Technology Review

Discussion of K-Means, Kernel K-means, K-Medians, K-Medoids – their application and comparisons

Author: Shruti Kalia (User: skalia2, Email Id: skalia2@illinois.edu)

Introduction

A cluster is a collection of data objects which are Similar (related) to one another within the same group or Dissimilar (unrelated) to the objects in other groups. Cluster analysis, which is also known as **clustering** or data segmentation is to partition a set of data points into a set of groups (clusters) that are similar; this is unsupervised learning as there are no predefined classes.

There are many clustering methods and one of them is the Partitioning algorithm based on the distance method, which is to partition data in a high dimensional space into multiple clusters – the type of data here, is text data, which is very popular in social media and the web. The methods to handle such data include K-Means, K-Medoids, K-Medians, which we are going to discuss in this review.

Body

The **K-Means clustering algorithm** considers every center is represented by the center of the cluster (or, centroid). Given K, the number of clusters, the K-Means clustering algorithm is outlined as below steps:
Step1: Select the K points as initial centroids
Step2: Repeat the below steps in a loop until the convergence criterion is satisfied

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., **mean** point) of each cluster because the initial randomly selected centroid may not be a very good one

This is an **efficient algorithm** for clustering with the computational time complexity of O(nkl), which is linear to the number of objects, where 'n' is the total number of objects in the dataset, 'k' is the required number of clusters we identified and 'l' is the number of iterations, k≤n, l≤n.

There are many ways to automatically determine the "best" K and in practice, one can run a range of values for K to select the best value.

The initialization becomes important if we want to find high-quality clusters. One of the methods proposed for better initialization of k seeds is K-Means++.

There are different kinds of measures that can be used to calculate the distance and assign to the closest centroid such as Euclidean distance (used often), Manhattan distance, and Cosine similarity.

K-Means method is **sensitive** to noise data and outliers. For example, if we look at a company's salary and add a very high salary, then the average salary of the whole company almost shifts a lot. So, there are variations like **K-medians or K-medoids** algorithms that overcome this outlier noise data problem.

In the **K-Medoids clustering algorithm**, we use the most centrally located object in the cluster (i.e., medoids) instead of the mean value of an object in a cluster as our centroid. The difference is K-Means selects the K virtual centroid, but K-Medoids finds K representative of real objects. The steps are as follows:

Step 1: Select the K points as initial medoids (i.e., representative objects)
Step 2: Repeat the below steps in a loop until the convergence criterion is satisfied

- Assign each remaining object to the nearest medoids
- Randomly select a non-medoid object
- Compute the total cost of swapping the medoid with a non-medoid object
- Swap medoid with the non-medoid object if it improves the clustering quality (i.e., total cost < 0)

K-Medoids have some disadvantages in that it is more costly than the K-Means method. It does not scale well for large data sets. The computational complexity is O(K(n-K)^2 which is quite **expensive.** But K-medoids prove superiority to K-means in the execution time, quality clustered classes, and being non-sensitive to outliers and reduction of noise.

In the **K-medians clustering algorithm**, as **medians** are **less sensitive** to outliers than means, therefore, medians are used as a reference point using the Manhattan (or, city block) distance measure. The steps are as follows:
Step1: Select K points as initial K Medians (i.e., representative objects)
Step2: Repeat the below steps in a loop until the convergence criterion is satisfied

- Assign every point to its nearest median
- Re-compute the median using the median of each individual feature

Kernel K-Means algorithm (or, density-based clustering) can be used to detect **non-convex clusters** whereas K-Means can only detect clusters that are **linearly separable**. The steps are as follows:
Step 1: Map data points in the input space onto a high-dimensional feature space using the kernel function
Step 2: Perform K-Means on the mapped feature space
The computational complexity of the Kernel K-Means algorithm is higher than K-Means as it needs to compute and store n x n kernel matrix generated from the kernel function on the original data. If the original data contains 'n' objects and if 'n' is large, then the n x n kernel matrix could be very large.

Let's consider the below example.

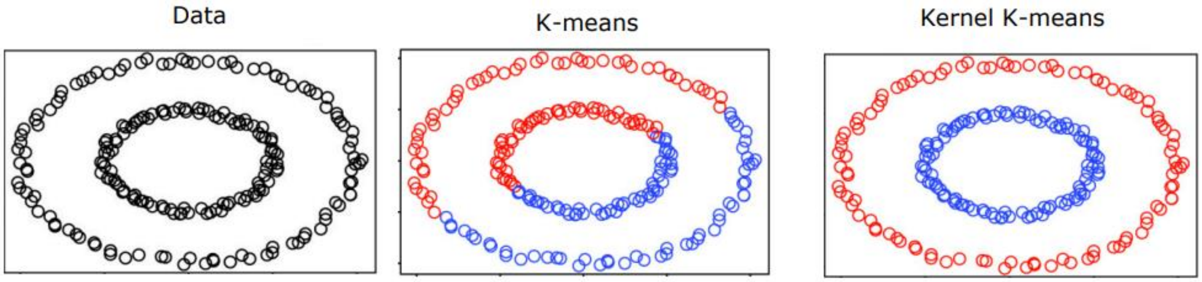

Diagram: Referred from below-mentioned Reference Number 3

The first diagram (Data) on left shows the original data points for a data set.

If we want to use K-Means to generate two clusters, we can find something linearly separable as seen in the second diagram (K-means). This doesn't generate a quality cluster and chops every cluster into two halves - the dense kernel is split, as also the ring.

However, when we use Kernel transformation mapped into a Kernel Matrix K for any two points, we see quality clusters of two different colors in the third diagram (Kernel K-means). The blue one is a core part of the center surrounded by the red one as a big ring. Thus, Kernel K-means can find "complex" clusters.

Conclusion

The problem of selecting the best algorithm is a difficult one. A good clustering algorithm ideally should produce groups with distinct non-overlapping boundaries, although a perfect separation cannot typically be achieved in practice. To determine which algorithm is good, it is a function of the type of data available and the particular purpose of analysis.

K-Means can typically be applied to data that has a smaller number of dimensions, is numeric, and is continuous, and below are its uses:

Prediction of Students' Academic Performance: The clustering algorithm serves as a good benchmark to monitor the progression of students' performance in higher institutions. It also enhances the decision-making by academic planners to monitor the candidates' performance semester by semester by improving on the future academic results in the subsequent academic session.

Spam filter: K-Means clustering techniques have proven to be an effective way of identifying spam. It works by looking at the different sections of the email (header, sender, and content). The data is then grouped together, and these groups can then be classified to identify which are spam.

Classifying network traffic: K-means clustering is used to group together characteristics of the traffic sources when the clusters are created as per the traffic types. By having precise information on traffic sources, we can grow our site and plan capacity effectively.

Fantasy Football and Sports: The challenge at the start of the season is that there is little information available to help identify the winning players. When there is little performance data available to train our model, we have an advantage for unsupervised learning. By using K-Means clustering, we can find similar players using some of their characteristics ultimately providing a better team more quickly at the start of the year, giving us an advantage.

Kernel K-Means in **health studies** is more successful to find high-risk individuals with cardiovascular diseases. It gives faster and consistent results as it is able to identify non-linear structures and is suitable for real-life data set.

References

1. Application of k-Means Clustering https://arxiv.org/ftp/arxiv/papers/1002/1002.2425.pdf
2. Usage of Kernel K-Means https://www.oatext.com/Usage-of-Kernel-K-Means-and-DBSCAN-cluster-algorithms-in-health-studies-An-application.php
3. Kernel Clustering http://www.cse.msu.edu/~cse902/S14/ppt/kernelClustering.pdf
4. 7 Innovative Uses of Clustering Algorithms https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/6224
5. Comparative Study between K-Means and K-Medoids https://www.irjet.net/archives/V6/i3/IRJET-V6I3154.pdf
6. An Enhancement Over K-Means https://arxiv.org/ftp/arxiv/papers/1706/1706.02949.pdf