Community Detection on 2-Hop Topology

HU, Pili and LI, Yichao

December 13, 2011

Abstract

Community detection in social networks has drawn great interest in recent decades, especially inspired by the emerge of wide online social network sites. In this paper, we study the two-hop topology from several observers in the "renren.com" network. Some important statistics are calculated first, along with some visualization. Then we compute mining(results) —**NEED-REVIEW-HERE**—

This document serves as the report of CSCI5180 course project[4, 5]. For more information, codes, and data, please refer to our open source repository[1].

Contents

1 Introduction **3**

2 Data Preparation and Preprocessing **3**

2.1 Data Source 3

3 Statistics **4**

4 Feature Selection **4**

4.1 Random Walk Based Techniques 4

4.2 Simple Proximity Measures 5

5 Single Feature Evalutaion **6**

5.1 Quality Functions 6

5.2 Receiver Operating Characteristics 7

6 Model Training and Testing **7**

7 Conclusion **7**

8 Future Work **7**

1 Introduction

Community detection has drawn great interest in the near decade. The explosion of online Social Network Sites make the study more practical but challenging. Traditional methodology aims at optimizing certain global metric. The resultant graph partition can thus be regarded as communities. The motivation of detecting communities on 2-hop topology comes in two folds:

1. **Data Availability**. The global topology of those SNS is kept confidential. However, users are allowed to view their buddies profile by default. This gives us the chance to construct a 2-hop subgraph from the whole graph. Take a step further, outsiders can always observe some forwarding sequences of tweet/status. This helps to reconstruct a subgraph more than 2-hop from the observer. The methodology developed here can thus be extended to such situations.

2. **Computation Availability**. Although those SNS providers have access to the global topology, it is computationally intractable to perform some well-developed global optimization. Local heuristics are widely used in many applications.

2 Data Preparation and Preprocessing

2.1 Data Source

The data source used in this paper is ”renren.com”, which is the most popular SNS in mainland China. At the mean while, ”renren.com” has the following good characteristics:

- Web UI of ”3g.renren.com” is very clean. Cawling through this interface doesn’t require any API calls.

- Renren is a real name dense graph connected mainly through real communities. Intuitively speaking, people who are close to the observer should be observed within this 2-hop topology with high probability.

- Institution label for all nodes in the observed subgraph is available on ”renren.com”. It makes model validation easier and automated. If the model is proven effective, we can adapt it to other SNS’s.

3 Statistics

4 Feature Selection

4.1 Random Walk Based Techniques

The rationale behind random walk is: the nearer a node is to observer, it can be reached with a higher probability by a random walker starting from observer. PageRank is one of the most classical work in this field.

We shortlist three variations[2] of the original PageRank, together with the explanation of using them in our context:

- PageRank with escape probability:

  $$\overrightarrow{v} = \alpha A^{\mathrm{T}} \overrightarrow{v} + (1-\alpha)\frac{\overrightarrow{1}}{n} \tag{1}$$

  where A is the transition matrix, typically the adjacency matrix, and α is the transfer ratio. $\overrightarrow{v}$ is the resultant PageRank value.

  In our graph, the link is very sparse for edge node, which represents the friend of observer's friend. They provide many dangling links, which influences the outcome of original PageRank algorithm. These nodes may appear as probability sinks. By introducing in escape probability, such sinks can be eliminated.

  Our graph is rooted and thus biased at the observer. So the rank output by this algorithm can be a proximity measure between other nodes and observer.

- Personalized PageRank:

  $$\overrightarrow{v} = \alpha A^{\mathrm{T}} \overrightarrow{v} + (1-\alpha)\frac{\overrightarrow{b}}{||\overrightarrow{b}||_1} \tag{2}$$

  The difference from eqn(1) is that, the all one's column vector $\overrightarrow{1}$ is substituted by a more general weight vector, also called the escape vector. This vector describes where and by what probability the random walker will escape to.

  With different escape vector $\overrightarrow{b}$, we can reach different goal:

  - Unsupervised learning. Denote the subscript of observer as o, then:

    $$b_i = \{ \begin{array}{cc} 1 & i = o \\ 0 & i \neq o \end{array} \tag{3}$$

    That means the random walker will always escape to the root node. Then the output can measure the proximity between root and other nodes.

- Semi-supervised learning. Denote the labeled set as L and their label as l_i. The personalized escape vector can be:

$$b_i = \{ \begin{array}{cc} l_i & i \in L \\ 0 & i \notin L \end{array} \tag{4}$$

Typically, the miner can choose a subset of the nodes, which are already known to share the same label with the observer. We assign those nodes in set L some positive weights. Then the random walker can escape to this set with different probability. The rantionale is, if one node is in our community, it is probable to be reached from some already known members.

4.2 Simple Proximity Measures

Yet random walk based techniques are good proximity measures in many context, there exists some simple but effective measures. We shortlist the following metrics, together with explanation of using them in our context:

- Common Neighbours:

$$Common(i, j) = |N(i) \cap N(j)| \tag{5}$$

It is straight forward that if node i shares more common neighbours with observer o, it is closer to observer.

- Adamic/Adar score:

$$Adamic/Adar(i, j) = \sum_{k \in N(i) \cap N(j)} \frac{1}{\log |N(k)| + 1} \tag{6}$$

Adamic/Adar can be seen as an extension of simple common neighbours, which take the neighbour's degree into consideration. The contribution from each common neighbour k is weighted as $\frac{1}{\log |N(k)|+1}$. Higher degree nodes may stant for public pages, or very well-known people in the network. Sharing such kind of common neighbour is a much weaker evidence that two people are in the same community. Thus the it is given a lower weight. The use of log scale stems from previous statistical studies, that node ranking tends to be Zipf distributed. [3]

- Jaccard's coefficient:

$$J(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \tag{7}$$

This coefficient can be seen as another extension of common neighbours. It is effective when the graph is sparse. The numerator calculates the common neighbours of two nodes. The denominator takes their own size into consideration. It's reasonable that two nodes with higher degree will have more common neighbours. The denominator act as a normalizer, which makes the output of different node pairs relative comparable.

5 Single Feature Evalutaion

5.1 Quality Functions

After we label the community of all nodes, we can calculate some quality functions to indicate how well the community is detected. Different researchers have different preference of quality functions, to name a few:

- Normalized cut:

$$Ncut(S) = \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\sum_{i\in S} d(i)} + \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\sum_{j\in \overline{S}} d(j)} \tag{8}$$

  where S is the set of one community, and $d(.)$ is the digree of a node, which is defined as $d(i) = \sum_j A_{ij}$.

  The smaller the value, the better community detection quality. Two numerators measure the number of inter community links, and the value is normalized by the number of intra community links.

- Conductance:

$$Conductance(S) = \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\min\{\sum_{i\in S} d(i), \sum_{j\in \overline{S}} d(j)\}} \tag{9}$$

  This quality function is positive related with eqn(8). When community sizes are highly skewed, conductance may provide better evaluation.

- Modularity:

$$Q = \sum_{i=1}^{K} \left[\frac{A(V_i, V_i)}{m} - \left(\frac{d(V_i)}{2m} \right)^2 \right] \tag{10}$$

  where

$$A(V_i, V_j) = \sum_{u\in V_i, v\in V_j} A_{uv} \tag{11}$$

  and

$$d(V_i) = \sum_{u\in V_i} d_u \tag{12}$$

  K is the number of communities, which is 2 in our case (we only distinguish whether the node is in the same community with observer or not).

The rationale behind modularity is the comparison with a random graph. The father resultant community is from a random graph, the better the quality. One of the advantages of modularity is that it is independent of the number of communities the graph is divided into[2].

Those quality functions are popular among different research groups. They also capture different characteristics of graphs. In this project, we'll check if these global metric can be used to evaluate our extreme local case.

5.2 Receiver Operating Characteristics

Draw the Receiver Operating Curve by varying proximity threshold. (Different TP and FP rate)

—NEED-REVIEW-HERE—

6 Model Training and Testing

7 Conclusion

8 Future Work

As is in our proposal [1], there are other metrics we want to try. Due to project time and report space, we only name a few in this section:

- Katz Score:

$$Katz(i,j) = \sum_{l=1}^{\infty} \beta^l A^l(i,j) \tag{13}$$

  The matrix series results in

$$Katz = (I - \beta A)^{-1} - I \tag{14}$$

  By tuning β, this score can measure number of paths between two nodes, with preferred length range.

- Simrank:

$$s(a,b) = \frac{\gamma}{|I(a)||I(b)|} \sum_{i \in I(a), j \in I(b)} s(i,j) \tag{15}$$

  This metric measures the expectation of γ^l, where l equals the time when two random walkers start from a and b meet. By proper matrix construction, Simrank can be solved as an eigenvalue problem.

For more information, interested readers are recommended to [2].

Acknowledgements

The authors would like to thank the following people for their contribution of personal data: Bo WANG, Hongshu LIAO, Huan CHEN, Shouxi LUO, Xiaoyu LUO, Xinyue ZHENG, Yan WANG, Yi LUO. The authors appreciate brain storm discussion with professor Wing Lau and Deyi Sun. The authors would like to thank course instructors and TAs of CSCI5180.

References

[1] Hu Pili and Li Yichao. Community detection on 2-hop topology. GitHub, https://github.com/Czlyc/2C-Web-Research, 12 2011. course project of CUHK/CSCI5180.

[2] C.C. Aggarwal. *Social network data analytics*. Springer-Verlag New York Inc, 2011.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[4] CSCI5180, Data Mining Lecture Notes, `http://www.cse.cuhk.edu.hk/~lwchan/teaching/csc5180.html`

[5] CSCI5180, Data Mining Tutorial Notes. (available in Moodle. Please contact the teacher/TA if you have interest)

[6] Receiver Operating Characteristics, `http://en.wikipedia.org/wiki/Receiver_operating_characteristic`

Appendix

Declaration

Meta tools like PageRank, Adamic/Adar, etc, are learned from the book *Social Network Data Analytics*. For original sources of these algorithms, please refer to the bibliography pages of that book.

The analysis/adaptation/extension/application of these tools and the use in community detection on 2-hop topology mentioned in this document is originality. Please refer to our open source repository for citation issues.