Variations and improvements of K-means

Yimeng Han (yimengh2)

Intro

There are two widely used clustering algorithms, partitional and hierarchical clustering. Hierarchical clustering recursively clusters two items at a time, while partitional clustering partition the entire set of items in a cluster into two more homogeneous clusters. K-mean is one of the most widely used algorithms belonging to partitional clustering. Its variations like K-medoids, K-medians and K-modes are also widely discussed partitional algorithms. These methods require input data, number of clusters expected, and K starting centers, and proceed by assigning data value to the nearest center of the clusters. However, even though there are lots of existing variations of K-means, people are still trying to improve the time complexity of K-means.

Body

Variations of K-Means can mainly be classified into different types.
The most significant type is by "choosing different representative prototypes for the clusters". Others are "choosing better initial centroid estimates" and "Applying some kind of feature transformation techniques". We will mainly discuss the variations belong to the first type. (Reddy & Vinzamuri).

One noticeable method is K-medoids clustering. Comparing to K-mean, it is more resilient to outliers and noise because it chooses the actual data points as the prototypes instead of calculating the mean points. "The K-medoids algorithm aims to minimize the absolute error criterion rather than the SSE." (Reddy & Vinzamuri). It repeatedly assigns each point to the cluster whose representative point is nearest to them. Then randomly choosing a non-representative point and computing the cost of swapping the point with the representative object. When the total cost of swapping is converging, the process stops. Thus, it is more robust to noise comparing to K-means, however, the computational complexity is also higher. When dealing with large datasets, K-means is more preferrable than K-medoids.

Another method K-modes is also very prevalent. The main difference between K-modes and K-means is that K-modes repeatedly assigning all data points to the cluster with nearest modes and recompute the modes of each cluster, rather than computing the means of the clusters. K-modes perform better than K-means when dealing with nonnumeric datasets. When dealing with categorical datasets, the data will first be transformed into new feature spaces. However, when applying K-means to the newly transformed data, it has proven to be very ineffective because SSE function and usage of the mean are not appropriate for the categorical data. K-modes is a non-parametric clustering technique, thus can better handling "categorical data and optimizing a matching metric (L0 loss function) without using any explicit distance metric". "The K-medoids algorithm aims to minimize the absolute error criterion rather than the SSE." (Reddy & Vinzamuri).

Similar to K-modes, K-medians uses the median of each points in the cluster to calculate the center of a cluster. Because it uses the mean coordinates of the clusters as the centers, the clusters retrieved by this technique thus will be denser and more robust to noise or outliers. However, the time complexity is also higher than K-means.

There are other types of variations, for example, Intelligent K-means trying to choose better initial centroid estimates (Reddy & Vinzamuri). However, even though there are so many variations of K-means clustering technique, the most popular and widely used one is still K-means. Even though K-means is sensitive to outliers or noises, in practical conditions, most of the time we are trying to get a quick solution to generate insights instead of trying to achieve clusters as accurate as possible. Furthermore, K-means can handle large datasets easily due to the low time complexity. We care about the time complexity and large datasets the most. Therefore, many researches are done to try to improve the speed of K-means, instead of the sensitivity to outliers or accuracy.

In paper "Making k-means even faster", they "present a new method of accelerating Lloyd's algorithm which builds on Ellan's algorithm" (Hamerly, 2010). The main algorithm is to use inequality to avoid calculating point-distance center; and can skip the loop of iterating k centers 80% of the time. When dealing with large datasets, this algorithm can parallelize easily for multithread processes. As a result of their experiment, the performance of the "hamerly clustering method" takes less CPU second in most cases with different number of datasets and centers. More specifically, this algorithm "is significantly faster than any competing algorithm in data of dimensions up to about 50" (Hamerly, 2010).

Conclusion

I presented the techniques and pros and cons of different variations of K-means clustering algorithm. Different variations have different advantages and disadvantages. For example, K-means work better for categorical data and K-medians is robust to outliers. However, the most well-liked among them in practice is still K-means because it is fast and can process large datasets. When dealing with clustering in real life, we mainly care about the general structure of the clusters, and care less about the accuracy influencing by noise. Therefore, some researchers tried to find ways to simply improve the time complexity of K-means instead of concentrating on the variations. The work of Hamerly provides a faster K-means algorithm.

References

Reddy, C. K., & Vinzamuri, B. (n.d.). A Survey of Partitional and Hierarchical Clustering Algorithms.

Miraoui, I. (2020, April 13). Clustering Algorithms: A One-Stop-Shop. Retrieved November 14, 2020, from https://towardsdatascience.com/clustering-algorithms-a-one-stop-shop-6cd0959f9b8f

10.4 - K-means and K-mediods. (n.d.). Retrieved November 14, 2020, from https://online.stat.psu.edu/stat555/node/88/

Hamerly, G. (2010). Making k-means even faster. Retrieved from https://epubs.siam.org/doi/pdf/10.1137/1.9781611972801.12