dK-Microaggregation: Anonymizing Graphs with Differential Privacy Guarantees

Masooma Iftikhar(✉), Qing Wang, and Yu Lin

Australian National University, Canberra, Australia
{masooma.iftikhar,qing.wang,yu.lin}@anu.edu.au

Abstract. With the advances of graph analytics, preserving privacy in publishing graph data becomes an important task. However, graph data is highly sensitive to structural changes. Perturbing graph data for achieving differential privacy inevitably leads to inject a large amount of noise and the utility of anonymized graphs is severely limited. In this paper, we propose a microaggregation-based framework for graph anonymization which meets the following requirements: (1) The topological structures of an original graph can be preserved at different levels of granularity; (2) ε-differential privacy is guaranteed for an original graph through adding controlled perturbation to its edges (i.e., edge privacy); (3) The utility of graph data is enhanced by reducing the magnitude of noise needed to achieve ε-differential privacy. Within the proposed framework, we further develop a simple yet effective microaggregation algorithm under a distance constraint. We have empirically verified the noise reduction and privacy guarantee of our proposed algorithm on three real-world graph datasets. The experiments show that our proposed framework can significantly reduce noise added to achieve ε-differential privacy over graph data, and thus enhance the utility of anonymized graphs.

Keywords: Privacy-preserving graph data publishing · Differential privacy · Graph data utility · dK-graphs · Graph anonymization

1 Introduction

Graph data analysis has been widely performed in real-life applications. For instance, online social networks are explored to analyze human social relationships, election networks are studied to discover different opinions in a community, and co-author networks are used to understand collaboration relationships among researchers [22]. However, such networks often contain sensitive or personally identifiable information, such as social contacts, personal opinions and private communication records. Publishing graph data can thus pose a privacy threat. To preserve graph data privacy, various anonymization techniques for graph data publishing have been proposed in the literature [1,11,14,24]. Nonetheless, even when a graph is anonymized without publishing any identity information, an individual may still be revealed based on structural information of a graph [11].

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 191–203, 2020.
https://doi.org/10.1007/978-3-030-47436-2_15

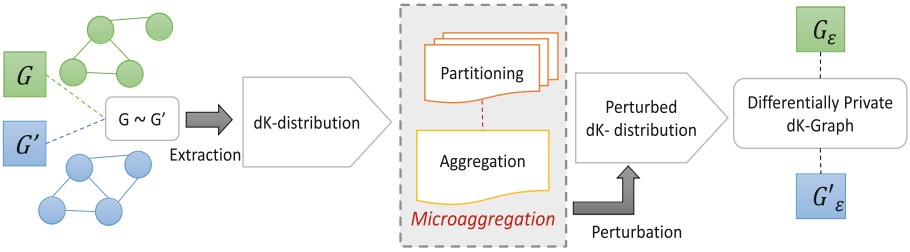

Fig. 1. A high-level overview of the proposed framework (*dK-Microaggregation*).

In recent years, differential privacy [5] has emerged as a widely recognized mathematical framework for privacy. A number of studies [10,18] have investigated the problem of publishing anonymized graphs under guarantee of differential privacy. However, graph data is highly sensitive to structural changes. Directly perturbing graph data often leads to inject a large amount of random noise and the utility of anonymized graphs is severely impacted. To deal with this issue, several works [19–22] have explored techniques of indirectly perturbing graph data through a graph abstraction model, such as the dK-graph model [16] and hierarchical random graph (HRG) model [2], or spectral graph methods. The central ideas behind these works are to first project a graph into a statistical representation (e.g., degree distribution and dendrogram), or a spectral representation (e.g., adjacency matrix), and then add random noise to perturb such representations. Although these techniques are promising, they can only achieve ε-differential privacy over a graph by injecting the magnitude of random noise proportional to the sensitivity of queries, which is fixed to global sensitivity. Due to the high sensitivity of graph data on structural changes, the utility of anonymized graphs being published by these works is still limited.

To alleviate this limitation, we aim to develop a general framework of anonymizing graphs which meets the following requirements: (1) The topological structures of an original graph can be preserved at different levels of granularity; (2) ε-differential privacy is guaranteed for an original graph through adding controlled perturbation to its edges (i.e., edge privacy [13]); (3) The utility of graph data is enhanced by reducing the magnitude of noise needed to achieve ε-differential privacy. We observe that the dK-graph model [15,16] for analyzing network topologies can serve as a good basis for generating structure-preserving anonymized graphs. Essentially, the dK-graph model generates dK-graphs by retaining a series of network topology properties being extracted from d-sized subgraphs in an original graph. In order to reduce the amount of random noise without compromising ε-differential privacy, we incorporate microaggregation techniques [4] into the dK graph model to reduce the sensitivity of queries. This enables to apply perturbation on network topology properties at a flexible level of granularity, depending on the degree of microaggregation.

Figure 1 provides a high-level overview of our proposed framework. Given two neighboring graphs $G \sim G'$, network topology properties such as dK-distributions [16] are first extracted from each graph. Then a dK-distribution goes through a microaggregation procedure, which consists of partition and aggregation. After that, the microaggregated dK-distribution is perturbed, yielding a ε-differentially private dK-distribution. Finally, based on the perturbed dK-distribution, ε-differentially private dK-graphs are generated. That is, for two neighboring graphs $G \sim G'$, their corresponding anonymized graphs generated by this framework are ε-indistinguishable.

Contributions. To summarize, our work has the following contributions: (1) We present a novel framework, called *dK-microaggregation*, that can leverage a series of network topology properties to generate ε-differentially private anonymized graphs. (2) We propose a distance constrained algorithm for approximating dK-distributions of a graph via microaggregation within the proposed framework, which enables us to reduce the amount of noise being added into ε-deferentially private anonymized graphs. (3) We have empirically verified the noise reduction of our proposed framework on three real-world networks. It shows that our algorithm can effectively enhance the utility of generated anonymized graphs by providing better within-cluster homogeneity and reducing the amount of noise, in comparison with the state-of-the-art methods.

2 Problem Formulation

Let $G = (V, E)$ be a simple undirected graph, where V is the set of nodes and E the set of edges in G. We use $deg(v)$ to denote the degree of a node v, and $deg(G)$ to denote the maximum degree of G.

Definition 1 (Neighboring graphs). *Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be* neighboring graphs*, denoted as $G \sim G'$, iff $V = V'$, $E \subset E'$ and $|E| + 1 = |E'|$.*

The dK-graph model [16] provides a systematic way of extracting subgraph degree distributions from a given graph, i.e. *dK-distributions*.

Definition 2 (dK-distribution). *A* dK-distribution *$dK(G)$ over a graph G is the probability distribution on the connected subgraphs of size d in G.*

Specifically, 1K-distribution captures a degree distribution, 2K-distribution captures a joint degree distribution, i.e. the number of edges between nodes of different degrees, and 3K-distribution captures a clustering coefficient distribution, i.e. the number of triangles and wedges connecting nodes of different degrees. When $d = |V|$, dK-distribution specifies the entire graph. For larger values of d, dK-distributions capture more complex properties of a graph at the expense of higher computational overhead [16]. To describe how a dK-distribution is extracted from a graph, we define the notion of dK function.

Definition 3 (DK FUNCTION). *Let* $\mathbb{G} = \{(V, E')|E' \subseteq V \times V\}$ *be the set of all graphs with the set* V *of nodes. A* dK function $\gamma^{dK} : \mathbb{G} \rightarrow \mathbb{D}$ *maps a graph in* $\mathbb{G}$ *to its dK-distribution in* $\mathbb{D}$ *s.t.* $\gamma^{dK}(G) = dK(G)$.

Following the previous work [16], we define *dK-graph* as a graph that can be constructed through reproducing the corresponding dK-distribution.

Definition 4 (DK-GRAPH). *A* dK-graph *over* $dK(G)$ *is a graph in which connected subgraphs of size* d *satisfy the probability distribution in* $dK(G)$.

Conceptually, a dK-graph is considered as an anonymized version of an original graph G that retains certain topological properties of G at a chosen level of granularity. In this paper, we aim to generate dK-graphs with ε-differential privacy guarantee for preserving privacy of structural information between nodes of a graph (edge privacy). We formally define *differentially private dK-graph* below.

Definition 5 (DIFFERENTIALLY PRIVATE DK-GRAPHS). *A randomized mechanism* $\mathcal{K}$ *provides* ε*-differentially private dK-graphs, if for each pair of neighboring graphs* $G \sim G'$ *and all possible outputs* $\mathcal{G} \subseteq range(\mathcal{K})$, *the following holds*

$$Pr[\mathcal{K}(G) \in \mathcal{G}] \leq e^{\varepsilon} \times Pr[\mathcal{K}(G') \in \mathcal{G}]. \tag{1}$$

$\mathcal{G}$ is a family of dK-graphs, and $\varepsilon > 0$ is the *differential privacy parameter*. Smaller values of ε provide stronger privacy guarantees [5].

3 dK-Microaggregation Framework

In this section, we present a novel framework *dK-Microaggregation* for generating ε-differentially private dK-graphs. Without loss of generality, we will use 2K-distribution to illustrate our proposed framework. This is due to two reasons: (1) As previously discussed in [15,16], the $d = 2$ case is sufficient for most practical purposes; (2) dK-generators for $d = 2$ have been well studied [9,15], whereas dK-generators for $d \geq 3$ have not been yet discovered [9]. Given a graph $G = (V, E)$, we have $2K(G) = \{(g, g', m)|m = |E_{(g,g')}|\}$ where (g, g') is a degree pair and $E_{(g,g')} = \{(v, v') \in E|g = deg(v) \wedge g' = deg(v')\}$ is the set of edges with the degree pair (g, g').

Previous studies [19,20] have shown that, changing a single edge in a graph may result in one or more changes on tuples in its corresponding dK-distribution. The following lemma states the maximum number of changes between the 2K-distributions of two neighboring graphs.

Lemma 1. *Let* $G \sim G'$ *be two neighboring graphs. Then* $\gamma^{dK}(G)$ *and* $\gamma^{dK}(G')$ *differ in at most* $4 \times g + 1$ *tuples, where* $d = 2$ *and* $g = max(\{deg(G), deg(G')\})$.

In our work, for each dK-distribution D, we want to generate D_{ε} that is an anonymized version of D satisfying ε-differential privacy. Thus, we view the response to a dK function γ^{dK} for $d = 2$ as a collection of responses to *degree queries*, one for each tuple in a 2K distribution.

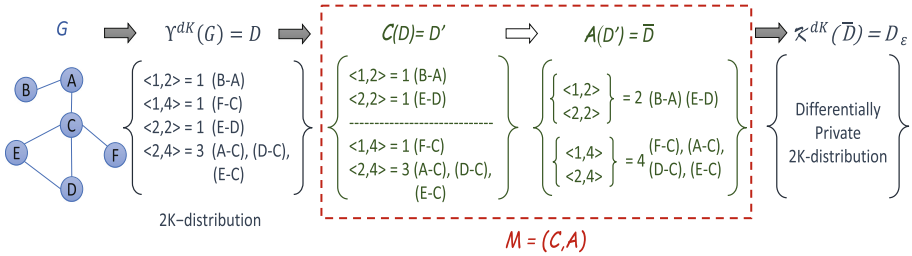

Fig. 2. An illustration of our proposed algorithms.

Definition 6 (Degree query). *A* degree query $q_t : \mathbb{G} \rightarrow \mathbb{R}$ *maps a degree pair* $t = (g_1, g_2)$ *in a graph* $G \in \mathbb{G}$ *to a frequency value in* $\mathbb{R}$ *s.t.* $(g_1, g_2, q_t(G)) \in \gamma^{dK}(G)$.

To guarantee ε-differential privacy for each q_t, we can add random noise into the real response $q_t(G)$, yielding a randomized response $q_t(G) + Lap(\Delta(q_t)/\varepsilon)$, where $\Delta(q_t)$ denotes the sensitivity of q_t and $Lap(\Delta(q_t)/\varepsilon)$ denotes random noise drawn from a Laplace distribution.

If we query D with a set of degree queries $\{q_t\}_{t \in D}$ and the response to each q_t satisfies ε-differential privacy, by the parallel composition property of differential privacy [17], we can generate D_ε that satisfies ε-differential privacy. However, the total amount of random noise being added into the responses can be very high, particularly when a graph is large. To control the amount of random noise and thus increase the utility of D_ε, we microaggregate similar tuples in D before adding noise. Thus, the dK function γ^{dK} is replaced by $\gamma^{dK} \circ \mathcal{M}$, i.e., we run γ^{dK} on the microaggregated dK-distribution $\overline{D}$ resulting from running a microaggregation algorithm $\mathcal{M}$ over D. The response to $\gamma^{dK} \circ \mathcal{M}$ is a collection of responses to *microaggregate degree queries*, one for each cluster in $\overline{D}$.

Definition 7 (Microaggregate degree query). *A* microaggregate degree query $q^*_T : \mathbb{G} \rightarrow \mathbb{R}$ *maps a set of degree pairs* T *in a graph* $G \in \mathbb{G}$ *to a frequency value in* $\mathbb{R}$ *s.t.* $q^*_T(G) = sum(\{q_t(G) | t = (g_1, g_2), t \in T, (g_1, g_2, q_t(G)) \in \gamma^{dK}(G)\})$.

Indeed, we can see that q_t is a special case of q^*_T since $q_t(G) = q^*_T(G)$ holds for $T = \{t\}$. By Lemma 1, we have the following lemma about q_t and q^*_T.

Lemma 2. *The sensitivity of both* q_t *and* q^*_T *on a graph* G *is upper bounded by* $(4 \times deg(G) + 1)$.

For each cluster in $\overline{D}$ that is resulted from running $\mathcal{M}$, only aggregated frequency value for a cluster of tuples is returned by a microaggregate degree query. Thus, $\gamma^{dK} \circ \mathcal{M}$ is less "sensitive" when the number of clusters in $\overline{D}$ is smaller. By Lemma 2 and the fact that changing one edge on a graph may lead to changes on multiple clusters in $\overline{D}$, we have the following lemma about the sensitivity of $\gamma^{dK} \circ \mathcal{M}$.

Lemma 3. *Let $C_1, \ldots C_n$ be the clusters in $\overline{D}$ resulting from running $\mathcal{M}$ over $\gamma^{dK}(G)$. Then the sensitivity of $\gamma^{dK} \circ \mathcal{M}$ is upper bounded by $(4 \times g + 1) \times n$.*

Generally, dK-microaggregation works in the following steps. First, it extracts a dK-distribution from a graph. Then, it microaggregates the dK-distribution and perturbs the microaggregated dK-distribution to generate ε-differentially private dK-distribution. Finally, a dK-graph is generated.

4 Proposed Algorithm

In this section, we discuss algorithms for microaggregating dK-distributions. Generally, a microaggregation algorithm for dK-distributions $\mathcal{M} = (\mathcal{C}, \mathcal{A})$ consists of two phases: (a) *Partition* - similar tuples in a dK-distribution are partitioned into the same cluster; (b) *Aggregation* - the frequency values of tuples in the same cluster are aggregated. As illustrated in Fig. 2, a 2K-distribution D is partitioned into multiple clusters by a clustering function $\mathcal{C}$, i.e. $\mathcal{C}(D) = D'$. Then, the frequency values of tuples in each cluster are aggregated by an aggregate function $\mathcal{A}$, i.e. $\mathcal{A}(D') = \overline{D}$.

MDAV-dK Algorithm. Given a dK-distribution $D = \gamma^{dK}(G)$ over a graph G, a simple way of microaggregating D is to partition D in such a way that each cluster contains at least k tuples. For this, we use a simple microaggregation heuristic, called *Maximum Distance to Average Vector* (MDAV) [4], which can generate clusters of the same size k, except one cluster of size between k and $2k - 1$. However, unlike a standard version of MDAV that aggregates each cluster by replacing each record in the cluster with a representative record, we perform aggregation to aggregate frequency values of tuples in each cluster into an aggregated frequency value. To avoid ambiguity, we call our MDAV-based algorithm for microaggregating dK-distributions the *MDAV-dK* algorithm.

It is well-known that, for many real-world networks such as Twitter, their degree distributions are often highly skewed. This often leads to highly skewed dK-distributions for such networks. However, due to inherent limitations of MDAV, e.g. the fixed-size constraint, MDAV-dK would suffer significant information loss when evenly partitioning a highly skewed dK-distribution into clusters of the same size. To address this issue, we propose an algorithm called *Maximum Pairwise Distance Constraint* (MPDC-dK).

MPDC-dK Algorithm. Unlike MDAV-dK, MPDC-dK aims to partition a dK-distribution into clusters under a distance constraint. Specifically, after partitioning, the distances between the corresponding degrees in any two tuples within a cluster should be no greater than a specified distance interval τ. Take a 2K-distribution D for example. Let $t_1 = (g_1, g'_1, m_1)$ and $t_2 = (g_2, g'_2, m_2)$ be two tuples in a cluster after applying MPDC-dK on D. Then, we say that these two tuples satisfy a distance constraint τ iff $\max(|g_1 - g_2|, |g'_1 - g'_2|) \leq \tau$. The clustering problem addressed by MPDC-dK is thus to generate the minimum number of clusters in which every pair of tuples from the same cluster satisfies such a distance constraint τ.

Algorithm 1: *MPDC-dK*

Input: D: dK-distribution; τ: distance interval
Output: D': set of clusters

```
1  D' := φ;
2  b_list := [ ];
3  foreach (g, g') ∈ D do
4      foreach b_i ∈ covering_boxes((g, g'), τ) do
5          Add b_i to b_list (if not exist) and increase the count of b_i by 1 in b_list.
6          Add (g, g') to b_i in b_list
7  while b_list is not empty do
8      b_max ← the box with the maximum count
9      d_max ← the degree pairs in b_max
10     D' := D' ∪ {d_max}
11     Remove b_max from b_list.
12     foreach (g, g') ∈ d_max do
13         foreach b_i ∈ covering_boxes((g, g'), τ) do
14             Remove (g, g') from b_i in b_list
15             Decrease the count of b_i in b_list by 1 and remove b_i if its count is 0
16 Return D'
```

The conceptual ideas behind our MPDC-dK algorithm design is to consider each degree pair (g, g') as coordinates in a two dimensional space, and also treat the above distance constraint τ as a τ-by-τ box, denoted by $((x, x'), \tau)$ and centered at (x, x'), in the same two dimensional space. Clearly, such a box corresponds to a cluster that satisfies the distance constraint τ, and a box $((x, x'), \tau)$ covers a degree pair (g, g') iff $x - \tau/2 \leq g \leq x + \tau/2$ and $x' - \tau/2 \leq g' \leq x' + \tau/2$. MPDC-dK employs a greedy algorithm to find the minimum number of boxes (i.e., clusters) that cover all degree pairs. MDPC-dK first enumerates all boxes that cover at least one degree pair and records the corresponding counts as the number of degree pairs being covered by these boxes. MDPC-dK then recursively selects a box with the maximum count (i.e., covering the maximum number of degree pairs) in a greedy manner, assigns these degree pairs in a new cluster, and removes them from other boxes until all boxes are empty. After that, MDPC-dK performs aggregation to aggregate the frequency values of tuples in each cluster into an aggregated frequency value.

Algorithm 1 describes the details of our MPDC-dK algorithm. Given a dK-distribution D, we start with initializing an empty cluster list D' (Line 1) and a list b_list to record each box and its corresponding degree pairs, and the total number of degree pairs covered by the box (Line 2). For each degree pair (g, g') in D, we enumerate boxes that cover (g, g') using a function *covering_boxes* (Line 4). For each enumerated box b_i we update the list by adding (g, g') to b_i and increment the count of b_i by 1 (Lines 5–6). After creating b_list, we iteratively select a box b_{max} with the maximum count for degree pairs (Line 8),

then generate a new cluster of degree pairs in d_{max}, and add it into the cluster list (Lines 9–10). We further remove b_{max} and all degree pairs in b_{max} from b_list and update the counts of affected boxes in b_list (Lines 11–15). The algorithm terminates when b_list is empty and returns a set of generated clusters D'.

5 Theoretical Discussion

Privacy Analysis. Here, we theoretically show that dK-graphs generated in our proposed framework are differentially private. Firstly, by Lemma 2 and 3, we can obtain a ε-differentially dK-distribution D_ε by microaggregating a dK-distribution and calibrating the amount of random noise according to the sensitivity of microaggregated degree queries. As D_ε only contains aggregated frequency values for clusters of tuples in a dK-distribution, we perform post-processing using a randomized algorithm f to randomly select tuples within each cluster of D_ε until the aggregated frequency value of the cluster is reached. Previously, Dwork and Roth [6] proved that differential privacy is immune to post-processing, i.e., the composition of a randomized algorithm with a differentially private algorithm is differentially private. This leads to the lemma below.

Lemma 4. *Let D_ε be a ε-differentially private dK-distribution and f be a randomized algorithm for post-processing D_ε. Then $f(D_\varepsilon)$ is also a ε-differentially private dK-distribution.*

Based on $f(D_\varepsilon)$, a dK-graph can be generated using a dK-graph generator [15,16]. Following Lemma 4, Definition 5, and the proposition of Dwork and Roth [6] on post-processing, we have the following theorem for our framework, which corresponds to a randomized mechanism $\mathcal{K} = \gamma^{dK} \circ \mathcal{M} \circ \mathcal{K}^{dK} \circ f \circ \widehat{\gamma}^{dK}$, where $\widehat{\gamma}^{dK} : \mathbb{D} \rightarrow \mathbb{G}$ is a dK-graph generator.

Theorem 1. *$\mathcal{K}$ generates ε-differentially private dK-graphs.*

Complexity Analysis. We analyze the time complexity of the algorithms MDAV-dK and MPDC-dK. For MDAV-dK with a constraint on the minimum size k of clusters, given a dK-distribution D as input, the complexity of MDAV-dK for clustering is similar to MDAV [4], i.e. $\mathcal{O}(n^2)$. For MPDC-dK with a constraint on the distance interval τ, in order to generate clusters, MPDC-dK needs to perform a sequential search over all degree pairs in D. Firstly, MPDC-dK needs to enumerate boxes for all the degree pairs, and each degree pair is covered by at most $(\tau+1)^2$ boxes (Line 4 of Algorithm 1), hence the cost of enumerating boxes is $\mathcal{O}(\tau^2 n)$ (Line 3–6 of Algorithm 1). Secondly, MPDC-dK sorts the boxes based on the corresponding degree pairs being covered, and selects and removes the box with the maximum count iteratively. Although it takes $\mathcal{O}(nlogn)$ to sort and greedily select the box with the maximum count for the first iteration, each later iteration only costs $\mathcal{O}(\tau^2 logn)$ (Line 8 of Algorithm 1) because each box overlaps with at most $4\tau^2$ other boxes and removing one box only affects the count of $\mathcal{O}(\tau^2)$ boxes (Lines 11–15 of Algorithm 1). Hence, the cost of selecting and removing boxes is $\mathcal{O}(\tau^2 nlogn)$ (Lines 7–15 of Algorithm 1). The overall complexity of MPDC-dK for clustering is $\mathcal{O}(\tau^2 nlogn)$.

6 Experiments

We have evaluated the proposed framework to answer the following questions:

- **Q1.** How does dK-microaggregation reduce the amount of noise added into dK-distributions while still providing ε-differential privacy guarantee?
- **Q2.** How does our microaggregation algorithms perform in providing better within cluster homogeneity for dK-distributions?
- **Q3.** What are the trade-offs between utility and privacy when generating differentially private dK-graphs?

Datasets. We used three network datasets in the experiments: (1) *polbooks*[1] contains 105 nodes and 441 edges. It is a network of books about US politics. (2) *ca-GrQc* (see footnote 1) contains 5,242 nodes and 14,496 edges. (3) *ca-HepTh* (see footnote 1) contains 9,877 nodes and 25,998 edges. Both *ca-GrQc* and *ca-HepTh* are scientific collaborative networks between authors and papers.

Baseline Methods. In order to evaluate our proposed framework, we considered the following methods: (1) ε-DP, which is a standard ε-differential privacy

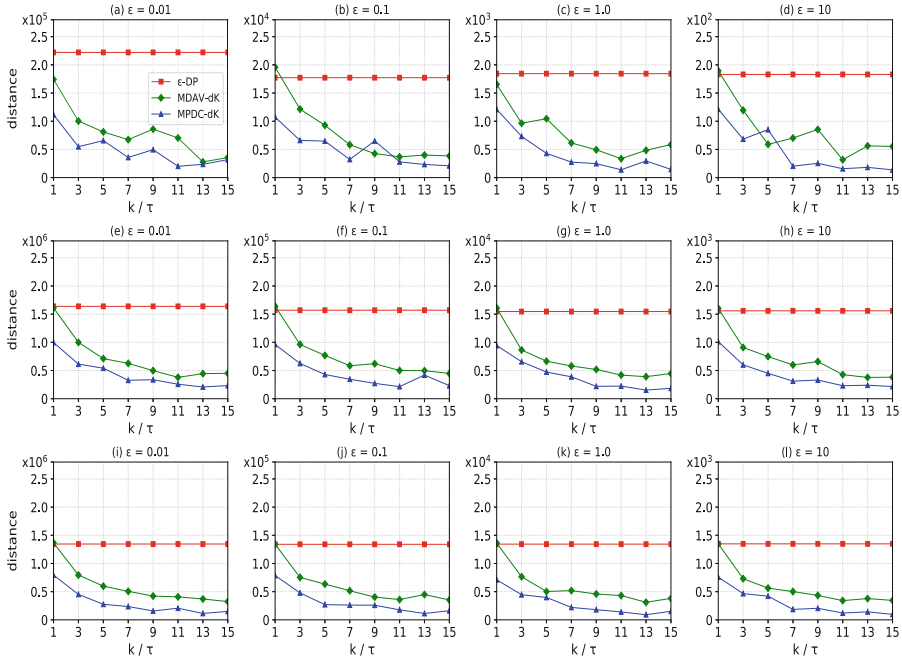

Fig. 3. Comparison on the Euclidean distance between original and perturbed dK-distributions under varying k, τ, and ε over three datasets: (a)–(d) *polbooks* dataset, (e)–(h) *ca-GrQc* dataset, and (i)–(l) *ca-HepTh* dataset.

[1] *polbooks* is available at http://networkrepository.com/polbooks.php; *ca-GrQc* and *ca-HepTh* are available at http://snap.stanford.edu/data/index.html.

algorithm in which noise is added using the Laplace mechanism [5]. (2) MDAV-dK which extends the standard microaggregation algorithm MDAV [4] for handling dK-distributions. (3) MPDC-dK is our proposed dK-microaggregation algorithm. We used *Orbis* [15] to generate 2K-distributions.

Evaluation Measures. We used Euclidean distance [19] to measure network structural error between original and perturbed dK-distributions. For clustering algorithms, we measure within-cluster homogeneity using the sum of absolute error [7] defined as $SAE = \sum_{i=1}^{N} \sum_{\forall x_j \in c_i} |x_j - \mu_i|$ where c_i is the set of tuples in cluster i and μ_i is the mean of cluster i.

Table 1. Performance of MDAV-dK under different values of k.

Datasets	Measures	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
polbooks	SAE	0	144.6	184.67	**224.84**	273.6	292.21	299.15	334.25
	# Clusters	161	53	32	**23**	17	14	12	10
ca-GrQc	SAE	0	1073.3	1476	**1810.5**	2166.8	2313.7	2555.5	2730
	# Clusters	1233	411	246	**176**	137	112	94	82
ca-HepTh	SAE	0	968.72	1304	1599.8	**1893.9**	2063	2232.9	2389.7
	# Clusters	1295	431	259	185	**143**	117	99	86

Table 2. Performance of MPDC-dK under different values of τ.

Datasets	Measures	$\tau=1$	$\tau=3$	$\tau=5$	$\tau=7$	$\tau=9$	$\tau=11$	$\tau=13$	$\tau=15$
polbooks	SAE	90.72	**192.15**	328.96	424.2	563.73	617.63	723.06	795.77
	# Clusters	68	**25**	13	8	7	5	3	3
ca-GrQc	SAE	725.38	**1732.1**	2630.6	3470.6	4262.9	5176.7	6170.1	7037.7
	# Clusters	483	**178**	98	61	42	35	26	20
ca-HepTh	SAE	841.87	**1761.8**	2773.3	3721.4	4719.2	5623.8	6402.6	7034.2
	# Clusters	412	**140**	73	37	34	24	19	15

Experimental Results. To verify the overall utility of ε-differentially private dK-distribution, we first conducted experiments to compare the structural error between original and perturbed dK-distributions generated by our algorithm MDAV-dK, MPDC-dK and the baseline method ε-DP. Figure 3 presents our experimental results. For ε-DP, we used the following privacy parameters $\varepsilon = [0.01, 0.1, 1.0, 10.0]$, which cover the range of differential privacy levels widely used in the literature [12]. The results for ε-DP is displayed as horizontal lines, as ε-DP does not depend on the parameters k and τ.

From Fig. 3, we can see that, for all three datasets, our proposed algorithms MDAV-dK and MPDC-dK lead to less structural error for every value of ε as compared to ε-DP. This is because, by approximating a query γ to $\gamma \circ \mathcal{M}$ via dk-microaggregation, the errors caused by random noise to achieve ε-differential

privacy are reduced significantly. Thus, dK-microaggregation introduces overall less noise to achieve differential privacy.

We then conducted experiments to compare the quality of clusters, in terms of within-cluster homogeneity, generated by MDAV-dK and MPDC-dK. The results are shown in Tables 1 and 2. We observe that, for values of k and τ at which MDAV-dK and MPDC-dK generate almost the same number of clusters, as highlighted in bold, MPDC-dK outperforms MDAV-dK by producing clusters with less SAE over all three datasets. This is consistent with the previous discussion in Sect. 4. As MPDC-dK always partitions degree pairs under a distance constraint rather than a fixed-size constraint, thus it generates more homogeneous clusters as compared to MDAV-dK.

Discussion. We analyze the trade-offs between utility and privacy of dK-graphs generated in the proposed framework. To enhance the utility of differentially private dK-graphs, we approximated an original query γ to $\gamma \circ \mathcal{M}$. This thus introduces two kinds of errors: one is random noise to guarantee ε-differential privacy, and the other one is due to microaggregation. We have noticed that, the second kind of error can be reduced by generating homogeneous clusters during microaggregation. On the other hand, for the first kind of error which depends on the sensitivity of $\gamma \circ \mathcal{M}$, it dominates the impact on the utility of differentially private dK-graphs generated via dk-microaggregation. By reducing sensitivity we can increase the utility of dK-graphs without compromising privacy.

7 Related Work

Graph data anonymization has been widely studied in the literature, and many anonymization techniques [1,11,14,24] have been proposed to enforce privacy over graph data. These techniques can be broadly categorized into three areas: nodes and edges perturbation, k-anonymity, and differential privacy. Perturbation-based approaches follow certain principles to process nodes and edges, including identity removal [14], edge modification [23], nodes clustering [11], and so on. Generally, k-anonymity approaches divide an original graph into at least k-sized blocks so that the probability that an adversary can re-identify one node's identity is at most $1/k$. Popular k-anonymity approaches for graph anonymization include k-candidate [11], k-neighborhood anonymity (k-NA) [24], k-degree anonymity (k-DA) [14], k-automorphism, and k-isomorphism (k-iso) [1].

Differential privacy on graph data can be roughly divided into two categories, namely: node differential privacy [3] and edge differential privacy [13]. In general, unlike k-anonymity, differential privacy approaches have mathematical proofs of privacy guarantee. Nevertheless, applying differential privacy on graph data limits utility because graph is highly sensitive to structural changes and adding noise directly into graph data can significantly degrade its utility. To address this issue, many approaches [19–22] perturb various statistical information of a graph by projecting graph data into other domains using feature-abstraction models [2,16]. This idea is appealing; however it leads to yielding less data utility

due to injecting random noise based on the global sensitivity to guarantee ε-differential privacy. Our aim is to anonymize graphs under ε-differential privacy using less sensitive queries. In this regard, we proposed a microaggregation-based framework which reduces the sensitivity via microaggregation, thus reducing the overall noise needed to achieve ε-differentially private graphs.

8 Conclusion

In this paper, we have formalized a general microaggregation-based framework for anonymizing graphs that preserves the utility of dK-graphs while enforcing ε-differential privacy. Based on the proposed framework, we have proposed an algorithm for microaggregating dK-distributions under a distance constraint. We have theoretically analyzed the privacy property of our framework and the complexity of our algorithm. The effectiveness of our work has been empirically verified over three real-world datasets. Future extensions to this work will consider zero knowledge privacy (ZKP) [8], to release statistics about social groups in a network while protecting privacy of individuals.

References

1. Cheng, J., Fu, A.W.C., Liu, J.: K-isomorphism: privacy preserving network publication against structural attacks. In: SIGMOD, pp. 459–470 (2010)
2. Clauset, A., Moore, C., Newman, M.E.: Hierarchical structure and the prediction of missing links in networks. Nature **453**(7191), 98–101 (2008)
3. Day, W.Y., Li, N., Lyu, M.: Publishing graph degree distribution with node differential privacy. In: SIGMOD, pp. 123–138 (2016)
4. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Min. Knowl. Discov. **11**(2), 195–212 (2005). https://doi.org/10.1007/s10618-005-0007-5
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
6. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. FnT-TCS **9**(3–4), 211–407 (2014)
7. Estivill-Castro, V., Yang, J.: Fast and robust general purpose clustering algorithms. In: Mizoguchi, R., Slaney, J. (eds.) PRICAI 2000. LNCS (LNAI), vol. 1886, pp. 208–218. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44533-1_24
8. Gehrke, J., Lui, E., Pass, R.: Towards privacy for social networks: a zero-knowledge based definition of privacy. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 432–449. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_26
9. Gjoka, M., Kurant, M., Markopoulou, A.: 2.5 k-graphs: from sampling to generation. In: INFOCOM, pp. 1968–1976 (2013)
10. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: ICDM, pp. 169–178 (2009)
11. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. In: PVLDB, pp. 102–114 (2008)

12. Iftikhar, M., Wang, Q., Lin, Y.: Publishing differentially private datasets via stable microaggregation. In: EDBT, pp. 662–665 (2019)
13. Jorgensen, Z., Yu, T., Cormode, G.: Publishing attributed social graphs with formal privacy guarantees. In: SIGMOD, pp. 107–122 (2016)
14. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: SIGMOD, pp. 93–106 (2008)
15. Mahadevan, P., Hubble, C., Krioukov, D., Huffaker, B., Vahdat, A.: Orbis: rescaling degree correlations to generate annotated internet topologies. In: SIGCOMM, pp. 325–336 (2007)
16. Mahadevan, P., Krioukov, D., Fall, K., Vahdat, A.: Systematic topology analysis and generation using degree correlations. In: SIGCOMM, pp. 135–146 (2006)
17. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: SIGMOD, pp. 19–30 (2009)
18. Proserpio, D., Goldberg, S., McSherry, F.: Calibrating data to sensitivity in private data analysis. In: PVLDB, pp. 637–648 (2014)
19. Sala, A., Zhao, X., Wilson, C., Zheng, H., Zhao, B.Y.: Sharing graphs using differentially private graph models. In: SIGCOMM, pp. 81–98 (2011)
20. Wang, Y., Wu, X.: Preserving differential privacy in degree-correlation based graph generation. Trans. Data Priv. **6**(2), 127–145 (2013)
21. Wang, Y., Wu, X., Wu, L.: Differential privacy preserving spectral graph analysis. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 329–340. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_28
22. Xiao, Q., Chen, R., Tan, K.L.: Differentially private network data release via structural inference. In: SIGKDD. pp. 911–920 (2014)
23. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: SDM, pp. 739–750 (2008)
24. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: ICDE, pp. 506–515 (2008)