Fast Approximation of Centrality

David Eppstein* Joseph Wang*

Abstract

Social studies researchers use graphs to model group activities in social networks. An important property in this context is the *centrality* of a vertex: the inverse of the average distance to each other vertex. We describe a randomized approximation algorithm for centrality in weighted graphs. For graphs exhibiting the small world phenomenon, our method estimates the centrality of all vertices with high probability within a $(1+\epsilon)$ factor in near-linear time.

1 Introduction

In social network analysis, the vertices of a graph represent agents in a group and the edges represent relationships, such as communication or friendship. The idea of applying graph theory to analyze the connection between the structural *centrality* and group process was introduced by Bavelas [4]. Various measurement of centrality [7, 14, 15] have been proposed for analyzing communication activity, control, or independence within a social network.

We are particularly interested in *closeness centrality* [5, 6, 24], which is used to measure the independence and efficiency of an agent [14, 15]. Beauchamp [6] defined the closeness centrality of agent a_j as

$$\frac{n-1}{\sum_{i=1}^{n} d(i,j)}$$

where $d(i,j)$ is the distance between agents i and j.[1] We are interested in computing centrality values for all agents. To compute the centrality for each agent, it is sufficient to solve the all-pairs shortest-paths (APSP) problem. No faster exact method is known.

The APSP problem can be solved by various algorithms in time $\mathcal{O}(nm + n^2 \log n)$ [13, 19], $\mathcal{O}(n^3)$ [12], or more quickly using fast matrix multiplication techniques [2, 11, 25, 26]. Faster specialized algorithms are known for graph classes such as interval graphs [3, 9, 23] and chordal graphs [8, 17], and the APSP problem can be solved in average-case in time $\mathcal{O}(n^2 \log n)$ for various types of random graph [10, 16, 20, 22]. Because these results are slow, specialized, or (with fast matrix multiplication) complicated and impractical, and because recent applications of social network theory to the internet may involve graphs with millions of vertices, it is of interest to consider faster approximations. Aingworth et al. [1] proposed an algorithm with an additive error of 2 for the unweighted APSP problem that runs in time $\mathcal{O}(n^{2.5}\sqrt{\log n})$. However this is still slow and does not provide a good approximation when the distances are small.

In this paper, we consider a method for fast approximation of centrality. We apply a random sampling technique to approximate the inverse centrality of all vertices in a weighted graph to within an additive error of $\epsilon\Delta$ with high probability in time $\mathcal{O}(\frac{\log n}{\epsilon^2}(n \log n + m))$, where ϵ is any fixed constant and Δ is the diameter of the graph.

It has been observed empirically that many social networks exhibit the *small world phenomenon* [21]: their diameter is bounded by a constant, or, equivalently, the ratio between the minimum and maximum distance is bounded. For such networks, the inverse centrality at any vertex is $\Omega(\Delta)$ and our method provides a near-linear time $(1+\epsilon)$-approximation to the centrality of all vertices.

2 The Algorithm

We now describe a randomized approximation algorithm RAND for estimating centrality. RAND randomly chooses k sample vertices and computes single-source shortest-paths (SSSP) from each sample vertex to all other vertices. The estimated centrality of a vertex is defined in terms of the average distance to the sample vertices.

Algorithm RAND:

1. Let k be the number of iterations needed to obtain the desired error bound.

2. In iteration i, pick vertex v_i uniformly at random from G and solve the SSSP problem with v_i as the source.

3. Let
$$\hat{c}_u = 1 / \sum_{i=1}^{k} \frac{n\, d(v_i, u)}{k(n-1)}$$
be the centrality estimator for vertex u.

It is not hard to see that, for any k and u, the expected value of $1/\hat{c}_u$ is equal to $1/c_u$.

*Dept. Inf. & Comp. Sci., UC Irvine, CA 92697-3425, USA, {eppstein,josephw}@ics.uci.edu.

[1]This should be distinguished from another common concept of graph centrality, in which the most central vertices minimize the maximum distance to another vertex.

LEMMA 2.1. (HOEFFDING [18]) *If $x_1, x_2, \ldots, x_k$ are independent, $a_i \leq x_i \leq b_i$, and $\mu = E[\sum x_i/k]$ is the expected mean, then for $\xi > 0$*

$$\Pr\left\{\left|\frac{\sum_{i=1}^{k} x_i}{k} - \mu\right| \geq \xi\right\} \leq 2e^{-2k^2\xi^2/\sum_{i=1}^{k}(b_i-a_i)^2}.$$

We need to bound the probability that the error in estimating the inverse centrality of any vertex u is at most ξ. This is done by applying Hoeffding's bound with $x_i = \frac{d(i,u)n}{(n-1)}$, $\mu = \frac{1}{c_u}$, $a_i = 0$, and $b_i = \frac{n\Delta}{n-1}$. Thus the probability that the difference between the estimated inverse centrality $1/\hat{c}_u$ and the actual inverse centrality $1/c_u$ is more than ξ is

$$\begin{aligned} \Pr\left\{\left|\tfrac{1}{\hat{c}_u} - \tfrac{1}{c_u}\right| \geq \xi\right\} &\leq 2 \cdot e^{-2k^2\xi^2/\sum_{i=1}^{k}(b_i-a_i)^2} \\ &= 2 \cdot e^{-2k^2\xi^2/k(\frac{n\Delta}{n-1})^2} \\ &= 2 \cdot e^{-\Omega(k\xi^2/\Delta^2)} \end{aligned}$$

For $\xi = \epsilon\Delta$, using $\Theta(\frac{\log n}{\epsilon^2})$ samples will cause the probability of error at any vertex to be bounded above by e.g. $1/n^2$, giving at most $1/n$ probability of having greater than $\epsilon\Delta$ error anywhere in the graph.

The total running time of algorithm is $\mathcal{O}(k \cdot m)$ for unweighted graphs and $\mathcal{O}(k(n \log n + m))$ for weighted graphs. Thus, for $k = \Theta(\frac{\log n}{\epsilon^2})$, we have an $\mathcal{O}(\frac{\log n}{\epsilon^2}(n \log n + m))$ algorithm for approximating centrality within an inverse additive error of $\epsilon\Delta$ with high probability.

Acknowledgements

We thank Dave Goggin for bringing this problem to our attention, and Lin Freeman for helpful comments on a draft of this paper.

References

[1] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.* 28(4):1167–1181, 1999.

[2] N. Alon, Z. Galil, and O. Margalit. On the exponent of the all pairs shortest path problem. *J. Comput. Syst. Sci.* 54(2):255–262, April 1997.

[3] M. J. Atallah, D. Z. Chen, and D. T. Lee. An optimal algorithm for shortest paths on weighted interval and circular-arc graphs. *Proc. 1st Eur. Symp. Algorithms*, pp. 13–24. Springer-Verlag, Lect. Notes in Comp. Sci. 726, 1993.

[4] A. Bavelas. A mathematical model for group structures. *Human Organization* 7:16–30, 1948.

[5] A. Bavelas. Communication patterns in task oriented groups. *J. Acoust. Soc. Amer.* 22:271–282, 1950.

[6] M. A. Beauchamp. An improved index of centrality. *Behavioral Science* 10:161–163, 1965.

[7] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *J. Math. Sociol.* 2:113–120, 1972.

[8] A. Brandstadt, V. Chepoi, and F. Dragan. The algorithmic use of hypertree structure and maximum neighborhood orderings. *Proc. 20th Int. Worksh. Graph-Theoretic Concepts in Computer Science*, pp. 65–80. Springer-Verlag, Lect. Notes in Comp. Sci. 903, 1994.

[9] D. Z. Chen, D. T. Lee, R. Sridhar, and C. N. Sekharan. Solving the all-pair shortest path query problem on interval and circular-arc graphs. *Networks* 31(4):249–257, 1998.

[10] C. Cooper, A. M. Frieze, K. Mehlhorn, and V. Priebe. Average-case complexity of shortest-paths problems in the vertex-potential model. *Proc. Worksh. Randomization and Approximation*, pp. 15–26, 1997.

[11] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computation* 9(3):251–280, March 1990.

[12] R. W. Floyd. Algorithm 97: shortest path. *Commun. ACM* 5(6):345, June 1962.

[13] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM* 34(3):596–615, July 1987.

[14] L. C. Freeman. Centrality in social networks: I. conceptual clarification. *Social Networks* 1:215–239, 1979.

[15] N. E. Friedkin. Theoretical foundations for centrality measures. *Amer. J. Sociol.* 96(6):1478–1504, May 1991.

[16] A. M. Frieze and G. R. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Math.* 10:57–77, 1985.

[17] K. Han, C. N. Sekharan, and R. Sridhar. Unified all-pairs shortest path algorithms in the chordal hierarchy. *Discrete Applied Math.* 77:59–71, 1997.

[18] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statistical Assoc.* 58(301):713–721, March 1963.

[19] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM* 24(1):1–13, January 1977.

[20] K. Mehlhorn and V. Priebe. On the all-pairs shortest path algorithm of Moffat and Takaoka. *Proc. 3rd Eur. Symp. Algorithms*, pp. 185–198. Springer-Verlag, Lect. Notes in Comp. Sci. 979, 1995.

[21] S. Milgram. The small world problem. *Psychol. Today* 2:60–67, 1967.

[22] A. Moffat and T. Takaoka. An all pairs shortest path algorithm with expected time $\mathcal{O}(n^2 \log n)$. *Proc. 26th IEEE Symp. Foundations of Computer Science*, pp. 101–105, 1985.

[23] R. Ravi, M. V. Marathe, and C. P. Rangan. An optimal algorithm to solve the all-pair shortest path problem on interval graphs. *Networks* 22:21–35, 1992.

[24] G. Sabidussi. The centrality index of a graph. *Psychometrika* 31:581–603, 1966.

[25] R. Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *J. Comput. Syst. Sci.* 51(3):400–403, December 1995.

[26] G. Yuval. An algorithm for finding all shortest paths using $N^{2.81}$ infinite-precision multiplications. *Inf. Proc. Lett.* 4(6):155–156, March 1976.