International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.1, Issue No.1, pp : 28-33 01 Nov. 2012

Evaluation of Alternative Centrality Measure Algorithm For Tracking Online Community In Social Network

Sanjiv Sharma, G.N. purohit
[1,2]Department of computer Science, Banasthali Vidyapith , Bansthali Rajasthan(INDIA)
er.sanjiv@gmail.com

Abstract— Network centrality is used to identify the most important/active people at the center of a network or those that are well connected. The tracking of single community in social networks is commonly done using some of the centrality measures employed in social network .The betweenness centrality measures has been used in SCAN (Social Cohesion Analysis of Network) method to track communities in social networks. This paper evaluates new alternative eigenvector centrality measures for tracking community and is more suitable compared to betweenness centrality measures algorithm.

***Keywords*—: Social Network Analysis, Centrality, betweenness centrality, SCAN, Communities, Eigenvector centrality.**

1. Introduction

Social network analysis [1] views social relationships in terms of network theory consisting of nodes and ties (also called edges, links, or connections). Nodes are the individual Communities within the network, and ties are the relationships between the Communities. Measures of centrality reflect the prominence of communities/units within a network. In graph theory and network analysis, there are various measures of the centrality of a vertex within a graph that determine the relative importance of the vertex/node in the graph/network.

The Internet has spawned different types of information sharing systems, including the Web. Recently, online social networks have gained significant popularity and are now among the most popular sites on the Web. Unlike the Web, which is largely organized around content, online social networks are organized around users. Participating users join a network, publish their profile and (optionally) any content, and create links to any other users with whom they associate. The resulting social network provides a basis for maintaining social relationships, for finding users with similar interests, and for locating content and knowledge that has been contributed or endorsed by other users.

Numerous centrality measures such as degree [3,8], closeness [3,22], betweenness [2,3,12,25] information [3,18], eigenvector [8,17], and dependence centrality [3] have been used for characterizing the social behaviour and connectedness of nodes within networks. The logic of using centrality measures is that people who are actively involved in one or

more subgroups will generally score higher with respect to centrality scores for the corresponding network.

Betweenness centrality is mostly used to find subgroup and to measure community membership [2], whereas degree and closeness centrality are used for characterizing influential members. Although network centrality measures are easy to calculate using computer programs such as Pajek [25] and UCINET [23,9], there has been no consensus among researchers as to the most meaningful centrality measure to use for finding subgroup members . In extremely large social networks, computational efficiency may become an issue in selecting a relevant centrality measure to use. Analyzing with UCINET, a betweenness centrality measure require large computation & time complexity as compare to eigenvector centrality and degree centrality is the easiest to calculate.

2. Background & Related Work

The Social Cohesion Analysis of Networks (SCAN) method was developed for automatically identifying subgroups of people in social networks that are cohesive over time [25]. The SCAN method is applied based on the premise that a social graph can be obtained from the online community interactions [19] where the links are untyped (i.e., there are no associated semantics). In the social graph, each link represents an interaction between two individuals where one individual has responded to the other's post in the online community. The SCAN method has been designed to identify cohesive subgroups on the basis of social networks inferred from online interactions around common topics of interest. The SCAN method consists of the following three steps:

1. Select: In the first step, the possible members of cohesive subgroups are identified.We set a cutoff value on a measure that is assumed to be correlated with likelihood of being a subgroup member, and then filter out people who fail to reach the cutoff value on that measure. We use betweenness centrality as this cutoff measure, since prior research has found that it does a fairly good job of identifying subgroup members although other centrality measures such as degree and closeness centrality could also be used.

By selecting a cutoff centrality measure, we obtain a subgraph of the original social graph where all members that have a centrality below the cutoff centrality measure are removed, resulting in a list of potential active members of subgroups.

2. Collect: Grouping these potential members into subgroups.

3. Choose: Choosing cohesive subgroups that have a similar membership over time.

Limitation: The SCAN method only focused on betweenness centrality; other centrality measures may be useful.

3. Existing betweenness centrality measure algorithms for tracking online communities

Betweenness centrality

Betweenness [14,15] is a centrality measure of a vertex within a graph (there is also edge betweenness, which is not discussed here). Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.

For a graph G: = (V, E) with n vertices, the betweenness $C_B(v)$ for vertex v is computed as follows:

1. For each pair of vertices (s,t), compute all shortest paths between them.

2. For each pair of vertices (s,t), determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).

3. Sum this fraction over all pairs of vertices (s,t).

Or, more succinctly:

$$C_B(v) = \sum_{s \neq v \neq t \in V}^{|Y|} \sigma_{st}(v) / \sigma_{st} \quad \text{.....(1)}$$

where σ_{st} is the number of shortest paths [12] from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v. Calculating the betweenness and closeness centralities of all the vertices in a graph involves calculating the shortest paths between all pairs of vertices on a graph. In calculating betweenness and closeness centralities of all vertices in a graph, it is assumed that graphs are undirected and connected with the allowance of loops and multiple edges. When specifically dealing with network graphs, oftentimes graphs are without loops or multiple edges to maintain simple relationships (where edges represent connections between two people or vertices). Brande's algorithm will divide final centrality scores by 2 to account for each shortest path being counted twice.

The sequential algorithm

The sequential algorithm is explained in [Brandes][21], and we won't repeat all of that here. There's also a cartoon of part of a sample run in the slides by Robinson linked to the web site.The basic idea is to identify shortest paths by breadth-first search (or BFS). Performing a BFS from a starting vertex s gives the lengths of the shortest paths from s to every other vertex. With a little bit of extra bookkeeping, this BFS can also count shortest paths from s to every t, and can even keep track of how many of those paths go through every other vertex v. Actually, the search from s requires two sweeps over the graph: The first sweep is the BFS, which computes σ_{st} for every t and also records information about the "predecessors" of each vertex reached in the search;the second sweep goes through the BFS tree in reverse order, updating the centrality scores of each vertex using its predecessors.

The search starting from vertex s computes the contribution to $C_B(v)$ from the inner sum in Equation (1) for every v. The whole algorithm consists of doing the search from every possible starting vertex s, adding up the contributions as they're computed.

There are several places to introduce parallelism in the algorithm. Probably the simplest is to parallelize the outermost loop over s; that is, to do several breadths-first searches on different processors

at the same time. A second approach is to parallelize the individual breadth-first searches, either by working on multiple nodes on a level simultaneously or by working on multiple neighbours of a single node.

Baseline algorithm

A baseline algorithm that uses Betweenness Centrality to detect communities in the graph implicitly defined by an MLog. An MLog is represented as a graph where the nodes represent people and each edge represents an instant message exchanged between two persons. The baseline algorithm is as follows:

- Detect connected components in graph.
- Compute Betweenness Centrality for each node in the subgraph.
- Remove the node with the highest Betweenness Centrality since it is a "boundary spanner".
- Create two new connected components and repeat from 2 until the subgraph has the desired number of nodes in the community.

4. Proposed alternative Eigenvector centrality measure algorithm for tracking online communities

Proposed alternative Eigenvector centrality measure algorithm for tracking online communities

Eigenvector centrality [17,19] is a measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Google's PageRank is a variant of the Eigenvector centrality measure.

Using the adjacency matrix to find eigenvector centrality : Let x_i denote the score of the i^{th} node. Let $A_{i,j}$ be the adjacency matrix of the network. Hence $A_{i,j} = 1$ if the i^{th} node is adjacent to the j^{th} node, and $A_{i,j} = 0$ otherwise. More generally, the entries in A can be real numbers representing connection strengths, as in a stochastic matrix.

For the i^{th} node, let the centrality score be proportional to the sum of the scores of all nodes which are connected to it. Hence

$$X_i = (1/\lambda)\sum_{j \in M(i)} (X_j) = (1/\lambda)\sum_{j=1}^{N} A_{i,j}(X_j) \quad \text{.....(2)}$$

where $M(i)$ is the set of nodes that are connected to the i^{th} node, N is the total number of nodes and λ is a constant. In vector notation this can be rewritten as

$X=(1/\lambda)AX$ or as the eigenvector equation $AX= \lambda X$

In general, there will be many different eigenvalues λ for which an eigenvector solution exists. However, the additional requirement that all the entries in the eigenvector be positive implies (by the Perron–Frobenius theorem) that only the greatest eigenvalue results in the desired centrality measure. The i^{th} component of the related eigenvector then gives the centrality score of the i^{th} node in the network. Power iteration is one of many eigenvalue algorithms that may be used to find this dominant eigenvector.Principal eigenvector of the (possibly valued) adjacency matrix of a network.

Eigenvector centrality is like a recursive version of degree centrality. The basic algorithm is as follows:

1. Start by assigning centrality score of 1 to all nodes (v_i = 1 for all i in the network)

2. Recompute scores of each node as weighted sum of centralities of all nodes in a node's neighborhood: v_i = sum_{j \in N} x_{ij}*v_j

3. Normalize v by dividing each value by the largest value

4. Repeat steps 2 and 3 until values of v stop changing.

A node is central to the extent that the node is connected to others who are central. An actor who is high on eigenvector centrality is connected to many actors who are themselves connected to many actors.

Power Method

An efficient method for finding the largest eigenvalue/eigenvector pair is the power iteration or power method. The power method is an iterative calculation that involves performing a matrix-vector multiply over and over until the change in the Iterate (vector) falls below a user

supplied threshold. We outline the power method algorithm below with the following pseudo-code:

```
X[n]=1; //create initial vector and set to a vector of all ones.

While (convergence criteria not met)

{   for(i=0;i<=n-1;i++)

        for(j=0;j<=n-1;j++)

            tmp[i]+=x[j]*A[i][j];   //metrix vector multiply

     for(k=0;k<=n-1;k++)

              norm_fector +=tmp[j]*tmp[k];  //calculate Euclidian norm

for(k=0;k<=n-1;k++)

            tmp[i]/=norm_fector;     //normlize the vector

X=tmp;

}
```

In the initialization step, the starting iterates is set to a vector of all ones, allowing the algorithm to behave deterministically, a useful property for performance analysis. Following initialization, the algorithm enters a while loop consisting of a matrix-vector multiply, followed by a calculation of the iterate's Euclidean norm, and finally a vector normalization. Interesting graphs tend to be sparse which means the adjacency matrix will generally be sparse, and can be efficiently represented in compressed row storage (CRS). In addition to being space efficient, storing the matrix in CRS improves the efficiency of the matrix-vector multiply in the loop. The pseudo-code for the sparse matrix-vector multiply is:

```
// sparse matrix vector multiply

for i=0:(n-1)

for j=0:(numNonZeroElements[i]-1)

colIdx = C(I,j);

tmp[i] += x[colIdx];
```

5. Evaluation of centrality measures algorithms

Assuming a Social network of friends in which all friends are connected to each other by friendship relations. Centrality measures select the important friends from the social network. In this section UCINET simulator [23] analyze the social network and evaluate eigenvector & betweenness centrality measures algorithm in following manner:

Step1: Suppose social network dataset contain name of friends Manish, abhishek, ajay, akhilesh ,aman, ashish , mitesh, neeraj, rajiv , ravi , shankar ,sumit,vijay ,vinay and each row represents the relationship among the friends :

shankar ashish sumit ajay vijay
ashish
abhishek akhilesh ravi ajay vijay Manish neeraj
sumit shankar ashish ajay vijay neeraj
mitesh shankar ashish sumit vijay Manish neeraj
vijay shankar ashish sumit ajay Manish neeraj
Manish ashish ajay aman rajiv neeraj
ashish rajiv vinay vikas
rajiv Manish ram sumit vikas anil
vinay shankar rajiv vikas anil
vikas ram rajiv vinay anil
neeraj ashish sumit ajay vijay Manish
vikram
anil rajiv vinay vikas

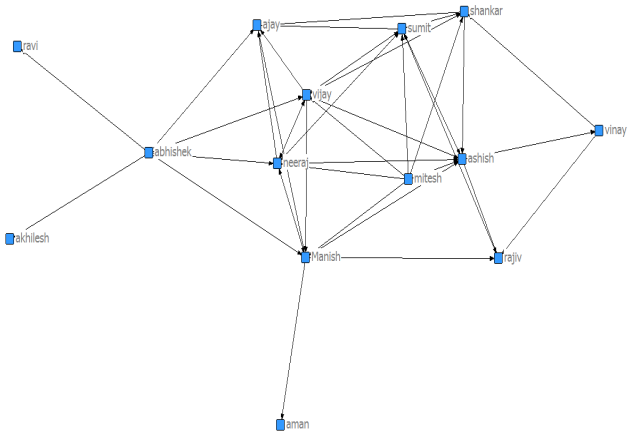
Step 2: After examine social network dataset; we can get following table 1 in form of text file (friends.txt). Table 1 shows comparatively analysis of both (eigenvector & Betweenness) measures & algorithm as per following details:

		1	2
S.no.	Name of person	Betweenness	Eigenvector
1	Manish	15.450	0.336
2	abhishek	0.000	0.223
3	ajay	0.000	0.297
4	akhilesh	0.000	0.035
5	aman	0.000	0.052
6	ashish	11.750	0.359
7	mitesh	0.000	0.318
8	neeraj	4.367	0.350
9	rajiv	8.917	0.180
10	ravi	0.000	0.035
11	shankar	5.033	0.283
12	sumit	7.533	0.337
13	vijay	6.867	0.388
14	vinay	2.083	0.127

Table1

Step 3: Graph 1 is generated by UCINET Simulator for tracking community from social network data set and objects represents nodes of graph. Graph 1 shows highly influential node for selecting

communities or tracking communities and also indicate that eigenvector centrality measure is more effective as compare to betweenness centrality measure.



Graph 1

Above simulation performed by UCINET Simulator [23] where two different centralities measures are compared on the basis of value generated by community data. Consequentially resulting graph show higher influential nodes. Eigenvector centrality shows vijay,ashish,and neeraj are highly influential nodes. Manish and ashish are the highly influential node in case of betweenness centrality. Network Centralization Index [23] represents overall centralization of the network. Graph1 is analyzed by simulator and simulator shows Betweenness centrality based network centralization Index [14] is 7.61%.Despite this, the overall network centralization is relatively low. Hence there cannot be a lot of "betweenness." In the sense of structural constraint, there is not a lot of "power" in this network. Similarly Eigenvector centrality based Network centralization index [17] is 31.76%. Hence eigenvector centrality provides a substantial amount of concentration or centralization in this whole network/graph1. That is, the power of individual actors varies rather substantially, and this means that, overall, positional advantages are rather unequally distributed in this network.

Betweenness centrality identifies most connected nodes on the most traveled paths. Eigenvector centrality considers nodes connected to other high degree nodes as highly central.To calculate the betweenness centrality [8] tells how "in-between" a node is within a complete graph by measuring the number of all shortest paths that pass through that node. The best known algorithm for that takes O(VE + V2logV) time and O(N+V) memory [21]. Eigenvector centrality [8] measures the importance of a node by the measure of its connectivity to other "important" nodes in the graph. The process of finding it is similar to belief propagation and the algorithm is iterative with the time complexity O(k.E). where k is the number of iterations needed before convergence, V is the number of vertices and E is the number of edges in the graph. In dense graphs, the number of edges E, tends to approach O(V2). With the increase in the E and the V value, the computation becomes super-linearly expensive. Eigen vector centrality take less time & space as compare to betweenness centrality . For example, to calculate Betweenness centrality and eigenvector centrality for a graph with V=656, and E=25000 took 142.5 s and 5.75 s respectively. Another example suppose graph of any social network contain 2,625 vertex and 99,999 edges. Betweenness centrality and eigenvector centrality algorithm took 570 s and 23 s respectively where overall throughput can be achieved by eigenvector centrality. After observation resultant eigenvector centrality measure is more suitable for “select” step of SCAN method [25] because eigenvector centrality can find or select high influential node in less time & easier manner.

6. Conclusion

This paper evaluates eigenvector and betweenness centrality measure algorithms for tracking online community in social network, according to the following observations. First, eigenvector centrality measures for tracking community are more suitable against betweenness centrality measures algorithm. Second a betweenness centrality measure has high calculation complexity as compare to eigenvector measure algorithms. Third, eigenvector measure algorithms provide better time complexity against betweenness centrality measure algorithms. Forth, eigenvector more desirable measure for SCAN method for selecting high influential node or communities.

References

[1] Garton L, Haythornthwaite C, Wellman B (1997) Studying online social networks. J Comput Mediated Commun 3(1):1–30.

[2] L. Freeman. A set of measures of centrality based on betweenness.Sociometry, 40(1):35–41, 1977

[3] Estrada E, Rodriguez-Velazquez AJ (2005) Subgraph centrality in complex networks. Phys Rev E 71:056103

[4] Backstrom L (2006) Group formation in large social networks: membership, growth, and evolution. In: KDD 06: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, ACM Press, pp 44–54

[5] Burt R (1982) Toward a structural theory of action: network models of social structure, perception and action. Academic, New York.

[6] Carrington PJ, Scott J, Wasserman S (2006) Models and methods in social network analysis. Cambridge University Press, New York, NY, USA

[7] Danon L, Duch J, Diaz-Guilera A, Arenas A (2005) Comparing community structure identification.J Stat Mech Theor Exp: P09008

[8] Freeman CL (1978) Centrality in social networks: Conceptual clarification. Social Networks 1:215–239

[9] Clauset A (2005) Finding local community structure in networks. Phys Rev E 72:026132

[10] Du N, Wu B, Pei X, Wang B, Xu L (2007) Community detection in large-scale social networks.In WebKDD/SNA-KDD ’07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, New York, NY, USA,pp 16–25

[11] N. Friedkin. Horizons of observability and limits of informal control in organizations. Social Forces, 62(1):57–77, 1983.

[12]G. Kahng, E. Oh, B. Kahng, and D. Kim. Betweenness centrality correlation in social networks. Phys. Rev. E, 67:01710–1, 2003.

[13]Danon L, Duch J, Diaz-Guilera A, Arenas A (2005) Comparing community structure identification.J Stat Mech Theor Exp: P09008

[14] M. Newman. A measure of betweenness centrality based on random walks. Social Networks, 27(1):39–54, 2005.

[15] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. Social Networks, 11:1–37, 1989.

[16] Newman EJM, Girvan M (2004) Finding and evaluating community structure in networks.Phys Rev E 69:026113

[17] Ruhnau B (October 2000) Eigenvector-centrality – a node-centrality? Social Networks 22(4):357–365

[18] Fortunato S, Latora V, Marchiori M (2004) Method to find community structures based on information centrality. Phys Rev E (Stat Nonlinear, Soft Matter Phys) 70(5):056104

[19] Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. Proc Natl Acad Sci USA 101(9):2658–2663

[20] Tantipathananandh C, Berger-Wolf YT, Kempe D (2007) A framework for community identification in dynamic social networks. In: KDD ’07: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY,USA, pp 717–726

[21] Traud LA, Kelsic DE, Mucha JP, Porter AM (2009) Community structure in online collegiate social networks, American Physical Society, 2009 APS March Meeting, March 16–20, pp.

[22] U. Brandes. A faster algorithm for betweenness centrality. Journal of Mathematical Sociology, 25(2):163–177, 2001.

[23] U. Brandes and C. Pich. Centrality estimation in large networks.I. J. of Bifurcation and Chaos, 17(7):2303–2318, 2007.

[24]Borgatti SP, Everett GM, Freeman CL (2002) Ucinet for windows: software for social network analysis. Analytic Technologies, Harvard, USA Science BV, Amsterdam, the Netherlands, pp 107–117.

[25] de Nooy W, Mrvar A, Batagelj V (2005) Exploratory social network analysis with pajek.Cambridge University Press, New York, USA.

[26] Chin A (January 2009) Social cohesion analysis of networks: a method for finding cohesive subgroups in social hypertext. PhD thesis, University of Toronto.