**CSE 5243 Lab#3 Report**

by Xuyang Zhang and Xiang Li

***1. Introduction***

**1.1 Data Extraction:**

The raw data are included inside twenty-two .sgm files, each of which contains nearly a thousand articles. However, only articles with body part are considered as a valid article. Furthermore, since we perform the clustering on both ‘topics’ and ‘places’, only the article with non-empty entry for the corresponding ‘topics’ or ‘places’ is counted. According to our results, the number of valid articles with non-empty entry for ‘places’ label is 15098 while the number of valid articles with non-empty entry for ‘topics’ label is 5928. The frequency of each meaningful word is stored in a ‘Trie’ data structure, which is later used to produce features.

**1.2 Feature:**

In consistence with the previous Lab, TF-idf was used as the word frequency metric. Two distinct feature maps were constructed for comparison.

*Feature1:* the words used for TF-idf are first extracted from top 5 most frequent words in each article, however, this time the TF-idf will always be updated by a larger value. With these words sorted by TF-idf values in non-ascending order, we choose 20% of this whole feature set.

*Feature2:* words used for TF-idf were from the collection of the top 5 frequent words from each document including body and title.

**1.3 Cluster Models:**

*a. K-means:*

K-means model from the package sklearn.cluster is used. K-means is a simple and fast unsupervised clustering algorithm, which classifies the given dataset through a certain number of clusters. It works for high dimensional features. However, K-means has several inherent disadvantages. The clustering results could be affected by the initialization of initial centroids. Moreover, K-means is not resistant to noise. Our utilization of the implemented K-means includes the initial parameter “kmeans ++” which can help with the initialization issue. This parameter can initialize the centroids to be (generally) distant from each other, leading to provably better results than random initialization [1].

*b. DBSCAN:*

Density-based spatial clustering of applications with noise (DBSCAN) locates regions of high density that are separated from one another by regions of low density. It is resistant to noises and can handle clusters of various shapes and sizes compared with K-means. However, the performance of DBSCAN might be affected when the input data is in high dimensionality [2].

***2. Results and Discussion:***

For each of the two selected clusters, we perform the clustering on both feature1 and feature2, with the corresponding feature array as the input. The information gain as reflected from the entropy difference is used to evaluate the performance of the clustering. The reference entropy is calculated before clustering. The results for ‘topics’ and ‘places’ are shown in separate figures. A larger value of the information gain yields better clustering performance. For DBSCAN cluster, we test two parameters: EPSILON=[0.1, 0.3, 1] (radius of a cluster) and minpts=[6, 10] (threshold of data counts). For the K-means cluster, the number of cluster is tested (K = [6, 10]).

*2.1. DBSCAN Cluster*

Figure Summary of the places DBSCAN Cluster information gain.

Above is the observation for ‘places’ label. Generally, DBSCAN clustering has more obvious effect for feature2 than for feature1 since feature1 contains much more information. Moreover, a smaller value of the minpts yields a greater information gain. That is because it is hard for a cluster to develop with a high threshold number. We can expect that for minpts=10, each cluster will have smaller number of data than those in minpts=6. In consequence, many data points will be left as noise yielding less information gain. As for the radius of the cluster, feature1 with minpts=10 has the best clustering performance with radius=0.1. For all the other cases, we see a radius of 0.3 yields the best performance. And a value of radius=1 results in a trivial information gain in all cases.

Figure Summary of the topics DBSCAN Cluster information gain.

Comparing Figure2 with Figure1, we see that ‘topics’ articles always yield better DBSCAN clustering performance than ‘places’ articles under the same parameter settings. That maybe because topics of articles always tend to be unique, however, the places of different articles may have some intersection.

*2.2 K-means Cluster*

Figure Summary of the places K-means Cluster information gain.

As shown above, for the places label articles K-means clustering, a larger number of clusters yields better clustering performance. That may be because the feature array for places is large enough so that more clusters will gain more information.

Figure Summary of the topics K-means Cluster information gain.

As can be seen, the topics label shows a decreasing clustering performance for Feature1. That may be because the number of raw data of articles with non-empty ‘topics’ label is smaller than that of ‘places’ label. And feature1 generates smaller size of the feature array. Therefore, Feature1 array for topics does not need that many clusters.

Comparing Figure2 with Figure1, we see that ‘topics’ articles always yield better K-means clustering performance than ‘places’ articles under the same parameter settings.

Reference Links:

[1]: <https://scikit-learn.org/stable/modules/clustering.html#k-means>

[2]:

Pang-Ning Tan, Michael Steinbach, Vipin Kumar,. (2017). *Introduction to Data Mining.* Pearson.