The Comparison of Top Leaders Algorithm and Other Algorithms

Final Report of Project in CS 6350 Machine Learning

Name: Pingfan Tang, ID: 00921446

1 Introduction

In this project I explored the Top Leaders algorithm [1], and compared it with several other community discovery algorithms. Community discovery is an important and interesting research field in the analytics of social network. By detecting communities in a social network, companies can adopt different marketing strategies and recommend different products for people in different communities, or provide personalized service for them. Thus, companies can make more profit, so community discovery is an interesting and practical research subject.

The Top Leaders algorithm is inspired by K-means algorithm. According to the knowledge we learned in the class, the essence of K-means clustering is a hard EM for Gaussian mixture model. So, in essence Top Leader algorithm is a kind of unsupervised learning algorithm.

Through this project, I learned Top Leaders algorithm is an effective method for community discovery, but its performance is usually inferior to that of spectral clustering algorithm, Girvan and Newman's divisive algorithm and Newman's greedy optimization of modularity algorithm. Moreover, the initial selection of leaders has an important influence on the final result. These will be shown and discussed in Section 4 and Section 5.

2 Top Leaders Algorithm

A social network can be represented by an undirected graph, and to discover communities in a network means to cluster vertices in a graph [4]. The Top Leaders algorithm is very similar to K-means algorithm. It first finds promising leader nodes in the given network, then iteratively updates communities and their corresponding leaders until there is no change in the communities. This process is shown in Algorithm 2.1.

Algorithm 2.1 Top Leaders Algorithm

Input: A social network G, and k, the desired number of communities.
Initialize a set *leaders* which contains k leaders.
repeat
  for all node $n \notin G$ **do**
    if $n \in leaders$ **then**
      use Algorithm 2.2 to associate n to a leader
  for all $l \in leader$ **do**
    $l = \arg\max_{n \in Community(l)} \text{Centrality}(n)$
until There is no change in the leaders.

In Algorithm 2.1, the "Centrality" of a node n in community C of size N is defined as $\text{Centrality}(n) = \frac{\deg(n,C)}{N-1}$ where deg(n,C) is the number of edges in C incident upon n.

The paper [1] gives four methods to initialize k leaders: "Naïve Initialization", "Top Global Leaders", "Top Leaders & not Direct Neighbour" and "Top Leaders & Few Neighbours in Common". According to this paper, "Top Leaders & Few Neighbours in Common" works best. The computation is using the most central node (node with largest degree) as the first leader, and then add the next central one to the current set of leaders if it has less than N_0 common neighbours with each leader in the current set.

After initialization, to associate each node to a specific leader, the Top Leaders algorithm calls Algorithm 2.2, where $\mathcal{N}(n, d)$ denotes the set of nodes in the neighbourhood depth d of node n.

Algorithm 2.2 Associate node n to its leader

Input: A social network G, node n, *leader* (a set of k leaders), and δ, γ
$depth = 1$
$CandList = leaders$
repeat
  $A = \{c \in CandList | \ \mathcal{N}(n, depth) \cap \mathcal{N}(c, 1) \geq \gamma\}$
  $CandList = \arg\max_{c \in A} |\mathcal{N}(n, depth) \cap \mathcal{N}(c, 1)|$
  $depth = depth + 1$
until $|CandList| \leq 1$ or $depth > \delta$
if $|CandList| = 0$ **then**
  associate n as an outlier
else if $|CandList| > 1$ **then**
  associate n as a hub
else
  associate n to $CandList$

3 Three Other Algorithms

In addition to the Top Leaders algorithm, I also explored the following three classic community discovery algorithms.

3.1 Spectral clustering algorithm

Spectral methods [3] generally refer to the following algorithm. If A is the adjacency matrix of a network, and D is a diagonal matrix with the degrees of the nodes along the diagonal, the Laplacian (or the normalized Laplacian) L is given by $L = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$. The eigenvectors corresponding to the two smallest non-zero eigenvalue of L eigenvectors define an embedding of the nodes of the network as points in a 2-dimensional space. So, we can use classical data clustering techniques K-means algorithm to derive the assignment of nodes to clusters.

3.2 Girvan and Newman's divisive algorithm

Newman and Girvan [5] proposed a divisive algorithm for community discovery, using ideas of edge betweenness. The betweenness of an edge e is defined as follows

$$\text{betw}(e) = \text{fration of shortest paths that use edge } e.$$

According to the above definition, edges with high betweenness scores are more likely to be the edges that connect different communities. That is, inter-community edges are designed to have higher edge betweenness scores than intra-community edges do. Therefore, by identifying and discarding such edges with high betweenness scores, one can disconnect the social network into its constituent communities. The general form of their algorithm is as follows:

Algorithm 3.1 Girvan and Newman's divisive algorithm

Input: A social network G, and k (the desired number of communities)
repeat
  Compute betweenness score for all edges in G
  Find the edge with the highest score and remove it from G
until There are k communities in G

3.3 Newman's greedy optimization of modularity algorithm

Newman [6] proposed a greedy agglomerative clustering algorithm for optimizing modularity. Modularity [2] is used to measure the goodness of a clustering of a graph, and higher modularity means better clustering. Suppose (V, E) is an undirected graph where $V = \{v_1, v_2, \cdots, v_n\}$ and $|E| = m$, and $A = (a_{ij})$ is the adjacency matrix of this graph . If V is divided into k clusters $\{V_1, V_2, \cdots, V_k\}$, then the modularity of this clustering is

$$Q = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} - \frac{d_i d_j}{2m}) \delta(v_i, v_j) \tag{3.1}$$

where d_i is the degree of v_i, d_j is the degree of v_j, and if v_i, v_j are in the same cluster then $\delta(v_i, v_j) = 1$ else $\delta(v_i, v_j) = 0$.

The basic idea of the algorithm is that at each stage, groups of vertices are successively merged to form larger communities such that the modularity of the resulting division of the network increases after each merge. At the start, each node in the network is in its own community, and at each step one chooses the two communities whose merger leads to the biggest increase in the modularity. We only need to consider those communities which share at least one edge, since merging communities which do not share any edges cannot result in an increase in modularity.

4 Experiment Result

I compare Top Leaders algorithm and three other algorithms on the following data sets: "Dolphin social network", "Zachary's karate club", "Books about US politics", where "Zachary's karate club" has been explored in [1], but the other two sets have not been tested for Top Leaders algorithm. To measure the performance of different algorithms, I use modularity and normalized cut as quality functions. Modularity is given by (3.1), and for a graph with vertices $V = \{v_1, v_2, \cdots, v_n\}$ and k communities $\{C_1, C_2, \cdots, C_k\}$, normalized cut is defined as [7, 8]

$$\text{Ncut} = \sum_{l=1}^{k} \left(\frac{\sum_{v_i \in C_l, v_j \notin C_l} A(i,j)}{\sum_{v_i \in C_l} \text{degree}(v_i)} + \frac{\sum_{v_i \in C_l, v_j \notin C_l} A(i,j)}{\sum_{v_i \notin C_l} \text{degree}(v_i)} \right)$$

where A is the adjacency matrix of this graph. The normalized cut of a community C_l is the sum of weights of the edges that connect C_l to the rest of the graph, normalized by the total edge weights of C_l

and that of the rest of the graph. Communities with low normalized cut are good communities, as they are well connected amongst themselves but are sparsely connected to the rest of the graph.

For all three data sets, I choose $k = 2$, $\delta = 10$ and $\gamma = 0$ in Algorithm 2.2, and choose the same initial nodes in Top Leaders algorithm and spectral clustering algorithm.

For "Dolphin social network" (**Dolphin**) I choose $N_0 = 1$ while using "Top leaders & Few Neighbours in Common" to obtain the initial leaders, but for "Zachary's karate club" (**Karate**) and "Books about US politics" (**Polbooks**), I choose $N_0 = 5$ in initialization. I use Matlab to implement these four algorithms, and use Python to obtain the adjacency matrix from gml file and visualize the result. For different data sets, the performance of Top Leaders algorithm (**TP**), spectral clustering algorithm (**SC**), Girvan and Newman's divisive algorithm (**GND**) and Newman's greedy optimization of modularity algorithm (**NGOM**) is shown in Table 4.1.

Table 4.1: The running result of four algorithms on three data sets.

		TL	SC	GND	NGOM
Dolphin	Modularity	0.3722	0.3787	0.3787	0.3854
	Ncut	0.2640	0.1812	0.1812	0.2309
Karate	Modularity	0.3715	0.3715	0.3582	0.3718
	Ncut	0.5132	0.5132	0.5649	0.5128
Polbooks	Modularity	0.4454	0.4424	0.4569	0.4472
	Ncut	0.2178	0.1860	0.1723	0.2089

The running results of Top Leaders algorithm on "Dolphin social network", "Zachary's karate club" and "Books about US politics", are shown in Figure 4.1, Figure 4.2 and Figure 4.3 respectively, where the yellow nodes are the final leaders of communities.

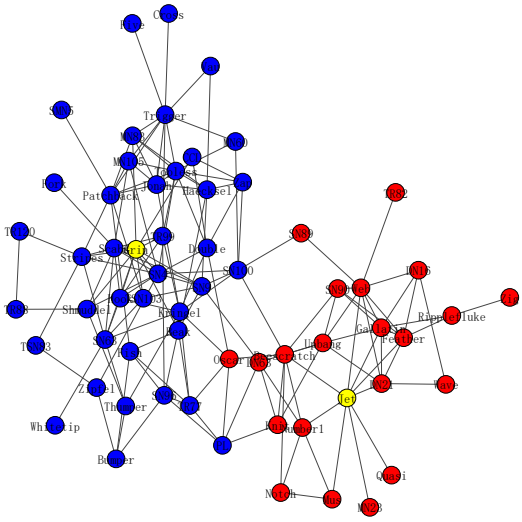

Figure 4.1: The result of Top Leaders algorithm ($N_0 = 1$) on Dolphin social network.

Due to the limit of the length of this report, we cannot list the pictorial results of three other algorithms, and they are similar to that of Top Leaders algorithm. Through the experiment, we found the result of Top Leaders algorithm is sensitive to the selection of initial leaders. For example, for the data set "Zachary's karate club", if we choose $N_0 = 1$ in the initialization of Top Leaders algorithm, the running result is

Modularity=0.0939, Ncut=1.3437, which is worse than the case $N_0 = 5$. The picture of this result is shown in Figure 4.4.

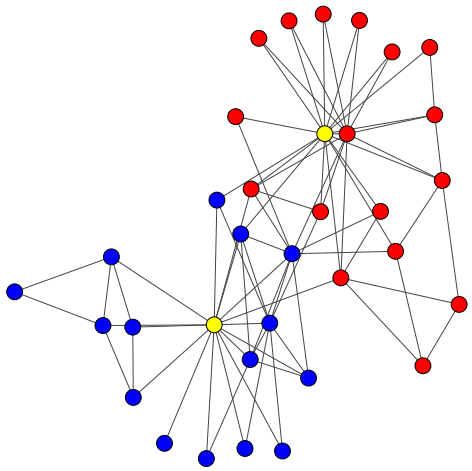

Figure 4.2: The result of Top Leaders algorithm ($N_0 = 5$) on Zachary's karate club.

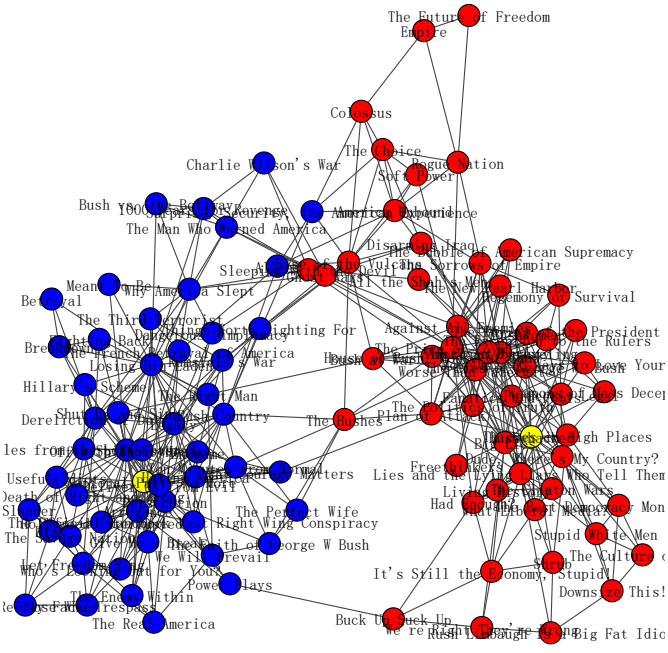

Figure 4.3: The result of Top Leaders algorithm ($N_0 = 5$) on Books about US politics.

5 Conclusion and Future Directions

From Table 4.1, Figure 4.1, Figure 4.2 and Figure 4.3, we can see Top Leaders algorithm is an effective method for community discovery, and we obtained the same result on "Zachary's karate club" as in [1]. In paper [1], the authors use "Purity" and "Adjusted Rand Index" as measure to show that their algorithm works better than some other algorithms, but these two evaluation metrics are based on the ground truth

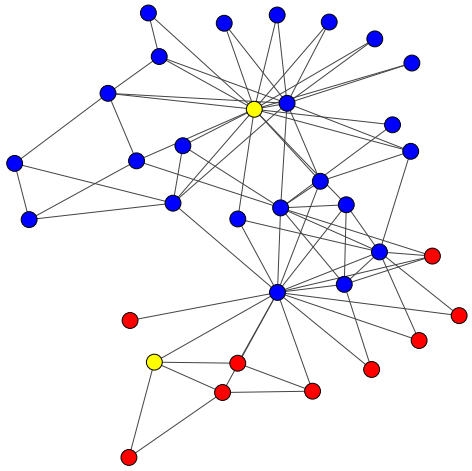

Figure 4.4: The result of Top Leaders algorithm ($N_0 = 1$) on Zachary's karate club.

which is usually unknown. In our project, we use "Modularity" and "Normalized Cut" to compare the performance of different algorithms. Under these two quality functions, Top Leaders algorithm usually cannot work better than SC, GND or NGOM, although it can produce a fine and reasonable result.

Moreover,through experiment we find Top Leaders algorithm is sensitive to the choice of the initial leaders. (GND and NGOM need not initialization or the choice of parameters.) In some special case as shown in Figure 4.4, a node connected to one leader may be associated to another leader, and a community may be not a connected component of the graph. Obviously, this is not a desired result. The main reason for this phenomenon is that the Top Leaders algorithm uses $|\mathcal{N}(n, depth) \cap \mathcal{N}(c, 1)|$ to measure the distance between the node n and the leader c. So, if **I had much more time for this project**, I will try to find another method to measure the distance between a node and a leader, for example using the length of the shortest path between nodes. I will also try other methods to choose initial leaders to make the algorithm more robust to the initialization and avoid the case in Figure 4.4.

References

[1] R. Rabbany, J. Chen, and O. R. Zaiane, *Top leaders community detection approach in information networks.* in SNA-KDD Workshop on Social Network Mining and Analysis, 2010.

[2] M.E.J.Newman and M.Girvan, *Finding and evaluating community structure in networks.* Phys. Rev. E, 69(2):026113, Feb 2004.

[3] U. Von Luxburg, *A tutorial on spectral clustering.* Statistics and Computing, 17(4):395–416, 2007.

[4] Charu C. Aggarwal, *Social Network Data Analytics.* Springer, New York, 2011.

[5] M.E.J.Newman and M.Girvan, *Finding and evaluating community structure in networks.* Phys. Rev. E, 69(2):026113, Feb 2004.

[6] M. E. J. Newman, *Fast algorithm for detecting community structure in networks.* Physical Review E, 69(6):066133, 2004.

[7] J. Shi and J. Malik, *Normalized Cuts and Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

[8] R.V. Solé and M. Montoya, *Complexity and fragility in ecological networks.* Proceedings of the Royal Society of London. Series B: Biological Sciences, 268(1480):2039, 2001.