K-MEANS-- CLUSTERING AND DIMENSIONALITY REDUCTION WITH RELATION TO PRINCIPAL COMPONENT ANALYSIS

JEN WANG

Abstract. In this paper, we briefly introduce k-Means-- clustering, a much lesser-known algorithm, and discuss its efficiency benefits over naive k-Means and k-Means++. Then we provide an in-depth interpretation of Ding and He's work[1] on the connection between Principal Component Analysis and k-Means clustering for dimensionality reduction for the case $k = 2$. We see that by construction, this connection holds accurately for the k-Means-- algorithm as well.

1. Introduction

k-Means clustering is an unsupervised algorithm for partitioning data points into k clusters based on a given distance metric for better sorting. In naive k-Means, also known as Lloyd's algorithm, the k centroids are initialized randomly. Then we assign each data point in the set to a cluster that has the minimum distance to it. Once finished, we recalculate the new centroids as the mean of the data points in the cluster and repeat the assignment procedure. This iteration is repeated until convergence such that all data points in the same cluster remain in the cluster after recalculating centroids. In this paper, we explore a slightly improved version of naive k-Means called k-Means--. The difference is in the initial centroid assignment. Instead of randomly assigning all centroids, we begin by choosing the first centroid at an initial location, then choosing the farthest point as the second centroid, and the farthest from the second as the third centroid and so on. The rest of the cluster grouping and centroid reassigning iterations until convergence are the same algorithm as naive k-Means. Thus, let $\mu_1^{(i)}, \mu_2^{(i)}, ..., \mu_k^{(i)}$ be the k centroids after the i-th iteration, where $i \in \mathbb{Z}_+$. The k-Means-- algorithm is constructed with initial centroids

$$\mu_1^{(1)} = \mathbf{0}^T, \mu_j^{(1)} := \arg\max_m \left\| x_m^{(1)} - \mu_{j-1}^{(1)} \right\|^2, \quad 2 \le j \le k, 1 \le m \le n,$$

where x_m a data point in all n data points. Next, let $C_j^{(i)}$ be the set of clustered points around centroid $\mu_j^{(i)}$ such that $x_m^{(i)} \in C_j^{(i)}$ if and only if

$$\left\| x_m^{(i)} - \mu_j^{(i)} \right\|^2 \equiv \arg\min_l \left\| x_m^{(i)} - \mu_l^{(i)} \right\|^2 \quad \forall l \in [1, k]. \tag{1}$$

Apply (1) such that all x_m are assigned to a centroid C_j, then recalculate centroids with

$$\mu_j^{(i+1)} := \frac{\sum_{m=1}^n x_m^{(i)} \mathbf{1}_{\{x_m^{(i)} \in C_j^{(i)}\}}}{\sum_{m=1}^n \mathbf{1}_{\{x_m^{(i)} \in C_j^{(i)}\}}}. \tag{2}$$

Repeating (1) and (2) in order until convergence yields the final centroids $\mu_1^*, \mu_2^*, ..., \mu_k^*$ with their respective optimal partition clusters $C_1^*, C_2^*, ..., C_k^*$.

The benefit of initializing centroids with maximum distance in between is to make sure we generate a set of centroids which covers the whole span of our data. This way, we avoid being stuck at a local minimum during centroid reassignment iteration, which can happen if we happen

to randomly initialize centroids that are too close to each other. A local minima has worst-case runtime $2^{\Omega(\sqrt{n})}$, which is undesirable for large-sized data sets.

2. Dimensionality Reduction

It is intuitive to think of k-Means clustering as a tool for dimensionality reduction, as it generalizes a cluster of points to a singular centroid value, thus reducing the dimensionality from n to k. A visual example is by down-sampling an image through reduction of the number of pixels and quantization of colors to achieve a lower quality version of the original image. Reducing file size comes with many practical uses, such as more efficient storage, faster computation, and noise elimination. The most popular method for dimensionality reduction is via linear algebra and matrices using Principal Component Analysis (PCA). Sometimes in machine learning, PCA is performed first, followed by k-means clustering to produce the most efficient result. In the following section, we attempt to analyze Ding and He's work on the connection between PCA and k-Means clustering[1]. More specifically, PCA can be seen as the continuous version of k-means clustering.

Let $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$ denote the original data matrix and $\mathbf{Y}$ denote the centered data matrix, where $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n})$ and $\mathbf{y_i} = \mathbf{x_i} - \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the mean of $\mathbf{x}$. The covariance matrix of the standardized data is

$$YY^T = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T.$$

The principal directions $\mathbf{U}_k$ and principal components $\mathbf{V}_k$ are eigenvectors satisfying

$$\mathbf{Y}\mathbf{Y}^T\mathbf{U}_k = \lambda_k \mathbf{U}_k, \mathbf{Y}\mathbf{Y}^T\mathbf{V}_k = \lambda_k \mathbf{V}_k, \mathbf{V}_k = \frac{Y^T\mathbf{U}_k}{\lambda_k^{1/2}}. \tag{3}$$

Recall the Singular Value Decomposition (SVD) of matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$ with rank $r \leq \min\{m, n\}$ is $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times r}$ contains left-singular vectors of $\mathbf{Y}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$ a diagonal matrix containing singular values of $\mathbf{Y}$, and $\mathbf{V} \in \mathbb{R}^{n\times r}$ contains right-singular vectors of $\mathbf{Y}$. Thus, Equation(3) are the defining equations for the SVD of $\mathbf{Y}$: $\mathbf{Y} = \sum_k \lambda_k^{1/2} \mathbf{U}_k \mathbf{V}_k^T$ (Golub & Van Loan, 1996). Visually, projecting values of $\mathbf{Y}$ onto principal direction $\mathbf{U}_k$ yields elements of $\mathbf{V}_k$.

To show the connection between PCA and k-Means, Ding and He's paper begins with the $k = 2$ case. Let us first rewrite equation (2) in vector form as the objective function of k-Means for centroid reassignment and omit iteration labels $\cdot^{(i)}$ for a general case. Then we want to minimize

$$D_j = \sum_{j=1}^{k} \sum_{i\in C_j} (\mathbf{x}_i - \mathbf{m}_j)^2.$$

The indicator function from the construction of μ_j in Equation(2) is written above as the second summation. Thus $\mathbf{m}_j = \sum_{i\in C_j} \mathbf{x}_i / n_j$ is the iterative calculation of centroid for cluster C_j, where $n_j = \sum_{i=1}^{n} \mathbf{1}_{\{i\in C_j\}}$ is the number of points inside C_j.

In the case of only two clusters C_1 and C_2, let

$$d(C_1, C_2) := \sum_{i\in C_1} \sum_{l\in C_2} (\mathbf{x}_i - \mathbf{x_l})^2$$

be the sum of squared difference between the clusters. And the objective function is

$$D_2 = \sum_{j=1}^{2} \sum_{i\in C_j} (\mathbf{x}_i - \mathbf{m}_j)^2.$$

Expanding the square and simplying, we obtain

$$\begin{aligned}
D_2 &= \sum_{j=1}^{2}\sum_{i\in C_j}(\mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{m}_j + \mathbf{m}_j^T\mathbf{m}_j) \\
&= \sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{j=1}^{2}\sum_{i\in C_j}\mathbf{x}_i^T\frac{\sum_{l\in C_j}\mathbf{x}_l}{n_j} + \sum_{j=1}^{2}n_j\mathbf{m}_j^T\mathbf{m}_j \\
&= \sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{j=1}^{2}\frac{1}{n_j}\sum_{i\in C_j}\sum_{l\in C_j}\mathbf{x}_i^T\mathbf{x}_l + \sum_{j=1}^{2}n_j\left(\frac{\sum_{l\in C_j}\mathbf{x}_l}{n_j}\right)^T\left(\frac{\sum_{i\in C_j}\mathbf{x}_i}{n_j}\right) \\
&= \sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{j=1}^{2}\frac{1}{n_j}\sum_{i\in C_j}\sum_{l\in C_j}\mathbf{x}_i^T\mathbf{x}_l + \sum_{j=1}^{2}\frac{1}{n_j}\sum_{i\in C_j}\sum_{l\in C_j}\mathbf{x}_i^T\mathbf{x}_l \\
&= \sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - \sum_{j=1}^{2}\frac{1}{n_j}\sum_{i,l\in C_j}\mathbf{x}_i^T\mathbf{x}_l.
\end{aligned}$$

For within cluster pairwise squared difference we have

$$\sum_{i,l\in C_j}(\mathbf{x}_i-\mathbf{x}_l)^2 = \sum_{i,l\in C_j}(\mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_l + \mathbf{x}_l^T\mathbf{x}_l) = 2n_j\sum_{i\in C_j}\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{i,l\in C_j}\mathbf{x}_i^T\mathbf{x}_l.$$

Therefore we have the expression

$$\sum_{i,l\in C_j}\mathbf{x}_i^T\mathbf{x}_l = n_j\sum_{i\in C_j}\mathbf{x}_i^T\mathbf{x}_i - \frac{1}{2}\sum_{i,l\in C_j}(\mathbf{x}_i-\mathbf{x}_l)^2.$$

Plugging this identity into the equation for D_2, we obtain

$$\begin{aligned}
D_2 &= \sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - \sum_{j=1}^{2}\frac{1}{n_j}\left(n_j\sum_{i\in C_j}\mathbf{x}_i^T\mathbf{x}_i - \frac{1}{2}\sum_{i,l\in C_j}(\mathbf{x}_i-\mathbf{x}_l)^2\right) \\
&= \frac{1}{2}\sum_{j=1}^{2}\frac{1}{n_j}\sum_{i,l\in C_j}(\mathbf{x}_i-\mathbf{x}_l)^2 \\
&= \frac{1}{2n_1}\sum_{i,l\in C_1}(\mathbf{x}_i-\mathbf{x}_l)^2 + \frac{1}{2n_2}\sum_{i,l\in C_2}(\mathbf{x}_i-\mathbf{x}_l)^2 \\
&= \frac{1}{2n_1}d(C_1,C_1) + \frac{1}{2n_2}d(C_2,C_2).
\end{aligned}$$

Define $n\overline{\mathbf{y}} := \sum_{i=1}^{n}\mathbf{y}_i$. Centering the dataset with $\mathbf{y}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ and $\sum_{i=1}^{n}\mathbf{y}_i^T\mathbf{y}_i = n\overline{\mathbf{y}^2}$ a constant, we construct J_D to be a new distance objective function

$$J_D := \frac{n_1n_2}{n}\left(2\frac{d(C_1,C_2)}{n_1n_2} - \frac{d(C_1,C_1)}{n_1^2} - \frac{d(C_2,C_2)}{n_2^2}\right).$$

Then the squared difference can be rewritten as

$$D_2 = n\overline{\mathbf{y}^2} - \frac{1}{2}J_D. \tag{4}$$

Thus, minimizing the squared difference objective is equivalent to maximizing J_D. The first term of J_D is the inter-cluster distance. Maximizing it means we want the clusters to be as far apart as possible. This is consistent with the k-Means-- algorithm as we choose the next cluster to be the farthest from the current cluster. The second and third terms of J_D are the within-cluster distance

among points in the same cluster. Minimizing them corresponds to only keeping points close to each other in the same cluster. The purpose of the new objective function is that J_D leads to a continuous solution of k-means clustering via the principal component.

Theorem 1. *For k-Means clustering where $k = 2$, the continuous solution of the cluster indicator vector is the principal component $\mathbf{V}_1$, i.e., clusters C_1, C_2 are*

$$C_1 = \{i \mid \mathbf{V}_1(i) \leq 0\}, \quad C_2 = \{i \mid \mathbf{V}_1(i) > 0\}. \tag{5}$$

The optimal value of the k-Means objective satisfies the bounds

$$n\overline{\mathbf{y}^2} - \lambda_1 < D_2 < n\overline{\mathbf{y}^2}. \tag{6}$$

Proof. Consider the squared distance matrix $\mathbf{L} = (l_{ij})$, where $l_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$. Let the centered distance matrix $\hat{\mathbf{L}}$ be the squared distance matrix subtracting its column and row means, i.e.

$$\hat{l}_{ij} = l_{ij} - l_{i.}/n - l_{.j}/n + l_{..}/n^2, \tag{7}$$

where $l_{i.} = \sum_j l_{ij}, l_{.j} = \sum_i l_{ij}$, and $l_{..} = \sum_i \sum_j l_{i,j}$. Define the cluster indicator vector to be

$$c(i) = \begin{cases} \sqrt{\frac{n_2}{nn_1}} & i \in C_1, \\ -\sqrt{\frac{n_1}{nn_2}} & i \in C_2. \end{cases} \tag{8}$$

The cluster indicator vector has zero mean and unit norm:

$$\sum_i c(i) = \sum_{i \in C_1} \sqrt{\frac{n_2}{nn_1}} - \sum_{i \in C_2} \sqrt{\frac{n_1}{nn_2}} = n_1\sqrt{\frac{n_2}{nn_1}} - n_2\sqrt{\frac{n_1}{nn_2}} = 0,$$
$$\sum_i c^2(i) = n_1\frac{n_2}{nn_1} + n_2\frac{n_1}{nn_2} = \frac{n_1 + n_2}{n} = 1.$$

It is straight forward to see that $\mathbf{c}^T\mathbf{L}\mathbf{c} = -J_D$. Furthermore, this property holds true for the centered matrix $\hat{L}$ as well, since the last three terms $-l_{i.}/n - l_{.j}/n + l_{..}/n^2$ in Equation(7) contribute to zero in $\mathbf{c}^T\hat{\mathbf{L}}\mathbf{c}$. Thus, maximizing J_D is equivalent to minimizing $\mathbf{c}^T\mathbf{L}\mathbf{c}$. Recall Rayleigh Quotient, which states for a symmetric matrix $\mathbf{A}$, finding the $\min \mathbf{x}^T\mathbf{A}\mathbf{x}$ subject to $\|\mathbf{x}\|^2 = 1$ is equivalent to solving the unconstrained problem

$$\min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}},$$

which has solution when $\mathbf{x}$ is the smallest eigenvector of $\mathbf{A}$. By construction, the cluster indicator matrix has unit norm; thus, the objective

$$\min \mathbf{c}^T\mathbf{L}\mathbf{c} = -J_D \quad \text{subject to } \|\mathbf{c}\|^2 = 1$$

has solution of the desired cluster indicator vector $\mathbf{c}$ whose eigenvalue is the smallest (most negative) of

$$\hat{\mathbf{L}}\mathbf{v} = \lambda\mathbf{v} \equiv \left[(\hat{\mathbf{L}} - \lambda\mathbf{I})\mathbf{v} = 0\right].$$

Since $\hat{\mathbf{L}}$ has row-sum equals to zero, one solution to $(\hat{\mathbf{L}} - \lambda\mathbf{I})\mathbf{v} = 0$ is when $\lambda = 0$ and $\mathbf{v} = (1, ..., 1)^T$. Let this eigenvector be denoted as $\mathbf{e} := (1, ..., 1)^T$. Since all other eigenvectors of $\hat{\mathbf{L}}$ are orthogonal to $\mathbf{e}$, i.e. $\mathbf{v}^T\mathbf{e} = 0$, the each eigenvector also have mean zero property: $\sum_i \mathbf{v}(i) = 0$. We can find a more precise description of these eigenvectors by performing some arithmetic on the vectors of the

squared distance matrix $\hat{\mathbf{L}}$.

$$\begin{aligned}
l_{i\cdot} &= \sum_{j=1}^{n} l_{ij} = \sum_{j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{j=1}^{n} \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j \\
&= n\mathbf{x}_i^2 2\mathbf{x}_i^T \sum_{j=1}^{n} \mathbf{x}_j + \sum_{j=1}^{n} \mathbf{x}_j^T\mathbf{x}_j = n\mathbf{x}_i^2 - 2n\mathbf{x}_i\overline{\mathbf{x}} + n\overline{\mathbf{x}^2}. \\
l_{\cdot j} &= n\mathbf{x}_j^2 - 2n\mathbf{x}_j\overline{\mathbf{x}} + n\overline{\mathbf{x}^2}. \\
l_{\cdot\cdot} &= \sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{x}_i^T\mathbf{x}_i - 2\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{x}_j^T\mathbf{x}_j \\
&= n\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 + n\sum_{j=1}^{n} \|\mathbf{x}_j\|^2 - 2n^2\overline{\mathbf{x}^2} \\
&= 2n\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - 2n^2\overline{\mathbf{x}^2} = 2n^2\overline{\mathbf{x}^2} - 2n^2\overline{\mathbf{x}^2} = 2n^2\overline{\mathbf{x}^2} = 0.
\end{aligned}$$

Plugging these quantities into Equation(7), we obtain

$$\begin{aligned}
\hat{l}_{ij} &= l_{ij} - \frac{l_{i\cdot}}{n} - \frac{l_{\cdot j}}{n} + \frac{l_{\cdot\cdot}}{n^2} \\
&= \left(\mathbf{x}_i^2 - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^2\right) - \left(\mathbf{x}_i^2 - 2\mathbf{x}_i^T\overline{\mathbf{x}} + \overline{\mathbf{x}^2}\right) - \left(\mathbf{x}_j^2 - 2\mathbf{x}_j^T\overline{\mathbf{x}} + \overline{\mathbf{x}^2}\right) \\
&= -2\mathbf{x}_i^T\mathbf{x}_j + 2\mathbf{x}_i^T\overline{\mathbf{x}} - 2\overline{\mathbf{x}^2} + 2\mathbf{x}_j^T\overline{\mathbf{x}} \\
&= -2(\mathbf{x}_i - \overline{\mathbf{x}})^T(\mathbf{x}_j - \overline{\mathbf{x}}) = -2\overline{\mathbf{y}^2}.
\end{aligned}$$

Equivalently, we can conclude that $\hat{\mathbf{L}} = -2\mathbf{Y}^T\mathbf{Y}$. Recall our goal is to maximize J_D, which is equivalent to minimizing $\hat{\mathbf{L}}$. Therefore, the continuous solution is achieved through the eigenvector corresponding to largest (most positive) eigenvalue of $\mathbf{Y}^T\mathbf{Y}$. By definition, this would be the principal component $\mathbf{V}_1$. Lastly, an upper bound on J_D can be found with $J_D = -\mathbf{c}^T\hat{\mathbf{L}}\mathbf{c} = 2\mathbf{c}^T\mathbf{Y}^T\mathbf{Y}\mathbf{c} < 2\lambda_1$. Thus Equation(4) yields $n\overline{\mathbf{y}^2} - \lambda_1 < D_2 < n\overline{\mathbf{y}^2}$, the bounds given in Theorem1.

□

References

[1] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. *Proceedings of the 21 st International Conference on Machine Learning*, 2004.