Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 8, no. 1, pp. 39–45 (2004)

Fast Approximation of Centrality

David Eppstein *Joseph Wang*
Donald Bren School of Information & Computer Sciences
University of California, Irvine
eppstein@ics.uci.edu josephw@ics.uci.edu

Abstract

Social scientists 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 $\tilde{O}(m)$ time.

Article Type	Communicated by	Submitted	Revised
concise paper	M. Fürer	March 2001	March 2004

A preliminary version of this paper appeared in the proceedings of SODA 2001.

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 processes was introduced by Bavelas [4]. Various measurement of centrality [7, 15, 16] have been proposed for analyzing communication activity, control, or independence within a social network.

We are particularly interested in *closeness centrality* [5, 6, 25], which is used to measure the independence and efficiency of an agent [15, 16]. 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 $O(nm + n^2 \log n)$ [14, 20], $O(n^3)$ [13], or more quickly using fast matrix multiplication techniques [2, 11, 26, 28], where n is the number of vertices and m is the number of edges in a graph. Faster specialized algorithms are known for graph classes such as interval graphs [3, 9, 24] and chordal graphs [8, 18], and the APSP problem can be solved in average-case in time $O(n^2 \log n)$ for various types of random graph [10, 17, 21, 23]. 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 $O(n^{2.5}\sqrt{\log n})$. Dor et al. [12] improved the time to $\tilde{O}(n^{7/3})$. However it 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 $O(\frac{\log n}{\epsilon^2}(n \log n + m))$, where $\epsilon > 0$ and Δ is the diameter of the graph.

It has been observed empirically that many social networks exhibit the *small world phenomenon* [22, 27]. That is, their diameter is $O(\log n)$ instead of $O(n)$. For such networks, our method provides a near-linear time $(1+\epsilon)$-approximation to the centrality of all vertices.

[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.

2 Preliminaries

We are given a graph $G(V, E)$ with n vertices and m edges, the distance $d(u, v)$ between two vertices u and v is the length of the shortest path between them. The diameter Δ of a graph G is defined as $max_{u,v \in V} d(u, v)$. We define the centrality c_v of vertex v as follows:

$$c_v = \frac{n-1}{\sum_{u \in V} d(u, v)}.$$

If G is not connected, then $c_v = 0$. Hence we will assume G is connected.

3 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
$$\hat{c}_u = \frac{1}{\sum_{i=1}^{k} \frac{n\, d(v_i, u)}{k(n-1)}}$$
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 $1/c_u$.

Theorem 1 $E\left[\frac{1}{\hat{c}_u}\right] = \frac{1}{c_u}$.

Proof: Each vertex has equal probability of $1/n$ to be picked at each round. The expected value for $\frac{1}{\hat{c}_u}$ is

$$\begin{aligned} E\left[\frac{1}{\hat{c}_u}\right] &= \frac{n}{n-1} \cdot \frac{1}{n^k} \cdot \frac{kn^{k-1} \sum_{i=1}^{n} d(v_i, u)}{k} \\ &= \frac{n}{n-1} \cdot \frac{\sum_{i=1}^{n} d(v_i, u)}{n} \\ &= \frac{1}{c_u}. \end{aligned}$$

□

In 1963, Hoeffding [19] gave the following theorem on probability bounds for sums of independent random variables.

Lemma 2 (Hoeffding [19]) *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 $\xi > 0$*

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

Theorem 3 *Let G be a connected graph with n vertices and diameter Δ. With high probability, algorithm RAND computes the inverse centrality estimator $\frac{1}{\hat{c}_u}$ to within $\xi = \epsilon\Delta$ of the inverse centrality $\frac{1}{c_u}$ for all vertices u of G, using $\Theta(\frac{\log n}{\epsilon^2})$ samples, for $\epsilon > 0$.*

Proof: 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 $x_i = \frac{d(v_i,u)n}{(n-1)}$, $\mu = \frac{1}{c_u}$, $a_i = 0$, and $b_i = \frac{n\Delta}{n-1}$.

We know $E[1/\hat{c}_u] = 1/c_u$. 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

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

For $\xi = \epsilon\Delta$, using $k = \alpha \cdot \frac{\log n}{\epsilon^2}$ samples, $\alpha \geq 1$, 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. □

Fredman and Tarjan [14] gave an algorithm for solving the *SSSP* problem in time $O(n \log n + m)$. Thus, the total running time of our algorithm is $O(k \cdot m)$ for unweighted graphs and $O(k(n \log n + m))$ for weighted graphs. Thus, for $k = \Theta(\frac{\log n}{\epsilon^2})$, we have an $O(\frac{\log n}{\epsilon^2}(n \log n + m))$ algorithm for approximation of the inverse centrality within an additive error of $\epsilon\Delta$ with high probability.

4 Conclusion

We gave an $O(\frac{\log n}{\epsilon^2}(n \log n + m))$ randomized algorithm with additive error of $\epsilon\Delta$ for approximating the inverse centrality of weighted graphs. Many graph classes such as unweighted paths, cycles, and balanced trees, have inverse centrality proportional to Δ. More interestingly, Milgram [22] showed that many social networks have bounded diameter and inverse centrality. For such networks, our method provides a near-linear time $(1 + \epsilon)$-approximation to the centrality of all vertices.

Acknowledgments

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. Gali, 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. In *Proceedings of First Annual European Symposium on Algorithms*, pages 13–24, 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. *Journal of the Acoustical Society of America*, 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. *Journal of Mathematical Sociology*, 2:113–120, 1972.

[8] A. Brandstadt, V. Chepoi, and F. Dragan. The algorithmic use of hypertree structure and maximum neighborhood orderings. In *Proceedings of the Graph-Theoretic Concepts in Computer Science*, pages 65–80, 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. Frieze, K. Mehlhorn, and V. Priebe. Average-case complexity of shortest-paths problems in the vertex-potential model. In *Proceedings of European Symposium on Algorithms*, pages 15–26, 1997.

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

[12] D. Dor, S. Halperin, and U. Zwick. All pairs almost shortest paths. In *Proceedings of the 37rd Annual IEEE Symposium on Foundations of Computer Science*, pages 452–461, 1996.

[13] R. W. Floyd. Algorithm 97: Shortest path. *Communication of the Association for Computing Machinery*, 5:345, 1962.

[14] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal for the Association of Computing Machinery*, 34:596–615, 1987.

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

[16] N. E. Friedkin. Theoretical foundations of centrality measures. *AJS*, 96(6):1478–1504, 1991.

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

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

[19] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of American Statistical Association*, 58:713–721, 1963.

[20] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the Association for Computing Machinery*, 24:1–13, 1977.

[21] K. Mehlhorn and V. Priebe. On the all-pairs shortest path algorithm of Moffat and Takaoka. In *Proceedings of the 3rd Annual European Symposium on Algorithms*, pages 185–198, 1995.

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

[23] A. Moffat and T. Takaoka. An all pairs shortest path algorithm with expected time $o(n^2 \lg n)$. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pages 101–105, 1985.

[24] 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.

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

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

[27] D. J. Watts. *Small worlds: the dynamics of networks between order and randomness.* Princeton University Press, Princeton, N.J., 1999.

[28] G. Yuval. An algorithm for finding all shortest paths using $n^{2.81}$ infinite-precision multiplications. *Information Processing Letters*, 4:155–156, 1976.