Review of some text clustering techniques

Chengyuan Ai(ai13@illinois.edu)

Introduction

Clustering technology is a cutting-edge technique in the field of machine learning. Machine learning mainly uses two types of methods: supervised learning and unsupervised learning. Clustering analysis is an unsupervised learning method, and in pattern recognition, for a given data sample with a known category label, the classification problem is trained to make it possible to classify samples with unknown labels. However, in the real world, a considerable amount of data has no known labels, and their categories are missing or require extensive manual labeling to obtain them. Clustering analysis is important to discover the intrinsic knowledge and the implied patterns from the data.

Text clustering is the application of clustering analysis in the field of text data. With the continuous development of information technology and network, the scale of text data is getting larger and larger, and there is a huge amount of potential information and knowledge hidden in these massive text data. At the same time, text data is a kind of unstructured data, and the application area of clustering for text data is also very wide. In the direction of information retrieval, text clustering can be used to cluster search engines and improve the accuracy of users' access to information; in the direction of information recommendation, clustering can also extract hot topics or discover events, automatically archive texts, and help improve text visualization.

Below, I will give a review of some text clustering techniques in detail, mainly Distance-based Partitioning Clustering algorithms: K-Means, kernel K-means, K-Medians, and K-Medoids.

Text clustering techniques

Clustering analysis divides a given data sample into several categories based on similarity, and the higher the similarity the more two objects are classified into the same category, which will eventually form the data into several clusters, which can be distinguished from each other based on their shape, size, and density.

Before performing text clustering analysis, we first have to numerate or vectorize the unstructured text data, i.e., the feature selection process. Text data, has some explicit features, such as word count, word frequency, number of stop words, average word length, etc. Of course, some non-displayed features can also be used to achieve better clustering results.

Text clustering is divided into three main methods: Distance-based Clustering, Agglomerative and Hierarchical Clustering, and Distance-based Partitioning Clustering. Here, we focus on the last algorithm.

Partitioning algorithms are widely used to efficiently create clusters of objects. There are four widely used algorithms as follows:

- **K-Means clustering**: In the initialization phase of the K-means algorithm, text objects are first considered as term vectors, and then the centroids of the initial clusters are randomly specified, and in each iteration, each point is assigned to the cluster with the nearest center. The center is the average of all points in the cluster, and the coordinates of the average point are the arithmetic mean of each dimension on all points in the cluster. Then the iterations are repeated until the similarity-based objective function is satisfied.
  Pros**:** simple principle (close to the centroid), easy to implement, and the space complexity and time complexity is low.
  Cons: first, it only considers the distance between sample points, and usually the results are spherical clusters. If the density of sample points is considered, the density-based method represented by the DBSCAN algorithm can find arbitrarily shaped clusters. Secondly, the K values and initial centroids are given by the user, which is easy to fall into the local optimum. Finally, it is easy to receive outliers due to the need to consider the distribution of all points.
- **K-Medoids**: Instead of using the mean value, the K-Medoids algorithm uses the most central object in the cluster, i.e., the medoids, as the reference point. The algorithm procedure is similar to K-Means above.
  Pros: K-medoids is more robust than K-means when noise and isolated points are present.
  Cons**:** K-medoids works well for small data sets, but not for large data sets, the time complexity of the step to compute the center of mass is O(n^2), and it runs slowly.
- **Kernel K-Means**: The kernel k-means is, in effect, a projection of each sample into a higher dimensional space, and then the processed data is clustered using the ordinary k-means algorithm idea.
  Pros: It can handle nonlinear data, while traditional k-means can only handle linear data.
  Cons: It takes longer computation time because of the need to first project the nonlinear data into the high-dimensional space.
- **K-Medians:** Instead of using the mean value, the K-Medians algorithm uses the median value in the cluster as the reference point. The algorithm procedure is similar to K-Means above.
  Pros**:** K-medians is more robust than K-means when noise and isolated points are present.
  Cons: The time complexity and space complexity is more than K-Means.

Conclusion

After the above-detailed introduction of the four Distance-based Partitioning Clustering algorithm, and the comparison of their advantages and disadvantages, we can conclude that different clustering methods have different characteristics and applicability conditions, which requires us to explore the text data carefully before performing text clustering analysis, to choose the optimal clustering technique according to the characteristics of the dataset and thus get better results and faster computation speed.

Reference

Aggarwal, C. C. (2015). Mining text data. In *Data mining* (pp. 429-455). Springer, Cham.

Dhillon, I. S., Guan, Y., & Kulis, B. (2004, August). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 551-556).

https://www.coursera.org/learn/cs-410/lecture/PsyKR/10-5-text-clustering-similarity-based-approaches