Social and Information Network Analysis : Final Project
Movie Rating Prediction

[1]Sung, Cheng-en [2]Huang, Hsiao-Ting
[1]*Department of Electrical Engineering National Cheng Kung University*
[2]*Department of Electrical Engineering National Cheng Kung University*
[1]*iamikari0383@gmail.com*, [2]*t654321ina@gmail.com*

Keywords: Community Detection

Abstract: The modern science of networks has brought significant advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure, or clustering, i.e., the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. Detecting communities is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs.

1 DESCRIPTION

Methods of detecting network communities: In this homework, we use three methods to detecting network communities. The three alogrithms are Louvian Algorithm, Label Propagation Algorithm and Speaker-Listener Label Propagation Algorithm. And in the zip file would include Louvian's code and SLPA's code.

1.1 Method 1: *Louvian Algorithm*

This algorithm try to optimized modularity value, which defined as a value between -1 and 1 that measures the density of links inside communities compared to links between communities. For a weighted graph modularity defined as:

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j) \tag{1}$$

Where

- A_{ij}: edges between i and j
- k_i & k_j are the sum of the weights of the edges attached to nodes i and j, respectively
- m is the sum of all of the edge weights in the graph
- c_i and c_j are the communities of the nodes
- δ is a simple delta function

Louvian Algorithm in Three phases
Initialize: each node assigned to its own community.

1. Put each node i to its neighbor nodes j who has the most modularity gain. Modularity gain defined as:

$$\Delta Q = [\frac{\sum_{in}+k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m}^2)] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m}^2) - (\frac{\sum_{k_i}}{2m}^2)] \tag{2}$$

   where
   - $\sum_{im}$ is sum of all the weights of the links inside the community i is moving into
   - $\sum_{tot}$ is the sum of all the weights of the links to nodes in the community
   - k_i is the weighted degree of i
   - $k_{i,in}$ is the sum of the weights of the links between i and other nodes in the community
   - and m is the sum of the weights of all links in the network.
2. In the second phase of the algorithm, it groups all of the nodes in the same community and builds a new network where nodes are the communities from the previous phase. Once the new network is created, the second phase has ended and the first phase can be re-applied to the new network.
3. Output each node belongs to which communities.

1.2 Method 2: *Label Propagation Algorithm*

Label propagation is an algorithm for finding communities. Label propagation has advantages in its

running time and amount of a priori information needed about the network structure (no network structure parameter is required to be known beforehand including number of communities at the end of this algorithm).

LPA in Four phases
Initialize: each node assigned to its own community. (i.e., For a given node x, $C_x^{(0)} = x$)

1. Arrange the nodes in the network in a random order and set it to X
2. For each $x \in X$ chosen in that specific order, let $C_x^{(t)} = \mathcal{M}(C_{x_{i1}}^{(t-1)}, \ldots, C_{x_{im}}^{(t-1)}, C_{x_{i(m+1)}}^{(t-1)}, \ldots, C_{x_{ik}}^{(t-1)})$. $\mathcal{M}$ here returns the label occurring with the highest frequency among neighbours.
3. Until iteration equal to MAX_ITERATION which is user define (e.x., MAX_ITERATION = 20)
4. Output each node belongs to which communities.

1.3 Method 3: *Speaker-Listener Label Propagation Algorithm*

Speaker-Listener Label Propagation Algorithm (SLPA) extend Label Propagation Algorithm (LPA) by regard each node as belongs to overlapping communities. Input of this algorithm contain two parameter, first is the maximum number of iteration (MAX_ITERATION), second is the threshold (θ) that judge whether one node is belong to one community. Again SLAP do not need to know the number of communities at the end of the algorithm.

SLPA in Five phases
Initialize: each node assigned to its own community. (i.e., For a given node x, $C_x^{(0)} = x$)

1. Arrange the nodes in the network in a random order and set it to X
2. For each $x \in X$ chosen in that specific order, let $C_x^{(t)} = \mathcal{M}(C_{x_{i1}}^{(t-1)}, \ldots, C_{x_{im}}^{(t-1)}, C_{x_{i(m+1)}}^{(t-1)}, \ldots, C_{x_{ik}}^{(t-1)})$. $\mathcal{M}$ here returns the label occurring with the highest frequency among neighbours.
3. Store the most popular label to corresponding node's memory
4. Until iteration equal to MAX_ITERATION which is user define (e.x., MAX_ITERATION = 20)
5. Output each node belongs to which communities.

We implement this algorithm and get highest score within all our team submission record (i.e., **NMI = 1.0000** and **ANC = 1.0000**)

2 ANALYSIS

Our team consider that the best method of this homework depends on lots of facters. Not only the method we use but also the parameters we changed also the performance evaluating way ... ,ect. Therefore, we will first analysis the results by the NMI and ANC performnce evaluations. Second, by the score we get from the race. Finally, the parameters we had changed to optimize our method.

2.1 The Performance Evaluation

This race take advantage of two commonly used evaluation metrics of clustering: one is Normalized Mutual Information (NMI) (3), and the other is Accuracy in the Number of Communities (ANC) (4).

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]} \tag{3}$$

Where:

- Y: Class label
- C: Cluster label
- $H(.)$: Entropy
- $I(Y,C)$: Mutual information between Y and C

$$ANC = 1 - \frac{||C^*| - |\hat{C}||}{2|C^*|} \tag{4}$$

Where:

- C^*: Set of ground truth communities
- $\hat{C}$: Set of dected communities

Obviously, after we knew the rules in the evaluation NMI and ANC, we can see that the score of ANC is more esier to get higher in the most of the time. Therefore, we are dedicated to get higher NMI to be our goal.
We would compare those method we had used in the following parts:

2.2 Compare Louvian, LPA and SLPA Algorithm

Part 1. the scores from the race
The following are the highest score of the three methods we get from the race:

Method	NMI	ANC
LPA	0.6125	0
Louvian	0.9177	0.7614
SLPA	1.0000	1.0000

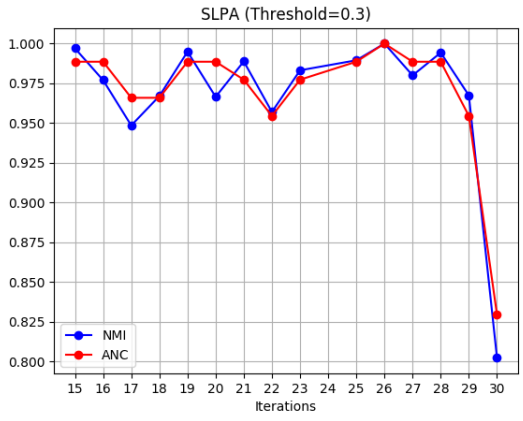
Part 2. Discussion:
LPA is the first method that we tried. And just like above's score, the accuracy is dramatically worse; after discussion, we found that there might be tow reasons that we get the lower score: First is that we have some bug in our code; secound is that as we know LPA is really unstable . So that might be a reason too.

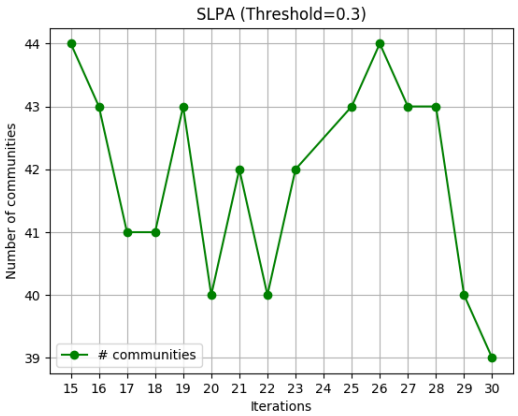
While we thinking about what's wrong in our LPA method, we try the Louvian method which also is compared in the race. And since it is the optimization of Modularity as the algorithm progresses. It obviously raised the accuracy of the result.

Finally, after searching lots of papers and algorithum, we found that SLPA might be a best solusiton to optimized the LPA Algorithm. When in the LPA Algorithm, each node had only one tag. However, in SLPA, this method set a list for each node to store historical tags. The tags updated each iteration were stored. By this way, we can get a better accuracy.

2.3 Compare SLPA Algorithm with different parameters

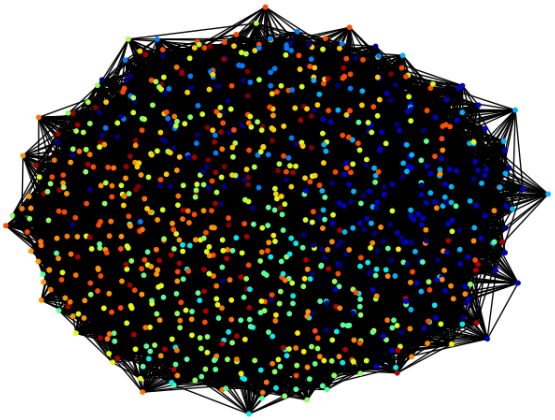
After we successfully running SLPA Algorithm. We started to change the two parameters of this method: One is iterations, the other is threshold. And we found that when the iterations= 26 and threshold= 0.3 will have the highest accuracy(NMI = 1.0000 ANC = 1.0000). Beside, we found that if the number of iteration higher than 28, NMI and ANC decrease dramatically (ex: iteration = 30, NMI = 0.8023, ANC = 0.8295)



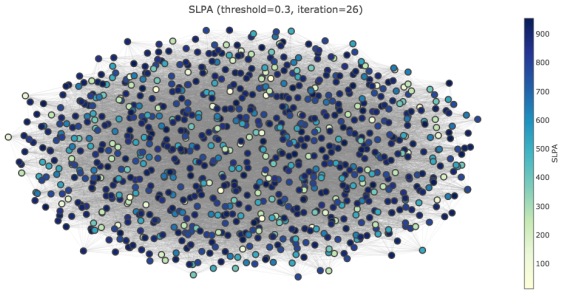


3 VISUALIZATION

3.1 Picture of Louvian Algorithm



3.2 Picture of SLPA Algorithm



4 REFERENCE

[1] Blondel et al., 2008 V.D. Blondel, J.-L. Guillaume, R. Lambiotte. Fast unfolding of communities in large networks J. Stat. Mech., 2008 (2008), p. P10008

[2] Jierui Xie , Boleslaw K. Szymanski , Xiaoming Liu, SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process, Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, p.344-349, December 11-11, 2011