A Sparse K-Means Clustering Algorithm

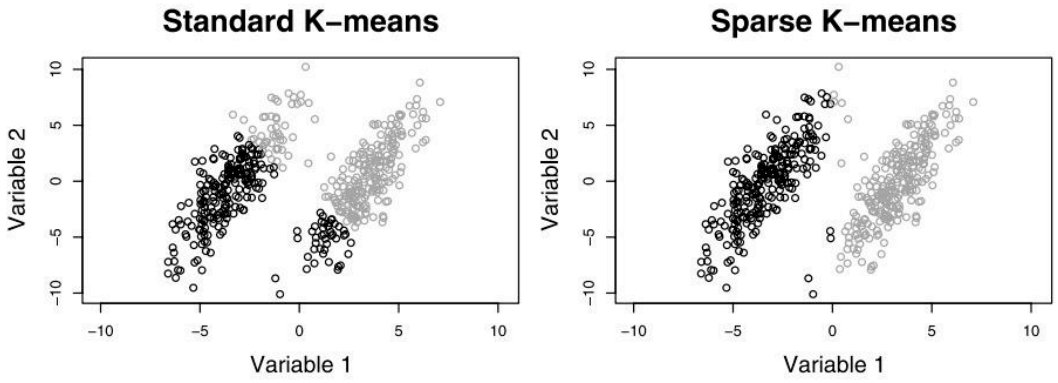
[redacted]

K-means is a broadly used clustering method which aims to partition *n* observations into *k* clusters, in which each observation belongs to the cluster with the nearest mean. The popularity of K-means derives in part from its conceptual simplicity (it optimizes a very natural objective function) and widespread implementation in statistical packages.

Unfortunately, it is easy to construct simple examples where K-means performs rather poorly in the presence of a large number of noise variables, i.e., variables that do not change from cluster to cluster. These types of datasets are commonplace in modern applications. Furthermore, in some applications it is of interest to identify not only possible clusters in the data, but also a relatively small number of variables that sufficiently determine that partition.

To address these problems, Witten and Tibshirani (2010) proposed an alternative to the classical K-means - called sparse K-means (SK-means) - which simultaneously finds the clusters and the important clustering variables.

As a motivating example, the following are two clustering results of 500 independent observations from a bivariate normal distribution. A mean shift on the first feature defines the two classes. The resulting data, as well as the clusters obtained using standard k-means clustering (k=2) and the sparse k-means clustering, can be seen below. Unlike standard k-means clustering, sparse k-means clustering automatically identifies a subset of the features to use in clustering the observations. Here it uses only the first feature, and consequently agrees quite well with the true class labels.



Let **X** denote an $n \times p$ data matrix, with n observations and p features. One way to reduce the dimensionality of the data before clustering is by performing PCA in order to obtain a matrix **A** of reduced dimensionality; then, the n rows of **A** can be clustered. However, this approach has a number of drawbacks. First of all, the resulting clustering is not sparse in the features, since each of the columns of **A** is a function of the full set of p features. Moreover, there is no guarantee that the new features in **A** providesthe best separation between subgroups.

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

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

is minimal, where n_k is the number of observations in cluster k and C_k contains the indices of the observations in cluster k. This is equivalent to:

$$WCSS = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \mu_{kj})^2$$

where μ_{kj} is the mean of feature j for all the elements in cluster k.

Note that if we define the *between-cluster sum of squares* (BCSS) as

$$BCSS = \sum_{j=1}^{p} \left(\sum_{i=1}^{n} (x_{ij} - \mu_j)^2 - \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2 \right)$$

where μ_j is the mean of feature j for all the elements in the dataset, then minimizing the WCSS is equivalent to maximizing the BCSS.

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

$$\max_{C_1,\ldots,C_k,w} \left\{ \sum_{j=1}^{p} w_j \left(-\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} (x_{ij} - \mu_j)^2 \right) \right\}$$

$$subject\ to\ \|w\|^2 \le 1, \|w\|_1 \le s,\ w_j \ge 0\ \forall j$$

where w_j is a weight corresponding to feature j and s is a tuning parameter. 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 solution. We instead maximize a weighted BCSS, subject to constraints on the weights. The *sparse K-means clustering criterion* is as follows:

$$\max_{C_1,\ldots,C_k,w} \{ \sum_{j=1}^{p} w_j (\sum_{i=1}^{n} (x_{ij} - \mu_j)^2 - \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2 \}$$

$$subj\,ect\,to \; \|w\|^2 \le 1, \|w\|_1 \le s, \; w_j \ge 0 \; \forall j$$

Some observations about this criterion:

1. If $w_1 = w_2 = \ldots = w_p$, then the criterion reduces to minimizing the WCSS, which can be solved by the standard K-means clustering algorithm.
2. The L1, or *lasso*, penalty on *w* results in sparsity for small values of the tuning parameter *s*. That is, some of the w_j's will equal zero.
3. The L2 penalty also serves an important role, since without it, at most one element of *w* would be non-zero in general.
4. The value of w_j can be interpreted as the contribution of feature *j* to the resulting sparse clustering: a large value of w_j indicates a feature that contributes greatly, and $w_j = 0$ means that feature *j* is not involved in the clustering.

The sparse K-means clustering maximizes the objective function by carrying out the following steps:

1) Initialize w as $w_1 = w_2 = \ldots = w_p = \frac{1}{\sqrt{p}}$

2) Iterate until the weight changes converge to 0 (see the stopping criterion on the next page)

    a. Holding w fixed, optimize the criterion with respect to $C_1,\ldots,C_k$. That is:

$$\min_{C_1,\ldots,C_k} \{ \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} w_j (x_{ij} - \mu_{kj})^2 \}$$

    by applying the standard K-means.

b. Holding $C_1,...,C_k$ fixed, optimize the criterion with respect to w by applying:

$$\max_{w} \{ \sum_{j=1}^{p} w_j a_j \}$$

$$subj\,ect\,to\ \|w\|^2 \le 1, \|w\|_1 \le s,\ w_j \ge 0\ \forall j$$

$$where\ a_j = \sum_{i=1}^{n} (x_{ij} - \mu_j)^2 - \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2 \qquad (f\,ixed)$$

Since in the dataset of the experiment have certain special characteristics, the general solution for this convex problem is not presented here. The simplified solution will be given in the experiment section.

3) The clusters are given by $C_1,\ldots,C_K$, and the feature weights corresponding to this clustering are given by $w_1,\ldots,w_p$.

In the stopping criterion for step 2, we stop when the sum of changes in weights is small in relation to the weights:

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

where r is the number of the current iteration.

The Experiment

To evaluate the performance of the Sparse K-means in relation to the classic K-means, I ran text classification experiments on the 20 newsgroups data set. 20 Newsgroups (as the title suggests) is a collection of newsgroup posts from 20 different newsgroups from the mid 1990s. There are approximately 1000 posts per newsgroup. I used the "20news-bydate" version, which has duplicates removed, posts are sorted within each newsgroup by date into train/test sets and newsgroup-identifying headers are discarded. My pre-processing consisted of (in order) (1) splitting on space characters, (2) lower-casing all alphabetic characters, (3) discarding tokens of length 25 or greater (required for the removal of binary code), (4) stemming each token (excluding stop words). I

computed a document vector for each post, consisting of the number of times each token occurred in that post. After running this pre-processing on the collection, 2,216,913 occurrences of 79970 distinct terms were found in 18832 documents.

In addition, at the end of the pre-processing, all the distinct terms were sorted in accordance to the variance of their frequency in a post across the dataset. It is possible choose the *t* terms with the highest frequency variance. This follows the concept that words with high frequency suggest a better separation of posts into groups.

The general solution for the convex problem in step 2b of the Sparse K-means clustering algorithm involves the soft-thresholding of the BCSS for each token. This proposition follows from the Karush-Kuhn-Tucker conditions (see e.g. Boyd & Vandenberghe 2004). However, since the number of occurrences of a token in a post is non-negative, optimizing weights can be found by applying the following simplified solution:

$$\mathrm{w} = \frac{\mathrm{S}(\mathrm{a}, \Delta)}{\|\mathrm{S}(\mathrm{a}, \Delta)\|} \quad \text{where a} \in R^{\mathrm{p}} \text{ such that}$$

$$a_j = \sum_{i=1}^{n} (x_{ij} - \mu_j)^2 - \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2$$

$$\Delta = 0 \text{ if that results in } \| \mathrm{w} \|_1 < \mathrm{s}\text{; otherwise, } \Delta > 0 \text{ is chosen so that } \| \mathrm{w} \|_1 = \mathrm{s}$$

$$\text{and } S(a, \Delta) \in R^{\mathrm{p}}\text{, } S(a, \Delta)[j] = \max(a_j - \Delta, 0)$$

While in the Sparse K-means algorithm, the weights are re-computed in each iteration, the weights in the Standard K-means algorithm are fixed. Since some words are generally more common than others and therefore are not good keywords for distinguishing between groups of posts, an inverse document frequency factor (IDF) was incorporated, which diminishes the weight of terms that occur very frequently across the dataset.

$$idf(term) = \log \frac{\#\text{ of posts in the dataset}}{\#\text{ of posts in the dataset that contain } term}$$

Therefore, for each document a normalized tf-idf vector (term count inverse document frequency) was computed.

Results

Performance was measured using CER (Classification Error Rate), that is, the number of misclassified posts in relation to the total number of posts. A lower CER indicates a better classification algorithm. The newsgroup of a cluster is set to be the newsgroup with the highest number of posts belonging to that cluster.

The following are the CER results for running K-means algorithm versions on the whole dataset (20 newsgroups):

Number of terms used	Standard K-means	Sparse K-means
100	0.8538	0.8505
1000	0.6686	0.6602
10000	0.6422	0.6313

The following are the CER results for running K-means algorithm versions on different number of newsgroups using 10000 terms with the highest frequency variance across the dataset (newsgroups were chosen randomly):

Number of newsgroups	Standard K-means	Sparse K-means
2	0.2676	0.2676
5	0.3932	0.3867
10	0.4804	0.4796
20	0.6422	0. 6313

References

Boyd, S.; Vandenberghe, L. Convex Optimization. Cambridge University Press; 2004.

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