Detecting Critical Links in Complex Network to Maintain Information Flow/Reachability

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4,5]

[1] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp
[5] School of Computing and Information Systems, University of Tasmania

Abstract. We address the problem of efficiently detecting critical links in a large network. Critical links are such links that their deletion exerts substantial effects on the network performance. Here in this paper, we define the performance as being the average node reachability. This problem is computationally very expensive because the number of links is an order of magnitude larger even for a sparse network. We tackle this problem by using bottom-*k* sketch algorithm and further by employing two new acceleration techniques: marginal-link updating (MLU) and redundant-link skipping (RLS). We tested the effectiveness of the proposed method using two real-world large networks and two synthetic large networks and showed that the new method can compute the performance degradation by link removal about an order of magnitude faster than the baseline method in which bottom-*k* sketch algorithm is applied directly. Further, we confirmed that the measures easily composed by well known existing centralities, e.g. in/out-degree, betweenness, PageRank, authority/hub, are not able to detect critical links. Those links detected by these measures do not reduce the average reachability at all, i.e. not critical at all.

Keywords: Social networks, Link deletion, Critical links, Node reachability

1 Introduction

Studies of the structure and functions of large complex networks have attracted a great deal of attention in many different fields such as sociology, biology, physics and computer science [26]. It has been recognized that developing new methods/tools that enable us to quantify the importance of each individual node and link in a network is crucially important in pursuing fundamental network analysis. Networks mediate the spread of information, and it sometimes happens that a small initial seed cascades to affect large portions of networks [30]. Such information cascade phenomena are observed in many situations: for example, cascading failures can occur in power grids (*e.g.*, the August 10, 1996 accident in the western US power grid), diseases can spread over networks

of contacts between individuals, innovations and rumors can propagate through social networks, and large grass-roots social movements can begin in the absence of centralized control (*e.g.*, the Arab Spring). These problems have mostly been studied from the view point of identifying influential nodes under some assumed information diffusion model. There are other studies on identifying influential links to prevent the spread of undesirable things. See Section 2 for related work.

We study this problem from a slightly different angle in a more general setting. Which links are most critical in maintaining a desired network performance? For example, when the desired performance is to minimize contamination, the problem is reduced to detecting critical links to remove or block. When the desired performance is to maximize evacuation or minimize isolation, the problem is to detect critical links that reduce the overall performance if these links do not function. This problem is mathematically formulated as an optimization problem when a network structure is given and a performance measure is defined. In this paper we define the performance as being the average node reachability with respect to a link deletion, i.e. average number of nodes that are reachable from every single node when a particular link is deleted. The problem is to rank the links in accordance with the performance and identify the most critical link(s).

Since the core of the computation is to estimate reachability, an efficient method of counting reachable nodes is needed. We borrow the idea of bottom-k sketch [11, 12] which can estimate the number of reachable nodes quite efficiently by sampling a small number of nodes. Although it is very efficient, it still is computationally heavy when applied to our problem because we have to compute reachability from every single node for a particular link deletion and repeat this for all nodes and take the average. We repeat this for all the links and rank the results. To cope with this difficulty, we introduce two acceleration techniques called marginal-link updating (MLU) and redundant-link skipping (RLS). These are designed to improve the computational efficiency of bottom-k sketch.

We have tested our method using two real-world benchmark networks taken from Stanford Network Analysis Project and two synthetic networks which we designed to control the structural properties. We confirmed that about an order of magnitude reduction of computation time is obtained by use of these two acceleration techniques over a baseline method in which no acceleration techniques are used and bottom-k sketch algorithm is applied directly. We also analyzed which acceleration technique works better in which situations. We further investigated whether other measures which can easily be composed by the well known existing centralities can detect critical links. We composed four measures each computed by degree centrality, betweenness centrality, PageRank centrality and authority/hub centrality, respectively. These four measures rank the links very differently and those identified critical according to these measures do not reduce the performance at all, i.e. they are not critical by no means. This series of experiments confirm that the proposed method is unique and can efficiently detect critical links.

The paper is organized as follows. Section 2 briefly explains studies related to this paper. Section 3 revisits bottom-k sketch algorithm and introduces two new acceleration techniques. Section 4 reports four datasets used and the experimental results: computa-

tional efficiency and comparison with other measures. Section 5 summarizes the main achievement and future plans.

2 Related Work

Finding critical links in a network is closely related to the problem of efficiently preventing the spread of undesirable things such as contamination and malicious rumors by blocking links. An effective method of blocking a limited number of links in a social network was presented to solve the contamination minimization problem under a fundamental information diffusion model such as the independent cascade and the linear threshold models [17]. Many studies were also made on exploring effective strategies for reducing the spread of infection by removing nodes in a network [1, 5, 6, 25]. Moreover, we note that the contamination minimization problem can be converse to the influence maximization problem, which has recently attracted much interest in the field of social network mining [16, 19, 23, 8, 9, 15, 3, 29, 31].

To find critical links in a network, we consider quantifying how influential each link is. It is closely related to quantifying how influential each node is in the network. To this end, several node-centrality measures have been presented in the field of social network analysis. Representative node-centrality measures include degree centrality [14], HITS (hub and authority) centrality [7], PageRank centrality [4] and betweenness centrality [14]. Here, note that for some node-centrality measures such as betweenness centrality, their computation becomes harder as the network size increases, since it needs to take the global network structure into account. Thus, several researchers presented methods of approximating such node-centralities [2, 27, 10]. Moreover, given an information diffusion model on a social network, influence degree centrality can be defined by evaluating the influence of each node. Unlike node-centrality measures derived only from network topology, influence degree centrality exploits a dynamical process on the network as well. An efficient method of simultaneously estimating the influence degrees of all the nodes was presented under the SIR model setting [22]. We note that influence degree centrality can also be employed for identifying super-mediators of information diffusion in the social network [28]. In this paper, we propose a method of efficiently evaluating how critical each link is in the network (i.e., calculating our new link-centrality measure). Since conventional node-centrality measures can naturally derive link-centrality measures, we also compare the proposed link-centrality measure with those link-centrality measures (see Section 4.3).

A bottom-k sketch [11, 12] used in this paper is a summary of a set of nodes, which is obtained by associating with each node in a network an independent random rank value drawn from a probability distribution. The bottom-k estimator includes the k smallest rank values, and the kth smallest one is used for the estimation. This estimate has a Coefficient of Variation (CV), which is the ratio of the standard deviation to the mean, that is never more than $1/\sqrt{k-2}$ and is well concentrated [11]. We can quite efficiently calculate the bottom-k sketch of each node in the network by orderly assigning the rank values from the smallest one to those nodes reachable by reversely following links over the network. Based on this framework, a greedy Sketch-based Influence Maximization (SKIM) algorithm has been proposed, and it has been shown that the

SKIM algorithm scales to graphs with billions of edges, with one to two orders of magnitude speedup over the best greedy methods [13]. Thus, we also develop our method of detecting critical links under the framework of the bottom-k sketching algorithm.

3 Proposed Method

Let $G = (\mathcal{V}, \mathcal{E})$ be a given simple network without self-loops, where $\mathcal{V} = \{u, v, w, \cdots\}$ and $\mathcal{E} = \{e = (u, v), f, g, \cdots\}$ are sets of nodes and directed links, respectively. Let $\mathcal{R}(v; G)$ and $\mathcal{Q}(v; G)$ be the sets of reachable nodes by forwardly and reversely following links from a node v over G, respectively, where note that $v \in \mathcal{R}(v; G)$ and $v \in \mathcal{Q}(v; G)$. Also, let $\mathcal{R}_1(v; G)$ and $\mathcal{Q}_1(v; G)$ be the sets of those nodes adjacent to v, i.e., $\mathcal{R}_1(v; G) = \{w \in \mathcal{R}(v; G) \mid (v, w) \in \mathcal{E}\}$ and $\mathcal{Q}_1(v; G) = \{u \in \mathcal{Q}(v; G) \mid (u, v) \in \mathcal{E}\}$, respectively. Here, we briefly revisit the bottom-k sketch [11, 12] and describe the way to estimate the number of the reachable nodes from each node $v \in \mathcal{V}$, i.e., $|\mathcal{R}(v; G)|$. First, we assign to each node $v \in \mathcal{V}$ a value $r(v)$ uniformly at random in $[0, 1]$. When $|\mathcal{R}(v; G)| \geq k$, let $\mathcal{B}_k(v; G)$ be the subset of the k smallest elements in $\{r(w) \mid w \in \mathcal{R}(v; G)\}$, and $b_k(v; G) = \max \mathcal{B}_k(v; G)$ be the k-th smallest element. Then, we can unbiasedly estimate the number of the reachable nodes from v by $H(v; G) = |\mathcal{B}_k(v; G)|$ if $|\mathcal{B}_k(v; G)| < k$ [1]; otherwise $H(v; G) = (k - 1)/b_k(v; G)$. Here note that for any $c > 0$, it is enough to set $k = (2 + c)\epsilon^{-2} \log |\mathcal{V}|$ to have a probability of having relative error larger than ϵ bounded by $|\mathcal{V}|^{-c}$ [11, 12]. Here, we can efficiently calculate the bottom-k sketch $\mathcal{B}_k(v; G)$ for each node $v \in \mathcal{V}$ by reversely following links $k|\mathcal{E}|$ times. Namely, we first initialize $\mathcal{B}_k(v; G) \leftarrow \emptyset$ and sort the random values as $(r(v_1), \cdots, r(v_i), \cdots, r(v_{|\mathcal{V}|}))$ in ascending order, i.e., $r(v_i) \leq r(v_{i+1})$. Then, from $i = 1$ to $|\mathcal{V}|$, for $w \in \mathcal{Q}(v_i; G)$, we repeatedly insert $r(v_i)$ into $\mathcal{B}_k(w; G)$ by reversely following links from v_i if $|\mathcal{B}_k(w; G)| < k$.

As described earlier, we focus on the problem of detecting a critical link $\hat{e} \in \mathcal{E}$, where the average number of reachable nodes maximally decreases by its removable. Let $G_e = (\mathcal{V}, \mathcal{E} \setminus \{e\})$ be the network obtained by removing a link e, then we can define the following objective function to be minimized with respect to $e \in \mathcal{E}$.

$$F_0(G_e) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} |\mathcal{R}(v; G_e)|. \tag{1}$$

In this paper, by using the estimation based on the bottom-k sketches, we focus on the following objective function.

$$F(G_e) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} H(v; G_e). \tag{2}$$

Here we can straightforwardly obtain a baseline method which re-calculates the bottom-k sketches, $\mathcal{B}_k(v; G_e)$, with respect to G_e for all nodes from scratch. However, the baseline method generally requires a large amount of computation for large-scale networks.

[1] $\mathcal{B}_k(v; G)$ can still be defined when $|\mathcal{R}(v; G)| < k$. In this case its cardinality is the number of reachable nodes from v.

In order to overcome this problem, by borrowing and extending the basic ideas of pruning techniques proposed in [21, 22], below we propose new acceleration techniques called marginal-link updating (MLU) and redundant-link skipping (RLS).

The MLU technique locally updates the bottom-k sketches of some nodes when removing links incident to a node with in-degree 0 or out-degree 0 in the network G. First, let $v \in \mathcal{V}$ be a node with in-degree 0, i.e., $|\mathcal{Q}_1(v;G)| = 0$. Here, note that by removal of a link from v to its child node w, say $e = (v,w)$ and $w \in \mathcal{R}_1(v;G)$, only the bottom-k sketch of node v changes, i.e., $\mathcal{B}_k(u;G) = \mathcal{B}_k(u;G_e)$ for any node $u \neq v$. Namely, we can locally update the bottom-k sketch of node v by computing $\mathcal{B}_k(v;G_e)$ as the k smallest elements in $\cup_{w \in \mathcal{R}_1(v;G_e)} \mathcal{B}_k(w;G)$. On the other hand, let $v \in \mathcal{V}$ be a node with out-degree 0, i.e., $|\mathcal{R}_1(v;G)| = 0$, then the bottom-k sketch of node v is $\mathcal{B}_k(v;G) = \{r(v)\}$. Here, note that by removal of a link to v from its parent node u, say $e = (u,v)$ and $u \in \mathcal{Q}_1(v;G)$, only the bottom-k sketch of node x such that $x \in \mathcal{Q}(v;G) \setminus \mathcal{Q}(v;G_e)$ possibly changes. Thus, by computing the bottom-$(k+1)$ sketch of any node $u \in \mathcal{V}$, i.e., $\mathcal{B}_{k+1}(u;G)$, in advance, we can locally update the bottom-k sketch of such a node x just by replacing $r(v) \in \mathcal{B}_k(x;G)$ with $b_{k+1}(x;G)$ unless $|\mathcal{B}_{k+1}(x;G)| \leq k$, by reversely following links from v as performed in the bottom-k sketches calculation.

The RLS technique selects each link $e \in \mathcal{E}$ for which $F(G_e) = F(G)$ and prune some subset of such links. Here, we say that a link $e = (v,w) \in \mathcal{E}$ is a *skippable link* if there exist some node $x \in \mathcal{V}$ such that $f = (v,x) \in \mathcal{E}$ and $g = (x,w) \in \mathcal{E}$, i.e., $x \in \mathcal{R}_1(v;G) \cap \mathcal{Q}_1(w;G)$, which means $|\mathcal{R}_1(v;G) \cap \mathcal{Q}_1(w;G)| \geq 1$. Namely, we can skip evaluating $F(G_e)$ for the purpose of solving our problem due to $F(G_e) = F(G)$. Moreover, we say that a link $e = (v,w) \in \mathcal{E}$ is a *prunable link* if $|\mathcal{R}_1(v;G) \cap \mathcal{Q}_1(w;G)| \geq 2$. Namely, we can prune such a link e for our problem by setting $G \leftarrow G_e$ due to $F((G_e)_f) = F(G_f)$ for any link $f \in \mathcal{E}$. For each node $v \in \mathcal{V}$, let $\mathcal{S}(v)$ and $\mathcal{P}(v)$ be sets of skippable and prunable links from v. We can calculate $\mathcal{S}(v)$ and $\mathcal{P}(v)$ as follows: for each child node $w \in \mathcal{R}_1(v;G)$, we first initialize $c(v,w;G) \leftarrow 0$, $\mathcal{S}(v) \leftarrow \emptyset$ and $\mathcal{P}(v) \leftarrow \emptyset$. Then, for each node $x \in \mathcal{R}_1(v;G)$, we repeatedly set $c(v,w;G) \leftarrow c(v,w;G) + 1$ and $\mathcal{S}(v) \leftarrow \mathcal{S}(v) \cup \{(v,w)\}$ if $\{w\} \in \mathcal{R}_1(x;G)$, and set $\mathcal{P}(v) \leftarrow \mathcal{P}(v) \cup \{(v,w)\}$ and $G \leftarrow G_{(v,w)}$ if $c(v,w;G) \geq 2$.

In our proposed method, the RLS technique is applied before the MLU techniques, because it is naturally conceivable that the RLS technique decreases the number of links in our network G. Clearly we can individually incorporate these techniques into the baseline method. Hereafter, we refer to the proposed method without the MLU technique as the RLS method, and the proposed method without the RLS technique as the MLU method. Since it is difficult to analytically examine the effectiveness of these techniques, we empirically evaluate the computational efficiency of these three methods in comparison to the baseline method.

4 Experiments

We evaluated the effectiveness of the proposed method using two benchmark and two synthetic networks.

Table 1. Basic statistics of networks.

No.	name	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|\mathcal{I}_0\|$	$\|\mathcal{O}_0\|$	$\|\mathcal{S}\|$	$\|\mathcal{P}\|$
1	CIT	34,546	421,578	2,393	6,320	302,248	176,224
2	DBA	35,000	351,317	5,984	5,999	85,815	24,690
3	DCN	35,000	350,807	4,996	8,868	289,398	175,211
4	P2P	36,682	88,328	26,960	229	1,502	29

4.1 Datasets

We employed two benchmark networks obtained from SNAP (Stanford Network Analysis Project) [2]. The first one is a high-energy physics citation network from the e-print arXiv [3], which covers all the citations within a dataset of $34,546$ papers (nodes) with $421,578$ citations (links). If a paper u cites paper v, the network contains a directed link from u to v. The second one is a sequence of snapshots of the Gnutella peer-to-peer file sharing network from August 2002 [4]. There are total of 9 snapshots of Gnutella network collected in August 2002. The network consists of $36,682$ nodes and $88,328$ directed links, where nodes represent hosts in the Gnutella network topology and links represent connections between the Gnutella hosts.

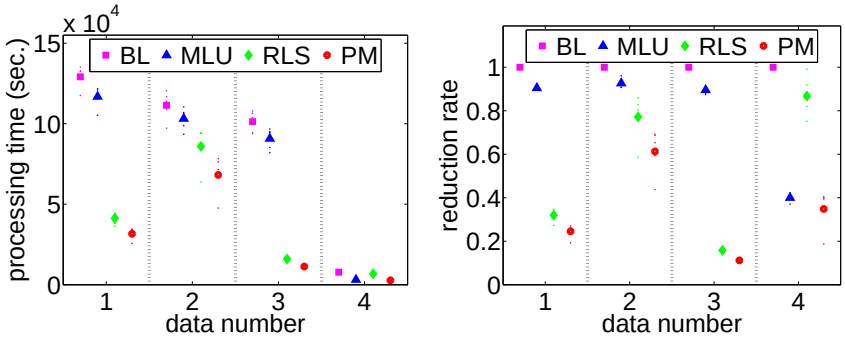
In addition, we utilized two synthetic networks with a DAG (Directed Acyclic Graph) property, which were generated by using the DCNN and DBA methods described in [21, 22], respectively. For the sake of convenience, we briefly revisit these methods. First, we explain the DCNN method. Here, we say that a pair of nodes $\{v, w\}$ is a potential pair if they are not directly connected, but have at least one common adjacent node. Then, we can summarize the DCNN method as an algorithm which repeats the following steps from a single node and an empty set of links while $|\mathcal{V}| < L$: 1) With probability $1 - \delta$, create a new node $u \in \mathcal{V}$, select a node $v \in \mathcal{V}$ at random, and add a link (u, v) or (v, u) arbitrary; 2) With probability δ, select a potential pair $\{v, w\}$ at random, and add a link (v, w) or (w, v) to be a DAG direction. Clearly, we can easily see that the DCNN method generates a DAG. In our experiments, we set $L = 35,000$ and $\delta = 0.1$ to make sure that the numbers of nodes and links can be roughly equal to $|\mathcal{V}| = 35,000$ and $|\mathcal{E}| = 350,000$, which can be a network with an intermediate size between the above two benchmark networks.

Next, we explain the DBA method. Here, we say that a node is selected by preferential attachment if its selection probability is proportional to the number of adjacent nodes. Then, we can summarize the DBA method as an algorithm which repeats the following steps from a DAG having M links generated by the DCNN method while $|\mathcal{V}| < L$: 1) With probability $1 - \delta$, create a new node $u \in \mathcal{V}$, select a node $v \in \mathcal{V}$ by preferential attachment, and create a link (u, v) or (v, u) arbitrary. 2) With probability δ, select a node $v \in \mathcal{V}$ at random, select another node $w \in \mathcal{V}$ by preferential attachment, and create a link (v, w) or (w, v) to be a DAG direction. Again, we can easily see that the DBA method generates a DAG. In our experiments, we also set $L = 35,000$, $\delta = 0.1$, and $M = 100$.

[2] https://snap.stanford.edu/

[3] https://snap.stanford.edu/data/cit-HepPh.html

[4] https://snap.stanford.edu/data/p2p-Gnutella30.html



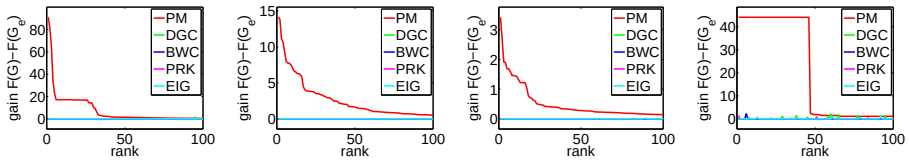
(a) Processing time (b) Reduction rate

Fig. 1. Computation time comparison.

In what follows, we refer to these two benchmark networks of citation and pear-to-pear and those generated by the DCNN an DBA methods as CIT, P2P, DCN and DBA networks. Table 1 summarizes the basic statistics of these networks, consisting of the numbers of nodes and links, $|\mathcal{V}|$ and $|\mathcal{E}|$, the numbers of in-degree 0 and out-degree 0 nodes, $|\mathcal{I}_0|$ and $|\mathcal{O}_0|$, and the numbers of skippable and prunable links, $|\mathcal{S}|$ and $|\mathcal{P}|$, where each network is also identified by its data number as shown in Tab. 1. From this table, we can conjecture that the RLS technique works well for the CIT and DCN networks, while the MNU technique for the P2P networks. Here note that the numbers of skippable and prunable links appearing in the networks generated by the DCNN method inevitably become larger than those generated by the DBA method because the DCNN method has a link creation mechanism between potential pairs.

4.2 Computational Efficiency

First, we evaluated the efficiency of the proposed method which calculates $F(G_e)$ for each link $e \in \mathcal{E}$. We compared the computation time of the baseline (BL), RLS, MLU, and proposed (PM) methods by performing five trials. Here, we used the same random value $r(v)$ assignment for each trial so that the bottom-k sketches of all the nodes are the same for any method, i.e., it is guaranteed that each method can produce the same result. Figure 1 shows the computation times of each method for five trials plotted by dots and the average values over these trials plotted by different markers as indicated in the figure, where we set $k = 64$ for calculation of the bottom-k sketches of all the nodes according to [13]. Figure 1(a) compares the actual processing times of these methods, where our programs implemented in C were executed on a computer system equipped with two Xeon X5690 3.47GHz CPUs and a 192GB main memory with a single thread within the memory capacity. Figure 1(b) compares the reduction rates of computation times for these methods from the BL method.



(a) Data 1 CIT (b) Data 2 DBA (c) Data 3 DCN (d) Data 4 P2P

Fig. 2. gain comparison of extracted links.

From Fig. 1(a), we can see that the computation times were improved largely for the CIT and CNN networks, modestly for the P2P network, and much less modestly for the DBA network, although the computation time of the BL method for the P2P network was smaller than those for the other networks. More specifically, as expected, we consider that the RLS technique worked quite well especially for the CIT and DCN networks, due to large numbers of skippable and prunable links in these networks as shown in Tab. 1. On the other hand, although the MLU technique is not so remarkably effective, we consider that this technique can steadily improve the reduction rate of computation times especially for the P2P network as shown in Fig. 1(b). In short, we can conjecture that the proposed method combining both the RLS and MLU techniques is more reliable than the other three methods in terms of computation time because it produced the best performance for all of the four networks. Reduction of computation time depends on network structures, but overall we can say that use of both techniques can increase the computational efficiency by about an order of magnitude. These results demonstrate the effectiveness of the proposed method.

4.3 Comparison with Conventional Centralities

As noted earlier, by solving our critical link detection problem that the average number of reachable nodes maximally decreases by link removable, we can obtain the value $F(G_e)$ for each link $e \in \mathcal{E}$ as a measure to evaluate the criticalness of the link e. Thus, we evaluated whether or not our measure $F(G_e)$ can actually provide a novel concept in comparison with some measures derived from conventional centralities.

As conventional centralities, we examined the degree centrality, the betweenness centrality, the PageRank centrality, and the eigenvalue centrality for network G, and straightforwardly extended these centralities so as to evaluate the criticalness of a given link e. The first measure $DGC(e)$ derived from degree centrality for a given link $e = (u, v) \in \mathcal{E}$ is defined as

$$DGC(e) = |\mathcal{Q}_1(u; G)| \times |\mathcal{R}_1(v; G)|, \tag{3}$$

where recall that $\mathcal{Q}_1(u; G)$ and $\mathcal{R}_1(v; G)$ are in-degree of node u and out-degree of node v, respectively. Namely, the first measure $DGC(e)$ highly evaluate a link from a node

with high in-degree to a node with high out-degree. Next, for a given link $e = (u, v) \in \mathcal{E}$, the second measure $BWC(e)$ derived from the betweenness centrality is defined as

$$BWC(e) = btw(u) \times btw(v), \quad btw(v) = \sum_{w \in \mathcal{V}} \sum_{x \in \mathcal{V}} \frac{nsp^G_{w,x}(v)}{nsp^G_{w,x}}, \tag{4}$$

where $btw(v)$ stands for the betweenness of a node $v \in \mathcal{V}$ and $nsp^G_{w,x}$ is the total number of the shortest paths between node w and node x in G and $nsp^G_{w,x}(v)$ is the number of the shortest paths between node w and node x in G that passes through node v. Namely, the second measure $BWC(e)$ highly evaluate a link between nodes with high betweenness centrality scores. Going on next, for a given link $e = (u, v) \in \mathcal{E}$, the third measure $PRK(e)$ derived from the PageRank centrality is defined as

$$PRK(e) = prk(u) \times prk(v), \tag{5}$$

where $prk(v)$ stands for the PageRank score of a node $v \in \mathcal{V}$, which is provided by applying the PageRank algorithm with random jump factor 0.15 [4]. Namely, the third measure $PRK(e)$ highly evaluates a link between nodes with high PageRank scores. Finally, for a given link $e = (u, v) \in \mathcal{E}$, the fourth measure $EIG(e)$ derived from eigenvector centrality is defined as

$$EIG(e) = auth(u) \times hub(v), \tag{6}$$

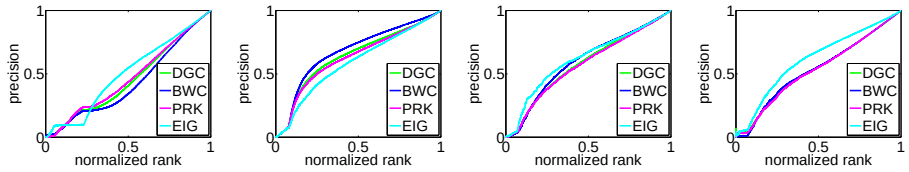
where $auth(u)$ and $hub(v)$ respectively stand for the authority and the hub scores of nodes $u, v \in \mathcal{V}$, which is provided by applying the HITS algorithm [7]. Namely, the fourth measure $EIG(e)$ highly evaluates a link from a node with a high hub score to a node with high authority score.

First, we examined how each of highly ranked links by these standard centralities, *i.e.*, DGC, BWC, PRK, and EIC, can decrease the average number of reachable nodes by its removal. Here, we measured the performance by the following gain $F(G) - F(G_e)$

$$F(G) - F(G_e) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} H(v; G) - \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} H(v; G_e). \tag{7}$$

Figures 2 shows our experimental results, where the vertical and horizontal axes stand for the rank until top-100 and the gain, respectively, and Figs. 2(a), 2(b), 2(c), and 2(d) correspond to the CIT, DBA, DCN, and P2P networks. We can see that our proposed method (denoted by PM) detected critical links having substantial amount of gains, where the curves for the top-100 links are somewhat different from each other depending on the network datasets. On the other hand, the gains for all of the top-100 links by those measures defined by the standard centralities were almost zeros for any dataset. In addition, we should emphasize that the gain curves shown in Fig. 2 could uncover some characteristics of these networks, i.e., the DCN network was relatively robust to a single link removal, and so on.

Next, we examined the similarity between our ranking based on $F(G_e)$ and the other ranking, *i.e.*, the DGC, BWC, PRK, and EIC ranking. Here, we measured the similarity



(a) Data 1 CIT (b) Data 2 DBA (c) Data 3 DCN (d) Data 4 P2P

Fig. 3. precision comparison of extracted links.

Table 2. Ranks by conventional cenralities to top-3 links by proposed measure.

	CIT				DBA			
rank	DGC	BWC	PRK	EIG	DGC	BWC	PRK	EIG
1	233,638	296,250	161,504	405,661	188,686	309,435	251,492	218,665
2	189,768	424	26,626	402,823	301,228	335,210	309,249	289,465
3	67,333	106,045	2,158	141,559	276,667	320,951	288,338	217,370
	DCN				P2P			
rank	DGC	BWC	PRK	EIG	DGC	BWC	PRK	EIG
1	284,501	269,488	252,472	168,049	42,915	88,077	82,028	81,162
2	302,753	296,190	293,284	312,283	44,008	79,543	75,022	75,128
3	311,040	311,040	338,940	186,370	74,710	74,947	85,107	65,655

between the top j links for our ranking method, denoted as a set $\mathcal{A}_j$, and those for the other ranking method, denoted as a set $\mathcal{A}'_j$, by the precision $Prec(j)$ defined by

$$Prec(j) = \frac{|\mathcal{A}_j \cap \mathcal{A}'_j|}{j}. \tag{8}$$

Figures 3 shows our experimental results, where the vertical and horizontal axes stand for normalized rank for all links and the precision, respectively, and Figs. 3(a), 3(b), 3(c), and 3(d) are those of the CIT, DBA, DCN, and P2P networks. We can see that the results were quite similar to each other regardless of any pair of the centrality measures and the networks although a slightly better precision curve swelling in the upper left corner was obtained especially for the DBA network.

Finally, Table 2 shows ranks by conventional centralities to top-3 links obtained by proposed measure. These results indicate that the rankings by these convention measures were substantially different from those of our measure. Namely, these experimental results suggest that our measure $F(G_e)$ could actually provide a novel concept in comparison with some measures derived from conventional centralities.

5 Conclusion

In this paper we have proposed a novel computational method that can detect critical links quite efficiently for a large network. The problem is reduced to finding a link that

reduces the network performance substantially with respect to its removal. Such a link is considered critical in maintaining the good performance. There are many problems that can be mapped to this critical link detection problem, e.g. contamination minimization be it physical or virtual, evacuation trouble minimization, road maintenance prioritization, etc.

Network performance varies with specific problem, but in general it is represented by the reachability performance, i.e. how many nodes are reachable from a node in the network on the average. This brings in computational issue because reachability must be estimated for all the nodes for a particular link removal and to find critical links this has to be repeated for all the links. The number of links is generally an order of magnitude larger than the number of nodes even for a sparse network that is encountered in actual practice. We used bottom-k sketch algorithm as a basis to count reachable nodes, which only uses k-samples to estimate the reachable nodes from a selected node. It has a sound theoretical background and been shown quite efficient and accurate for a k which is far smaller than the number of nodes in the network. Our contribution is to introduce two new acceleration techniques to further reduce the bottom-k sketch computation by clever local update and redundant computations pruning. The first technique MLU (marginal-link updating) locally updates the bottom-k sketches of some nodes when removing links incident to a node with in-degree 0 or out-degree 0 in the network. The second techniqu RLS (redundant-link skipping) selects each link that does not affect the performance with respect to its removal and prune some subset of such links.

We have tested the performance of the proposed method using four networks with about 35, 000 nodes and 90,000 to 420,000 links. Two were taken from Stanford Network Analysis Project and the other two were artificially generated to control the network structure. We verified that the acceleration techniques indeed work for all these four networks of different characteristics and can reduce the computation time by about an order of magnitude. MLU works better for networks with many nodes with in-degree 0 or out-degree 0. RLS works better for networks with many prunable links. We have further evaluated how other measures based on conventional centralities work in estimating our performance measure, i.e. the average number of reachable nodes by a link removal. We have composed four measures, each based on degree centrality, betweenness centrality, PageRank centrality and authority/hub centrality, respectively. All of these measures are not able to detect critical links that were detected by the proposed method. Links detected by these measures do not show any performance degradation, i.e. not critical at all. We can conclude that no existing measure can find critical links.

There are many things to do. Reachability computation is a basic operation and is a basis for many applications. We continue to explore techniques to further reduce computation time. Our immediate future plan is to apply our method to a real world application and show that it can solve a difficult problem efficiently, e.g. identifying important hot spots in transportation network or evacuation network.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under award

number FA2386-16-1-4032, and JSPS Grant-in-Aid for Scientific Research (C) (No. 26330261).

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature 406, 378–382 (2000)
2. Boldi, P., Vigna, S.: In-core computation of geometric centralities with hyperball: A hunderd billion nodes and beyond. In: Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops (ICDMW'13). pp. 621–628 (2013)
3. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14). pp. 946–957 (2014)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30, 107–117 (1998)
5. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. In: Proceedings of the 9th International World Wide Web Conference. pp. 309–320 (2000)
6. Callaway, D.S., Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Network robustness and fragility: Percolation on random graphs. Physical Reveiw Letters 85, 5468–5471 (2000)
7. Chakrabarti, S., Dom, B., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J.: Mining the web's link structure. IEEE Computer 32, 60–67 (1999)
8. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09). pp. 199–208 (2009)
9. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10). pp. 88–97 (2010)
10. Chierichetti, F., Epasto, A., Kumar, R., Lattanzi, S., Mirrokni, V.: Efficient algorithms for public-private social networks. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15). pp. 139–148 (2015)
11. Cohen, E.: Size-estimation framework with applications to transitive closure and reachability. Journal of Computer and System Sciences 55, 441–453 (1997)
12. Cohen, E.: All-distances sketches, revisited: Hip estimators for massive graphs analysis. In: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 88–99 (2015)
13. Cohen, E., Delling, D., Pajor, T., Werneck, R.F.: Sketch-based influence maximization and computation: Scaling up with guarantees. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 629–638 (2014)
14. Freeman, L.: Centrality in social networks: Conceptual clarification. Social Networks 1, 215–239 (1979)
15. Goyal, A., Bonchi, F., Lakshmanan, L.: A data-based approach to social influence maximization. Proceedings of the VLDB Endowment 5(1), 73–84 (2011)
16. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03). pp. 137–146 (2003)
17. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, 9:1–9:23 (2009)

18. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions fot sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). pp. 2046–2051 (2009)
19. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). pp. 1371–1376 (2007)
20. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20, 70–97 (2010)
21. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Efficient analysis of node influence based on sir model over huge complex networks. In: Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA'14). pp. 216–222 (2014)
22. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Speeding-up node influence computation for huge social networks. International Journal of Data Science and Analytics 1, 1–14 (2016)
23. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). pp. 420–429 (2007)
24. Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., Hurst, M.: Patterns of cascading behavior in large blog graphs. In: Proceedings of 2007 SIAM International Conference on Data Mining (SDM'07). pp. 551–556 (2007)
25. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E 66, 035101 (2002)
26. Newman, M.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
27. Ohara, K., Saito, K., Kimura, M., Motoda, H.: Resampling-based framework for estimating node centrality of large social network. In: Proceedings of the 17th International Conference on Discovery Science (DS'14). pp. 228–239. LNAI 8777 (2014)
28. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Super mediator - a new centrality measure of node importance for information diffusion over social network. Information Sciences 329, 985–1000 (2016)
29. Song, G., Zhou, X., Wang, Y., Xie, K.: Influence maximization on large-scale mobile social network: A divide-and-conquer method. IEEE Transactions on Parallel and Distributed Systems 26, 1379–1392 (2015)
30. Watts, D.: A simple model of global cascades on random networks. Proceedings of the National Academy of Sciences of the United States of America 99, 5766–5771 (2002)
31. Zhou, C., Zhang, P., Zang, W., Guo, L.: On the upper bounds of spread for greedy algorithms in social network influence maximization. IEEE Transactions on Knowledge and Data Engineering 27, 2770–2783 (2015)