

WORKSHOP ON INTERNET OF THINGS AND ARTIFICIAL INTELLIGENCE

Space Efficient Decremental Betweenness Algorithm for Directed Graphs

Reynaldo Gil-Pons¹, Leticia Arco²

1-Reynaldo Gil Pons. CERPAMID, Cuba, rey@cerpamid.co.cu

2-Leticia Arco. Vrije Universiteit Brussels, Belgium, leticiaarco@gmail.com

Abstract: Betweenness is one of the most popular centrality measures in the analysis of social networks. Its computation has a high cost making it implausible for relatively large networks. The dynamic nature of many social networks opens up the possibility of developing faster algorithms for the dynamic version of the problem. In this work, we propose a new decremental algorithm to compute betweenness centrality of all nodes in directed graphs extracted from social networks. Our algorithm uses linear space, making it suitable for large scale applications. The experimental evaluation on a variety of networks has shown our algorithm is faster than recalculation from scratch and competitive with recent approaches.

Keywords: Social Network Analysis; Betweenness Centrality; Dynamic Algorithms.

1. Introduction

Centrality is one of the most important concepts in the analysis of social networks. Among centrality measures, one of the most popular is betweenness centrality [2]. The betweenness of a node is a measure of the control this node has on the communication paths in the network. Therefore, it can be used to rank nodes according to their relative importance in a graph. Computing the betweenness of all nodes in a network has a high cost, so efficiency is the target of much related research. Nowadays, most graphs are inherently dynamic. When a graph suffers small changes, recomputing betweenness from scratch would be very inefficient.

OFFICIAL TEMPLATE FOR THE SUBMISSION OF PAPERS
II INTERNATIONAL SCIENTIFIC CONVENTION
“II ICCUCLV 2019”

JUNE 23th – 30th, 2019
CAYOS DE VILLA CLARA. CUBA.



In this paper, we present a space efficient algorithm to compute the betweenness centrality of all nodes in a directed decremental network. Its space complexity is linear in the size of the input graph and its time complexity is like that of algorithm **icentral** [3]. In the worst case, our algorithm is equivalent to recalculating betweenness in the biconnected component where the removed edge resides, plus some linear overhead. To the best of our knowledge, it is the first algorithm calculating betweenness centrality in decremental directed graphs, showing better performance than recalculation, and at the same time, having less than quadratic space complexity. It works with disconnected graphs, characteristic very important in real world applications.

2. An efficient algorithm to compute the betweenness centrality

The proposed algorithm is a generalization of **icentral** to deal with directed graphs.

Definition 1: Let G be a graph and let G^* be the graph G after deleting an edge (u, v) . We define affected component as the biconnected component of the undirected version of G to which the newly deleted edge (u, v) belonged.

The main obstacle in generalizing **icentral** is that, in directed graphs, when an edge is deleted, betweenness values of nodes outside the affected component can change as well. In the next theorem, we prove a formula allowing computing those changes efficiently.

Theorem 1: Let x be a node outside the affected component A , and let s be the articulation point inside the component such that its removal disconnects x from A . Then after the update, the betweenness of x changes by:

$$\delta_s(x) \cdot (reach^*(s) - reach(s)) + \delta_s^r(x) \cdot (reach^{*r}(s) - reach^r(s))$$

where $reach(s)$ equals the number of nodes z such that there exists a shortest path from z to x passing through A , superscript r means the function is applied to the reversed graph, and superscript $*$ indicates the function is applied to the updated graph.

Using the aforementioned theorem, we designed an algorithm that first updates the betweenness of all nodes inside the affected component, using a procedure very similar

OFFICIAL TEMPLATE FOR THE SUBMISSION OF PAPERS
II INTERNATIONAL SCIENTIFIC CONVENTION
“II ICCUCLV 2019”

JUNE 23th – 30th, 2019
CAYOS DE VILLA CLARA. CUBA.



to algorithm **brandes** [1] and **icentral**, and then that of all nodes outside the affected component in linear time and memory.

It is possible to modify slightly the proposed approach to work with graphs with arbitrary positive weights, by using Dijkstra algorithm instead of BFS. In graphs with multiples edges, parallel edges can be substituted with the edgewith smallest weight, and then obtain a simple graph with the same betweenness. On the other hand, it is straightforward to parallelize the most time-consuming part of the algorithm, the computation of the betweenness changes inside the affected component. As the δ values corresponding to affected sources are computed independently, these computations could be done by different nodes in a parallel environment. In this environment, good speedups are expected, similar to those in [3].

4. Conclusions

In this work, an algorithm for computing betweenness in decremental directed graphs has been proposed. Its memory usage is linear allowing it to scale to large graphs. Its time complexity is similar to that of algorithm proposed in [3], despite of handling the more general case of directed and possibly disconnected graphs. Experiments have shown our algorithm can be of practical interest for large directed graphs, mostly when faster algorithms [4] are not feasible due to their quadratic memory usage.

5. Bibliographical references

- [1] Brandes, U.: A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology* 25(2), 163–177 (2001)
- [2] Freeman, L.C.: A Set of Measures of Centrality Based on Betweenness(1977)
- [3] Jamour, F., Skiadopoulos, S., Kalnis, P.: Parallel Algorithm for Incremental Betweenness Centrality on Large Graphs. *IEEE Transactions on Parallel and Distributed Systems* (2017).
- [4] Nasre, M., Pontecorvi, M., Ramachandran, V.: Decremental all-pairs ALL shortest paths and betweenness centrality. *Lecture Notes in Computer* 8889, 766–778 (2014).

Contact Information
convencionuclv@uclv.cu
www.uclv.edu.cu