

**CSE 494 CSE/CBS 598 (Fall 2007): Numerical Linear Algebra for Data  
Exploration— Clustering**  
Instructor: Jieping Ye

## 1 Introduction

- One important method for data compression and classification is to organize data points in clusters: A cluster is a subset of the set of data points that are close together, using some distance measure.
- A loose definition of clustering could be the process of organizing data into groups whose members are similar in some way. A cluster is therefore a collection of data points which are “similar” between them and are “dissimilar” to the data points belonging to other clusters
- One can compute the mean value of each cluster separately, and use the means as representatives of the clusters. Equivalently, the means can be used as basis vectors, and all the data points are represented by their coordinates with respect to this basis.
- Clustering algorithms can be applied in many fields:
  - Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
  - Biology: classification of plants and animals given their features;
  - WWW: document clustering; clustering weblog data to discover groups of similar access patterns.
- There are several methods for computing a clustering. One of the most important is the k-means algorithm.

## 2 K-means Clustering

- We assume that we have  $n$  data points  $\{x_i\}_{i=1}^n \in \mathbb{R}^m$ , which we organize as columns in a matrix

$$X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}.$$

- Let  $\Pi = \{\pi_j\}_{j=1}^k$  denote a partitioning of the data in  $X$  into  $k$  clusters:

$$\pi_j = \{v \mid x_v \text{ belongs to cluster } j\}.$$

- Let the mean, or the centroid, of the cluster be

$$c_j = \frac{1}{n_j} \sum_{v \in \pi_j} x_v,$$

where  $n_j$  is the number of elements in  $\pi_j$ .

- We describe K-means algorithm based on the Euclidean distance measure.

- The tightness or coherence of cluster  $\pi_j$  can be measured as the sum

$$q_j = \sum_{v \in \pi_j} \|x_v - c_j\|^2.$$

- The closer the vectors are to the centroid, the smaller the value of  $q_j$ . The quality of a clustering can be measured as the overall coherence,

$$Q(\Pi) = \sum_{j=1}^k \sum_{v \in \pi_j} \|x_v - c_j\|^2.$$

- In the k-means algorithm we seek a partitioning that has optimal coherence, in the sense that it is the solution of the minimization problem  $\min_{\Pi} Q(\Pi)$ .
- The K-means algorithm
  - Initialization: Choose  $k$  initial centroids.
  - Form  $k$  clusters by assigning all data points to the closest centroid.
  - Recompute the centroid of each cluster.
  - Repeat the second and third steps until convergence.
- The initial partitioning is often chosen randomly. The algorithms usually has rather fast convergence, but one cannot guarantee that the algorithm finds the global minimum.

### 3 Spectral Relaxation for K-means Clustering

- Despite the popularity of K-means clustering, one of its major drawbacks is that it is prone to local minima. Much research has been done on computing refined initial points and adding explicit constraints to the sum-of-squares cost function for K-means clustering so that the search can converge to better local minimum.
- Zha *et al.* tackled the problem from a different angle: formulate the sum-of-squares minimization in K-means as a trace maximization problem with special constraints: relaxing the constraints leads to a maximization problem that possesses optimal global solutions.

– *Spectral Relaxation for K-means Clustering.* H. Zha, X. He, D. Ding, and H. Simon. NIPS 2001.

#### 3.1 Spectral Relaxation

- Recall that the  $n$  data points  $\{x_i\}_{i=1}^n \in \mathbb{R}^m$ , which we organize as columns in a matrix

$$X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}.$$

are partitioned into  $k$  clusters:  $\Pi = \{\pi_j\}_{j=1}^k$  as

$$\pi_j = \{v \mid x_v \text{ belongs to cluster } j\}.$$

The mean, or the centroid, of the cluster is

$$c_j = \frac{1}{n_j} \sum_{v \in \pi_j} x_v,$$

where  $n_j$  is the number of elements in  $\pi_j$ .

- For a given partition  $\Pi$ , the associated sum-of-squares cost function is defined as

$$Q(\Pi) = \sum_{j=1}^k \sum_{v \in \pi_j} \|x_v - c_j\|^2.$$

- Let  $e$  be the vector of all ones with appropriate length. It is easy to see that  $c_j = X_j e / n_j$ , where  $X_j$  is the data matrix of the  $j$ -th cluster.
- The sum-of-squares cost function of the  $j$ -th cluster is

$$q_j = \sum_{v \in \pi_j} \|x_v - c_j\|^2 = \|X_j - c_j e^T\|_F^2 = \|X_j(I_{n_j} - ee^T/n_j)\|_F^2.$$

- Note that  $I_{n_j} - ee^T/n_j$  is a projection matrix and

$$(I_{n_j} - ee^T/n_j)^2 = I_{n_j} - ee^T/n_j.$$

It follows that

$$q_j = \text{trace} \left( X_j (I_{n_j} - ee^T/n_j) X_j^T \right) = \text{trace} \left( (I_{n_j} - ee^T/n_j) X_j^T X_j \right).$$

Therefore,

$$Q(\Pi) = \sum_{j=1}^k q_j = \sum_{j=1}^k \left( \text{trace} \left( X_j^T X_j \right) - \frac{e^T X_j^T X_j e}{\sqrt{n_j}} \right).$$

- Define the  $n$ -by- $k$  orthogonal matrix  $Y$  as follows

$$Y = \begin{pmatrix} e/\sqrt{n_1} & & & \\ & e/\sqrt{n_2} & & \\ & & \ddots & \\ & & & e/\sqrt{n_k} \end{pmatrix} \quad (1)$$

Then

$$Q(\Pi) = \text{trace} \left( X^T X \right) - \text{trace} \left( Y^T X^T X Y \right).$$

The minimization of  $Q(\Pi)$  is equivalent to the maximization of  $\text{trace} \left( Y^T X^T X Y \right)$  with  $Y$  is of the form in Eq. (1).

- Ignoring the special structure of  $Y$  and let it be an arbitrary orthonormal matrix, we obtain a relaxed maximization problem

$$\max_{Y^T Y = I_k} \text{trace} \left( Y^T X^T X Y \right).$$

- It turns out the above trace maximization problem has a closed-form solution.
  - Theorem (Ky Fan): Let  $H$  be a symmetric matrix with eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  and the corresponding eigenvectors  $U = [u_1, \dots, u_n]$ . Then

$$\lambda_1 + \dots + \lambda_k = \max_{Y^T Y = I_k} \text{trace}(Y^T H Y).$$

Moreover, the optimal  $Y^*$  is given by  $Y^* = [u_1, \dots, u_k]Q$  with  $Q$  an arbitrary orthogonal matrix of size  $k$  by  $k$ .

- We may derive the following lower bound for the minimum of the sum-of-squares cost function:

$$\min_{\Pi} Q(\Pi) \geq \text{trace}(X^T X) - \max_{Y^T Y = I_k} \text{trace}(Y^T X^T X Y) = \sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2(X),$$

where  $\sigma_i(X)$  is the  $i$ -th largest singular value of  $X$ .

- It is easy to see from the above derivation that we can replace  $X$  with  $X - ae^T$  where  $a$  is an arbitrary vector. If we choose  $a$  to be the mean of all data in  $X$ , then relaxed maximization problem is equivalent to Principal Component Analysis (PCA).
- Let  $Y^*$  be the  $n$ -by- $k$  matrix consisting of the  $k$  largest eigenvectors of  $X^T X$ . Each row of  $Y^*$  corresponds to a data vector. This can be considered as transforming the original data vectors which live in a  $m$ -dimensional space to new data vectors which now live in a  $k$ -dimensional space. One might be attempted to compute the cluster assignment by applying the ordinary K-means method to those data vectors in the reduced dimension space.
- More references
  - *K-means Clustering via Principal Component Analysis*. Chris Ding and Xiaofeng He. *ICML 2004*.
  - *A Unified View of Kernel k-means, Spectral Clustering and Graph Partitioning*. I.S. Dhillon, Y. Guan, and B. Kulis. *UTCS Technical Report #TR-04-25*.  
[http://www.cs.utexas.edu/users/kulis/pubs/spectral\\_techreport.pdf](http://www.cs.utexas.edu/users/kulis/pubs/spectral_techreport.pdf)

## 4 Matrix Approximations using Clustering

- Given any partitioning  $\Pi = \{\pi_j\}_{j=1}^k$  of the data in  $X$  into  $k$  clusters, we can approximate a document vector by the closest mean (centroid) vector. In other words, if a document vector is in cluster  $\pi_j$ , we can approximate it by the mean vector  $c_j$ .
- This leads to the matrix approximation  $X \approx \hat{X}$  such that, for  $1 \leq i \leq n$ , its  $i$ -th column is the mean vector closest to the data point  $x_i$ .
- We can express  $\hat{X}$  as low-rank matrix approximation as follows:

$$\hat{X} = [c_1, c_2, \dots, c_k] \mathcal{I}$$

where  $\mathcal{I} \in \mathbb{R}^{k \times n}$  indicates the cluster membership. More specifically,  $\mathcal{I}_{ij} = 1$ , if  $x_j$  belongs to the  $i$ -th cluster  $\pi_i$  and  $\mathcal{I}_{ij} = 0$  otherwise. Denote  $C = [c_1, c_2, \dots, c_k]$ .

- The matrix approximation  $\hat{X}$  has rank at most  $k$ . It is thus natural to compare the approximation power of  $\hat{X}$  to that of the best possible rank- $k$  approximation to the data matrix  $X$  based on SVD.
  - Best rank- $k$  approximation:  $X_k = U_k \Sigma_k V_k^T$ , where  $U_k$  and  $V_k$  consist of the top  $k$  left and right singular vectors of  $X$ , respectively, and  $\Sigma_k$  contains the top  $k$  singular values.
- Empirical studies showed that, for each fixed  $k$ , the approximation error for the  $k$ -truncated SVD is significantly lower than that for  $\hat{X}$ .

## 5 Concept Decompositions

- It can be shown that by approximating each document vector by a linear combination of the concept vectors it is possible to obtain significantly better matrix approximations.

– *Concept Decompositions for Large Sparse Text Data using Clustering. I.S. Dhillon and D.S. Modha. Machine Learning, 2001.*

- Given any partitioning  $\Pi = \{\pi_j\}_{j=1}^k$  of the data in  $X$  into  $k$  clusters. Let  $\{c_j\}_{j=1}^k$  denote the  $k$  cluster centroids. Define the concept matrix as a  $m \times k$  matrix such that, for  $1 \leq j \leq k$ , the  $j$ -th column of the matrix is the centroid (concept) vector  $c_j$ , that is,  $C = [c_1, c_2, \dots, c_k]$ . Assuming linear independence of the  $k$  concept vectors, it follows that the concept matrix has rank  $k$ .

- For any partitioning of the data, we define the corresponding concept decomposition  $\tilde{X}_k$  of the data matrix  $X$  as the least-squares approximation of  $X$  onto the column space of the concept matrix  $C$ . We can write the concept decomposition as an  $m \times n$  matrix  $\tilde{X}_k = CZ^*$ ; where  $Z^*$  is a  $k \times n$  matrix that is to be determined by solving the following least-squares problem:

$$Z^* = \arg \min_Z \|X - CZ\|_F^2.$$

- It is well known that a closed-form solution exists for the least-squares problem above, namely,

$$Z^* = (C^T C)^{-1} C^T X.$$

- Although the above equation is intuitively pleasing, it does not constitute an efficient and numerically stable way to compute the matrix  $Z^*$ . Instead, we can use the QR decomposition of the concept matrix.

– Let  $C = QR$  be the thin QR decomposition of  $C$ , then  $Z^* = (R^T R)^{-1} R^T Q^T X$ . It follows that

$$\tilde{X}_k = CZ^* = QR (R^T R)^{-1} R^T Q^T X = QQ^T X.$$

Here we assume that  $R$  is nonsingular, i.e., the  $k$  cluster centroids in  $C$  are linearly independent.

- Show that the concept decomposition  $\tilde{X}_k$  is a better matrix approximation than  $\hat{X}_k$ .

### 5.1 Empirical Observations

- The approximation power (when measured using the Frobenius norm) of concept decompositions is comparable to the best possible approximations by truncated SVD. An important advantage of concept decompositions is that they are computationally more efficient and require much less memory than truncated SVD.
- When applied for document clustering, concept decompositions produce concept vectors which are localized in the word space, are sparse, and tend towards orthonormality. In contrast, the singular vectors obtained from SVD are global in the word space and are dense. Nonetheless, the subspaces spanned by the concept vectors and the leading singular vectors are quite close in the sense of small principal angles between them.