A Maximum Variance Approach for Graph Anonymization

Hiep H. Nguyen, Abdessamad Imine, Michael Rusinowitch

A Maximum Variance Approach for Graph Anonymization

Hiep H. Nguyen[1], Abdessamad Imine[1], and Michaël Rusinowitch[1]

LORIA/INRIA Nancy-Grand Est
France
{huu-hiep.nguyen,michael.rusinowitch}@inria.fr,abdessamad.imine@loria.fr

Abstract. Uncertain graphs, a form of uncertain data, have recently attracted a lot of attention as they can represent inherent uncertainty in collected data. The uncertain graphs pose challenges to conventional data processing techniques and open new research directions. Going in the reserve direction, this paper focuses on the problem of anonymizing a deterministic graph by converting it into an uncertain form. The paper first analyzes drawbacks in a recent uncertainty-based anonymization scheme and then proposes *Maximum Variance*, a novel approach that provides better tradeoff between privacy and utility. Towards a fair comparison between the anonymization schemes on graphs, the second contribution of this paper is to describe a quantifying framework for graph anonymization by assessing privacy and utility scores of typical schemes in a unified space. The extensive experiments show the effectiveness and efficiency of Maximum Variance on three large real graphs.

1 Introduction

Graphs represent a rich class of data observed in daily life where entities are represented by vertices and their connections are characterized by edges. With the appearance of increasingly complex networks, the research community requires large and reliable graph data to conduct in-depth studies. However, this requirement usually conflicts with privacy protection of data contributing entities. Specifically in social networks, naive approaches like removing user ids from social graphs are not effective, leaving users open to privacy risks. Structural attacks to re-identify or de-anonymize users are shown feasible [1] [10]. Recent surveys on security and privacy issues in OSNs (e.g. [9]) enumerate real-world breaches and possible defenses. Anonymization is such an effective countermeasure with many schemes proposed recently [25, 12, 26, 4, 22, 20, 6, 5].

Given a social graph, the existing anonymization methods fall into four main categories. The first category includes *random* additions, deletions and switches of edges to prevent the re-identification of nodes or edges. The methods of second category provide k-anonymity [19] by *deterministic* node/edge additions or deletions, assuming attacker's background knowledge regarding some property of its target node. The methods falling in the third category assign probabilities to edges to add uncertainty to the true graph. Finally, the fourth class of techniques cluster nodes into super nodes of size at least k. Note that the last two classes of schemes induce *possible world* models, i.e., we can retrieve sample graphs that are consistent with the anonymized output graph.

The third category is the most recent class of methods which leverage the semantics of edge probability to inject uncertainty to a given deterministic graph, converting it into an uncertain one. In [2], Boldi et al. introduced the concept of *(k,ϵ)-obfuscation*, where $k \geq 1$ is a desired level of obfuscation and $\epsilon \geq 0$ is a tolerance parameter. However, this approach exposes several shortcomings in the selection of potential edges and the formulation of minimizing the standard deviation of the sampling distribution. We clarify these points in Section 3.2.

In this paper, we introduce *Maximum Variance* (MV) approach based on two crucial observations. First, we observe that nodes gain better privacy if their incident uncertain edges constitute large degree variance. To avoid the trivial solution of all edges having probabilities 0.5 and to keep the expected node degrees for utility, we formulate a quadratic program with constraints that the expected node degrees should be as in the true graph. Second observation emerges naturally from the formation of real networks that display community structure where new links are largely formed by *transitivity.* Therefore, we propose adding potential edges only by distance 2 (*friend-of-friend*). The extensive experiments show the elegance and effectiveness of MV over (k, ϵ)-obfuscation. We believe that the present work suggests an extensible approach for graph anonymization. Our contributions are summarized as follows:

- We analyze several disadvantages in the previous work [2], showing that their pursuit of minimum standard deviation σ has high impact on privacy (Section 3).
- We propose MV, a novel anonymization scheme also based on the semantics of uncertain graphs (Sections 4 and 5). MV provides better privacy and utility by using two key observations. It proposes *nearby* potential edges and tries to maximize the *variance* of node degrees by a simple quadratic program.
- Towards a fair comparison between the anonymization schemes on graphs, this paper describes a generic quantifying framework by putting forward the distortion measure (Section 6). Rather than Shannon entropy-based or min entropy-based privacy scores with a parameter k as in previous work, the framework utilizes the *incorrectness* concept in [17] to quantify the re-identification risks of nodes. As for the utility score, we select typical graph metrics [2] [23].
- We evaluate the MV approach on three large real graphs and show its outperformance over (k, ϵ)-obfuscation (Section 7).

2 Related Work

2.1 Anonymization of Deterministic Graphs

There is a vast literature on graph perturbation that deserves a survey. In this section, we enumerate only several groups of ideas that are related to our proposed scheme.

Anonymizing unlabeled vertices for node privacy In unlabeled graphs, node identifiers are numbered in an arbitrary manner after removing their labels. The attacker aims at reidentifying nodes solely based on their structural information. For this line of graphs, node privacy protection implies the link privacy. Techniques of adding and removing edges, nodes can be done randomly or deterministically. Random perturbation is a naive approach and usually used as a baseline method. More guided approaches consist of *k-neighborhood*[25], *k-degree*[12, 6, 5], *k-automorphism*[26], *k-symmetry*[22], *k-isomorphism*[4] and k^2-*degree*[20]. These schemes provide k-anonymity [19] semantics and most of them rely on heuristics to avoid combinatorial intractability, except optimal solutions based on dynamic programming [12, 6, 5]. K-automorphism, k-symmetry, and k-isomorphism can resist *any* structural attacks by exploiting the inherent symmetry in graph. K-symmetry partitions a graph into automorphic orbits and duplicate subgraphs. k^2-degree addresses the friendship attacks, based on the vertex degree pair of an edge. Ying and Wu [23] propose a spectrum preserving approach which wisely chooses edge pairs to switch in order to keep the spectrum of the adjacency matrix not to vary too much. The clearest disadvantage of the above schemes is that they are inefficient, if not infeasible, on large graphs.

Apart from the two above categories, perturbation techniques have other categories that capitalize on *possible world* semantics. Hay et al. [10] generalize a network by clustering nodes and publish graph summarization of super nodes and super edges. The utility of this scheme is limited. In another direction, Boldi et al. [2] take the uncertain graph approach. With edge probabilities, the output graph can be used to generate sample graphs by independent edge sampling. Our approach belongs to this class of techniques with different formulation and better privacy/utility tradeoff. Note that in *k-symmetry*[4], the output sample graphs are also possible worlds of the intermediate symmetric graph.

Anonymizing labeled vertices for link privacy If nodes are labeled, we are only concerned about the link disclosure risk. For example, Mittal et al. [13] employ an edge rewiring method based on random walks to keep the mixing time tunable and prevent link re-identification by Bayesian inference. This method is effective for social network based systems, e.g. Sybil defense, DHT routing. Link privacy is also described in [23] but only for Random Switch, Random Add/Del.

Min entropy, Shannon entropy and incorrectness measure We now survey some commonly used notions of privacy metrics. *Min entropy* [18] quantifies the largest probability gap between the posterior and the prior over all items in the input dataset. K-anonymity has the same semantics with the corresponding min entropy of $\log_2 k$. So k-anonymity based perturbation schemes in the previous subsection belong to min entropy. Shannon entropy argued in [3] and [2] is another choice of privacy metrics. The third metrics that we use in MV is the *incorrectness* measure from location privacy research [17]. Given the prior information (e.g. node degree in the true graph) and the posterior information harvested from the anonymized data, incorrectness measure is the number of incorrect guesses made by the attacker. This measure gauges the *distortion* caused by the anonymization algorithm.

Table 1: List of notations

Symbol	**Definition**
$G_0 = (V, E_{G_0})$	true graph
$\mathcal{G} = (V, E, p)$	uncertain graph constructed from G_0
$G = (V, E_G) \sqsubseteq \mathcal{G}$	sample graph from $\mathcal{G}$
$d_u(G)$	degree of node u in G
$d_u(\mathcal{G})$	expected degree of node u in $\mathcal{G}$
n_p	number of potential edges, i.e. $\lvert E\rvert = \lvert E_{G_0}\rvert + n_p$
$\mathcal{N}(u)$	neighbors of node u in $\mathcal{G}$
$\Delta(d)$	number of d-degree nodes in G
R_σ	truncated normal distribution on [0,1]
$r_e \leftarrow R_\sigma$	a sample from the distribution R_σ
p_i (p_{uv})	probability of edge e_i (e_{uv})
$Hi_G(u)$	signature Hi of node u in graph G

2.2 Mining Uncertain Graphs

Uncertain graphs pose big challenges to traditional mining techniques. Because of the exponential number of possible worlds, naive enumerations are intractable. Typical graph search operations like k-Nearest neighbor and pattern matching require new approaches [15] [27] [24]. Those methods answer threshold-based queries by using pruning strategies based on Apriori property of frequent patterns.

3 Preliminaries

This section starts with common definitions and assumptions on uncertain graphs. It then reveals several shortcomings in the main competitor [2]. Table 1 summarizes notations used in this paper.

3.1 Uncertain Graph

Let $\mathcal{G} = (V, E, p)$ be an uncertain undirected graph, where $p : E \to [0, 1]$ is the function that gives an existence probability to each edge. The common assumption is on the *independence* of edge probabilities. Following the *possible-worlds* semantics in relational data [7], the uncertain graph $\mathcal{G}$ induces a set $\{G = (V, E_G)\}$ of $2^{|E|}$ deterministic graphs (worlds), each is defined by a subset of E. The probability of $G = (V, E_G) \sqsubseteq \mathcal{G}$ is:

$$Pr(G) = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e)) \quad (1)$$

Note that deterministic graphs are also uncertain graphs with all edges having probabilities 1.

3.2 (k, ϵ)-obfuscation and Its Limitations

In [2], Boldi et al. extend the concept of *k-obfuscation* developed in [3].

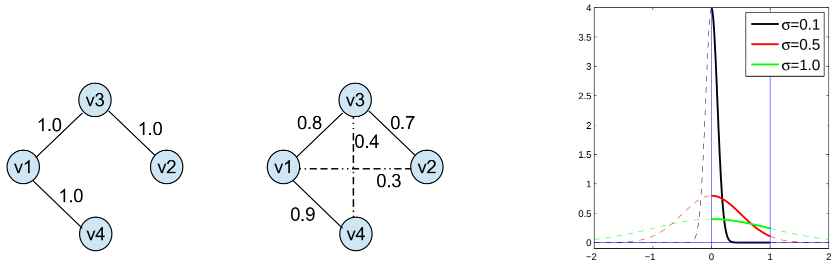
Definition 1. *(k,ϵ)-obfuscation [2]. Let P be a vertex property, $k \geq 1$ be a desired level of obfuscation, and $\epsilon \geq 0$ be a tolerance parameter. The uncertain graph $\mathcal{G}$ is said to k-obfuscate a given vertex $v \in G$ with respect to P if the entropy of the distribution $Y_{P(v)}$ over the vertices of $\mathcal{G}$ is greater than or equal to $\log_2 k$:*

$$H(Y_{P(v)}) \geq \log_2 k \tag{2}$$

The uncertain graph $\mathcal{G}$ is a (k, ϵ)-obfuscation with respect to property P if it k-obfuscates at least $(1 - \epsilon)n$ vertices in $\mathcal{G}$ with respect to P.

Given the true graph G_0 (Fig.1a), the basic idea of (k, ϵ)-*obf* (Fig.1b) is to transfer the probabilities from existing edges to potential (non-existing) edges. The edge probability is sampled from the truncated normal distribution R_σ (Fig. 1c). For each existing sampled edge e, it is assigned a probability $1 - r_e$ where $r_e \leftarrow R_\sigma$ and for each non-existing sampled edge e', it is assigned a probability $r_{e'} \leftarrow R_\sigma$.

Table 2 gives an example of how to compute degree entropy for the uncertain graph in Fig. 1b. Here vertex property P is the node degree. Each row in the left table is the degree distribution for the corresponding node. For instance, $v1$ has degree 0 with probability $(1 - 0.8).(1 - 0.3).(1 - 0.9) = 0.014$. The right table normalizes values in each column (i.e. in each degree value) to get distributions $Y_{P(v)}$. The entropy $H(Y_{P(v)})$ for each degree value is shown in the bottom row. Given $k = 3, \log_2 k = 1.585$, then $v1, v3$ with true degree 2 and $v2, v4$ with degree true 1 satisfy (2). Therefore, $\epsilon = 0$.



(a) (b) (c)

Fig. 1: (a) True graph (b) An obfuscation with potential edges (dashed) (c) Truncated normal distribution on [0,1] (bold solid curves)

While the idea is quite interesting as a guideline of how to come up with an uncertain version of the graph, the specific approach in [2] has two drawbacks. First, it formulates the problem as the minimization of σ. With small values of σ, r_e highly concentrates around zero, so existing sampled edges have probabilities nearly 1 and non-existing sampled edges are assigned probabilities almost 0. By the simple rounding technique, the attacker can easily reveal the true graph.

Table 2: The degree uncertainty for each node (left) and normalized values for each degree (right)

	deg=0	deg=1	deg=2	deg=3
v1	0.014	0.188	0.582	0.216
v2	0.210	0.580	0.210	0.000
v3	0.036	0.252	0.488	0.224
v4	0.060	0.580	0.360	0.000

$Y_{P(v)}$	deg=0	deg=1	deg=2	deg=3
v1	0.044	0.117	0.355	0.491
v2	0.656	0.362	0.128	0.000
v3	0.112	0.158	0.298	0.509
v4	0.187	0.362	0.220	0.000
$H(Y_{P(v)})$	1.404	1.844	1.911	0.999

Even if the graph owner only publishes sample graphs, re-identification attacks are still effective. As we show in Section 7, the $H2_{open}$ risk in the uncertain graph produced by [2] may be up to 50% of the true graph while it is only 2% in our approach. Also note that in [2], the found values of σ vary in a wide range from 10^{-1} to 10^{-8}. Second, the approach in [2] does not consider the locality (subgraph) of nodes in selecting pairs of nodes for establishing potential edges. As shown in [8], *subgraph-wise perturbation* effectively reduces structural distortion.

4 Maximum Variance Approach

In this section, we present two key observations underpinning the MV approach.

4.1 Observation #1: Maximum Degree Variance

We argue that efficient countermeasures against structural attacks should hinge on node degrees because the degree is the fundamental property of nodes in unlabeled graphs and if a node and its neighbors have degree changed, the re-identification risk is reduced significantly. Consequently, instead of replicating local structures as in k-anonymity based approaches [25, 12, 26, 4, 22, 20], we can deviate the attacks by changing node degrees *probabilistically.* For example, node $v1$ in Fig.1a has degree 2 with probability 1.0 whereas in Fig.1b, its degree gets four possible values $\{0, 1, 2, 3\}$ with probabilities $\{0.014, 0.188, 0.582, 0.216\}$ respectively. Generally, given edge probabilities incident to node u as $p_1, p_2, ..p_{d_u(\mathcal{G})}$, the degree of u is a sum of independent Bernoulli random variables, so its expected value is $\sum_{i=1}^{d_u(\mathcal{G})} p_i$ and its variance is $\sum_{i=1}^{d_u(\mathcal{G})} p_i(1 - p_i)$. If we naively target the maximum (local) degree variance without any constraints, the naive solution is at $p_i = 0.5$ for every incident edge i. However, such an assignment distorts graph structure severely and deteriorates the utility. Instead, we should use the constraint $\sum_{i=1}^{d_u(\mathcal{G})} p_i = d_u(G_0)$. Note that the *minimum variance* of an uncertain graph is 0 and corresponds to the case $\mathcal{G}$ has all edges being deterministic, e.g. when $\mathcal{G} = G_0$ and in edge-switching approaches. In the following section, we show an interesting result relating the *total* degree variance with graph edit distance.

4.2 Variance with edit distance

The *edit distance* between two deterministic graphs G, G' is defined as:

$$D(G, G') = |E_G \setminus E_{G'}| + |E_{G'} \setminus E_G| \tag{3}$$

A well-known result about the expected edit distance between the uncertain graph $\mathcal{G}$ and the deterministic graph $G \sqsubseteq \mathcal{G}$ is:

$$E[D(\mathcal{G}, G)] = \sum_{G' \sqsubseteq \mathcal{G}} Pr(G')D(G, G') = \sum_{e_i \in E_G} (1 - p_i) + \sum_{e_i \notin E_G} p_i \tag{4}$$

Correspondingly, the variance of edit distance is defined as

$$Var[D(\mathcal{G}, G)] = \sum_{G' \sqsubseteq \mathcal{G}} Pr(G')[D(G, G') - E[D(\mathcal{G}, G)]]^2 \tag{5}$$

We prove in the following Theorem that the variance of edit distance is the sum of edge variances and does not depend on the choice of G.

Theorem 1. *Assume that $\mathcal{G}(V, E, p)$ has k uncertain edges $e_1, e_2, ..., e_k$ and $G \sqsubseteq \mathcal{G}$ (i.e. $E_G \subseteq E$). The edit distance variance is $Var[D(\mathcal{G}, G)] = \sum_{i=1}^{k} p_i(1 - p_i)$ and does not depend on the choice of G.*

Proof. See Appendix A.1.

4.3 Observation #2: Nearby Potential Edges

As indicated by Leskovec et al. [11], real graphs reveal two temporal evolution properties: *densification power law* and *shrinking diameters*. Community Guided Attachment (CGA) model [11], which produces densifying graphs, is an example of a hierarchical graph generation model in which the linkage probability between nodes decreases as a function of their relative distance in the hierarchy. With regard to this observation, (k, ϵ)-obfuscation, by heuristically making potential edges solely based on node degree discrepancy, produces many inter-community edges. Shortest-path based statistics will be reduced due to these edges. MV, in contrast, tries to mitigate the structural distortion by proposing only *nearby* potential edges before assigning edge probabilities. Another evidence is from [21] where Vazquez analytically proved that the *Nearest Neighbor* can explains the power-law for degree distribution, clustering coefficient and average degree among the neighbors. Those properties are in very good agreement with the observations made for social graphs. Sala et al. [16] confirmed the consistency of Nearest Neighbor model in their comparative study on graph models for social networks.

5 Algorithms

This section describes steps of MV to convert the input deterministic graph into an uncertain one.

5.1 Overview

The intuition behind the new approach is to formulate the perturbation problem as a *quadratic programming* problem. Given the true graph G_0 and the number of potential edges allowed to be added n_p, the scheme has three phases. The first phase tries to partition G_0 into s subgraphs, each one with $n_s = n_p/s$ potential edges connecting nearby nodes (with default distance 2, i.e. *friend-of-friend*). The second phase formulates a quadratic program for each subgraph with the constraint of unchanged node degrees to produce the uncertain subgraphs $s\mathcal{G}$ with maximum edge variance. The third phase combines the uncertain subgraphs $s\mathcal{G}$ into $\mathcal{G}$ and publishes several sample graphs. The three phases are illustrated in Fig. 2.

By keeping the degree of nodes in the perturbed graph, our approach is similar to the *edge switching* approaches (e.g.[23]) but ours is more subtle as we do it implicitly and the switching occurs not necessarily on pairs of edges.

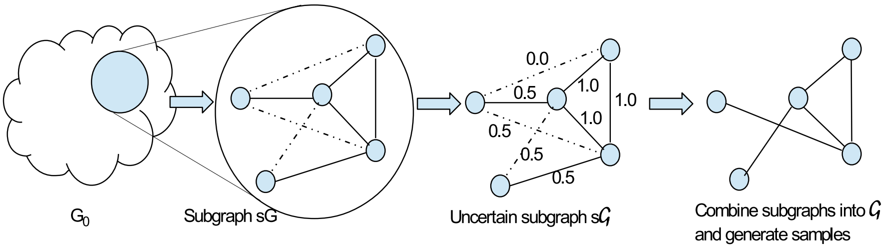

G0
Subgraph sG
0.0
0.5
1.0
1.0
1.0
0.5
0.5
0.5
Uncertain subgraph s$\mathcal{G}$
Combine subgraphs into $\mathcal{G}$
and generate samples

Fig. 2: Maximum Variance approach

5.2 Graph Partitioning

Because of the complexity of exact quadratic programming (Section 5.3), we need a pre-processing phase to divide the true graph G_0 into subgraphs and run the optimization on each subgraph. Given the number of subgraphs s, we run *METIS* [1] to get almost equal-sized subgraphs with minimum number of inter-subgraph edges. Each subgraph has n_s potential edges added before running the quadratic program. This phase is outlined in Algorithm 1.

5.3 Quadratic Programming

By assuming the independence of edges, the total edge variance of $\mathcal{G} = (V, E, p)$ for edit distance (Theorem 1) is:

$$Var(E) = \sum_{i=1}^{|E|} p_i(1 - p_i) = |E_{G_0}| - \sum_{i=1}^{|E|} p_i^2 \tag{6}$$

The last equality in (6) is due to the constraint that the expected node degrees are unchanged (i.e. $\sum_{i=1}^{d_u(\mathcal{G})} p_i = d_u(G_0)$), so $\sum_{i=1}^{|E|} p_i$ is equal to $|E_{G_0}|$. By

[1] http://glaros.dtc.umn.edu/gkhome/views/metis

Algorithm 1 Partition-and-Add-Edges

```
Input: true graph $G_0 = (V, E_{G_0})$, number of subgraphs $s$, number of potential edges
    per subgraph $n_s$
Output: list of augmented subgraphs $gl$
$gl \leftarrow$ METIS$(G_0, s)$.
for $sG$ in $gl$ do
    $i \leftarrow 0$
    while $i < n_s$ do
        randomly pick $u, v \in V_{sG}$ and $(u, v) \notin E_{sG}$ with $d(u, v) = 2$
        $E_{sG} \leftarrow E_{sG} \cup (u, v)$
        $i \leftarrow i + 1$
return $gl$
```

targeting the maximum edge variance, we come up with the following quadratic program.

$$\begin{aligned} &\text{Minimize} && \sum_{i=1}^{|E|} p_i^2 \\ &\text{Subject to} && 0 \leq p_i \leq 1 \\ & && \sum_{v \in \mathcal{N}(u)} p_{uv} = d_u(G_0) \quad \forall u \end{aligned}$$

The objective function reflects the privacy goal (i.e. the sampled graphs do not highly concentrate around the true graph) while the expected degree constraints aims to preserve the utility.

By dividing the large input graphs into subgraphs, we solve independent quadratic optimization problems. Because each edge belongs to at most one subgraph and the expected node degrees in each subgraph are unchanged, it is straightforward to show that the expected node degrees in G_0 are also fixed.

6 Quantifying Framework

This section introduces a unified framework for privacy and utility quantification of anonymization methods in which the concept of incorrectness is central to privacy assessment.

6.1 Privacy Measurement

We focus on structural re-identification attacks under various models of attacker's knowledge as shown in [10]. We quantify the privacy of an anonymized graph as the *sum* of re-identification probabilities of all nodes in the graph. We differentiate *closed-world* from *open-world* adversaries. For example, when a closed-world adversary knows that Bob has three neighbors, this fact is exact. An open-world adversary, in this case, would learn only that Bob has at least three neighbors. We consider the result of structural query Q on a node u as the node signature $sig_Q(u)$. Given a query Q, nodes having the same signature form an *equivalence class*. So given the true graph G_0 and an output anonymized graph G^*, the privacy score is measured as in the following example.

Example 1. Assuming that we have signatures of G_0 and signatures of G^* as in Table 3, the re-identification probabilities in G^* of nodes 1,2 are $\frac{1}{3}$, of nodes 4,8 are $\frac{1}{2}$, of nodes 3,5,6,7 are 0s. And the privacy score of G^* is $\frac{1}{3}+\frac{1}{3}+\frac{1}{2}+\frac{1}{2}+0+0+0+0 = 1.66$. Note that the privacy score of G_0 is $\frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{2}+\frac{1}{2}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = 3$, equal to the number of equivalence classes.

Table 3: Example 1

Graph	**Equivalence classes**
G_0	$s_1\{1,2,3\}, s_2\{4,5\}, s_3\{6,7,8\}$
G^*	$s_1\{1,2,6\}, s_2\{4,7\}, s_3\{3,8\}, s_4\{5\}$

We consider two privacy scores in this paper.

- **H1** score uses node degree as the node signature, i.e. we assume that the attacker knows *apriori* degrees of all nodes.
- $\mathbf{H2_{open}}$ uses the *set* (not multiset) of degrees of node's friends as the node signature. For example, if a node has 6 neighbors and the degrees of those neighbors are $\{1,2,2,3,3,5\}$, then its signature for $H2_{open}$ attack is $\{1,2,3,5\}$.

Higher-order scores like $H2$ (exact multiset of neighbors' degrees) or $H3$ (exact multiset of neighbor-of-neighbors' degrees) induce much higher privacy scores of the true graph G_0 (in the order of $|V|$) and represent less meaningful metrics for privacy. The following proposition claims the *automorphism-invariant* property of structural privacy scores.

Proposition 1. *All privacy scores based on structural queries [10] are automorphism-invariant, i.e. if we find a non-trivial automorphism G_1 graph of G_0, the signatures of all nodes in G_1 are unchanged.*

Proof. G_1 is an automorphism of G_0 if there exists a permutation $\pi : V \to V$ such that $(u,v) \in E_{G_0} \leftrightarrow (\pi(u), \pi(v)) \in E_{G_1}$. For $H1$ score, it is straightforward to verify that $H1_{G_1}(u) = H1_{G_0}(\pi(u))$ according to the definition of π.

For $H2_{open}$ score, we prove that $\forall d_v \in H2_{G_0}(u)$ we also have $d_v \in H2_{G_1}(\pi(u))$ and vice versa. Because $d_v \in H2_{G_0}(u) \to (u,v) \in E_{G_0} \to (\pi(u), \pi(v)) \in E_{G_1}$. Note that $d_{\pi(v)} = d_v$ ($H1$ unchanged), so $d_v \in H2_{G_1}(\pi(u))$.

The reverse is proved similarly. This argument can also apply to any structural queries (signatures) in [10]. □

6.2 Utility Measurement

Following [2] and [23], we consider three groups of statistics for utility measurement: degree-based statistics, shortest-path based statistics and clustering statistics.

Degree-based statistics

- Number of edges: $S_{NE} = \frac{1}{2}\sum_{v \in V} d_v$
- Average degree: $S_{AD} = \frac{1}{n}\sum_{v \in V} d_v$
- Maximal degree: $S_{MD} = \max_{v \in V} d_v$
- Degree variance: $S_{DV} = \frac{1}{n}\sum_{v \in V}(d_v - S_{AD})^2$
- Power-law exponent of degree sequence: S_{PL} is the estimate of γ assuming the degree sequence follows a power-law $\Delta(d) \sim d^{-\gamma}$

Shortest path-based statistics

- Average distance: S_{APD} is the average distance among all pairs of vertices that are path-connected.
- Effective diameter: S_{ED} is the 90-th percentile distance among all path-connected pairs of vertices.
- Connectivity length: S_{CL} is defined as the harmonic mean of all pairwise distances in the graph.
- Diameter : S_{Diam} is the maximum distance among all path-connected pairs of vertices.

Clustering statistics

- Clustering coefficient: $S_{CC} = \frac{3N_\Delta}{N_3}$ where N_Δ is the number of triangles and N_3 is the number of connected triples.

All of the above statistics are computed on sample graphs generated from the uncertain output $\mathcal{G}$. In particular, to estimate shortest-path based measures, we use Approximate Neighbourhood Function (ANF) [14]. The diameter is lower bounded by the longest distance among all-destination bread-first-searches from 1,000 randomly chosen nodes.

7 Evaluation

In this section, our evaluation aims to show the effectiveness and efficiency of the MV approach and verify its outperformance over the (k, ϵ)-obfuscation. The effectiveness is measured by privacy scores (lower are better) and the relative error of utility. The efficiency is measured by the running time. All algorithms are implemented in Python and run on a desktop PC with *Intel*® Core i7-4770@ 3.4Ghz, 16GB memory. We use *MOSEK*[2] as the quadratic solver.

Three large real-world datasets are used in our experiments [3]. `dblp` is a co-authorship network where two authors are connected if they publish at least one paper together. `amazon` is a product co-purchasing network. If a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. `youtube` is a video-sharing web site that includes a social network. The graph sizes $(|V|, |E|)$ of `dblp`, `amazon` and `youtube` are (317080, 1049866), (334863, 925872) and (1134890, 2987624) respectively. We partition `dblp`, `amazon` into 20 subgraphs and `youtube` into 60 subgraphs. The sample size of each test case is 20.

[2] http://mosek.com/

[3] http://snap.stanford.edu/data/index.html

7.1 Effectiveness and Efficiency

We assess privacy and utility of MV by varying n_p (the number of potential edges). The results are shown in Table 4. As for privacy scores, if we increase n_p, we gain better privacy (lower sums of re-identification probabilities) as we allow more edge switches. Due to the expected degree constraints in the quadratic program, all degree-based metrics vary only a little. By contrast, (k, ϵ)-obfuscation (cf. Table 5) does not have such advantages. The heuristics used in [2] only reaches low relative errors at small values of σ. Unfortunately, these choices give rise to privacy risks (much higher $H1, H2_{open}$ scores).

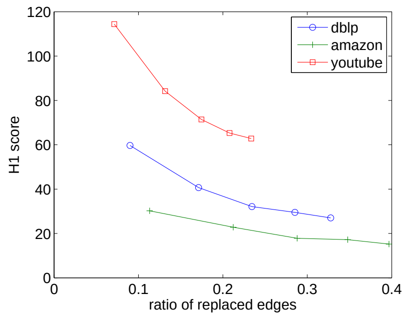


Fig. 3: $H1$ score

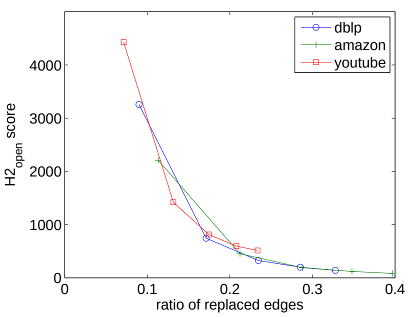


Fig. 4: $H2_{open}$ score

We observe the near *linear* relationships between $H1$, $rel.err$ and the number of replaced edges $|E_{G_0} \setminus E_G|$ in Figures 3, 5 and near *quadratic* relationship of $H2_{open}$ against $|E_{G_0} \setminus E_G|$ in Fig.4. The ratio of replaced edges in Figures 3,4 and 5 is defined as $\frac{|E_{G_0} \setminus E_G|}{|E_{G_0}|}$.

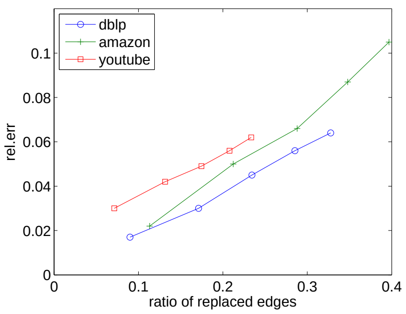


Fig. 5: Relative error

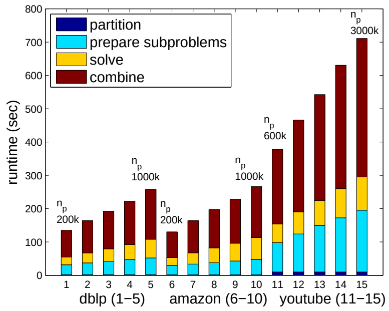


Fig. 6: Runtime of MV

The runtime of MV consists of time for (1) partitioning G_0, (2) adding friend-of-friend edges to subgraphs, (3) solving quadratic subproblems and (4) combining uncertain subgraphs to get $\mathcal{G}$. We report the runtime in Fig.6. As we can see, the total runtime is in several minutes and the partitioning step is nearly negligible. Increasing n_p gives rise to runtime in steps 2,3 and 4 and the trends are nearly linear. The runtime on `youtube` is three times longer than on the other two datasets, almost linear to their sizes.

Table 4: Effectiveness of MV (k denotes one thousand)

n_p	$H1$	$H2_{open}$	S_{NE}	S_{AD}	S_{MD}	S_{DV}	S_{CC}	S_{PL}	S_{APD}	S_{ED}	S_{CL}	S_{Diam}	rel.err
dblp	199	125302	1049866	6.62	343	100.15	0.306	2.245	7.69	9	7.46	20	
200k	59.7	3257.2	1049774	6.62	342.3	100.73	0.279	2.213	7.66	9.3	7.43	19.5	**0.017**
400k	40.7	744.0	1049813	6.62	343.5	101.26	0.255	2.189	7.56	9.1	7.33	18.9	**0.030**
600k	32.1	325.7	1050066	6.62	343.4	101.73	0.235	2.173	7.46	9.0	7.25	17.7	**0.045**
800k	29.5	199.2	1049869	6.62	345.9	102.07	0.219	2.163	7.45	9.0	7.24	17.0	**0.056**
1000k	27.0	140.7	1049849	6.62	345.4	102.29	0.205	2.155	7.34	9.0	7.15	17.0	**0.064**
amazon	153	113338	925872	5.53	549	33.20	0.205	2.336	12.75	16	12.10	44	
200k	30.2	2209.1	925831	5.53	551.5	33.83	0.197	2.321	12.38	16.1	11.72	40.5	**0.022**
400k	22.8	452.4	925928	5.53	550.2	34.40	0.182	2.306	11.88	15.3	11.28	37.1	**0.050**
600k	17.8	188.4	925802	5.53	543.9	34.79	0.167	2.296	11.60	15.0	11.04	36.9	**0.066**
800k	17.2	118.8	925660	5.53	550.0	35.11	0.154	2.289	11.33	14.4	10.81	34.5	**0.087**
1000k	15.2	82.4	925950	5.53	551.8	35.43	0.142	2.282	11.13	14.1	10.62	31.8	**0.105**
youtube	978	321724	2987624	5.27	28754	2576.0	0.0062	2.429	6.07	8	6.79	20	
600k	114.4	4428.8	2987898	5.27	28759	2576	0.0065	2.373	6.19	7.8	5.97	18.6	**0.030**
1200k	84.2	1419.2	2987342	5.26	28754	2576	0.0064	2.319	6.02	7.2	5.82	17.9	**0.042**
1800k	71.4	814.4	2987706	5.27	28745	2577	0.0062	2.287	5.97	7.1	5.78	17.2	**0.049**
2400k	65.3	595.5	2987468	5.26	28749	2577	0.0060	2.265	5.96	7.1	5.77	16.6	**0.056**
3000k	62.8	513.7	2987771	5.27	28761	2578	0.0058	2.251	5.89	7.1	5.71	16.4	**0.062**

Table 5: (k, ϵ)-obfuscation

σ	$H1$	$H2_{open}$	S_{NE}	S_{AD}	S_{MD}	S_{DV}	S_{CC}	S_{PL}	S_{APD}	S_{ED}	S_{CL}	S_{Diam}	rel.err
dblp	199	125302	1049866	6.62	343	100.15	0.306	2.245	7.69	9	7.46	20	
0.001	72.9	40712.1	1048153	6.61	316.0	97.46	0.303	2.244	7.74	9.4	7.50	20.0	**0.018**
0.01	41.1	24618.2	1035994	6.53	186.0	86.47	0.294	2.248	7.82	9.5	7.59	19.8	**0.077**
0.1	19.7	7771.4	991498	6.25	164.9	64.20	0.284	2.265	8.08	10.0	7.85	20.0	**0.128**
amazon	153	113338	925872	5.530	549	33.20	0.205	2.336	12.75	16	12.10	44	
0.001	55.7	55655.9	924321	5.52	479.1	31.73	0.206	2.340	12.14	15.2	11.65	33.2	**0.057**
0.01	34.5	39689.8	915711	5.47	299.7	27.18	0.220	2.348	12.40	15.6	11.91	32.4	**0.101**
0.1	19.2	16375.4	892140	5.33	253.9	21.87	0.232	2.374	12.52	15.5	12.06	31.4	**0.144**
youtube	978	321724	2987624	5.27	28754	2576.0	0.0062	2.429	6.07	8	6.79	20	
0.001	157.2	36744.6	2982974	5.26	28438	2522.6	0.0062	2.416	6.24	8.0	6.01	19.5	**0.022**
0.01	80.0	22361.7	2940310	5.18	26900	2282.6	0.0061	2.419	6.27	8.0	6.04	19.0	**0.043**
0.1	23.4	5806.9	2624066	4.62	16353	970.8	0.0070	2.438	6.59	8.1	6.36	20.4	**0.160**

7.2 Comparative Evaluation

Table 6 compares MV and (k, ϵ)-obfuscation. Beside the default strategy *NearBy* (nb), we include *Random* (rand) strategy for potential edges (i.e. selecting pairs of nodes uniformly on V). The column *tradeoff* is $\sqrt{H2_{open}} \times rel.err$ as we conjecture the quadratic and linear curves of $H2_{open}$ and $rel.err$ respectively (Figures 4 and 5). Clearly, MV provides better privacy-utility tradeoffs.

In addition to the re-identification scores $H1$ and $H2_{open}$, we also compute ϵ for $k \in \{30, 50, 100\}$ to have a fair comparison with (k, ϵ)-obfuscation. Table 6 justifies the better performance of MV. Our approach results in lower relative errors (better utility), lower privacy scores as well as smaller tolerance ratio ϵ (better privacy). Moreover, the worse results of *Random* strategy confirm our second observation in Section 4.3.

The number of potential edges used in MV could be 20% of $|E_{G_0}|$, much less than that of (k, ϵ)-obfuscation (100% for $c = 2$ [2]). Columns $|E_{G_0} \setminus E_G|, |E_G \setminus E_{G_0}|$ show the difference of edge sets between G_0 and samples generated from $\mathcal{G}$. Because the expected degrees are preserved, $|E_{G_0} \setminus E_G| \simeq |E_G \setminus E_{G_0}|$ in MV and are higher than those of (k, ϵ)-obfuscation where the number of edges is preserved only at small σ, i.e. we allow more edge changes while not sacrificing the utility.

Table 6: MV vs. (k, ϵ)-obfuscation (lower tradeoff is better)

	PRIVACY							UTILITY	
graph	$H1$	$H2_{open}$	$\|E_{G_0} \setminus E_G\|$	$\|E_G \setminus E_{G_0}\|$	$\epsilon(k = 30)$	$\epsilon(k = 50)$	$\epsilon(k = 100)$	rel.err	tradeoff
`dblp`	199	125302			0.00238	0.00393	0.00694		
$\sigma = 0.001$	72.9	40712.1	6993.0	5280.2	0.00039	0.00122	0.00435	0.018	**3.61**
$\sigma = 0.01$	41.1	24618.2	19317.3	5444.9	0.00051	0.00062	0.00082	0.077	**12.03**
$\sigma = 0.1$	19.7	7771.4	65285.1	6916.8	0.00179	0.00199	0.00245	0.128	**11.33**
(nb)200k	59.7	3257.2	94508.0	94416.5	0.00033	0.00077	0.00152	0.017	**0.99**
(nb)600k	32.1	325.7	246155.6	246355.3	0.00017	0.00029	0.00085	0.045	**0.82**
(rand)200k	118.4	7390.4	69838.0	69917.2	0.00092	0.00206	0.00347	0.036	**3.09**
(rand)600k	56.4	369.4	202425.5	202530.8	0.00045	0.00077	0.00171	0.078	**1.50**
`amazon`	153	113338			0.00151	0.00218	0.00456		
$\sigma = 0.001$	55.7	55655.9	6158.9	4607.4	0.00048	0.00119	0.00293	0.065	**13.40**
$\sigma = 0.01$	34.5	39689.8	14962.0	4801.3	0.00038	0.00052	0.00066	0.114	**21.33**
$\sigma = 0.1$	19.2	16375.4	39382.6	5650.3	0.00068	0.00102	0.00190	0.145	**18.46**
(nb)200k	30.2	2209.1	104800.9	104759.9	0.00023	0.00032	0.00065	0.022	**1.03**
(nb)600k	17.8	188.4	266603.7	266533.7	0.00015	0.00023	0.00047	0.066	**0.91**
(rand)200k	87.8	7728.6	76417.8	76400.4	0.00071	0.00111	0.00190	0.112	**9.88**
(rand)600k	43.2	353.0	222055.3	222276.3	0.00042	0.00065	0.00106	0.175	**3.30**
`youtube`	978	321724			0.00291	0.00402	0.00583		
$\sigma = 0.001$	157.2	36744.6	19678.5	15028.5	0.00143	0.00232	0.00421	0.022	**4.28**
$\sigma = 0.01$	80.0	22361.7	62228.55	14914.3	0.00060	0.00105	0.00232	0.043	**6.38**
$\sigma = 0.1$	23.4	5806.9	378566.0	15007.5	0.00038	0.00052	0.00074	0.160	**12.20**
(nb)600k	114.4	4428.8	213097.3	213371.4	0.00047	0.00063	0.00108	0.030	**2.00**
(nb)1800k	71.4	814.4	521709.9	521791.6	0.00040	0.00052	0.00090	0.049	**1.38**
(rand)600k	733.5	108899.3	32325.3	32273.0	0.00092	0.00096	0.00105	0.018	**5.76**
(rand)1800k	345.0	9888.8	216297.2	216160.3	0.00107	0.00134	0.00204	0.050	**5.01**

8 Conclusion

In this work, we propose a novel anonymization scheme for social graphs based on edge uncertainty semantics. To remedy the drawbacks in previous work, our MV approach exploits two key observations: maximizing degree variance while keeping the expected values unchanged and using nearby potential edges. Furthermore, we promote the usage of incorrectness measure for privacy assessment in a unified quantifying framework rather than Shannon entropy or min-entropy (k-anonymity). The experiments demonstrate the outperformance of our method over the (k, ϵ)-obfuscation. Our work may incite several directions for future research including (1) deeper analysis on the privacy-utility relationship (e.g. explaining the near-linear or near-quadratic curves) in MV (2) generalized uncertainty models for graph anonymization with constraint of unchanged expected node degrees.

References

1. L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, pages 181–190. ACM, 2007.
2. P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *Proceedings of the VLDB Endowment*, 5(11):1376–1387, 2012.

3. F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In *ICDE*, pages 924–935. IEEE, 2011.
4. J. Cheng, A. W.-c. Fu, and J. Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *SIGMOD*, pages 459–470. ACM, 2010.
5. S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh. Why waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. *Social Network Analysis and Mining*, 3(3):381–399, 2013.
6. S. Chester, B. M. Kapron, G. Srivastava, and S. Venkatesh. Complexity of social network anonymization. *Social Network Analysis and Mining*, 3(2):151–166, 2013.
7. N. Dalvi and D. Suciu. Management of probabilistic data: foundations and challenges. In *PODS*, pages 1–12. ACM, 2007.
8. A. M. Fard, K. Wang, and P. S. Yu. Limiting link disclosure in social network analysis through subgraph-wise perturbation. In *EDBT*, pages 109–119. ACM, 2012.
9. H. Gao, J. Hu, T. Huang, J. Wang, and Y. Chen. Security issues in online social networks. *Internet Computing, IEEE*, 15(4):56–63, 2011.
10. M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, 1(1):102–114, 2008.
11. J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.
12. K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD*, pages 93–106. ACM, 2008.
13. P. Mittal, C. Papamanthou, and D. Song. Preserving link privacy in social network based systems. In *NDSS*, 2013.
14. C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: A fast and scalable tool for data mining in massive graphs. In *KDD*, pages 81–90. ACM, 2002.
15. M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *Proceedings of the VLDB Endowment*, 3(1-2):997–1008, 2010.
16. A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao. Measurement-calibrated graph models for social network experiments. In *WWW*, pages 861–870. ACM, 2010.
17. R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *SP*, pages 247–262. IEEE, 2011.
18. G. Smith. On the foundations of quantitative information flow. In *Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.
19. L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
20. C.-H. Tai, P. S. Yu, D.-N. Yang, and M.-S. Chen. Privacy-preserving social network publication against friendship attacks. In *KDD*, pages 1262–1270. ACM, 2011.
21. A. Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):056104, 2003.
22. W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang. k-symmetry model for identity anonymization in social networks. In *EDBT*, pages 111–122. ACM, 2010.
23. X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *SDM*, volume 8, pages 739–750. SIAM, 2008.
24. Y. Yuan, G. Wang, H. Wang, and L. Chen. Efficient subgraph search over large uncertain graphs. *Proc. VLDB Endow*, 4(11), 2011.
25. B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, pages 506–515. IEEE, 2008.

26. L. Zou, L. Chen, and M. T. Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, 2(1):946–957, 2009.

27. Z. Zou, J. Li, H. Gao, and S. Zhang. Mining frequent subgraph patterns from uncertain graph data. *Knowledge and Data Engineering, IEEE Transactions on*, 22(9):1203–1218, 2010.

A Proof of Theorems

A.1 Proof of theorem 1

Proof. We prove the result by induction.

When $k = 1$, we have two cases of G_1: $E_{G_1} = \{e_1\}$ and $E_{G_1} = \emptyset$. For both cases, $Var[D(\mathcal{G}_1, G_1)] = p_1(1 - p_1)$, i.e. independent of G_1.

Assume that the result is correct up to $k-1$ edges, i.e. $Var[D(\mathcal{G}_{k-1}, G_{k-1})] = \sum_{i=1}^{k-1} p_i(1 - p_i)$ for all $G_{k-1} \sqsubseteq \mathcal{G}_{k-1}$, we need to prove that it is also correct for k edges. We use the subscript notations $\mathcal{G}_k, G_k$ for the case of k edges. We consider two cases of G_k: $e_k \in G_k$ and $e_k \notin G_k$.

Case 1. The formula for $Var[D(\mathcal{G}_k, G_k)]$ is

$$Var[D(\mathcal{G}_k, G_k)] = \sum_{G'_k \sqsubseteq \mathcal{G}_k} Pr(G'_k)[D(G'_k, G_k) - E[D(\mathcal{G}_k, G_k)]]^2$$

$$= \sum_{e_k \in G'_k} Pr(G'_k)[D(G'_k, G_k) - E[D_k]]^2 + \sum_{e_k \notin G'_k} Pr(G'_k)[D(G'_k, G_k) - E[D_k]]^2$$

The first sum is $\sum_{G'_{k-1} \sqsubseteq \mathcal{G}_{k-1}} p_k Pr(G'_{k-1})[D_{k-1} - E[D_{k-1}] - (1 - p_k)]^2$.

The second sum is $\sum_{G'_{k-1} \sqsubseteq \mathcal{G}_{k-1}} (1 - p_k) Pr(G'_{k-1})[D_{k-1} - E[D_{k-1}] + p_k)]^2$.

Here we use shortened notations D_k for $D(G'_k, G_k)$ and $E[D_k]$ for $E[D(\mathcal{G}_k, G_k)]$.

By simple algebra, we have $Var[D(\mathcal{G}_k, G_k)] = Var[D(\mathcal{G}_{k-1}, G_{k-1})] + q_k(1 - q_k) = \sum_{i=1}^{k} p_i(1 - p_i)$.

Case 2. similar to the Case 1. □