k-means clustering

Sylesh Suresh

March 2018

1 Introduction

So far, we have learned about supervised learning algorithms, which analyze data with class labels. Clustering algorithms, however, are unsupervised learning techniques used to separate unlabeled data into different groups, or clusters. These clusters group similar data points; data points of each cluster will be more similar to each other than to data points of other clusters. Unlike the supervised learning algorithms, which decide what class new data points belong to based on previously seen labeled data points, clustering algorithms assign labels to data points themselves. k-means clustering is one of the simplest and most popular clustering algorithms.

2 Algorithm

The k-means clustering algorithm is as follows:

1. Randomly pick k points to be centroids.
2. Assign each data point to its nearest centroid.
3. Move each centroid to the center of the data points which were assigned to it.
4. Repeat steps 2 and 3 until no data point has its centroid assignment changed or until a specified maximum iteration count has been reached.

k, the number of clusters, is a hyperparameter chosen beforehand. The distance measure is the squared Euclidean distance:

$$d(\boldsymbol{x}, \boldsymbol{y})^2 = \sum_{j=1}^{n} (x_j - y_j)^2 = ||\boldsymbol{x} - \boldsymbol{y}||^2$$

for n dimensions.

The k-means algorithm is essentially an iterative method of minimizing the within-cluster sum of squares (WCSS):

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} d(\boldsymbol{x}, \boldsymbol{\mu_i})^2$$

where C_i is the ith cluster and μ_i is the centroid of the ith cluster.

3 Limitations

One of the key limitations of k-means clustering is the need to specify the number of clusters as a hyperparameter; in many cases, the number of clusters may not be obvious from looking at the data (particularly when the feature space has many dimensions and is thus not easily visualizable). Another limiting factor is the computational expense of the algorithm if the number of maximum iterations is set particularly high (or if it is not set at all). One way to ameliorate this problem is to provide a tolerance; if the change in SSE is less than or equal to this tolerance, the algorithm will terminate. However, this solution is susceptible to convergence to a local minimum.

4 k-means++

An alternative algorithm to the classic k-means algorithm is k-means++. the algorithm is as follows:

1. Initialize an empty set M to store the k centroids about to be selected.
2. Randomly choose the first centroid from the input samples and store it in M.
3. For each sample x_i that is not in M, find the minimum squared distance $d(x_i, M)$ to any centroid in the set.
4. Select the next centroid using the probability distribution $d(x_i, M) / \sum_j d(x_j, M)$.
5. Repeat steps 2 and 3 until k centroids are chosen.
6. Use classic k-means algorithm to fine tune centroid locations.

This algorithm initializes the centroids more judiciously such that they are farther apart and thus provides more consistent results than the classic algorithm.