

Zaitzeff, Alexander

Spectral Graph Theory for Weighted Digraphs

Faculty Mentor: Jeffrey Humpherys, Mathematics

For digraphs weighted and unweighted, one important application is ranking: Given a directed graph, whether it be the Internet or a social network, which node (representing a web page or a person) is the most important? There are many different methods to find answer this question. A few are highest indegree, closeness centrality¹, betweenness centrality², eigenvector centrality, Katz Centrality³, and PageRank⁴. Our idea is to use sparsity, or the idea that in a network only has a few important nodes, to determine the ranking on a graph.

A directed graph can be represented as a matrix. The matrix has eigenvalues and eigenvectors. For eigenvector centrality the values in the left Perron eigenvector (the eigenvector corresponding to the largest eigenvalue) correspond to a centrality measure. The entries with the highest values in the left Perron eigenvector correspond to the most important, or most central, nodes in a graph. This method has the effect that the importance of a node depends on the number and the quality of its links incoming links⁵.

In my and my professor's measure we perform L1 minimization in a neighborhood of the Perron eigenvector. This finds a vector that is sparse (few non-zero entries) and close to the Perron eigenvector. Then we take the entries in the 'sparse' vector as importance where the highest entries in the 'sparse' vector correspond to the most important nodes in the graph. We do this on the matrix itself and the graph as viewed as a random walk (Pretend you are walking on the graph and the node you go to next is determined by picking an outgoing edge at random) as two separate rankings.

To test our method we generated various graphs and ran our ranking algorithm on the graph as well as highest indegree, closeness centrality, betweenness centrality, eigenvector centrality, Katz Centrality, and PageRank to compare.

The table shows the rankings of the top nodes on a power law graph – a structure that is characteristic of networks found in real data – on 40 nodes. The column 'sparNor' is the algorithm run on the 0-1 matrix that represents graph and the column 'sparRW' is the algorithm run on the random walk matrix representation of the graph. The figures

¹ Alex Bavelas. Communication patterns in task-oriented groups. Journal of the acoustical society of America, 1950.

² Linton C Freeman. A set of measures of centrality based on betweenness. Sociometry, pages 35–41, 1977.

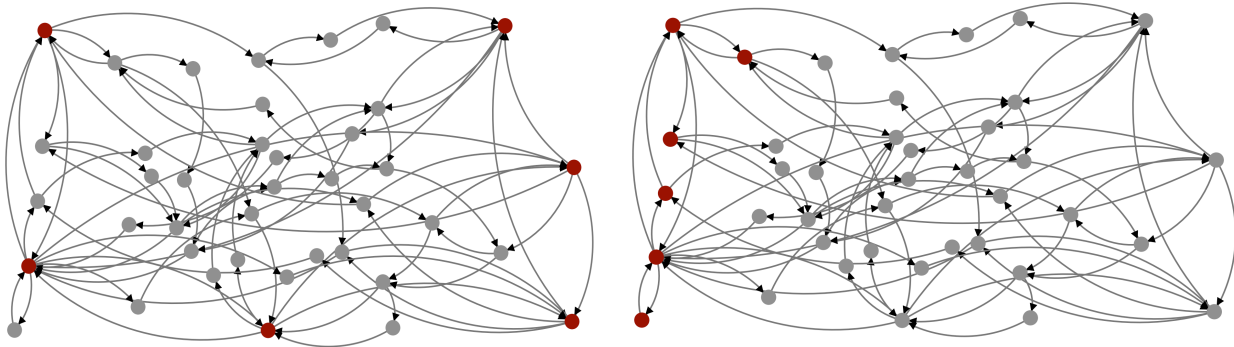
³ Leo Katz. A new status index derived from sociometric analysis. Psychometrika, 18(1):39–43, 1953.

⁴ Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18):3825–3833, 2012.

⁵ Mark EJ Newman. The mathematics of networks. The new palgrave encyclopedia of economics, 2(2008):1–12, 2008.

show the graph with the top six most important nodes highlighted for eigenvector centrality and our algorithm run on the 0-1 matrix.

Top Nodes	In-degree	Close-ness	Between-ness	Eigen-vector	Katz	Page-rank	Spar-Nor	Spar-RW
First	30	30	30	30	30	30	2	30
Second	26	4	15	2	15	4	30	10
Third	15	26	26	15	2	2	29	26
Forth	10	15	4	10	4	26	39	2
Fifth	2	14	2	26	26	10	12	15
Sixth	4	2	6	4	10	15	16	4



Figures 1 and 2 – The red nodes are the six most important given by eigenvalue centrality (Right) and by 'sparse' vectors on the 0-1 matrix (Left).

Our algorithm picks up on some nodes that are important that one can see by looking at the graph that are not picked up by the other algorithms. Node 12, which is in the bottom left hand corner, communicates with what is the most important node in the graph according to the other algorithms. This shows promise that our algorithm is catching structure in the graph that other algorithms miss.

Further research needs to find what are the characteristics of the nodes that the algorithm picks up on, how to better pick the neighborhood around the Perron eigenvector, and then extend this to larger, different types, and real-world graphs.

In conclusion, my professor and I have looked at another way of finding importance in network structures using the idea that only a few of the nodes are important. This method has good results on small test cases.