A Framework for Feature Selection in Clustering

Witten and Tibshirani

May 21, 2020

Introduction

- Let X denotes an $n \times p$ data matrix, with n observations and p features.
- Suppose that we wish to cluster the observations, and we suspect that the true underlying clusters differ only with respect to some of the features.
- We propose a method for sparse clustering, which allows us to group the observations using only an adaptively chosen subset of the features.

Clustering

- Kmeans
- Hierarchical clustering
- GMM, DBSCAN,manifold learning(t-SNE)

Dissimilarity

- Clustering methods require some concept of the dissimilarity(or distance) between pairs of observations.
- Throughout this paper, we will assume that d is additive in the features.That is

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} d_{i,i',j}.$$

- In this paper, we take d to be squared Euclidean distance, but other dissimilarity measures are possible.

Matrix decomposition

One way to reduce the dimensionality of the data before clustering is by performing a matrix decomposition.

- One can approximate the $n \times p$ data matrix X as $X \approx AB$ where A is a $n \times q$ matrix and $B$is a $q \times p$ matrix, $q \ll p$.
- For instance, PCA, Non-negative Matrix Factorization.

Matrix decomposition

However, these approaches have a number of drawbacks.

- The resulting clustering is not sparse in the features.
- There is no guarantee that A contains the signal that one is interested in detecting via clustering.
- For example, the principal components with largest eigenvalues do not necessarily provide the best separation between subgroups.

GMM

- One can model the rows of X as independent multivariate observations drawn from a mixture model with K components
- usually a mixture of Gaussians is used.That is, given the data, the log-likelihood is

$$\sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k f_k \left(\mathbf{X}_i; \mu_k, \mathbf{\Sigma}_k \right) \right],$$

where f_k is a Gaussian density parametrized by its mean μ_k and covariance matrix Σ_k. The EM algorithm can be used to fit his model.

feature selection in GMM

We can maximize the penalized log-likelihood

$$\sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k f_k \left(\mathbf{X}_i ; \mu_k, \mathbf{\Sigma}_k \right) \right] - \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_{kj}| ,$$

where $\Sigma_1 = \cdots = \Sigma_K$ is taken to be a diagonal matrix.

- The L_1 penalty is applied to the elements of μ_k.
- Some of the elements of μ_k will be exactly zero.
- If, for some variable j, $\mu_{kj} = 0$ for all $k = 1, \ldots, K$, then the resulting clustering will not involve feature j.

COSA

- Friedman and Meulman (2004) propose clustering objects on subsets of attributes (COSA).
- Let C_k denote the indices of the observations in the kth of the K clusters. Then, the COSA criterion is

$$\text{minimize}_{C_1,\ldots,C_K,\mathbf{w}} \left\{ \sum_{k=1}^{K} a_k \sum_{i,i' \in C_k} \sum_{j=1}^{p} (w_j d_{i,i',j} + \lambda w_j \log w_j) \right\}$$

$$\text{subject to } \sum_{i=1}^{p} w_j = 1, \quad w_j \geq 0 \quad \forall j.$$

Here, a_k is some function of the number of elements in cluster k, $w \in \mathbb{R}^p$ a vector of feature weights.

COSA

- It can be seen that this criterion is related to a weighted version of K-means clustering.
- Unfortunately, this proposal does not truly result in a sparse clustering, since all variables have nonzero weights for $\lambda > 0$.
- An extension of (3) is proposed in order to generalize the method to other types of clustering, such as hierarchical clustering.

Let $\mathbf{X}_j \in \mathbb{R}^n$ denote feature j. Many clustering methods can be expressed as an optimization problem of the form

$$\underset{\mathbf{\Theta} \in D}{\text{maximize}} \left\{ \sum_{j=1}^{p} f_j\left(\mathbf{X}_j, \mathbf{\Theta}\right) \right\}$$

where $f_j(\mathbf{X}_j, \mathbf{\Theta})$ is some function that involves only the jth feature of the data.
K-means and hierarchical clustering are two such examples, as we show in the next few sections.

We define sparse clustering as the solution to the problem

$$\begin{aligned} &\text{maximize}_{\mathbf{w},\mathbf{\Theta}\in D}\left\{\sum_{j=1}^{p} w_j f_j\left(\mathbf{X}_j,\mathbf{\Theta}\right)\right\} \\ &\text{subject to } \|\mathbf{w}\|^2 \le 1, \quad \|\mathbf{w}\|_1 \le s, \\ &w_j \ge 0 \quad \forall j, \end{aligned} \tag{5}$$

where w_j is a weight corresponding to feature j and s is a tuning parameter, $1 \le s \le \sqrt{p}$.

Optimization

We optimize (5) using an iterative algorithm:

- holding $\mathbf{w}$ fixed, we optimize (5) with respect to $\boldsymbol{\Theta}$
- holding $\boldsymbol{\Theta}$ fixed, we optimize (5) with respect to $\mathbf{w}$.

In general, we do not achieve a global optimum of (5) using this iterative approach.
However, we are guaranteed that each iteration increases the objective function.

Optimization

To optimize (5) with respect to $\mathbf{w}$ with $\mathbf{\Theta}$ held fixed, we note that the problem can be rewritten as

$$\begin{aligned} &\underset{\mathbf{w}}{\text{maximize}} \left\{ \mathbf{w}^T \mathbf{a} \right\} \\ &\text{subject to} \qquad \|\mathbf{w}\|^2 \leq 1, \quad \|\mathbf{w}\|_1 \leq s \\ &\qquad\qquad\quad w_j \geq 0 \qquad \forall j, \end{aligned} \tag{6}$$

where $a_j = f_j(\mathbf{X}_j, \mathbf{\Theta})$.

Optimization

This can be solved by KKT conditions.

K-means

K-means clustering minimizes the within-cluster sum of squares (WCSS). That is, it seeks to partition the n observations into K sets, or clusters, such that the WCSS

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

is minimal, where n_k is the number of observations in cluster k. Note that if we define the between-cluster sum of squares (BCSS) as

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

then minimizing the WCSS is equivalent to maximizing the BCSS.

Sparse K-means

One could try to develop a method for sparse K-means clustering by optimizing a weighted WCSS, subject to constraints on the weights: that is,

$$\begin{gathered}\text{maximize}_{C_1,\ldots,C_K,\mathbf{w}} \left\{ \sum_{j=1}^{p} w_j \left(- \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right) \right\} \\ \text{subject to } \|\mathbf{w}\|^2 \leq 1, \quad \|\mathbf{w}\|_1 \leq s \\ w_j \geq 0 \quad \forall j.\end{gathered}$$

Since each element of the weighted sum is negative, the maximum occurs when all weights are zero, regardless of the value of s. This is not an interesting question.

Sparse K-Means

We instead maximize a weighted BCSS, subject to constraints on the weights. Our sparse K-means clustering criterion is as follows:

$$\text{maximize}_{C_1,\ldots,C_K,\mathbf{w}} \left\{ \sum_{j=1}^{p} w_j \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right) \right\}$$

$$\text{subject to} \quad \|\mathbf{w}\|^2 \leq 1, \quad \|\mathbf{w}\|_1 \leq s$$
$$w_j \geq 0 \quad \forall j.$$

The weights will be sparse for an appropriate choice of the tuning parameter s, which should satisfy $1 \leq s \leq \sqrt{p}$.

Algorithm for Sparse K-Means clustering

- Initialize $\mathbf{w}$ as $w_1 = \cdots = w_p = 1/\sqrt{p}$.
- Iterate until convergence:
  - Holding w fixed, optimize with respect to $C_1, \ldots, C_K$.
  - Holding $C_1, \ldots, C_K$ fixed, optimize with respect to $\mathbf{w}$.
- The clusters are given by $C_1, \ldots, C_K$, and the feature weights corresponding to this clustering are given by $w_1, \ldots, w_p$.

Selection of Tuning Parameter

- The sparse K-means clustering algorithm has one tuning parameter w.
- We assume that K, the number of clusters, is fixed.
- Note that one cannot simply select s to maximize the objective function , since as s is increased, the objective will increase as well.
- Instead, we apply a permutation approach that is closely related to the gap statistic.

Algorithm to select tuning parameter s

- Obtain permuted datasets $\mathbf{X}_1, \ldots, \mathbf{X}_B$ by independently permuting the observations within each feature.
- For each candidate tuning parameter value s:
  - Compute the objective function

$$O(s) = \sum_j w_j \left(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i' \in C_k} d_{i,i',j}\right)$$

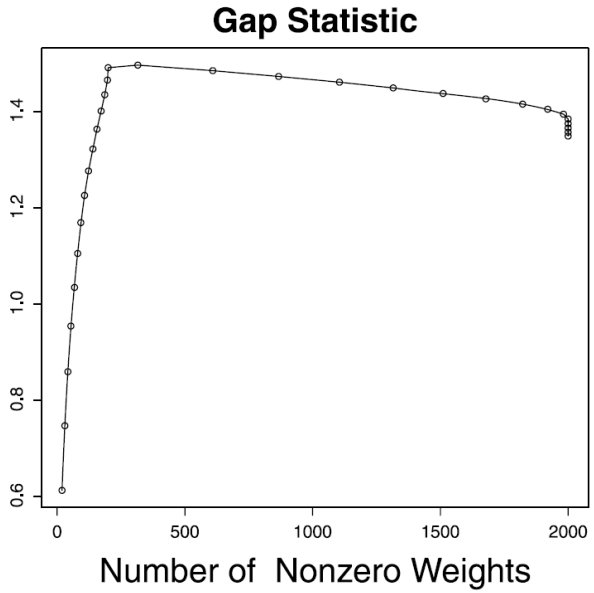
  - For $b = 1, 2, \ldots, B$, compute $O_b(s)$.
  - Calculate $Gap(s) = \log(O(s)) - \frac{1}{B}\sum_{b=1}^{B}\log(O_B(s))$.
- Choose s^* corresponding to the largest value of $Gap(s)$.

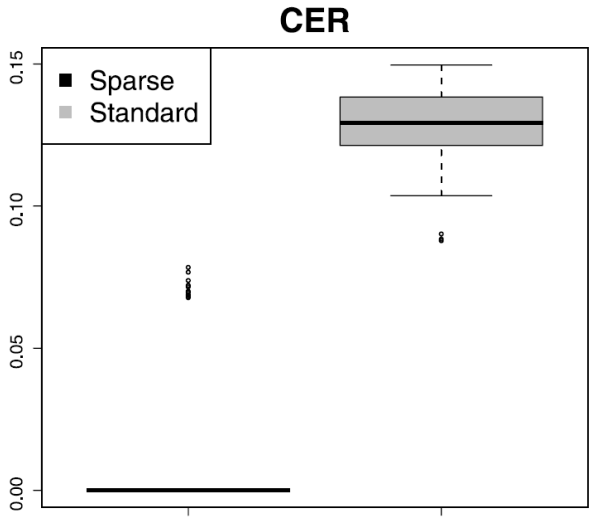
Tuning parameter selection

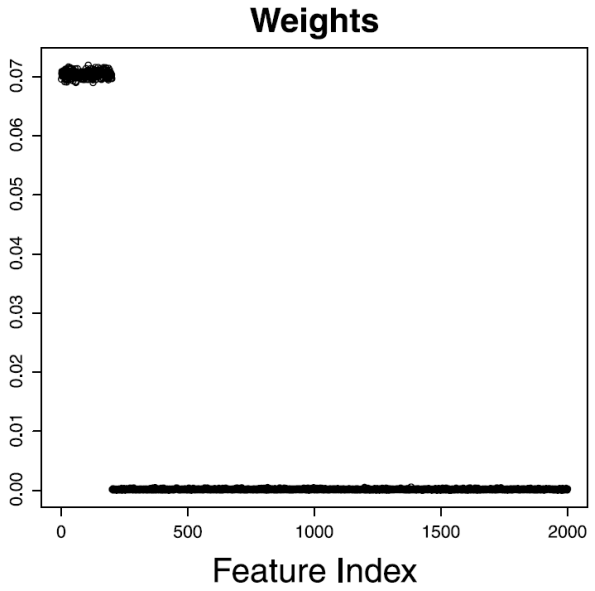
- Note that while there may be strong correlations between the features in the original data $\mathbf{X}$, the features in the permuted datasets $\mathbf{X}_1, ..., \mathbf{X}_B$ are uncorrelated with each other.
- The gap statistic measures the strength of the clustering obtained on the real data relative to the clustering obtained on null data that does not contain subgroups.
- The optimal tuning parameter value occurs when this quantity is greatest.

Tuning parameter selection

We apply this method to a simple example with 6 equally sized classes, where $n = 120, p = 2000, and 200$ features differ between classes.
In the figure we have used the classification error rate (CER) for two partitions of a set of n observations.







A Comparison of Sparse and Standard K-Means

- We compare the performances of standard and sparse K-means in a simulation study where $q = 50$ features differ between $K = 3$ classes.
- $X_{ij} \sim N(\mu_{ij}, 1)$ independent; $\mu_{ij} = \mu\,(1_{i \in C_1, j \leq q} - 1_{i \in C_2, j \leq q})$.
- Datasets were generated with various values of μ and p, with 20 observations per class.

A Comparison of Sparse and Standard K-Means

Table 1. Standard 3-means results for Simulation 1. The reported values are the mean (and standard error) of the CER over 20 simulations. The μ/p combinations for which the CER of standard 3-means is significantly less than that of sparse 3-means (at level $\alpha = 0.05$) are shown in bold

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	**0.07 (0.01)**	0.184 (0.015)	0.22 (0.009)	0.272 (0.006)
$\mu = 0.7$	**0.023 (0.005)**	0.077 (0.009)	0.16 (0.012)	0.232 (0.01)
$\mu = 0.8$	**0.013 (0.004)**	0.038 (0.007)	0.08 (0.005)	0.198 (0.01)
$\mu = 0.9$	**0.001 (0.001)**	0.013 (0.005)	0.048 (0.008)	0.102 (0.013)
$\mu = 1$	0.002 (0.002)	0.004 (0.002)	0.013 (0.004)	0.05 (0.006)

A Comparison of Sparse and Standard K-Means

Table 2. Sparse 3-means results for Simulation 1. The reported values are the mean (and standard error) of the CER over 20 simulations. The μ/p combinations for which the CER of sparse 3-means is significantly less than that of standard 3-means (at level $\alpha = 0.05$) are shown in bold

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	0.146 (0.014)	**0.157 (0.016)**	**0.183 (0.015)**	0.241 (0.017)
$\mu = 0.7$	0.081 (0.011)	**0.049 (0.008)**	**0.078 (0.013)**	**0.098 (0.013)**
$\mu = 0.8$	0.043 (0.008)	**0.031 (0.007)**	**0.031 (0.005)**	**0.037 (0.006)**
$\mu = 0.9$	0.015 (0.006)	**0.005 (0.003)**	**0.014 (0.004)**	**0.014 (0.004)**
$\mu = 1$	0.009 (0.004)	0.004 (0.002)	**0.001 (0.001)**	**0.002 (0.002)**

A Comparison of Sparse and Standard K-Means

Table 3. Sparse 3-means results for Simulation 1. The mean number of nonzero feature weights resulting from the method for tuning parameter selection of Section 3.2 is shown; standard errors are given in parentheses. Note that 50 features differ between the three classes

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	41.35 (0.895)	167.4 (7.147)	243.1 (31.726)	119.45 (41.259)
$\mu = 0.7$	40.85 (0.642)	195.65 (2.514)	208.85 (19.995)	130.15 (17.007)
$\mu = 0.8$	38.2 (0.651)	198.85 (0.654)	156.35 (13.491)	106.7 (10.988)
$\mu = 0.9$	38.7 (0.719)	200 (0)	204.75 (19.96)	83.7 (9.271)
$\mu = 1$	36.95 (0.478)	200 (0)	222.85 (20.247)	91.65 (14.573)

A Comparison With Other Approaches

We compare the performance of sparse K-means to a number of competitors:

- COSA
- The model-based clustering approach of Raftery and Dean(2006)
- The penalized log-likelihood approach of Pan and Shen (2007)
- PCA followed by 3-means clustering.

A Comparison With Other Approaches

Table 4. Results for Simulation 2. The quantities reported are the mean and standard error (given in parentheses) of the CER, and of the number of nonzero coefficients, over 25 simulated datasets

Simulation	Method	CER	Num. nonzero coef.
Small simulation:	Sparse K-means	0.112 (0.019)	8.2 (0.733)
$p = 25, q = 5$,	K-means	0.263 (0.011)	25 (0)
10 obs. per class	Pan and Shen	0.126 (0.017)	6.72 (0.334)
	COSA w/Hier. Clust.	0.381 (0.016)	25 (0)
	COSA w/K-medoids	0.369 (0.012)	25 (0)
	Raftery and Dean	0.514 (0.031)	22 (0.86)
	PCA w/K-means	0.16 (0.012)	25 (0)
Large simulation:	Sparse K-means	0.106 (0.019)	141.92 (9.561)
$p = 500, q = 50$,	K-means	0.214 (0.011)	500 (0)
20 obs. per class	Pan and Shen	0.134 (0.013)	76 (3.821)
	COSA w/Hier. Clust.	0.458 (0.011)	500 (0)
	COSA w/K-medoids	0.427 (0.004)	500 (0)
	PCA w/K-means	0.058 (0.006)	500 (0)

The Sparse Hierarchical Clustering Method

- Note that hierarchical clustering takes as input a $n \times n$ dissimilarity matrix U.
- The clustering can use any type of linkage— complete, average, or single.
- If U s the overall dissimilarity matrix $\{\sum_j d_{i,i',j}\}_{i,i'}$,then standard hierarchical clustering results.
- In this section, we cast the overall dissimilarity matrix $\{\sum_j d_{i,i',j}\}_{i,i'}$ in the form (4), and then propose a criterion of the form (5) that leads to a reweighted dissimilarity matrix that is sparse in the features.

The Sparse Hierarchical Clustering Method

Since scaling the dissimilarity matrix by a factor does not affect the shape of the resulting dendrogram, we ignore proportionality constants in the following discussion. Consider the criterion

$$\begin{aligned} &\underset{\mathbf{U}}{\text{maximize}} \left\{ \sum_j \sum_{i,i'} d_{i,i',j} U_{i,i'} \right\} \\ &\text{subject to} \quad \sum U_{i,i'}^2 \leq 1. \end{aligned} \tag{14}$$

Let U^* optimize (14). It is not hard to show that $U^*_{i,i'} \propto \sum_j d_{i,i',j}$, and so performing hierarchical clustering on U^* results in standard hierarchical clustering.

The Sparse Hierarchical Clustering Method

we modify (14) by multiplying each element of the summation over j by a weight wj, subject to constraints on the weights:

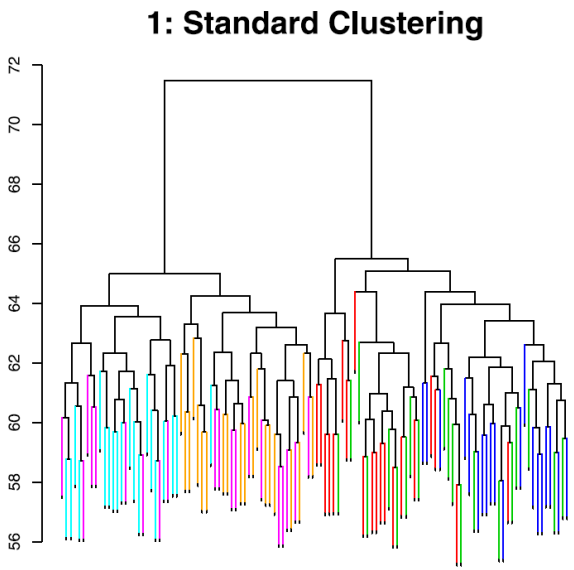
$$\underset{\mathbf{w},\mathbf{U}}{\text{maximize}} \left\{ \sum_j w_j \sum_{i,i'} d_{i,i',j} U_{i,i'} \right\}$$

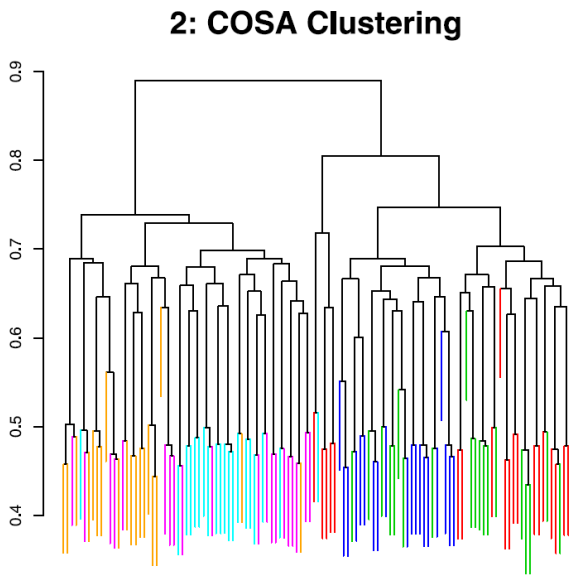
$$\text{subject to} \quad \begin{array}{l} \sum_{i,i'} U_{i,i'}^2 \leq 1, \quad \|\mathbf{w}\|^2 \leq 1, \quad \|\mathbf{w}\|_1 \leq s \\ w_j \geq 0 \quad \forall j \end{array} \tag{15}$$

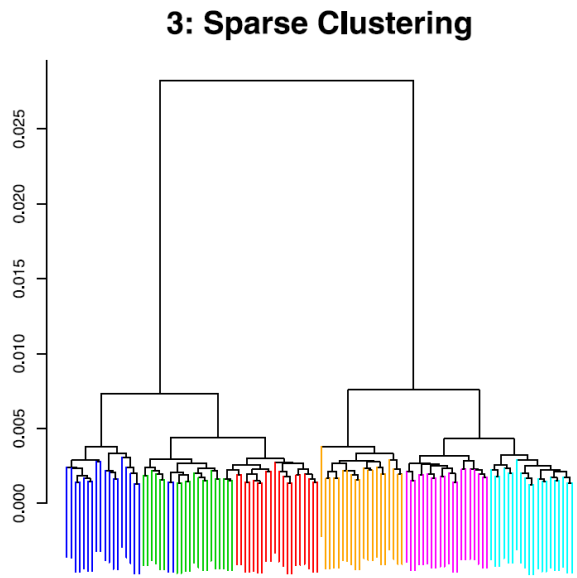
Algorithm for sparse hierarchical clustering

- Initialize w as $w_1 = \cdots = w_p = 1/\sqrt{p}$
- Iterate until convergence:
  - update u
  - update w
- Rewrite u as a $n \times n$ matrix U.
- Perform hierarchical clustering on the $n \times n$ dissimilarity matrix U.

Sparse Hierarchical clustering







Reanalysis of a breast cancer dataset

We performed four versions of hierarchical clustering with Eisen linkage on the 62 observations that were assigned to the four classes:

- Sparse hierarchical clustering of all 1753 genes, with the tuning parameter chosen to yield 496 nonzero genes
- Standard hierarchical clustering using all 1753 genes.
- Standard hierarchical clustering using the 496 genes with highest marginal variance.
- COSA hierarchical clustering using all 1753 genes.

Reanalysis of a breast cancer dataset