Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 143 (2018) 786–793

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Advances in Computing and Communication (ICACC-2018)

Node Differential Privacy in Social Graph Degree Publishing

Kamalkumar R. Macwan[a], Sankita J. Patel[b]

[a]*PhD Scholar, Sardar Vallabhbhai National Institute of Technology , Surat-395007, Gujarat, India*
[b]*Assistant Professor, Sardar Vallabhbhai National Institute of Technology , Surat-395007, Gujarat, India*

Abstract

Effective analysis of social network dataset has gained attention in various fields. The published dataset is often limited by the privacy of individual user whose data is included in it. Existing anonymization methods to publish the entire dataset or the approaches to release specific results may not provide user privacy. But, the differential privacy (DP) technique provides strong guarantees about the privacy of the participants in a released database. For the social graph data, differential privacy is divided into two parts: edge and node differential privacy. Edge-DP protects the relationship between two nodes represented as an edge whereas node-DP provide privacy to all edges connected to a node. For node-DP, we have proposed projection approach to restrict degree of a node below a certain value, to reduce the sensitivity of user related queries. The existing graph projection techniques lose much useful information. Our proposed projection approach retains more information by accessing node according to their increasing degree value. The degree distribution of social dataset under node-DP is generated to release the information regarding user's degree.

Keywords: Differential Privacy ; Degree Distribution ; Node-DP; Social Network Data Publishing

1. Introduction

Social network data publishing has enabled researchers to utilize the data in various fields. The statistics from social dataset also contain useful information, such as number of connections, public opinion poll, or number of active users in any community/group. Unfortunately, there is a trade-off between publishing the original dataset to the research community and the privacy of the users whose data is present in that dataset. Researchers often spread more data that contain useful information than their expectation[1]. Then the question arise, how the social graph should be shared publicly for analysis purpose without compromising user privacy? One way to do so is to apply stronger anonymization techniques [2, 3] to modify the original social graph structure in precise ways that have more user privacy but retain the utility of the modified graph structure. An adversary can deanonymize published dataset

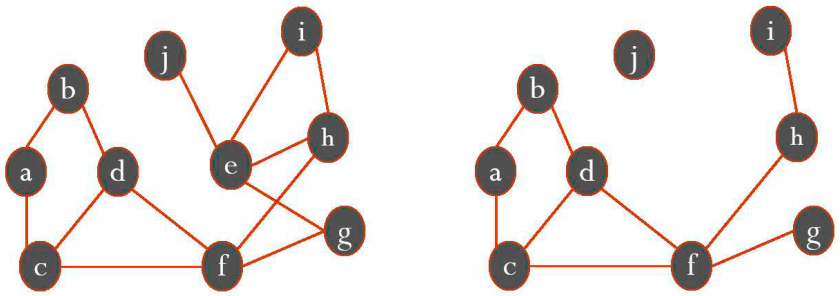
E-mail address: kamal.macwan@yahoo.com

1877-0509
10.1016/j.procs.2018.10.388

using public information [1, 4]. In recent years, the differential privacy approach[5] has gradually become the standard privacy definition in research on private data publishing. The main idea behind of this technique is to inject randomized noise into query results to hide the impact of individual user of the published dataset.

1.1. Social Network Data Publishing

sensitive. Let's assume that the analysts are interested in result of counting queries. The returned answer is generic enough not to reveal some sensitive information of any user. Now, after removing a user's data, the obtained result from the same query may reveal some sensitive information of the user whose data is removed. Thus, it fails to safeguard the sensitive information of the participants of the published dataset. For example, for given social network in fig. 1(a), the result of the query "How many user have 4 connections?"is 2. But, the same query produce answer as 1 for node neighbouring graph G'shown in fig. 1(b). Although these results do not reveal any user identity, it produce the useful information that the removed node(user) has 4 connections.



(a) Original Social Graph G (b) Node Neighbouring Graph G'

Fig. 1. Published Social Network Graph

A simple solution could be to inject some random noise to the actual result. The result may be distributed according to the Laplace distribution to have result near to the true value. This way the analyst receive the useful information while still having some uncertainty about it. Even though same query is applied more than one time, the generated answer is slightly differing because of the random noise being added to the true result. Thus, it might help them to deduce much closer value to the actual result but it helps to provide privacy to the participants.

1.2. ϵ-Differential Privacy

ϵ-Differential privacy depends on some query and result perturbation to provide privacy guarantee. This can be achieved in differential privacy by adding some random noise to the query output. This is realized by using the methods such as Laplace distribution and the normal distribution with variance depending on ϵ and the query's sensitivity. The notion of ϵ-differential privacy [5] is defined based on the concept of neighbouring dataset.

Definition 1. *"A randomized algorithm A satisfies ϵ-differential privacy(ϵ-DP) when for any two neighbouring databases, D and D′, denoted by $D \backsimeq D'$,*

$Pr[A(D) \in S] \leqslant exp(\epsilon) \times Pr[A(D') \in S]$

where $S \subseteq Range(A)$ and ϵ is a parameter for privacy level. "

The output generated from differentially private algorithm for neighbouring datasets does not have significant change[5]. The differential privacy for social graph data is divided into two variants: in edge differential privacy(edge-DP), two graphs are neighbouring if they differ on a single edge; in node differential privacy(node-DP), two graphs are neighbouring if they differ on a node and all edges incident to that node.

1.3. Challenges

Privacy preserving social network data publishing is concerned with social network data publishing while preserving the social network users' privacy. An adversary can utilize different graph structural properties to carry out privacy related attacks. The main challenge is to guarantee DP in large dataset. Graph projection method should retain more number of edges while restricting the maximum degree below some threshold value. It is also challenging to maintain usefulness of responses for social network queries while guaranteeing privacy. Moreover, existing notions of privacy require too much noise to be added to true answers and can hardly produce responses that are useful. By focusing on specific query, we have the chance to guarantee privacy with much less noise.

1.4. Contribution

In this proposed work, we investigate the problem of publishing the degree distribution of a graph under node-DP. The goal of this work is to publish node degree histogram that approximates the true distribution of G as much as possible while satisfying node-DP. Graph projection method is applied to restrict the maximum degree of node to some threshold. The proposed graph projection method, "ordered edge insertion", retains more edges compared to other existing techniques. The projected graph is minimal and has sensitivity of $2\theta+1$ to publish the degree histogram. The objective is to retain edges of the nodes having lower degree first and then to proceeds for higher degree nodes in order to achieve θ-bounded graph.

2. Related Work

The differential privacy[5] was first considered for tabular data. The databases contains array or sets where each entry corresponds to an individual's information. The individual data may be modified without affecting other entries. In order to provide user privacy, differentially private function is used to inject Laplace noise of the possible maximum change in the function. The differential privacy technique has been also applied in network trace anonymization [6], and data compression techniques [7]. Other work focused specifically on applying differential privacy to simple graph structures such as degree distributions [8]. The same technique is extended to the social network data represented as graph, where node represents the user/community and an edge represents the relationship between them. It is used to count the number of edges, degree of nodes and occurrences of small subgraph.

2.1. Preliminaries

We consider a social network $G = (V, E)$ as a simple, undirected graph, contains no multiple edges, where V is a set of vertices representing individual users and $E \subseteq V \times V$ is the set of edges representing the relationship between individuals.

Definition 2. *(Node Distance) "For given graph G and G′, the node distance $d_{node}(G, G')$ is defined as the minimum number of nodes in G′ that need to be changed to obtain G. "*

The modification of graph contains insertion or removal of nodes with set of edges connected to those nodes or just change in their adjacency list. So, a change to one node can modify the adjacency list of other nodes. The node distance is defined in eq. 1, where k is the number of nodes in the largest induced subgraph of G which is equal to the corresponding induced subgraph of G′.

$$d_{\text{node}}(G, G') = max\{|V_G|, |V_{G'}|\} - k \tag{1}$$

Definition 3. *(Node Neighbours) "Given graph G and G′ are called as node neighbours if $d_{node}(G, G')$=1. "*

Similarly, the edge distance $d_{\text{edge}}(G, G')$ is also defined as the minimum number of edges in G′ that need to be changed to obtain G. Nissim[9] proposed a local measure of sensitivity as Local sensitivity.

Definition 4. *(Local Sensitivity) "For a function $f : G_n \rightarrow R$ and a graph $G \in G_n$, the local sensitivity of function f at G is defined in eq. 2, where the maximum is taken over all node neighbours G' of G and G_n represents the set of graph on at most n nodes. "*

$$LS_f(G) = max_{G'} \parallel f(G) - f(G') \parallel_1 \tag{2}$$

The global sensitivity is defined as $\triangle f = max_G LS_f(G)$. Now, the most basic framework to achieve differential privacy, Laplace noise is sealed according to the global sensitivity of the desired statistic f. It is done by just adding Laplace noise to each entry of f.

Definition 5. *(Laplace Mechanism[8]) "The algorithm $A(G) = f(G) + Lap(\triangle f/\epsilon)^p$ is ϵ-node-private. "*

2.2. Node Differential Privacy

A privatized query Q satisfies node-DP if it satisfies the differential privacy for all the pairs of graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ where $V_2 = V_1 - x$ and $E_2 = E_1 - (v_1, v_2)$ where $v_1 = x$ OR $v_2 = x$ for some $x \in V_1$. In node privacy, if the true world is a given social network G, the neighbouring possible worlds are ones in which an arbitrary node, and all edges connected to it, are removed from or added to G. This privacy guarantee to protect all the individuals (participants and their connections). An attacker in possession of generated result R will not be able to determine whether a person x appears in the population at all.

Gehrke[10] suggest a stronger privacy model named as zero knowledge privacy. The method demonstrate that the privacy can be achieved for several tasks in extremely dense graphs. As the technique used for computation in small subgraph of a larger graph, it is not directly applicable to sparse graph. Blocki[11] also consider node-DP algorithms to analyse sparse graphs. They made assumption regarding the beliefs about the dataset that a querier may have to quantify over a restricted class of datasets. Existing works[12, 13] have considered publishing the degree sequence or distribution and extended the technique to synthetic generation . But, these approaches become impractical for node-DP, where the sensitivity of *hist(G)* becomes 2(|V|-1). The common method to provide node-DP is to add noise to the output query. The noise function depends on the maximum possible change in query result due to addition or removal of individual user. So, satisfying node-DP is much harder as removal of one node may cause removal of |V|-1 edges, where V is the set of all nodes. In order to lower this change, a graph projection method is considered as key technique for node-DP.

2.3. Graph Projection Methods

The graph projection method modify a graph to be θ-degree bounded G^θ, in which maximum degree is not more than θ [14]. As this projection operation results into change of degree of nodes , retaining more information is important factor. The main challenge in graph projection method is to retain as much information as possible while bounding maximum degree to θ. Table 1 list out the existing graph projection methods and their properties. The existing approaches fail to retain more number of edges in projection result. Edge selection process for insertion/removal of edge for graph projection method is crucial.

3. The Proposed Graph Projection Method

As the high sensitivity for satisfying node-DP is big challenge, the graph projection step manages to make it low. Removing one node may affect other |V|-1 nodes in the graph, where V is the set of nodes. This sensitivity can be restricted to some value by keeping threshold value for the degree of nodes. Such transformed graph is known as

Table 1. Existing graph projection methods

Projection Method	Approach	Limitation
Truncation method[15]	Remove all nodes with degree larger than θ.	It removes more edges than necessary and has maximum change of 2θ+1 in query result.
Edge Removal[11]	Traverse the edges in arbitrary order and remove each edge that is connected to node that has degree > θ.	It fails to retain more number of edges and have maximum change of $p(2\theta)^{(p-1)}$ in subgraph query result, where p is the number of nodes in subgraph.
Edge addition[14]	It performs edge addition operation according to stable order of edges.	The number of retain edges is not possbile maximum value.

θ-bounded graph. In order to achieve θ-bounded graph, the degree should be reduced to θ for the nodes having degree higher than θ. The original node set V is divided into two different sets based on their degree and threshold value θ for projection:

- $V_L : v \in V$ where degree$(v) \leq \theta$
- $V_R : v \in V$ where degree$(v) > \theta$

In our proposed work, we initialize each node with degree as 0. Edge insertion operation is performed to connect these nodes to increase their degree to value θ. Now, to retain the same degree for the nodes of set V_L, we start the edge insertion operation from the nodes having lower degree. The neighbouring node having minimum degree is considered as candidate node for edge insertion operation. This helps to retain edge-set for lower degree nodes. The edge insertion is performed only if both the connecting nodes having degree < θ.

Algorithm 1 Projection by ordered edge-insertion

$E_\theta \longleftarrow \emptyset$
$d(v) \longleftarrow 0, \forall v \in V$
$V'[\,]$ = sorted node-list based on their degree
for each $v \in V'$ **do**
  if $d[v] < \theta$ **then**
    select $u \in$ neigh(v) where $d[u] =$ *minimum* and $d[u] < \theta$
    $E_\theta = E_\theta \cup (u, v)$
    $d[u]$++
    $d[v]$++
  end if
end for
return $G_\theta = (V, E_\theta)$

Algorithm 1 shows the steps for projection operation by ordered edge-insertion approach. For given original graph $G = (V, E)$, the sorting operation for nodes takes $O(|V|^2)$ time. The candidate node selection operation takes $O(|V|)$ time for each node. So, the total time complexity of the proposed projection method is $O(|V|^2)$. The generated projection result from algorithm 1 is maximal in the sense that no more edges can be inserted without increasing degree of some nodes above θ.

4. Experimental Analysis

In this section, we present our experimental study on real dataset to evaluate the performance of the proposed graph projection method. We have also compared the effectiveness of our graph projection method on differentially private result. The experiments are conducted on an Intel Core, 2 Quad CPU, 3.20 GHz machine with 4GB main memory running Windows 7 OS. The various graph projection approaches are implemented in Python programming language. We have compared practical results of our proposed projection approach with existing projection methods.

4.1. Utility Measurement

The graph projection method converts the original graph $G(V, E)$ into θ-bounded graph $G^{\theta}(V^{\theta}, E^{\theta})$. In order to evaluate the effectiveness of the projection method, we use the distance measurement between true distribution and the published distribution. Several metrics are listed here [8, 14, 15] :

- **L1 Distance :** The L1 distance between any two distribution d_1 and d_2 can be computed as:

$$\| d_1 - d_2 \|_1 = \sum_{i=1}^{n} | d_1(i) - d_2(i) | \tag{3}$$

  where, n = max(length(d_1),length(d_2)). The distribution having length less than n is padded with 0 for comparison.
- **Preserved Edge Ratio (PER) :** It is the ratio of number of edges preserved in the θ-bounded graph to the number of edges in the original graph. For given original graph $G(V, E)$ and projected graph $G^{\theta}(V^{\theta}, E^{\theta})$, it is defined as:

$$PER = \frac{| E^{\theta} |}{| E |} \tag{4}$$

- **Kolmogorov Smirnov(KS) Distance :** It compares the cumulative distribution function(CDF) for two different distribution. The KS-Distance between two cumulative distributions is used to test the closeness between them. It is defined in equation 5, where, CDF_{d1} and CDF_{d2} are two different cumulative distribution of d_1 and d_2 distribution respectively.

$$KS(d_1, d_2) = max_i | CDF_{\mathrm{d1}}(i)) - CDF_{\mathrm{d2}}(i) | \tag{5}$$

4.2. Datasets

We conduct our experiments on two datasets: Facebook and Loc-Brightkite. Both datasets contains undirected graphs without self-loop and multiple edges. These datasets are available at network repository [16].

1. **Facebook :** This social network data was collected from survey participants using Facebook app. This dataset contains 4039 nodes and 88234 edges.
2. **Loc-Brightkite :** It contains the check-in location information shared by the users. This friendship network contains 58,228 nodes and 214,078 edges.

4.3. Evaluating Projection Method

The proposed projection approach is compared with three existing graph projection methods: **T** (Truncation) [15] , **ER** (Edge Removal)[11], **Seq** (Sequential Edge Addition) [14]. Table 2 and table 3 shows the results of L1 Distance and PER comparison of degree distribution results for various projection methods. The experiment is performed for different values of θ but keeping ϵ as 0.5 only. The column **Pro**(Proposed Approach) contains the obtained results from our proposed "Ordered edge insertion" projection method.

Table 2. L1 Distance and PER comparison(Facebook dataset)

	L1 Distance				PER			
θ	**T**	**ER**	**Seq**	**Pro**	**T**	**ER**	**Seq**	**Pro**
10	4080	7159	3619	**3236**	0.0092	0.0092	0.1825	**0.1998**
25	3630	5625	2496	**2310**	0.0636	0.0636	0.3736	**0.4076**
50	2471	3615	1538	**1538**	0.1953	0.1952	0.5787	**0.6191**
100	1309	1790	1006	**1004**	0.4579	0.4579	0.7994	**0.8351**
200	863	903	671	746	0.8766	0.8766	0.9646	**0.9656**

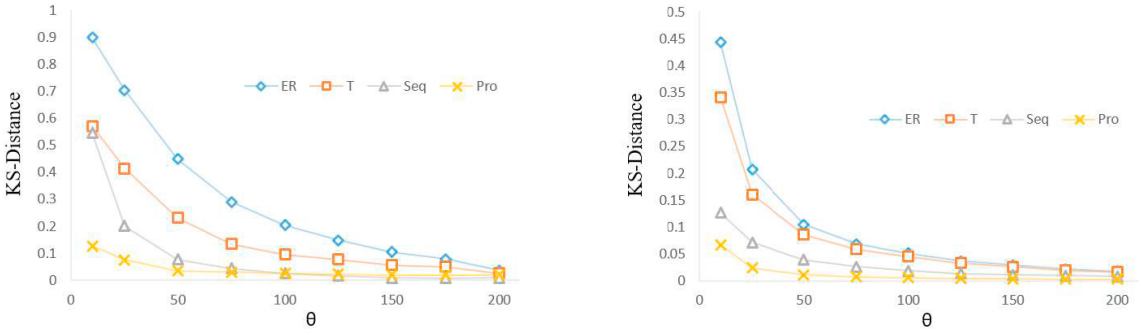
Table 3. L1 Distance and PER comparison (Loc-Brightkite dataset)

	L1 Distance				PER			
θ	**T**	**ER**	**Seq**	**Pro**	**T**	**ER**	**Seq**	**Pro**
10	41701	50695	15550	**12434**	0.1395	0.1395	0.4044	**0.4540**
25	20704	23965	7260	**4900**	0.3303	0.3303	0.6097	**0.6451**
50	10492	12161	3454	**2410**	0.5241	0.5241	0.7530	**0.7756**
100	5682	6085	1678	**1403**	0.7050	0.7050	0.8691	**0.8826**
200	2641	2730	1098	**1042**	0.8698	0.8697	0.9497	**0.9520**

Table 2 shows that Sequential and our proposed approach have the lowest L1 distance compared to other two methods for Facebook dataset ($|V|$=4039, $|E|$=88234). Lower L1 distance represent that it maintain the shape of the degree distribution after the projection. On the other hand, our proposed approach preserve the most number of edges, has greater PER value compared to all three approaches. The experimental result for Loc-Brightkite dataset ($|V|$ =58228, $|E|$=214078) is shown in table 3. Here, for any value of θ, our proposed approach have lowest L1 distance value and highest PER value. So, our proposed approach preserves the shape of the distribution and number of edges for large dataset too.

4.4. Evaluating θ-Cumulative Histogram

Here, we have compared the Node-DP dataset for various graph projection results. The degree distribution is converted into θ-Cumulative Histogram to publish the useful results. Figures 2(a) and 2(b) show the θ-cumulative distribution on node-DP results for various graph projection methods. The KS-distance is used to test the closeness of resultant cumulative histogram to the original one. Our proposed approach have significant better results compared to other existing methods for small value of θ. The edge removal and truncation approach are not able to produce maximal projected graph. The edge insertion operation in our proposed approach depends on node degree only, irrespective of edge ordering.



(a) Facebook dataset (b) Loc-Brightkite dataset

Fig. 2. Comparison of θ-Cumulative results for Node-DP

5. Conclusion

The proposed 'ordered edge insertion" projection method preserves as much information of a graph as possible and produce maximal graph. It helps to limit the sensitivity in order to apply Laplace noise on the θ-cumulative histogram. The experimental results show that our proposed approach have significant improvement over the existing approaches. The edge preservation rate for our proposed approach is almost 4% higher than sequential edge insertion method for any value of θ. The resultant θ-cumulative histogram for our proposed approach is also closer than other techniques.

References

[1] A. Narayanan and V. Shmatikov. (2008) "Robust de-anonymization of large sparse datasets," in *Security and Privacy, IEEE Symposium on. : 111–125.*

[2] K. Liu and E. Terzi. (2008) "Towards identity anonymization on graphs," in Proceedings of the ACM SIGMOD international conference on Management of data: 93–106.

[3] K. R. Macwan and S. J. Patel. (2017), "k-degree anonymity model for social network data publishing," Advances in Electrical and Computer Engineering **17** *(4): 117-124.*

[4] L. Backstrom, C. Dwork, and J. Kleinberg. (2007) "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," Proceedings of the 16th international conference on World Wide Web, ACM: 181–190.

[5] C. Dwork. (2008) "Differential privacy: A survey of results," International Conference on Theory and Applications of Models of Computation, Springer: 1–19.

[6] F. McSherry and R. Mahajan. (2010) "Differentially-private network trace analysis," ACM SIGCOMM Computer Communication Review **40** *(4): 123–134.*

[7] X. Xiao, G. Wang, and J. Gehrke, (2011) "Differential privacy via wavelet transforms," IEEE Transactions on Knowledge and Data Engineering **23** *(8): 1200–1214.*

[8] M. Hay, C. Li, G. Miklau, and D. Jensen. (2009) "Accurate estimation of the degree distribution of private networks," Data Mining, ICDM'09. Ninth IEEE International Conference on IEEE: 169–178.

[9] K. Nissim, S. Raskhodnikova, and A. Smith. (2007) "Smooth sensitivity and sampling in private data analysis," Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, ACM: 75–84.

[10] J. Gehrke, E. Lui, and R. Pass. (2011) "Towards privacy for social networks: A zero-knowledge based definition of privacy," Theory of Cryptography Conference, Springer: 432–449.

[11] J. Blocki, A. Blum, A. Datta, and O. Sheffet. (2013) "Differentially private data analysis of social networks via restricted sensitivity," Proceedings of the 4th conference on Innovations in Theoretical Computer Science, ACM: 87–96.

[12] D. Proserpio, S. Goldberg, and F. McSherry. (2012) "A workflow for differentially-private graph synthesis," Proceedings of the 2012 ACM workshop on Workshop on online social networks: 13–18.

[13] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev. (2011) "Private analysis of graph structure," Proceedings of the VLDB Endowment, **4** *(11): 1146–1157.*

[14] W.-Y. Day, N. Li, and M. Lyu. (2016) "Publishing graph degree distribution with node differential privacy," Proceedings of the 2016 International Conference on Management of Data, ACM: 123–138.

[15] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. (2013) "Analyzing graphs with node differential privacy," Theory of Cryptography, Springer: 457–476.

[16] J. Leskovec. (2017) "Stanford large network dataset collection," http://snap.stanford.edu/data, accessed: 2017-09-19.