ATTACKING SIMILARITY-BASED LINK PREDICTION IN SOCIAL NETWORKS

Kai Zhou
Computer Science and Engineering,
Washington University in St. Louis
St. Louis, MO
zhoukai@wustl.edu

Tomasz P. Michalak
Institute of Informatics, University of Warsaw
Warsaw, Poland
tpm@mimuw.edu.pl

Marcin Waniek
Computer Science, Khalifa University of Science and Technology
Abu Dhabi, UAE
mjwaniek@gmail.com


ABSTRACT
Link prediction is one of the fundamental problems in computational social science. A particularly common means to predict existence of unobserved links is via structural similarity metrics, such as the number of common neighbors; node pairs with higher similarity are thus deemed more likely to be linked. However, a number of applications of link prediction, such as predicting links in gang or terrorist networks, are adversarial, with another party incentivized to minimize its effectiveness by manipulating observed information about the network. We offer a comprehensive algorithmic investigation of the problem of attacking similarity-based link prediction through link deletion, focusing on two broad classes of such approaches, one which uses only local information about target links, and another which uses global network information. While we show several variations of the general problem to be NP-Hard for both local and global metrics, we exhibit a number of well-motivated special cases which are tractable. Additionally, we provide principled and empirically effective algorithms for the intractable cases, in some cases proving worst-case approximation guarantees.

KEYWORDS
Computational social science; link prediction; security and privacy; adversarial attacks

ACM Reference Format:

1 INTRODUCTION
Link prediction is a fundamental problem in social network analysis. A common approach to predicting a target link \((u, v)\) is to use an observed (sub)network to infer the likelihood of the existence of this link using a measure of similarity, or closeness, of \(u\) and \(v\); we call this similarity-based link prediction \([1, 11, 19, 25]\). For example, if \(u\) and \(v\) are individuals who have many friends in common, it may be natural to assume that they are themselves friends. Representational power of social networks implies very broad application of link prediction techniques, ranging from friend recommendations to inference of criminal and terrorist ties. A crucial assumption in conventional similarity-based link prediction approaches is that the observed (sub)network is measured correctly. However, insofar as link prediction may reveal relationships which associated parties prefer to keep hidden—either for the sake of privacy, or to avoid being apprehended by law enforcement—it introduces incentives to manipulate network measurements in order to reduce perceived similarity scores for target links.

In order to systematically study the ability of an “adversary” to manipulate link prediction, we formulate attacks on link prediction as an optimization problem in which the adversary aims to minimize the total weighted similarity scores of a set of target links by removing a limited subset of edges from the observed subnetwork. We present a comprehensive study of this algorithmic problem, focusing on two important subclasses of similarity metrics: local metrics, which make use of only local information about the target link, and global metrics, which use global network information. We show that the problem is in general NP-Hard even for local metrics, and our hardness results are stronger for the commonly used Katz and ACT global similarity metrics (for example, the problem is hard for these metrics even if there is only a single target link).

On the positive side, we exhibit a number of important special cases when the problem is tractable. These include attacks on local metrics when there is a single target link, or a collection of target nodes (such as gang members) with the goal of hiding links among them. Additionally, we present practical algorithms for the intractable cases, including global similarity metrics. In a number of such settings, we are able to provide provable approximation guarantees. Finally, we demonstrate the effectiveness of the approaches we develop through an extensive experimental evaluation.

Related Work. Link prediction has been extensively studied in multiple domains such as social science \([11]\), bioinformatics \([2]\), and security \([7]\). There are two broad classes of approaches for link prediction: the first based on structural similarity \([11, 13]\) and the second using learning \([1, 14, 18, 20]\). This work is focused on the
former, which commonly use either local information \([9, 28]\), rely on paths between nodes \([8, 12]\), or make use of random walks \([4]\) (we view the latter two categories as examples of global metrics).

Our work is connected to several efforts studying vulnerability of social network analysis (SNA). Michalak et al. \([15]\) suggest considering strategic considerations in SNA, but do not offer algorithmic analysis. Waniek et al. study attacks against centrality measures and community detection \([21, 22]\). There is considerable literature on hiding or anonymizing links on networks (e.g., \([23, 24, 26]\)), but these approaches allow arbitrary graph modifications and are in any case heuristic, often proposing randomly swapping or rerouting edges. In contrast, we provide the first comprehensive algorithmic study of the problem of hiding links by merely deleting observed edges (i.e., preventing them from being observed), and the first strong positive algorithmic results.

2 PROBLEM FORMULATION

2.1 Similarity Metrics

One of the major approaches for link prediction both in the network science literature and in practice is via the use of similarity metrics \([11]\). Specifically, suppose we wish to know whether a particular link \((u, v)\) connecting nodes \(u\) and \(v\) exists. A structural similarity metric \(\text{Sim}(u, v)\) quantifies the extent to which the nodes \(u\) and \(v\) have shared topological properties, such as shared neighbors, with the idea that higher similarity scores imply greater likelihood that \(u\) and \(v\) are connected. Below, we will distinguish two types of similarity metrics: local, which only use information about the nodes and their immediate neighbors, and global, which make use of global information about the network.

2.2 Attack Model

At the high level, our goal is to remove a subset of observed edges in order to minimize perceived similarity scores of a collection of target (and, presumably, existing) links. This could be viewed both from the perspective of vulnerability analysis, where the goal of link prediction is to identify relationships among malicious parties (such as gang members), or privacy, where the “attacker” is not malicious, but rather aims to preserve privacy of a collection of target relationships.

To formalize the problem, consider an underlying graph \(G = (V, E)\) representing a social network, where \(V\) is the set of nodes and \(E\) is the set of edges. This graph is not fully known, and instead an analyst obtains answers for a collection of edge queries \(Q\) from the environment, where for each query \((u, v) \in Q\), they observe the associated edge if \((u, v) \in E\), and determine that the edge doesn’t exist otherwise. The partially constructed graph \(\hat{G}_Q = (V_Q, E_Q)\) based on the queries \(Q\) is then used to compute similarity metrics \(\text{Sim}(u', v')\) for any potential edges \((u', v') \notin Q\).

An attacker has a collection of target links \(H\) they wish to hide, and can remove a subset of at most \(k\) edges in \(E_Q \equiv E \cap Q\) to this end. While there are many ways to express the attacker’s objective mathematically, a relatively natural and general approach is to minimize the weighted sum of similarity scores of links in \(H\):

\[
\min_{E_a \subseteq E_Q} \sum_{(u, v) \in H} w_{uv} \text{Sim}(u, v; E_a), \quad \text{s.t. } |E_a| \leq k, \tag{1}
\]

where \(w_{uv}\) is the weight representing the relative importance of hiding the link \((u, v)\), and we make explicit the dependence of similarity metrics on the set of removed edges \(E_a\). Henceforth, we simplify notation by keeping this dependence implicit.

3 ATTACKING LOCAL SIMILARITY METRICS

Our analysis covers nine representative local similarity metrics (summarized in the supplement) that are commonly used in the state-of-the-art link prediction algorithms. We first systematically divide local metrics into two sub-class: Common Neighbor Degree (CND) and Weighted Common Neighbor (WCN) metrics, depending on their special structures. Next, we show that attacking all local metrics is NP-Hard. We follow this negative result with an approximation algorithm exhibiting a solution-dependent bound. Finally, we present polynomial-time algorithms for well-motivated special cases.

We begin by introducing some notation. We denote \(U = \{u_i\}\) as the union of end-nodes, termed target nodes, of the target links in \(H\). Assume \(|U| = n\). Let \(W = \{w_1, w_2, \ldots, w_m\}\) be the set of common neighbors of the target nodes, where each \(w_i \in W\) connects to at least two nodes in \(U\). Let \(N(u_i, u_j)\) denote the set of common neighbors of \(u_i\) and \(u_j\). For any node \(u_i \in V\), let \(d(u_i)\) be its degree. We use a decision matrix \(X \in \{0, 1\}^{m \times n}\) to denote the states of edges among the nodes in \(W\) and \(U\), where the entry \(x_{ij}\) in the \(i\)-th row and \(j\)-th column of \(X\) equals 1 if there is an edge between \(w_i\) and \(u_j\); otherwise, \(x_{ij} = 0\). We will say the attacker erases \(x_{ij}\) (when \(x_{ij} = 1\)) to denote the fact that the attacker deletes the edge between \(w_i\) and \(u_j\) (thus setting \(x_{ij}\) as 0).

3.1 Classification of Local Metrics

We now make a useful distinction between two classes of local metrics that use somewhat different local information.

**Definition 3.1.** A metric \(\text{Sim}\) is a CND metric if the corresponding total similarity \(f_2\) has the form \(\sum_{r=1}^{m} W_r \frac{\sum_{i,j \in H \setminus \{u_i, u_j\}} x_{ij}^r}{f_r(S_r)}\), where \(f_r\) is a metric-dependent increasing function of \(S_r\), the sum of \(r\)-th row of decision matrix \(X\), and \(W_r\) is an associated weight.

The metrics Adamic-Adar (AA), Resource Allocation (RA), and Common Neighbors (CN) are CND metrics. We note that the sum \(\sum_{i,j \in H \setminus \{u_i, u_j\}} x_{ij}^r\) is over all links in \(H\) for simplicity, we write the sum as \(\sum_{ij}\) henceforth.

**Definition 3.2.** A metric \(\text{Sim}\) is a WCN metric if

- it has the form \(\text{Sim}(u_i, u_j) = \frac{|N(u_i, u_j)|}{g(d(u_i), d(u_j), |N(u_i, u_j)|)}\), where \(g\) is strictly increasing in \(d(u_i)\) and \(d(u_j)\). That is, \(g(d(u_i) - t, d(u_j) - s) \leq g(d(u_i), d(u_j))\) for any valid non-negative integers \(t\) and \(s\) and any valid value of \(|N(u_i, u_j)|\).
- \(\text{Sim}\) is strictly increasing in \(|N(u_i, u_j)|\). That is, \(\text{Sim}(|N(u_i, u_j)| - t) \leq \text{Sim}(|N(u_i, u_j)|)\) for any valid non-negative integer \(t\) and any valid values of \(d(u_i)\) and \(d(u_j)\).

The WCN metrics include many common metrics, such as Jaccard, Sorensen, Salton, Hub Promoted, Hub Depressed, and Leicht.
By the above definitions, we know a rational attacker will only delete edges between nodes in $W$ and nodes in $U$, since deleting other types of edges will either decrease $d(u_i)$ or $d(w_j)$, causing the similarity to increase. Thus, the total similarity $f_t$ is fully captured by the decision matrix $X$. As a result, attacking local similarities is formulated as an optimization problem, termed as $\text{Prob-Local}$:

$$\min_X f_t(X), \quad \text{s.t.} \ \text{Sum}(X^0 - X) \leq k,$$

where $X^0$ is the original decision matrix and $\text{Sum}(\cdot)$ denotes the element-wise summation.

### 3.2 Hardness Results

We start by making no restrictions on the set of target links $H$. In this general case, we show that attacking all local metrics is NP-hard.

**Theorem 3.3.** Attacking local similarity metrics is NP-Hard.

**Proof.** As attacking local similarity metrics is modelled as an optimization problem, we consider the corresponding decision problem: can an attacker delete up to $k$ edges such that the total similarity $f_t$ is no greater than a constant $0$? We note that the minimum possible $f_t$ for all local metrics in a connected graph is $0$. Thus, we consider the decision problem $P_L$, which is to decide whether one can we delete $k$ edges such that $f_t = 0$.

We use the vertex cover problem for reduction. Let $P_{VC}$ denote the decision version of vertex cover, which is to decide whether there exists a vertex cover of size $k$ given a graph $G$ and an integer $k$.

Given an instance of vertex cover (i.e., a graph $G = (V, E)$ and an integer $k$), we construct our decision problem $P_L$ as follows. We first construct a new graph $Q$ in the following steps:

- For each node $v_i \in V$, create a node $v_i$ for graph $Q$.
- Add another node $w$ to $Q$ and connect $w$ to each $v_i$.
- Add $n = |V|$ nodes $u_1, \ldots, u_n$ and add an edge between each pair of nodes $(u_i, v_i)$.
- Add an edge between $(u_i, u_{i+1})$ for $i = 1, 2, \ldots, n - 1$.

The set $H$ of target links is then $H = \{(v_i, v_j)\} \subseteq Q$ if and only if $(v_i, v_j)$ is an edge in $G$. Our decision problem $P_L$ is then constructed regarding this graph $Q$ and target set $H$.

Now, we show $P_L$ and $P_{VC}$ are equivalent. We use CN metrics as an example and show that the same proof can be applied to other local metrics by slightly modifying the constructed graph $Q$.

First, we show if there is a vertex cover of size $k$ in graph $G$, then we can delete $k$ edges such that $f_t(H) = 0$. Suppose $V_c$ is a vertex cover with $|V_c| = k$. Without loss of generality, let $V_c = \{v_1, \ldots, v_k\}$. Then we show that deleting $k$ edges $(v_1, w), \ldots, (v_k, w)$ will make $f_t(H) = 0$. Let $(v_i, v_j)$ be an arbitrary target link. Then $(v_i, v_j)$ corresponds to an edge in $G$. By the definition of vertex cover, we have at least one of $v_i$ and $v_j$ is in $V_c$. We can assume $v_i \in V_c$. Since $v_i$ and $v_j$ have only one common neighbor $w$ in $Q$, deleting $(v_i, w)$ will make $CN(v_i, v_j) = 0$. As $(v_i, v_j)$ is arbitrarily selected, we have $CN(v_i, v_j) = 0$ for any target link $(v_i, v_j) \in H$. Thus, we have found $k$ edges whose deletion will make $f_t(H) = 0$.

Second, we show if we can delete $k$ edges to make $f_t(H) = 0$ in $Q$, the we can find a vertex cover of size $k$ in $G$. Suppose we found $k$ edges whose deletion will make $f_t(H) = 0$. Then each deleted edge must be $(v_i, v_j)$ for some $i = 1, \ldots, n$, since deleting other types of edges will not decrease $f_t(H)$. Without loss of generality, we assume the $k$ deleted edges are $(v_1, v_1), \ldots, (v_k, v_k)$. We then show that $V_c = \{v_1, \ldots, v_k\}$ forms a vertex cover in $G$. Since $\forall (v_i, v_j) \in H, CN(v_i, v_j) \geq 0$, $f_t(H) = 0$ means that $CN(v_i, v_j) = 0$ for very target link. As each target link $(v_i, v_j)$ initially has one common node $w$ we know at least one of $v_i$ and $v_j$ is in set $V_c$; otherwise, $CN(v_i, v_j) = 1$ making $f_t(H) > 0$. As each $(v_i, v_j)$ corresponds to an edge in $G$, we know each edge in $G$ has at least one end node in $V_c$. By definition, $V_c$ is a vertex cover of size $k$.

As a result, $P_L$ and $P_{VC}$ is equivalent, proving that minimizing CN metric is NP-Hard. The other local metrics are different variations of CN metrics. To make the above proof applicable for other metrics, we need to construct graph $Q$ such that $f_t(H) = 0$ if and only if there is no common neighbors between each pair of target link. To achieve this, we can slightly modify the graph $Q$ constructed previously for CN metric. For CND metrics, we can add some isolated nodes to $Q$ and connect $w$ with each of the isolated nodes. For WCN metrics, we can add some isolated nodes for each node $v_i$ and connect each isolated node with $v_i$ to make sure that the degree of each $v_i$ is always positive. Then the previous proof holds for other local metrics. $\square$

### 3.3 Practical Attacks

Since in general attacking even local metrics is hard, we have two ways of achieving positive results: approximation algorithms and restricted special cases. We start with the former, and exhibit several tractable special cases thereafter.

To obtain an approximation algorithm for the general case, we use submodular relaxation. Specifically, we bound the denominator of each term of $f_t$ by constants as if all the budget were assigned to decrease that single term, arriving at an upper bound $f_{tu}$ for the original objective $f_t$.

For WCN metrics, let $g_{ij}$ be the denominator of $\text{Sim}(u_i, u_j)$. For each $g_{ij}$, we bound it by $L_{ij} \leq g_{ij} \leq U_{ij}$, where $L_{ij}$ is obtained when $k$ edges are deleted and $U_{ij}$ is obtained when no edge is deleted. Take Sørensen metric as an example, where $\text{Sim}(u_i, u_j) = \frac{2 \cdot |N(u_i) \cap N(u_j)|}{d(u_i) + d(u_j)}$. Then $d_i^0 + d_j^0 - k \leq d(u_i) + d(u_j) \leq d_i^0 + d_j^0$, where $d_i^0$ and $d_j^0$ denote the original degrees of $u_i$ and $u_j$, respectively. In this way, each similarity is bounded as

$$\frac{|N(u_i) \cap N(u_j)|}{U_{ij}} \leq \frac{|N(u_i) \cap N(u_j)|}{L_{ij}} \\ \text{Let } f_{tu}^{\text{WCN}} = \sum_{i,j} \frac{|N(u_i) \cap N(u_j)|}{L_{ij}}, \text{ and } f_t^{\text{WCN}} = \sum_{i,j} \frac{|N(u_i) \cap N(u_j)|}{U_{ij}}. \text{ Then } f_t^{\text{WCN}} \leq f_t \leq f_{tu}^{\text{WCN}}.$$

Similarly, for CND metrics, the denominator in each term $f_t(S_r)$ is bounded by $f_t(S_r^0) - k \leq f_t(S_r) \leq f_t(S_r^0)$, where $S_r^0$ denotes the sum of the $r$th row of the original decision matrix $X^0$. Then $f_t^{\text{CND}} \leq f_t \leq f_{tu}^{\text{CND}}$, where $f_t^{\text{CND}} = \sum_{r=1}^{m} W_r \sum_{i,j} f_t(S_r)$. Due to the similarity between the structures of $f_t^{\text{WCN}}$ and $f_t^{\text{CND}}$, we will focus on $f_t^{\text{WCN}}$ and
omit the superscript \( WCN \) in the following analysis. The proposed approximation algorithm and the associated bound analysis are also applicable for \( f_{1CND} \).

Optimizing Bounding Function. We now consider minimizing \( f_{tu} \). Let \( S' \) be the set of edges that the attacker chooses to delete. Then set \( S' \) is associated with a decision matrix \( X' \). For any \( S \subset S' \), we have \( X \geq X' \), where \( X \) is the matrix associated with \( S \) and \( \geq \) denotes component-wise comparison. Define a set function \( F(S) = f_{tu}(X') - f_{tu}(X) \). Clearly, \( F(\emptyset) = 0 \). Then minimizing \( f_{tu} \) is equivalent to

\[
\max_{S \subseteq E_Q} F(S), \quad \text{s.t. } |S| \leq k. \tag{3}
\]

**Theorem 3.4**. \( F(S) \) is a monotone increasing submodular function.

**Proof.** Assume \( S \subset S' \), we need to show \( F(S) \leq F(S') \). It is equivalent to show \( f_{tu}(X') \geq f_{tu}(X) \). Let \( C_i \) be the \( i \)-th column of \( X \). Then \( \{N(u_i, u_j) = (C_i, C_j)\} \), where \( (C_i, C_j) \) denotes their inner product. Now, \( f_{tu}(X) = \sum_{ij} \frac{w_{ij}}{L_{ij}} (C_i C_j) \), where the weights \( w_{ij} \) and \( L_{ij} \) are constants. Since \( X \geq X' \), we have \( (C_i, C_j) \geq (C_i', C_j') \) for every pair of \( i, j \). Thus, \( f_{tu}(X) \geq f_{tu}(X') \). That is, \( F(S) \) is monotone increasing.

Let an edge \( e \notin S' \) be associated with the \( p \)-th row and \( q \)-th column entry in \( X \). Let \( e \cup S' \) be associated with a matrix \( X' \), where the only difference between \( X' \) and \( X \) is that \( x'_{ij} = 0 \) while \( x_{ij} = 1 \).

Similarly, let \( e \cup S' \) be associated with a matrix \( X' \). Define \( \Delta(e|S) = F(e \cup S) - F(S) \) and \( \Delta(e|S') = F(e \cup S') - F(S') \). Then we need to show \( \Delta(e|S) \geq \Delta(e|S') \).

\[
\Delta(e|S) = f_{tu}(X) - f_{tu}(X') = \sum_{ij} \frac{w_{ij}}{L_{ij}} (C_i C_q) - \sum_{ij} \frac{w_{ij}}{L_{ij}} (C_i' C_q') = \sum_{j} \frac{w_{jq}}{L_{jq}} x_{pq} - \sum_{j} \frac{w_{jq}}{L_{jq}} x'_{pq} = \sum_{j} \frac{w_{jq}}{L_{jq}} x_{pq},
\]

where the sum \( \sum_{j} \) is over all pairs of \( (i, j) \) such that \( (u_i, u_j) \in H \).

The second equality holds as deleting edge \( e \) will only affect the \( q \)-th column. The last equality holds since \( x_{pq} = 1 \) and \( x'_{pq} = 0 \).

Similarly, we can obtain \( \Delta(e|S') = \sum_{j} \frac{w_{jq}}{L_{jq}} x_{pq} \). Then \( \Delta(e|S) \geq \Delta(e|S') \). Since \( \sum_{j} \frac{w_{jq}}{L_{jq}} x_{pq} \geq 0 \), we have \( \Delta(e|S) \geq \Delta(e|S') \). By definition, \( F(S) \) is submodular.

\( \Box \)

Problem (3) is to maximize a monotone increasing submodular function under cardinality constraint. The typical greedy algorithm for such type of problems achieves a \((1 - 1/e)\)-approximation of the maximum. In particular, the greedy algorithm will delete the edge that will cause the largest increase in \( F(S) \) step by step until \( k \) edges are deleted. Suppose the greedy algorithm outputs a sub-optimal set \( S' \), which corresponds to a minimizer \( X_u' \) of \( f_{tu}(X) \). Then we take the value \( f_t(X_u') \) as the approximation of \( f_t(X') \), where \( X' \) is the optimal minimizer of \( f_t \). We term this approximation algorithm as \textit{Approx-Local}.

\textbf{Bound Analysis.} We theoretically analyze the performance of our proposed approximation algorithm \textit{Approx-Local}.\(^1\) Let \( X^*, X_u^* \), and \( X_t^* \) be the minimizers of \( f_t, f_{tu} \), and \( f_{lt} \), respectively. Define the gap between \( f_t \) and \( f_{tu} \) as \( \alpha(X) = f_{tu}(X) - f_t(X) \), which is a function of the decision matrix \( X \).

**Theorem 3.5.** \( \alpha(X) \) is an increasing function of \( X \).

**Proof.** Consider a particular term of \( \alpha(X) \), which is denoted as \( \alpha_{ij}(X) = \frac{w_{ij}}{L_{ij}} (C_i^X C_j^X) - \frac{w_{ij}}{L_{ij}} (C_i^Y C_j^Y) \). Let \( C_i^X \) denote the \( i \)-th column of \( X \). For simplicity, write \( g(d(u_i, d(u_j), (C_i^Y, C_j^Y)) \). Let \( \Delta X \) be the minimizers of \( f_{tu}(X) \).

Let \(\Delta_{ij}(X) = \frac{w_{ij}}{L_{ij}} (C_i^X C_j^X) - \frac{w_{ij}}{L_{ij}} (C_i^Y C_j^Y) \).

Then \( \Delta_{ij}(X) = \alpha_{ij}(X) \). As \( g \) is strictly increasing in \( d(u_i) \) and \( d(u_j) \), it is increasing in \( X \). Then we have \( g(X) \geq g(Y) \). Thus,

\[
\alpha_{ij}(X) - \alpha_{ij}(Y) \geq \frac{w_{ij}}{L_{ij}} (\frac{(C_i^X - C_i^Y) - (C_j^X - C_j^Y)}{g(Y)}) \geq 0.
\]

The last inequality holds as \( L_{ij} \) is the lower bound (i.e., \( L_{ij} \leq g(Y) \)). As \( \alpha(X) \) is the weighted sum over all pair of target links, we have \( \alpha(X) \geq \alpha(Y) \).

\( \Box \)

Theorem 3.5 states that the gap between the total similarity and its upper bound function is closing as we delete more edges (i.e., \( X \) becomes smaller). We further provide a solution-dependent bound of \( g = f_{tu}(X_u^*) - f_t(X^*) \), which measures the gap between the minimum of \( f_t \) output by our proposed algorithm and the real minimum.

\[
g \leq f_{tu}(X_u^*) - f_t(X^*) \leq f_{tu}(X_u^*) - f_{lt}(X_t^*) \leq f_{tu}(X_u^*) - f_{lt}(X_t^*) \leq 0.
\]

Such a gap depends on the solutions \( X_u^* \) and \( X_t^* \). We evaluate the gap through extensive experiments in Section 5.

\subsection{3.4 Tractable Special Cases}

We identify two important special cases for which the attack models are significantly simplified. The first case considers attacking a single target link and optimal attacks can be found in linear time for all local metrics. The second case considers attacking a group of nodes and the goal is to hide all possible links among them. We demonstrate that optimal attacks in this case can be found efficiently for the class of CND metrics.

Due to the space limit, we only highlight some key observations and present some important results. The full analysis is in the extended version [27] of the paper.

\subsection{3.4.1 Attacking a Single Link}

When the target is a single link \((u, v)\), the attacker will focus only on the links connecting \( u \) or \( v \) with their common neighbors, denoted as \( N(u, v) = \{w_j\}_{j=1}^{N_{iu}} \). Let \( x_{iu} = 0 \) denotes that attacker chooses to delete the link between \( w_i \) and \( u \) or \( x_{iu} = 1 \) otherwise.
where \( g \) is a non-decreasing function of \( d(w_i) \).

To minimize a CND, the attacker will remove edges incident to common neighbors \( w \) in increasing order of degree \( d(w) \). In fact, this algorithm is optimal and has a time complexity \( O(|N(u, v)|) \).

For WCN metrics, consider a tuple \((u, w, v)\) where \( w \) is a common neighbor of \( u \) and \( v \). We divide the links surrounding \((u, w, v)\) into four sets: \( E_1 = \{(u, w)\}, E_2 = \{(v, w)\}, E_3 = \{(u, v)\}, \) and \( E_4 = \{(v, s)\}, \) where \( s \) denotes a non-common neighbor of \( u \) and \( v \). As the attacker deletes links from \( E_0 \), there are four possible states of the tuples between \( u \) and \( v \). In state 1, both \((u, w)\) and \((w, v)\) are deleted. In state 2, \((u, w)\) is deleted while \((w, v)\) is not. In state 3, \((w, v)\) is deleted while \((u, w)\) is not. In state 4, neither \((u, w)\) nor \((w, v)\) is deleted. We use integer variables \( y_1, y_2, y_3 \) to denote the number of tuples in state 1, 2, 3, respectively. Furthermore, let \( y_4 \) and \( y_5 \) be the number of deleted edges from \( E_3 \) and \( E_4 \), respectively. In this way, the vector \((y_1, y_2, y_3, y_4, y_5)\) fully captures an attacker’s strategy.

Proposition 3.7. A WCN metric can be written as \( \text{Sim}(u, v) = f(y_1, y_2, y_3, y_4, y_5) \) such that \( f \) is decreasing in \( y_2 \) and \( y_3 \) and \( f \) is increasing in \( y_4 \) and \( y_5 \). Our analysis shows that in an optimal attack, \( y_1^* = y_2^* = y_3^* = 0 \) and \( y_4^* + y_5^* = k \). That is, the attacker will always choose \( k \) edges from \( E_1 \cup E_2 \) to delete. The following theorem then specifies how the attacker can optimally choose edges.

Theorem 3.8. The optimal attack on WCN metrics with a single target link selects arbitrary \( y_1^* \) links from \( E_1 \) and \((k - y_1^*) \) links from \( E_2 \) to delete with the constraint that for any selected links \((u, w_1) \in E_1 \) and \((v, w_2) \in E_2, w_1 \neq w_2 \). The value of \( y_1^* \) is the solution of a single-variable integer optimization problem.

The time complexity of solving the single-variable integer optimization problem is bounded in \( O(k) \).

3.4.2 Attacking A Group of Nodes: We consider the special case where 1) the target is a group of nodes \( U \) and the links between each pair of nodes in \( U \) consist the target link set \( H \); 2) each link in \( H \) has equal weight. In this case, optimal attacks on CND metrics can be found in polynomial time.

Proposition 3.9. For CND metrics, the total similarity \( f_i \) has the form \( \sum_{i=1}^{M} f_i(S_i) \), where \( S_i \) is the sum of the \( i \)-th row of \( X \) and \( f_i(S_i) \) is a convex increasing function of \( S_i \).

Proposition 3.9 states that \( f_i \) for CND metrics can be written as a sum of independent functions, where each function \( f_i \) is a convex increasing function. We then propose a greedy algorithm termed Greedy-CND to minimize \( f_i^{\text{CND}} \). In essence, Greedy-CND takes as the input \( S^0 \), which is the row sum of the initial decision matrix \( X \), and decreases an entry in \( S^0 \) whose decreasing causes the maximum decrease in \( f_i^{\text{CND}} \) step by step until an upper bound of \( k \) edges are deleted. This algorithm turns out to be optimal, as we prove in the extended version of the paper.

4 ATTACKING GLOBAL METRICS

In this section, we analyze attacks on two common global similarity metrics: Katz and ACT. We begin with attacks on a single link and show that finding optimal attack strategies is NP-hard even for a single target link.

Let \( A \in \{0, 1\}^{N \times N} \) and \( D \) be the adjacency matrix and degree matrix of the graph \( G \), respectively. The Laplacian matrix is defined as \( L = D - A \). The pseudo-inverse of \( L \) is \( L^+ = (L - E)^{-1} + E \), where \( E \) is an \((N \times N)\) matrix with each entry being \( \frac{1}{N} \). We use a binary vector \( y \in \{0, 1\}^M \) to denote the states of edges in \( E_Q \), where \( y_i = 0 \) iff the \( i \)-th edge in \( E_Q \) is deleted. Finally, \( y \leq y' \) \((A \leq A')\) is a component-wise inequality between vectors (matrices).

4.1 Problem Formulation for Katz Similarity

The Katz similarity is a common path-based similarity metric \([8]\).

For a pair of nodes \((u, v)\), Katz similarity is defined as

\[
\text{Katz}(u, v) = \sum_{l=1}^{\infty} \beta^l |\text{path}_{u,v}^l| = (\beta A + \beta^2 A^2 + \beta^3 A^3 + \cdots)_{uv},
\]

where \( |\text{path}_{u,v}^l| \) denotes the number of paths of length \( l \) between \( u \) and \( v \). \( \beta > 0 \) is a parameter and \((\cdot)_{uv} \) denotes the entry in the \( u \)-th row and \( v \)-th column of a matrix. By definition, the adjacency matrix \( A \) is fully captured by the vector \( y \). Thus, \( \text{Katz}(u, v) \) is a function of \( y \), written as \( \text{Katz}_{uv}(y) \). As one would expect, it is an increasing function of \( y \).

Lemma 4.1. \( \text{Katz}_{uv}(y) \) is an increasing function of \( y \).

Proof. Let \( A \) and \( A' \) be the corresponding adjacency matrices of \( y \) and \( y' \). If \( y \leq y' \), we have \( A \leq A' \). Now, consider the \( j \)-th term of the Katz similarity matrix \( K \), which is \( \beta^j A^j \). As every entry in \( A \) is non-negative and \( \beta > 0 \), we have \( \beta^j A^j \leq \beta^j A'^j \) for every \( j \). Thus, \( \text{Katz}_{uv}(y) \leq \text{Katz}_{uv}(y') \).

As a result, deleting a link will always decrease \( \text{Katz}_{uv}(y) \), and the attacker would therefore always delete \( k \) links in \( E_Q \) if \( E_Q \) has at least \( k \) links. Thus, minimizing Katz for a particular target link \((u, v)\) is captured by Prob-Katz:

\[
\min_y \text{Katz}_{uv}(y), \quad \text{s.t.} \quad \sum_{i=1}^{M} y_i = M - k, y \in \{0, 1\}^M.
\]

4.2 Problem Formulation for ACT

The second global similarity metric we consider is based on ACT, which measures a distance between two nodes in terms of random walks. Specifically, for a pair of nodes \((u, v)\), \( \text{ACT}(u, v) \), is the expected time for a simple random walker to travel from a node \( u \) to node \( v \) on a graph and return to \( u \). Since \( \text{ACT}(u, v) \) is a distance metric, the attacker’s aim is to maximize \( \text{ACT}(u, v) \), defined as

\[
\text{ACT}(u, v) = V_G(L^u_{uu} + L^v_{uv} - 2L^u_{uv}),
\]

where \( V_G \) is the volume of the graph \([4]\).

Directly optimizing \( \text{ACT}(u, v) \) is hard. Indeed, deleting an edge may either increase or decrease \( \text{ACT}(u, v) \), so that unlike other metrics, \( \text{ACT} \) is not monotone in \( y \). Fortunately, Ghosh et al. \([5]\) show that when edges are unweighted (as in our setting), \( \text{ACT}(u, v) \) can be defined in terms of Effective Resistance \( \text{ER} \): \( \text{ACT}(u, v) = V_G \text{ER}(u, v) \). It is also not difficult to see that both the volume \( V_G \) and \( \text{ER} \) can be represented in terms of \( y \).

We begin by investigating the effect of deleting an edge on \( \text{ER}(y) \). We use a well-known result by Doyle and Snell \([3]\) to this end.
Lemma 4.2 ([3]). The effective resistance between two nodes is strictly increasing when an edge is deleted.

The following lemma is then an immediate corollary.

Lemma 4.3. ER(y) is a decreasing function of y.

As a result, maximizing ER(y) would always entail deleting all allowed edges. Let t be the maximum number of edges that can be deleted. Then, maximizing ER(y) can be formulated as Prob-ER:

\[
\max_y \text{ ER}(y), \quad \text{s.t. } \sum_{i=1}^M y_i = M - t, y \in \{0,1\}^M.
\]

However, while ER(y) increases as we delete edges, volume \( V_G = 2 \sum_{i=1}^M y_i \) decreases. Fortunately, since volume is linear in the number of deleted edges, we reduce the problem of optimizing ACT to that of solving Prob-ER by solving the latter for \( t = 0, \ldots, k \), and choosing the best of these in terms of ACT. Similarly, hardness of Prob-ER implies hardness of optimizing ACT. Consequently, the rest of this section focuses on solving Prob-ER.

4.3 Hardness Results

We prove that minimizing Katz and maximizing ER between a single pair of nodes by deleting edges with budget constraint are both NP-hard.

Theorem 4.4. Minimizing Katz similarity and maximizing ACT distance is NP-hard even if \( H \) contains a single target link.

Proof. We consider the decision version of minimizing Katz, termed as Prob-\( P_K \), which is to decide whether one can delete \( k \) edges to make \( \text{Katz}(u,v) \leq q \) given a graph \( Q \) and a target node pair \( (u,v) \) in \( Q \). Similarly, we consider the decision version of maximizing ER, termed as Prob-\( P_E \): which is to decide whether one can delete \( k \) edges to make \( \text{ER}(u,v) \geq q \) given a graph \( Q \) and a target node pair \( (u,v) \) in \( Q \).

We use the Hamiltonian cycle problem, termed Prob-\( P_H \), for reduction. Prob-\( P_H \) is to decide whether there exists a circle that visits each nodes in a given connected graph \( G \) exactly once (thus called Hamiltonian circle).

By the definition of Katz similarity, \( \text{Katz}(u,v) \) is minimized when the graph is a string with \( u \) and \( v \) as two end nodes and all others as inner nodes in that string; that is the graph over that set of nodes is a Hamiltonian path with \( u \) and \( v \) as end nodes. We denote the minimum value of \( \text{Katz}(u,v) \) in this case as \( \text{min}_K \). Similarly, by the definition of effective resistance, \( \text{ER}(u,v) \) is maximized when the graph is also a Hamiltonian path over that set of nodes with \( u \) and \( v \) as the two end nodes. We assume that all edges have equal resistance. We denote the maximum value of \( \text{ER}(u,v) \) in this case as \( \text{max}_E \).

We then set \( q = \text{min}_K \) in the decision problem Prob-\( P_E \) and set \( q = \text{max}_E \) in Prob-\( P_K \). As a result, the two decision problems Prob-\( P_E \) and Prob-\( P_K \) are then both equivalent to the following decision problem, termed Prob-\( P_S \): given a graph \( Q \) and two nodes \( u \) and \( v \) in \( Q \), can we delete \( k \) edges such that the remaining graph \( S \) forms a string (i.e., a Hamiltonian path) with \( u \) and \( v \) as two end nodes?

Now the reduction. Given an instance of Hamiltonian circle (i.e., a graph \( G = (V,E) \)), we construct a new graph \( Q \) from \( G \) in the following steps:

- Select an arbitrary node \( w \) in \( G \). Let \( N(w) = \{l_1, l_2, \ldots, l_W \} \) be the neighbors of \( w \), where \( W = |N(w)| \).
- Add two nodes \( u \) and \( v \).
- Add edge \( (u, w) \) and edges \( (v, l_i), \forall l_i \in N(w) \).

The resulting graph is then the graph \( Q \) in decision problem Prob-\( P_S \), where the budget \( k = W + |E| - |V| \). We now show that problem Prob-\( P_H \) and problem Prob-\( P_S \) are equivalent.

First, we show if there exists a Hamiltonian circle in \( G \), then we can delete \( k = W + |E| - |V| \) edges such that the measurement (Katz or ER) between \( u \) and \( v \) in graph \( Q \) is \( q \). Assume the Hamiltonian circle travels to \( w \) through edge \( (l_i, w) \) and leaves \( w \) through edge \( (w, l_j) \). We then 1) delete \((W-1)\) edges \((v, l_i)\) for each \( l_i \in N(W) \) and \( l_i \neq l_j \); 2) delete all \(|E| - |V|\) edges in \( G \) that do not appear in the Hamiltonian circle; 3) delete edge \((w, l_j)\). Thus, we deleted a total of \(|E| - |V|\) edges. After deleting all these \( k \) edges, in the remaining graph \( S \), there exists a Hamiltonian path between \( w \) and \( l_i \). As \( u \) only connects to \( w \) and \( v \) only connects to \( l_i \), the remaining graph forms a Hamiltonian path between \( u \) and \( v \). As a result, the measurement between \( u \) and \( v \) equals \( q \).

Second, we show if we can remove \( k = W + |E| - |V| \) edges from \( Q \) such that the remaining graph \( S \) forms a Hamiltonian path between \( u \) and \( v \), then we can find a Hamiltonian circle in the graph \( G \). Suppose in the reaming string, \( v \) connects to \( l_i \) and \( w \) connects to \( l_j \). As \( u \) connects only to \( w \) in graph \( Q \), \( u \) must connect to \( w \) in \( S \). From the construction of \( Q \), the total number of edges of \( Q \) is \(|E| + W + 1\). After deleting \( k \) edges, the remaining number of edges is \(|V| + 1\). Excluding the two edges \((u, w)\) and \((v, l_i)\), we know there are \(|V| - 1\) edges among the node set \( V \) of the original graph \( G \). As the remaining graph is connected, there must exist a Hamiltonian path between \( w \) and \( l_i \). As \((w, l_i)\) is an edge in the graph \( G \), we have found a Hamiltonian circle in \( G \), consisting of the Hamiltonian path between \( w \) and \( l_i \) plus the edge \((w, l_i)\).

Thus, decision problem Prob-\( P_S \) is NP-complete; minimizing Katz and maximizing ER (ACT) are NP-hard. ☐

4.4 Practical Attack Strategies

While computing an optimal attack on Katz and ACT is NP-Hard, we now devise approximate approaches which are highly effective in practice.

4.4.1 Attacking Katz Similarity. To attack Katz similarity, we transform the attacker’s optimization problem into that of maximizing a monotone increasing submodular function. We begin with the single-link case (i.e., \( H \) is a singleton), and subsequently generalize to an arbitrary \( H \). We define a set function \( f(S_p) \) as follows. Let \( S_p \subseteq E_Q \) be a set of edges that an attacker chooses to delete. Let \( A_p \) be the adjacency matrix of the graph \( G_Q \) after all the edges in \( S_p \) are deleted. Define

\[
f(S_p) = \beta A_p + \beta^2 A_p^2 + \beta^3 A_p^3 + \cdots
\]

Since there is a one-to-one mapping between the set \( S_p \) and the matrix \( A_p \), the function \( f(S_p) \) is well-defined. We note that \( f(S_p) \) gives the Katz similarity matrix of the graph \( G \) after all the edges in \( S_p \) are deleted. We further define a set function

\[
g_{uv}(S_p) = (K - f(S_p))_{uv},
\]
where \( K = f(\emptyset) \) (the Katz similarity matrix when no edges are deleted) and \((\cdot)_{uv}\) denotes the \(u\)th row and \(v\)th column of a matrix.

Clearly, when \( S_p = \emptyset, g_{uv}(S_p) = 0 \).

Then, Prob-Katz is equivalent to

\[
\max_{S_p \subseteq E_I} g_{uv}(S_p), \quad \text{s.t. } |S_p| = k
\]  

(4)

**Theorem 4.5.** The set function \( g_{uv}(S_p) \) is monotone increasing and submodular.

Proof. To prove that \( g_{uv} \) is monotone increasing, we need to show that \( \forall S_p, S_q \subseteq \emptyset, g_{uv}(S_p) \leq g_{uv}(S_q) \). It is equivalent to show \( (f(S_p))_{uv} \leq (f(S_q))_{uv} \). We note that \((f(S_p))_{uv} \) and \((f(S_q))_{uv} \) are the Katz similarity between \( u \) and \( v \) after the edges in \( S_p \) and \( S_q \) are deleted, respectively. Theorem 4.1 states that the Katz similarity will decrease as more edges are deleted. Since \( S_p \subseteq S_q \), we have \( f(S_p) \geq f(S_q) \). Thus, \( g_{uv}(S_p) \leq g_{uv}(S_q) \).

Next, we prove \( g_{uv} \) is submodular. Let \( e \in E_I \setminus S_q \) be an edge between node \( i \) and node \( j \) in the graph. Let \( G \) be an \( n \times n \) matrix where \( G_{ij} = G_{ji} = 1 \) and the rest of the entries are 0. Then we have the set \( S_p \triangleq \emptyset \) is associated by \( G - S_p \) and \( S_q \subseteq \emptyset \) is associated with \( Y_q \). For a set \( S \), let \( \Delta(e|S) = f(S \cup e) - f(S) \). Then we need to show

\[
\Delta(e|S) \leq \Delta(e|S_q).
\]

Denote the \( t \)th item of \( \Delta(e|S) \) as \( \Delta^t(e|S) \). In the following, we will first prove \( \Delta^t(e|S_p) \leq \Delta^t(e|S_q) \) by induction. Assume that the holds for \( t = s \) (it’s straightforward to verify the case for \( t = 1 \) and \( t = 2 \)). That is

\[
\\beta^t[(A_p - G)^s - (A_p)^s] \leq 0.
\]

(5)

When \( t = s + 1 \), we have

\[
\begin{align*}
\Delta^t(e|S_p) & = (A_p - G)^{s+1} - (A_p)^{s+1} - (A_q - G)^{s+1} + (A_q)^{s+1} \\
& = (A_p - G)^sA_p - (A_p)^sA_q + (A_q)^sA_q \\
& \leq (A_p - G)\beta^s(A_p - G)^sA_q + (A_q)^sA_q
\end{align*}
\]

The inequality comes from the fact that \( (A_p - G) \geq (A_q - G) \) when \( G \geq 0 \). Furthermore, since \( S_p \subset S_q \), we have \( A_p - A_q = F \) for some \( F \geq 0 \). Thus, \( \Delta(e|S_p) \leq \Delta(e|S_q) \).

Next, for the multi-link case, the total similarity \( f_t = \sum_{i,j} w_{ij}K_{ij} \). Let \( F(S) \) be a function of the set of deleted edges, defined as

\[
F(S) = \beta A_S + \beta^2 A^2_S + \beta^3 A^3_S + \cdots,
\]

where \( A_S \) denotes the adjacency matrix after all edges in \( S \) are deleted. Note that \( F(S) \) gives the Katz similarity matrix when edges in \( S \) are deleted. Further define \( g_{ij}(S) = (K^0 - F(S))_{ij} \), where \( K^0 \) is the original Katz similarity matrix. Let \( G_t(S) = \sum_{i,j} w_{ij}g_{ij}(S) \). By definition, we have \( G_t(S) = \sum_{i,j} w_{ij}K^0_{ij} - f_t \).

Thus, minimizing \( f_t \) is equivalent to

\[
\max_{S \subseteq E_Q} G_t(S), \quad \text{s.t. } |S| \leq k.
\]

The following result is then a direct corollary of Theorem 4.5.

**Corollary 4.6.** \( G_t(S) \) is monotone increasing and submodular.

Proof. This is an immediate conclusion of two results. First, \( g_{ij}(S) \) is monotone increasing and submodular in \( S \) as proved in Theorem 4.5. Second, a positive linear combination of submodular functions is submodular \[16\]. As \( G_t(S) \) is the sum of \( g_{ij}(S) \), \( G_t(S) \) is monotone increasing and submodular. \( \square \)

As a result, minimizing the total Katz similarity is equivalent to maximizing a monotone increasing submodular function under cardinality constraint. We can achieve a \((1 - 1/e)\) approximation by applying a simple iterative greedy algorithm in which we delete one edge at a time that maximizes the marginal impact on the objective. We call this resulting algorithm Greedy-Katz.

**4.4.2 Attacking ACT.** From the analysis of minimizing Katz similarity, it is natural to investigate submodularity of the effective resistance or ACT as a function of the set of edges. Unfortunately, counter examples show that the effective resistance is neither submodular nor supermodular.

Our first step is to approximate the objective function \( ER(u, v) \) based on the results by Von Luxburg et al. \[17\], who show that \( ER(u, v) \) can be approximated by \( \frac{1}{d(u)} + \frac{1}{d(v)} \) for large geometric graphs as well as random graphs with given expected degrees. Consequently, we use the approximation \( ER(u, v) \approx ER_{ap}(u, v) = \frac{1}{d(u)} + \frac{1}{d(v)} \). Then the total effective resistance is approximated as

\[
ER(H) \approx ER_{ap}(H) = \sum_{ij} w_{ij}(\frac{1}{d(u)} + \frac{1}{d(v)}) = \sum_{i=1}^n W_i \sum_{j=1}^n w_{ij},
\]

where \( W_i > 0 \) is some constant weight associated with each \( u_i \). Let \( D_i \) be the original degree of node \( u_i \) and \( z_i \) be an integer variable denoting the number of deleted edges connecting to \( u_i \). Then maximizing \( ER_{ap}(H) \) is equivalent to

\[
\max \sum_{i=1}^n \frac{W_i}{D_i} - z_i, \quad \text{s.t. } \sum_{i=1}^n z_i \leq k, z_i \in [0, k].
\]

(6)

We assume that deleting edges would not make the graph disconnected. That is \( \forall i \in [1, n], k < D_i \).

We formulate the above problem as a linear integer program. Specifically, let \( \Delta_{ij} \) be the decrease in \( ER_{ap}(H) \) after \( j \) edges connecting to node \( u_i \) are deleted. As any such \( j \) edges will cause the same decrease, the value of each \( \Delta_{ij} \) for \( j = 0, 1, \ldots, k \) could be efficiently computed in advance. We use a binary variable \( h_{ij} = 1 \) to denote that the attacker chooses to delete such \( j \) edges; otherwise, \( h_{ij} = 0 \). Then problem (6) is equivalent to

\[
\max \sum_{h=0}^k \sum_{i,j=0}^n \frac{W_{ij}}{D_{ij}} - \Delta_{ij} h_{ij}, \quad \text{s.t. } \sum_{i=1}^n \sum_{j=0}^k h_{ij} \leq k, \forall i, \sum_{j=0}^k h_{ij} \leq 1.
\]
The above problem is a linear program of \((k + 1) \times n\) binary variables with \((n + 1)\) linear constraints. A numerical solution [6] gives the number of edges incident to each node that needs to be deleted.

5 EXPERIMENTS

Our experiments use two classes of networks: 1) randomly generated scale-free networks and 2) a Facebook friendship network [10]. In the scale-free networks, the degree distribution satisfies \(P(k) \propto k^{-\gamma}\), where \(\gamma\) is a parameter.

Baseline algorithms. We compare our algorithms with two baseline algorithms. We term the first one as RandomDel, which randomly deletes the edges connected to the target nodes. The second baseline, termed GreedyBase, is a heuristic algorithm proposed in [23]. This algorithm will try to delete the link whose deletion will cause the largest decrease in the number of “closed triads” as defined in [23]. Our experiments show that while the performance of GreedyBase varies regarding different metrics, RandomDel performs poorly for all metrics (Fig. 1). Henceforth, we only compare our algorithm with GreedyBase for global metrics (Fig. 2 and Fig. 3).

For local metrics, we evaluate Approx-Local in the general case. We consider a target set of size 20. We select RA (CND metric) and Sørensen (WCN metric) as two representatives, for which the results are presented in Fig. 1. All similarity scores are scaled to 1.0 when no edges are deleted. Due to space limit, we only present the results on one scale of the scale-free network \((n = 1000, \gamma = 2.0)\) and Facebook network \((n = 786, m = 12291)\). A more comprehensive set of experiments is presented in the extended version.

We note that deleting a relatively small number of links can significantly decrease the similarities of a set of target links. The gap between the upper and lower bound functions, which reflects the approximation quality of Approx-Local, is within 20% of the original similarity.

For global metrics, we evaluate Greedy-Katz and Local-ACT regarding a set of target links \(|H| = 20\) on different scales of networks. As shown in Fig. 2 and Fig. 3, the performances are significantly better than those of the baseline algorithm. Additional results for the special cases are provided in the extended version.

6 CONCLUSION

We investigate the problem of hiding a set of target links in a network via minimizing the similarities of those links, by deleting a limited number of edges. We divide similarity metrics associated with potential links into two broad classes: local metrics (CND and WCN) and global metrics (Katz and ACT). We prove that computing optimal attacks on all these metrics is NP-hard.

For local metrics, we proposed an algorithm minimizing the upper bounds of local metrics, which corresponds to maximizing submodular functions under cardinality constraints. Furthermore, we identify two special cases, attacking a single link and attacking a group of nodes, where the first case ensures optimal attacks for all local metrics and the latter ensures optimal attacks for CND metrics. For global metrics, we prove that even when attacking a single link, both the problem of minimizing Katz and that of maximizing ACT are NP-Hard. We then propose an efficient greedy algorithm (Greedy-Katz) and a principled heuristic algorithm (Local-ACT) for the two problems, respectively. Our experiments show that our algorithms are highly effective in practice and, in particular, significantly outperform a recently proposed heuristic. Overall, the results in this paper greatly advance the algorithmic understanding of attacking similarity-based link prediction.

ACKNOWLEDGMENT

This research was partially supported by the National Science Foundation (IIS-1905558) and Army Research Office (W911NF-16-1-0069 and MURI W911NF-18-1-0208). Tomasz P. Michalak was supported by the Polish National Science Centre grant 2016/23/B/ST6/03599.
REFERENCES


