A Probability Mapping-based Privacy Preservation Method for Social Networks

Qingru Li[1[0000-0003-2532-045X]], Yahong Wang[1[0009-0004-4906-4846]], Fangwei Wang[1(✉)[0000-0003-3888-8167]], Zhiyuan Tan[2[0000-0001-5420-2554]], and Changguang Wang[1(✉)[0000-0002-2054-9215]]

[1]College of Computer and Cyberspace Security, Hebei Normal University, Shijiazhuang 050024, China

{fw_wang, wangcg}@hebtu.edu.cn

[2] School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, EH10 5DT, UK

Abstract. The mining and analysis of social networks can bring significant economic and social benefits. However, it also poses a risk of privacy leakages. Differential privacy is a de facto standard to prevent such leaks, but it suffers from the high sensitivity of query functions. Although projection is a technique that can reduce this sensitivity, existing methods still struggle to maintain a satisfactory level of sensitivity in query functions. This results in lower data utility and an inevitable risk of privacy leakage. To prevent the disclosure of user privacy, we need to significantly reduce the sensitivity of the query functions and minimize the error of the projected values with respect to the original values. To address this issue, we first explore the influence of mapping and projection on reducing the sensitivity of query functions. We then propose a Probability Mapping (PM) algorithm, based on multi-armed bandit, which however tends to generate mapped graphs with a wide range of degrees and containing considerable nodes with high degrees. Thus, we develop a new Probability Projection (PP) algorithm to overcome these weaknesses. Finally, we propose four histogram publishing algorithms built upon PM and PP, namely PMTC, PPTC, PMCTC and PPCTC. Extensive experimental results on three different sized datasets show that PM and PP not only retain more edge information and reduce the error but also improve the data availability.

Keywords: Differential Privacy; Multi-Armed Bandit; Mapping; Projection; Histogram; Social Networks

1 Introduction

The amount of data shared on social networks has increased significantly and contains a lot of personal information and social connections. This sharing has led to a significant breach of privacy. For instance, in March 2020, some data sets from Sina Weibo were stolen by attackers, which resulted in the exposure of information belonging to 538 million Weibo users. Similarly, in June 2021, LinkedIn was hacked, and data of approximate 500 million users were stolen. As a result, researchers are now focusing on the issue of privacy protection for social networks, and the development of social networks with privacy protection features is a current research priority.

Nowadays, there are various techniques available to safeguard data privacy. Some of these include *K*-anonymity [1], *L*-diversity [2], *t*-closeness [3], and *m*-invariance [4]. While these methods can effectively prevent the disclosure of user information, they may not be able to protect against background knowledge attacks. However, the differential privacy mechanism has a solid mathematical foundation and is capable of defending against such attacks. As a result, this mechanism is commonly employed in the release of private data on social media platforms [5].

When publishing differential privacy data, it can be difficult to balance query function sensitivity and data availability. To reduce query function sensitivity, projection is a common technique that maps an original graph to a graph with nodes having a maximum degree of θ. However, current projection methods may compromise privacy or lose original information. To address these issues, we introduce a new projection method, Probability Projection (PP), that limits edge increase through probability and node degree thresholds. Our proposed method provides better privacy protection with less information loss. Overall, our contributions include developing a novel projection method that enhances privacy and preserves data integrity.

This paper is structured as follows. Section 2 reviews related research on differential privacy algorithms. Section 3 provides definitions and explanations of social networks, differential privacy, and the multi-armed bandit. Section 4 introduces the Probability Selection Algorithm Based on Multi-Armed Bandit (PSMAB), Probability Mapping (PM), PP, and four histogram publishing methods based on PM and PP. Section 5 conducts experimental verification analysis and measures the data utility of the proposed algorithms. Section 6 summarizes the paper.

2 Related Work

Data publishing using differential privacy [6] involves adding random noise to real datasets. This approach can be divided into two categories: edge differential privacy and node differential privacy. Edge differential privacy is where the two graphs are adjacent only if they differ on a single edge. On the other hand, node differential privacy is where deleting a node and all its connected edges in a graph results in a new graph that is adjacent to the original graph.

2.1 Edge Differential Privacy

To the problem of losing original information and weak privacy protection in edge differential privacy, Lv et al. [7] developed an edge removal projection algorithm based on the Triangle-count Sort Edge Removal (TSER) algorithm. This algorithm preserves more triangles in the original graph and enhances the availability of data. Zhou et al. [8] also proposed a model for generating social networks called the structure-attribute social network model. This model introduces uncertainty graphs into network partitions. However, it has high time complexity and low data validity. Huang et al. [9] proposed the Privacy-preserving approach Based on Clustering and Noise (PBCN). This approach makes the published graph resistant to degree and graph structure attacks while maintaining high execution efficiency. Gao et al. [10] proposed a differential private graph model that combines sequences dK-1, dK-2, and dK-3 and adds three levels of rewiring algorithms. This model preserves the original graph structure information to a greater extent.

However, edge differential privacy is vulnerable to attacks that involve re-identifying nodes in anonymized graph data. Besides, it is designed to protect the relationship between two entities, which is represented in the form of an edge, from being disclosed, but in social networks, a node and its associated edges represent all of a person's data, whereas an edge cannot represent all of a person's data.

2.2 Node Differential Privacy

On contrast, node differential privacy offers a greater level of privacy protection compared to edge differential privacy. In order to address the issue of large errors in algorithm under node differential privacy, several methods have been proposed. Day et al. [11] introduced π_θ, which involves adding edges while ensuring that the degree of the two connected nodes does not exceed the node degree threshold. This helps reduce errors in fitting the true degree distribution. Meanwhile, Zhang et al. [12] proposed a Sequential Edge Removal (SER) algorithm to decrease the global sensitivity of histogram publication. However, both π_θ and SER suffer from uncertainty in edge ordering, which limits their ability to preserve edge information to the fullest extent.

Liu et al. [13] put forth two algorithms for releasing the intensity distribution and analyzed the impact of projection on node intensity histograms through introspection. Ding et al. [14] suggested the Best Adaptation (BA) strategy, which involves removing the connection edge of a node that has the largest number of triangles adjacent to it. However, while this method improves data availability, it also reduces privacy protection effects. Prasad et al. [15] introduced FlowGraph (FG), which constructs a weighted graph by creating new nodes between a source node v and a sink node u. It then calculates the maximum flow from v to u, removes v from the maximum flow graph, and constructs the degree distribution. By combining the Lipschitz extension and the generalized exponential mechanism, FG greatly published a degree distribution that approximates the original graph and is more accurate than previous algorithms.

However, current node differential privacy methods often suffer from issues such as loss of original information and significant errors before and after projection. To address these challenges, this paper aims to minimize errors and preserve more of the original data while enhancing data availability and ensuring privacy protection.

3 Preliminaries

This section introduces the preliminaries to our proposed probability mapping-based privacy preservation method. They include social network graphs and differential privacy.

3.1 Social Network Graph

A social network can be represented by a graph, which is defined as $G = (V, E)$. In this graph, $V = \{v_1, v_2, \dots, v_n\}$ represents a set of users, and $E = \{e_1, e_2, \dots, e_m\}$ represents a set of relationships between users.

3.2 Differential Privacy

Differential privacy is a technique that allows for sharing information about a group of individuals, while protecting their personal privacy by obscuring their data. The following are the key concepts in contemporary differential privacy.

ε -Differential Privacy [6]. A random algorithm $M: D \rightarrow M(D)$ satisfies ε -Differential Privacy if any two neighboring datasets D and D' maintains the following relationship.

$$P_r[M(D) \in S] \leq e^{\varepsilon} \cdot P_r[M(D') \in S], \tag{1}$$

where the probability P_r is controlled by the randomness of the algorithm M, $S \in Range(D)$ and ε is a parameter for privacy level.

Global Sensitivity [16]. For any function f, given two adjacent datasets D and D' that differ at most one record, the global sensitivity of f is defined as:

$$\Delta f = \|f(D) - f(D')\|_1, \tag{2}$$

where $\|f(D) - f(D')\|_1$ is the $L1$ normal form. The higher global sensitivity is, the less data is available and more noise is added.

Laplace Mechanism. Given a dataset D, there exists a function $f: D \rightarrow R$ with sensitivity Δf. If the mechanism M satisfies ε -Differential Privacy, its output satisfies:

$$M(D) = f(D) + Lap(\frac{\Delta f}{\varepsilon}), \tag{3}$$

where $Lap(\frac{\Delta f}{\varepsilon})$ is a random noise following the Laplacian distribution.

3.3 Multi-Armed Bandit

The multi-armed bandit model has n arms. Each pull of the arm α results in a reward. The reward of each arm follows the same function r. The goal of the multi-armed bandit model is to find the arm with the most appropriate reward after t iterations.

We define the reward function r for each arm as follows:

$$r = \alpha \times L + \beta \times p + \gamma \times \Delta f, \tag{4}$$

where L represents the error of the histogram, p represents the utility of the probability, Δf represents the privacy loss, and α, β, and γ are the weight. The errors obtained by each probability are set as the reward function of each arm.

4 Proposed Method

Our Probability Mapping (PM) and Probability Projection (PP) algorithms are built upon the Probability Selection Algorithm Based on Multi-Armed Bandit (PSMAB).

They only differ in the node degree threshold that restricts the addition of edges. See Sections 4.1, 4.2 and 4.3 for the details of PSMAB, PM and PP algorithms.

The process flowchart for safeguarding social network privacy through PM or PP is illustrated in Fig. 1. We start by inputting a social network G and a node degree threshold θ. Then, we select probability p based on PSMAB. Finally, we obtain four triangle count histograms: PM-based Triangle Counting (PMTC), PP-based Triangle Counting (PPTC), PM-based Cumulative Triangle Count (PMCTC), and PP-based Cumulative Triangle Count (PPCTC) histograms, through processing by PM or PP.

Fig. 1. Flowchart of publishing algorithm of triangle count histogram in social networks based on PM or PP.

4.1 The Probability Selection Algorithm Based on Multi-Armed Bandit

A probability selection algorithm based on multi-armed bandit (PSMAB) is a method for selecting an action from a group of options that have uncertain rewards. The algorithm aims to balance the exploration of new actions that may provide greater rewards with exploiting the best known action. We came up with PSMAB inspired by [17]. The process flowchart of PSMAB is shown in Fig. 2, where r_i denotes the reward of arm i, and L_{p_i} and L_i denote the probability error and total error corresponding to r_i, respectively.

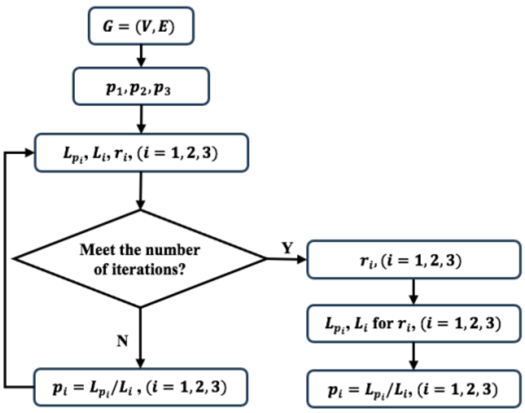


Fig. 2. Flowchart of PSMAB.

PSMAB first initializes the reward, probability, and reward function of each arm. In each iteration, PSMAB starts with the comparison between the generated random number r and an initial parameter lr. If r is less than lr, PSMAB randomly selects an arm and then updates its reward. If r is greater than lr, PSMAB selects the arm with the largest reward and updates its reward. Then, the probability error L_p and total error

L for each arm are calculated based on the reward and probability of the three arms, respectively. The ratio of L_p to L for each arm is then obtained using Eq. (5).

$$p = L_p / L, \tag{5}$$

where L_p is the probability error, and L is the total error caused by the probability, noise, and node degree threshold. p is the new probability for each arm. The three arms with the latest probabilities are carried into the next iteration. Then, p, L_p, and L of each arm are recalculated. To account for the randomness of probabilities, this process is repeated 10 iterations. Once complete, the probability with the medium value r_i is selected as the proper probability. This is due to the fact that when there is a larger error, fewer edges are retained, resulting in decreased data utility. Conversely, when there is a smaller error, more original data must be retained, which can lead to a weaker privacy protection effect.

4.2 Probability Mapping Algorithm

This section proposes a novel edge addition algorithm, named Probability Mapping (PM) that is built upon PSMAB. In the PM algorithm, the first step involves creating a new graph G' by mapping all the nodes in the original graph G. However, no edges are added at this stage. Next, PM traverses through the edge set of the original graph and generates a random number ra. ra is then compared with the probability p chosen by PSMAB. If ra is less than p, the edge $e = (u, v)$ is added, and the degrees of the nodes at each end of the edge are incremented by one. On the other hand, if ra is greater than or equal to p, no edge is added. Once the traversal is complete, the new mapped graph $G' = (V, E')$ is generated. The pseudocode of PM is shown in Algorithm 1.

Algorithm 1: PM Algorithm.

Input: An original social network $G = (V, E)$, probability p, and an edge ordering $\Lambda = < e_1, e_2, \dots, e_n >$.
Output: A mapped graph $G' = (V, E')$.

```
1: d(v) ← 0 for each v ∈ V, E' ← ϕ
2: for e = (u,v) ∈ Λ do
3:   if random number ra < p
4:     E' ← E' ∪ {e}
5:     d(u) ← d(u) + 1; d(v) ← d(v) + 1
6:   end if
7: end for
8: return G' = (V, E')
```

4.3 Probability Projection Algorithm

There are several nodes in the mapped graph with large degrees, and the range of degree values is also quite broad. This can result in increased errors in the histogram. To address this issue, we introduce the Probability Projection (PP) algorithm, which is an enhancement of the PM algorithm. PM and PP follow the same principles, with the difference being that PP imposes a stricter condition by adding a node degree threshold for adding edges. The pseudocode for PP is shown in Algorithm 2. Compared to the existing methods of π_θ [11] and FG [15], PP can reduce the sensitivity of query functions, minimize the error between the original and projected data, and preserve more original information. This reduces the risk of user privacy breaches.

The total time complexity of both Algorithm 1 and Algorithm 2 is $O\ (|E|)$, as the majority of time is consumed in the process of traversing the edges. The total space complexity of both algorithms is $O\ (|E| + |V|)$ because they both need to create new memory to save nodes and edges during the process of constructing a mapping graph and a projection graph.

Algorithm 2: PP Algorithm.

Input: An original social network $G = (V, E)$, probability p, node degree threshold θ and an edge ordering $\Lambda = < e_1, e_2, \ldots, e_n >$.
Output: A projected graph $G'' = (V, E'')$.

```
d(v) ← 0 for each v ∈ V, E'' ← φ
for e = (u,v) ∈ Λ do
    if d(u) < θ & d(v) < θ then
        if random number ra < p
            E'' ← E'' ∪ {e}
            d(u) ← d(u) + 1; d(v) ← d(v) + 1
        end if
    end if
end for
return G'' = (V, E'')
```

4.4 Publishing Algorithm for Triangle Counting Histogram

For counting triangles in social networks, we create two types of histograms: PM-based Triangle Counting histogram (PMTC) and PP-based Triangle Counting histogram (PPTC).

The process of generating PMTC involves generating a probability from PSMAB, creating a mapped graph G' from PM and generating a histogram of the corresponding triangle count. To make the histogram publishable, we add noise to each bucket. This results in a histogram that can be used to count triangles in social networks. The process of generating PPTC is similar to that of PMTC.

Theorem 1. Given two adjacent graphs G and G' that only differ by one node, the following equation holds:

$$\Delta_{PMTC} = \|PMTC(G) - PMTC(G')\|_1 < 4p + 1.$$

Proof: Suppose the graphs $G = (V, E)$ and $G' = (V, E')$ differ by only one node v'. We refer to the set of all triangles that exists solely in G' as T, and the number of triangles in this set as m. All triangles in the set T have a common node v' at least. Removing the node v' from the graph G' effectively removes all triangles from the set T. Every triangle in T change the triangle count result of the other two nodes, both nodes have a difference of 1. Because each bucket in the triangle count histogram result represents the number of nodes corresponding to the number of triangles.

In the worst-case scenario, all triangles in the set T have only one common node v', which means that the number of nodes influenced by triangles in the set T is at most

$2m$. Every node under the influence of the set T will cause a difference of at most 2 in the histogram. The difference, caused by the removal of the node v', in the histogram is 1. Therefore, the difference between the graphs $G = (V, E)$ and $G' = (V, E')$ is $4m + 1$.

In PMTC, the number (i.e., m) of the triangles in the set T is determined by the ratio of the probability p to the value range of a random number. As a random number range between 0 and 1, the ratio between p and the random number also falls in the range of [0,1]. Thus, Δ_{PMTC} is always less than $4p + 1$.

Theorem 2. Given two adjacent graphs G and G' that only differ by one node, the following equation holds:

$$\Delta_{PPTC} = \|PPTC(G) - PPTC(G')\|_1 < 4p\theta + 1.$$

Proof: The proof procedure is similar to Theorem 1. The difference is that a node degree threshold θ is set in PPTC to limit the addition of edges, and the difference caused in the histogram is $4p\theta + 1$ at most.

4.5 Publishing Algorithm for Cumulative Triangle Counting Histogram

To minimize noise-caused errors, this section proposes two enhanced publishing algorithms: PM-based Cumulative Triangle Count histogram (PMCTC) and PP-based Cumulative Triangle Count histogram (PPCTC).

Theorem 3. Given two adjacent graphs G and G' that only differ by one node, the following equation holds:

$$\Delta_{PMCTC} = \|PMCTC(G) - PMCTC(G')\|_1 < 2p + 1.$$

Proof: Using the notations in Theorem 1. Suppose the node v' connects to m triangles, then removing it from G' will cause a shift in all the bins of the cumulative histogram by 1. The maximum number of nodes affected by deleting the node v' is 2m. In the cumulative histogram, the difference of histogram caused by every $2m$ nodes is 1, which means that the total difference between the graphs $G = (V, E)$ and $G' = (V, E')$ in the cumulative histogram is $2m + 1$.

In PMCTC, the probability p is used to decide whether to add or delete an edge. The value of m is determined by the ratio of p to the value range of a random number. As the random numbers range between 0 and 1, the ratio between p and the random number also falls in the range of [0,1]. This means that Δ_{PMCTC} is always less than $2p + 1$.

Theorem 4. Given two adjacent graphs G and G' that only differ by one node, the following equation holds:

$$\Delta_{PPCTC} = \|PPCTC(G) - PPCTC(G')\|_1 < 2p\theta + 1.$$

Proof: The proof procedure is similar to Theorem 2 and Theorem 3, and the difference caused by PPCTC is $2p\theta + 1$ at most.

The probability p takes a value ranging from $[0,1]$. Taking the probability value into Theorems 1-4 shows that the algorithms proposed in this paper reduce the sensitivity of the query functions compared with π_θ [11] and FG [15]. The sensitivity of π_θ [11] is $2\theta + 1$, and the sensitivity of FG [15] is 6θ.

5 Experimental Results and Analysis

5.1 Datasets

Three real-world datasets from [18], as shown in Table 1, were used in our experiments. Tri_Num is the number of node triangles in the dataset, and Max_degree is the maximum degree of nodes in the dataset. Each network dataset was pre-processed and converted into an undirected graph.

Table 1. Information of the datasets.

Graph	$\|V\|$	$\|E\|$	Tri_Num	Max_degree
Facebook	4,039	88,234	1,612,010	1,045
Email-Enron	36,692	183,831	727,044	1,383
Twitter	75,879	1,768,149	1,768,149	81,306

The initial range of probability for PSMAB is set to $[0,1]$. However, when the probability ranges from [0,0.5], it retains little of the original graph information, which significantly affects the availability of data. On the other hand, when the probability is [0.9,1], too much original graph information is retained, resulting in too much noise. To balance between the availability of data and retaining original graph information, the multi-armed bandit model has three arms that correspond to different probabilities: $p = 0.6$, $p = 0.7$, and $p = 0.8$. The selection of the specific probability is performed by iterations based on PSMAB depicted in Fig. 2.

5.2 Performance Indicators

$L1$ *error*, *edge retention rate*, and KS *distance* are used to evaluate the experimental results, and their definitions are as follows:

- $\boldsymbol{L1\ error}$: The $L1$ *error* shows the difference between the two histograms obtained before and after PM or PP algorithm. The $L1$ *error* is defined in Eq. (6):

$$L = \sum_{i=1}^{n} |f(x_i)' - f(x_i)| , \tag{6}$$

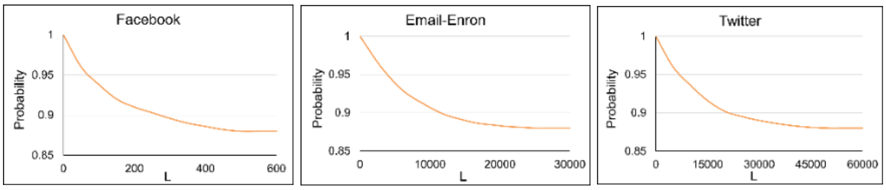
  where $f(x_i)'$ represents the frequency of each degree value after any one of the four algorithms is processed in this paper, and $f(x_i)$ represents the frequency of each degree value in the original social network graph.
- ***Edge retention rate***: It can be expressed as $(|E'|/|E|)$, where E' represents the edge in the graph after projection, and E represents the edge in the original graph.
- $\boldsymbol{KS\ distance}$: The smaller the $KS\ distance$, the more similar the histogram after noise is added to the original histogram, and the higher the data availability. The $KS\ distance$ is defined in Eq. (7):

$$K = max|f(x)' - f(x)| , \tag{7}$$

  where $f(x)'$ represents the histogram data distribution after adding noise, and $f(x)$ represents the histogram data distribution of the original social network.

5.3 Experimental Results and Analysis

The Effect of Error on Probability Selection. We explored the effect of total error on probability in PSMAB by setting a node degree threshold of θ = 128. Considering the randomness of the probability and Laplacian noise, the averaged results shown in Fig. 3 were obtained from 100 trials.

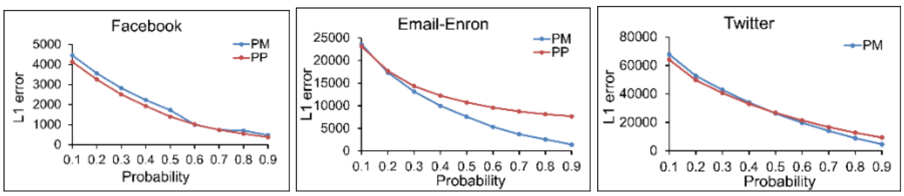


(a) Result on Facebook. (b) Result on Email-Enron. (c) Result on Twitter.

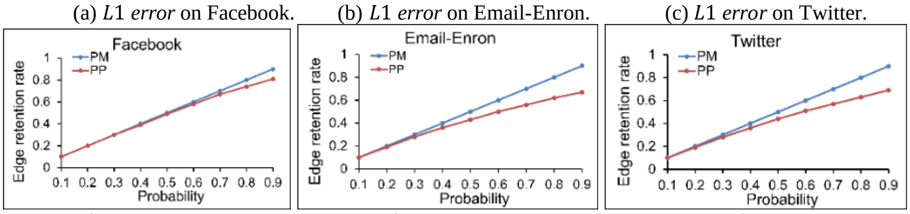
Fig. 3. The effect of different errors on probability selection.

It can be seen from Fig. 3 that the probability tends to stabilize at a constant value of 0.88 as the error increases across the three datasets. The reason behind this phenomenon can be attributed to the fact that PSMAB algorithm restricts the addition of edges based on probability and node degree threshold, while ignoring the dataset size. PSMAB takes into account the balance between preserving privacy and maximizing data utility, hence it retains more data information while minimizing the error rate.

Comparison Between PM and PP. We conducted a study to compare PM and PP by setting a node degree threshold of $\theta = 128$. We compared the $L1$ *errors* and the *edge retention rates* ($|E'|/|E|$) of PM and PP against the various probabilities, respectively across three datasets. The averaged results shown in Fig. 4 were obtained from 100 trials.



(a) $L1$ *error* on Facebook. (b) $L1$ *error* on Email-Enron. (c) $L1$ *error* on Twitter.



(d) $|E'|/|E|$ on Facebook. (e) $|E'|/|E|$ on Email-Enron. (f) $|E'|/|E|$ on Twitter.

Fig. 4. Comparison between PM and PP under different probabilities across 3 datasets.

As can be seen in Fig. 4, the $L1$ *errors* of both PM and PP show a decreasing trend as the probability increases, while the $|E'|/|E|$ show an increasing trend. PP has a smaller $L1$ *error* than PM on smaller datasets (e.g., Facebook) and a higher $L1$ *error* on larger datasets (e.g., Email-Enron and Twitter). However, the $|E'|/|E|$ of PM is higher than that of the PP on any dataset. Because PP has stricter restrictions on adding edges compared with PM, and the relationship between nodes and edges is more intensive in a large social network, deleting edges is more likely to lead to a larger $L1$ *error*.

In summary, PM is more suitable for large datasets with complex relationships, while PP is more suitable for small datasets with simpler relationships.

Comparison with Other Projection Algorithms. Comparing PP with π_θ [11] and FG [15], two metrics (i.e., the $L1$ *error* of the node degree histogram and the $|E'|/|E|$ before and after projection) are used. In this experiment, a node degree threshold of θ = 128 was set. Considering the randomness of the probability and Laplacian noise, the averaged results shown in Tables 2 and 3 were obtained from 100 trials.

Table 2. Comparison of $L1$ *error* of our methods with other projection methods.

Dataset	**FG**	$\boldsymbol{\pi_\theta}$	**PP**
Facebook	1092	801	611
Email-Enron	13602	12577	7126
Twitter	17116	15293	12623

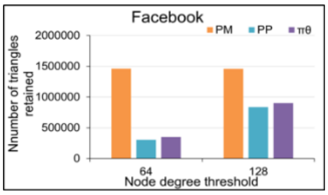
Table 3. Comparison of the edge retention rate of our methods with other projection methods.

Dataset	**FG**	$\boldsymbol{\pi_\theta}$	**PP**
Facebook	0.90	0.88	0.86
Email-Enron	0.75	0.74	0.71
Twitter	0.77	0.74	0.74

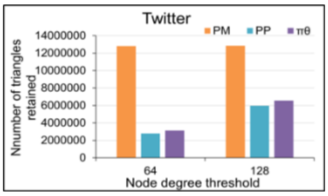
The experimental results in Table 2 show that PP has a smaller $L1$ *error* than the other three methods, which implies that the error caused by PP is smaller and a better distribution shape is maintained. Moreover, based on Theorems 2 and 4, it can be inferred that PP reduces the sensitivity of the query function, which reduces the risk of user privacy leakages. The stricter the conditions for adding edges, the lower the edge retention rate, but the more edge information is retained. Our purpose is to reduce the sensitivity of the query function and the error before and after projection while ensuring that more edge information is retained.

In conclusion, PP not only significantly reduces the $L1$ *error* of the degree histogram and the sensitivity of the query function on each dataset, but also retains more edge information, effectively mitigating the risk of user privacy leakage.

Comparison of Triangle Retention Numbers. This section compares the numbers of triangles retained by PMTC and PPTC with that of π_θ [11]. On different datasets, we set different node degree thresholds. Considering the randomness of the probability, the averaged results shown in Fig. 5 were obtained from 100 trials.



(a) Result on Facebook.

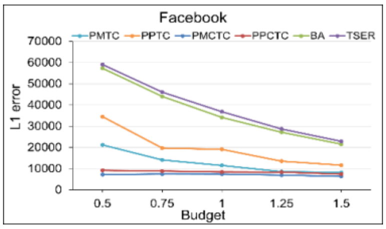


(b) Result on Twitter.

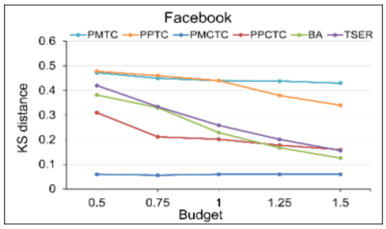
Fig. 5. Comparison of number of triangles retained.

It can be seen from Fig. 5 that PM retains more triangles than other algorithms. Besides, the number of triangles retained by PP and π_θ is increasing as the node degree threshold increases. Because in PM, the probability is the basis for adding edges, and the node degree threshold does not affect the results of PM, while the results of other algorithms are affected by the node degree threshold. The larger the node degree threshold, the more edges are added. In addition, the number of triangles retained by PP is less than that of π_θ. An explanation is that PP is limited by probability and node degree threshold, while π_θ is limited only by node degree threshold.

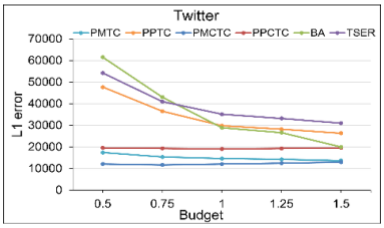
Comparison with Other Publishing Algorithms. PMTC, PPTC, PMCTC, and PPCTC are compared with TSER [7] and BA [14] on the Facebook and Twitter datasets. The results are shown in Fig. 6.



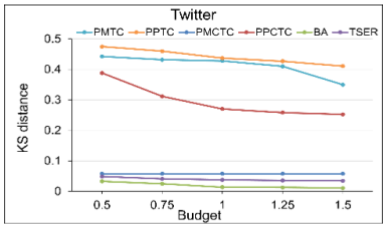
(a) $L1$ *error* on Facebook.



(b) KS *distance* on Facebook.



(c) $L1$ *error* on Twitter.



(d) KS *distance* on Twitter.

Fig. 6. Cumulative histogram comparison of $L1$ *error* and KS *distance*.

Fig. 6 shows that the $L1$ *error* of the four algorithms proposed is better than BA and TSER. As the privacy budget increases, the $L1$ *error* of the four methods shows a decreasing trend. Due to the randomness of probability, the KS *distance* generated by the four algorithms may produce the same result under different privacy budgets. As the privacy budget increases, the added noise decreases, the data availability increases and more original graph information is preserved. It can be seen from Fig. 6 on the smaller dataset (Facebook) that the $KS\ distance$ of PPCTC is slightly higher than BA when the budget is greater than 1.25. Because BA retains more edges while maintaining a data distribution more similar to the original social network. However, PPCTC does not take into account the number of node triangles connected to the edges in the process of adding the edges. While on the larger dataset (Twitter), the $KS\ distance$ of PPCTC and PMCTC is higher than BA and TSER. Because in PPCTC and PMCTC, the probability is used as a constraint for adding edges, and the probability is random. In addition, the $L1\ error$ and $KS\ distance$ of PPCTC are higher than PMCTC on any dataset. Because PPCTC has stricter restrictions on adding edges.

Comparing PMTC, PPTC, PMCTC and PPCTC, Fig. 6 shows that the $L1\ error$ and $KS\ distance$ of PMTC and PPTC are higher than PMCTC and PPCTC. Because from Theorems 1-4 the sensitivity of PMTC and PPTC is higher than that of PMCTC and PPCTC, leading to the addition of more noise and a larger error in the histogram.

6 Conclusions

In this paper, we have presented the probability selection technology based on PSMAB. We have also developed PM and PP to anonymize triangles in large social networks using node differential privacy. Based on PM and PP, we have built four methods for displaying triangles. These include the PMTC and PPTC histogram publishing algorithms, as well as the cumulative PMCTC and PPCTC histogram publishing algorithms. Extensive experiments were conducted to validate the probability effects of PM and PP. The experimental results show PP algorithm achieves a smaller $L1\ error$ rate than those of the existing algorithms. Furthermore, PMCTC and PPCTC have higher data usage and lower global sensitivity than other algorithms. However, it should be noted that our research focuses on data from static social networks. In real-world scenarios, social networks change dynamically in real time. Consequently, the application of these algorithms to dynamic social networks is a key field of future research.

Acknowledgement. This research was funded by NSFC under Grant 61572170, Natural Science Foundation of Hebei Province under Grant F2021205004, Science Foundation of Returned Overseas of Hebei Province Under Grant C2020342, and Key Science Foundation of Hebei Education Department under Grant ZD2021062.

References

1. Sweeney, L.: K-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems **10**(05), 557-570 (2002). https://doi.org/10.1142/S0218488502001648
2. Machanavajjhala, A., Gehrke, J., Kifer, D., Muthuramakrishnan, V.: l-diversity: privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) **1**(1), 1-52 (2007). https://doi.org/10.1145/1217299.1217302
3. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: privacy beyond k-anonymity and l-diversity. 2007 IEEE 23rd International Conference on Data Engineering, 106-115 (2006)

4. Xiao, X., Tao, Y.: M-invariance: towards privacy preserving re-publication of dynamic datasets. Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 689-700 (2007)
5. Jiang, H., Pei, J., Yu, D., Yu, J., Gong, B., Cheng, X.: Applications of differential privacy in social network analysis: A survey. IEEE Transactions on Knowledge and Data Engineering **35**(1), 108-127 (2021). https://doi.org/10.1109/TKDE.2021.3073062
6. Dwork, C.: Differential privacy. International colloquium on automata, languages, and programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 1-12 (2006)
7. Lv, T., Li, H., Tang, Z., Fu, F., Cao, J., Zhang, J.: Publishing triangle counting histogram in social networks based on differential privacy. Security and Communication Networks **2021,** 1-16 (2021). https://doi.org/10.1155/2021/7206179
8. Zhou, N., Long, S., Liu, H.: Structure-attribute social network graph data publishing satisfying differential privacy. Symmetry **14**(12), 2531-2541 (2022).
9. Huang, H., Zhang, D., Xiao, F., Wang, K., Gu, J., Wang, R.: Privacy-preserving approach PBCN in social network with differential privacy. IEEE Transactions on Network and Service Management **17**(2), 931-945 (2020). https://doi.org/10.1109/TNSM.2020.2982555
10. Gao, T., Li, F.: Protecting social network with differential privacy under novel graph model. IEEE Access **8**(23), 185276-185289 (2020).
11. Day, W. Y., Li, N., Min, L.: Publishing graph degree distribution with node differential privacy. In: Proceedings of the 2016 International Conference on Management of Data, pp. 123-138. (2016)
12. Zhang, Y., Wei, J., Li, J.: Graph degree histogram publication method with node-differential privacy. Journal of Computer Research and Development **56**(3), 508-520 (2019). https://doi.org/10.7544/issn1000-1239.2019.20170886
13. Liu, G., Ma, X., Li, W.: Publishing node strength distribution with node differential privacy. IEEE Access **8**(23), 217642-217650 (2020)
14. Ding, X., Sheng, S., Zhou, H., Zhang, X., Bao, Z., Zhou, P., Hai, J.: Differential private triangle counting in large graphs. IEEE Transactions on Knowledge and Data Engineering **34**(11) 5278-5292 (2021). https://doi.org/10.1109/TKDE.2021.3052827
15. Sofya, R., Adam, S.: Efficient Lipschitz extensions for high-dimensional graph statistics and node private degree distributions. arXiv preprint arXiv:1504.07912, (2015)
16. Wu, X., Zhang, Y., Shi, M., Li, P., Li, R., Xiong, N.: An adaptive federated learning scheme with differential privacy preserving. Future Generation Computer Systems **127**(1), 362-372 (2022). https://doi.org/10.1016/j.future.2021.09.015
17. Odeyomi, Olusola T.: Differential privacy in social networks using multi-armed bandit. IEEE Access **8**, 11817-11829 (2022)
18. Jure, L., Andrej, K., http://snap.stanford.edu/data, 2014/06.