Using Clustering for Community Search

Jianyu Jiang (3030044036)

November 29, 2017

1 Problem Definition

In this assignment, you are required to implement and benchmark community search algorithm using clustering. The algorithms should first compute user similarity by vertex similarity and personal page-rank and should perform a K-means clustering.

2 Design and Implementation

The algorithm has three steps. First, we compute the similarity matrix by vertex similarity or personalized PageRank. Then, we perform a K-Means algorithm with this similarity matrix. Then, we compare the K-Means results.

We compare the result with three models: Purity, Entropy and Normalized mutual information (NMI).

If $W = w_1, w_2, ..., w_k$ is the set of clusters and $C = c_1, c_2, ..., c_j$ is the set classes. Then,

$$purity(W, C) = \frac{1}{N} \sum_k \max_j w_j \bigcap c_j \tag{1}$$

. a perfect clustering has a purity of 1.

For entropy,

$$H(W) = -\sum_k P(w_k) log P(w_k) = -\sum_k \frac{|w_k|}{N} log \frac{|w_k|}{N} \tag{2}$$

The minimum of $H(W)$ is 0 if the clustering is random with respect to class membership. In that case, knowing that a document is in a particular cluster does not give us any new information about what its class might be.

For NMI, it is always a value between 0 to 1.

2.1 Design and Implementation

We implemented the algorithm using python. We implemented the personalized PageRank ourselves and use KMeans implementation from sklearn.

2.2 Benchmark

Evaluation of the algorithm is focused of Purity, Entropy and NMI of using different similarity matrix and size of clusters. Table 1 shows the result.

k	**Purity-PR**	**Purity**	**Entropy-PR**	**Entropy**	**NMI-PR**	**NMI**
2	0.5459	0.5754	0.01233	0.7395	0.02577	0.0123
4	0.5470	0.5940	0.0257	0.8900	0.00488	0.0286
8	0.5481	0.5946	0.4995	0.9424	0.00545	0.02910
16	0.5765	0.5962	1.042	1.0290	0.01531	0.03244

Table 1: Evaluation result.