Spectral relaxation for K-means clustering

Nikita Petrashen, Daniil Svirskiy, Nikita Drobyshev, Dinar Sharafutdinov

December 21, 2019

Contents

Introduction

In our project we will try to show, that minimization of cost function for K-means clustering can be refolmulated as a trace maximization problem of associated Gram matrix. K-means algorithm tends to minimize the total quadratic deviation of the points of the clusters from the centers of these clusters. The main goal of our work is to solve one of the problems of coordinate descent method which is used in the K-means algorithm of clustering. Coordinate descent search method is prone to local minima, but the problem can be reformulated as a trace maximization problem with special constraints. This way to look at the problem leads to the global solution. Further in the report, we are going to explain all the aspects of the solution. Our project was inspired by the paper [1].

1 Problem statement

Derive an another approach to the problem of K-means clustering using trace maximization formulation, obtain the solution via pivoted QR and then compare this method to K-means and p-K-means (K-means on low-dimensional embeddings of the data vectors which are obtained using SVD).

2 Problem study

2.1 K-means

One of the most popular methods for solving clustering problems is the k-means method. Suppose we have an observation matrix $\mathbb{A}$ of $m \times n$ dimension, where m is a number of features and n – number of elements (observations). K-means clustering aims to divide n observations into $k(\leq n)$ clusters. Matrix $\mathbb{A}$ is a set of m dimensional vectors a_i so it can be represented as: $\mathbb{A} = [a_1, \dots, a_n]$. Problem can be written as a minimization problem for associated sum-of-squares cost function:

$$ss(\Pi) = \sum_{i=1}^{k} \sum_{s=1}^{s_i} ||a_s^{(i)} - m_i||^2, \tag{1}$$

where

$$m_i = \sum_{s=1}^{s_i} a_s^{(i)} / s_i \tag{2}$$

Here we denote s_i as a total number of elements in cluster i and $a_s^{(i)}$ is the s-th element in this cluster.

In other words, our task is to find a matrix of permutations $\mathbb{E}$ which will give:

$$\mathbb{AE} = [\mathbb{A}_1, \dots, \mathbb{A}_k],\ \mathbb{A}_i = [a_1^{(i)}, \dots, a_{s_i}^{(i)}]. \tag{3}$$

K-means algorithm (also named as Lloyd's algorithm) can be described as follows:

1. First step is to take initial set of k means $m_1^{(0)}, \dots, m_k^{(0)}$ which are chosen randomly. Here index (0) denotes first step of itertaion. The number of clusters k is chosen manually.
2. Now we assign each observation to the cluster whose mean has the least squared Euclidean distance. Mathematically set $s_i^{(t)}$ on the iteration step t for cluster i is formed as:

$$s_i^{(t)} = \{a_s : ||a_s - m_i^{(t)}||^2 \le ||a_s - m_j^{(t)}||^2 \forall j, 1 \le j \le k\},$$

   where each a_s assigned only to one cluster $s_i^{(t)}$.
3. Finally, we update mean values in cluster i by:

$$m_i^{(t+1)} = \sum_{s=1}^{s_i^{(t)}} a_s^{(i)} / s_i^{(t)}.$$

Using this algorithm we can face several problems:

- The global minimum of the total quadratic deviation $ss(\prod)$ is not guaranteed to be achieved, but only one of the local minima.
- The result depends on the choice of the initial cluster means $m_1^{(0)}, \dots, m_k^{(0)}$; their optimal choice is unknown.
- The number of clusters k must be known in advance.

In the next part we will show how this problem can be reformulated as a trace maximization problem which tends to a global optimal solution.

2.2 Connection between K-means and Spectral Relaxation

Let's consider $i-th$ element of the sum from (1):

$$ss_i = \sum_{s=1}^{s_i} ||a_s^{(i)} - m_i||^2. \tag{4}$$

Then we rewrite our sums (2) and (4) in a matrix form which gives:

$$m_i = \sum_{s=1}^{s_i} a_s^{(i)} / s_i = 1/s_i \sum_{s=1}^{s_i} a_s^{(i)} = \mathbb{A}_i e / s_i, \tag{5}$$

$$ss_i = \sum_{s=1}^{s_i} ||a_s^{(i)} - m_i||^2 = (a_1^{(i)} - m_i, a_1^{(i)} - m_i) + \ldots + (a_{s_i}^{(i)} - m_i, a_{s_i}^{(i)} - m_i),$$

sum of scalar products can be expressed as:

$$(a_1^{(i)} - m_i, a_1^{(i)} - m_i) + \ldots + (a_{s_i}^{(i)} - m_i, a_{s_i}^{(i)} - m_i) = ||\mathbb{A}_i - m_i e^T||_F^2, \tag{6}$$

where $e^T = [1, \ldots, 1]$ is a vector with ones of apropriate dimension and $\mathbb{A}_i$ is a matrix of $i - th$ cluster.

Now if we put representation of m_i from (5) into (6) we get:

$$||\mathbb{A}_i - m_i e^T||_F^2 = ||\mathbb{A}_i(I_{s_i} - ee^T/s_i)||_F^2, \tag{7}$$

where I_{s_i} is an identity matrix of $s_i \times s_i$ dimension.

We know that Frobenius norm is a norm $||A||_F = \sqrt{trace(A^*A)}$ so we can write (7) as:

$$||\mathbb{A}_i(I_{s_i} - ee^T/s_i)||_F^2 = trace((I_{s_i} - ee^T/s_i)\mathbb{A}_i^T\mathbb{A}_i(I_{s_i} - ee^T/s_i)), \tag{8}$$

here we use the fact that $(I_{s_i} - ee^T/s_i)^T = (I_{s_i} - ee^T/s_i)$.

Now using a property of trace $trace(AB) = trace(BA)$ and the fact that that $(I_{s_i} - ee^T/s_i)$ is a projection matrix so $(I_{s_i} - ee^T/s_i)^2 = (I_{s_i} - ee^T/s_i)$ we can get:

$$trace((I_{s_i} - ee^T/s_i)\mathbb{A}_i^T\mathbb{A}_i(I_{s_i} - ee^T/s_i)) = trace(\mathbb{A}_i(I_{s_i} - ee^T/s_i)\mathbb{A}_i^T) \tag{9}$$

and finally:

$$trace(\mathbb{A}_i(I_{s_i} - ee^T/s_i)\mathbb{A}_i^T) = trace(\mathbb{A}_i^T\mathbb{A}_i - (\frac{e^T}{\sqrt{s_i}})\mathbb{A}_i^T\mathbb{A}_i(\frac{e}{\sqrt{s_i}})), \tag{10}$$

which gives expression for $ss(\prod)$:

$$ss(\prod) = \sum_{i=1}^{k} trace(\mathbb{A}_i^T\mathbb{A}_i) - (\frac{e^T}{\sqrt{s_i}})\mathbb{A}_i^T\mathbb{A}_i(\frac{e}{\sqrt{s_i}}). \tag{11}$$

Now we build matrix $\mathbb{X}$ of $n \times k$ size from elements $(\frac{e}{\sqrt{s_i}})$ which gives:

$$\mathbb{X} = \begin{pmatrix} (\frac{e_{s_1}}{\sqrt{s_1}}) & & & \\ & (\frac{e_{s_2}}{\sqrt{s_2}}) & & \\ & & \vdots & \\ & & & (\frac{e_{s_k}}{\sqrt{s_k}}) \end{pmatrix}, \tag{12}$$

where $e_{s_1}^T$ is a vector of ones with s_1 length.
Sum-of-squares cost function for $\mathbb{A}$ matrix now can be written as:

$$ss(\prod) = trace(\mathbb{A}^T\mathbb{A}) - trace(\mathbb{X}^T\mathbb{A}^T\mathbb{A}\mathbb{X}). \tag{13}$$

It's minimization is equivalent to:

$$max(trace(\mathbb{X}^T\mathbb{A}^T\mathbb{A}\mathbb{X})), \tag{14}$$

Now if let x_i be the cluster indicator vector so:

$$x_i^T = [0, \dots, 0, \underbrace{1, \dots, 1}_{s_i}, 0, \dots 0] \tag{15}$$

and the norm of x_i is $||x_i|| = \sqrt{s_i}$.
Using this matrix $\mathbb{X}$ we can write trace from (14) as:

$$trace(\mathbb{X}^T\mathbb{A}^T\mathbb{A}\mathbb{X}) = \sum_{i=1}^{k} \frac{x_i^T A^T A x_i}{x_i^T x_i} = \sum_{i=1}^{k} \frac{||Ax_i||^2}{||x_i||^2}. \tag{16}$$

Multiplication of Ax_i gives us elements of $i-th$ cluster, so we can rewrite (16) as:

$$\sum_{i=1}^{k} \frac{||Ax_i||^2}{||x_i||^2} = \sum_{i=1}^{k} s_i \left|\left| \frac{A_i e}{s_i} \right|\right|^2 = \sum_{i=1}^{k} s_i ||m_i||^2. \tag{17}$$

Ignoring the special structure of $\mathbb{X}$ problem also can be considered as a relaxed maximization problem assuming that $\mathbb{X}$ is an orthonormal matrix.

Ky-Fan theorem from article [1] gives us the fact that maximal value of $\underset{\mathbb{X}^T\mathbb{X}=I_k}{max}\ trace(\mathbb{X}^T\mathbb{A}^T\mathbb{A}\mathbb{X})$ is equal to the sum of first k largest eigenvalues of Gram matrix $\mathbb{A}^T\mathbb{A}$. Also it states that the optimal matrix $X^*$is given by k eigenvectors which correspond to the largest eigenvalues. With this we can obtain a lower bound for the cost function (1).

Now let matrix $\mathbb{X}_k$ be a $n \times k$ matrix consisting of k largest eigenvectors of matrix $\mathbb{A}^T\mathbb{A}$. This matrix transforms original data vectors from m-dimensional space to k-dimensional.

One can notice that this process is very familiar with low-rank approximation but here we try to show matrix $\mathbb{A}$ cluster structure. It is necessary to mention here that this process requires $k \geq m$ since otherwise there is less then k non-zero eigenvalues. Our computational experiments show that the accuracy of the metod increases as the value $\frac{k}{m}$ increases.

2.3 Pivoted QR

Now we can show how to make a cluster assignment using pivoted QR decomposition. Let's assume that we already have the best partition that minimizes

$ss(\prod)$ for a given matrix $\mathbb{A} = [\mathbb{A}_1, \ldots, \mathbb{A}_k]$. Building a Gram matrix gives:

$$\mathbb{A}^T\mathbb{A} = \begin{bmatrix} \mathbb{A}_1^T\mathbb{A}_1 & 0 & \ldots & 0 \\ 0 & \mathbb{A}_2^T\mathbb{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbb{A}_k^T\mathbb{A}_k \end{bmatrix} + E \tag{18}$$

Here we assume that overlaps between $i-th$ and $j-th$ clusters are close to zero ($\|E\|$ is small).

Our next step is to build a matrix $\mathbb{Y}_k$ which columns are the largest eigenvectors of each cluster:

$$\mathbb{Y}_k = \begin{pmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_k \end{pmatrix}. \tag{19}$$

$$A_i^T A_i y_i = \mu_i y_i.$$

Connection between eigenvalues of matrix $\mathbb{A}^T\mathbb{A}$ and maximal eigenvalues of $\mathbb{A}_i^T\mathbb{A}_i$ is set by Davis-Kahan $sin(\Theta)$ theorem and it can be shown that:

$$\mathbb{X}_k \equiv [x_1, \ldots, , x_k] = \mathbb{Y}_k\mathbb{V} + O(\|E\|), \tag{20}$$

where $\mathbb{X}_k$ is a matrix of k largest eigenvectors of $\mathbb{A}^T\mathbb{A}$ and $\mathbb{V}$ is an $k \times k$ orthogonal matrix. Ignoring the $O(\|E\|)$ term we can write:

$$\mathbb{X}_k^T = [\underbrace{y_{11}v_{1,\ldots}y_{1s_1}v_1}_{cluster\ 1}, \ldots, \underbrace{y_{k1}v_{k,\ldots}y_{ks_k}v_k}_{cluster\ 2}] \tag{21}$$

The key notion here in that all ν_i are orthogonal to each other. This form for $\mathbb{X}_k^T$ can be obtained as follows: we select the column in $\mathbb{X}_k^T$ and orthogonalize each other column against it. Suppose it belongs to the cluster i. Then the residuals of columns from the same cluster will have small norm (because they are almost collinear), and every other residual's norm will be not so small (because of orthogonality). Then we select the next column with the largest norm and repeat the process. After k steps we will have the desired form for $\mathbb{X}_k^T$. It is easilly shown that this process is equivalent to QR decomposition with column pivoting. Finally, we get

$$\mathbb{X}_k^T P = Q[R_{11}, R_{12}],$$

where $R_{11}$is a k by k upper-triangular matrix. We then compute

$$\widehat{R} = R_{11}^{-1}[R_{11}, R_{12}]P^T,$$

which corresponds to the representation of our data points in the "cluster basis". Cluster lable for each point is then assigned by determining the largest in absolute value component of the corresponding column in the matrix $\widehat{R}$.

3 Computational complexity

3.1 K-means (Lloyd's algorithm which is discussed in our project)

Computing distances from the centroids is $O(kmn)$.

Recomputing the centroind centers is $O(kmn)$. So the total complexity is $O(klmn)$, where $l.$ is the number of iterations.

3.2 p-Kmeans

First we need to multiply $\mathbb{A}^T$ and $\mathbb{A}$ which is $O(mn^2)$ and compute first k eigenvectors of $\mathbb{A}^{\mathrm{T}}\mathbb{A}$ which is $O(kn^2)$. Then we apply Lloyd's algorithm using low-dimensional representation of our data, so the cost is $O(klkn)$, k is the new dimensionality of our space). Total cost is $O(mn^2 + kn^2 + k^2ln)$.

3.3 p-QR

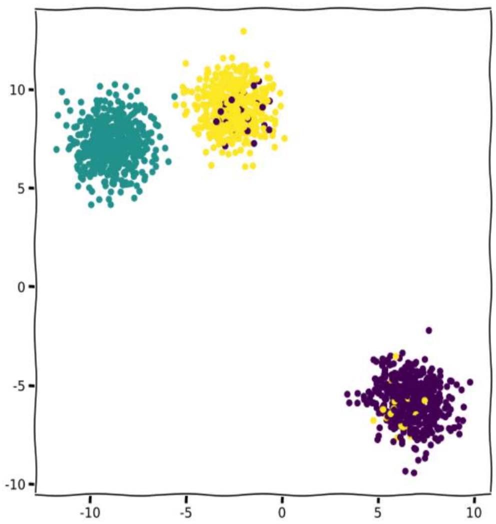
First we need to multiply $\mathbb{A}^T$ and $\mathbb{A}$ which is $O(mn^2)$ and compute first k eigenvectors of $\mathbb{A}^T\mathbb{A}$ which is $O(kn^2)$, then compute PQR decomposition which is $O(kn^2)$ and the inverse of R_{11} which is $O(k^3)$. Total cost is $O(mn^2+kn^2+k^3)$.

4 Experimental results

We conducted our experiments using two datasets. The first one is a toy sklearn blobs dataset, the second one is 20newsgroups. We used sklearn implementation of K-means and our own implementations of p-Kmeans and p-QR, which may have had its influence on the results since sklearn is a highly optimized and effective package.

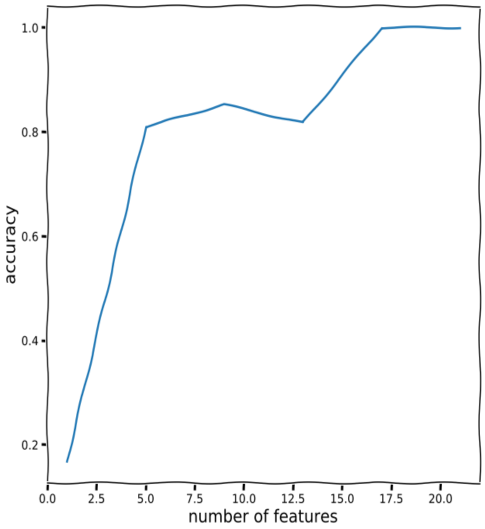
Obviously for blobs dataset with 3 clusters, 1500 samples and six features an accuracy of K-means is 100%. For the p-Kmeans, it is 87% (figure 1).

Figure 1. Prediction of 3 clusters for blobs dataset (p-Kmeans)



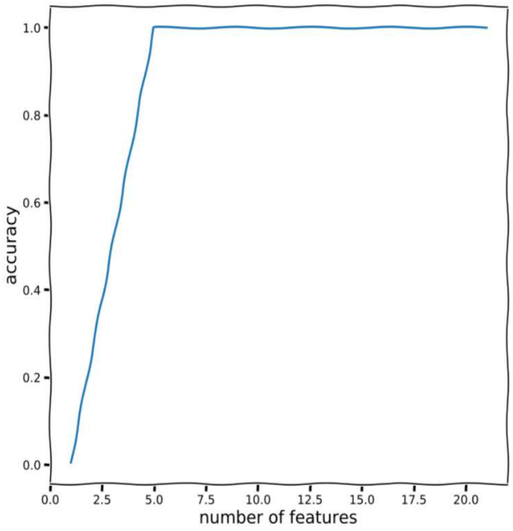
Then we checked connection between p-Kmeans accuracy and number of features. We found that when the number of features becomes substantially bigger than the number of clusters, then the algorithm works well (figure 2).

Figure 2. Connection between p-Kmeans accuracy and number of features



For p-QR, we got similar results, but it needs even a smaller number of features to get the high accuracy (figure 3).

Figure 3. Connection between p-QR accuracy and number of features



Finally, we compared results of p-QR and K-means for 20newsgroups dataset and got that for a bigger amount of clusters p-QR has higher accuracy as well as it is more stable overall (table 1).

Table 1. Accuracy of p-QR and K-means for 20newsgroups

	# categories	accuracy
K-means	2	0.80
	4	0.67
	20	0.35
p-QR	2	0.77
	4	0.75
	20	0.38

In table 2, you can see the comparison in terms of time cost. K-means is the fastest one since sklearn is a quite optimised package. But the results of p-QR is not so bad and much better than p-Kmeans even though the implementation is not optimised.

Table 2. Time costs for three algorithms

	#samples	time
K-means	100	15 ms
	1000	25 ms
	5000	45 ms
p-Kmeans	100	7 ms
	1000	1.5 s
	5000	1 min 8 s
p-QR	100	7 ms
	1000	206 ms
	5000	15 s

5 Conclusions

Although not really used in practice (we have not found any implementations of it in popular packages (sklearn.cluster.SpectralClustering implements another algorithm)), this is a very interesting approach to the problem of K-means clustering from the point of view of spectral properties of the data distribution. The referred paper has inspired many other research in this field (over 700 citations).

6 Teamwork

Our work was divided into four parts: Theory, Coding, Report and Presentation. First several days we spend on the understanding of main theoretical aspects. Then we prioritised parts of work. For Nikita Petrashen and Daniil Svirskiy priorities were theory, coding and report. For Nikita Drobyshev and Dinar Sharafutdinov: coding, theory, presentation.

7 References

1. Spectral relaxation for K-means clustering, Zha et al 2001, NIPS.

2. nla.skoltech.ru

3. https://github.com/dinarkino/NLA-project-Spectral-relaxation-for-K-means-clustering