COSI 126A BONUS QUESTIONS

Jie Tang

APRIL 26, 2020

1. Implement K-means in 10 lines (5 points)

Attached in the codes files

2. Implement comprehensive K-means including initialization, multiple runs, outliers without line number constraint (10 points)

Attached in the codes files

3. Proof of the equivalence of element-wise and matrix-wise K-means objective function in Chapter 4 Page 4 (5 points)

Let the data $X = \langle x_1, x_2, \vec{x_3}, ..., x_n \rangle^\top$, x_i is a row vector representing the i-th point

For any clustering result, the centroid is $M = \langle m_1, m_2, \vec{m_3}, ..., m_n \rangle^\top$,

Let $H_{lk} = 1$ if $X_l \in C_l$, 0 otherwise,

Notice that the l-th row of H is a k-dimension row vector $\vec{h_l} = \langle 0, 0, ..., 0, 1, 0, ..., 0 \rangle$, which only have only non-zero element equal to 1 in the k-th place, k is the cluster label of x_l.

The l-th row of HM is $\vec{h_l} * M = \langle 0, 0, ..., 0, \vec{1}, 0, ..., 0 \rangle \langle m_1, m_2, \vec{m_3}, ..., m_n \rangle^\top = m_k$

Let $m_l^{'} = m_k$, for $x_l \in C_k$. $m_l^{'}$ is the closed certroid of x_l.

$HM = \left\langle m_1^{'}, m_2^{'}, m_3^{'}, ..., m_n^{'} \right\rangle^\top$ and $X - HM = \left\langle x_1 - < m_1^{'}, x_2 - m_2^{'}, x_3 - m_3^{'}, ..., x_n - m_n^{'} \right\rangle^\top$

$||X - HM||_F^2 = \sum_{l=1}^{N} ||x_l - m_l^{'}||^2$, and by definition, $m_l^{'}$ is the certroid for x_l,

$\sum_{l=1}^{N} ||x_l - m_l^{'}||^2$ can be written as $\sum_{k=1}^{K} \sum_{x_l \in C_k} ||x_i - m_k||^2$

So element-wise and matrix-wise K-means objective function are equivalent.

4. Proof of the graph interpretation of K-means in Chapter 4 Page 5 (5 points)

Let the data $X = \langle x_1, x_2, \vec{x_3}, ..., x_n \rangle^\top$, x_i is a row vector representing the i-th point. For a clustering with K clusters, we rearrange the rows, grouping points in the same row together. $X = [X_i, X_2, ..., X_K]^\top$, $X_k = [x_1^{(k)}, ..., x_{n_i}^{(k)}]^\top$, $n_k = |C_k|$, and SSE remain unchanged.

Then SSE can be written as $\sum_{k=1}^{K} \sum_{i=1}^{n_k} ||x_i^k - m_k||^2$, m_k is a row vector representing the k-th centroid

For the k-th cluster, let $e_k = \langle 1, 1, \vec{...}, 1 \rangle$, the dimension of e_k is n_k, $m_k = \frac{\sum_{i=1}^{n_k} x_i^k}{n_k}$

$SSE_k = \sum_{i=1}^{n_k} ||x_i^k - m_k||^2 = ||X_k - m_k^\top e_k||_F^2 = ||(I_{n_k} - e_k^\top e_k / n_k) X_k||_F^2$

$= trace(((I_{n_k} - e_k^\top e_k/n_k)X_k)^\top ((I_{n_k} - e_k^\top e_k/n_k)X_k)$

$= trace((X_k^\top (I_{n_k} - e_k^\top e_k/n_k)^\top)(I_{n_k} - e_k^\top e_k/n_k)X_k)$

Notice that $I_{n_k} - e_k^\top e_k/n_k$ is symmetric and a projection matrix,

$(I_{n_k} - e_k^\top e_k/n_k)^\top (I_{n_k} - e_k^\top e_k/n_k) = (I_{n_k} - e_k^\top e_k/n_k)^2 = (I_{n_k} - e_k^\top e_k/n_k)$

Then $SSE_k = trace((X_k^\top (I_{n_k} - e_k^\top e_k/n_k)X_k) = trace((I_{n_k} - e_k^\top e_k/n_k)X_k X_k^\top) =$

$trace(X_k X_k^\top - (e_k^\top e_k/n_k)X_k X_k^\top) = trace(X_k X_k^\top) - trace((\frac{e_k^\top}{\sqrt{n_k}})X_k X_k^\top (\frac{e_k}{\sqrt{n_k}})^\top)$

So $SSE = \sum_{k=1}^{K} SSE_i = \sum_{k=1}^{K} [trace(X_k X_k^\top) - trace((\frac{e_k^\top}{\sqrt{n_k}})X_k X_k^\top (\frac{e_k}{\sqrt{n_k}}))]$

Now we can define a N*K matrix,

$$Q = \begin{bmatrix} \frac{e_1^\top}{\sqrt{n_1}} & & \\ & \ddots & \\ & & \frac{e_K^\top}{\sqrt{n_K}} \end{bmatrix}$$

$SSE = trace(XX^\top) - \sum_{k=1}^{K} trace((\frac{e_k^\top}{\sqrt{n_k}})X_k X_k^\top (\frac{e_k}{\sqrt{n_k}}))] = trace(XX^\top) - trace(Q^\top XX^\top Q)$.

In order to minimize SSE, we need to maximize $trace(Q^\top XX^\top Q)$

5. Two moons challenge via K-means in Chapter 4 Page 6 (20 points)

The assumption of K-means is that the data should be globular. However, when it comes to the two moon shapes, it is not that case anymore. K-means won't work in this scenario. What we need to do is to do data transformation. It's quite similar to using kernel on the data.

i) we should set K to a large number, such as 50, and run K-means many times(like 100). Then we could get 50 labels for each point;
ii) we use the mean of the points that have the same labels to stand for the whole cluster's points. And we run K-means on those new labeled means. However, we should cut down the number of K;
iii) repeat ii) until we get the number of K that we want;

The rationale behind this is that, we first divide the data into many clusters, and gradually, we cluster points that are close to each other. In the end, the points will be clustered into moon shapes.

6. Implement Kernel K-means (5 points)

Attached in the codes files

8. How many layers for the current deepest neural network? (5 points)

Check out this paper from Huang et al. from less than one week ago on ArXiv : [1603.09382v1] Deep Networks with Stochastic Depth

Excerpt from the abstract : "with stochastic depth we can increase the depth of residual networks even beyond 1200 layers and still yield meaningful improvements in test error (4.91%) on CIFAR-10"

9. 59-sampling for greedy K-means (5 points)

This scenario occurs when we are doing global K-means.
Global K-means is an initialization method of K-means. This initialization method attempts to find the global minimum of our objective function, rather than a local minimum like random initialization and K-means++ methods do.

The idea of this method is: Suppose we are given a data set X and want K clusters from it. We initially start out by solving the clustering problem of k=1 clusters, for which we know the centroid will simply be the average of all the data points. Denote this solution as m1. Now we want to solve the clustering problem of k = 2 by running k-means on the centroids (m1, xn) where xn $\in$ X. So we do this for all our points in the data set, keep the partition obtained

from the clustering, and then calculate the SSE for each one. The partition with the lowest SSE becomes our solution for the clustering problem of k =2, and this partition becomes our new centroids. We repeat this process until we have the optimal solution to the clustering problem of k = K. We see that in this way, we iteratively solve the clustering problem of k = K by finding the optimal solution to the clustering problem k = K -1 and trying every point as the next centroid, keeping the partition with the minimum SSE.

This initialization method is very costly, since we have to run k-means for every data point, which makes this algorithm very slow. That's why we want to do sampling when we try to find another centroid.

At least 59 samples should be chose so that at least one of them is located in the top 5% candidates points with 95% confidence interval.

This is actually is a probability problem. We assume that there are n points in the dataset, and we choose x points so that all of them are below 95%, and the probability is less than (1 - 95%). As a result, we can get the following equation:
$(1 - 5\%)^x$ <= 1-95%
Thus: X >= 59.

10. Rule number of d items (5 points)

Given d unique items:
1. Total number of itemsets = 2^d -1.
This is because for each item, we have two options, to be chosen or not to be chosen. However, we should exclude the empty itemset. Thus the total number of itemsets should be 2^d -1.

2. Total number of possible association rules:

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

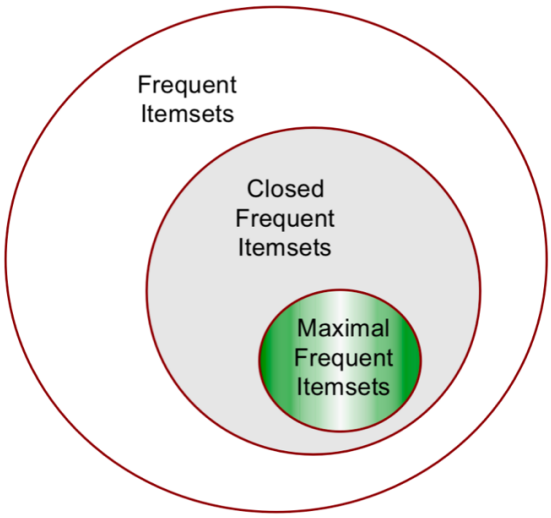
$$= 3^d - 2^{d+1} + 1$$

First, we count the number of ways to create an itemset that forms the left hand side of the rule. We assume there are k items in the left hand side. Next, for each size k itemset selected for the left-hand side, count the number of ways to choose the remaining d-k items to form the right-hand side of the rule. The left hand side possible itemsets should be
$\sum_{k=1}^{d-1}[\binom{d}{k} * \sum_{j=1}^{d-k}\binom{d-k}{j}] = 3^d - 2^{d+1} + 1$

11. The differences between maximal set and closed frequent set in terms of information loss (5 points)

Answer: First, let look at the definitions of both maximal frequent set and closed frequent set. An itemset is maximal frequent if it is frequent and none of its immediate supersets is frequent.
An itemset X is closed if X is frequent and none of its immediate supersets has the same support as the itemset X. On the contrary, X is not closed if at least one of its immediate supersets has support count as X.

Only based on the definitions of both maximal frequent set and closed frequent set, there is no relationship between the two. However, when we look at the information loss, the closed frequent itemset will contain the maximal set. Just as the picture below shows. There would be some information loss from closed frequent itemset to maximal itemset.



12. How to apply association rules for clustering? Find a paper in this topic (5 points)

Htun Zaw Oo, Nang Saing Moon Kham. “Pattern Discovery Using Association Rule Mining on Clustered Data.” International Journal of New Technology and Research (IJNTR), Vol. 4, Issue 2, February 2018, Pages 07-11.
In this paper, the aim is to find frequent user access pattern from web log entries. Combined effort of clustering and association rule mining is used to apply for pattern discovery.

13. Does the min-support have the same property with support for pruning the tree? (5 points)

Usually, we use min-Apriori to find word associations. It is analogous to traditional association analysis, an itemset is considered to be a collection of words, while its support measures the degree of association among the words. And we call this support min-support. The support of an itemset can be computed based on the normalized frequency of its corresponding words.

In min-Apriori, the association among words in a given document is obtained by taking the minimum value of their normalized frequencies, i.e., min(word1,word2) = min(0.3,0.6) = 0.3. The support of an itemset is computed by aggregating its association over all the documents.

The min-support measure defined in min-Apriori has the following desired properties, which makes it suitable for finding word associations in documents:
1. min-support increases monotonically as the normalized frequency of a word increases.
2. min-support increases monotonically as the number of documents that contain the word increases.
3. min-support has an anti-monotone property just like the support for pruning the tree. For example, consider a pair of itemsets {A,B} and {A, B,C}. Since min({A,B}) > min({A, B,C}), s({A,B}) > s({A, B,C}). Therefore, support decreases monotonically as the number of words in an itemset increases.