Fully-Dynamic Approximation of Betweenness Centrality

Elisabetta Bergamini and Henning Meyerhenke

Institute of Theoretical Informatics
Karlsruhe Institute of Technology (KIT), Germany
{elisabetta.bergamini,meyerhenke}@kit.edu

Abstract. Betweenness is a well-known centrality measure that ranks the nodes of a network according to their participation in shortest paths. Since an exact computation is prohibitive in large networks, several approximation algorithms have been proposed. Besides that, recent years have seen the publication of dynamic algorithms for efficient recomputation of betweenness in evolving networks. In previous work we proposed the first semi-dynamic algorithms that recompute an approximation of betweenness in connected graphs after batches of edge insertions.

In this paper we propose the first fully-dynamic approximation algorithms (for weighted and unweighted undirected graphs that need not to be connected) with a provable guarantee on the maximum approximation error. The transfer to fully-dynamic and disconnected graphs implies additional algorithmic problems that could be of independent interest. In particular, we propose a new upper bound on the vertex diameter for weighted undirected graphs. For both weighted and unweighted graphs, we also propose the first fully-dynamic algorithms that keep track of this upper bound. In addition, we extend our former algorithm for semi-dynamic BFS to batches of both edge insertions and deletions.

Using approximation, our algorithms are the first to make in-memory computation of betweenness in fully-dynamic networks with millions of edges feasible. Our experiments show that they can achieve substantial speedups compared to recomputation, up to several orders of magnitude.

Keywords: betweenness centrality, algorithmic network analysis, fully-dynamic graph algorithms, approximation algorithms, shortest paths.

1 Introduction

The identification of the most central nodes of a network is a fundamental problem in network analysis. Betweenness centrality (BC) is a well-known index that ranks the importance of nodes according to their participation in shortest paths. Intuitively, a node has high BC when it lies on many shortest paths between pairs of other nodes. Formally, BC of a node \( v \) is defined as

\[
\! c_B(v) = \frac{1}{n(n-1)} \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}},
\]

where \( n \) is the number of nodes, \( \sigma_{st} \) is the number of shortest paths between two nodes \( s \) and \( t \) and \( \sigma_{st}(v) \) is the number of these paths that go through node \( v \). Since it depends on all shortest paths, the exact
computation of BC is expensive: the best known algorithm [5] is quadratic in the number of nodes for sparse networks and cubic for dense networks, prohibitive for networks with hundreds of thousands of nodes. Many graphs of interest, however, such as web graphs or social networks, have millions or even billions of nodes and edges. For this reason, approximation algorithms [6,9,1] must be used in practice. In addition, many large graphs of interest evolve continuously, making the efficient recomputation of BC a necessity. In a previous work, we proposed the first two approximation algorithms [4] (IA for unweighted and IA_W for weighted graphs) that can efficiently recompute the approximate BC scores after batches of edge insertions or weight decreases. IA and IA_W are the only semi-dynamic algorithms that can actually be applied to large networks. The algorithms build on RK [19], a static algorithm with a theoretical guarantee on the quality of the approximation, and inherit this guarantee from RK. However, IA and IA_W target a relatively restricted configuration: only connected graphs and edge insertions/weight decreases.

Our Contributions. In this paper we present the first fully-dynamic algorithms (handling edge insertions, deletions and arbitrary weight updates) for BC approximation in weighted and unweighted undirected graphs. Our algorithms extend the semi-dynamic ones we presented in [4], while keeping the theoretical guarantee on the maximum approximation error. The transfer to fully-dynamic and disconnected graphs implies several additional problems compared to the restricted case we considered previously [4]. Consequently, we present the following intermediate results, all of which could be of independent interest. (i) We propose a new upper bound on the vertex diameter VD (i.e. number of nodes in the shortest path(s) with the maximum number of nodes) for weighted undirected graphs. This can improve significantly the one used in the RK algorithm [19] if the network’s weights vary in relatively small ranges (from the size of the largest connected component to at most twice the vertex diameter times the ratio between the maximum and the minimum edge weights). (ii) For both weighted and unweighted graphs, we present the first fully-dynamic algorithm for updating an approximation of VD, which is equivalent to the diameter in unweighted graphs. (iii) We extend our previous semi-dynamic BFS algorithm [4] to batches of both edge insertions and deletions. In our experiments, we compare our algorithms to recomputation with RK on both synthetic and real dynamic networks. Our results show that our algorithms can achieve substantial speedups, often several orders of magnitude on single-edge updates and are always faster than recomputation on batches of more than 1000 edges.

2 Related Work

2.1 Overview of Algorithms for Computing BC

The best static exact algorithm for BC (BA) is due to Brandes [5] and requires $\Theta(nm)$ operations for unweighted graphs and $\Theta(nm + n^2 \log n)$ for graphs with positive edge weights. The algorithm computes a single-source shortest path