Maths behind K-means Clustering

Andrew

July 2023

Warning! This document is written by a CS undergraduate trying to understand Maths after a long time of struggling. Please let me know immediately if anything is wrong or unacceptable to exist.

1 Initial rules

Since K-means is an **unsupervised** learning algorithm, our input is a mere dataset with no labels. All we have at the first step is an observation matrix with columns are the total amount of observations and rows are the number of dimensions (features) for each observation. Now, let's start defining our observation and label matrix. Our observation matrix should have a form like this

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} \in R^{d \times N} \text{ where}$$
$$\mathbf{d} : \text{observation dimension (number of features)}$$
$$\mathbf{N} : \text{number of observations}$$

and our label matrix should have a similar form

$$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix} \in R^{k \times N} \text{ where}$$
$$\mathbf{k} : \text{number of clusters}$$
$$\mathbf{N} : \text{number of observations}$$

An observation will have its corresponding label vector y. Each element of the vector represents its binary value of that corresponding centroid. For example, suppose we have a dataset of real-estate properties having 2 features for each observation, which are the total area and prices. Then our observation matrix will have 2 rows (2 features) and N columns (depends on how large our dataset). Suppose we have magically known there are 4 groups those properties will be classified into (4 clusters), then our label matrix should have 4 rows and N columns. Now, in order to express that observation x_1 belongs to the 3^{rd} centroid, its corresponding label vector y_1 will have the form $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$. Since each observation can only belong to a single centroid, if x_1 is classified into group k, then $y_{ik} = 1$ and $y_{ij} = 0, \forall j \neq k$. So, we've just form our conditions for the label vectors.

$$y_{ij} \in \{0, 1\}, \sum_{j=1}^{K} y_{ij} = 1$$

2 Construct loss functions and optimizing method

2.1 Loss function

Let's call our centroids as $\mathbf{m}$, then we'll need to calculate the loss for each centroid. In this problem, for a 2-d dataset (points as observations), the loss for each point can be calculated as the Euclidean distance from that point to its centroid. The loss from point $x_1(a,b)$ to its centroid $m_k(c,d)$ is

$$d\left(x_1, m_k\right) = \sqrt{\left(a-c\right)^2 + \left(b-d\right)^2} = \|\mathbf{x_1} - \mathbf{m_k}\|_2$$

Note: the number $_2$ is the power of the difference between two points in the square root

Next, we'll square it to get rid of the square root, generalize it and apply the label vector as $y_{ik} = 1$, $y_{ij} = 0$, $\forall j \neq k$

$$y_{ik}\|\mathbf{x_i} - \mathbf{m_k}\|_2^2$$

this expression means that only when the vector label y_{ik} is true (has value 1), we are able to calculate the loss from that point to its corresponding centroid. Otherwise, we'll still calculate the distance $d(x_i, m_j)$ but since $y_{ik} = 0$, we'll get nothing but 0. To generalize this math for the rest of the label vector, we'll have

$$\sum_{j=1}^{K} y_{ik}\|\mathbf{x_i} - \mathbf{m_k}\|_2^2$$

Generalize for the whole dataset, we'll get our loss function

$$Loss(Y, M) = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ik}\|\mathbf{x_i} - \mathbf{m_k}\|_2^2$$

Now we've got a new variables $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_k \end{bmatrix}$, which are the coordinates of all centroids. Since it is a loss function, we'll need to find its smallest value (optimize) and therefore lead us to solving this equation:

$$\text{Y, M} = argmin_{Y,M}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\|\mathbf{x_i} - \mathbf{m_j}\|_2^2$$

$$\text{subject to: } y_{ij} \in \{0,1\}\ \forall i,j, \sum_{j=1}^{K} y_{ij} = 1, \forall i$$

Note: the equation above means that we'll have to find matrices Y, M such that the loss function will have its *min* value. The *argmin* expression means that instead of finding the usual *min* of the equation, which is the smallest value of the loss function, we now want to find the *argument value* resulting to that *min*.

Another note (Why not?): after completing the equation, this actually resembles the *Sum of squared errors (SSE)* which has the following formula

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

with y_i as our observations and $\hat{y}_i$ as their corresponding centroids. Take a look and ponder over it for a while, you'll see the relation.

2.2 Let's optimize

In machine learning, optimizing our loss function usually comes with taking its derivative (the slope) and tweak our weights such that its derivative will move in an opposite direction to that slope. Therefore the loss function will be reduced and ultimately we'll get its global minimum (if we're lucky enough). This is called Gradient Descent. In our problem, however, we'll use a quite similar process but we still hold the same goal, which is finding the minimum of our loss function. But it won't be easy as we're having 2 variables Y and M. So we'll try to ignore 1 of the variables and take the derivative of the other (for me it looks like partial derivative).

2.2.1 Ignore M, find Y

"Suppose we've found our centroids, let's label our observations such that the loss function achieve the min value". For an easier grasp, labeling for **all** observations can be seen as labeling for **a single** observation.

$$y_i = \underset{y_i}{\operatorname{argmin}} \sum_{j=1}^{K} y_{ij} \|\mathbf{x_i} - \mathbf{m_j}\|_2^2$$

$$\text{subject to: } y_{ij} \in \{0, 1\}\ \forall j,\ \sum_{j=1}^{K} y_{ij} = 1$$

Can you see the pattern? This equation means that finding the *min* value depends on the distance $\|\mathbf{x_i} - \mathbf{m_j}\|_2$, since y_{ij} can only be 1 for a single centroid, otherwise it equals 0. Consequently, this equation shows us that **the observation x_i belongs to its closest centroid**.

2.2.2 Find M, ignore Y

"Suppose we've labeled our observations, let's calculate their new centroids such that the loss function achieve the *min* value". Similar to Y, the equation can be shortened as follow

$$m_j = \underset{m_j}{\operatorname{argmin}} \sum_{i=1}^{N} y_{ij} \|\mathbf{x_i} - \mathbf{m_j}\|_2^2$$

This equation means that we're calculating for a single centroid. Remember the condition $y_{ij} \in \{0, 1\},\ \forall i, j$? Now we only choose one centroid, which means a single defined j, and choose all observations. $y_{ij} = 1$ only when that label vector of x_i has the right j $(j = k)$. Therefore, this equation means we're finding the value of a particular centroid such that the sum of the squared distance from all of its points to itself will have the smallest value. Now we're gonna take the derivative of this loss function just as the *Gradient Descent* I've mentioned earlier. Remember, since we're taking a partial derivative with respect to m_j, y_{ij} will be considered as a constant.

$$\frac{\partial l(m_j)}{\partial m_j} = \sum_{i=1}^{N} 2y_{ij}(x_i - m_j)\frac{\partial (x_i - m_j)}{\partial m_j} = 2\sum_{i=1}^{N} y_{ij}(m_j - x_i)$$

Solve this derivative when it equals 0, we'll get

$$m_j \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} y_{ij} x_i$$

$$\Rightarrow m_j = \frac{\sum_{i=1}^{N} y_{ij} x_i}{\sum_{i=1}^{N} y_{ij}}$$

Can you see the pattern? The *numerator* is the **sum of all observations** in that cluster and the *denominator* is the **total number of observations** in that same cluster. So, it is clear that the centroid m_j is the **mean of all observations in cluster** j.