CSE5243 Assignment 4

Man Cao(cao.235), Lilong Jiang(jiang.573)

November 11, 2013

1 Work Separation

Lilong mainly worked on hierarchical clustering. Man mainly worked on K-means. In fact there were a lot of overlapping during the work, we exchanged various ideas and wrote the this report together.

2 Input

After eliminating documents without topics, 11367 documents are left.
The input file has the following format:

```
{'NEWID':<value>, 'TOPICS':[value1, value2, ...], 'PLACES':[value1, value2, ...]}
{<term1>:<value1>, <term1>:<value2>, ...}
```

Note that each document corresponds to two lines: the first line contains the metadata of the document, the second line is the frequency vector.

3 Metrics

Cosine similarity and Jaccard similarity are used in this assignment.

4 Algorithms

4.1 Hierarchical Clustering

4.1.1 Single Link

We use the single-link to define the inter-cluster similarity. In every iteration, the two clusters with minimum distance will be merged.

4.1.2 Implemtation Details

Since the similarity between cluster1 and cluster2 is the same as the similarity between cluster2 and cluster1. We only need to store the *upper triangle of the matrix*.
Also considering the expensive time complexity of finding the min distance in the matrix, we

transform the proximity matrix to a proximity list and *sort* the list before the clustering. In this way, we only check the distance from the last position. For every document, we records which cluster it belongs to and for every cluster, we record which documents are in it.
The sorting time for the proximity list is $O(N^2 logN)$ since there exist $O(\frac{N^2}{2})$elements in the list. There are N iterations in total and in every iterator, we need to find the next minimum distance and update the mapping relationship between the documents and clusters. The time complexity of finding the next minimum distance and updating the mapping relationship in N iterations is $O(N^2)$, so the total time complexity is $O(N^2 logN)$.

4.2 K-means Clustering

We implement the naïve K-means algorithm as described in the slides from class:

1. Randomly select K distinct documents as the initial centroids of K clusters.
2. For each document, compute similarities between the document and each of the K centroids; assign the document to the cluster whose centroid has largest similarity with the document.
3. For each cluster, recompute its centroid, which is the mean of all documents in the cluster; also compute the distance between the new centroid and old centroid.
4. If, for any cluster, the distance between the new centroid and old centroid is greater than thresold value, repeat from step 2.

The distance between two documents d1 and d2 is simply $1.0 - similarity(d1, d2)$. In our experiment, the threshold value is 0.001.

Interestingly, we observed that it is possible that K-means does not converge. In one case, we ran K-means for nearly 40 hours with K=16 and cosine similarity, and it did not give an output. Then we ran it again with the same configuration, and it just finished in 35 minutes.

We found proof states that for large scale of data, it is quite possible for K-means to oscillate between two or more partitions and never converge [1] [2]. One solution is to compute the variance of Sum of Squared Error (SSE) every time the algorithm tries to move an element to a different cluster, considering the slight change of the centroids of the two involved clusters. If such movement can reduce the SSE, then move it; otherwise do not move the element. We did not implement this approach, because it should be much slower, and we do not see the oscillation case very often.

5 Evaluation

5.1 Scalability

The time for clustering 2, 4, 8, 16 and 32 clusters with Hierarchical Clustering and K-means is show in Fig.

[1] `http://www.clustan.com/k-means_critique.html#FailureToConverge`
[2] "Some methods for classification and analysis of multivariate observations", MacQueen, Berkeley Symposium, 1967, p. 288

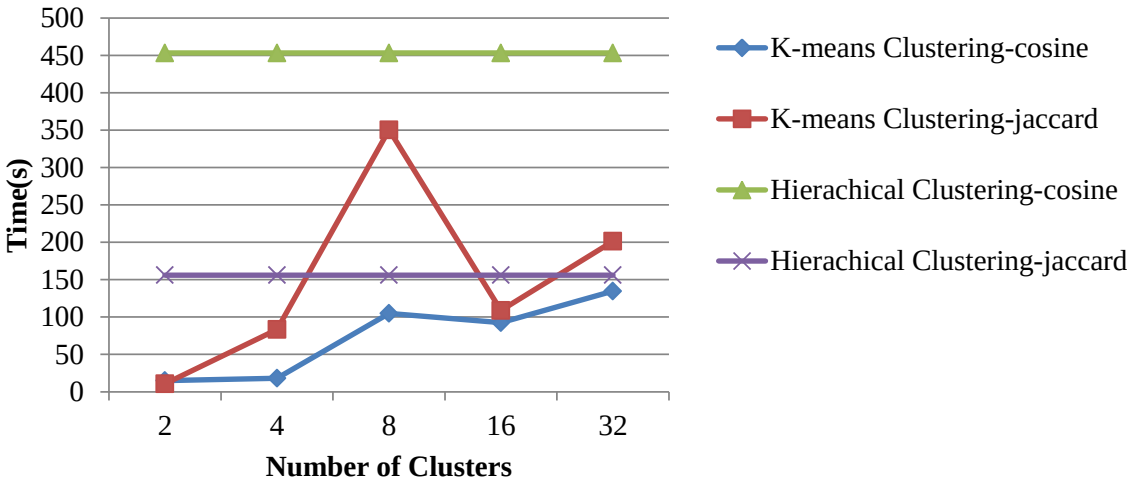

Figure 1: The average precision in percentage for each classifier over 5-fold cross validation, for two M values.

5.2 Quality

5.2.1 Entropy

For a document with multiple topics, each topic receives a *vote* of $1/n$, where n is the number of topics in the document. The probability of a topic t within a cluster C_x is then:

$$P(t|C_x) = \frac{\sum_{docs} v_{it}}{\sum_{docs} \sum_{topics} v_{ij}} \tag{1}$$

where v_{ij} is the vote of topic j from document i.

In order to make sure the entropy falls within the range of $[0, 1.0]$, we use m as the base of the logarithm, where m is the number of distinct topics in the cluster. That is:

$$m = |C_x.topics| \tag{2}$$

The entropy of the clustering is the sum of weighted entropy of each cluster.

5.2.2 Skew

The skew is measured as variance of the cardinalities of different clusters.

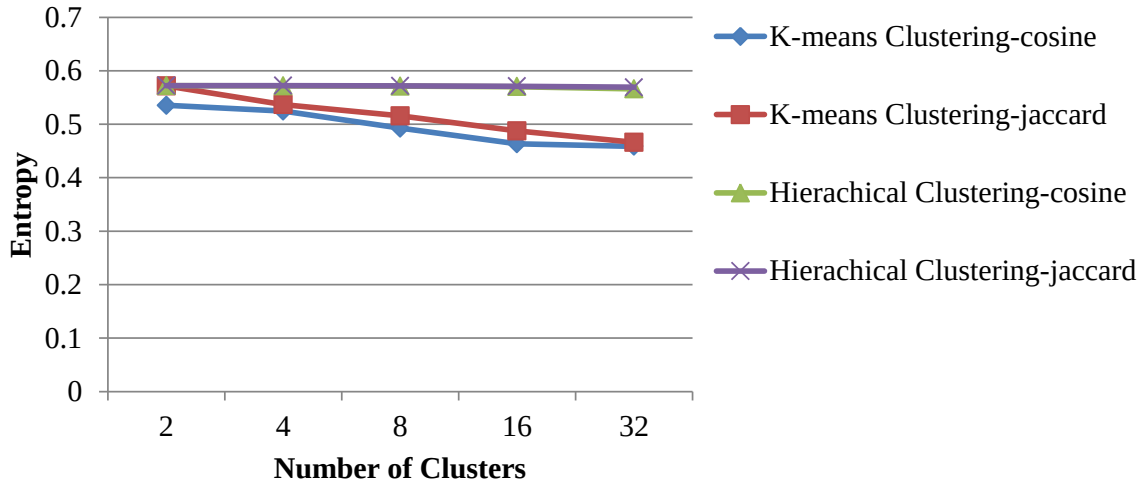

Figure 2: The average precision in percentage for each classifier over 5-fold cross validation, for two M values.

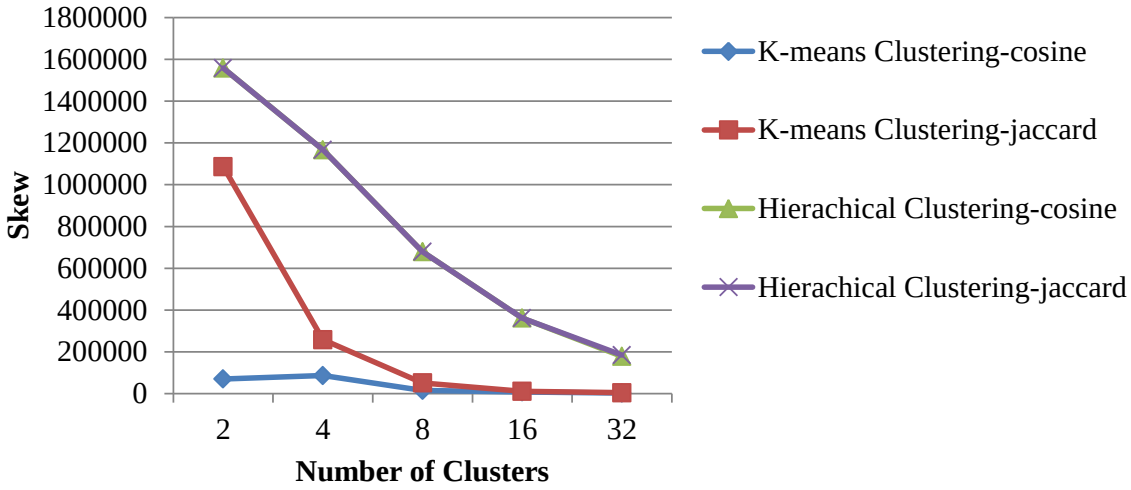

Figure 3: The average precision in percentage for each classifier over 5-fold cross validation, for two M values.