Experimental Evaluation of Feature Selection Methods for Clustering

Martin Azizyan[1], Aarti Singh[1], and Wei Wu[2]

[1]Machine Learning Department, Carnegie Mellon University
[2]Lane Center for Computational Biology, Carnegie Mellon University

Abstract

Background. Recently several clustering methods have been proposed which perform feature selection with the goal of finding structure in high dimensional data that is confined to a small set of features. Most of these methods are stated in terms of non-convex optimization objectives, and are computationally intractable to solve exactly in general. In this project we consider two such methods, both of which are stated as modifications of the famous K–means objective. Approximation algorithms proposed for these methods require initialization with some clustering, which is taken to be the result of standard K–means.

Aim. Our goal is to experimentally characterize the behavior of these algorithms, specifically the type of structure that they can and cannot discover, how they differ in terms of performance, and what effect the approximation methods have on the clustering and feature selection results.

Data. We use a dataset containing a detailed phenotypic characterization of 378 participants in a lung disease study in terms of 112 demographic, environmental, and medical features. The variable types are mixed, including continuous, ordinal, and nominal features.

Methods. Synthetic datasets are designed to elucidate the performance and behavior of the methods in consideration, both in terms of the optimal solutions for the optimization problems and the quality of the solutions obtained from the approximation algorithms. We formulate greedy K–means, a new feature selection method for clustering, inspired by the two methods in consideration. We propose a few initialization methods as alternatives to K–means initialization. We also propose a clustering criterion for mixed variable types. This criterion can directly be adapted to perform feature selection with the same approach used in greedy K–means.

Results. Based on synthetic experiments our proposed greedy K–means method seems to never perform much worse than the other two, and it may be advantageous for interpretability because it gives exactly sparse solutions.

We show examples where the previously proposed initialization method of using the K–means solution performs arbitrarily poorly, and that alternate initialization is needed. Among the initialization methods we consider, the one that seems to perform best is random support based initialization, and greedy K–means lends itself to that method naturally.

We show that the result of applying these methods to the lung disease data is unstable in terms of the resulting clustering, and that the selected features are highly influenced by correlations between variables, which warrants further investigation to ensure there is any significant *non*-linear structure in the data found by the clustering methods.

Conclusions. We provide insights regarding the challenges presented by noisy, high-dimensional data when applying the feature-sparse clustering methods in consideration. We demonstrate the practical importance of not ignoring the approximate nature of the optimization procedures available for these methods. Since linear structure (i.e. correlations) are overwhelmingly likely to exist in real-world data sets, our results indicate that care needs to be taken regarding the type of structure discovered when interpreting the meaning of selected features.

Contents

1 Introduction

Cluster analysis is one of the fundamental tools for exploratory data analysis, and clustering methods such as K–means have been widely used to extract useful structure from data. However, the effectiveness of these methods is diminished in modern high-dimensional datasets, where the large number of measured features

of the data can overwhelm any existing signal with noise. In other areas of machine learning and statistics, sparse methods, and feature selection in particular, have been used successfully to help overcome this "curse of dimensionality". The problem of clustering with feature selection (Section 2) has seen significant recent interest [24, 28, 11, 23, 20, 17, 2, 15].

Feature-sparse clustering algorithms may be beneficial in high dimensional problems for several reasons. Sparsity may facilitate the discovery of cluster structure that is confined to only a small subset of the features, and difficult to detect when using all features due to noise. Also, a sparse clustering can be expected to improve interpretability, which is an important factor in exploratory analysis. Another possible benefit may exist in settings where measuring each feature for a new sample incurs some additional cost. In that case, a clustering that relies on a smaller set of dimensions would reduce the cost of classifying future observations.

In this project we consider two methods from the literature [24, 28], which we refer to as Regularized K–means (described in Section 6.2), and Sparcl (Section 6.3). Both methods add (locally) convex penalties to the K–means objective to perform feature selection. In general obtaining an optimal solution to the resulting optimization problems is computationally intractable, so approximation algorithms are used which perform iterative procedures that require an initial clustering as input.

Our goal is to experimentally characterize the behavior of these algorithms (Section 5). We are specifically interested in the following:

1. What type of structure can each method recover, and what type of structure do they fail to recover?
2. How do the methods differ?
3. What approximations can we hope to recover, and how do they relate to the true optima of the objectives?

We use synthetic datasets (Section 6), as well as a dataset containing detailed phenotype information of participants in a lung disease study (Section 4).

Our contributions are:

- The formulation of a new method for feature-sparse clustering, greedy K–means (Section 6.4), which uses discrete, non-convex penalties to induce sparsity, and yet appears no more difficult to approximate than the other two methods in consideration.
- A clustering criterion for mixed variable types (Section 6.9) that can be easily adapted to perform feature selection in a manner analogous to greedy K–means.
- Discovery of possibly undesirable behavior of Regularized K–means due to the type of convex sparsifying penalty used (Section 7.3.1).
- Exploration of the important role played by the initialization method chosen for the approximation algorithms (Sections 7.3.3-7.3.5), and description of some initialization methods that may be of practical interest (Section 6.5).
- Identification of issues regarding the interpretability of the clustering results obtained from the lung disease data with feature selection (Section 7.4), namely: (a) a strong effect of inter-dimension correlations on the selection of features, which indicates that linear structure (more usefully estimated with methods such as PCA and sparse PCA, than clustering) may play much too strong a role in the types of objectives being considered, and (b) instability in the partitioning of the subjects within the best several solutions in terms of the objective given by multiple random runs of the approximation algorithm.

For a more detailed exposition and discussion of these and other findings, see Section 8.

2 Feature-sparse clustering

Given a data matrix $X \in \mathbb{R}^{n \times d}$, the goal of clustering is to find a partition of each n rows of the matrix (a.k.a. samples, observations, points) into some (prespecified or data dependent) number K of clusters that optimizes

a certain objective. In the case of K–means, one of several equivalent statements of the objective is minimizing the total sum of squared distances from each point to the mean of its assigned cluster. Adding feature selection to this would mean that in the end, we get a subset of the features and a clustering which is defined only in terms of those features. With feature selection, the goal is to simultaneously find a subset of features, and a clustering defined only in terms of those features. In particular, the resulting feature-sparse clustering needn't have any structure on the features that were not selected.

Here the notion of feature selection is distinct from pre-screening features, and from selecting features as a post-processing step to clustering. Pre-screening can be useful to remove features that are highly noisy, contain missing entries, or are not useful for other reasons. After a clustering is obtained, informative post-processing steps may include the selection of features that are most mutually informative with the *fixed* cluster labels. Such techniques can be useful in some contexts, but in this project we are concerned with the problem of feature selection simultaneously with clustering.

3 Background and related work

The notion of variable selection for clustering has seen some recent interest in the literature. Existing work spans a range of approaches.

Two of the methods we consider in this project due to [24, 28] are stated in terms of the K–means objective (see e.g. [14]), with modifications that result in feature-sparse results. A related K–means based method is [11], where instead of overall variable selection, the goal is sparsity in the differences between the means of all individual *pairs* of clusters.

There are also a number of model-based methods [23, 20, 17, 15, 31], typically formulated as penalized Gaussian mixture models. For instance, [31] considers a set of penalized maximum likelihood objectives, where the penalties are designed to induce various types of sparsity in the component means and covariances of the Gaussian mixture model.

Yet another algorithm is given by [2], which iteratively selects features which have small within-cluster variance (similar to K–means), and small correlation with all already selected features. The justification for this approach is that adding a variable that is highly correlated to one that has already been selected does not add more information, even if the variable in consideration by itself has small within-cluster variance. Several different methods for trading off between the within-cluster variance and correlation are given.

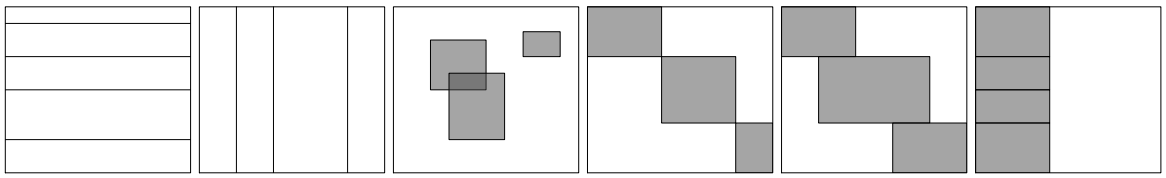


Figure 1: Some types of biclustering problems: (1a) clustering observations, (1b) clustering features, (1c) arbitrary overlapping biclusters, (1d) exclusive row and column biclusters, (1e) exclusive row biclusters, (1f) feature-sparse clustering.

Clustering with feature selection can be seen as an instance of the notion of biclustering [16, 25, 5]. In general, biclustering is the problem of finding groups of submatrices in the data, i.e. sets of rows and columns, such that the entries in each submatrix are similar in some sense. These submatrices are referred to as *biclusters*. Different constraints on the shapes of the biclusters result in different types of biclustering problems. See Figure 1 for schematic descriptions of some of the types of biclustering problems that have been considered.

4 Lung disease phenotype data

The dataset is composed of $d = 112$ features from $n = 378$ participants in a lung disease study [19]. The features include demographic information (e.g. age, race, ethnicity, gender, number of children and siblings), environmental information (e.g. pets, smoking status), medical history (including family members), severity of symptoms, use of medication, and the results of a range of medical tests. The variable types in the data are mixed, with 60 binary, 14 discrete (ordinal) non-binary, 36 non-discrete, and 2 categorical (with 4 and 5 values, resp.) features.

Previous analysis by [19] on similar data classified five phenotype groups of the disease with a hierarchical clustering technique based on 34 of the variables. With post-processing techniques (see Section 2), they showed that the cluster assignment could be recovered with 80% accuracy based on just 3 features.

More recent analysis by [30] using all 112 features identified six clinically meaningful subject clusters. Post-processing revealed a subset of 51 variables which could be used to reconstruct the subject classification with 88% accuracy.

5 Approach

The clustering methods we analyze (outlined in Section 6) are stated in terms of non-convex optimization objectives based on criteria that simultaneously encapsulate the goals of clustering with feature selection. We consider several random algorithms for obtaining approximate solutions to these optimization problems, since in general finding the exact optima is computationally intractable. Hence, there are two separate issues to keep in mind. One is the nature of the optimization objectives themselves in terms of the true optima (which we can't find in general), and the other is the quality of the approximate results we can compute in practice.

The factors we consider when analyzing the performance of each method are:

1. Feature selection error — are the "true" relevant dimensions identified?
2. Clustering error — how close is the approximate clustering to the clustering given by the relevant dimensions?
3. How close is the objective value of the approximate solution to the true optimum? It is of particular interest whether finding a solution that is close to optimal in terms of the objective implies low feature selection or clustering error.

These criteria can only be evaluated directly on synthetic data where ground truth can be established, and the notion of *true* relevant dimensions is well defined by design. We describe several synthetic datasets in Section 6, each constructed to elucidate different aspects of the methods in consideration.

Using the lung disease data it is impractical to find the true optima, and there is no clear notion of true relevant features or clustering. Consequently the above metrics cannot be directly applied. Our approach is to compare several of the obtained approximate solutions with the highest objective to *one other*, and explore the stability of the corresponding selected features and clustering results. The intuition is that if two solutions can look very dissimilar despite both having nearly the best found objective, then there may be reason to doubt the fidelity of the method in producing consistently meaningful results. Another element of our analysis is the characterization of the approximate solutions in terms of easily measurable properties of the data, in order to illuminate the true geometric meaning of the algorithm outputs.

6 Methods

We describe three K–means based methods for feature-sparse clustering, two from existing literature (Sections 6.2 and 6.3), and a third original method inspired by those (Section 6.4), as well as an alternative to K–means type criteria for data with mixed variable types (Section 6.9). The algorithms we consider require

an initialization; we describe several initialization methods in Section 6.5. To analyze the behavior of these methods, we use several synthetic datasets defined in Sections 6.6, 6.7, and 6.8.

All computation was performed in R [22].

6.1 Notation

For a matrix A, we refer to its i'th row as A_i, its j'th column as $A_{(j)}$, and the value in the i'th row and j'th column as $A_{i(j)}$. For an integer $m \geq 1$ we define $[m] = \{1, \ldots, m\}$, and for integers $m \leq n$, $[m..n] = \{m, m+1, \ldots, n\}$. For simplicity, throughout this document we assume that any data set has been preprocessed to have zero mean.

6.2 Regularized K–means

The Regularized K–means [24] objective is stated as

$$\min_{C \in \mathbb{R}^{K \times d}, L \in [K]^n} \frac{1}{n} \sum_{k=1}^{K} \sum_{i:L_i=k} \|X_i - C_k\|^2 + \sum_{j=1}^{d} \lambda_j \|C_{(j)}\| \tag{1}$$

where we could take $\lambda_1 = ... = \lambda_d = \lambda \geq 0$, or penalize dimensions adaptively according to their importance by setting $\lambda_j = \lambda / \|\widetilde{C}_{(j)}\|$, where $\widetilde{C} \in \mathbb{R}^{K \times d}$ is the matrix of cluster means given by L, i.e. the unpenalized cluster centers. In either case, λ controls the degree of sparsity of the resulting cluster centers. Higher λ corresponds to increasingly sparse solutions, and setting $\lambda = 0$ recovers the unregularized K–means objective.

The algorithm proposed in [24] for solving the Regularized K–means objective is an iterative coordinate descent procedure similar to Lloyd's algorithm. Given an initial clustering L and penalization parameter $\lambda \geq 0$,

1. For $j = 1, \ldots, d$, compute

$$C_{(j)} = \arg\min_{y \in \mathbb{R}^K} \frac{1}{n} \sum_{k=1}^{K} \sum_{i:L_i=k} (X_{i(j)} - y_k)^2 + \lambda_j \|y\| \tag{2}$$

   where $\lambda_j = \lambda$ in the non-adaptive version, or $\lambda_j = \lambda / \|\widetilde{C}_{(j)}\|$ where $\widetilde{C}$ is set to the means of each cluster based on the current value of L in the adaptive version.

2. For $i = 1, \ldots, n$, compute $L_i^* = \arg\min_{k \in [K]} \|X_i - C_k\|$.

3. If $L^* \neq L$, continue from step 1 with $L = L^*$. Otherwise, stop and return L and C.

6.3 Sparcl

The Sparcl objective [28] is a weighted version of the "maximum between-cluster sum of squares" formulation of K–means:

$$\begin{aligned} \max_{w \in \mathbb{R}^d, C \in \mathbb{R}^{K \times d}, L \in [K]^n} \quad & \sum_{j=1}^{d} w_j \left(\sum_{i=1}^{n} X_{i(j)}^2 - \sum_{k=1}^{K} \sum_{i:L_i=k} (X_{i(j)} - C_{k(j)})^2 \right) \\ \text{subject to} \quad & \|w\|^2 \leq 1, \|w\|_1 \leq s, w_j \geq 0 \ \forall j \in [d] \end{aligned} \tag{3}$$

where s is the parameter that controls the degree of sparsity.

The R package **sparcl** [29] includes an iterative algorithm for solving this objective. After computing an initial clustering with standard K–means, the algorithm alternates between computing the optimal w for fixed L and C (with a closed form soft-thresholding type operation), and computing L and C for fixed w using K–means on a reweighted dataset Z with $Z_{(j)} = \sqrt{w_j} X_{(j)}$ for each j. Since one of the factors we investigate in this work is the effect of initialization, we modify this code slightly to allow initialization with an arbitrary clustering.

Remark. Unlike Regularized K–means, the Sparcl objective does not reduce to K–means for any value of the sparsity parameter s. This is because even if $s \geq \sqrt{d}$ (which guarantees that the L1 constraint on w is inactive), the constraint $\|w\|^2 \leq 1$ still means that, in general, w is not constant.

6.4 Greedy feature selection

Consider the following equivalent formulation of the Regularized K–means objective (1), with uniform penalization per dimension:

$$\begin{aligned} \max_{C \in \mathbb{R}^{K \times d},\, L \in [K]^n} \quad & \sum_{j=1}^{d} \left(\sum_{i=1}^{n} X_{i(j)}^2 - \sum_{k=1}^{K} \sum_{i:L_i=k} (X_{i(j)} - C_{k(j)})^2 \right) \\ \text{subject to} \quad & \sum_{j=1}^{d} \|C_{(j)}\| \leq t \end{aligned}$$

for some $t \geq 0$. Here we have made two modifications — first, the penalty on cluster centers is expressed in terms of a constraint, as opposed to the Lagrangian form; and second, the within-cluster sum of squares is replaced by its between-cluster counterpart.

If we assume that the L2 penalty $\|C_{(j)}\|$ is intended to serve as a convex relaxation of the indicator function $\mathbf{I}(C_{(j)} \neq \mathbf{0})$, then we recover an un-relaxed objective by replacing the constraint on the cluster centers with $\sum_{j=1}^{d} \mathbf{I}(C_{(j)} \neq \mathbf{0}) \leq p$, for some integer p (in general, this results in a different clustering, and there is no exact relationship between the parameters p and t in terms of the resulting solutions). That objective, in turn, can be rewritten as

$$\begin{aligned} \max_{w \in \mathbb{R}^d,\, C \in \mathbb{R}^{K \times d},\, L \in [K]^n} \quad & \sum_{j=1}^{d} w_j \left(\sum_{i=1}^{n} X_{i(j)}^2 - \sum_{k=1}^{K} \sum_{i:L_i=k} (X_{i(j)} - C_{k(j)})^2 \right) \\ \text{subject to} \quad & \|w\|_1 \leq p,\ w_j \in \{0,1\}\ \ \forall j \in [d] \end{aligned} \tag{4}$$

by introducing indicator variables $w_j = \mathbf{I}(C_{(j)} \neq \mathbf{0})$. Note that if the optimal solution has $w_j = 0$ for some j, then the corresponding $C_{(j)}$ is arbitrary, i.e. any $C_{(j)} \in \mathbb{R}^K$ results in the same objective, since the contribution from dimension j to the between-cluster sum of squares is nullified. We can use the convention that for any such j, $C_{(j)} = \mathbf{0}$.

Incidentally, this objective can also be obtained from Sparcl (3), with the constraints $\|w\| \leq 1$ and $w_j \geq 0$ replaced by $w_j \in \{0,1\}$. We call (4) the greedy K–means objective, for reasons that will become apparent below.

The similarity of greedy K–means to Regularized K–means and Sparcl suggests two approximate optimization procedures. One way is to perform three-way coordinate ascent by iterating over reassigning L, C, and w in turn, similar to the solution for Regularized K–means proposed by [24]. The other way is to alternate between applying standard K–means on the dataset reweighted by w, and computing w based on the K–means solution, similar to the solution used for Sparcl in [28]. In both cases, w is computed by greedily selecting the dimensions which maximize the between-cluster sum of squares, with L and C fixed.

With a slight modification, these greedy methods can accommodate the case where there is a group (or multiple groups) of features that should either be all selected, or all not selected. One such instance is when there is a categorical feature that has been expressed through dummy variables. For example, suppose dimensions 1 through r (for some $r < d$) are indicator variables for a categorical feature that takes on r values. Then the constraints in (4) can be replaced by

$$\frac{1}{r} \sum_{j=1}^{r} w_j + \sum_{j=r+1}^{d} w_j \leq p,\ w_j \in \{0,1\}\ \forall j \in [d],\ w_1 = w_2 = \ldots = w_r$$

where the factor $\frac{1}{r}$ is added so that the group of indicator features has the same combined weight as all other features. Either of the optimization procedures above can be easily modified so that in the greedy feature selection step, features 1 to r are treated as one.

6.5 Initialization methods

All the algorithms discussed above require initialization with some clustering of the data. The papers that propose those algorithms assume that the (or, more appropriately, a) K–means solution will be used to do so. In Section 7.3.5 we demonstrate that even initializing with *the* true K–means clustering can, in general, result in suboptimal solutions. Hence, we are also interested in evaluating some other initialization methods. The first method applies to all algorithms in consideration, while the next two are specific to greedy K–means.

6.5.1 Random centroid initialization

Perhaps the most common initialization method for K–means type algorithms is the following to sample K points as centroids uniformly at random from the data (without replacement), and assigning each point to its nearest centroid in terms of Euclidean distance. We refer to this as random centroid initialization (or just random initialization).

This is not the only commonly used initialization method for K–means type algorithms, nor is it typically the for ordinary K–means. For instance, K–means++ [3] is an alternative initialization method based on a non-uniform random sampling of points as centroids from the data. It has good theoretical and practical properties, *specifically* for K–means. Intuitively, K–means++ tends to give an initialization that is in some sense close to the K–means optimum, and hence often results in a better solution than simply picking centroids uniformly at random. However, this runs counter to our goal when using random initialization, which is to explore solutions to the sparse clustering objectives that are *different* from the K–means initialized solutions.

6.5.2 Random support initialization

The greedy K–means objective, along with the iterative solution methods in consideration, suggest an alternate initialization. Instead of beginning the algorithm by setting a value for the cluster assignments, we could instead initialize the support set for the feature weights w. Specifically, if we are searching for a p-sparse solution, we can pick w such that $\sum_j w_j = p'$ uniformly at random for some p'. The obvious choice for p' is $p' = p$, but picking a slightly larger value may be beneficial (see Section 7.3.6).

6.5.3 Support covering initialization

We also consider a non-random method for initializing w in greedy K–means. A brute force exhaustive search over all $\binom{d}{p}$ possible values of p-sparse weights w quickly becomes intractable. However, as we will see in Section 7.3.6, it is reasonable to expect that an optimal p-sparse weight vector w^* can be found by initializing with some w that has $\sum_j w_j = p' > p$, and $\sum_j w_j^* w_j = q < p$. In other words, we may expect to find a near-optimal p-sparse feature set by iterating over the set of all initial w with some $p' > p$ non-zero entries such that for any p-sparse w^*, there is some w that overlaps with w^* on at least q features. The design of such a set of initial values for w is identical to the covering design problem, which is defined as follows [10].

Given integers $q \le p' \le d$, let $U = [1..d]$ and $\mathscr{A} = \{A \subset U : |A| = q\}$. Then a (d, p', q)-covering design is a set $\mathscr{B} \subseteq 2^U$ such that each $B \in \mathscr{B}$ has $|B| = p'$, and that for all $A \in \mathscr{A}$ there exists $B \in \mathscr{B}$ with $A \subseteq B$. Exactly computing optimal covering designs, or even their sizes, is a hard problem, but there is an online repository [9] of covering designs for a wide range of values.

For instance, for the example in Section 7.3.6 where we seek a solution with $p = 5$ features out of $d = 30$, we use a $(30, 10, 4)$-covering from the repository (i.e. $p' = 10$ and $q = 4$) of size 210, which is quite close to a theoretical lower bound of 165. For comparison, in this example $|\mathscr{A}| = 27405$, and $\binom{30}{5} = 142506$.

6.6 Distribution UNSHRINKABLE

This two-dimensional dataset composed of three Gaussian components is generated as follows. First, we sample 400, 200, and 200 points respectively from unit covariance Gaussians with means $(2, 3)$, $(-9, -1)$, and $(-5, -1)$. Then, we normalize both dimensions of the sample to zero mean and unit variance.

6.7 Distribution TWO-CLUSTER

For any $d \geq 2$, we generate a random d dimensional data set which has one "relevant" feature, and $d-1$ irrelevant, i.e. noise features, as follows. In the first dimension, which is the feature, we sample $n/2$ points from $\mathcal{N}(-\mu, 1-\mu^2)$ and $n/2$ points from $\mathcal{N}(\mu, 1-\mu^2)$, where $\mu = 0.975$, and $1-\mu^2$ is used for the variance of each component so that the overall variance is 1. All remaining dimensions are noise features, and we sample $X_{i(j)}$ independently from $\mathcal{N}(0,1)$ for $i \in [n]$ and $j \in [2..d]$. We use $n = 100$ samples in all experiments with this distribution. A sample from this distribution for $d = 2$ is shown in Figure 2. Of course, this is merely a toy distribution for evaluating the methods described in Section 6, and in practice the problem of selecting only one feature for clustering is neither challenging, nor is it likely to be of significant interest.

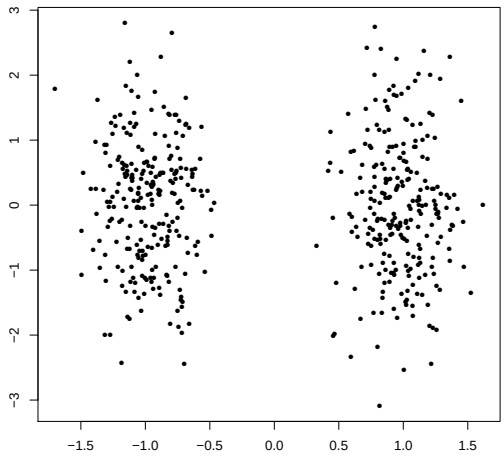


Figure 2: A sample of $n = 500$ points from the TWO-CLUSTER synthetic dataset.

6.8 Distribution TWO-GROUPS

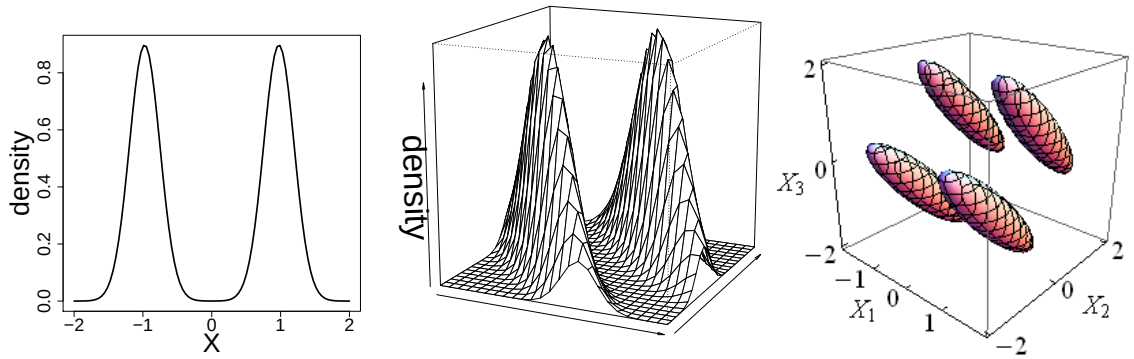
In this section, we describe a type of distribution that relates to a different notion of feature selection than that of Section 6.7. Instead of thinking of dimensions as relevant vs. noise, here we think of clusterings of the data as sparse or non-sparse based on the size of the groups of dimensions they are based on, i.e. a feature is relevant if it is part of the smallest group of features that gives a good clustering. This is a more general view of feature selection for clustering. It is also likely to be more practically useful, since for any given data set it is not realistic to expect that there are a large number of dimensions that, jointly, do not contain any significant cluster structure.

Specifically, we will have two groups of features, one much smaller than the other, so that each group of feature is composed of two well-separated clusters, but the respective cluster labels are independently distributed. When searching for a clustering into $K = 2$ partitions with a certain level of sparsity, the smaller group of features will be the better choice.

For $\mu \in (0,1)$ and $d \geq 1$, define the distribution Pancakes(μ, d) over $\mathbb{R}^d$ as the mixture

$$\frac{1}{2}\mathcal{N}\left(-\mu\frac{\mathbf{1}_d}{\sqrt{d}}, I_d - \mu^2\frac{\mathbf{1}_d\mathbf{1}_d^T}{d}\right) + \frac{1}{2}\mathcal{N}\left(\mu\frac{\mathbf{1}_d}{\sqrt{d}}, I_d - \mu^2\frac{\mathbf{1}_d\mathbf{1}_d^T}{d}\right)$$

where $\mathbf{1}_d$ and I_d are the d dimensional all ones vector and identity matrix, respectively. Pancakes(μ, d) has mean 0 and unit covariance, and for large μ is concentrated near two parallel discs (see Figures 3a and 3b). Since for large μ the two mixture components are nearly disjoint, the population within cluster sum of squares given by the component indicators is approximately $d - \mu^2$.



(a) Pancakes$(c,1)$ (b) Pancakes$(c,2)$ (c) $d_1 = 1, d_2 = 2, \mu_1 = \mu_2 = c$

Figure 3: Figures 3a and 3b — densities of the one and two dimensional Pancakes ($c = 0.975$). Figure 3c — level set of product of a one dimensional and a two dimensional Pancakes.

Now consider a random $X \in \mathbb{R}^d$ given by $X = (X_1, X_2)$, where $X_1 \sim$ Pancakes(μ_1, d_1) and $X_2 \sim$ Pancakes(μ_2, d_2) independently, and $d = d_1 + d_2$ (Figure 3c). Also, let L_1 and L_2 be the indicators of the component X_1 and X_2 are drawn from.

Marginally, the features $S_1 = [1..d_1]$ and $S_2 = [d_1 + 1..d]$ of X, corresponding to X_1 and X_2, are each composed of pairs of well separated components. Hence, we can imagine the distribution of X as being composed of four clusters defined by the pair of indicators (L_1, L_2). However, there is no "good" partition of $\mathbb{R}^d$ into $K = 2$ clusters in terms of X.

We use the values $d_1 = 5$, $d_2 = 25$, $\mu_1 = 0.975$, and $\mu_2 = 0.999$ in Section 7.3.5, so that the first group of features defines a feature-sparse clustering, and the second group of features defines a less sparse clustering with a better between-cluster separation.

6.9 Hybrid modeling mixed variable types

For data with mixed variable types, it is not immediately clear that K–means type methods are ideal. The K–means objective can be thought of as the (negative) log likelihood of the data under a Gaussian mixture model with identity covariances. For mixed variable types, then, an obvious modification would be to model each variable with the appropriate distribution, which we refer to as a hybrid model. We model each feature within a cluster as an independent univariate Gaussian for continuous features; a Binomial for ordinal (discrete) features; and a categorical random variable for categorical (nominal) features. The hybrid (hard-assignment) maximum likelihood criterion gives the objective

$$\max_{L \in [K]^n, \theta} \prod_{k=1}^{K} \prod_{i:L_i=k} \prod_{j=1}^{d} f_{m(j)}(X_{i(j)}; \theta_{k(j)})$$

where $f_{m(j)}$ is the density of the model chosen for feature j, and $\theta_{k(j)}$ are the parameters of the model. Equivalently, this objective can be stated in terms of maximizing the log likelihoods. If there are d_G, d_B, and d_c

Gaussian, Binomial, and categorical random variables, respectively, then the objective can be written as

$$\max_{L\in[K]^n} \left[\max_{\mu,\sigma\in\mathbb{R}^{K\times d_G}} \sum_{j=1}^{d_G}\sum_{i=1}^{n} \log\frac{1}{\sigma_{L_i(j)}} - \frac{(X^G_{i(j)}-\mu_{L_i(j)})^2}{2\sigma^2_{L_i(j)}} \right] + \left[\max_{p^B\in[0,1]^{d_B}} \sum_{j=1}^{d_B}\sum_{i=1}^{n} X^B_{i(j)}\log p^B_j + (s_j - X^B_{i(j)})\log(1-p^B_j) \right] + \left[\max_{p^c_{(j)}\in\mathscr{P}_{m_j}, j\in[d_c]} \sum_{j=1}^{d_c}\sum_{x=1}^{m_j}\sum_{i:X^c_{i(j)}=x} \log p^c_{x(j)} \right] \tag{5}$$

where s_j are the sizes of the binomial features (hence $X^B_{i(j)} \in \{0,\dots,s_j\}$), m_j are the number of categories for categorical features ($X^c_{i(j)} \in \{1,\dots,m_j\}$), and $\mathscr{P}_m = \{p \in [0,1]^m : \sum_{i=1}^m p_i = 1\}$ is the probability simplex.

Optimizing (5) with a coordinate ascent procedure would be straightforward. The cluster assignment step is obvious, and the maximum likelihood parameter values are simply empirical per-cluster means, variances, and category probabilities. We could also directly incorporate feature selection analogous to, say, greedy K–means.

7 Results

7.1 Optimizing the Regularized K–means objective

Recall the sub-problem (2) in the iterative solution to Regularized K–means proposed by [24] (Section 6.2):

$$C_{(j)} = \operatorname*{arg\,min}_{y\in\mathbb{R}^K} \frac{1}{n}\sum_{k=1}^{K}\sum_{i:L_i=k}(X_{i(j)}-y_k)^2 + \lambda_j\|y\|$$

which must be solved for each $j = 1,\dots,d$, with L fixed. This problem can be stated in terms of a group LASSO penalized regression objective and solved e.g. with the R package **grplasso** [18]. The design matrix required to do so could be a matrix in $\{0,1\}^{n\times K}$ where row i contains exactly one non-zero entry in position L_i. However, the special structure of this design matrix allows for a more direct solution. Specifically, for each j, by taking the subdifferential of (2) we find

$$C_{k(j)} = \begin{cases} 0 & \text{if} \quad \frac{2}{n}\|\vec{n}\circ\widetilde{C}_{(j)}\| \le \lambda_j, \\ \left(1+\frac{\lambda_j n}{2\vec{n}_k\|C_{(j)}\|}\right)^{-1}\widetilde{C}_{k(j)} & \text{o.w.} \end{cases} \qquad \text{for } k = 1,\dots,K \tag{6}$$

where $\circ$ is the Hadamard (entrywise) product, $\vec{n} \in \mathbb{R}^K$ is the vector of cluster sizes, $\widetilde{C}$ are the unpenalized cluster centers (dependence on L is kept implicit for brevity). Note that this is a closed form solution for deciding if $C_{(j)} = \mathbf{0}$, but not for finding $C_{(j)}$ when it is non-zero since $\|C_{(j)}\|$ appears in the second line. However, when $C_{(j)} \neq \mathbf{0}$, by taking the squares of both sides of (6), summing over k, and dividing by $\|C_{(j)}\|^2$, we have

$$1 = \sum_{k=1}^{K}\left(\|C_{(j)}\| + \frac{\lambda_j n}{2\vec{n}_k}\right)^{-2}\widetilde{C}^2_{k(j)}.$$

This equation, when $\frac{2}{n}\|\vec{n}\circ\widetilde{C}_{(j)}\| > \lambda_j$, has a unique solution $\|C_{(j)}\| \in (0, \|\widetilde{C}_{(j)}\|]$ that can be found numerically using any number of univariate solvers. Once the value of $\|C_{(j)}\|$ is known, $C_{(j)}$ itself can be obtained in closed form from (6). From our experience, this method has a significant performance advantage over using the group LASSO formulation.

7.2 Hybrid likelihood criterion for binary features

The mixed modeling approach of Section 6.9 replaces per-cluster variance as the minimization objective with entropy. Figure 4 shows the cluster variance and entropy (up to a multiplicative constant) of a binary feature modeled as a Bernoulli random variable, which measures the contribution of the feature to the total within cluster sum of squares and log likelihood of a clustering, respectively. The high similarity of the two curves indicates that the two criteria may be near equivalent in practice. Considering the complexity of the hybrid criterion, and the fact that standard K–means algorithms can not be used directly when optimizing it, hybrid modeling may be unnecessary for binary features.

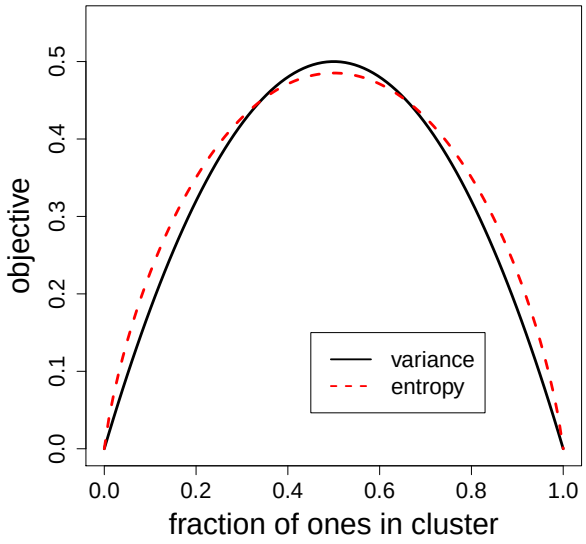


Figure 4: Variance and entropy of a binary feature (up to a multiplicative constant).

7.3 Synthetic experiments

7.3.1 Effect of direct cluster center penalization on UNSHRINKABLE

The Regularized K–means idea of directly penalizing cluster parameters (similar approach taken in [11]) can have a potentially undesirable effect.

In Figure 5 we show the Regularized K–means solutions (initialized by the K–means solution) with increasing sparsity parameter λ on the UNSHRINKABLE dataset (Section 6.6). The horizontal axis alone is sufficient to differentiate the three original clusters. However, as λ increases, all three cluster centers are shrink towards zero, until the cluster shown in blue is left nearly empty. In this example, Sparcl (for s such that one feature is selected) and greedy K–means (with $p = 1$) identify the horizontal axis as the relevant feature. The resulting clustering is identical to what we would obtain by applying K–means to the horizontal axis directly.

7.3.2 K–means clustering of TWO-CLUSTER

Using the TWO-CLUSTER data described in Section 6.7, Figure 6 shows the clustering error achieved by K–means on this random data, as a function of the number of noise dimensions (i.e. $d - 1$). Note that 0.5 is the maximal possible clustering error between any two clusterings with $K = 2$. The K–means solutions here were computed with multiple random restarts of the Hartigan–Wong [12] algorithm for K–means. This is the default method used in the R implementation of K–means, and it performs very well in practice (in terms of

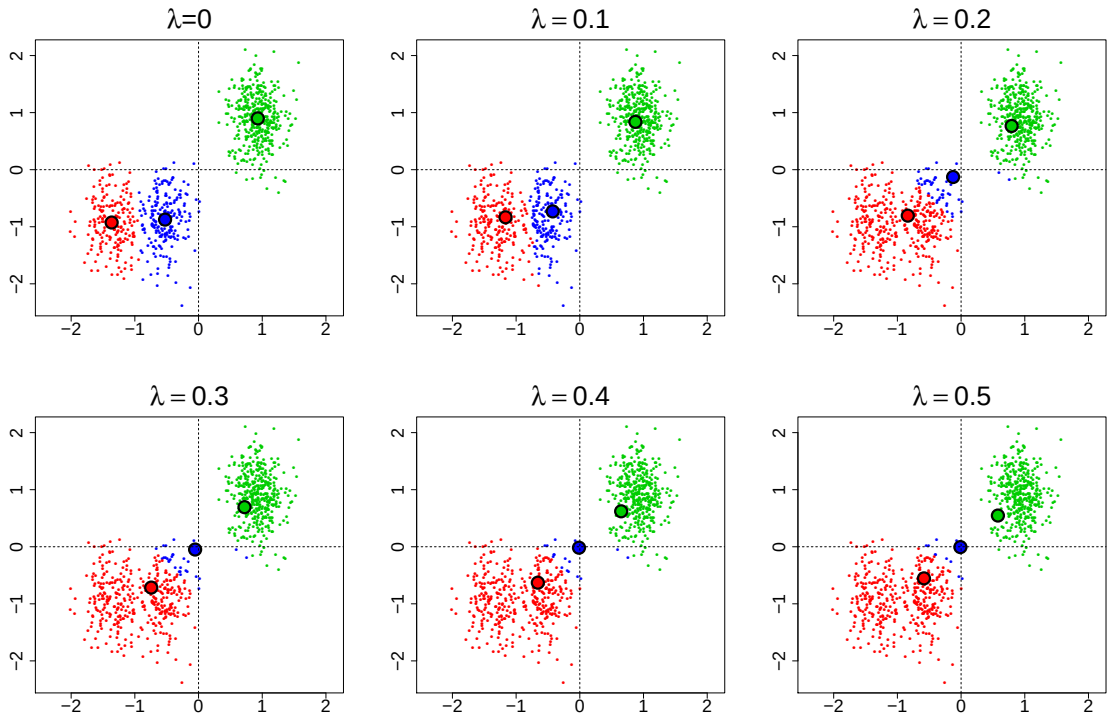


Figure 5: The clustering given by Regularized K–means in a two dimensional dataset with increasing regularization parameter for $K = 3$. Color indicates cluster membership, and the three large points are the cluster centers. In the last three plots, the cluster corresponding to the middle centroid is nearly empty.

finding a solution with optimal or near-optimal objective value). Hence, the error curve in Figure 6 appears to characterize the K–means objective itself, and not the approximate algorithm we use to solve it.

7.3.3 Difficulty of approaching optimal objective value

In Figure 7, we show the distributions of the objectives of the K–means initialized solution and the random centroid initialized solution of the three methods in Section 6 on the same data as above. We use 100 draws of the TWO-CLUSTER dataset (Section 7.3.2) with $d = 11$ and $d = 21$ (i.e. 10 and 20 noise dimensions, respectively). For each dataset, we initialize each algorithm 20 times with random centroids. We use the objective given by initializing with the true labels as a baseline to compute suboptimality, so that results from different random draws of the data can be directly compared. Initializing with the true labels also gave solutions that were the same (or nearly the same) as K–means using only the relevant feature, which indicates that the design of the synthetic distribution is such that the true optima of all three algorithms in consideration, if found, are ideal for feature selection. The figures show the rapid fall in the probability that initializing with K–means will result in a nearly optimal solution, as well as the difficulty of using random centroid initialization.

We also observed that there was no strong dependence between the degree of suboptimality of the K–means initialized solutions, and the randomly initialized solutions (note that Figure 7 is insufficient to draw this conclusion).

7.3.4 Effect of feature selection on clustering error for TWO-CLUSTER

Figure 8 shows the performance in terms of clustering error of all methods (initialized with K–means) applied to the same data distribution used above, with parameters set so that exactly one feature is selected. The

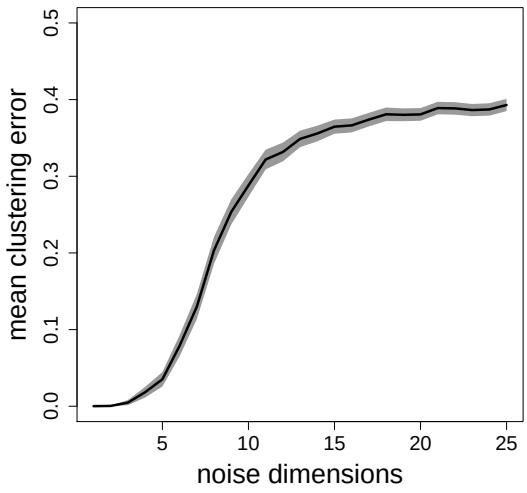


Figure 6: K–means clustering error for the dataset of Section 7.3.2 (with 95% Gaussian confidence intervals).

number of dimensions ranges from 1 to 15 (i.e. the data is from $d = 2$ to $d = 16$ dimensional). There is no significant difference in the clustering error of the four methods in terms of average clustering error, but there is one when we measure the percentage of random draws where the clustering error is exactly zero. This is due to the fact feature selection results in zero error when the correct feature is identified, but in general *increases* the error when an incorrect feature is selected. In *all* random instantiations, initializing with the true cluster labels resulted in zero error, as in the previous section.

A more detailed look at the $d = 11$ case To better understand these results, we look in more detail at the results using $d = 11$ dimensions, i.e. 10 noise features (Figure 10). We pick sparsity parameters for all three methods so that exactly one non-zero feature is selected.

Figure 9 shows that for all three methods being considered (initialized with K–means), there appears to be a critical value for the clustering error of the K–means solution, such that above that value feature selection doesn't help, but below that value feature selection identifies the relevant feature, and reduces the clustering error to 0.

Figure 11 shows a heatmap of the joint distribution of the clustering errors of K–means and Sparcl (density estimate is over-smoothed — all observed errors less than 0.2 for Sparcl were exactly 0). A similar distribution holds for all three methods. Unsurprisingly, for all three methods zero clustering error also coincides exactly with correct identification of the relevant feature.

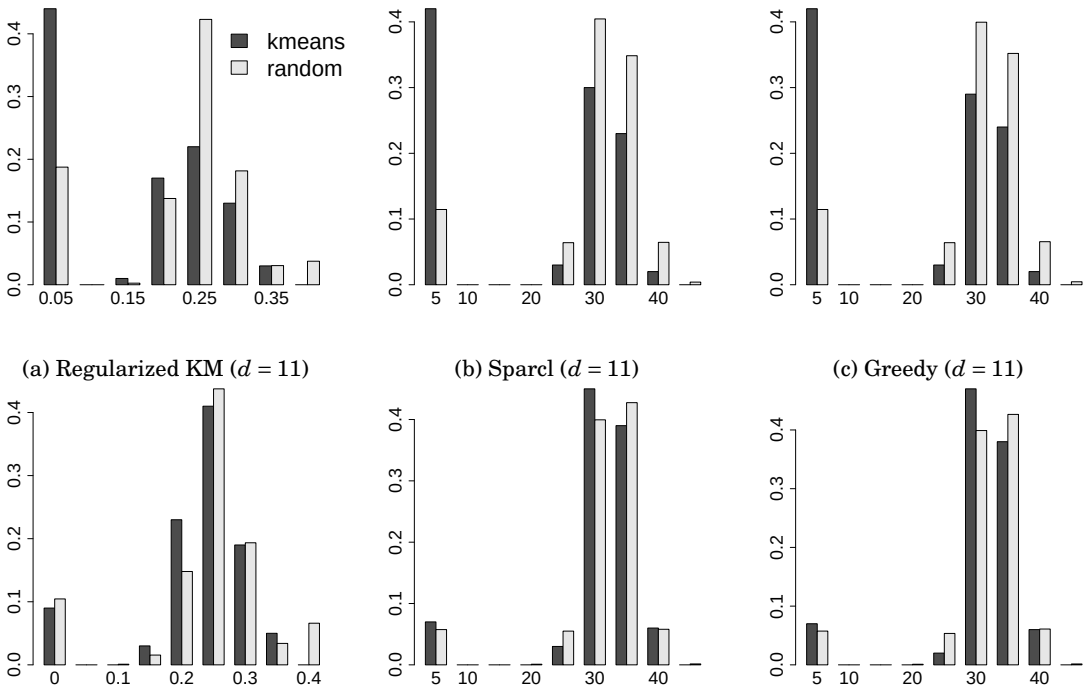
7.3.5 K–means initialization with infinite data on TWO-GROUPS

In this section we analyze the results given by K–means initialization using infinite data from the distribution described in Section 6.8. For simplicity, in this section and the next we focus only on the greedy K–means objective.

When μ_1 and μ_2 are large (near 1), the population within cluster sum of squares for the $K = 2$ clusterings given by L_1 and L_2 are approximately

	WCSS on S_1	WCSS on S_2	Total
L_1	$d_1 - \mu_1^2$	d_2	$d_1 + d_2 - \mu_1^2$
L_2	d_1	$d_2 - \mu_2^2$	$d_1 + d_2 - \mu_2^2$.

Note that, since X_1 and X_2 are drawn independently, the mutual error of the clustering based on L_1 and L_2 is exactly 0.5.



(a) Regularized KM ($d = 11$) (b) Sparcl ($d = 11$) (c) Greedy ($d = 11$)

(d) Regularized KM ($d = 21$) (e) Sparcl ($d = 21$) (f) Greedy ($d = 21$)

Figure 7: Histograms of suboptimality in the objective resulting from K–means initialization and random centroid initialization (probability over all randomness on the vertical axis). For each dataset 20 random initial clusterings were used, and overall 100 random datasets were drawn with $d = 11$ and $d = 21$ dimensions. Parameters were fixed to $\lambda = 0.3$ for Regularized K–means, $s = 1.01$ for Sparcl, and $p = 1$ for greedy K–means, which were chosen so that a typical solution was concentrated on a small number of dimension. The horizontal axis is the (absolute value of) the difference between the objective of a solution and the best approximation of the true optimum we were able to obtain.

Since $d_1 = 5$, $d_2 = 25$, $\mu_1 = 0.975$, and $\mu_2 = 0.999$, the K–means solution with $K = 2$ will be that given by L_2, and the features in S_1 will be irrelevant to the resulting clustering. On the other hand, the greedy K–means solution with $p = 5$ should select exactly the features in S_1, since the clusters corresponding to S_2 are not well-separable with such few features, as is easy to verify empirically.

Initializing greedy K–means for $p = 5$ (or, in fact, for any $p \leq d_2$) with the clustering given by L_2 will always result in w such that $w_j = 0$ for any $j \in S_1$. This is because, due to the statistical independence of X_1 and X_2, the conditional distribution of X_1 conditioned on $L_2 = 1$ is identical to the distribution of X_1 conditioned on $L_2 = 2$, and so the between cluster sum of squares for all $j \in S_1$ will be exactly 0 for any clustering based on any subset of S_2.

7.3.6 Random centroids, random features, and set covers

In the previous section we showed that applying greedy K–means to the TWO-GROUPS distribution with K–means initialization does not result in the optimal solution, even with inifinite data. In this section, we would like to evaluate the performance of random centroid initialization (Section 6.5.1) on the same distribution, as well as random support initialization (Section 6.5.2) and covering initialization (Section 6.5.3). As in the previous section, here we consider only the greedy K–means objective, though it may be possible to obtain similar results for Sparcl and Regularized K–means as well.

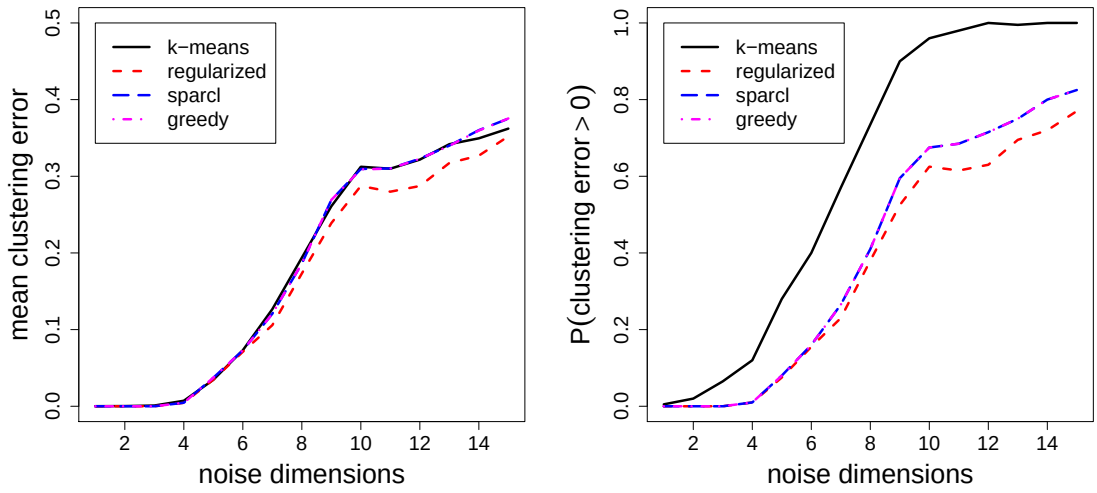
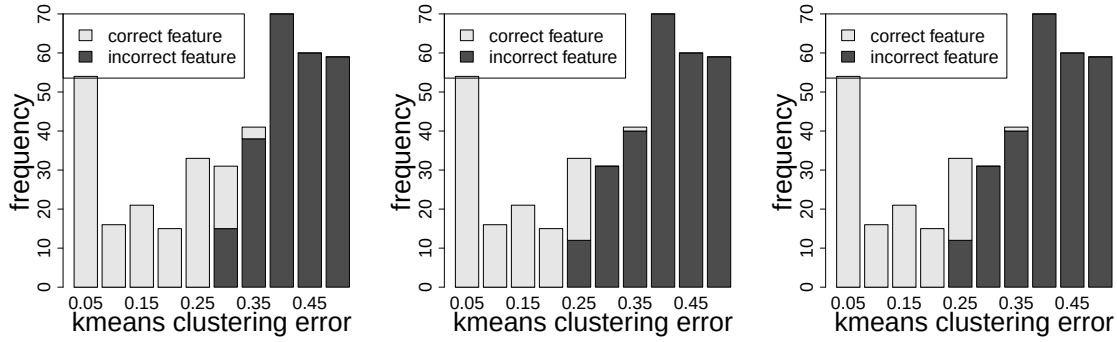


Figure 8: Errors are calculated based on the sparsity parameter that results in a single selected feature, and averaged over 200 random draws of the data set.



(a) Regularized KM (b) Sparcl (c) Greedy

Figure 9: Histograms of clustering error of K–means, separated to show the conditional distribution of whether each method identifies the relevant feature or not (out of 400 random samples with $d = 11$ features).

Since we can't actually work with infinite data, we sampled a single data set with $n = 1000$ points from the distribution with the fixed parameter settings stated in the last section. More accurately, instead of sampling directly from that distribution, we sampled $n/4 = 250$ points from the conditional distributions for all four values for the pair (L_1, L_2), to avoid confounding the results with uninteresting random factors. As expected, the K–means solution[1] agreed exactly with L_2, i.e. the clustering given by the group of $d_2 = 25$ features in S_2. Of course, this also meant that the error between the K–means solution and the clustering L_1, given by the group of $d_1 = 5$ features in S_1, was 0.5 (i.e. maximal).

Initializing greedy K–means for $p = 5$ with the K–means solution resulted in a clustering that had error 0.474 with L_1 (i.e. S_1 was not identified), and error 0.304 with L_2. Indeed, we designed the Pancakes distribution specifically so that its clustering could not be approximated based on a small subset of its dimensions (namely 5 out of 25, in this case).

We performed 2000 rounds of random centroid initialization, and in exactly *one* instance did we reach zero clustering error. In every other case, the error was least 0.245 (Figure 12a). (Note that, by design of this data set, zero clustering error is achieved after selecting $p = 5$ features if and only if the features in S_1 are selected.)

[1]We were able to obtain what appeared to be the optimal K–means solution using multiple restarts of the Hartigan–Wong algorithm.

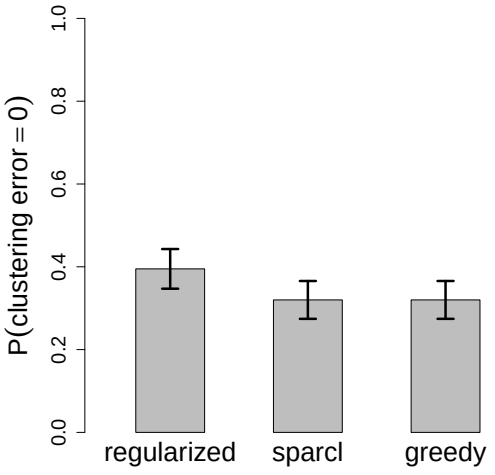


Figure 10: Fraction of random runs (out of 400) with 0 clustering error (±95% CIs).

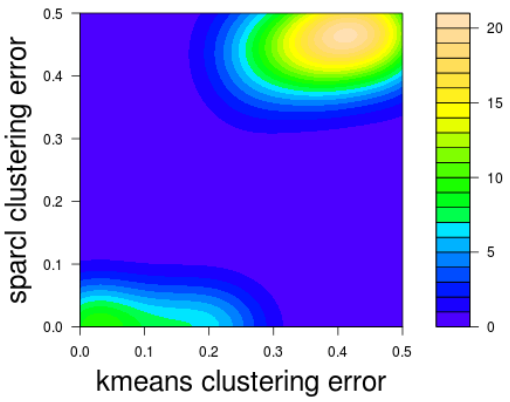


Figure 11: Bimodal distribution of clustering error after feature selection.

Figure 12b shows the results of greedy K–means after 1000 random support initializations (Section 6.5.2) with $\sum_j w_j = p' = 10$. We use $p' = 10$, and not $p' = 5$, because we believe that the algorithm is more robust to the presence of irrelevant features (those in S_2) in the initial w, than to the absence of relevant features (those in S_1). For reference, note that there are $\binom{30}{5} = 142,506$ possible values for w with $\|w\|_1 = 5$.

Out of those 1000 random initializations of w, the optimal solution was found in 9 cases. Of these, in 2 instances all relevant features were included in the initial w (along with 5 irrelevant features), and in the other 7 instances four of the relevant features were included (along with 6 irrelevant features).

Using the 210 elements of a $(30, 10, 4)$-covering (Section 6.5.3) for initialization, we obtain the results in Figure 12c. We saw above that we can tolerate not including all 5 relevant features in the initialization, which is why we chose to use a $(30, 10, 4)$-covering. Three out of the 210 elements of the covering result in an optimal solution of greedy K–means. One had all 5 relevant dimensions in the initial set, and the other two had 4.

7.4 Lung disease data

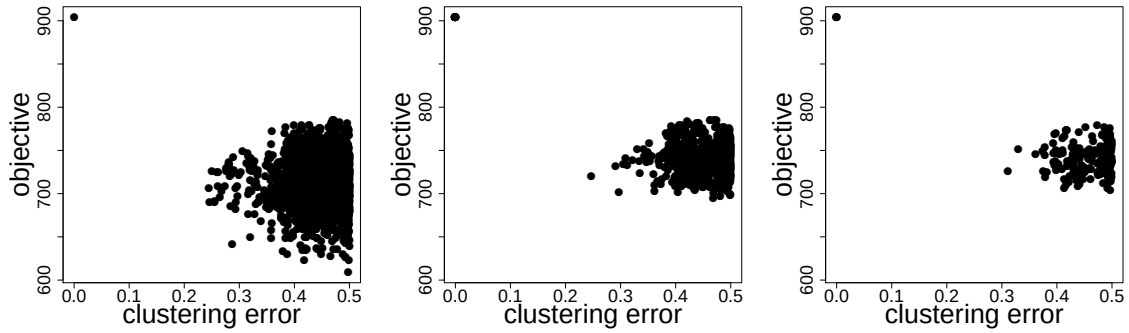
In this section we apply the methods discussed above to the lung disease dataset described in Section 4.

For K–means initialized results, a K–means solution was computed using the Hartigan–Wong algorithm [12], and appeared stable over multiple restarts.

We express the categorical variables through dummy indicator features, which is the common treatment of such features for K–means and similar methods [13, 1]. We standardize each dimension to mean 0 and variance 1, except the dummy features for the categorical variables, which we scale so that the total variance of each group of dummy features is 1.

7.4.1 Choice of K

In clustering, choosing the number of clusters is a famously difficult problem, with a large volume of work on the subject (see e.g. [26, 6, 21, 32]). Specifically in the context of Regularized K–means, [24] proposed a method to do so based on a type of bootstrap procedure. Here we use the value $K = 6$, which was the number of clusters chosen in a previous analysis of the dataset by [30].



(a) Random centroid (b) Random support (c) Support covering

Figure 12: Scatterplots of clustering error vs. objective of three greedy K–means initialization methods (higher objective preferred). In the second and third plots, the point in the top left corner, corresponding to the best objective, has multiplicity 9 and 3, respectively.

7.4.2 Decreasing cluster sizes found by Regularized K–means

We showed an example in Section 7.3.1 of potentially undesirable results from the direct cluster center penalization of this method. It appears that something similar may be occurring here — for $\lambda \approx 0.29$, with 63 non-zero features remaining (over half), two of the 6 clusters contain only 4 points (with K–means initialization). When $\lambda \approx 0.47$, there are still 40 non-zero features, but now two clusters contain 2 points each, and a third cluster only 7. Sparcl and greedy K–means do not share this behavior — the smallest clusters contain typically at least 20 points regardless of the degree of sparsity. For this reason, we do not report results from the Regularized K–means method henceforth.

7.4.3 Objective values

Here we compare the results given by Sparcl and greedy K–means when sparsity is high for different initializations. In particular, we fix the parameter $s = 2.8$ for Sparcl which results in between 9 to 11 selected features, and for greedy K–means we select $p = 10$ features.

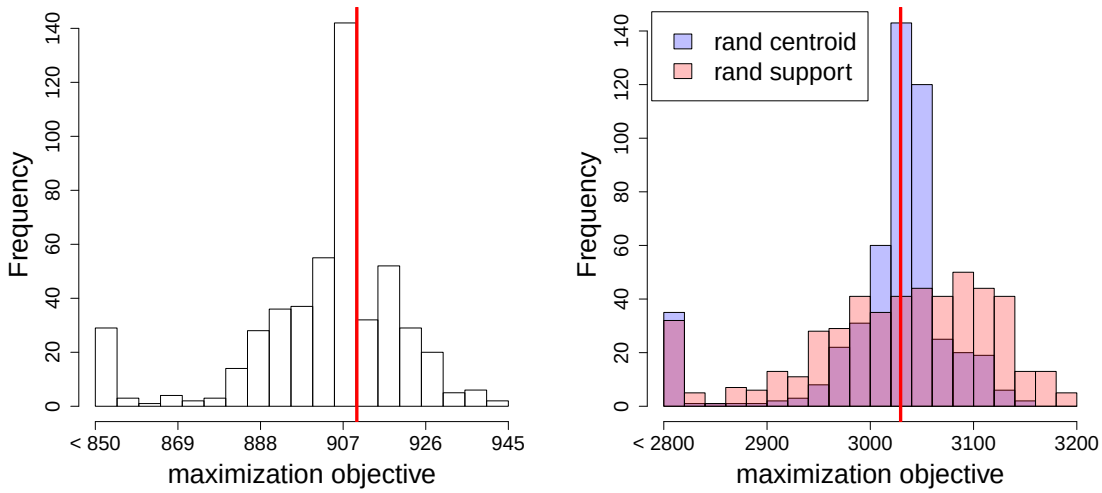
In Figure 13a we plot the distribution of the random centroid initialized Sparcl objective with the above parameters over 500 restarts, and that of the K–means initialized solution. Figure 13b shows the same for greedy K–means, along with 500 runs of the random support initialization described in Section 6.5.2 (with 30 initial random features). Unfortunately, support covering initialization (Section 6.5.3) is not an option here, since the overall number of dimensions is too high.

We make a few observations based on these results. First, K–means initialization does not lead to better objectives than the average random initialization. Second, random support initialization seems to have a significant advantage at the higher end of the distribution of objectives over random centroid initialization. Also, since the upper tail of the distributions of the objectives do not appear to decay very quickly, it would be reasonable to expect that with further random runs we would continue finding increasingly better solutions.

In the synthetic experiments above, the results of Sparcl and greedy K–means were near identical. Since random support initialization, which is defined more naturally for greedy K–means, performs better than random centroid and K–means initialization here, below we only present the greedy K–means results.

7.4.4 Selected features and correlation, clustering stability

In the previous section we saw that the objective of greedy K–means on the lung disease data given by random support initializations can be higher than by K–means initialization. Here we compare the selected features for a range of values for the sparsity level p. We present the results with K–means initialization, and with 50



(a) Sparcl objectives

(b) Greedy K–means objectives

Figure 13: The Sparcl and greedy K–means objectives (higher = better) over 500 random centroid initializations, as well as 500 random support initializations (with 30 features) of greedy K–means. The vertical lines show the K–means initialized solutions.

random support initializations with size of initial set chosen according to

$$30 + (p-1)\frac{d-30}{d-1}$$

where d is the total number of features and p is the given sparsity level. In all instances with $p \leq 90$ features selected, the highest objective from random support initialization was higher than that of K–means initialization.

Figure 14 shows the selected features by K–means initialization, and the average w values given by the random support initializations (black = 1, white = 0), in terms of the objective. In both cases, features are ordered according to the order with which they appear in the K–means initialized solutions as p increases. Though largely similar, there are key differences between the two. For instance, two of the features selected by K–means initialization with $p \leq 12$ are ranked much lower by the second method.

Figure 15 shows, for each p, the distance between each pair of clusterings with the 5 highest objective values as computed above. The clustering distance is measures as the fraction of points where the two cluster label vectors disagree (minimized over the equivalent cluster label permutations). What is noteworthy here is the large variation in clusterings despite the similarity of the corresponding objective values.

Figure 16 is a box plot of the absolute values of the rows (or equivalently columns) of the covariance matrix of the data. The features are ordered according to the highest average w values in the top 5 solutions (in terms of objective) given by random support initialization. The overall trend is obvious — features with higher overall correlations are selected earlier — but there are exceptions. For instance, the set of correlations of the first 4 features are lower than the next 4 in terms of the median and confidence intervals. However, the feature correlations in the first group take on more outlying values on the higher end. In fact, from Figure 17, which shows the covariance matrix between the first 50 features as ordered in Figure 16 (with black = 1, white = 0), we see that the group of first 4 features appear to be more highly correlated with *each other*, than any other group.

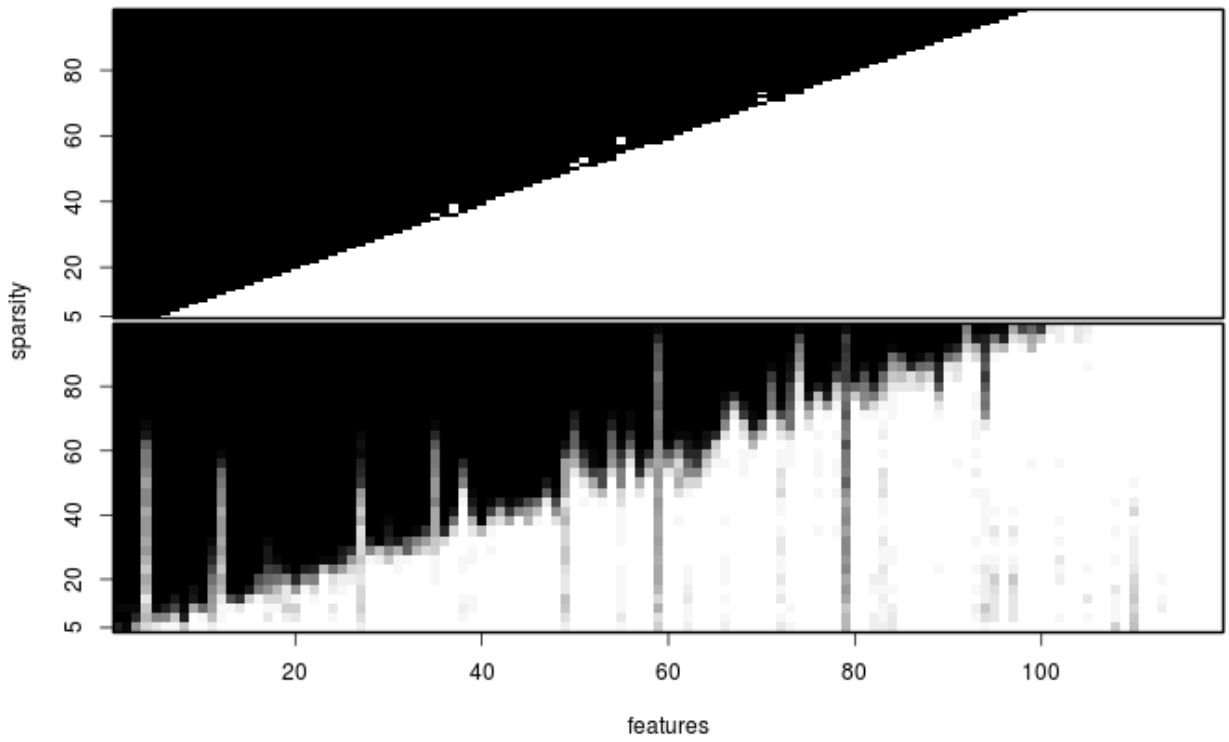


Figure 14: Comparison of features selected by greedy K–means with K–means initialization (top) and random support initialization. In the top figure, black indicates that a feature was selected. In the bottom figure, darkness indicates the fraction of random initializations where a feature is selected. Features ordered according to order of appearance in K–means initialized solutions.

8 Discussion

The performance results of the three methods in consideration — Regularized K–means, Sparcl, and greedy K–means — are nearly identical on the synthetic experiments of Sections 7.3.2-7.3.6.

Section 7.3.4 shows a small (but statistically significant) advantage of Regularized K–means over Sparcl and greedy K–means on the TWO-CLUSTER dataset (Figure 10). We don't have an explanation for this apparent advantage of Regularized k–means. We hypothesize that the difference may be due to the fact that in this particular dataset, the Regularized K–means penalty, which effectively constrains cluster centers to be in a small ball around the origin, acts in a similar manner as the re-weighting scheme used in [4], hence slightly increasing the signal to noise ratio of the data.

However, if this hypothesis is correct, then this small superiority of Regularized K–means would likely not generalize to more complex datasets. Moreover, the three cluster example in Section 7.3.1 with the UNSHRINKABLE distribution demonstrates a setting where Regularized K–means gives results that are difficult to justify as advantageous in any situation. There, the direct penalization of cluster centers in this method results in the merging of (feature-sparse) clusters that appear well-separated. In fact, it is not difficult to design examples where Regularized K–means would find only empty or near-empty clusters due to this type of penalization. We observe similar behavior on the lung disease data (Section 7.4.2), and it appears that the higher dimensionality and larger value of K exacerbate the issue.

The difference between Sparcl and greedy K–means is more subtle. Based only on our synthetic experiment results, the two methods are exactly identical. Also, though we only apply the random support (Section 6.5.2) and support covering (Section 6.5.3) initialization techniques to greedy K–means, they can easily be adapted

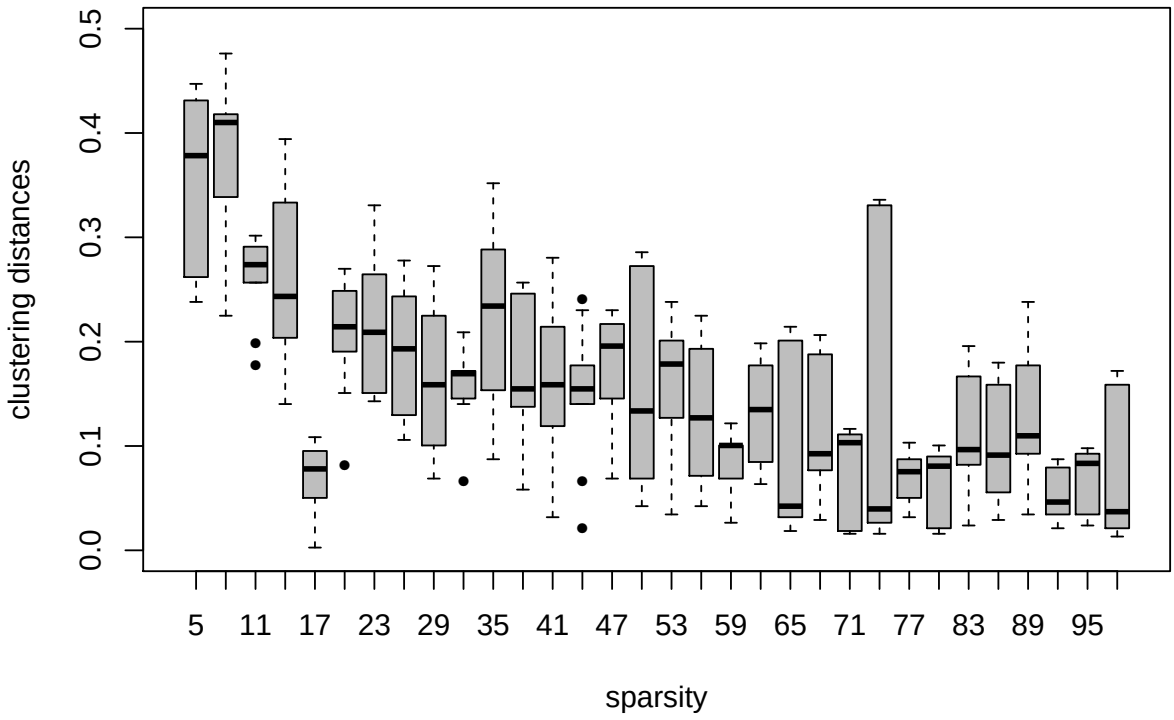


Figure 15: Clustering distances between all pairs of top 5 solutions (in terms of the objective), for each sparsity level p.

to Sparcl as well. The most significant apparent differences are that greedy K–means gives features weights that are exactly 0 or 1, while in Sparcl a feature may have a weight that is arbitrarily small but non-zero; also, while greedy K–means is identical to K–means for $p \geq d$, the same is not in general true of Sparcl for any value of the sparsity parameter s. Specifically, when $s \geq \sqrt{d}$, though the sparsity inducing L1 constraint on the feature weights w in Sparcl is inactive, the constraint $\|w\|^2 \leq 1$ remains active. Hence, Sparcl will result in weights w where features with lower within-cluster variance are weighed higher. For these reasons, the greedy K–means solutions may be more easily interpretable in practice.

In all synthetic experiments with the TWO-CLUSTER and TWO-GROUPS datasets, the solutions given by initializing with the true cluster labels performed perfectly in terms of feature selection and clustering error. Moreover, the objective values of the resulting solutions were better (higher or lower, as appropriate) than those solutions that did not identify the correct features and clustering. In other words, with the exception of the undesirable behavior of Regularized K–means on the UNSHRINKABLE data, the issue at hand is the problem of finding a sufficiently good approximation to the true optimum of the nonconvex objectives in consideration, rather than the properties of the optimal solutions themselves.

The prescribed approximation methods in the original proposals of Regularized K–means [24] and Sparcl [28] both used a K–means clustering of the data to initialize iterative coordinate descent type algorithms. The results of Section 7.3.3 demonstrate that such an initialization can often give poor solutions. The random alternatives that can eventually find better solutions require large numbers of attempts. In a more realistic dataset, with hundreds of observations and dimensions, the number of required initializations to find a good solution may be even higher. Hence, it is important to better understand the relationship between the initialization of the methods and the qualities of the resulting solutions, and in particular the factors that determine when K–means initialization is successful.

Figures 9 and 11 (Section 7.3.4) demonstrate that in terms of clustering error, the feature selection methods

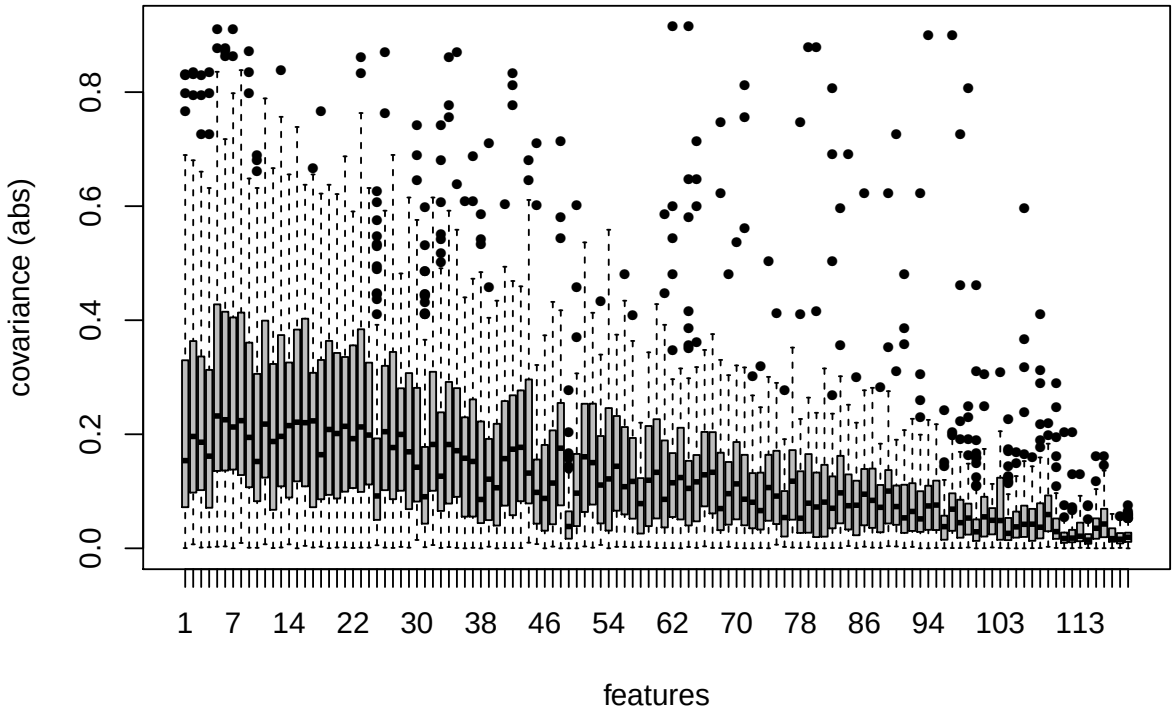


Figure 16: Box plot of rows (or columns) of absolute values of data covariance matrix. Features ordered according to result of random support initializations with top 5 objectives.

initialized by K–means succeed whenever the K–means solution itself is within some threshold of the true sparse clustering (in fact, the same holds for random initialization).

In Section 7.3.2, we see rapid deterioration in the percent of random draws of the distribution of Section 6.7 where the K–means solution agrees with the "true" cluster labels defined by the relevant feature. It is worth noting that the main reason why K–means gives solutions which are so different from the true cluster labels is due to random correlations in the data. Though in distribution this synthetic dataset has identity covariance, any given sample will not. As we increase the dimensionality of the data set (while keeping the sample size fixed at $n = 100$), the distribution of the empirical covariance will increasingly vary from identity. The K–means objective is sensitive both to linear and non-linear structure, and for data sets with highly non-spherical covariances, K–means can be seen as a discretized version of principal component analysis (PCA) [8]. We were able to verify the above intuition by applying K–means on the data used in Figure 6 after "whitening" it, i.e. transforming to identity empirical covariance, which in practice resulted in zero clustering error with very few exceptions.

Though it may seem like the above observation obviates the need for any feature selection methods on this type of data, we believe this not to be the case for several reasons. First, the whitening operation is only well-defined (well-behaved) when $d < n$ ($d \ll n$). Second, whitening in general results in features which cannot be expressed as sparse combinations of the original data features, so semantic interpretation of sparsity is no longer possible. And finally, as we see in Section 7.3.5, it is possible to design spherical datasets where the sparse clustering cannot be found with K–means. The results discussed above rely on random correlations in the data to produce K–means solutions that are significantly different from the clustering according to the relevant feature. With infinite data, initializing all three methods with the K–means solution would result in the relevant feature being identified. In Section 7.3.5, we show that for the TWO-GROUPS distribution described in Section 6.8, even with infinite data K–means initialization does not give the desired solution. Hence, the

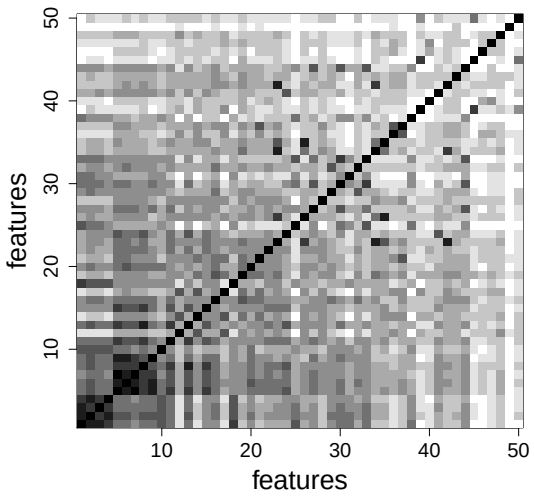


Figure 17: Covariance of first 50 features. Feature ordering same as in Figure 16.

TWO-GROUPS distribution demonstrates the limitation of iterative coordinate descent-type methods based on initialization with the K–means solution, *even with infinitely many samples*. In light of this observation, the advantage shown in Sections 7.3.6 and 7.4.3 of the support-based initialization methods we propose over the standard random centroid initialization method may be of significant practical value.

Correlations between features also play a role in the results obtained for the lung disease dataset, as seen in Section 7.4.4. The features selected by greedy K–means tend to be the ones that are highly correlated with others, even after multiple rounds of random support initialization. From the point of view of the objective, this is not surprising — it is a modification of the K–means objective, and as observed above, K–means is sensitive to linear structure. However, it is not clear whether the features selected by greedy K–means for any given value of p, and the corresponding feature-sparse clustering, imply the discovery of any non-linear structure that could not have been found using a sparse principal component analysis [33, 7]. Recently developed optimization methods for sparse PCA improve the practicality of the approach for moderately sized datasets [27]. It would be interesting to apply those methods to the lung disease data and compare the selected features with those of greedy K–means.

In Section 6.9 we proposed a hybrid modeling maximum likelihood method for dealing with mixed variable types. The behavior of this method on the lung disease dataset was difficult to compare to that of the K–means based methods, due to the instability of the iterative coordinate descent/ascent approximation algorithms and the complexity of the data. The similarity (up to a multiplicative constant) of the variance and entropy of a Bernoulli random variable demonstrated in Section 7.2 gives reason to believe that in a purely binary dataset, K–means based and Bernoulli likelihood-based methods would give similar results. (Or, at least, that the clustering given by one method would be near-optimal in terms of the objective of the other. The actual clusterings could be substantially different, since the non-convexity of the objectives in question allows for the existence of multiple distant near-optimal local optima.) However, this result does not give any indication of the behavior of hybrid modeling for non-binary ordinal and categorical features, as well as the effect of including the variance as a model parameter for Gaussian features.

9 Limitations and future work

This work is not intended to be a complete analysis of the lung disease dataset. Among the numerous possible directions for further inquiry is to compare the features selected in Section 7.4 with the features that previous analysis found to be highly informative of subject categorization (see Section 4).

We were only able to investigate two of the feature-sparse clustering methods from the literature. To our

knowledge, there is no existing comparison between the other methods mentioned in Section 3. While these methods have differences, many are similar in the manner in which they induce sparsity, which is often the addition of a typical sparsifying penalty to some non-convex clustering objective. Taking a unifying view of all such methods may be informative.

Our analysis of the mixed variable type modeling technique outlined in Section 6.9 was minimal. Due to the instability of the results of the random approximate algorithms that are available for optimizing the type of objectives in consideration, we found it difficult to make any meaningful comparison of the hybrid modeling method and the K–means based methods. A way to overcome this challenge would be to use further synthetic datasets comprised of discrete or mixed features, designed to be sufficiently simple so that the results of the different methods can be more easily interpreted.

Other possible future directions of inquiry:

- Though the K–means objective is in general NP-hard to solve, there are methods that under certain reasonable conditions are guaranteed to give good approximations to the true optimum. Similarly, though in general every objective considered in this paper is NP-hard, it may be possible to guarantee properties of approximate solutions under certain conditions. This may require the design of algorithms that do not rely on coordinate descent/ascent, which does not perform well in our experience even for standard K–means.

- We saw in Section 7 the effect that correlations in the data can have. Though this behavior makes sense from the perspective of the objective, in practice it may be undesirable if there is a group of features whose correlation is obvious and uninteresting (for instance, the height and weight of patients). Decorrelating the data does not seem appropriate, since it does not preserve the semantic meaning of the original features. Non-spherical Gaussian mixture models may address this issue, but they may be problematic in a high-dimensional problem with small sample size due to the inevitably larger number of model parameters.

10 Conclusion

We analyzed two feature-sparse clustering methods from the literature and formulated a third method, greedy K–means, which (unlike the first two) uses a discrete non-convex statement of the feature-sparsity constraint, but nonetheless appears no more difficult to optimize. Our proposal may give results with fewer idiosyncrasies due to the explicit, discrete nature of the formulation.

The importance of initialization of the approximate optimization algorithms for these methods was highlighted with synthetic experiments. We demonstrated that the previously prescribed practice of initialization with the standard K–means clustering can perform arbitrarily poorly, and proposed a few alternate approaches that can give better results at the cost of multiple restarts of the algorithms. Specifically, we considered support-based initialization, which appears to perform better than the canonical initialization method for K–means based on random centroids. For relatively small dimensional cases, we proposed support covering initialization, which is a deterministic enumeration based method. For larger datasets, random support initialization performed the best, where we randomly choose an initial set of features that may be a few times larger than the desired sparsity of the solution.

We addressed the issue of clustering in the presence of mixed variable types with the formulation of a hybrid mixture model, which can be easily adapted to perform feature selection in a manner analogous to greedy K–means. For binary features, however, there appears to be only a minor difference between the K–means and hybrid criteria.

We applied greedy K–means to the lung disease phenotype dataset. The set of selected features over a range of sparsity parameter was computed, and followed a clear approximately monotonous trend. This trend largely agreed with the degree to which each feature was correlated with all other features, indicating that correlation plays a strong role in determining which features are selected The clusterings given by the near-optimal solutions found for a specific value of the sparsity parameter were highly unstable, indicating the need for careful consideration when interpreting the results.

We raised three questions in the beginning of this report. Following are the answers we were able to discern.

1. **What type of structure can each method recover, and what type of structure do they fail to recover?**

   In most synthetic experiments the true optima of all objectives seemed to recover the structure of interest (at least as far as could be verified, considering the intractability of proving a solution to be optimal), even in the case where the most sparse clustering is not the only one. However, clusters in dimensions orthogonal to the direction of predominant linear structure are unlikely to be discovered by these methods, even with infinite computational power.

2. **How do the methods differ?**

   On the high end of sparsity, the results of Regularized K–means are distorted by the type of regularization which forces cluster centroids to be near the origin. Near the other extreme, Sparcl does not smoothly transition to standard K–means, unlike Regularized and greedy K–means. Finally, greedy K–means gives exactly sparse results, whereas in the other two methods a selected feature can be arbitrarily close to irrelevant.

3. **What approximations can we hope to recover, and how do they relate to the true optima of the objectives?**

   Even in the relatively simple synthetic examples used, it took thousands of random iterations to discover the optimal solutions of interest. Moreover, the transition from optimal to sub-optimal was dramatic, meaning that the approximation quality of sub-optimal solutions was very low. This indicates that, especially for more complex, realistic data, approximation is a major issue.

   In fact, the accuracy of the results given by the algorithms we investigated was dominated by the initialization used. Hence, perhaps it is more appropriate from a practical perspective to regard these methods as somewhat sophisticated thresholding operators that find a sparse solution in some sense "near" the clustering used to initialize them, as opposed to being clustering algorithms themselves.

Acknowledgements

We thank Roy Maxion for helpful suggestions. This research is supported in part by NSF grants IIS-1116458 and CAREER award IIS-1252412.

References

[1] Laura Anderlucci and Christian Hennig. Clustering of categorical data: a comparison of a model-based and a distance-based approach. *Communications of Statistics - Theory and Methods*, 2013.

[2] Jeffrey L Andrews and Paul D McNicholas. Variable selection for clustering and classification. *ArXiv e-prints*, March 2013.

[3] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[4] Spencer Charles Brubaker and Santosh Vempala. Isotropic pca and affine-invariant clustering. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 551–560. IEEE, 2008.

[5] Stanislav Busygin, Oleg Prokopyev, and Panos M Pardalos. Biclustering in data mining. *Computers & Operations Research*, 35(9):2964–2987, 2008.

[6] Gilles Celeux and Gilda Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, 13(2):195–212, 1996.

[7] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *The Journal of Machine Learning Research*, 9:1269–1294, 2008.

[8] Chris HQ Ding and Xiaofeng He. Principal component analysis and effective k-means clustering. In *SDM*, 2004.

[9] Dan Gordon. La Jolla covering repository. `http://www.ccrwest.org/cover.html`, 2013.

[10] Daniel M Gordon, Oren Patashnik, and Greg Kuperberg. New constructions for covering designs. *Journal of Combinatorial Designs*, 3(4):269–284, 1995.

[11] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Pairwise variable selection for high-dimensional model-based clustering. *Biometrics*, 2009.

[12] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[13] Christian Hennig and Tim F Liao. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369, 2013.

[14] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[15] Hyangmin Lee and Jia Li. Variable selection for clustering by separability based on ridgelines. *Journal of Computational and Graphical Statistics*, 21(2):315–337, 2012.

[16] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(1):24–45, 2004.

[17] Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.

[18] Lukas Meier. *grplasso: Fitting user specified models with Group Lasso penalty*, 2013. R package version 0.4-3.

[19] Wendy C. Moore, Deborah A. Meyers, Sally E. Wenzel, W. Gerald Teague, Huashi Li, Xingnan Li, Ralph D'Agostino, Mario Castro, Douglas Curran-Everett, Anne M. Fitzpatrick, Benjamin Gaston, Nizar N. Jarjour, Ronald Sorkness, William J. Calhoun, Kian F. Chung, Suzy A. A. Comhair, Raed A. Dweik, Elliot Israel, Stephen P. Peters, William W. Busse, Serpil C. Erzurum, and Eugene R. Bleecker. Identification of asthma phenotypes using cluster analysis in the severe asthma research program. *American Journal of Respiratory and Critical Care Medicine*, 181(4):315–323, 2010.

[20] Wei Pan and Xiaotong Shen. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8:1145–1164, 2007.

[21] Dan Pelleg and Andrew W Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.

[22] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[23] Adrian E Raftery and Nema Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

[24] W. Sun, J. Wang, and Y. Fang. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012.

[25] Amos Tanay, Roded Sharan, and Ron Shamir. Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 9:26–1, 2005.

[26] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[27] Vincent Q. Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. In *Advances in Neural Information Processing Systems 26*, pages 2670–2678. 2013.

[28] Daniela M. Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713+, 2012.

[29] Daniela M. Witten and Robert Tibshirani. *sparcl: Perform sparse hierarchical clustering and sparse k-means clustering*, 2013. R package version 1.0.3.

[30] Wei Wu, Eugene Bleecker, Wendy Moore, William W Busse, Mario Castro, Kian Fan Chung, William J Calhoun, Serpil Erzurum, Benjamin Gaston, Elliot Israel, Douglas Curran-Everett, and Sally E Wenzel. Unsupervised phenotyping of severe asthma research program participants using expanded lung data. *Journal of Allergy and Clinical Immunology (in press)*, 2014.

[31] Benhuai Xie, Wei Pan, and Xiaotong Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2:168, 2008.

[32] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[33] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.