Finding High Betweenness Centrality Vertices in Large Networks

Vladimir Ufimtsev and Sanjukta Bhowmick

Department of Computer Science, University of Nebraska at Omaha

Introduction. Betweenness centrality (BC) is a widely applied network measure for identifying important vertices in complex networks. BC measures the importance of a vertex with respect to the flow of information in a network, based on the number of shortest paths that pass through that vertex. Specifically, the BC of vertex v is defined as [3]: $BC(v) = \sum\limits_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the total number of shortest paths in G between nodes s and t, and $\sigma_{st}(v)$ is the total number of shortest paths in G between s and t that pass through v. Most algorithms for computing BC have to compute the values for all the vertices in the network i.e. it is not possible to compute the BC just for a specific vertex without having to compute BC for the whole network. For example, the popular Brandes method [1] for obtaining BC, cumulatively computes the values for every vertex in the network. Although the algorithm has polynomial complexity, the execution time $O(V * E)$ is still prohibitive for large-scale networks. However, in practice only the vertices with the highest BC values are required. Here we show how we can use group testing to obtain the d-highest BC vertices, in shorter time than computing the BC of all vertices (and then sorting to find the highest ones).

Group Testing to Identify High BC Vertices

We focus on identifying only the top-highest BC vertices in the network, and it is the identity not the actual BC value of the vertex that is important. We use our proposed algorithm (for calculating the BC of a specified vertex) in a group testing based technique to identify the vertices in the network that have the highest BC [7]. The central idea of group testing is that if there is a small percentage of defective units in a large population of units, it is more efficient (requires less tests) to test the units in carefully selected groups, using for example principles of superimposed code theory, rather than testing each unit separately. Group testing has a vast amount of applications including pattern matching and DNA library screening [4, 5]. In our application of group testing, the defective units correspond to vertices with high BC. We use a group testing design based on a Latin square to determine the number of groups (tests) and what vertices are part of each group. According to this design, for each test the specified vertices are grouped into a single supervertex and the betweenness centrality of that specified supervertex is calculated using our single vertex BC algorithm. If the BC value of the supervertex (group) is high (exceeds a threshold) then the result is designated as positive (1) otherwise it is negative (0). Upon completion of all the tests, the vertices that have the highest BC rank are identified. Theoretically this method guarantees that we find at least two highly ranked vertices in $3\lceil\sqrt{n}\rceil$ tests in a network on n vertices.

Preliminary Results

Using our group testing algorithm we performed experiments over a set of ten networks collected from the DIMACS Implementation Challenge Set [2] and the Stanford Network Analysis Project [6].

To evaluate the efficiency and accuracy of the method we analyze how many of the nodes identified by group testing are actually high ranking. Using the Brandes algorithm [1] we obtain the full ranking for each network and see what rank the nodes identified by group testing have. The group testing design we are using (based on Latin squares) is successful if it identifies at least the top 2 vertices. The results are given in Table 1 (reproduced from [8]). Out of the ten networks, group testing was successful on six networks (top six rows of the table), and found low ranked (below rank 10) vertices for the other four (the last four rows of the table).

On further study we observed that the networks for which group testing failed to find the high ranked vertices were the ones that were most sensitive to small perturbations in the network structure.

Therefore group testing can be also used as a method to classify networks that are robust to noise from networks that are more sensitive.

Table 1. Finding High BC Vertices Using Group Testing on Real-World Networks. The best threshold and the vertices obtained using that threshold are given. The vertices are represented by their rank, as per their BC values obtained using the Brandes method.

Name	**Vertices**	**Edges**	# of Tests	Threshold	**High BC Vertices**
Karate	34	156	18	55%	**1st, 2nd**
Chesapeake	39	340	21	30%	**1st, 2nd**
AS20000102	6474	13233	243	12%	**1st, 2nd, 3rd**
AS20000101	3570	7391	180	16%	**1st, 2nd**
Caida	16301	65910	384	21%	**1st, 2nd, 3rd**
C. Elegans	453	4050	66	35%	**1st, 2nd, 4th** **10th** + 3 low ranked
Les Mis.	77	508	27	45%	**1st, 10th**, +3 low ranked
GrQc	5242	28980	219	80.3%	20 low ranked
HepTh	9877	51971	300	76%	6 low ranked
Power Grid	4941	13188	213	84%	6 low ranked

Each group of vertices can be formed and the corresponding tests can be executed independently. Therefore group testing is perfectly parallelizable. In each test we are only required to know the BC of the supervertex. However, most of the current BC computing algorithms focus on cumulatively finding the BC of all the vertices, and cannot identify the BC of only a specific vertex.

Therefore, in order to efficiently apply our group testing algorithm, we have developed an algorithm that computes the BC of one vertex only. The efficiency of the algorithm is related to the size of and the number of chordless cycles in the graph. If the graph is chordal, i.e. the largest chordless cycle is of length three, then we can compute the BC of a designated vertex in time proportional to execute one breadth first search. For larger chordless cycles, in the worst case, we have to execute a BFS for each cycle. Therefore, if the number of chordless cycles in the graph is q, then computing the BC of a vertex would take time $O(q * V)$.

In this talk we will present our results on group testing over a wider set of networks, we will demonstrate how group testing is an effective method for finding the highest BC vertices in robust networks, and how we can identify sensitive networks using this method. Finally we will present our algorithm for finding the BC of a single vertex, along with the scalability results for group testing.

References

1. U. Brandes , Faster Algorithm for Betweenness Centrality *J. Math. Sociol.*, 25, 163 (2001)
2. DIMACS 10th Implementation Challenge http://www.cc.gatech.edu/dimacs10/archive/clustering.shtml (2011)
3. L.C. Freeman, A Set of Measures of Centrality Based on Betweenness, *Sociometry*, 40, 35 (1977)
4. A.J. Macula, L.J. Popyack , A group testing method for finding patterns in data, *Discrete Appl. Math.* 144, no. 1-2, 149–157, (2004)
5. A.J. Macula, Probabilistic nonadaptive group testing in the presence of errors and DNA library screening, *Combinatorics and Biology* (Los Alamos, NM, 1998). Ann. Comb. 3, no. 1, 61–69, (1999)
6. Stanford Network Analysis Project (SNAP) http://snap.stanford.edu/index.html
7. V. V. Ufimtsev, A Scalable Group Testing Based Algorithm for Finding d-highest Betweenness Centrality Vertices in Large Scale Networks,Poster, *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011

8. V. V. Ufimtsev, S. Bhowmick, Application of Group Testing in Identifying High Betweenness Centrality Vertices in Complex Networks *KDD Workshop on Mining and Learning with Graphs*, (2013)