K-Path Centrality: A New Centrality Measure in Social Networks

Tharaka Alahakoon Rahul Tripathi Nicolas Kourtellis Ramanuja Simha Adriana Iamnitchi

Department of Computer Science and Engineering, University of South Florida, Tampa FL

alahakoo@mail.usf.edu tripathi@cse.usf.edu nkourtel@mail.usf.edu rsimha@mail.usf.edu anda@cse.usf.edu

Abstract

This paper proposes an alternative way to identify nodes with high betweenness centrality. It introduces a new metric, κ-path centrality, and a randomized algorithm for estimating it, and shows empirically that nodes with high κ-path centrality have high node betweenness centrality. Experimental evaluations on diverse real and synthetic social networks show improved accuracy in detecting high betweenness centrality nodes and significantly reduced execution time when compared to known randomized algorithms.

Categories and Subject Descriptors G.2.2 [*Graph Theory*]: Graph algorithms and network problems

General Terms theory, algorithms, measurement

Keywords betweenness centrality, social network analysis, algorithms, experimental evaluation

1. Introduction

Social network analysis tools have been used in various fields such as physics, biology, genomics, anthropology, economics, organizational studies, psychology, and IT. The recent phenomenal growth of online social networks exacerbates the need for such tools that are scalable for applications in military, government, and for commercial purposes, to name only a few. Some of the relevant network metrics are local, such as degree centrality, while others capture global structural properties of the graph, such as the *betweenness centrality*. This important global graph metric is a centrality index that quantifies the importance of a node or an edge as a function of the number of shortest paths that traverse it.

Node betweenness centrality is relevant to problems such as identifying important nodes that control flows of information between separate parts of the network and identifying causal links to influence other entities behavior, such as genes in genomics or customers in marketing studies. Betweenness centrality has been used to analyze various social or general networks [13, 18, 19], to identify influential nodes surrounded by other influential nodes in social networks [14], and to measure network traffic in communication networks [21].

Node betweenness centrality, however, is computationally expensive. The best known algorithm for computing exact betweenness centrality of all vertices is Brandes' algorithm [5], which takes time $O(nm)$ on unweighted graphs and $O(nm + n^2 \log n)$ on weighted graphs. Some randomized algorithms for estimating betweenness centrality have been proposed in the literature [3, 6, 12], but the accuracy of these randomized algorithms decreases and the execution time increases considerably with the increase in the network size. Variants of betweenness centrality, such as flow betweenness [8] and random-walk betweenness [17], take computation time at least of the order nm. Thus, existing approaches for determining node betweenness centrality, which work well on small networks, are infeasible for networks with millions of nodes and edges.

We introduce a new approach for identifying highly influential nodes based on their betweenness centrality score, according to the following observations. First, we observe that the value of the betweenness centrality is irrelevant: it is the relative "importance" of nodes (as measured by betweenness centrality) that matters. Second, we observe that for the vast majority of applications, it is sufficient to identify categories of nodes of similar importance: thus, identifying the top 1% most important nodes is significantly more relevant than precisely ordering the nodes based on their relative betweenness centrality. Third, we observe that distant nodes in (social) networks are unlikely to influence each other [10]. Finally, we use the observation that influence may not be restricted to shortest paths [22]. Capturing these observations, we introduce a new distance-based centrality index called *κ-path centrality*, present a randomized algorithm for estimating it, and show empirically that nodes with high κ-path centrality have high betweenness centrality.

SNS'11, April 10, 2011, Salzburg, Austria.

The contributions of this paper are:
– A new node centrality measure, κ-path centrality, that is intuitively more appropriate for very large social networks because it limits graph exploration to a useful neighborhood of κ social hops around each node. The supporting intuition is twofold: first, in social networks, distant nodes are unlikely to influence each other, and thus the (long) shortest path that connects them is irrelevant in practice. Second, shortest paths are not always the choice for information transmission, as information may travel on less optimal paths.
– A randomized algorithm that estimates the κ-path centrality index for all nodes in a network of size n, up to an additive error of at most $n^{1/2+\alpha}$ with probability at least $1-1/n^2$ in time $O(\kappa^3 n^{2-2\alpha}\log n)$, where $\alpha \in [-1/2, 1/2]$ controls the tradeoff between accuracy and computation time.
– An empirical demonstration on a set of real and synthetic social networks that nodes with high κ-path centrality have high betweenness centrality.
– An experimental demonstration that the running time of our randomized algorithm for estimating κ-path centrality is orders of magnitude lower than the runtime of the best known algorithms for computing exact or approximate betweenness centrality, while maintaining higher accuracy, especially in very large networks.

2. Node Betweenness Centrality

Node betweenness centrality is a global centrality index that quantifies how much a vertex controls the information flow between all pairs of vertices in a graph. In this section, we review the formal definition of node betweenness centrality and briefly overview algorithms used in the experimental evaluation, that compute exact and approximate betweenness of all vertices in a graph.

2.1 Definition and Notations

Let $G = (V, E)$ be any (directed or undirected) graph, described by the set of vertices V and the set of edges E. The number of vertices (edges) in G is denoted by n (respectively, m). Let W be a non-negative weight function on the edges of G, assuming without loss of generality that each edge e of G has $W(e) = 1$ if G is unweighted. We define the *length* of any path ρ in G as the sum of weights of edges in ρ. A *shortest path* from s to t is a path of minimum length; its length is denoted by $d_G(s,t)$. Let $P_s(t)$ denote the *set of predecessors* of t on shortest paths from s to t. Let σ_{st} denote the *number of shortest paths* from s to t and, for any $v \in V$, let $\sigma_{st}(v)$ denote the number of shortest paths from s to t that go through v. Note that $d_G(s,s) = 0$, $\sigma_{ss} = 1$, and $\sigma_{st}(v) = 0$ if $v \in \{s,t\}$ or if v does not lie on any shortest path from s to t. The *betweenness centrality* index of a vertex v is the summation over all pairs of end vertices of the fractional count of shortest paths going through v.

DEFINITION 2.1. **(Betweenness Centrality [2, 9])** *For every vertex $v \in V$ of a weighted graph $G(V, E)$, the* betweenness centrality $\mathcal{C}_B(v)$ *of v is defined by*

$$\mathcal{C}_B(v) = \sum_{s \neq v} \sum_{t \neq v,s} \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

2.2 Brandes' Algorithm

Brandes' algorithm [5] for computing betweenness centrality defines the notion of the *dependency score* of any source vertex s on another vertex v as $\delta_{s\star}(v) = \sum_{t \neq s,v} \frac{\sigma_{st}(v)}{\sigma_{st}}$. Notice that the betweenness centrality $\mathcal{C}_B(v)$ of any vertex v can be expressed in terms of dependency scores as follows: $\mathcal{C}_B(v) = \sum_{s \neq v} \delta_{s\star}(v)$. The following recurrence relation on $\delta_{s\star}(v)$ is significant to Brandes' algorithm:

$$\delta_{s\star}(v) = \sum_{u : v \in P_s(u)} \frac{\sigma_{sv}}{\sigma_{su}}(1 + \delta_{s\star}(u)).$$

The algorithm takes as input a graph $G = (V, E)$ and an array W of edge weights and outputs the betweenness centrality $\mathcal{C}_B[v]$ of every $v \in V$. The running time of Brandes' algorithm on weighted graphs is $\mathcal{O}(nm + n^2 \log n)$ if the min-priority queue Q is implemented by a *Fibonacci heap*. Using BFS instead of Dijkstra's algorithm when the input graph is unweighted, the running time reduces to $\mathcal{O}(nm)$.

2.3 RA-Brandes Algorithm

Adapting the technique of Eppstein and Wang [7] for estimating the closeness centrality, Jacob et al. [12] and, independently, Brandes and Pich [6] proposed a randomized approximation algorithm for estimating the betweenness centrality of all vertices in any given graph. This algorithm, which we refer to as *Randomized-Approximate Brandes* or in short *RA-Brandes*, is different from Brandes' algorithm in only one main respect: Brandes' considers dependency scores $\delta_{s\star}(\cdot)$ of all n start vertices s, whereas RA-Brandes considers these scores of only a multiset $\mathcal{S}$ of $\Theta((\log n)/\epsilon^2)$ vertices. The multiset $\mathcal{S}$ is selected by choosing vertices uniformly at random with replacement. The estimated betweenness centrality $\widehat{\mathcal{C}}_B[v]$ of any vertex v is then defined as the scaled average of these scores: $\widehat{\mathcal{C}}_B[v] = \frac{n}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \delta_{s\star}(v)$.

The running time of RA-Brandes on unweighted graphs is $O(\frac{\log n}{\epsilon^2}(m+n))$, and on weighted graphs is $O(\frac{\log n}{\epsilon^2}(m + n\log n))$. The algorithm guarantees computing, for each vertex v, an approximation $\widehat{\mathcal{C}}_B[v]$ that is within $\mathcal{C}_B[v] \pm \epsilon n(n-1)$ with high probability $1 - 1/n^{\Omega(1)}$.

2.4 AS-Brandes Algorithm

Bader et al. [3] proposed a randomized algorithm for estimating the betweenness centrality of all vertices in any given graph. Their algorithm is based on the *adaptive sampling* technique of Lipton and Naughton [16] used in an algorithm for estimating the size of the transitive closure of a directed graph. The adaptive sampling technique requires selecting a multiset of start vertices by sampling vertices adaptively in the sense that the number of vertices chosen varies with

the information gained from each sample. Because of its similarity to Brandes' algorithm and application of adaptive sampling technique, we refer to this algorithm as *Adaptive-Sampling Brandes* or in short *AS-Brandes*.

AS-Brandes considers dependency scores of only a multiset $\mathcal{S}$ of at most T start vertices. It estimates betweenness centrality of any vertex v by noting how fast the sum of dependency scores for v reach a threshold cn, where $c \geq 2$ is supplied to the algorithm. To this end, for each vertex v, the algorithm maintains a running sum $RS[v]$ of dependency scores $\delta_{s\star}(v)$ for start vertices s and it records in a variable $k[v]$, the number of start vertices used for v until $RS[v]$ becomes greater than cn; $k[v]$ is set to T if $RS[v]$ never exceeds cn. The estimated betweenness centrality $\widehat{\mathcal{C}}_B[v]$ of any vertex v is then defined as the scaled average of these scores over $k[v]$ samples: $\widehat{\mathcal{C}}_B[v] = n \cdot \frac{RS[v]}{k[v]}$.

Since AS-Brandes considers only T start vertices while Brandes' considers n, AS-Brandes should be roughly $\Omega(n/T)$ times faster than Brandes'. It guarantees that, for $0 < \epsilon < 0.5$, if the betweenness centrality $\mathcal{C}_B[v]$ of a vertex v is at least n^2/t for some constant $t \geq 1$, then with probability at least $1 - 2\epsilon$, its estimated betweenness centrality $\widehat{\mathcal{C}}_B[v]$ is within $(1 \pm 1/\epsilon) \cdot \mathcal{C}_B[v]$ with ϵt samples of start vertices.

3. K-Path Centrality

As introduced in [17], the random-walk betweenness centrality is based on the traversal of the network with absorbing random walks. Assume the traversal of a message (e.g., news or rumor) originating from some source s over a network and intending to finally reach some destination t in the network along a path, and assume that each node in the network has only its own local view (i.e., has information only of its outgoing neighbors). Thus, when the message is at a current node v, the node v forwards the message based on its local view to one of its outgoing neighbors chosen uniformly at random. The message continues to travel in this manner until it reaches the destination node t, and then stops.

The notion of κ-path centrality is based on a similar assumption regarding the random traversal of a message from a source s. However, we make two further assumptions in order to reduce the computation time without deviating much from the above random walk model. First, we consider message traversals along simple paths only, i.e., paths in which vertices do not repeat. As non-simple paths do not correspond to the intuitive notion of ideal message traversals in a social network, their consideration in the computation of centrality indices is a noisy factor. To discount non-simple paths, we assume that each intermediate node v on a partially traversed path forwards the message to a neighbor chosen randomly, with probability inversely proportional to edge weights, from the current set of unvisited neighbors; the message traversal is assumed to stop if all the outgoing neighbors of the current node v already appear in the path up to v. Although choosing a random neighbor in this manner at each step requires the premise that the message carries the history of the path traversed so far, this premise is needed to express the average contribution of any simple path in the overall information flow and to efficiently simulate such random simple paths. Second, we assume that the message traversals are only along paths of at most κ links (edges), where κ is a parameter dependent on the network. It has been found in many studies on social networks that message traversals typically take paths containing few links [10], and so this seems to be a reasonable assumption in the context of social networks. Based on these assumptions, we define κ-path centrality:

DEFINITION 3.1. **(κ-Path Centrality)** *For every vertex v of a graph $G = (V, E)$, the κ-path centrality $\mathcal{C}_k(v)$ of v is defined as the sum, over all possible source nodes s, of the probability that a message originating from s goes through v, assuming that the message traversals are only along random simple paths of at most κ edges.*

3.1 Estimating K-Path Centrality

We present a randomized approximation algorithm for estimating the κ-path centrality of all vertices in any graph. The algorithm takes as input a graph $G = (V, E)$, a non-negative weight function W on the edges of G, and parameters $\alpha \in [-1/2, 1/2]$ and integer $\kappa = f(m, n)$, and runs in time $O(\kappa^3 n^{2-2\alpha} \ln n)$. For each vertex v, it outputs an estimate of $\mathcal{C}_\kappa(v)$ up to an additive error of $\pm n^{1/2+\alpha}$ with probability at least $1 - 1/n^2$. We refer to this algorithm as *Randomized-Approximate κpath* or in short *RA-κpath*.

```
Input: Graph G = (V, E), Array W of edge weights, α ∈ [−1/2, 1/2],
       and integer κ
Output: Array Ĉ_κ of κ-path centrality estimates
begin
    foreach v ∈ V do
        count[v] ← 0; Explored[v] ← false;
    end
    /* S is a stack and n = |V| */
    T ← 2κ²n^{1−2α} ln n; S ← ∅;
    for i ← 1 to T do
        /* simulate a message traversal from s containing ℓ edges */
        s ← a vertex chosen uniformly at random from V;
        ℓ ← an integer chosen uniformly at random from [1, κ];
        Explored[s] ← true; push s to S; j ← 1;
        while (j ≤ ℓ and ∃(s, u) ∈ E s.t. !Explored[u]) do
            v ← a vertex chosen randomly from {u | (s, u) ∈ E
                and !Explored[u]} with probability
                proportional to 1/W(s, v);
            Explored[v] ← true; push v to S;
            count[v] ← count[v] + 1;
            s ← v; j ← j + 1;
        end
        /* reinitialize Explored[v] to false */
        while S is nonempty do
            pop v ← S; Explored[v] ← false;
        end
    end
    foreach v ∈ V do
        Ĉ_κ[v] ← κn · count[v]/T;
    end
    return Ĉ_κ;
end
```

The algorithm performs $T = 2\kappa^2 n^{1-2\alpha} \ln n$ iterations (the expression for T comes from the analysis of the algorithm).

In each iteration, a start vertex $s \in V$ and a walk length $\ell \in [1, \kappa]$ are chosen uniformly at random, and then a random walk consisting of ℓ edges from s is performed that essentially simulates a message traversal from s in G using the assumption made in Definition 3.1. The number of times any vertex v is visited over all the random walks is recorded in a variable $\mathrm{count}[v]$. The estimated κ-path centrality $\widehat{\mathcal{C}}_\kappa[v]$ of any vertex v is then defined as the scaled average of the times v is visited over T walks: $\widehat{\mathcal{C}}_\kappa[v] = \kappa n \cdot \frac{\mathrm{count}[v]}{T}$.

THEOREM 3.2. *The algorithm* RA-κpath *runs in time* $O(\kappa^3 n^{2-2\alpha} \log n)$*, and outputs, for each vertex* v*, an estimate* $\widehat{\mathcal{C}}_\kappa[v]$ *of* $\mathcal{C}_\kappa[v]$ *up to an additive error of* $\pm n^{1/2+\alpha}$ *with probability* $1 - 1/n^2$.

The proof, omitted here due to space constraint, can be found in the full version of this paper [1].

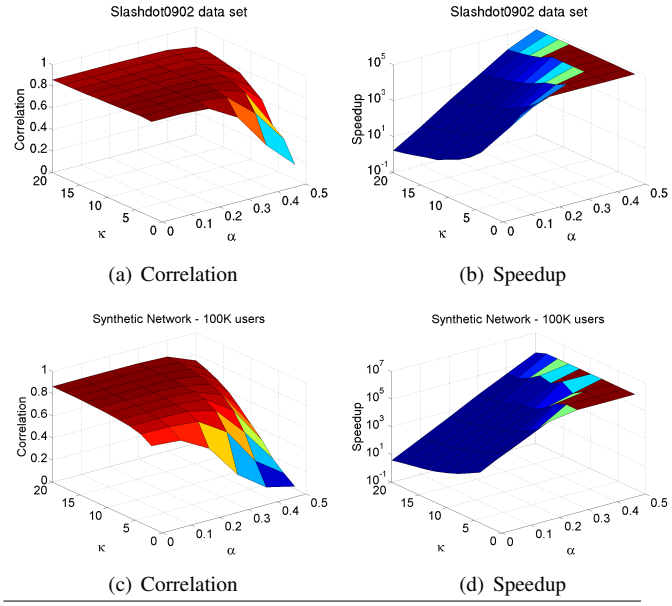
4. Experimental Evaluation

In order to assess the performance of RA-κpath, we compare its accuracy and running time with that of Brandes', RA-Brandes and AS-Brandes. We performed experiments on both real and synthetic social networks. The real networks were selected from various online sources to cover a wide range of application domains and scales, and are presented in Table 1. In order to test the performance of RA-κpath on social graphs that maintain consistent social properties with increase in their size, we created a set of synthetic networks using a synthetic social network generator based on the model in [20]. We used this generator to produce networks with $1K$, $10K$, $50K$ and $100K$ nodes. All experiments were done on a cluster with identical nodes with two AMD Opteron processors at 2.2 GHz and 4GB RAM.

4.1 Performance Metrics

For evaluating the accuracy of κ-path centrality in estimating the relative importance of a node as per the betweenness centrality index, we choose two accuracy metrics. The first metric, called RA-κpath correlation, is the correlation between the approximate κ-path centrality values computed by RA-κpath and the exact betweenness centrality values computed by Brandes' algorithm. We apply the same approach to measure the accuracy of the two approximation algorithms, RA-Brandes and AS-Brandes, in estimating the node betweenness centrality. We refer to these metrics as RA-Brandes and AS-Brandes correlation, respectively.

The second accuracy metric captures the ability to identify the top N% high centrality betweenness nodes. We measure the percentage of the overlap between the top $N\%$ nodes as returned by a particular approximation algorithm (out of RA-κpath, RA-Brandes, and AS-Brandes) and the top $N\%$ nodes as identified by Brandes'. We refer to this metric as top $N\%$ RA-κpath, top $N\%$ RA-Brandes, and top $N\%$ AS-Brandes, respectively.



(a) Correlation (b) Speedup

(c) Correlation (d) Speedup

Figure 1. RA-κpath correlation and RA-κpath speedup in a real social network and a synthetic social network.

For evaluating the run time performance, we determine the ratio of the execution time of each of the three approximation algorithms over our implementation of Brandes'. We refer to this metric as RA-κpath speedup, RA-Brandes speedup, and AS-Brandes speedup, respectively.

4.2 Comparison with Brandes' Algorithm

We compute the correlation and speedup of RA-κpath with respect to Brandes' for the real and synthetic social networks for κ varying from 2 to 20 in increments of 2 and α varying from 0 to 0.5 in increments of 0.1. Due to space limitation, we present the speedup and correlation results of one real and one synthetic social network in Figure 1.

We found that, as α decreases, the correlation of RA-κpath increases and its speedup decreases (both with respect to Brandes'). The maximum correlation achieved is in the range of 0.70 to 0.95 and the maximum speedup achieved is in the range of 10^2 to 10^6, depending on the values of α, κ, and the network size. A general observation from these results is that we can achieve a near optimal performance of RA-κpath in both correlation and speedup performance metrics when, for a network of n vertices and m edges, α is set to 0.2 and κ is set to $\ln(n+m)$. We use these values of α and κ in the following experiments that compare the performance of RA-κpath with RA-Brandes and AS-Brandes.

4.3 Comparison with RA- and AS-Brandes

Figures 2(a) and 2(b) show the correlation and speedup results of RA-κpath, RA-Brandes, and AS-Brandes with respect to Brandes' on real networks. These results were obtained for $\epsilon = 0.5$ for RA-Brandes, and $s = 20$ and $c = 5$ for AS-Brandes. This choice of parameters for AS-Brandes has also been used in [3]. The results demonstrate the superiority

Real Networks	Number of Vertices	Number of Edges	Directed/ Undirected	Weighted/ Unweighted	Source	Network Type
Kazaa	2,424	13,354	Undirected	Weighted	[11]	File sharing
SciMet	2,729	10,416	Undirected	Unweighted	[4]	Citation
CA-CondMat	23,133	186,936	Undirected	Unweighted	[15]	Co-authorship
Cit-HepPh	34,546	421,578	Directed	Unweighted	[15]	Citation
Email-Enron	36,692	367,662	Undirected	Unweighted	[15]	Email communication
Soc-Epinions1	75,879	508,837	Directed	Unweighted	[15]	Social
Soc-Slashdot0922	82,168	948,464	Directed	Unweighted	[15]	Social

Table 1. Summary information of the real networks used in this study.

Network / nodes	RA-K	RA-B	AS-B	RA-K	RA-B	AS-B	RA-K	RA-B	AS-B
	1%	1%	1%	5%	5%	5%	10%	10%	10%
Kazaa / 2.4K	**79.2**	58.3	58.3	**72.7**	64.5	66.9	72.3	79.3	**79.8**
SciMet / 2.7K	**85.2**	48.1	44.4	**77.9**	66.2	64.0	**76.5**	70.2	69.1
CA-CondMat / 23.1K	**74.5**	48.1	48.9	**76.6**	73.2	72.4	76.9	**81.8**	**81.8**
Cit-HepPh / 34.5K	**71.3**	53.9	47.8	**66.1**	61.2	61.4	66.3	68.9	**69.7**
Email-Enron / 36.7K	75.1	**79.0**	76.8	63.8	88.5	**89.1**	65.6	**92.7**	**92.7**
Soc-Epinions1 / 75.9K	**80.6**	70.2	71.0	75.0	**90.2**	90.0	72.7	94.8	**95.0**
Soc-Slashdot0922 / 82.2K	**85.9**	67.4	67.7	85.2	**88.8**	88.3	78.4	**92.1**	92.0
synthetic / 1K	**83.0**	70.0	65.0	**82.4**	70.6	69.6	**77.3**	70.1	69.7
synthetic / 10K	**88.3**	58.0	58.4	**82.4**	67.8	67.8	**78.7**	78.5	78.5
synthetic / 50K	**86.6**	61.6	60.8	**81.7**	76.5	77.0	77.5	83.5	**83.8**
synthetic / 100K	**87.5**	61.0	60.4	**81.4**	79.7	79.8	77.1	84.4	**84.6**

Table 2. Percentage overlap of the top $N\%$ nodes as computed by the three algorithms with respect to the exact betweenness centrality values. The speedups of the three algorithms were first matched to set the parameter values and then the algorithms with the parameter values set were ran for their percentage overlap of the top $N\%$ nodes. The values in bold denote the highest in the respective (N-value) category.

of RA-κpath over the other algorithms in both performance metrics for most of the real networks examined.

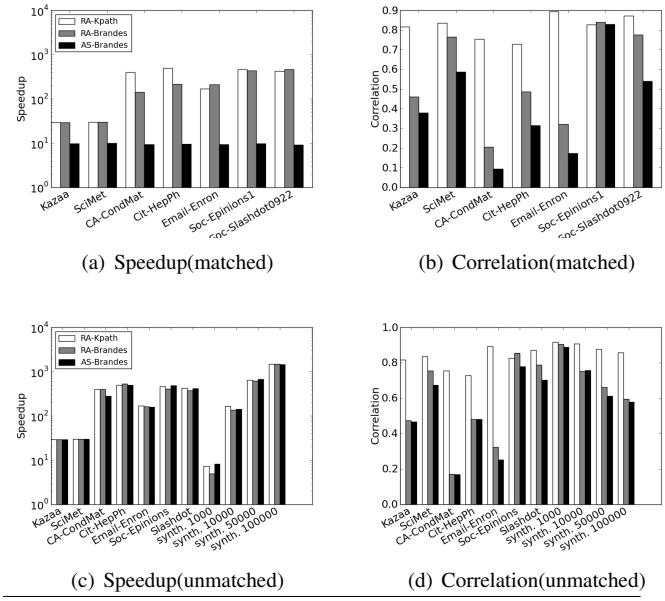
However, we believe that the choice of parameter values $\epsilon = 0.5$ and $s = 20$ is not suitable for the sizes of the networks we examined: For example, in [3] where these values for parameters s and c are used in AS-Brandes, the largest networks evaluated have $< 10K$ nodes and $< 50K$ edges. For this reason, we decided to match the speedups of the three algorithms in order to infer less biased parameter values for AS-Brandes and RA-Brandes. We thus performed several experiments with various values of ϵ (for RA-Brandes) and s (for AS-Brandes), and settled on the following heuristic that helped us to closely match the speedups of the three algorithms with respect to Brandes' algorithm:
– $\epsilon = 2 \times ((\text{RA-}\kappa\text{path speedup}) \times \ln(n)/n)^{1/2}$ and
– $s = 2 \times (\text{RA-}\kappa\text{path speedup})$
Intuition for this choice of ϵ: RA-Brandes considers dependency scores of $\Theta((\ln n)/\epsilon^2)$ start vertices while Brandes' considers these scores of n start vertices, and so RA-Brandes speedup can be estimated to $\Theta(n\epsilon^2/\ln n)$; setting this estimate to RA-κpath speedup yields the above expression for ϵ. The intuition for the choice of s has similar reasoning.

Figures 2(c) and 2(d) show that the correlations of RA-κpath, RA-Brandes, and AS-Brandes vary widely when their speedups are matched. In many cases RA-κpath outperforms the other two algorithms by a factor of 0.1 to 0.6, depending on the network. This degradation of correlation in the case of RA-Brandes and AS-Brandes is expected as these algorithms contain an inherent trade-off between speedup and accuracy. For the synthetic social networks, the values presented are averaged over ten executions on ten independently generated networks for each size.

Table 2 shows top $N\%$ RA-κpath (RA-K), top $N\%$ RA-Brandes (RA-B), and top $N\%$ AS-Brandes (AS-B), for the real and synthetic social networks and for $N = 1$, 5, and 10. These results are obtained after the algorithms were matched in speedup as mentioned earlier. Overall, RA-κpath outperforms the other two algorithms by a factor of 20% to 30%, in identifying the top 1% most important nodes in all the sizes and types of networks. There is a decrease in the performance of RA-κpath when we consider the top 5% and top 10%. This performance deterioration for large N could be due to the arbitrary ordering of low κ-path centrality nodes arising from closeness in their values.

5. Summary

In this paper, we introduced a new graph centrality index called κ-path centrality and presented a randomized algo-



(a) Speedup(matched) (b) Correlation(matched)

(c) Speedup(unmatched) (d) Correlation(unmatched)

Figure 2. (a,b) Unmatched speedup and respective correlation of RA-κpath, RA-Brandes, and AS-Brandes using default parameters ($\alpha = 0.2$, $\kappa = \ln(n+m)$, $\epsilon = 0.5, s = 20$, and $c = 5$). (c) Matched speedup of the three algorithms to set parameter values, (d) respective correlation using the set values.

rithm RA-κpath for estimating its value for all vertices. Our experimental evaluation demonstrates that this centrality metric can be used to scalably estimate the relative importance of nodes as per the betweenness centrality index: the correlation between the two centrality indices reaches from 0.70 to 0.95 for all network sizes for a speedup gain of up to 6 orders of magnitude for networks with more than $10,000$ nodes. Our experiments show that RA-κpath is very effective in identifying the top 1% or the top 5% nodes in the exact betweenness score, outperforming previously known approximate betweenness centrality algorithms AS-Brandes and RA-Brandes. The near optimal performance of RA-κpath in both correlation and speedup performance metrics can be achieved when its parameters are set to $\alpha = 0.2$ and $\kappa = \ln(n+m)$, where n and m are the number of nodes and the number of edges in the network, respectively.

Through our experiments, we have shown that κ-path centrality can be used as an alternative to node betweenness centrality since (a) κ-path centrality closely models the spread of information in a network and allows to quantify the influence of any node in the network and (b) the speedup performance of RA-κpath for estimating κ-path centrality surpasses those achieved by existing methods of computing exact or approximate betweenness centrality values.

Acknowledgments

This research was partially supported by the NSF under Grants No. CNS-0831785 and CNS-0952420. We acknowledge using services provided by Research Computing, USF.

References

[1] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi. K-path centrality: A new centrality measure in social networks. http://www.csee.usf.edu/∼tripathi/kpath-centrality.pdf.

[2] J. Anthonisse. The rush in a directed graph. Technical Report BN9/71, Stichting Mathematisch Centrum, Amsterdam, Netherlands, 1971.

[3] D. Bader, S. Kintali, K. Madduri, and M. Mihail. Approximating betweenness centrality. In *WAW*, pages 124–137, 2007.

[4] V. Batagelj and A. Mrvar. Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/, 2006.

[5] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[6] U. Brandes and C. Pich. Centrality estimation in large networks. *I. J. of Bifurcation and Chaos*, 17(7):2303–2318, 2007.

[7] D. Eppstein and J. Wang. Fast approximation of centrality. *J. of Graph Algorithms & Applications*, 8(1):39–45, 2004.

[8] C. Freeman, S. Borgatti, and D. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, 1991.

[9] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[10] N. Friedkin. Horizons of observability and limits of informal control in organizations. *Social Forces*, 62(1):57–77, 1983.

[11] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *INFOCOM*, pages 952–963, 2004.

[12] R. Jacob, D. Koschützki, K. Lehmann, L. Peeters, and D. Podehl. Algorithms for centrality indices. In *Network Analysis*, pages 62–82. Springer-Verlag *LNCS #3418*, 2005.

[13] H. Jeong, S. Mason, A. Barabási, and Z. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41, 2001.

[14] G. Kahng, E. Oh, B. Kahng, and D. Kim. Betweenness centrality correlation in social networks. *Phys. Rev. E*, 67:01710–1, 2003.

[15] J. Leskovec. Stanford large network dataset collection. http://snap.stanford.edu/data/, 2009.

[16] R. Lipton and J. Naughton. Estimating the size of generalized transitive closures. In *VLDB*, pages 165–171, 1989.

[17] M. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[18] M. Ortiz, J. Hoyos, and M. Lopez. The social networks of academic performance in a student context of poverty in mexico. *Social Networks*, 26(2):175–188, 2004.

[19] Y. Said, E. Wegman, W. Sharabati, and J. Rigsby. Social networks of author-coauthor relationships. *Computational Statistics and Data Analysis*, 52:2177–2184, 2008.

[20] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Zhao. Measurement-calibrated graph models for social network experiments. In *WWW*, pages 861–870, 2010.

[21] B. Singh and N. Gupte. Congestion and decongestion in a communication network. *Phys. Rev. E*, 71(5):055103, 2005.

[22] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11:1–37, 1989.