Homework04

CSC240/440 Data Mining

Beilei Guo & Liwei Jiang

**What kinds of clustering methods we use?**

For this mini-project, we are working on a public dataset created in University of California, Irvine (UCI). This is a dataset made especially for studying clustering analysis. The dataset is Bag of Words dataset, in which a vocabulary is present and the counts of the words (term-frequency) in different documents is provided. In this project, we used three clustering methods, including *k*-Means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and spectral clustering.

**How do we use those clustering methods?**

We first dropped the first three lines of the document - docword.kos.txt. Then, we vectorized the raw data. The raw data records the count of words in each document. Words are recorded by word\_id(int), which could directly be used as dimensions to generate the document vectors. We then did normalization because *k*-Means clustering is “isotropic” in all directions of space and therefore tends to produce round clusters instead of elongated clusters. Normalization will set variances equal, thus avoid putting more weight on variables with smaller variance. Before applying clustering methods, we utilized t-distributed stochastic neighbor embedding (t-SNE) in our project. TSNE is a statistical method for visualizing high-dimensional data by modeling objects by two- or three-dimensional points which allow us to visualize high dimensional data like the document vectors.

The first clustering algorithm we used is *k*-Means, which is a centroid-based partitioning technique that uses the centroid of a cluster to present that cluster. We first selected two as the optimum number of clusters. Results shown that one cluster has 3103 words in the vocabulary while the other one has 327 words. We visualized the frequency difference between two blog types using a histogram (shown below) and traced the high frequency words back to the vocab.kos.txt document.

Chart

Description automatically generated

Words that have a relative high frequency in the first blog type are: “account”, “Bush”, “electoral”, “governor”, “house”, “Kerry”, “November”, “poll”, “polls”, “republicans”, “senate”. Words that have a relative high frequency in the second blog type are “bush”, “campaign”, “dean”, “democratic”, “general”, “house”, “Iraq”, “Kerry”, “poll”, “president”, “war”. We conclude that the first blog type represents republican articles, and the second blog type represents democratic articles. It is worth noting that the high frequency criteria we used for two blog types are different (0.5 for the first cluster and 2 for the second cluster).

We then selected nine as the optimum number of clusters. The visualized results shown that there are nine clusters in the dataset.

Chart, scatter chart

Description automatically generated

Table 1. Visualization of the *k*-Means results

After *k*-means, we conducted spectral clustering using existing libraries. Spectral clustering algorithm is a group of methods that are effective in high dimensional data applications. Through using affinity matrix, spectral clustering could construct new dimensions instead of facing a direct challenge to the high-dimensional data. It performs well particularly in the case of non-Gaussian clusters where K-Means fail to give good results.  We selected nine as the value of k since it maximized the eigengap and produce better spectral clustering results. The visualization shown that some points are clustered differently with *k*-means.

Chart, scatter chart

Description automatically generated

Table 2. Visualization of the Spectral Clustering results

At last, we conducted DBSCAN clustering. DBSCAN Clustering is a density-based clustering non-parametric algorithm which grows clusters according to the density of neighborhood objects. The DBSCAN algorithm basically requires 2 parameters: eps and minPoints. Based on the distance of the dataset, we choose 0.8 as our eps value. Meanwhile, we choose 100 as our minPoints value since we have a large dataset. Results of the DBSCAN method is shown below:

Chart, scatter chart

Description automatically generated

Table 3. Visualization of DBSCAN results

Comparing the visualized results of three clustering methods, we think k-Means produce the most high-quality clusters.

**Why do we select those methods?**

We choose *k*-Means, DBSCAN and spectral clustering because they perform well on large/medium n\_samples and medium/small n\_clusters. Based on the nature of our dataset and results of the first clustering method, we know that we have a relatively large n\_samples and a rather small n\_clusters.

We choose *k*-Means because it guarantees convergence and is relatively simple to implement and. It also generalizes to clusters of different shapes and sizes. The k-means method has its disadvantages. For instance, we need to manually specify *k* in advance. Besides, in *k*-Means, it is not guaranteed to converge to the global optimum and often terminates at a local optimum. Meanwhile, the k-means method is not suitable for discovering clusters of very different size. In addition, centroids can be dragged by outliers in *k*-Means. A small number of outliers can substantially influence the mean value and get their own cluster instead of being ignored. Moreover, the k-means method does not work well with high dimensional data. As the number of dimensions increase, k-means, as a distance-based algorithm, converges to a constant value between any given examples.

Therefore, to deal with the outlier problem we choose DBSCAN and spectral clustering as our second and third clustering method. The DBSCAN algorithm has a notion of noise and is robust to outliers. Meanwhile, unlike k-means, which assumes that clusters are convex shaped, DBSCAN could discover clusters with nonconvex shapes. To deal with the curse of dimensionality, we choose spectral clustering as our third clustering method. Like DBSCAN, spectral clustering produce clusters that do not follow a fixed shape or pattern, therefore could be used for data of different shapes and sizes. In addition, spectral clustering added a pre-clustering step to the algorithm, thus is a great method to cluster high-dimensional data.