Scalable Similarity Estimation in Social Networks: Closeness, Node Labels, and Random Edge Lengths

Edith Cohen
Microsoft Research SVC
edith@cohenwang.com

Daniel Delling
Microsoft Research SVC
dadellin@microsoft.com

Fabian Fuchs*
KIT, Germany
fabian.fuchs@kit.edu

Andrew V. Goldberg
Microsoft Research SVC
goldberg@microsoft.com

Moises Goldszmidt
Microsoft Research SVC
moises@microsoft.com

Renato F. Werneck
Microsoft Research SVC
renatow@microsoft.com

ABSTRACT

Similarity estimation between nodes based on structural properties of graphs is a basic building block used in the analysis of massive networks for diverse purposes such as link prediction, product recommendations, advertisement, collaborative filtering, and community discovery. While local similarity measures, based on properties of immediate neighbors, are easy to compute, those relying on global properties have better recall. Unfortunately, this better quality comes with a computational price tag. Aiming for both accuracy and scalability, we make several contributions. First, we define *closeness similarity*, a natural measure that compares two nodes based on the similarity of their relations to all other nodes. Second, we show how the all-distances sketch (ADS) node labels, which are efficient to compute, can support the estimation of closeness similarity and shortest-path (SP) distances in logarithmic query time. Third, we propose the randomized edge lengths (REL) technique and define the corresponding REL distance, which captures both path length and path multiplicity and therefore improves over the SP distance as a similarity measure. The REL distance can also be the basis of closeness similarity and can be estimated using SP computation or the ADS labels. We demonstrate the effectiveness of our measures and the accuracy of our estimates through experiments on social networks with up to tens of millions of nodes.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms**]: Miscellaneous; G.2.2 [**Discrete Mathematics**]: Graph Theory; H.2.8 [**Database Management**]: Database Applications—*Data Mining*; H.3.2 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Clustering

Keywords

Closeness Similarity; Social Networks; All-Distances Sketches; Random Edge Lengths

*Intern at Microsoft Research during this work.

COSN'13, October 7–8, 2013, Boston, Massachusetts, USA.

http://dx.doi.org/10.1145/2512938.2512944.

1. INTRODUCTION

Methods that estimate the similarity of two nodes based solely on the graph structure are the heart of approaches for analyzing massive graphs and social networks [4, 21, 24, 27, 30]. These methods constitute the basic building blocks in algorithms for link prediction (friend recommendations), collaborative filtering, product placement, advertisement, and community discovery, to name a few.

The underlying *similarity measures* can be classified as *local* or *global*. Local measures are based only on the relation of immediate neighborhoods and include the number of common neighbors [28] and Adamic-Adar [4], which weighs each common neighbor inversely to the logarithm of its degree. Global measures are defined with respect to the global graph structure and include RWR (random walk with a specified restart probability) [34, 27], which extends hitting time and PageRank [10]; KATZ [24], which sums the number of distinct paths, where path contribution decreases exponentially with its length; SimRank [22], which is a recursively defined similarity measure based on neighbor similarity; resistance distance (viewing network as an electric circuit) and the related commute time (symmetric hitting time) [9]; the shortest-path (SP) distance; and the minimum cut. We refer the reader to [27] for an overview.

Local measures are fairly effective for some applications, but global measures can assign meaningful similarity scores to pairs that are more than two hops apart, and therefore have a better recall [27]. This capability comes at a price: global measures are often computationally more expensive, making them infeasible for application to networks and graphs with tens to hundreds of million of nodes, which are the norm in today's social networks, such as Twitter, Facebook, or LinkedIn.

Aiming for both accuracy and scalability, we introduce novel measures based on global properties and novel estimation techniques. Our approach consists of three components.

The first is a definition of the *closeness similarity* between nodes, which is based on the similarity of their distances to all other nodes in the network. Closeness similarity is specified with respect to three functions (*distance*, *decay*, and *weight*) and naturally extends the classic closeness centrality [31, 32] and several measures based on local and global properties.

The second component, *all-distances sketch* (ADS) labels, is a sketching technique that assigns a label to each node in the graph. ADS labels were initially developed for estimating the number of nodes reachable from (or within a certain distance of) a given node [6, 7, 12, 16, 19, 29], then extended to estimate closeness centrality [13, 15]. Although we need to compute an ADS label for each node in a graph, the above papers show that this can be

done efficiently, with a logarithmic total number of traversals per edge. These computations can also be done offline, leading to extremely efficient (online) queries, consisting of simple operations on the (small) ADS labels. We show here that the ADS labels of two nodes can be used effectively to estimate both their SP distance and their closeness similarity.

As the third component in our approach, we introduce the *randomized edge lengths* (REL) technique. We define the *REL distance*, which is the expected SP distance with respect to this randomization. While both shorter path length and multiplicity strongly correlate with similarity in social networks, the SP distance does not account for multiplicity. The REL distance compensates for this semantic deficiency of SP distances while retaining the computational advantages of this benchmark similarity measure [27]. Another property of the REL distance that is not shared by the SP distance but holds for popular global measures (such as RWR, KATZ, and resistance distance) is that it can be used to identify situations in which the similarity value of two nodes highly depends on the presence of a third node (a "critical hub"). This is important because a critical hub between nodes may overstate actual similarity and also the number of pairs with respect to which a node is a critical hub can be used as a measure of betweenness centrality. Lastly, REL can be integrated with both ADSs (for efficiency) and closeness similarity (for accuracy).

We demonstrate both the effectiveness and the scalability of our approaches through a large-scale experimental study on three real social networks. The first two, DBLP and arXiv, are undirected networks representing co-authorships. The third, the so-called mention graph in Twitter, is directed and has tens of millions of nodes. We can answer arbitrary similarity queries (even global ones) on real social networks of such scale in a few *microseconds*, taking the full structure of the graph into account. Our experiments compare our approach against well-established algorithms representing both local (Adamic-Adar) and global (RWR) measures of similarity. As proxy for ground truth we use metadata associated with the networks (the text in paper titles, abstracts, and tweets), and use standard similarity measures from information retrieval to evaluate the degree of similarity between the nodes in the network. For validation, we also consider one synthetic *small world* network [26], for which we actually hold the ground truth.

In summary, the contributions of our work are:

- A definition of closeness similarity in graphs with a suitable parametrization, capable of capturing both local and global definitions.
- The application of all-distances sketch (ADS) labels to estimate distances and closeness similarities with theoretical performance guarantees.
- The application of randomized edge lengths (REL) to increase the accuracy of the similarity estimation.
- Experiments demonstrating the scalability and effectiveness of our approach on large-scale networks with comparisons to other representative approaches.

The remainder of this paper is organized as follows. We give a precise definition of ADSs and related notions in Section 2. Section 3 discusses how ADSs can be used to obtain distance estimates, and gives novel theoretical bounds for their accuracy. Section 4 introduces the notion of closeness similarity, studies two natural special cases (based on SP distances and Dijkstra ranks), and shows how they can be efficiently approximated using ADSs. Section 5 introduces the concept of randomized edge lengths (REL). The results of our experimental evaluation are provided in Section 6. We conclude with final remarks in Section 7.

2. PRELIMINARIES

We consider both directed and undirected networks. For nodes v, u, we use d_{vu} to denote the shortest-path (SP) distance from v to u. Let π_{vu} denote the *Dijkstra rank* of u with respect to v, defined as its position in the list of nodes ordered by increasing distance from v. For simplicity, we assume consistent tie-breaking to make distances distinct.

For two nodes u, v, we use the notation $\Phi_{<u}(v) = \{j | \pi_{vj} < \pi_{vu}\}$ for the set of nodes that are closer to v than u is. For $d \geq 0$ and node v, $N_d(v)$ (the d neighborhood of v) is the set of nodes of distance at most d from v, and $N_{<d}(v)$ is the set of nodes that are of distance less than d from v. We use $n_d(v) \equiv |N_d(v)|$.

For directed graphs, we distinguish between *forward* and *backward* Dijkstra ranks ($\overrightarrow{\pi}_{vi}$, $\overleftarrow{\pi}_{vi}$) and neighborhoods ($\overrightarrow{\Phi}_{<u}(v)$, $\overleftarrow{\Phi}_{<u}(v)$,$\overrightarrow{N}_d(v)$,$\overleftarrow{N}_d(v)$). The forward relations are with respect to the graph with edges oriented forward. The backward relations are with respect to the graph with edges reversed.

For a numeric function $r : X \to [0, 1]$ over a set X, the function $\mathrm{k}^{\mathrm{th}}_r(X)$ returns the k-th smallest value in the range of r on X. If $|X| < k$ then we define $\mathrm{k}^{\mathrm{th}}_r(X) = 1$.

The *all-distances sketch* (ADS) labels are defined with respect to a random rank assignment to nodes such that $\forall v, r(v) \sim U[0, 1]$, i.e, they are independently drawn from the uniform distribution on $[0, 1]$:

$$\mathrm{ADS}(v) = \{(u, d_{vu}) \mid r(u) < \mathrm{k}^{\mathrm{th}}_r(\Phi_{<u}(v))\} .$$

In other words, a node u belongs to $\mathrm{ADS}(v)$ if u is among the k nodes with lowest rank r within the ball of radius d_{vu} around v. (For simplicity, we abuse notation and often interpret $\mathrm{ADS}(v)$ as a set of nodes, even though it is actually a set of *pairs*, each consisting of a node and a distance.) Since the inclusion probability of a node is inversely proportional to its Dijkstra rank, the expected size of $\mathrm{ADS}(v)$ is $\mathsf{E}[|\mathrm{ADS}(v)|] \leq k \ln n$, where n is the number of nodes reachable from v [12, 16]. A detailed example is given in the appendix.

Moreover, a set of ADSs (one for each node) with respect to the same r can be computed efficiently, using $O(k \ln n)$ traversals per edge in expectation. The two proposed ADS computations (see overview in [13]) are based on performing pruned Dijkstra-based single-source shortest path computations [12, 16] or (for unweighted edges) dynamic programming [29, 19, 6, 7].

For a node $u \in \mathrm{ADS}(v)$, we define

$$p_{vu} = \mathrm{k}^{\mathrm{th}}_r(\Phi_{<u}(v)) \tag{1}$$

to be the k-th smallest rank value amongst nodes that are closer to v than u is. By definition, $\mathrm{k}^{\mathrm{th}}_r(\Phi_{<u}(v)) = \mathrm{k}^{\mathrm{th}}_r(\{i \in \mathrm{ADS}(v) | d_{vi} < d_{vu}\})$ (the k-th smallest amongst nodes in $\mathrm{ADS}(v)$ that are closer to v than u is). Therefore, p_{vu} can be computed from $\mathrm{ADS}(v)$ for all $u \in \mathrm{ADS}(v)$. The value p_{vu} is the inclusion probability $\Pr[i \in \mathrm{ADS}(v)]$, conditioned on fixing r on all nodes in $\Phi_{<u}(v)$ (but not fixing $r(u)$). Under this conditioning, i is included if and only if $r(i) < p_{vu}$. Since $r(i) \sim U[0, 1]$, this happens with probability p_{vu}. We use p_{vu} to obtain estimators for node centrality and pairwise relations.

Another useful function that can be computed from $\mathrm{ADS}(v)$ with respect to a distance $x > 0$ is the threshold rank, defined as

$$\tau_v(x) = \mathrm{k}^{\mathrm{th}}(N_{<x}(v)). \tag{2}$$

The threshold rank is the maximum rank value that suffices for a node at distance x from v to be included in $\mathrm{ADS}(v)$, fixing the ranks of all other nodes. When $u \in \mathrm{ADS}(v)$, then $p_{vu} = \tau_v(d_{vu})$. We use τ in the analysis of our similarity estimators. The inverse function

$$\tau_v^{-1}(y) = \max\{d_{vi} \mid \mathrm{k}_r^{\mathrm{th}}(\Phi_{<i}(v)) > y\} \quad (3)$$

is the maximum distance for which the threshold rank is larger than y. This function gives a lower bound on the distance d_{vi} for a node $i \notin \mathrm{ADS}(v)$. It also allows us to identify nodes that are far away from a set of nodes.

For directed graphs, we distinguish between the forward and backward ADSs: $\overrightarrow{\mathrm{ADS}}(v)$ is computed with respect to $\overrightarrow{\Phi}_v$ and $\overleftarrow{\mathrm{ADS}}(v)$ is computed with respect to $\overleftarrow{\Phi}_v$. Accordingly, we use the forward and backward notation with the functions p and τ ($\overrightarrow{p}_{uv}$, $\overleftarrow{p}_{uv}$, $\overrightarrow{\tau}$, $\overleftarrow{\tau}$).

As a relative error measure for the quality of our estimators we use the *coefficient of variation (CV)*, defined as the ratio of root of square error to mean.

3. DISTANCE ESTIMATION

We explore the use of ADSs as *distance labels*. Distance labels enable the computation of the SP distance d_{ij} from the labels of the nodes i and j. Node labels that have the same format as ADSs (2-hop labels [14]) have been successfully used for exact distances in road networks [3] and medium-size unweighted social graphs [5]. These exact labels, however, are much larger and much more expensive to compute than ADSs.

We show here how ADSs can be used to obtain bounds on the distance. Without loss of generality, we use the directed notation. If we are lucky and $j \in \overrightarrow{\mathrm{ADS}}(i)$ or $i \in \overleftarrow{\mathrm{ADS}}(j)$, we can determine the exact distance d_{ij} from $\overrightarrow{\mathrm{ADS}}(i)$ or $\overleftarrow{\mathrm{ADS}}(j)$. Otherwise, we can obtain upper and lower bounds, which we denote by $\overline{d}_{ij}$ and $\underline{d}_{ij}$, respectively. The upper bound is obtained by treating the ADSs as 2-hop labels [3, 5, 14], namely considering the shortest path through a node in $\overrightarrow{\mathrm{ADS}}(i) \cap \overleftarrow{\mathrm{ADS}}(j)$:

$$\overline{d}_{ij} \leftarrow \min\{d_{iv} + d_{vj} \mid v \in \overrightarrow{\mathrm{ADS}}(i) \cap \overleftarrow{\mathrm{ADS}}(j)\} \quad (4)$$

To study the quality of the upper bound given by Equation (4), we consider its *stretch*, defined as the ratio between the bound and the actual SP distance. We bound the stretch of (4) on undirected graphs as a function of the parameter k. As our first contribution, we show that the bounds on stretch, query, space, and construction time asymptotically match those of the Thorup-Zwick distance oracles [33].

THEOREM 3.1. *On undirected graphs, with constant probability,*

$$\overline{d}_{vu} \leq \left(2\left\lceil\frac{\log n}{\log k}\right\rceil - 1\right) d_{vu}.$$

PROOF. Let $d \equiv d_{vu}$. We have the relation $N_d(u) \subset N_{2d}(v) \subset N_{3d}(u) \subset N_{4d}(v) \subset \ldots$ Consider the ratio $\frac{n_{id}(v)}{n_{(i+1)d}(u)}$ for $i \geq 1$. We look at the smallest i where $\frac{n_{id}(v)}{n_{(i+1)d}(u)} > c/k$. When this holds, we are very likely to have a common node that is in $\mathrm{ADS}(v) \cap \mathrm{ADS}(u) \cap N_{id}(v)$. Simply, the k smallest ranked nodes in $N_{(i+1)d}(u)$ are a uniform sample, so one of them hits $N_{id}(v)$ with probability $1 - (1 - c/k)^k \approx 1 - 1/e^c$. Finally, since there are at most n nodes, we must have $i \leq \log_{k/c} n$. □

This means the stretch is $\log n$ if we choose k to be constant, and $2a-1$ if we pick $k = n^{1/a}$ (for some integer a). The bound can be slightly tightened by looking at the maximum h such that, for some increasing $n_1 \leq n_2 \leq \ldots \leq n_h$, $k\sum_i n_i/n_{i+1} < 1$ (bounding the sum rather than the minimum ratio). This would give as stretch $\log(n)/\log\log(n)$ with fixed k.

A naive implementation of ADS intersection takes $O(k \ln n)$ time and works well in practice for small k. To improve query time when k is large (while still guaranteeing the same bound on the stretch), we further observe that the smallest ranked node in $N_{id}(v)$ is likely to be one of the k smallest in $N_{(i+1)d}(u)$. Therefore, it suffices to test for inclusion in $\mathrm{ADS}(u)$ only the nodes in $\mathrm{ADS}(v)$ that would have been included when $k = 1$. This would reduce query time to $O(\ln n)$. We note that asymptotic query times can be further reduced by a more careful use of the ADSs, but the details are mainly of theoretical interest and outside the scope of this work.

A clear advantage of the ADSs over the Thorup-Zwick summaries is that they are also "oracles" for other powerful measures: closeness similarity, as we show here, neighborhood sizes [12], and closeness centralities [13, 15]. In practice, the distance estimates obtained by the ADSs are much better than these worst-case bounds for arbitrary graphs. As we show in Section 6, we observe very small stretch even for directed graphs. We now provide a theoretical justification for these observations by considering structural properties of social networks.

For two nodes i, j we are likely to get a good upper bound if a good "intermediate" vertex is likely to be present at the intersection of the ADSs. More precisely, suppose nodes v_h are ordered by increasing $d_h \equiv d_{iv_h} + d_{v_hj}$, i.e., by the upper bound they yield on the distance. Note that

$$\begin{aligned}\Pr[v_h \in \mathrm{ADS}(i) \cap \mathrm{ADS}(j)] &\geq \\ \min\left\{1, \frac{k}{|\overrightarrow{\Phi}_i(v_h) \cup \overleftarrow{\Phi}_j(v_h)|}\right\} &\geq \frac{k}{\overrightarrow{\pi}_{iv_h} + \overleftarrow{\pi}_{jv_h}}.\end{aligned}$$

This is because a sufficient condition for v_h to be in the intersection is that it is k-th in the random permutation induced on v_h and the nodes $\overrightarrow{\Phi}_i(v_h) \cup \overleftarrow{\Phi}_j(v_h)$. These events are negatively correlated for different nodes. Therefore, by summing over potential intermediate nodes, we are $(1 - 1/e^c)$ likely to get an upper bound at most d_s when

$$k\sum_{h=1}^{s} \frac{1}{\max\{\overrightarrow{\pi}_{iv_h}, \overleftarrow{\pi}_{jv_h}\}} \geq c.$$

A sufficient condition to obtain an upper bound d_s with probability $1 - 1/e^c$ is that there is $x < d_s$ so that the Jaccard similarity of $N_x(i)$ and $N_{d_{ij}-x}(j)$ is at least c/k.

Qualitatively, the stretch is lower when there are many potential "hub" nodes v that lie on almost shortest paths ($d_{ij} \leq (1+\epsilon)(d_{iv} + d_{vj})$). Since the presence of multiple short paths is an indicator of similarity, this suggests that the approximate distance might be better in capturing some aspects of similarity than the exact distance.

3.1 Better Bounds

Our experiments show that simply applying Equation (4) (even for fairly small values of k) is enough to obtain very accurate results for social networks. For even better upper bounds, one can use more information and trade accuracy for computation time. For example, the formula

$$\begin{aligned}\overline{d}_{ij} \leftarrow \min\{d_{ix} + d_{xz} + d_{zy} + d_{yj} \mid x \in \overrightarrow{\mathrm{ADS}}(i), \quad (5)\\ y \in \overleftarrow{\mathrm{ADS}}(j), z \in \overrightarrow{\mathrm{ADS}}(x) \cap \overleftarrow{\mathrm{ADS}}(y)\}\end{aligned}$$

uses neighboring ADSs to obtain a tighter upper bound, but the associated query time is $O(k^2 \ln^2 n)$.

Moreover, we could use ADSs to obtain *lower bounds* on the distance between any two nodes as well. More precisely, the following lower bound on d_{vu} can be obtained from $r(u)$ and $\mathrm{ADS}(v)$:

$$\Delta(u,v) = \begin{cases} d_{vu}, \text{ if } u \in \mathrm{ADS}(v) \\ \tau_v^{-1}(r(u)), \text{ if } u \notin \mathrm{ADS}(v). \end{cases}$$

For directed graphs we use $\overrightarrow{\Delta}$ and $\overleftarrow{\Delta}$, according to the direction of the ADS of v. A tighter bound can be obtained using both $\mathrm{ADS}(u)$ and $\mathrm{ADS}(v)$:

$$\underline{d}_{ij} \leftarrow \max \begin{cases} \overrightarrow{\Delta}(j,i), \\ \overleftarrow{\Delta}(i,j), \\ \forall v \in \overrightarrow{\mathrm{ADS}}(j),\ \overrightarrow{\Delta}(v,i) - d_{jv} \\ \forall v \in \overleftarrow{\mathrm{ADS}}(i),\ \overleftarrow{\Delta}(v,j) - d_{vi}. \end{cases} \tag{6}$$

The last two conditions state that, if there is a node v in $\mathrm{ADS}(j)$ and distance d from v which we know has distance at least $d + x$ from i, then we get a lower bound of x on the distance between i and j.

4. CLOSENESS SIMILARITY

Using only the distance between two nodes to measure their similarity has obvious drawbacks. In this section, we introduce the concept of *closeness similarity*, which measures the similarity of two nodes based on their views of the full graph. More precisely, we consider the distance from each of these two nodes to all other nodes in the network and measure how much these two distance vectors differ. This is computationally expensive, but ADSs allow an efficient estimation of this view, as Section 4.1 will show.

Closeness similarity is specified with respect to a *distance function* δ_{ij} between nodes, a *distance decay* function $\alpha(d)$ (a monotone non-increasing function of distances), and a *weight function* $\beta(i)$ of node IDs. The basic expression for closeness similarity is

$$S_{\alpha,\beta}(v,u) = \sum_i \alpha(\max\{\delta_{vi}, \delta_{ui}\})\beta(i), \tag{7}$$

but we actually prefer the following Jaccard form, where the similarity is always in $[0,1]$:

$$J_{\alpha,\beta}(v,u) = \frac{\sum_i \alpha(\max\{\delta_{vi}, \delta_{ui}\})\beta(i)}{\sum_i \alpha(\min\{\delta_{vi}, \delta_{ui}\})\beta(i)}. \tag{8}$$

LEMMA 4.1. *If $\alpha(d) \geq 0$ is monotone non-increasing and $\beta(x) \geq 0$ is nonnegative, then $J_{\alpha,\beta}(u,v) \in [0,1]$ for all pairs u, v, and $J_{\alpha,\beta}(v,v) = 1$ for any v.*

PROOF. The first claim follows from monotonicity of α, which implies that $\alpha(\min\{\delta_{vi}, \delta_{ui}\}) \geq \alpha(\max\{\delta_{vi}, \delta_{ui}\})$. The second from $\alpha(\min\{\delta_{vi}, \delta_{vi}\}) = \alpha(\max\{\delta_{vi}, \delta_{vi}\}) = \alpha(\delta_{vi})$. □

The notion of closeness similarity is quite general and extremely powerful, as this section will show. A potential drawback is that it is expensive to compute exactly, but Section 4.1 will show that we can use ADSs to approximate it accurately and efficiently for all choices of α and β and two natural choices of the distance function δ. In terms of effectiveness, our experiments will show that simply setting $\alpha(x) \equiv 1/x$, $\beta \equiv 1$, and $\delta_{vi} \equiv \pi_{vi}$ (Dijkstra ranks) is enough to obtain very good results.

In general, however, the weighting β can be anything that depends on readily-available parameters of a node, such as its degree or metadata (e.g., gender or interests obtained through text analysis). Similarity is then measured with respect to these weights. Using $\beta \equiv 1$ weighs all nodes equally, in which case we omit β from the notation. A weight function that increases with degree can be used to emphasize contribution of highly connected nodes to the similarity. Alternatively, a decreasing β may make sense in collaborative filtering applications, where the proximity of two nodes to a third one is more meaningful when the third node is less central.

In the remainder of this section, we consider closeness similarity with respect to both *SP distance* (where $\delta_{ij} \equiv d_{ij}$), in Section 4.1.1, and *Dijkstra rank* (position in the nearest-neighbor list, with $\delta_{ij} = \pi_{ij}$), in Section 4.1.2. We explain the semantic difference between these choices in Section 4.2.

The notion of closeness similarity generalizes several existing measures. In particular, when δ is the SP distance, the Adamic-Adar (AA) measure [4] can be expressed as $S_{\alpha,\beta}$ using $\alpha(x) = 0$ when $x \geq 2$ and $\alpha(x) = 1$ otherwise, and $\beta(u) = 1/\log|\Gamma(u)|$, where $|\Gamma(u)|$ is the degree of u. Using $\alpha(x) = 1/(1+x)^c$ (for some choice of c) gives us a global variant of AA. Similarly, the intersection size of d-neighborhoods, which naturally extends the local *common neighbors* measure, can be expressed using $\beta(u) \equiv 1$ and $\alpha(x) = 1$ if $x \leq d$ and 0 otherwise.

Closeness similarity is also tightly related to the classic notion of *closeness centrality*, which measures the importance or relevance of nodes in a social network based on their distances to all other nodes. The closeness centrality $C_{\alpha,\beta}(i)$ of a node i is simply the closeness similarity $S_{\alpha,\beta}(i,i)$ between the node and itself. Use of $\alpha(x) = 1/x$ specifies Harmonic mean centrality (used in [32] with resistance distances and evaluated in [8] with SP distances) and $\alpha(x) = 2^{-x}$ specifies exponential decay with distance [20]. The number of reachable nodes from v can be expressed using $\alpha(x) = 1$. Estimators for $C_\alpha(v)$ from $\mathrm{ADS}(v)$ were studied in [13, 15] and have coefficient of variation (CV) $\leq 1/\sqrt{2(k-1)}$, which means that the relative error rapidly decreases with k.

In directed networks such as Twitter, we can consider closeness similarity with respect to either forward and backward edges. The forward similarity compares nodes based on how they relate to "the world," whereas backward similarity compares them based on how the world relates to them. To simplify the technical presentation, we focus on undirected graphs. Results easily extend to directed graphs by appropriately considering forward or backward paths.

4.1 Estimating Closeness Similarity

The exact computation of (7) and (8) is expensive, since it amounts to performing two single-source shortest-path searches for each pair of nodes. We will show, however, that we can obtain good estimates from $\mathrm{ADS}(u)$ and $\mathrm{ADS}(v)$. Since the expected time to compute the whole set of ADS labels is within a $k \ln n$ factor of a single-source shortest-path computation, this is very appealing. We also provide tight theoretical bounds on the quality of these estimates.

Our estimates $\hat{S}_{\alpha,\beta}(u,v)$ and $\hat{J}_{\alpha,\beta}(u,v)$ are obtained by separately estimating $\alpha(\max\{\delta_{vi}, \delta_{ui}\})$ and (when using the Jaccard form) $\alpha(\min\{\delta_{vi}, \delta_{ui}\})$ for each node i. We then substitute the estimate values $\hat{\alpha}(\max\{\delta_{vi}, \delta_{ui}\})$ and $\hat{\alpha}(\min\{\delta_{vi}, \delta_{ui}\})$ in the respective expressions (7) and (8). Since the estimate values are positive only for nodes $i \in \mathrm{ADS}(u) \cup \mathrm{ADS}(v)$, the computation is very efficient. In order to obtain a small relative error with arbitrary β, the random ranks underlying the ADS computation have to be drawn with respect to β (as done for centrality computation in [13]). This is explained in Section 4.3.

The estimates we obtain for each summand, and therefore each sum, are nonnegative. They are also unbiased for SP distances, and nearly so for Dijkstra ranks. The estimate on each summand

(with respect to each node i) typically has high variance: since each ADS has expected size $k \ln n$, for most nodes i we have limited information from $\mathrm{ADS}(u)$ and $\mathrm{ADS}(v)$. We can, however, bound the error of the aggregate estimate. To do so, we consider each variant (based on SP distances and Dijkstra ranks) separately.

4.1.1 SP Distance Closeness

We use the L* estimator of Cohen and Kaplan [17, 18], which is the unique estimator that is unbiased, nonnegative, and monotone (estimate is non-increasing with $r(i)$). The L* estimator is also variance-optimal in a Pareto sense [17]. We explicitly derive the estimator for functions α that are monotone non-increasing with the maximum distance $\alpha(\max\{d_{vi}, d_{ui}\})$ and with the minimum distance $\alpha(\min\{d_{vi}, d_{ui}\})$. We state the estimator, as applied to $\mathrm{ADS}(u)$ and $\mathrm{ADS}(v)$, and our error bounds.

Our estimator for $\alpha(\max\{d_{vi}, d_{ui}\})$ is defined by the following lemma.

LEMMA 4.2. *The L* estimate of* $\alpha(\max\{d_{vi}, d_{ui}\})$ *is*

$$\hat{\alpha}(\max\{d_{vi}, d_{ui}\}) = \begin{cases} i \notin \mathrm{ADS}(u) \cap \mathrm{ADS}(v)\text{:} & 0 \\ i \in \mathrm{ADS}(u) \cap \mathrm{ADS}(v)\text{:} & \frac{\alpha(\max\{d_{vi}, d_{ui}\})}{p_{i,u,v}} \end{cases},$$

where $p_{i,u,v} = \min\{p_{vi}, p_{ui}\}$*, with* p_{vi} *and* p_{ui} *as defined in* (1)*.*

PROOF. When we have no information (no upper bound on $\max\{d_{vi}, d_{ui}\}$), the estimate is 0. The value $p_{i,u,v}$ is the conditional probability (conditioned on fixed ranks of all other nodes) of the event $i \in \mathrm{ADS}(u) \cap \mathrm{ADS}(v)$. Therefore the estimator is simply the inverse-probability estimate conditioned on fixed ranks. □

Note that we have all the information needed to compute the estimate. When $i \notin \mathrm{ADS}(u) \cap \mathrm{ADS}(v)$, the estimate is 0. When $i \in \mathrm{ADS}(u) \cap \mathrm{ADS}(v)$, we know both d_{ui} and d_{vi} and can compute p_{vi} and p_{ui}. Lastly, to estimate the sum $\sum_i \alpha(\max\{d_{vi}, d_{ui}\})\beta(i)$ it suffices to only consider nodes that belong to the intersection of the ADSs, because they are the only ones with a positive estimate.

Our estimate for $\alpha(\min\{d_{vi}, d_{ui}\})$ is defined as follows.

LEMMA 4.3. *The L* estimate of* $\alpha(\min\{d_{vi}, d_{ui}\})$ *is*

$$\hat{\alpha}(\min\{d_{vi}, d_{ui}\}) = \begin{cases} i \notin \mathrm{ADS}(u) \cup \mathrm{ADS}(v): & 0 \\ i \in \mathrm{ADS}(u) \setminus \mathrm{ADS}(v): & \frac{\alpha(d_{ui})}{p_{ui}} \\ i \in \mathrm{ADS}(v) \setminus \mathrm{ADS}(u): & \frac{\alpha(d_{vi})}{p_{vi}} \\ i \in \mathrm{ADS}(u) \cap \mathrm{ADS}(v): & \\ \quad \textit{if } p_{vi} \le p_{ui}: & \\ \frac{\alpha(\min\{d_{vi}, d_{ui}\}) - (p_{ui} - p_{vi})\frac{\alpha(d_{ui})}{p_{ui}}}{p_{vi}} & \\ \quad \textit{if } p_{ui} < p_{vi}: & \\ \frac{\alpha(\min\{d_{vi}, d_{ui}\}) - (p_{vi} - p_{ui})\frac{\alpha(d_{vi})}{p_{vi}}}{p_{ui}} & \end{cases}.$$

PROOF. We apply the general derivation of the L* estimator, presented in [17]. In our context, we consider the outcome as a function of $r(i)$, conditioned on the ranks on all other nodes being fixed. The outcome is the occurrence of the events $i \in \mathrm{ADS}(u)$ and $i \in \mathrm{ADS}(v)$, as a function of $r(i)$. The outcome depends on the relation of $r(i)$ to the two threshold values $\tau_v(d_{vi})$ and $\tau_u(d_{ui})$ as in (2), which are solely determined by the ranks of $\Phi_{<i}(v)$ and $\Phi_{<i}(u)$. In particular, they do not depend on $r(i)$.

To obtain the L* estimate, we need to consider the (tightest) lower bound we can obtain on $\alpha(\min\{d_{vi}, d_{ui}\})$ from the outcome. This is the infimum of the function on all possible data (distances d_{ui} and d_{vi}) that are consistent with the outcome. Note that from the outcome for a given $r(i) = y$, we have enough information to determine the outcome, and the respective lower bound function, for all $r(i) \ge y$.

We now provide the estimator for all possible outcomes.

The first case is $i \notin \mathrm{ADS}(u) \cup \mathrm{ADS}(v)$, which happens when $r(i) > \max\{\tau_v(d_{vi}), \tau_u(d_{ui})\}$. This is consistent with i not even being reachable from v or i (or being very far). We therefore do not have an upper bound on the minimum distance (and thus do not have a lower bound on $\alpha(\min\{d_{vi}, d_{ui}\})$). We use the estimate $\hat{\alpha}(\min\{d_{vi}, d_{ui}\}) = 0$.

The second case is $i \in \mathrm{ADS}(u) \setminus \mathrm{ADS}(v)$. This can only happen if $\tau_u(d_{ui}) \ge \tau_v(d_{vi})$ and when $r(i) \in (\tau_v(d_{vi}), \tau_u(d_{ui})]$. In this case we know d_{ui}, and we have that $p_{ui} = \tau_u(d_{ui})$. Note that d_{ui} is trivially an upper bound on the minimum distance. It is also the tightest bound we can get from the information we have. We can only obtain a lower bound of $\tau_v^{-1}(r(i))$ (see Equation (3)) on d_{vi}, but we can not upper bound it. If $\tau_v^{-1}(r(i)) \ge d_{ui}$ then we know that $d_{vi} \ge d_{ui}$ and thus we know that the minimum distance is d_{ui}. Otherwise, we only know that the minimum distance is between $\tau_v^{-1}(r(i))$ and d_{ui} and the best upper bound we have is d_{ui}. Either way, whether we know the minimum distance or not, the estimate is $\hat{\alpha}(\min\{d_{vi}, d_{ui}\}) = \frac{\alpha(d_{vu})}{\tau_v(d_{ui})}$. The case $i \in \mathrm{ADS}(v) \setminus \mathrm{ADS}(u)$ is symmetric.

The remaining case is when $i \in \mathrm{ADS}(v) \cap \mathrm{ADS}(u)$. In this case we can determine d_{vi}, d_{ui}, and both thresholds $\tau_v(d_{vi}) = p_{vi}$ and $\tau_u(d_{ui}) = p_{ui}$. Assuming $\tau_v(d_{vi}) \le \tau_u(d_{ui})$ (the other case is symmetric), the L* estimate is the solution for x of the unbiasedness constraint: $\alpha(\min\{d_{vi}, d_{ui}\}) = (\tau_u(d_{ui}) - \tau_v(d_{vi}))\frac{\alpha(d_{ui})}{\tau_u(d_{ui})} + \tau_v(d_{vi})x$. □

Note again that we have all the information we need to compute the estimate. We know d_{vi} and p_{vi} if and only if $r(i) \le p_{vi}$, which happens if and only if $i \in \mathrm{ADS}(v)$, and symmetrically for u.

With these estimators, the Jaccard closeness similarity can be estimated in the natural way:

$$\hat{J}_{\beta,\alpha}(v,u) = \frac{\sum_{i \in \mathrm{ADS}(u) \cap \mathrm{ADS}(v)} \hat{\alpha}(\max\{d_{vi}, d_{ui}\})\beta(i)}{\sum_{i \in \mathrm{ADS}(u) \cup \mathrm{ADS}(v)} \hat{\alpha}(\min\{d_{vi}, d_{ui}\})\beta(i)}. \quad (9)$$

The estimates for both the numerator and denominator are unbiased, since our estimates for each summand are unbiased. The estimate on the ratio is biased, but we can bound the root of the mean square error of the estimate:

THEOREM 4.1. *When* $\alpha(x)$ *is non-increasing and node ranks are drawn according to* $\beta(i)$*, the root of the expected square error of* $\hat{J}_{\beta,\alpha}(v,u)$ *is* $O(1/\sqrt{k})$*.*

PROOF. The coefficient of variation of the estimate on $\sum_i \alpha(\min\{d_{vi}, d_{ui}\})$ is bounded by $1/\sqrt{2(k-1)}$. The RMSE (square root of expected square error) on the estimate $\sum_i \alpha(\max\{d_{vi}, d_{ui}\})$ is bounded by $\sum_i \alpha(\min\{d_{vi}, d_{ui}\})/\sqrt{2(k-1)}$. □

4.1.2 Dijkstra Rank Closeness

A natural choice for the distance decay function is $\alpha = 1/x$, based on the harmonic mean. With Dijkstra ranks, we suggest the use of the variant $\alpha(x) = \min\{1, h/x\}$ for some integer $h > 1$. With $\alpha(x) = 1/x$, the weights assigned to the h closest nodes are in the range $[1/h, 1]$, which can make the measure less robust when these nodes are actually similar. By using $h > 1$, we give the h closest nodes the same α weight, increasing robustness to variations in their ordering.

We now consider the choice $h = k$. Making the appropriate substitutions in Equation (8) we have

$$J^*(u,v) = \frac{\sum_i \min\{1, \frac{k}{\max\{\pi_{vi},\pi_{ui}\}}\}}{\sum_i \min\{1, \frac{k}{\min\{\pi_{vi},\pi_{ui}\}}\}}. \quad (10)$$

It turns out that we can obtain a particularly simple estimator for J^*:

$$\hat{J}^*(u,v) = \frac{|\text{ADS}(u) \cap \text{ADS}(v)|}{|\text{ADS}(u) \cup \text{ADS}(v)|}. \quad (11)$$

This is due to the following relations:

LEMMA 4.4. *For any three nodes i, u, and v,*

$$\Pr[i \in \textit{ADS}(u) \cap \textit{ADS}(v)] \approx \min\left\{1, \frac{k}{\max\{\pi_{vi},\pi_{ui}\}}\right\}$$

$$\Pr[i \in \textit{ADS}(u) \cup \textit{ADS}(v)] \approx \min\left\{1, \frac{k}{\min\{\pi_{vi},\pi_{ui}\}}\right\}$$

PROOF. Recall that the probability that a node i appears in $\text{ADS}(v)$ is $\min\{1, k/\pi_{vi}\}$. The probabilities that the node appears in the intersection of two or more ADSs are highly positively correlated. The joint probability is close to the minimum of the two, which is $\frac{k}{\max\{\pi_{vi},\pi_{ui}\}}$. More precisely, for $k = 1$, the joint probability is between $\frac{k}{\pi_{vi}+\pi_{ui}}$ (if the sets of preceding nodes in the permutations are disjoint) and $\frac{k}{\max\{\pi_{vi},\pi_{ui}\}}$ (if the sets of preceding nodes are maximally overlapping). Asymptotically, when k is larger, the expected size of the intersection of ADSs more closely approximates (10). □

Note that a nice property of these approximations is that we can work with a very compact representation of ADSs. Since each ADS has expected size $k \ln n$, to estimate size of union and intersection we can hash node IDs to a smaller domain of size $O(\log k + \log\log n)$. We can then store the ADS as an unordered set, obtaining a significant reduction in storage needed for the labels for very large graphs.

These approximations quickly converge with k. Using linearity of expectation, it follows that the sum with respect to $\alpha(\max\{\pi_{vi},\pi_{ui}\})$ can be approximated by the size of the intersection $|\text{ADS}(u) \cap \text{ADS}(v)|$, and the sum with respect to $\alpha(\min\{\pi_{vi},\pi_{ui}\})$ is approximated by the size of the union $|\text{ADS}(u) \cup \text{ADS}(v)|$.

We now consider estimating closeness similarity for more general forms of α. One complication is that, in contrast to the distance d_{ui}, the Dijkstra rank π_{ui} of a node i is not readily available when $i \in \text{ADS}(u)$. Fortunately, we can still obtain an estimate with good concentration:

LEMMA 4.5. *Conditioned on $i \in \text{ADS}(v)$, the estimate $\hat{\pi}_{vi} = 1 + |\widehat{\Phi_{<i}(v)}|$, where*

$$|\widehat{\Phi_{<i}(v)}| = \sum_{j \in \text{ADS}(v) \cap \Phi_{<i}(v)} \frac{1}{p_{vj}}, \quad (12)$$

is unbiased and has CV at most $1/\sqrt{2(k-1)}$.

PROOF. The Dijkstra rank $\pi_{vi} = 1 + |\Phi_{<i}(v)|$ is equal to 1 plus the number of nodes that are closer to v than i (we assume unique distances in this definition). The best estimator we are aware of for $|\Phi_{<i}(v)|$ is the RC estimator [13], which is (12). This RC estimate is equal to the exact size when $|\Phi_{<i}(v)| \leq k$ (i.e., $\pi_{vi} \leq k+1$) and an estimate that always exceeds k otherwise. The CV of the estimate is upper-bounded by $1/\sqrt{2(k-1)}$. Therefore for large k we can get good concentration. Note that we are able to compute this estimate only when $i \in \text{ADS}(v)$ (because otherwise we can not determine the nodes in $\text{ADS}(v)$ that are closer to v than i is). Note however that the estimate is independent of the inclusion of i in $\text{ADS}(v)$. Therefore, we get good concentration and unbiasedness when the estimate is conditioned on $i \in \text{ADS}(v)$. □

We can then substitute the estimated Dijkstra ranks $\hat{\pi}_{vi}$ instead of distances in the SP distance closeness similarity estimators.

4.2 SP Distance versus Dijkstra Rank

We now explain the qualitative difference between closeness similarities based on distances and Dijkstra ranks, which prompted us to consider both. The measures behave differently in heterogeneous networks where nodes have different distance distributions. For intuition, we relate the problem to the classic information retrieval problem of similarity of documents. For duplicate elimination, we may want to deem documents similar based on both topic and size. When searching for query matches, we may want to factor out the size and compare only based on topic. In our social network context, the Dijkstra rank closeness captures "topic similarity": the similarity of two nodes in terms of the *relative* order of their relations to other nodes. SP distance closeness uses the absolute strength of the relations.

To make this concrete, consider measuring the similarity of two nodes u and v. Suppose that the 100 nearest neighbors of u are much closer to u than the 100 nearest neighbors of v are to v. If these sets are sufficiently similar, we may want to say that u and v are similar in that they both view the world (or, if there is direction, the world relates to them) in a similar way. In this case, we should use Dijkstra rank closeness. To identify nodes that are similar in both topic and level of involvement, we should use distance-based closeness.

In our experiments on social networks, we use TF-IDF cosine similarity as a proxy for ground truth (see Section 6.1.2). With this measure, the similarity between nodes is only based on topics (and not the number) of publications or tweets associated with them. As explained, this makes Dijkstra rank closeness (and Equation (10)) a better fit.

4.3 Using Node Weights

In order to retain the CV bound of our closeness similarity estimators of Theorem 4.1 when β is not uniform, we need to make slight modification to the ADS computation. We briefly state these modifications. The rank $r(u)$ we assign to each node u is exponentially distributed with parameter $\beta(u)$. This can be done by drawing $y \sim U[0,1]$ (independently for each node) and computing $r(u) = -\ln(y)/\beta(u)$. The intuitive explanation for this choice is that the expected rank of a node is $1/\beta(u)$. Therefore, nodes with higher $\beta(u)$, which also have a higher contribution to closeness similarity, are more likely to obtain a lower rank value $r(u)$ and therefore are more likely to be represented in the ADSs of other nodes.

The estimators we apply need to be modified accordingly, since we are using a different distribution. The only difference is the computation of the probabilities p_{ui} and p_{vi}. Instead of simply using (1), we compute the probability that an exponential random variable with parameter $\beta(u)$ is smaller than $\text{k}^{\text{th}}_r(\Phi_{<u}(v))$. We obtain

$$p_{vu} = 1 - e^{-\beta(u)\text{k}^{\text{th}}_r(\Phi_{<u}(v))}.$$

Note that, because exponential random variables are not bounded, we need to define $\text{k}^{\text{th}}_r(X) = +\infty$ when $|X| < k$. The intuition behind the analysis is that (under an appropriate scaling) a node

of weight $\beta(i)$ is treated like $\beta(i)$ nodes of weight 1 under uniform scaling. Note that the mapping is not exact, as all $\beta(i)$ copies are correlated, but then the correlations between the copies work in our favor: if *any* one of the copies was represented in the decomposed problem then all of them would be represented in the weighted instance we are actually treating. For more details, we refer the reader to [12, 13, 15]. (Our problem here is more general, but the technique of handling node weights is the same)

5. RANDOMIZED EDGE LENGTHS

In this section, we introduce a similarity measure that is based on the SP distance and can be computed by SP computations, but is sensitive to path multiplicities as well as length. Our intuitive expectation from a similarity measure is that the similarity between u and v increases when there are more paths between u and v and when the paths are shorter. On the simple patterns in Figure 1, we expect similarity to increase with r (the number of paths) and decrease with ℓ (the SP distance). All common similarity measures satisfy this expectation in a *weak* sense: similarity is non-increasing with path lengths and is non-decreasing with multiplicity. Popular global measures such as KATZ, RWR, or resistance distance satisfy this expectation in a *strong* sense: similarity strictly decreases with ℓ and strictly increases with r. The SP distance and the minimum cut, however, satisfy the expectation only in a weak sense: the SP distance does not depend on r but does decrease with ℓ, whereas the minimum cut does not depend on ℓ but (for patterns B and C) increases with r. These two measures thus fail to capture some essential property of the connectivity between the nodes.

Given a network with link multiplicities w_e, we define the *REL distance* to be the *expected* shortest path distance with respect to random edge lengths that are drawn independently from an exponential distribution with parameter w_e. This is the same as assigning to each edge e a random length $-(\ln u_e)/w_e$, where $u_e \sim U[0,1]$ are independent.

The use of the exponential distribution is natural here: if we have multiple parallel links with multiplicities $\{w_i\}$, their effect on the REL distance is the same as that of a single link with multiplicity $\sum_i w_i$, which is what we intuitively want to happen. This holds because the minimum of several exponentially distributed random variables distributes like a single exponential random variable with parameter that is the sum of the original parameters.

The REL distance can be estimated using standard SP computations: we simply draw a set of random lengths and compute SP distances with respect to these lengths. We repeat this m times (for some m) for accuracy and take the average distance.

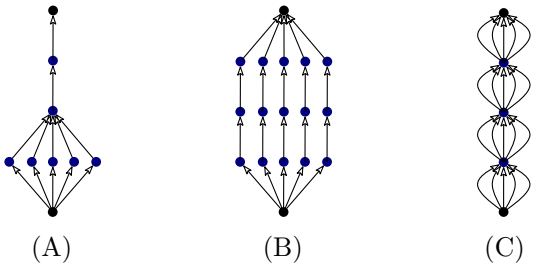


Figure 1: Connection between two nodes in simple networks. Pattern (B) consists of r disjoint paths of length ℓ. Pattern (C) has a path of length ℓ with r parallel edges between consecutive nodes. Pattern (A) has r disjoint paths of length 2 to an intermediate node and a path of length $\ell-2$ from the intermediate node to the other end point. All paths in these examples are of integral length $\ell > 2$.

LEMMA 5.1. *The REL distance estimator is unbiased and has CV* $\leq 1/\sqrt{m}$.

PROOF. Unbiasedness is immediate. The CV does not exceed that of a single exponential random variable. □

Note that REL distance is computationally not much more expensive than SP distance, but has several advantages, which we discuss in the remainder of this section.

5.1 Edge Robustness

Intuitively, a similarity measure is more robust when it is less sensitive to removal of edges [11, 21], or additions of spurious (random) edges to the network, which connect nodes that are otherwise very dissimilar. In this sense, the REL distance is much more robust than the SP distance, and behaves more similarly to KATZ and RWR.

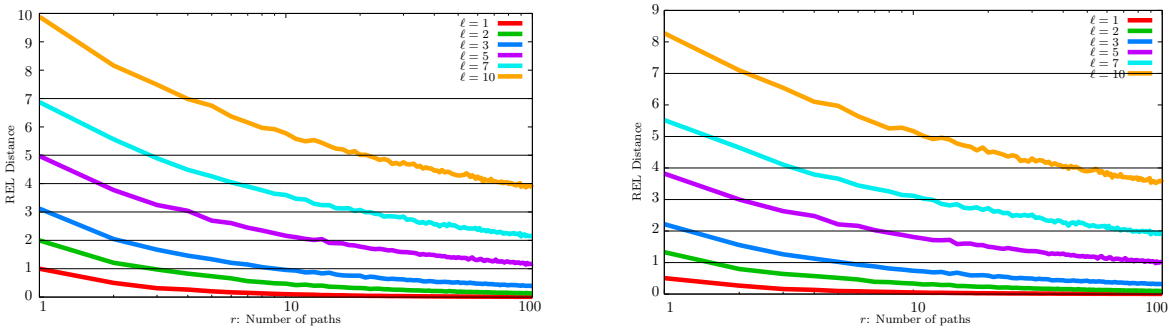
Consider again the patterns in Figure 1 and the relative similarity scores on these patterns by different measures. (C) is the most robust to random edge removals and (A) is the least robust. Therefore, we would like our similarity measure to have (C) $\succ$ (B) $\succ$ (A). The SP measure does not distinguish between the three cases, all having the same SP distance of ℓ, so we get (A) $\approx$ (B) $\approx$ (C). The same holds for RWR (PageRank, hitting time) regardless of the restart probability. The KATZ measure would give (C) $\succ$ (A) $\approx$ (B), since case (C) has r^ℓ paths and there are only r paths in cases (A) and (B). Thus, KATZ fails to distinguish between (A) and (B). The resistance distance is ℓ/r in both (B) and (C), thus not distinguishing between the two, but correctly giving higher resistance distance $\ell - 2 + 1/r$ in case (A). The same relation holds for the minimum cut, where the cut value is r for (B) and (C) but only 1 for (A). Thus, with both resistance distance and minimum cut, we obtain (C) $\approx$ (B) $\succ$ (A). Lastly, the REL distance gives the relation we want: (C) $\succ$ (B) $\succ$ (A). To see why, note that (C) has REL distance ℓ/r; this is much lower than (B), which has REL distance $\approx \ell - r\sqrt{\ell}$ (the standard deviation for a single length-ℓ path is $\sqrt{\ell}$ and we use an approximation for $r \ll \sqrt{\ell}$); this in turn is lower than (A), which has REL distance between $\ell - 2 + 2/r$ and ℓ.

5.2 Capturing Hub Nodes

Intuitively, when modeling or representing the relations of a node u it is important to identify when the similarity of a pair (u, w) is highly dependent on the presence of a single node v. That is, if v is removed, the similarity of (u, w) would significantly decrease. In such cases, we say that v is a *local hub* for the pair (u, w). For the relations between end points in the patterns in Figure 1, there are no local hubs in (B), all $\ell - 1$ middle nodes are local hubs in (C), and there are $\ell - 2$ local hubs in (A). We say that a similarity measure captures local hubs when we can identify that node v is a local hub for (u, w) from the similarities of (u, v), (v, w) and (u, w). We can identify local hubs with KATZ and RWR using the product relation $s_{uw} \approx s_{uv}s_{vw}$, with minimum cut using the relation $s_{uw} \approx \min\{s_{uv}, s_{vw}\}$, and with REL and resistance distance using the sum relation $s_{uw} \approx s_{uv} + s_{vw}$. SP distance, in contrast, does not capture local hubs: it is possible that $d_{uw} = d_{uv} + d_{vw}$ (as in pattern (B)) even when v is not a local hub for (u, w).

5.3 REL with ADSs

We have seen that REL has several desirable properties. As defined, however, it is very expensive to compute: it requires the computation of multiple shortest paths between two nodes, with differ-



(a) μ (b) $\mu - \sigma/2$

Figure 2: Variants of REL distance for two nodes connected by r disjoint paths of length ℓ: (a) standard REL; (b) REL distance minus half the standard deviation.

ent cost functions. Once again, we can use ADSs to approximate the REL computation efficiently and accurately.

We repeat the ADS computation t times, all with respect to the same random node ranking, but using different draws of edge lengths. Using the same node ranking makes the union of the ADSs more similar: a node that is included in one iteration is more likely to be included in another. That way, we can produce a single compact representation of all t sets of ADSs where each node appears with some aggregate information (average, quantile, sample SD of its distance or estimated Dijkstra rank in the different iterations). To estimate closeness similarity, we can use the t sets or work with the aggregate representation (which has a slightly different interpretation). We note that when using REL with closeness centrality estimation, we could randomize over node ranks as well. In an one-time centrality computation, the ADSs are computed but do not need to be stored [13], so there is no advantage for overlap.

A side advantage of REL when using Dijkstra ranks is that insignificant differences between nodes that can have very different Dijkstra ranks under fixed distances are ironed out. When differences are not significant, the REL distributions have significant overlap, which means that nodes would alternate relative Dijkstra rank in different repetitions.

5.4 Parametrized REL

With REL, as with other distance measures, many different patterns yield the same similarity score. We may want to be able to tune that by a parameter which controls the relative significance of different path lengths. Such a parameter is present in KATZ (the base of the exponent) and RWR (the restart probability) measures.

We propose a way to gain such control with REL. Instead of only considering the expectation of the distance, we can consider its distribution (more precisely, the variance of the REL distance random variable). Longer paths have smaller variance for the same expectation, and the sample variance and sample standard deviation σ are as easy to estimate as the sample mean μ. We propose using $\mu - c\sigma$ (for some constant $c \geq 0$) in order to increase the cost of longer paths and $\mu + c\sigma$ to achieve the opposite. Alternatively, we can also use a quantile of the distribution.

To understand this parametrization, we consider the REL distance $a(r, \ell)$ from u ro v when they are connected by r disjoint paths of length ℓ. The REL distance increases with ℓ and decreases with r (as discussed earlier, all the similarity measures we mention are non-increasing with ℓ and non-decreasing with r). Figure 2(a) shows the standard REL distance as a function of r for different values of the path length ℓ. The figure shows the relation between paths of different lengths. For example: a single length-7 path has the same REL distance as 4 disjoint length-10 paths. A single length-1 path (single edge) has the same REL distance as 3 disjoint length-2 paths or 8 disjoint length-3 paths.

Figure 2(b) shows the relation when using a modified REL distance of $\mu - \sigma/2$, which makes longer paths have higher cost. We can see that now a single path of length 7 has the similarity of 7 length-10 paths. Similarly, it takes 6 length-2 paths or 30 length-3 paths to match a single length-1 path.

6. EXPERIMENTS

We now present experimental results that illustrate the usefulness of the concepts introduced in this paper. Our focus is on testing the most natural variant of each major concept we dealt with: distance estimation (using ADSs), closeness similarity, and randomized edge lengths. Our general approach is to compare the graph-based measures we study with independent similarity measures based on metadata associated with each node.

6.1 Setup and Methodology

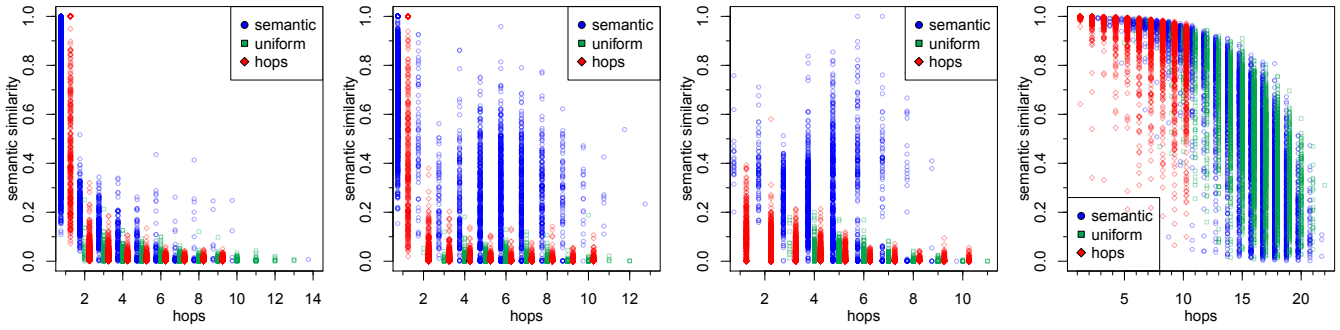
Our code is written in C++ and compiled with Microsoft Visual C++ 2012. Our test machine runs Windows Server 2008 R2 and has 384 GiB of DDR3-1066 RAM and two 8-core Intel Xeon E-5-2690 2.90 GHz CPUs, each with 8×64 KB of L1, 8×256 KB of L2, and 20 MB of shared L3 cache. All executions are single-threaded and run in memory.

6.1.1 Data Sets

We consider the largest (strongly) connected component of three real-world social networks (arXiv, DBLP, and Twitter) as well as a synthetic one (small world). The first two columns of Table 1 give key statistics for these networks. (We will discuss the remaining columns in Section 6.1.4.)

Table 1: Key statistics of our data sets, together with label generation effort (time and nodes per label) and in-memory query times for label intersection ($k = 3$, $m = 1$)

graph	nodes [$\times 10^6$]	edges [$\times 10^6$]	prep. [h:m]	label size	query [μs]
arXiv	0.43	28.68	0:02	37.85	1.22
DBLP	1.06	9.16	0:02	39.09	1.64
twitter	29.64	603.87	8:05	50.82	3.51
smallworld	1.00	5.98	0:02	40.74	1.35



(a) arXiv (b) DBLP (c) Twitter (d) Small World

Figure 3: Plots of all pairs sampled according to each distribution (uniform, hop-based, and semantic).

The first two networks come from DBLP and arXiv data.[1] In both cases, the raw data contains a list of articles with titles and authors; in addition, arXiv has abstracts and DBLP has venues. In the corresponding network, nodes represent authors and edges indicate co-authorship. We set the length of an edge to be the inverse of the number of common papers by the corresponding authors.

The third social network is the Twitter *mention graph*, built from tweets of December 2011. Each node represents a distinct user, and there is an edge between v and u if user v sends a tweet directed at user u (using the @ sign). This network is directed and weighted by 1/(number of mentions).

The final network we consider is a synthetic realization of a small world (SW) network [26], which is undirected and unweighted. It consists of an $N \times N$ toroidal grid with additional "long-distance" edges. For each node, we add an edge to a random node at L_1 distance d in the grid, where the probability of picking a particular value of d is proportional to d^{-2}.

6.1.2 Semantic-based Similarity

To evaluate the graph-based similarity measures we tested, we compare them against *semantic-based* similarity measures. These are based on individual properties of the entities represented by each node, with no direct information about the graph itself.

For the small world instance, we take the semantic similarity between nodes v and u to be the inverse of the L_1 distance between the corresponding points in the original metric space.

For real-world social networks, we measure semantic similarity using the metadata associated with each node. We map each node to a *document*, seen as a bag (multiset) of all terms (words) uttered by the user/author. For arXiv, this consists of the titles and abstracts of all articles written by the author; for DBLP, we use titles and venues; for Twitter, all tweets sent by the user, as long as at least two-thirds of the characters in the tweet are ASCII printable (this filters out most messages in non-Western languages, for which word-based similarity may not be an adequate ground truth). We measure the similarity between two nodes by comparing the corresponding documents. We first reweight terms using the standard TF-IDF (term frequency-inverse document frequency) method [23] (to deemphasize stop-words and other common terms), then compute the cosine similarity between the corresponding vectors.

More precisely, let $f(t, d)$ be the frequency of t, i.e., the number of times a term t appears in document d. Let the *logarithmically scaled term frequency* be $\text{tf}(t, d) = \ln(f(t, d) + 1)$. The *inverse document frequency* is defined as $\text{idf}(t, D) = \ln(|D|/|\{d \in D : t \in d\}|)$, where D is the entire corpus (set of documents). For a fixed document d and term t, $\text{tfidf}(t, d, D)$ is defined as $\text{tf}(t, d) \cdot \text{idf}(t, D)$. Let T be the set of all terms that appear in the entire corpus. Conceptually, each document d can be seen as a $|T|$-dimensional vector $v(d)$ whose t-th entry represents $\text{tfidf}(t, d, D)$. The *normalized semantic similarity* between two documents d_a and d_b is defined as the dot product of $v(d_a)$ and $v(d_b)$, divided (for normalization) by the product of the L_2 norms of $v(d_a)$ and $v(d_b)$. This is the similarity measure we use in our experiments.

6.1.3 Query Distribution

We evaluate the quality of our similarity measures on sample pairs of nodes. We pick such pairs using three distributions: *uniform*, *hop-based*, and *semantic-based*. Each has about 5000 pairs, and Figure 3 compares them. Each sampled pair (v, u) corresponds to a point; its x coordinate indicates the number of hops between v and u and its y coordinate indicates the semantic similarity. The color/shape of each point indicates the distribution it came from.

In the natural *uniform* distribution, each pair consists of two nodes picked independently and uniformly at random. Although it was often used in previous studies, this distribution is heavily biased towards pairs that are semantically quite different, since their nodes tend to be far apart in the graph.

In most real-life applications, however, we are interested in evaluating nodes that have some nontrivial degree of similarity. The *hop-based* distribution ensures that enough such pairs are evaluated, as follows. First, we pick a *center node* v uniformly at random. We then run a breadth-first search on the entire graph from v (disregarding any edge costs) and, for each $i \in \{1, 2, 3, \ldots, 10\}$, we pick a node u_i uniformly at random between all nodes that are exactly i hops away from v, creating a pair (v, u_i). We repeat this process with 500 different center nodes v.

The third distribution is *semantic-based*. Our semantic-based similarity measures are real numbers between 0 and 1. We split this range into twenty equal-width buckets $[0, 0.05), [0.05, 0.10), [0.10, 0.15), \ldots, [0.95, 1.00]$ and pick up to 250 pairs of nodes within each interval. We do so by picking pairs uniformly at random and assigning them to the appropriate bucket, keeping only the first 250 in each case. (For some inputs, the probability of hitting some of the higher buckets is extremely small; we may leave some buckets incomplete after sampling 20 million pairs unsuccessfully.)

As Figure 3 shows, most pairs in the uniform distribution are far apart and have extremely low semantic similarity. The hop-based distribution picks more pairs with higher semantic similarity, but

[1]Available at `http://export.arxiv.org/oai2` and `http://dblp.uni-trier.de/xml/`.

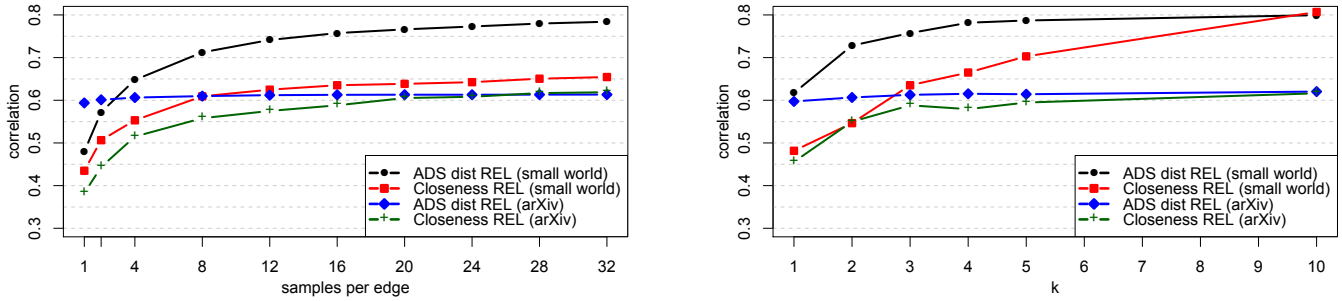


Figure 4: Parameter tests. On the left, we fix $k = 3$ and vary the number m of samples per edge. On the right, we fix $m = 16$ and vary k. For each set of parameters, we report Spearman's correlation relative to the semantic similarity.

they also tend to be very close together (in terms of hop distances). In general, the semantic-based distribution picks a more diverse set of pairs, with less correlation between hop distance and semantic similarity. We believe this distribution gives more insight into real applications.

Note that the two co-authorship networks (arXiv and DBLP) are quite different, due to different paper population and metadata. The small world network is quite distinct from other networks, with different structure, node degrees, and metadata.

6.1.4 Algorithms

To test the concepts introduced in this paper, we computed ADSs based on Dijkstra's algorithm, as described in [12, 13, 16]. Labels are represented in the obvious way, as pairs of hubs and distances, sorted by hub id to allow for a straightforward implementation of intersection in linear time [3]. Table 1 reports the time to generate labels (with $k = 3$ and $m = 1$, as defined in Sections 2 and 5) on our networks, the average number of nodes per ADS, and the average time to compute the similarity between two nodes from their ADSs. Note that preprocessing times are reasonable (a few hours for Twitter), and queries are extremely fast (a few microseconds). We want to stress that we did not tune our preprocessing code too much; our main focus is exploring the feasibility of ADS-based approaches for estimating similarity. We thus optimized our code enough to handle the Twitter data, but did not exploit further optimizations such as multi-threading or cache locality.

We implemented two ADS-based approaches to measure similarity. The first, *ADS distance*, simply computes the upper bound on the SP distance using Equation (4). Smaller distances indicate higher similarity. The second approach we consider is *closeness* similarity based on Dijkstra rank, using Equation (11).

We apply each variant to the original graph as well as to a version with randomized edge lengths (REL). For the latter, we use a single random rank function, and sample $m = 16$ lengths for each edge (see Section 6.2). The resulting preprocessing time is thus 16 times longer than those reported in Table 1.

To measure the quality of the results obtained, we compute Spearman's rank correlation coefficient [25] (or *Spearman's correlation*, for short) between each structure-based similarity measure and the appropriate semantic similarity, which we take as ground truth. Since Spearman's correlation is rank-based, it is a good fit to evaluate measures that operate on different scales. For consistency, when evaluating a similarity measure for which lower values are meant to indicate higher similarity (such as distance-based methods), we negate the similarity measure when computing Spearman's correlation. For all methods tested, therefore, positive values are better. Note that zero indicates a lack of correlation.

6.2 Parameter Setting

Both methods we study (ADS distance REL and Closeness REL) have two parameters: the threshold k for inclusion of nodes in ADS labels and the number m of samples per edge (for randomizing its length). We performed some preliminary explorations on the data to determine their values. Figure 4 shows how our similarity measures are affected (in terms of quality) when we fix one of these parameters and vary the other. As already mentioned, quality is measured in terms of Spearman's correlation to the semantic similarity (ground truth). We consider two representative graphs, small world and arXiv, and use the semantic-based distribution of pairs.

As predicted by our theoretical analysis, the trade-offs we must contend with are straightforward: increasing either parameter leads to better expected quality, at the cost extra time and space. (Recall that the preprocessing effort of the ADSs depends linearly on both k and m.) This monotonicity is very important in practice, since it eliminates the risk of overfitting to a specific input. In contrast, parameters such as the restart probability of RWR do not have this property.

Figure 4 shows that setting $k = 3$ and $m = 16$ is a reasonable trade-off between quality and efficiency. We therefore use these parameters in our experiments.

6.3 Results

Our main experimental results are reported in Tables 2, 3, and 4 (one table for each distribution of pairs described in Section 6.1.3). For each method and each graph, we give the Spearman's correlation between the similarity values they compute and the corresponding semantic similarities. The best result for each experiment (graph and distribution) is highlighted in bold.

Besides "ADS dist" and "Closeness" (and their respective versions augmented with REL), we also evaluate some competing measures. The *distance* measure is the actual graph distance between two nodes (computed explicitly with Dijkstra's algorithm). The *hops* measure is similar, but uses unit edge lengths. As a proxy for local methods, we use the *Adamic-Adar* measure [4], which adds up the number of common neighbors between v and u, weighted by the reverse logarithm of their degrees. As a proxy for global similarity measures, we consider *random walk with restarts* (RWR) [34], where the similarity between v and other nodes depends on the stationary distribution of a random walk from v, in the spirit of the rooted PageRank algorithm [10, 34]. The restart probability of this random walk must be tuned for different graphs, so we consider four different values (from 0.00 to 0.75). Our implementation of RWR is relatively slow, hence we only test it for one distribution on two networks. However, even the most efficient implementation of RWR [34] is orders of magnitude slower than ADS-based approaches.

Table 2: Uniform distribution: Spearman's correlation between each measure and semantic similarity.

measure	arXiv	DBLP	Twitter	SW
Adamic-Adar	0.097	0.034	0.107	0.000
hops	0.350	0.221	0.536	0.623
distance	**0.470**	**0.319**	0.191	0.623
ADS dist	0.462	0.318	0.196	0.519
ADS dist REL	0.419	0.314	0.242	**0.769**
Closeness	0.039	0.015	0.461	0.413
Closeness REL	0.063	0.034	**0.612**	0.666

Table 3: Hop-based distribution: Spearman's correlation between each measure and semantic similarity.

measure	arXiv	DBLP	Twitter	SW
Adamic-Adar	0.570	0.457	0.420	0.486
hops	0.645	0.468	**0.678**	0.831
distance	**0.648**	**0.507**	0.447	0.831
ADS dist	0.617	0.497	0.448	0.839
ADS dist REL	0.512	0.454	0.471	0.947
Closeness	0.379	0.249	0.518	0.877
Closeness REL	0.404	0.320	0.637	**0.949**

Table 4: Semantic distribution: Spearman's correlation between each measure and semantic similarity. (Entries marked "—" were not tested.)

measure	arXiv	DBLP	Twitter	SW
Adamic-Adar	0.626	0.746	0.548	0.000
hops	**0.752**	0.748	0.169	0.767
distance	0.590	0.634	−0.140	0.767
RWR-0.75	—	0.734	—	0.286
RWR-0.50	—	0.737	—	0.617
RWR-0.25	—	0.740	—	0.791
RWR-0.00	—	0.500	—	**0.915**
ADS dist	0.566	0.637	−0.127	0.671
ADS dist REL	0.614	0.584	−0.155	0.865
Closeness	0.641	0.742	0.613	0.609
Closeness REL	0.634	**0.752**	**0.649**	0.808

It is clear from the tables that no single measure dominates. This is to be expected, given that these are different networks that have evolved under different circumstances and with different semantics in mind. In DBLP, for example, an edge between nodes indicates an intentional collaboration that resulted in a paper. An edge in Twitter may result from a direct mention resulting from a re-tweet (which is common in some virtual chats), a casual reply, or a discussion/conversation.

There are, however, some obvious patterns we can discern. As expected, the effectiveness of local similarity measures is quite limited for arbitrary queries. Although Adamic-Adar works reasonably well for the hop-based distribution (Table 3), its performance varies wildly for other query distributions. The standard distance-based similarity tends to perform better in co-authorship and small world networks, since it can handle long-range queries appropriately. Interestingly, the hop-based measure is not far behind. In fact, on Twitter counting hops is consistently better than measuring the actual distances, indicating that the widely-used edge weighting scheme (inversely proportional to the frequency of communication) is not a good choice for this network.

The main drawback of the distance-based similarity measure is that evaluating long-range pair similarity requires costly Dijkstra computations. With ADS dist, we can approximate such distances in microseconds (see Table 1), which is orders of magnitude faster. Moreover, Tables 2 to 4 indicate that the results we obtain from both measures are comparable in quality. This is not surprising: for real-world social networks, we measured an average stretch below 10%. Adding randomized edge lengths (ADS dist REL) can lead to even better similarity estimates. This is particularly evident for small world networks, indicating that randomization is indeed effective in accounting for the underlying path multiplicity.

Remarkably, we can obtain even better results (comparable to RWR with tuned parameters) with our new ADS-based closeness similarity measure, which is just as cheap to compute. Comparing how two nodes are related to the entire network can be significantly more effective than measuring distances directly, especially when combined with randomized edge lengths. This is particularly true for Twitter. The method does have limitations, however. As Table 2 shows, closeness similarity does not work very well when nodes are far apart in the graph, as tends to be the case for pairs picked according to the uniform distribution in co-authorship networks (see Figures 3(a) and 3(b)). But in typical applications (such as community detection), when one is interested in ranking nearby (but not necessarily neighboring) nodes, ADS-based closeness similarity excels.

7. FINAL REMARKS

We have introduced global similarity measures based on all-distances sketches that are extremely efficient to compute. Our experiments on social networks with tens of millions of nodes show that these new measures are quite powerful in practice. We can compute the similarity between two arbitrary nodes in a few microseconds with accuracy that is at least as good as, and often better than, existing approaches. Another advantage is that our algorithms are simple to implement, naturally lending themselves to distributed and external memory scenarios, including within relational databases [2].

We stress that our experiments are done at full scale rather than with local subsamples of the networks. In fact, we observed that taking small samples of Twitter, for example, may significantly bias the results, leading to potentially inaccurate conclusions.

Given the diversity of social networks and their semantics, it is unrealistic to expect any single method based on structural properties of the graph to consistently dominate all others. The particular (global) approaches that we tested in our experiments are quite robust to the choice of input. That said, there are other possible instantiations of the parameters in our definitions of closeness similarity, as well as combinations with the REL techniques that we are planning to investigate and characterize further. A natural avenue for future work is to reduce the preprocessing effort (ADS computation), which is currently higher than we would like. Finally, another important direction is the extension of our techniques to computing seeded communities [1], where a group of nodes are presented and the task is to find a community spawning from this group.

8. REFERENCES

[1] I. Abraham, S. Chechik, D. Kempe, and A. Slivkins. Low-distortion inference of latent similarities from a multiplex social network. In *SODA*, pages 1853–1872, 2013.

[2] I. Abraham, D. Delling, A. Fiat, A. V. Goldberg, and R. F. Werneck. HLDB: Location-Based Services in Databases. In *GIS*, pages 339–348. ACM Press, 2012.

[3] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck. Hierarchical Hub Labelings for Shortest Paths. In *ESA*, volume 7501 of *LNCS*, pages 24–35. Springer, 2012.

[4] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27, 2005.

[5] T. Akiba, Y. Iwata, and Y. Yoshida. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *SIGMOD*, pages 349–360, 2013.

[6] P. Boldi, M. Rosa, and S. Vigna. HyperANF: approximating the neighbourhood function of very large graphs on a budget. In *WWW*, 2011.

[7] P. Boldi, M. Rosa, and S. Vigna. Robustness of social networks: Comparative results based on distance distributions. In *SocInfo*, pages 8–21, 2011.

[8] P. Boldi and S. Vigna. Studying network structures for IR: The impact of size. `http://ecir2012.upf.edu/ecir_paolo_boldi.pdf`, 2012.

[9] B. Bollobás. *Modern graph theory*. Springer, 1998.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[11] S. Chechik, Y. Emek, B. Patt-Shamir, and D. Peleg. Sparse reliable graph backbones. *Information and Computation*, 210:31 – 39, 2012.

[12] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, 55:441–453, 1997.

[13] E. Cohen. All-distances sketches, revisited: Scalable estimation of the distance distribution and centralities in massive graphs. Technical Report cs.DS/1306.3284, arXiv, 2013.

[14] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick. Reachability and distance queries via 2-hop labels. *SIAM J. Comput.*, 32(5):1338–1355, 2003.

[15] E. Cohen and H. Kaplan. Spatially-decaying aggregation over a network: model and algorithms. *J. Comput. Syst. Sci.*, 73:265–288, 2007.

[16] E. Cohen and H. Kaplan. Summarizing data using bottom-k sketches. In *PODC*, 2007.

[17] E. Cohen and H. Kaplan. A case for customizing estimators: Coordinated samples. Technical Report cs.ST/1212.0243, arXiv, 2012.

[18] E. Cohen and H. Kaplan. What you can do with coordinated samples. In *RANDOM*, 2013.

[19] P. Crescenzi, R. Grossi, L. Lanzi, and A. Marino. A comparison of three algorithms for approximating the distance distribution in real-world graphs. In *TAPAS*, 2011.

[20] C. Dangalchev. Residual closeness in networks. *Phisica A*, 365, 2006.

[21] A. Das Sarma, S. Gollapudi, M. Najork, and R. Panigrahy. A sketch-based distance oracle for web-scale graphs. In *WSDM*, pages 401–410, 2010.

[22] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *KDD*. ACM, 2002.

[23] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[24] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[25] E. S. Keeping. *Introduction to Statistical Inference*. Dover, 1962.

[26] J. M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC*. ACM, 2000.

[27] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007.

[28] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Phys. Rev. E 64, 025102*, 2001.

[29] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: a fast and scalable tool for data mining in massive graphs. In *KDD*, 2002.

[30] R. Panigrahy, M. Najork, and Y. Xie. How user behavior is related to social affinity. In *WSDM*, pages 713–722, 2012.

[31] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[32] K. A. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11, 1989.

[33] M. Thorup and U. Zwick. Approximate distance oracles. In *ACM STOC*, pages 183–192, 2001.

[34] H. Tong, C. Faloutsis, and J.-H. Pan. Fast random walk with restart and its applications. In *ICDM*. IEEE, 2006.

APPENDIX

This appendix contains a detailed example of a network and corresponding ADSs. For node h in the network in Figure 5, the forward ADS for $k = 1$ is $\overrightarrow{\text{ADS}}(h) = (h, e, k, a)$. For $k = 2$ we have $\overrightarrow{\text{ADS}}(h) = (h, e, g, f, i, k, a)$. Node j has forward distance $\overrightarrow{d}_{hj} = 11$ and forward Dijkstra rank $\overrightarrow{\pi}_{hj} = 4$ with respect to h. The backwards ADS from a for $k = 1$ is $\overleftarrow{\text{ADS}}(a) = (a)$ since a is the node with lowest rank. For $k = 2$, $\overleftarrow{\text{ADS}}(a) = (a, c, d, b, e, k)$.

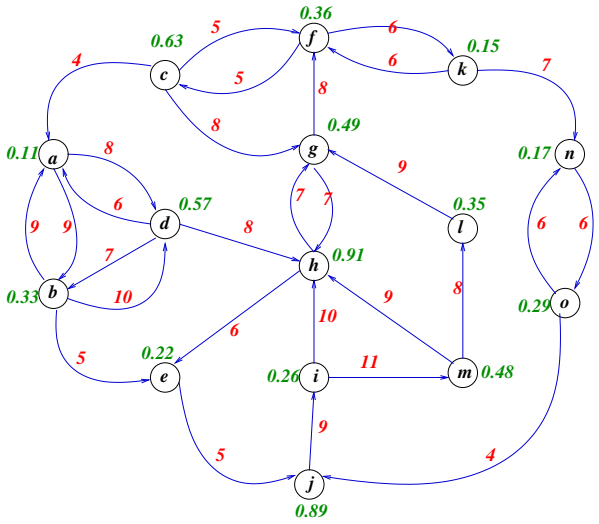


Figure 5: A directed network with lengths associated with edges and random rank values associated with its nodes. The nodes sorted by increasing distance from h, together with their distance, are $(h, 0)$, $(e, 6)$, $(g, 7)$, $(j, 11)$, $(f, 15)$, $(c, 20)$, $(i, 20)$, $(k, 21)$, $(a, 24)$, $(n, 28)$, $(m, 31)$, $(b, 31)$, $(d, 32)$, $(o, 34)$, $(l, 39)$. The nodes sorted by reversed increasing distance from node a are $(a, 0)$, $(c, 4)$, $(d, 6)$, $(b, 9)$, $(f, 9)$, $(g, 12)$, $(e, 14)$, $(k, 15)$, $(h, 16)$, $(j, 19)$, $(n, 22)$, $(o, 23)$, $(l, 25)$, $(i, 28)$, $(m, 39)$.