity measures suitable for the K-modes and K-prototypes methods, including the dissimilarity schemes introduced by Sangam and Om (2018) and Jia and Song (2020).

The findings indicate that by integrating the framework proposed by Witten and Tibshirani (2010) with the cluster algorithms K-modes and K-prototypes, the accuracy of clustering can be enhanced, particularly when dealing with datasets that contain numerous redundant features, as opposed to using these K-modes and K-prototypes independently. Furthermore, our sparse clustering methods improve interpretability of clusters by selecting a subset of important features.

The contributions of this paper are as follows. We propose sparse clustering methods that are capable of handling datasets consisting only of categorical data, as well as datasets that con-

tain both numerical and categorical data. Additionally, we examine the performance of hybrid dissimilarity measures proposed in the literature for the K-prototypes model in the context of sparse clustering.

The remainder of the paper is structured as follows. In Section 2, we provide an overview of relevant literature on sparse clustering and methods to handle mixed type data. Then, section 3 mentions the data that is used in this study and how it is obtained. The methods used in this study are detailed in Section 4. Furthermore, in Section 5 we present our findings with a discussion. In Section 6 we draw conclusions based on our result. Additionally, we discuss the limitations of our study and directions for further research.

2 Literature

This thesis touches upon two aspects of cluster analysis. Firstly, we consider a setting where there are a large amount of features p. It is likely that not all features are equally important for the clustering and hence we seek to "filter out" all the non-important features. Secondly, we address datasets which contain both numerical and categorical data.

One idea to reduce the amount of variables in a dataset is through dimension reduction. The original $n \times p$ dataset matrix $\mathbf{X}$, where n denotes the observations and p the features, is approximated as $\mathbf{X} \approx \mathbf{AB}$. Then, the $n \times q$ matrix $\mathbf{A}$ can be used instead of $\mathbf{X}$. Examples of dimension reduction methods are principal component analysis (PCA) and singular value decomposition (SVD). Other papers introduce their own dimension reduction methods. Roweis and Saul (2000) introduce locally linear embedding (LLE) which is based on local symmetries of linear reconstructions. Lee and Seung (1999) propose dimension reduction through non-negative matrix factorization which uses non-negativity constraints. However, Witten and Tibshirani (2010) note that these methods have some drawbacks. Namely, the reduced matrix $\mathbf{A}$ is not sparse as it is a function of the full set of p features and there is no insurance that $\mathbf{A}$ contains the signal that one is interesting in detecting via clustering. In fact, Chang (1983) states that the practice of performing PCA is not justified in general and shows that components with larger eigenvalues do not necessarily contain more information about the cluster structure of data. Śmieja et al. (2019) also note that dimensionality reduction leads to a loss of information about the original data. Hence, this thesis puts more focus on sparsity instead of dimensionality reduction.

There are various papers that focus on addressing feature sparsity. One approach is to model the observations in a data matrix $\mathbf{X}$ from a mixture model, usually a mixture of Gaussian (GMM). Pan and Shen (2007) encourage sparsity by maximizing a log-likelihood function subject to a LASSO-type penalty that is chosen to yield sparsity in features. A similar concept is used in the methods in this paper. While the model is well-suited for feature selection, problems can arise when a GMM has to estimate a $p \times p$ covariance matrix when $p \gg n$. Nonetheless, there are ways to adress this problem.

Friedman and Meulman (2004) introduce Clustering Objects on Subsets of Attributes (COSA) to cluster attribute value data. This method also minimizes an objective function subject to constraints on the feature weights. However, this method does not ultimately result in sparse clustering as none of the coefficients will be set to zero.

Instead, Witten and Tibshirani (2010) propose a general framework which can be thought of as a simpler version of the COSA proposal, capable of sparse clustering. The framework provides a general approach for obtaining sparse versions of various clustering methods. In this thesis, we build upon this frame due to its effectiveness and simplicity. In particular, we employ the framework to develop a novel approach for sparse clustering in datasets that contain both numerical and categorical data.

In practise, datasets usually contain both numerical and categorical variables. As categorical variables may contain valuable information, it is crucial that methods can handle a mixture of both data types. Chavent et al. (2022) extend the framework by Witten and Tibshirani (2010) to handle categorical data through group sparse K-means with a L_1 group penalty. Specifically, the data is split into two groups: categorical and numerical. The categorical variables are dummy coded, and all data is scaled. However, current work is still done one finding appropriate criteria for the group penalty.

Rather, this paper takes a different approach. We employ the K-prototypes model by Huang (1997a), which is an extension of the K-means method that can handle numerical and categorical variables. The K-prototypes model sets a pre-defined dissimilarity measure for numerical data, usually squared Euclidean distance, and utilizes a weighted simple matching scheme for categorical variables. However, this approach may still be sensitive to numerical variables with higher variance. Consequently, this thesis also explores alternative dissimilarity measures. These dissimilarity metrics aim to address the problem by establishing a common scale for all variable types. For instance, Sangam and Om (2018) introduce a hybrid dissimilarity coefficient that operates on the same scale for both categorical and numerical features. Moreover, this coefficient preserves the clustering characteristics of high intra-cluster similarity and low inter-cluster similarity. Similarly, Jia and Song (2020) present a novel hybrid dissimilarity coefficient designed for the K-prototype algorithm. They argue that this measure retains the characteristics of different types of data as well, and effectively improves the clustering accuracy and clustering effectiveness.

Additionally, if the dataset consists of only categorical variables, then the K-modes by Huang (1997b) can be employed. The K-modes algorithm essentially functions as a variation of the K-prototypes model but exclusively handles categorical variables.

Hence, there are multiple approaches that can potentially lead to sparsity in features for cluster analysis. In this thesis, we build upon the work of the framework of Witten and Tibshirani (2010) to achieve sparse clustering. Then, by utilizing the K-prototypes algorithm by Huang (1997a) we aim to achieve sparse clustering for mixed type data. Alternatively, by employing the K-modes method outlined in Huang (1997b), we aim to achieve clustering for categorical datasets. Furthermore, different dissimilarity measures are explored to determine which measure fits best in our application.

3 Data

One aim of this thesis is to replicate parts of the paper Witten and Tibshirani (2010). Specifically, we try to replicate the application of the sparse K-means method on single nucleotide polymorphism (SNP) data. The original study uses sparse clustering in order to identify distinct

populations in SNP data. The dataset is part of the third phase of the International Hap Map project. The data can be obtained here `https://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2008-07_phaseIII/hapmap_format/forward/`. This phase covers 1301 samples from different human populations. However, in line with the approach taken by Witten and Tibshirani (2010), our analysis will specifically concentrate on a subset of 315 samples, focusing exclusively on data from three specific populations: 71 samples represent African ancestry in Southwest USA, 162 samples represent Utah residents with Northern and Western European ancestry from the CEPH collection, and 82 samples represent Han Chinese in Beijing, China. Furthermore, only SNP data which is available in all three populations is used. This results in a data set with dimensions 315×17026. Then, we code AA as 2, Aa as 1, and aa as 0.

However, the data contains missing values. Witten and Tibshirani (2010) impute the missing values via a method called weighted K-nearest neighbors. The method is introduced by Troyanskaya et al. (2001), and we refer to this paper for more information about the method. We used the publicly available R package "pamr" to impute the missing values, available from `https://cran.r-project.org/web/packages/pamr/index.html`

Furthermore, we simulate data to assess the effectiveness of our proposed methods. Similar to the simulation study in Witten and Tibshirani (2010), the dataset is constructed with three groups, where each group differs in $q = 50$ features. The numerical features are generated with the following distribution $X_{ij} \sim \mathrm{N}(\mu_{ij}, 1)$, where $\mu_{ij} = \mu\,(1_{i\in C_1, J\leq q} - 1_{i\in C_2, J\leq q})$. The categorical features in the simulation data are sampled from $X_{ij} = [x_1, x_1, x_3]$, where $X_{ij} \sim P(X_{ij})$. Specifically, for $j \leq 50$, the probabilities of X_{ij} taking the values x_1, x_2, and x_3 are denoted as p_1, p_2, and p_3, respectively. For features with indices $j > 50$, the values were randomly sampled from $X_{ij} = [x_1, x_1, x_3]$ with equal probabilities $p_1 = p_2 = p_3$. The objective is to evaluate the performance of the methods under increased feature dimensionality, to assess their ability to handle noise.

Moreover, we use a dataset that contains information about gene expressions, which originates from a proof-of-concept study published by Golub et al. (1999). These data were used to classify patients with acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The dataset contains measurements corresponding to AML and ALL samples from Bone Marrow and Peripheral Blood. Hence, our objective is to accurately cluster patients with AML and ALL by identifying the most crucial genes associated with these diseases. The dataset is available from `https://www.kaggle.com/datasets/crawford/gene-expression?select=data_set_ALL_AML_independent.csv`. It contains both an initial dataset with dimensions 38×1648 and an independent dataset with dimensions 34×6418, which were both used in Golub et al. (1999). The dataset contains a numerical and categorical value for each gene expression. The numerical values represent gene expression levels that have been re-scaled to ensure comparability across samples. These values provide quantitative measurements indicating the intensity of gene activity. The categorical call columns classify genes as either present (P), absent (A), or marginal (M) based on the hybridization signal intensity, indicating the likelihood of gene expression. For a more detailed understanding of the call columns, we refer to Liu et al. (2002).

4 Methodology

Here, we introduce the methods used for our research. To begin, the sparse clustering framework by Witten and Tibshirani (2010) is discussed in Section 4.1. We then discuss the tuning of the hyper parameter in this framework in Section 4.2. This framework is then applied to the K-means algorithm in Section 4.3. Next, Section 4.4 describes the K-prototypes model and Section 4.5 addresses the tuning of the hyper parameter in K-prototypes. We then present alternative dissimilarity measures for the K-prototypes model in Section 4.6. Specifically, Section 4.6.1 describes the hybrid dissimilarity measure by Sangam and Om (2018) and section 4.6.2 covers the dissimilarity measure by Jia and Song (2020). Then, we present the algorithm for the sparse K-prototypes method in Section 4.7. Furthermore, the algorithm for sparse K-modes is introduced in Section 4.8. And finally, Section 4.9 discusses the cluster center initialization scheme by Ji et al. (2015).

4.1 Framework

Witten and Tibshirani (2010) introduce a general approach to the problem of sparse clustering. Suppose we wish to cluster n observation on p dimensions, where $\mathbf{X}$ is an $n \times p$ data matrix. Let $\mathbf{X}_j \in \mathbb{R}^n$ indicate feature j. Then, sparse clustering can then be seen as a solution to the following problem

$$\max_{\mathbf{w};\Theta \in D} \quad \left\{ \sum_{j=1}^{p} w_j f_j\left(\mathbf{X}_j, \Theta\right) \right\} \tag{1}$$

$$\text{subject to} \quad ||\mathbf{w}||^2 \leq 1, \quad ||\mathbf{w}||_1 \leq s, \quad w_j \geq 1 \quad \forall j, \tag{2}$$

where w_j is a weight for feature j, $f_j\left(\mathbf{X}_j, \Theta\right)$ is a function that only involves feature j, Θ is a parameter which restricted in a set D, and s is a tuning parameter, $1 \leq s \leq \sqrt{p}$. The value of w_j can be seen as the importance of feature j for the sparse clustering. A high value of w_j implies that the feature plays a larger role in the sparse clustering, whereas a low value means the opposite. The L_1, or LASSO, penalty in in (2) on $\mathbf{w}$ results in sparsity for small values of s, while the L_2, or Ridge, penalty ensures that, in general, there are multiple non-zero elements in $\mathbf{w}$. Furthermore, in (1) it is necessary that $f_j\left(\mathbf{X}_j, \Theta\right) > 0$, since w_j would simply be 0 if $f_j\left(\mathbf{X}_j, \Theta\right) \leq 0$.

A suitable choice for $f_j\left(\mathbf{X}_j, \Theta\right)$ in our methods is the average between-cluster distance. It is defined as follows

$$f_j\left(\mathbf{X}_j, \Theta\right) = \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right), \tag{3}$$

where $d_{i,i',j}$ denotes the dissimilarity between observation i and observation i' along feature j. The first part in this equation denotes the average distance between all observations, while the second part denotes the average distance between all observation in their respective cluster. Intuitively, important features in clustering should have a greater between-cluster distance than less important features, and thus they should be assigned a higher weight.

Moreover, the problem in (1) subject to (2) is solved iteratively for $\mathbf{w}$ and $f_j(\mathbf{X}_j, \Theta)$. For fixed $\mathbf{w}$, $f_j(\mathbf{X}_j, \Theta)$ is typically optimized using standard clustering procedure to a weighted version of the data set. For fixed $f_j(\mathbf{X}_j, \Theta)$, $\mathbf{w}$ is optimized according to a preposition. The preposition in Witten and Tibshirani (2010) states that the solution to this convex problem is $\mathbf{w} = \frac{S(\mathbf{a}_+, \Delta)}{||S(\mathbf{a}_+, \Delta)||_2}$, where $\mathbf{a}_+$ denotes the positive part of $\mathbf{a}$. Here, $\mathbf{a}$ is a vector which contains the values of $f_j(\mathbf{X}_j, \Theta)$ for every feature j. Moreover, the soft-thresholding operator S is defined as $S(x, c) = \text{sign}(x)(|x| - c+)_+$. If $||\mathbf{w}||_1 \leq s$, then $\Delta = 0$; otherwise, $\Delta > 0$ is chosen such that $||\mathbf{w}||_1 = s$.

The resulting algorithm is as follows for maximizing (1) subject to constraints 2.

1. Initialize $\mathbf{w} = w_1, \ldots, w_k = \frac{1}{\sqrt{p}}$.
2. Iterate until stopping criterion:
   (a) Holding $\mathbf{w}$ fixed, solve the following with respect to the clusters $C_1, \ldots, C_k$. using a clustering procedure.

$$\min_{C_1,\ldots,C_k} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} w_j d_{i,i',j} \tag{4}$$

   (b) Holding $C_1, \ldots, C_k$ fixed, solve equation (1) with respect to $\mathbf{w}$ via: $\mathbf{w} = \frac{S(a_+, \Delta)}{||S(a_+, \Delta)||_2}$, where

$$a_j = \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right) \tag{5}$$

   and $\Delta = 0$ if that results in $||\mathbf{w}||_1 < s$, otherwise $\Delta > 0$ is chosen such that $||\mathbf{w}||_1 = s$.
3. The weights for the features are stored in $\mathbf{w}$ and cluster assignments are stored in $C_1, \ldots, C_k$.

Throughout the paper we refer to this algorithm to solve the respective problem.

4.2 Parameter tuning sparse framework

The framework for sparse clustering has a tuning parameter s which is the L_1 bound on $\mathbf{w}$. The task of determining the optimal number of clusters K falls beyond the scope of this thesis and is therefore assumed to be fixed. The method to choose s is a permutation approach which is related to the gap statistic of Tibshirani and Walther (2005). Witten and Tibshirani (2010) introduce the following algorithm for the selection of s:

1. Obtain permuted data sets $\mathbf{X}_1, \ldots, \mathbf{X}_B$ through independent permutation of the observations within feature.
2. For every candidate tuning parameter s:
   (a) Compute $O(s) = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j}$, which is the objective after performing sparse K-means with tuning parameter s on data X.
   (b) For $b = 1, \ldots, B$, compute $O_b(s)$, which is the objective after performing sparse K-means with tuning parameter s on data $\mathbf{X}_b$.
   (c) Calculate $\text{Gap}(s) = \log(O(s)) - \frac{1}{B} \sum_{b=1}^{B} \log(O_b(s))$
3. Select s* as the maximum Gap(s) value or, alternatively, choose s* as the smallest value where Gap(s*) is within one standard deviation of $\log(O_b(s))$ from the maximum Gap(s) value.

Witten and Tibshirani (2010) note that the gap-statistic is not always a reliable measure that selects the best clusters. Hence, further research could focus on tuning of the parameter s. One could for example tune s based on the silhouette coefficient. Nonetheless, tuning of the parameter s is outside the scope of this thesis.

4.3 Sparse K-means

We employ K-means to cluster data with numerical data. K-means is a clustering method which aims to partition n observations with p numeric features into k clusters. It minimizes the within-cluster distance as in

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} \sum_{j=1}^{p} d_{i,i',j}, \tag{6}$$

where n_k is the amount of observations in cluster k and C_k contains the indices of observations in cluster k. Generally, $d_{i,i',j}$ can denote any dissimilarity measure between observations i and i' along feature j. However, in this thesis, the squared Euclidean distance is used, $d_{i,i',j} = \left(x_{i,j} - x_{i',j}\right)^2$.

It is relatively straightforward to employ the K-means method in the framework for sparse clustering. In the algorithm denoted in Section 4.1, Equation 4 is minimized using standard K-means algorithm. Witten and Tibshirani (2010) note that the algorithm generally does not result in a global optimum of criterion 1, since the criterion is convex and uses K-means in Step 2(a), which is not guaranteed to result in a global optimum.

Witten and Tibshirani (2010) have implemented this algorithm for sparse K-means in a publicly available R package, available from `https://cran.r-project.org/web/packages/sparcl/index.html`.

4.4 K-prototypes

To handle the combination of numerical and categorical data, we employ the K-prototypes method by Huang (1997a). This approach is an extension upon the standard K-means method designed to handle both types of data. It aims to minimize the same objective in standard K-means (6). The difference lies in the fact that the dissimilarity measures are defined in different ways for numerical and categorical variables. Assume that there are m numerical variables and $p - m$ categorical variables. The dissimilarity measure between observations i and i', is defined as follows in K-prototypes

$$d_{i,i',j} = \sum_{j=1}^{m} \left(x_{i,j} - x_{i',j}\right)^2 + \gamma \sum_{j=m+1}^{p-m} \delta(x_{i,j}, x_{i',j}), \text{ with } \delta(x_{i,j}, x_{i',j}) = \begin{cases} 0 & \text{if } x_{i,j} = x_{i',j} \\ 1 & \text{if } x_{i,j} \neq x_{i',j} \end{cases} \tag{7}$$

Here, δ is the simple matching dissimilarity, and γ is a tuning parameter weight which controls for the trade off between the Euclidean distance and simple matching distance δ for factor variables. By increasing γ, the algorithm puts more focus on minimizing categorical variables.

4.5 Parameter tuning K-prototypes

For balanced contributions in cluster analysis, uniform scaling across all features is preferred as this prevents the variables with higher variances dominating over those with smaller variances. It is thus important to define γ on the same scale. Huang (1997a) suggests setting γ around the average standard deviation σ of the numeric variables. However, we will use the method introduced by Szepannek (2018), which provides different data based heuristics for the choice of γ. They argue that the average variance $\bar{\sigma}^2$ or standard deviation $\bar{\sigma}$ over all numeric variables is related to the concentration c_{cat} of all categorical variables. c_{cat} is computed by either averaging $c_j^1 = 1 - \sum_c p_{jc}^2$ or $c_j^2 = 1 - \max_c p_{jc}$ over all categorical variables where p_{jc} denotes the proportion of a category c for feature j. We can then set the tuning parameter using $\gamma = \frac{\sigma^t}{c_{cat}}$, $t \in \{1, 2\}$. We use $t = 2$ and $c_j = 1$ to calculate γ as this is the default value recommended. Szepannek (2018) notes that that this should be considered a starting point for further analysis; the explicit choice of γ should be done carefully based on the application context.

However, another approach which aims at achieving equal scale between numerical and categorical variables is through different dissimilarity measures. We discuss these in the next sections.

4.6 Alternative dissimilarity measures

In this section, we define two novel hybrid dissimilarity coefficients proposed by two papers. These dissimilarity measures are defined in a manner that ensures the range of dissimilarity is consistent across all variables. We will employ the hybrid dissimilarity measures by Sangam and Om (2018) and Jia and Song (2020).

4.6.1 Hybrid dissimilarity coefficient in Sangam and Om (2018)

Sangam and Om (2018) propose a new hybrid dissimilarity measure for categorical and numerical variables in the context of K-prototypes. Let q_j^k denote the feature j of a prototype k, and $\delta_{j,l}^k$ reflect the amount of observations in cluster k that have the category or level l for feature j. Additionally, let $|c_k|$ indicate the number of observation in cluster k. The dissimilarity measure for categorical variables is a weighted Hamming dissimilarity function adopted for categorical attribute space

$$\zeta_{i,k,j}^c = \begin{cases} 1 - weight(\delta_{j,l}^k) & \text{if } x_{i,j} = q_j^k \\ 1 & \text{if } x_{i,j,l} \neq q_j^k \end{cases} \tag{8}$$

where the weight is given by

$$weight(\delta_{j,l}^k) = \frac{\delta_{j,l}^k}{|c_k|} \frac{\delta_{j,l}^k}{\eta}, \quad \text{where } \eta = \sum_{k=1}^{K} \delta_{j,l}^i \tag{9}$$

The weight function $weight(\delta_{j,l}^k)$ consists of two parts. The first fraction $\frac{\delta_{j,l}^k}{|c_k|}$ considers the relative frequency of the level l for a categorical feature j. As such, if two prototypes have the same level for a feature, this object will be more similar to the cluster with the highest proportion of this level. The second fraction takes the inter-cluster similarity into account, by

calculating the frequency of this level on the same feature for different clusters. The interval for this dissimilarity measure is defined on [0, p]

The dissimilarity measure for numerical data is the Minkowski-distance-based dissimilarity function adopted for numerical attribute space.

$$d^n_{i,k,j} = \sum_{j=1}^{m} \frac{|x_{i,j} - x_{i',j}|}{|max(j) - min(j)|} \tag{10}$$

Here, $max(j)$ and $min(j)$ are the maximum and minimum value of the j-th feature respectively. This dissimilarity measure is defined on the interval [0, $m - p$].

It is now straightforward to obtain a dissimilarity measure for mixed type data. We simply combine the two dissimilarity measures in (8) and (10):

$$d_{i,k,j} = \sum_{j=1}^{m} \frac{|x_{i,j} - q^k_j|}{|max(j) - min(j)|} + \sum_{j=p+1}^{p-m} \zeta_{i,i',j}, \tag{11}$$

Sangam and Om (2018) note that this distance function for mixed data has equal weightage for either-type attributes since it is defined on the same scale with respect to their dimensionality. Furthermore, they suggest to use a standard simple matching scheme for categorical variables in the initial step to speed up the clustering process.

4.6.2 Hybrid dissimilarity coefficient in Jia and Song (2020)

Jia and Song (2020) propose a similar hybrid dissimilarity measure for K-prototypes. The dissimilarity measure for categorical variable is similar to that in Equation (8), however weights are added based on entropy of the features. The entropy measure quantifies the impurity or disorder within a data set. Let $E^k_{n,j}$ denote the entropy for feature j in cluster k and n_j the number of levels for a feature j. Then, the entropy can be computed as follows

$$E_{n,j} = -\frac{1}{n_j} \sum_{l=1}^{n_j} \frac{\delta^k_{j,l}}{|c_k|} \log\left(\frac{\delta^k_{j,l}}{|c_k|}\right) \tag{12}$$

where $\delta^k_{j,l}$ is similar as in Equation (8) and (9). In order to reduce the influence of a categorical feature with many levels on clustering, (12) is divided by the number of levels n_j. Furthermore, as a high entropy value indicates a high impurity for a feature, the influence of this feature should be lowered relative to features with lower entropy measures. First, all entropy values are normalized. Jia and Song (2020) do not mention their normalization process, so we employ $L2$-normalization. Additionally, as the entropy measure ranges between 0 and 1, the weight of a feature is $1 - E_{n,j}$ since a higher entropy measure indicates higher disorder. The weight b_j for feature j is then simply the proportion based on the total weights

$$\beta_j = \frac{1 - E_{n,j}}{\sum_{j=1}^{m} (1 - E_{n,j})} \tag{13}$$

Then, the quantified dissimilarity coefficient for categorical variables between an observation

i and cluster prototype k is a weighted Hamming distance with weights based on entropy

$$d_{i,k,j} = \sum_{j=1}^{m} \beta_j \zeta_{i,k',j} \tag{14}$$

where $\zeta_{i,k',j}$ is the same as in Equation (8).

Furthermore, Jia and Song (2020) argue that direct calculation of data of different orders of magnitude will not only increase the difficulty of calculation, but also cause a large error between the calculated results and the real situation. Hence, Max-Min Standardization is adopted to calculate the quantified dissimilarity coefficient for numerical variables:

$$x'_{i,j} = \frac{x_{i,j} - \min(j)}{\max(j) - \min(j)} \tag{15}$$

$$d_{i,k,j} = \sum_{j=1}^{p} \sqrt{|x'_{i,j} - q_j^k|^2} \tag{16}$$

Jia and Song (2020) state that numerical vectors are treated as a whole (vector) while categorical features are treated as m-dimensional vectors. Hence, $1+m$ dimensional vectors are involved in the calculation of a dissimilarity coefficient. This results in the following weighted hybrid dissimilarity coefficient

$$d_{i,k,j} = \frac{1}{1+p} \sum_{j=1}^{p} \sqrt{|x'_{i,j} - q_j^k|^2} + \frac{p}{1+p} \sum_{j=p+1}^{m} \beta_j \zeta_{i,k',j} \tag{17}$$

4.7 Sparse K-prototypes

We implement sparse K-prototypes by employing K-prototypes in the algorithm denoted in Section 4.1. Step 2(a) in Equation 4 is minimized using standard K-prototypes algorithm. We have defined three ways to calculate the dissimilarity measures for the K-prototypes method in the previous sections. The first is shown in Equation (7) which combines the Euclidean distance and a simple matching scheme. The second hybrid dissimilarity measure in Equation (11) combines a Minkowski-distance-based dissimilarity with a weighted Hamming distance. The third dissimilarity measure in Equation (17) combines Max-Min standardized numerical variables with an entropy weighted Hamming distance.

Moreover, Step 2(b) becomes a two-step approach. We estimate the weights separately for numerical and categorical type data. Firstly, this is done to ensure fair allocation of the weights between categorical and numerical variables. Secondly, this separation may be necessary due to the inherent differences in the distribution of the dissimilarity measures, even if the measures are defined on the same scale. Categorical variables typically assume fixed values, either 1 or between 0 and 1, while numerical variables freely range within their respective feature ranges. These differences typically result in different ranges for the between cluster distance for features. Remember that the weights are determined using a soft-thresholder which assigns weighs based on the between-cluster distance for each feature. By estimating the weights separately for numerical and categorical variables, we prevent the bias of assigning higher or lower weights

to a particular type of feature solely due to the inherent differences in between-cluster feature distance. However, it is possible to adjust the weights accordingly if there is prior knowledge indicating that one type of feature might play a larger role in the clustering. Moreover, further analyzing the between-cluster distance for each feature can provide insight into whether a feature is important for the clustering.

4.8 Sparse k-modes

It may also be the case that data only contains categorical data. In that case, Huang (1997b) proposes an algorithm called K-modes which is an extension of the K-means method to handle categorical values. The K-modes handles categorical values similar to the K-prototypes method in (7). In K-modes, the dissimilarity $d_{i,i'}$ based on simple matching for two observations along feature j looks as follows

$$d_{i,i'} = \sum_{j=1}^{p} \delta(x_{i,j}, x_{i',j}), \text{ with } \delta(x_{i,j}, x_{i',j}) = \begin{cases} 0 & \text{if } x_{i,j} = x_{i',j} \\ 1 & \text{if } x_{i,j} \neq x_{i',j} \end{cases} \tag{18}$$

In general, one could define different different dissimilarity measures. Aside from the simple matching scheme, we use the discussed categorical dissimilarity measures for the K-prototypes algorithm. Namely, the weighted Hamming dissimilarity measure in Equation (8) and the weighted Hamming dissimilarity measure with entropy weights in Equation (14).

In similar vein to the sparse K-means algorithm, we can achieve sparse clustering using the following modifications. In the algorithm denoted in Section 4.1, Step 2(a) is minimized using standard K-modes algorithm. Moreover, like the sparse K-means and sparse K-prototypes methods, K-modes produces locally optimal solutions that are dependent on the initial cluster centers.

4.9 Cluster initialization

In general, clustering methods such as K-means, K-prototypes and K-modes are efficient but sensitive to initial cluster center conditions and outliers. The iterative nature of cluster methods results in locally optimal solutions that are not necessarily global optima. Nevertheless, there are different ways to mitigate the effects of random initialization in clustering methods. One approach is to repeat the algorithm with random cluster initializations and choose the one that results in small intra-cluster distance and large inter-cluster distance. Another option is to select the initial clusters based on the density of each object. In our algorithms, we incorporate both repeated initialization and a centroid initialization scheme proposed by Ji et al. (2015). This scheme takes both the centrality and distance of each data object into account to determine the initial centroids.

First, let $NborS(x_i)$ denote the neighbor-set of x_i which is given by:

$$NborS(x_i) = \{x_{i'} | d_{i,i'} \leq \tau, \text{for } x_{i'} \in \mathbf{X}\} \tag{19}$$

where $\tau > 0$ is the neighbor threshold which is set in advance; the higher the value of τ is, the more data objects the neighbor-set $NborS(x_i)$ includes. As an heuristic, τ is set to the 33% percentile of all dissimilarities measures. To make contributions of numerical and categorical the same, a simple matching scheme is applied to categorical variables and the dissimilarity for numerical variables is as follows:

$$d^n_{i,i',j} = \sum_{j=1}^{p} \left(\frac{x_{i,j} - x_{i',j}}{max(j) - min(j)} \right)^2 \tag{20}$$

Then, define the centrality of an data object x_i as follows

$$Cen(x_i) = \frac{|NborS(x_i)|}{|NborSMax|} \tag{21}$$

where $|\cdot|$ is the cardinality of a set and, $NborSMax = \max_{x_i \in \mathbf{X}}(NborS(x_i)$ is the neighor-set with the most elements. Based on this concept of centrality, the probability of an object x_i to be the first cluster is

$$P_1(x_i) = Cen(x_i) \tag{22}$$

However, if centrality is only considered to determine the subsequent centers, then data objects in the same clusters may be chosen. Furthermore, if we only consider distance, then outliers can be picked. Hence, a combination of the two is used. Let $Q_l = \{q_1, \ldots, q_l\}$ be the set of acquired cluster centers. Then, the probability of data objects in $\mathbf{X}$ to be $l+1$ cluster center is

$$P_{l+1}(x_i) = \min_{k \in Q_l} d_{i,k} \times Cen(x_i) \tag{23}$$

This probability takes both the centrality and distance to the nearest cluster into account. Intuitively, outliers are less likely to become the next cluster due to their low centrality, whereas data objects already belonging to the same cluster have a reduced probability due to their proximity to the existing cluster center. The initialization schemeis as follows

1. Calculate the probability to be the first cluster center as in (21) for each x_i and set the object with the highest value as the initial cluster
2. If the desired amount of clusters is reached, go to step 3. Otherwise, for each data object x_i calculate the probability to be the next cluster center as in (23).
3. Output initial clusters is given by Q_l

Naturally, this initialization scheme can be applied to the K-modes algorithm as well. In this case, the dissimilarity measure is the simple matching scheme as only categorical variables are take into account.

4.10 Evaluation of the methods

In order to evaluate the performance of the clustering methods, we use the Classification Error Rate (CER). Consider two partitions each of length n, denoted by P and Q. Partition P represents the true class labels, while partition Q represents a partition obtained through clustering. Let $1_{P(i,i')}$ be an indicator whether partition P places observations i and i' in the same group,

and the same applies for $1_{Q(i,i')}$. Then, the CER is defined as

$$\text{CER} = \sum_{i>i'} \frac{|1_{P(i,i')} - 1_{Q(i,i')}|}{\binom{n}{2}} \tag{24}$$

If partitions P and Q agree perfectly, the CER is equal to zero.

5 Results

Section 5 presents the findings in this thesis. We first replicate the application of the sparse K-means method by Witten and Tibshirani (2010) on SNP data in Section 5.1. Then, we evaluate the effectiveness of the proposed sparse K-modes and sparse K-prototypes method on simulated data in sections 5.2 and 5.3 respectively. Lastly, we apply our proposed methods in Section 5.4 on gene expression data to classify patients with AML and ALL.

5.1 SNP Data

Firstly, we try to replicate an application of the sparse K-means method. Witten and Tibshirani (2010) use sparse K-means to try to identify populations in SNP data. In line with replicating the study results of Witten and Tibshirani (2010), we aim to use a comparable selection of candidate tuning parameters values for s. However, the specific values for s in their study are not mentioned. Hence, we define our grid as follows: We take an equally sized grid of 20 values between 1 and $\sqrt{p}$, where in this case $p = 17026$. An overview of the results is shown in figure 1 which displays the gap statistic (a) and CER (b) values as a function of the number of nonzero weights, and the values of the weights for the tuning parameter that resulted in the minimal CER in the center panel and smallest amount of nonzero features.

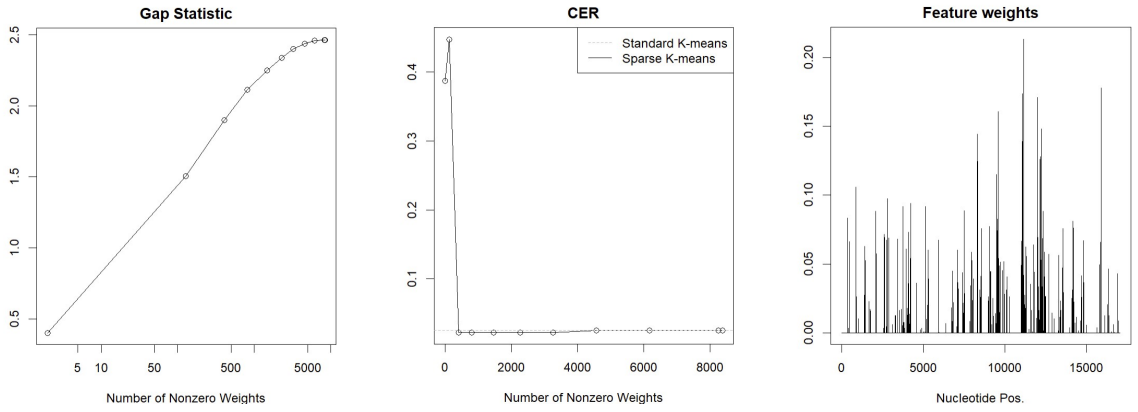

Figure 1: *Left*: The gap statistics obtained as a function of the number of SNPs with nonzero weights. *Center*: The CERs obtained utilizing standard 3-means and sparse 3-means clustering, for a range of values of the tuning parameter. *Right*: Weights for all 17026 features.

Similarly to Witten and Tibshirani (2010), the results show that sparse 3-means improves upon the standard 3-means method in terms of interpretability. Sparse 3-means can select a subset of features which can differentiate between the three populations. Furthermore, the CER

of sparse 3-means is 0.022, which is lower compared to the CER value of 0.025 for the standard 3-means method. The small difference in CER values may be explained as follows. It is possible that within the SNP data, there is a small number of features (SNPs) that exhibit distinct variations specific to each population. In contrast, there is a large subset of features (SNPs) that exhibit minimal variation among the three populations. By reducing the influence of these shared SNPs and focusing on the unique features, the sparse method could have improved the clustering performance. Additionally, it can be noted that the gap statistic does not perform well with this data. The gap statistic increases as the number of nonzero weights increase, even though this does not lead to the lowest CER value.

However, there are some discrepancies with the results of Witten and Tibshirani (2010). While the CER values and gap statistic measures are similar, the number of nonzero weights in our replication study seem to converge to 8369 nonzero features weights (no matter how high the tuning parameter s). Whereas the original study obtained the maximum of 17026 nonzero weights as their tuning parameter increases. This is likely to the lack of precise knowledge regarding the tuning parameters for both the imputation and the sparse 3-means method. Nonetheless, the conclusions drawn from these results remain similar to the original paper.

5.2 Simulation study K-modes and sparse K-modes

A simulation study is conducted to evaluate the standard K-modes and sparse K-modes methods under various dissimilarity specifications. We compare three dissimilarity measures; the Simple Matching (SM) scheme in (7), the weighted hamming dissimilarity (WHD) in (8) and the entropy weighted hamming dissimilarity (EWHD) in (14). In this simulated data, 50 categorical features differ between the three groups. The categorical features are sampled from $X_{ij} = [\text{“A”}, \text{“B”}, \text{“C”}]$, where $X_{ij} \sim P(X_{ij})$ for all $j \leq 50$. Each group has length $n = 20$. The probabilities for the first simulation are shown in Table 1. For $j > 50$, each value has an equal chance of $\frac{1}{3}$ within each group.

Table 1: Probability distribution for three groups in Simulation 1 and 2

Group	**Outcome**		
	$X_{ij} = x_1$	$X_{ij} = x_2$	$X_{ij} = x_3$
Group 1	0.6	0.25	0.15
Group 2	0.15	0.6	0.25
Group 3	0.25	0.15	0.6

Table 2: Probability distribution for three groups in Simulation 3

Group	**Outcome**		
	$X_{ij} = x_1$	$X_{ij} = x_2$	$X_{ij} = x_3$
Group 1	0.8	0.1	0.1
Group 2	0.1	0.8	0.1
Group 3	0.1	0.1	0.8

The result of Simulation 1 are displayed in table 3, showcasing the classification error rates (CER) for the standard K-Modes and sparse K-Modes variations. Firstly, standard 3-Modes tends to perform better than sparse 3-Modes when $p = 50$. However, as the value of p increases, the best performing model tends to be a sparse 3-Modes model. This reflects the fact that 3-Modes utilizes all variables, whereas the Sparse 3-Modes tries to find a subset of features. Consequently, 3-Modes is at an advantage when $p = 50$. However, as noise is introduced, it

becomes apparent that sparse 3-modes gains the upperhand by selecting a subset of important features.

Secondly, we note that the SM outperforms the other dissimilarity measures in both the standard 3-Modes and the Sparse 3-Modes. Additionally, the performance of the (MD,WHD) and (MM,EWHD) measures is poor, for both 3-Modes and Sparse 3-modes when $p = 500$. An analysis into the clusters obtained by each method reveals that the WHD and EWHD measures sometimes cluster all observations into a single cluster ($p = 200$ and $p = 500$). This phenomenon occurs because these dissimilarity measures assign high values to a feature even if it is important. These measures take both inter-cluster and intra-cluster similarity into account, resulting in low dissimilarity values only when both intra-cluster similarity and inter-cluster dissimilarity are maximized. In the simulated data, this is often not the case since $p = 0.6$, which means that the homogeneity within the clusters is low. Consequently, these measures perform poorly as they fail to provide low enough dissimilarity values to effectively separate the groups. In contrast, the SM scheme demonstrates better capability of separating groups ($p = 200$) as it simply assigns either the maximum value one or zero. Therefore, the algorithm identifies instances where intra-cluster dissimilarity is minimized for certain features, allowing for effective separation of groups by providing lower dissimilarity values. Thus, SM exhibits better distinguishability between important and non-important features, as it can produce lower dissimilarity values.

Table 3: 3-Modes and sparse 3-Modes results for simulation 1. The values denote the mean (and standard error) of the CER over 20 simulations

	p = 50	p = 200	p = 500
3-Modes(SM)	**0.000 (0.000)**	0.194 (0.021)	0.423 (0.003)
3-Modes(WHD)	0.005 (0.003)	0.455 (0.005)	0.514 (0.034)
3-Modes(EWHD)	0.008 (0.003)	0.433 (0.005)	0.528 (0.034)
Sparse 3-Modes(SM)	0.007 (0.003)	**0.032 (0.007)**	**0.161 (0.044)**
Sparse 3-Modes(WHD)	0.047 (0.020)	0.190 (0.033)	0.529 (0.054)
Sparse 3-Modes(EWHD)	0.028 (0.013)	0.159 (0.034)	0.562 (0.057)

5.3 Simulation study K-prototype and sparse K-prototypes

A simulation is run on the standard K-prototypes and sparse K-prototypes methods under different dissimilarity measures. We compare three dissimilarity measures. The first is the standard K-prototype measure in Equation (7) which is a combination of the Euclidian distance (ED) and Simple matching (SM) scheme. The second measure is the hybrid dissimilarity proposed by Sangam and Om (2018) in Equation (11). This hybrid dissimilarity combines the Minkowski distance (MD) with a weighted Hamming dissimilarity (WHD). The third dissimilarity measure is introduced Jia and Song (2020) in Equation (17). The categorical dissimilarity measure extends the WHD by incorporating weights based on entropy for each feature (EWHD). In addition, numerical attributes are handled using Max-Min standardization (MM).

The data is generated according to Section 3. In the simulated dataset, there are 50 features, consisting of 25 numerical and 25 categorical variables, that differ between the three groups.

The numerical features are generated with the following distribution $X_{ij} \sim \mathrm{N}(\mu_{ij}, 1)$, where $\mu_{ij} = \mu\,(1_{i\in C_1, j\leq q} - 1_{i\in C_2, j\leq q})$. The value of μ is set to $\mu = 1$. The categorical features are sampled from $X_{ij} =$ ["A", "B", "C"], where $X_{ij} \sim P(X_{ij})\ \forall j \leq 50$. The probability distribution is shown in table 1. For $j > 50$, each categorical value has an equal chance of $\frac{1}{3}$ within each group.

The results of the Simulation 2 are shown in table 4. Firstly, similar to the K-Modes simulation, we find that the standard 3-Prototypes models tend to perform better than the Sparse 3-Prototype models when $p = 50$. However, as p increases, the Sparse 3-Prototypes models demonstrate better performance due to the feature selection capabilities. Secondly, the (ED,SM) measures tend to perform better than the other dissimilarity schemes. This worse performance of the (MD, WHD) and (MM,EWHD) measures is mostly attributed to the influence of the categorical dissimilarity measures, specifically the WHD and EWHD measures, rather than the numerical dissimilarity measures themselves. Similar to Simulation 1, these measures fail to provide low enough dissimilarity values to effectively cluster the groups, primarily due to the lack of homogeneity within features. Furthermore, when $p = 500$, the values for (MD+WHD) and (MM+EWHD) are left empty as they set all the categorical weights to zero. In line with the objective of the simulation, which aimed to assess the effectiveness of combining numerical and categorical variables, we did not to apply the sparse 3-means method to the remaining numerical variables. This observation further reinforces our assertion that these measures are less effective in cases where the homogeneity within features is not as distinct.

Table 4: 3-Prototypes and sparse 3-Prototypes results for simulation 3. The values denote the mean (and standard error) of the CER over 20 simulations

	p = 50	p = 200	p = 500
3-Prototypes(ED,SM)	**0.002 (0.001)**	0.280(0.027)	0.519 (0.015)
3-Prototypes(MD,WHD)	0.010 (0.056)	0.466 (0.005)	0.585 (0.016)
3-Prototypes(MM,EWHD)	0.028 (0.046)	0.528 (0.018)	0.628 (0.010)
sparse 3-Prototypes(ED,SM)	0.056 (0.006)	**0.088 (0.018)**	**0.183** (0.038)
sparse 3-Prototypes(MD,WHD)	0.023 (0.011)	0.184 (0.044)	-
sparse 3-Prototypes(MM,EWHD)	0.083 (0.012)	0.183 (0.056)	-

However, it is worth noting that the WHD and EWHD measures may have the potential to perform well if the important features are more pronounced within each cluster. It is likely that a value of $p = 0.6$ is too low for these measures to operate effectively. To investigate this, we run a third simulation. The setup for this simulation is similar to that of Simulation 2. However, in this case, we use a different probability distribution for the categorical values, which is outlined in Table 2. Additionally, we perform the analysis for values of p equal to 200, 500, and 1000.

The result of Simulation 3 are shown in Table 5. We note that with $p = 200$, there is already a distinct separation among the clusters, as most CER values are close to zero. However, it becomes evident that when $p = 500$, the clustering performance of the standard 3-Prototypes method is affected by the additional noise. In contrast, the sparse 3-Prototypes method demonstrates an advantage in this scenario, as it has the ability to selectively choose features. In this

case, sparse 3-Prototypes has the upper hand as it can select features.

Moreover, when $p = 1000$, the combination of (MM,EWHD) tend to exhibit better performance compared to (ED,SM) and (MD,WHD). One potential explanation for this observation is that the impact of noise on categorical variables is further mitigated by the entropy weights employed in EWHD. These weights assign higher importance to features that exhibit clearer homogeneity. Therefore, by considering not only inter and intra-cluster similarity, as in WHD, but also incorporating homogeneity through entropy, the performance of EWHD was further improved.

Table 5: 3-Prototypes and sparse 3-Prototypes results for simulation 3. The values denote the mean (and standard error) of the CER over 20 simulations

	p = 200	p = 500	p = 1000
3-Prototypes(ED,SM)	0.030 (0.014)	0.422 (0.023)	0.489 (0.017)
3-Prototypes(MD,WHD)	0.000 (0.000)	0.186 (0.028)	0.393 (0.030)
3-Prototypes(MM,EWHD)	0.001 (0.001)	0.076 (0.011)	0.393 (0.034)
sparse 3-Prototypes(ED,SM)	0.000 (0.000)	0.038 (0.021)	0.222 (0.061)
sparse 3-Prototypes(MD,WHD)	0.000 (0.000)	0.001 (0.001)	0.149 (0.054)
sparse 3-Prototypes(MM,EWHD)	0.000 (0.000)	0.017 (0.006)	0.056 (0.030)

We can summarize the main findings of the simulations with the following points. First, the sparse K-modes and sparse K-prototypes demonstrate better robustness to noise in the data relative to the standard K-modes and standard K-prototypes. Secondly, we mainly investigated the categorical dissimilarity measures, as the numerical measures seemed to perform well across all methods. The main challenge resided in defining suitable categorical dissimilarity measures. Among the investigated measures, the SM scheme emerged as the most suitable choice when dealing with less homogeneous features. However, in cases where the important features are more prominent as in Simulation 3, the EWHD measure, particularly when the data contained a lot of noise, and the WHD measure demonstrated better robustness compared to the SM measure. Hence, different measures may exhibit better performance depending on the characteristics of the dataset. Additionaly, we note that these findings are based on simulated data, and thus may not accurately reflect the true performance or capabilities in real-world scenarios.

Lastly, we will briefly cover the performance of the cluster initialization scheme in Section 4.9. In general, we observed that the cluster initialization approach selects data points as centers, with each point belonging to a separate cluster. However, as more noise is introduced, this scheme occasionally selects two points from the same cluster. Notably, the categorical measure SM demonstrates greater resilience to the initial cluster selection and has the ability to navigate towards the actual clusters if the initial cluster contain observations from the same clusters. On the other hand, the performance of the categorical measures WHD and EWHD proved to be more dependent on the initial clusters, resulting in poorer performance when the initial clusters were not the most optimal ones.

5.4 Gene Expression Data

In this section, we apply sparse K-modes and sparse K-prototypes on gene expression data to identify patients with AML and ALL. We first cluster the training dataset which has dimensions 38×1648. It is important to note that we exclusively utilize the SD scheme for the K-modes algorithm since both the WHD and EWHD measures resulted in clustering all observations into a single cluster. This suggests that these particular measures are not suitable for this type of data. Consequently, we also only use the (ED,SM) dissimilirity combination for the K-prototypes method. The corresponding CERs and feature weights are presented in Figure 2.

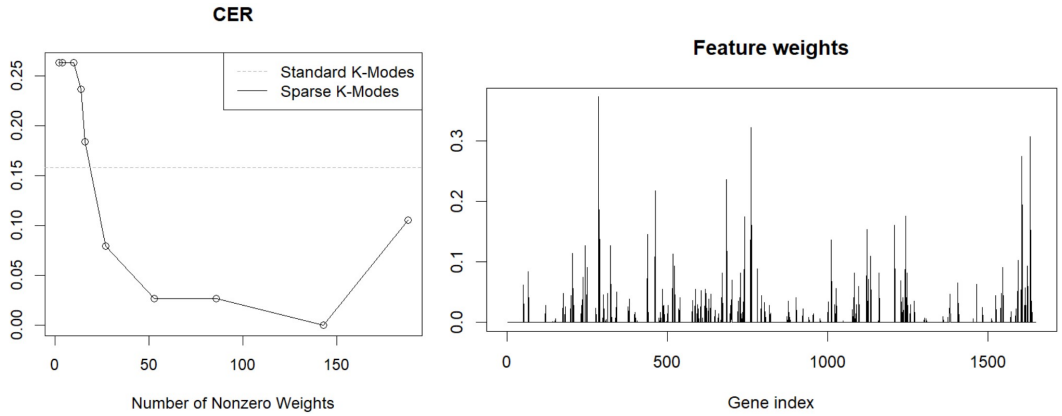

Figure 2: *Left*: The CERs obtained utilizing standard 2-means and sparse 2-means clustering, for a range of values of the tuning parameter *Right*: Weights for all 1648 features.

Firstly, we note that the sparse 2-modes achieves a CER value of zero when the amount of nonzero weights equals 148. The standard 2-modes algorithm has a CER value of 0.15. Hence, the sparse 2-modes results in a more accurate and meaningful cluster by considering a subset of the most important genes. Further investigation of these genes can be conducted based on the features with nonzero weights. However, we note that the performance of the sparse K-modes is dependent on the tuning parameter, as shown in the left plot of Figure 2.

Next, we apply standard 2-modes and sparse 2-modes on the independent dataset with dimensions 34×7130. The results are shown in Figure 3. Firstly, the sparse 2-modes has a CER of 0.03, misclassifying just a single data object, whereas the standard 2-modes has a CER of 0.18. Hence, sparse 2-modes improves upon the standard 2-modes in both accuracy and interpretability of clusters. Secondly, the CER value for sparse 2-modes achieves its lowest score when the amount of nonzero weights is six. This observation suggests that the genes primarily responsible for the clustering may not be concentrated within the range of the first 1648 gene features. As illustrated in the right plot of Figure 3, the most influential indexes are found outside this range.

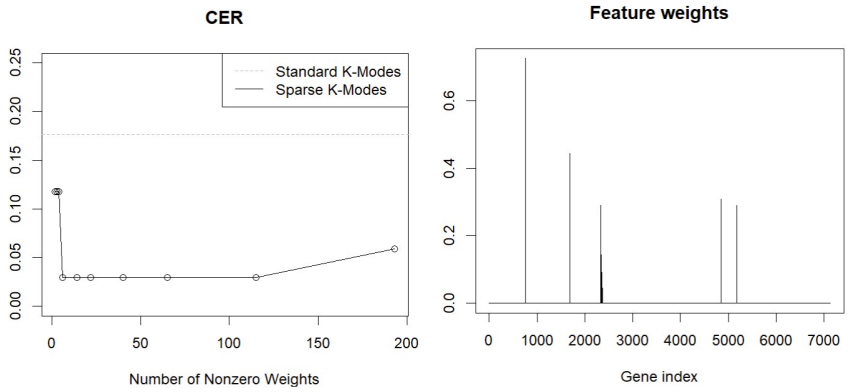

Figure 3: *Left*: The CERs obtained utilizing standard 2-means and sparse 2-means clustering, for a range of values of the tuning parameter. *Right*: Weights for all 7130 features.

We now apply standard and sparse K-prototypes on the complete initial dataset with numerical and categorical data, to determine whether the inclusion of numerical data enhances the clustering results. We show the CERs in Figure 4. The results show that the sparse K-prototypes performs worse than the standard K-prototypes. One possible explanation for this result is that numerical features may not significantly contribute to the clustering. Consequently, we verify this by using the sparse 2-prototypes in a supervised manner. In Step 2(a) of the algorithm described in Section 4.1, we input the correct cluster assignments and continue with the algorithm. By comparing the final clustering result with the correct assignments, we aimed to identify features that primarily drive cluster separation. After testing, it is the case that 11 features, comprising five categorical and six numerical variables, obtained positive weights during clustering, ultimately leading to the correct clustering. This phenomenon highlights certain key points. Firstly, it shows how dependent the algorithm is on the initial clusters. The employed initialization scheme proved to be ineffective for our dataset, which highlights the significance of the initial clusters in clustering algorithms. Secondly, our results suggest the potential necessity to explore alternative dissimilarity measures for the K-prototypes model. Other dissimilarity measures have been proposed in the literature which we did not cover in this study.

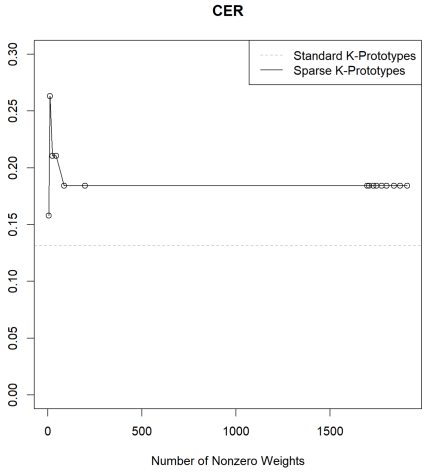

Figure 4: The CERs obtained utilizing standard 2-means and sparse 2-means clustering, for a range of values of the tuning parameter

6 Conclusion

In this thesis project we investigate the follow research question, "How can the feature selection framework in clustering proposed by Witten and Tibshirani (2010) be extended to handle datasets that contain either only categorical variables or a mix of numerical and categorical data?". We employed the framework in conjunction with the K-Modes algorithm to handle datasets consisting solely of categorical features. Additionally, we applied the framework to the K-Prototypes algorithm to address datasets containing both numerical and categorical variables. Through simulation experiments and analysis of gene expressions, we find that the integration of the framework with K-Modes and K-Prototypes resulted in a selection of truly relevant features for the clustering. As a result, we observe improvements in both interpretability and cluster accuracy when compared to using K-Modes and K-Prototypes independently. Hence, we can extend the feature selection framework proposed by Witten and Tibshirani (2010) to effectively handle data sets with only categorical data and mixed type data sets by incorporating the K-Modes and the K-Prototypes methods respectively.

Our study has several implications. Firstly, our findings demonstrate the effectiveness of the proposed sparse clustering methods for handling high-dimensional mixed data types. This suggests that these methods can be valuable in various domains where data contains a combination of numerical and categorical variables. Secondly, our study highlights the potential application of unsupervised sparse clustering in identifying patients in gene expression data, specifically in classifying patients with AML and ALL. Additionally, we showed that our methods can also be used in a supervised manner, although clustering methods are usually employed in an unsupervised setting.

Nonetheless, this paper raises several discussion points. Firstly, the application of the sparse K-mode and, in particular, the K-prototypes methods on gene expression data revealed that the performance of our methods is highly dependent on the tuning parameters and initial cluster selection. Further research could thus focus on an initialization scheme that is more robust to noise. Moreover, one could focus on tuning of the optimal L_1 bound in the sparse clustering framework. Choosing the optimal tuning parameter in cluster analysis is generally challenging since it is an unsupervised learning technique. Additionally, different dissimilarity measures can be explored for the proposed sparse K-prototype and sparse K-modes models.

References

Chang, W.-C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 32(3):267–275.

Chavent, M., Cottrell, M., Lacaille, J., Mourer, A., and Olteanu, M. (2022). Sparse weighted k-means for groups of mixed-type variables. pages 1–10.

Friedman, J. and Meulman, J. (2004). Clustering objects on subsets of attributes. 94305:1–30.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer:

class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537.

Huang, Z. (1997a). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pages 21–34. Citeseer.

Huang, Z. (1997b). A fast clustering algorithm to cluster very large categorical data sets in data mining. *Dmkd*, 3(8):34–39.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304.

Ji, J., Pang, W., Zheng, Y., Wang, Z., Ma, Z., and Zhang, L. (2015). A novel cluster center initialization method for the k-prototypes algorithms using centrality and distance. *Applied Mathematics & Information Sciences*, 9(6):2933.

Jia, Z. and Song, L. (2020). Weighted k-prototypes clustering algorithm based on the hybrid dissimilarity coefficient. *Mathematical Problems in Engineering*, 2020:1–13.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Learning the parts of objects by non-negative matrix factorization.*, 401(6755):788–791.

Liu, W.-m., Mei, R., Di, X., Ryder, T. B., Hubbell, E., Dee, S., Webster, T. A., Harrington, C., Ho, M.-h., Baid, J., et al. (2002). Analysis of high density expression microarrays with signed-rank call algorithms. *Bioinformatics*, 18(12):1593–1599.

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8(41):1145–1164.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Sangam, R. S. and Om, H. (2018). An equi-biased k-prototypes algorithm for clustering mixed-type data. *Sādhanā*, 43:1–12.

Śmieja, M., Hajto, K., and Tabor, J. (2019). Efficient mixture model for clustering of sparse high dimensional binary data. *Data Mining and Knowledge Discovery*, 33:1583–1624.

Szepannek, G. (2018). clustMixType: User-Friendly Clustering of Mixed-Type Data in R. *The R Journal*, 10(2):200–208.

Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays . *Bioinformatics*, 17(6):520–525.

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

A Programming code