Community Detection on 2-Hop Topology (Proposal)

HU, Pili

December 12, 2011

1 Background

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 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 Objectives

- Output a simple crawler for "renren.com".
- Output visualization solutions.
- Output some important statistics.
- Adapt several algorithms in related field to the 2-hop scenario.

4 Study Outline

1. Visualization tools: NodeXL, R, Graphviz, etc. The challenge is how to position the data points on 2-D surface when only topology (rather than proximity measure) is known. The visualization is performed and improved at every stage of the study. Key observations from preliminary visualization can assist the successive algorithm development.

2. Random walk based techniques. The rationale 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. Several variations exist, to name a few:

   - PageRank with escape probability:

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

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

   - Personalized PageRank:

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

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

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

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

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

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

- Simrank:

$$s(a,b) = \frac{\gamma}{|I(a)||I(b)|} \sum_{i \in I(a), j \in I(b)} s(i,j) \tag{5}$$

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.

3. Simple but effective graph-based proximity measures:

- Katz Score:

$$Katz(i,j) = \sum_{l=1}^{\infty} \beta^l A^l(i,j) \tag{6}$$

The matrix series results in

$$Katz = (I - \beta A)^{-1} - I \tag{7}$$

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

- Common Neighbours:

$$Common(i,j) = |N(i) \cap N(j)| \tag{8}$$

- Adamic/Adar score:

$$Adamic/Adar(i,j) = \sum_{k \in N(i) \cap N(j)} \frac{1}{\log |N(k)|} \tag{9}$$

Adamic/Adar can be seen as an extension of simple common neighbours, which take the neighbours degree into consideration.

- Jaccards coefficient:

$$J(i,j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \tag{10}$$

This coefficient can be seen as another extension of common neighbours. It is effective when the graph is sparse.

4. Evaluation of obtained proximity measures:

- Draw the Receiver Operating Curve by varying proximity threshold. (Different TP and FP rate)
- Evaluate some popular quality functions:
  - Normalized cut:

$$Ncut(S) = \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\sum_{i\in S} d(i)} + \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\sum_{j\in \overline{S}} d(j)} \tag{11}$$

  - Conductance:

$$Conductance(S) = \frac{\sum_{i\in S, j\in \overline{S}} A(i,j)}{\min\{\sum_{i\in S} d(i), \sum_{j\in \overline{S}} d(j)\}} \tag{12}$$

  - Modularity:

$$Q = \sum_{i=1}^{k} \left[\frac{A(V_i, V_i)}{m} - \left(\frac{d(V_i)}{2m} \right)^2 \right] \tag{13}$$

  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 in our extreme local case.

5. Combination of those obtained proximity measures:
   - Train an MLP neural network to combine those proximity measures together to predict institution labels.
   - Do response fit using decision tree to see whether there exists dominant effective measurements.

   Tools used in this stage may be "weka". Our graph scale is about 100k vertexes and 200k edges for each observer. The MLP should support 100k data points with 6 to 10 features. Whether this is tractable using weka or not depends.

5 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.

`https://github.com/Czlyc/2C-Web-Research`