Perturbation based Technique for Privacy Preserving Social Network Data

A PROJECT REPORT

Submitted by

PRABAKARAN A
Reg. No. 11MSE1108

in partial fulfillment for the award of the degree of

Master of Science

in

Software Engineering (5 Year Integrated Programme)

School of Computing Science and Engineering
VIT University
Vandalur - Kelambakkam Road, Chennai - 600 127

April - 2016

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore ▪ Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

School of Computing Science and Engineering

DECLARATION

I hereby declare that the project entitled **Perturbation based Technique for Privacy Preserving Social Network Data** submitted by me to the School of Computing Science and Engineering, VIT Chennai, 600 127 in partial fulfillment of the requirements of the award of the degree of **Master of Science** in **Software Engineering (5 year Integrated Programme)** is a bona-fide record of the work carried out by me under the supervision of **DR.N.MAHESWARI**. I further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place: Chennai
Date:

Signature of Candidate
(PRABAKARAN A)

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore ▪ Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled **Perturbation based Technique for Privacy Preserving Social Network Data** is prepared and submitted by **PRABAKARAN A (Reg. No. 11MSE1108)** to VIT Chennai, in partial fulfilment of the requirement for the award of the degree of **Master of Science** in **Software Engineering (5 year Integrated Programme)** is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Guide/Supervisor

Name: DR.N.MAHESWARI
Date:

Program Chair

Name: DR.N.MAHESWARI
Date:

Examiner

Name:
Date:

Examiner

Name:
Date:

(Seal of SCSE)

Acknowledgement

It is my pleasure to express with deep sense of gratitude to DR.N.MAHESWARI, Professor, SCSE VIT University, for her constant guidance, continual encouragement and understanding; more than all, she taught me patience in my endeavour. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Data Mining. I would like to express my gratitude to Dr.L.JEGANATHAN, Dean SCSE, for providing with an environment to work in and for his inspiration during the tenure of the course. I express ingeniously my whole-hearted thanks to Management, Program Chair, all faculty members and staff members working as limbs of our university for their not-self-cantered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project

PRABAKARAN A
Reg. No. 11MSE1108

Abstract

Social Network Analysis and Mining (SNAM) techniques have drawn significant attention in the recent years due to the popularity of online social media. Despite the advance of SNAM, the utility of a social network is highly affected by its completeness. However, a modest privacy gains may reduce a substantial SNAM utility. It is a challenge to make a balance between privacy and utility in social network data sharing and integration. In order to share useful information among different organizations without violating the privacy policies and preserving sensitive information, by generalization and probabilistic approach of social network integration. Particularly, by generalizing social networks to preserve privacy and integrating the probabilistic models of the shared information for SNAM. In addition to the current social network anonymity de-identification techniques, a business transaction warehouse is essentially a social network, in which weights are attached to network edges that are considered to be confidential. In such a business transaction social network, weight can represent the cost of one transaction between two business entities, the physical distance between two stores, to name a few. Perturbing the weights of some edges is for preserving data privacy when the network is published, while retaining the shortest path and the approximate length of the path between some pairs of nodes is required in the original network. Two privacy-preserving strategies are used. The first strategy is based on a Gaussian randomization multiplication, the second one is a Greedy perturbation algorithm based on graph theory. In particular, the second strategy not only yields an approximate length of the shortest path while maintaining the shortest path between selected pairs of nodes, but also maximizes privacy preservation of the original weights. Experimental results of both strategies are discussed.

School of Computing Science and Engineering, VIT University, Chennai

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Social networks have rapidly grown in popularity over the past decade and have become an interesting area of research. Preserving user privacy in social network data published for research purposes is an important concern

1.1 Background

Due to recent advances in computer and network, gathering and collecting data concerning different individuals and organizations becomes relatively easy. Establishing and researching social networks have become a major interest in data mining communities. There are a variety of social networks published so far for research purpose, including those for epidemiologists [24], sociologists [8], zoologists, intelligence communities (terrorism networks), and much more

A social network is a special graph structure made of entities and connections between these entities. The entities, or nodes, are abstract representations of either individuals or organizations that are connected by one or more attributes. The connections, or edges, denote relationships or interactions between these nodes. Connections can be used to represent financial exchanges, friend relationships, conflict likelihood, web links, sexual relations, disease transmission (epidemiology), etc.

Social networks typically contain a large amount of private information and are good sources for data analysis and data mining. The need to protect confidential, sensitive, and security information from being disclosed motivates researchers to develop privacy preserving techniques for social networks. One of the major challenges, therefore, is to approach an optimal tradeoff between se-

School of Computing Science and Engineering, VIT University, Chennai

curing the confidential information and maximizing the social networks utility analysis.

Recent study of privacy preservation in social networks focuses on the de-identification process to protect the privacy of individuals while preserving the patterns between small communities [5, 10, 21] Such de-identification processes are often helpful when the individuals identity is considered to be confidential, such as a patients identity.

However, the individual identity is not always considered to be confidential. For example, a recent tool called ArnetMiner [14] has been developed to allow mining the academic research network through a public web portal. Each node of this network represents a researcher. An edge exists between two nodes if the corresponding researchers share a co-authorship. Another feature that is supported by the system is the association search between two researchers, which enumerates all possible topics that connect one researcher to the other and show how closely the two researchers are connected. In this case, since all data needed to compute such network are obtained from public web pages or databases, privacy of identity is not a big concern. However, it is important to realize that the network derived from these public data makes implicit knowledge explicit and more specific, such as the association between individuals.

Next, another example of weighted social networks is given, which is thoroughly studied in [13]. The social network represents an automotive business network between Japanese corporations and American suppliers in North America. The background behind this example is that many Japanese automotive companies have already taken roots in North America, and American suppliers would seek access to such a profitable subcontract market. On one hand, the existence of a long-term and loyal connection between Japanese first-tier suppliers and auto makers plays a key role in making decisions. So these preferences surely prevent American suppliers from obtaining contracts easily. On the other hand, since most first-tier suppliers are sensitive to importing cost and have U.S. political pressure to avoid mass outsourcing, they prefer to collaborate with the qualified local American suppliers. Therefore, it is practical and economical to become a subcontractor of these lower-level suppliers. For every potential American supply contractor, it is desirable to obtain a comprehensive business network that can guide them in finding the most economical business path.

However, due to the fierce competition between suppliers, managers may not be willing to disclose the true transaction expenses to their adversaries. Otherwise, their adversaries could probably reduce the quotation below the price obtained in a secret bidding competition. Hence, suppliers would like to preserve their trans-

action expenses (edge weights) before the business network is published. At the same time, some global and local utilities of the social networks, such as the optimal supply chains (the lowest cost path between companies) and the corresponding lengths, are probably desired to be maintained for future analysis.

The focus is on publishing a social network which maintains the utility of the shortest paths while perturbing the actual weight between a pair of entities. The edge between two nodes is often associated with a quantitative weight that reflects the affinity between the two entities. The weighted graph allows deeper understanding about relationships between entities within the network. The shortest path between a pair of nodes is a path such that the sum of the weights of its constituent edges is to be minimum. The shortest path is a major data utility which has applications in different fields. So each node in this business graph represents a company or a supplier (or an agent), the edge denotes business relationship and the weight of the edge represents the transaction expenses according to some measures (such as per month, per person or per transaction) between the two entities [19]. As an abstract business network in figure 1.1, the bold numbers beside edges are the transaction expenses per month (the unit is million/month).

In this business example, for example, Company A wants to purchase some products or services, in the future, from Company D which cannot directly access each other due to some trade barriers. Company A needs to choose some trade intermediate suppliers who have the most competitive path (the shortest path of price) between themselves and Company D (maybe these suppliers need other suppliers to connect Company D). If the weights of the business social network are perturbed as in figure 1.2 but the shortest paths (and the corresponding lengths) are well preserved, Company A may be able to make an intelligent decision based on this privacy-preserving social network without having to know confidential details of the relationship between agents and Company D.

According to the algorithms, the perturbed graph preserves the same shortest paths and maintains the shortest path lengths close to the true values. Moreover, the total privacy of all edge weights is maximized by the methods. Here, the more weight of an edge changes, the more edges privacy is preserved. As the example in figure 1.1, the true expense between Agent 2 (or Supplier 2) and Company D is lower than that between Agent 3 (or Supplier 3) and Company D, but in the perturbed network as in figure 1.2, the expense between Agent 2 and Company D is higher than that between Agent 3 and Company D. So in a bidding competition, the business secret between Agent 2 and Company D is blind to Agent 3 (Agent 2s adversary) even if the perturbed business network is published. After a series of perturbations, the final perturbed version is in figure 1.2. The shortest path

between Company A and Company D is the same as the original one and the corresponding perturbed length is close to the original one.

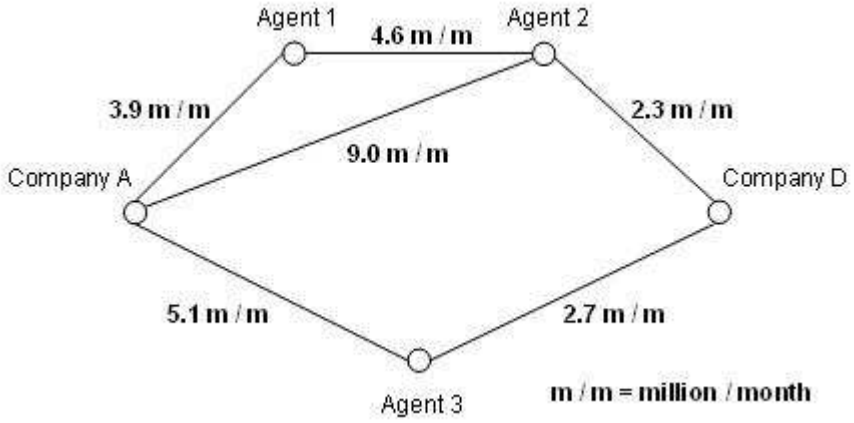

Figure 1.1: Original business network. All nodes in this gure represent either a company

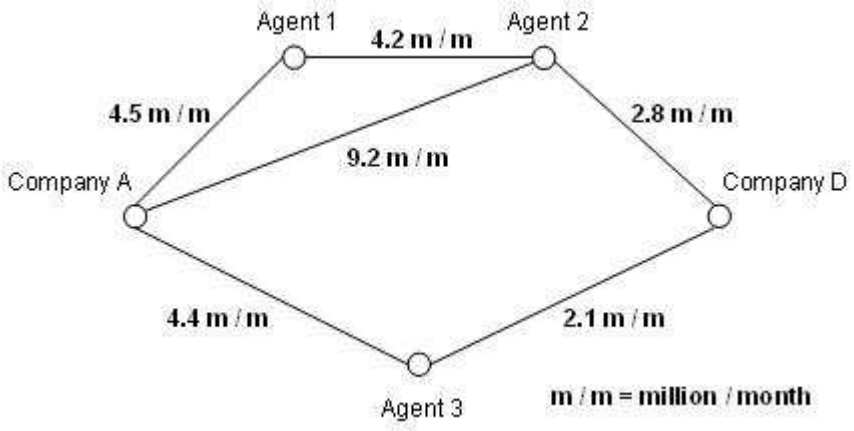

Figure 1.2: Perturbed business network

To utilize the privacy-preserving social network analysis, each people (or organization) has a local (private) weighted graph before perturbation. The process of information sharing and perturbation can be done either in a distributed environment or a central situation. In a distributed environment, each person perturbs

the individual local weighted graph, and then publishes the perturbed weights to the public. After all edges perturbation and publication, a global perturbed graph will be composed of individuals local perturbed graphs. In a central case, assume that there exists a trusted third-party which will absolutely never collude with anyone. Each person submits the original graph structure along with edges weights to the trusted third-party which then perturbs the whole graph with the aid of the analysis algorithms. After the central perturbation, the third-party releases the perturbed social network to the public.

Although just revealing the shortest paths and hiding all weights of edges between any two nodes can achieve privacy preservation in some cases, the unweighted shortest paths cannot have the same utility as the weighted ones in a real world. For example, in figure 1.1, if all weights are hidden and it only shows Company A that (Agent 1 $\rightarrow$ Agent 2$\rightarrow$ Company D), (Agent 3$\rightarrow$ Company D) are the shortest paths between Agent 1 and Company D, and Agent 3 and Company D, respectively, Company A cannot choose an optimal one between the two paths to Company D just based on the unweighted shortest paths. In this unweighted graph, the two unweighted shortest paths are equivalent to some extent, but actually they are essentially different for Company A, since the shortest path (Agent 3$\rightarrow$ Company D) is shorter (and more economical) than the path (Agent 1 $\rightarrow$ Agent2 $\rightarrow$ Company). D). Therefore, it is needed to preserve the shortest paths as well as the corresponding shortest paths lengths which facilitate business decision-making in a competitive environment.

1.2 Problem Formulation

The edge weights are perturbed while the shortest paths between pairs of nodes are preserved without adding or deleting any node and edge. For this purpose, two perturbation strategies [20] are used, Gaussian randomization multiplication and greedy perturbation. The two strategies serve different purposes. The Gaussian method mainly focuses on preserving the lengths of the perturbed shortest paths within some bounds of the original ones but does not guarantee the same shortest path after perturbation. The advantages of the greedy perturbation algorithm over the Gaussian algorithm are that it can keep the same shortest paths during the perturbation, in addition to keeping the perturbed shortest path lengths close to those of the original ones.

1.3 Related Work

In privacy-preserving data mining, various techniques have been developed to maintain the data utility without disclosing the original data and guarantee that the data mining analysis results are as close to those based on the original data as possible. Generally, among various privacy-preserving data mining and analysis techniques, it mention two main categories. Methods in the first category modify data mining algorithms so that they allow data mining operations on distributed datasets without knowing the exact values of the data or without direct access to the original datasets. Methods in the other category perturb the values of the datasets to protect privacy of the data values. These methods are designed to perturb the whole dataset or the confidential parts of the dataset using matrix decomposition or signal processing techniques and randomization addition.

In social networks, the data is not meaningfully represented by a tabular or matrix. Hence, most people do not use traditional matrix-based algorithm to preserve privacy. They emphasize the protection of social entitys identification via de-identification techniques. For example, Hay et al [21]. and Zhou et al [5]. presented a framework to add and delete some unweighted edges in social networks to prevent attackers from accurately re-identifying the nodes based on background information about the neighborhood. Read et al [24]. and Rogers [8] theoretically defined a family of attacks based on random graph theory and link mining prospect. They first added some distinguishable nodes into the social network before it is collected and published, and after that they used the known added nodes to differentiate the original graph patterns. Zheleva et al [10] Developed a model in which nodes are not labeled but edges are labeled which are sensitive and should be hidden. They hid and removed some edges based on edge clustering techniques.

The above methods all focus on preserving either node or edge privacy. This work focus on emphasizing edge weight privacy. Data owners may not want to release the exact weight of each edge, but would like to keep the shortest paths of a set of nodes and the lengths of the corresponding shortest paths as unperturbed as possible, for the data analysis purpose.

Chapter 2

Literature Review

2.1 Current Status of Traditional Data Privacy Preservation

In the past decade, there have been a large number of privacy-preserving data mining literature. Many researchers attempt to develop techniques to maintain data utilities without disclosing the original data and to produce data analysis results that are as close to those based on the original data as possible. Among those techniques, there are two main categories. Methods in the first category modify data mining algorithms so that they allow data mining operations on distributed datasets without knowing the exact values of the data or without directly accessing the original dataset. Methods in the other category perturb the values of the dataset to protect privacy of the data attributes. These methods pay more attention to perturbing the whole dataset or the confidential parts of the dataset by using distributions of certain noises [15] [1, 7, 12, 30].

In the second category, perturbation techniques are divided into two subcategories, data addition and data multiplication, both of which are easy to implement and practically useful. For instance, Tendick [22] perturbed each attribute in the dataset independently of the other attributes by the addition of a multivariate normal distribution e with the mean 0 in the form of A = A + e.

For the data additive perturbation strategy, although individual data items are distorted, the aggregate properties of the original data can be accurately maintained. These properties may facilitate data clustering [23] and classification [23] and finding association rules. Data multiplicative perturbation is also good for privacy-preserving data mining. This technique dramatically distorts the origi-

nal data, but maintains inter-data distances which are also effective for distance specific applications such as clustering and classification [7, 15]. The difference between the two perturbation strategies is that, in the former strategy, only the aggregate distribution properties are available for data mining and the individual data behavior is hidden, while in the latter case it can keep more data-specific properties such as distances which can facilitate more diverse data mining tasks.

Recently, in addition to data addition and data multiplication strategies, matrix decomposition and factorization techniques have been used to distort numerical valued datasets in the applications of privacy-preserving data mining. In particular, singular value decomposition (SVD) [27, 28] and non negative matrix factorization (NMF) have been shown to be very effective in providing high level data privacy preservation and maintaining high degree data utilities.

In recent years, however, it is noticed that the perturbed or distorted datasets from certain data perturbation techniques may not be safe if an attacker has some background information about the original datasets . In practice, it is unlikely that an attacker has no idea about the original dataset [12, 25, 26] other than the public perturbed version. The common sense, statistical measure, reference, and even a small amount of leakage may dramatically help the attacker weaken the privacy of the dataset. Guo and Wu [25,26] calculated a useful upper bound and lower bound about the difference between the original dataset and the estimated dataset which is computed from the perturbed dataset by spectral filtering techniques. Aggarwal [23] presented that, in the data additive perturbation, the privacy is susceptible from a known public dataset in a high dimensional space.

Their works have mentioned the use of background information probably possessed by the attacker in either data additive perturbation or multiplicative strategies, and they needed much more background information to support their privacy breach analysis. Attention will be paid to privacy breach analysis of the perturbed dataset with one single background record in a general data perturbation.

Besides, there are several classes of data distortion or perturbation methods. For example, one class is focused on data anonymization. Briefly, on one hand, the data anonymization strategy removes certain parts of the dataset such as unique and confidential identifiers, e.g., social security numbers or drivers license numbers or credit card numbers. Sweeney [18] demonstrated that this strategy may not be safe to guarantee identification privacy because the intruders can discover certain secret information by exploiting relationships among other attributes. On the other hand, the data randomization perturbation preserves data utilities such as patterns and association rules by using the additive random noise.

2.2 Current Status of Social Networks Privacy Preservation

In addition to a large amount of traditional privacy preserving data mining literature, more and more researchers have paid their attention to preserving privacy of social networks. This section provides a brief survey on privacy preserving social networks. Much progress has been made in studying the properties of social networks, such as degree distribution (the degree of a node tells how many edges connect this node to other ones) [29] , network topology (isomorphism) , growth models (network temporal attraction to new members) , small-world effect (the average shortest path length for social networks is empirically small), and community identification (functional group transformation).

In social networks, the data is not meaningfully represented by a tabular or matrix. Hence, most people do not use traditional matrix-based algorithms to preserve privacy. They emphasize the protection of social entitys identification via de-identification techniques [18]. . For example, Hay et al. and Zhou et al [4]. presented a framework to add and delete some unweighted edges in social networks to prevent attackers from accurately re-identifying the nodes based on background information about the neighborhood. Read et al. and Rogers [9] defined a family of attacks based on random graph theory and link mining prospect. They first added some distinguishable nodes into the social network before it is collected and published, and after that they used the known added nodes to differentiate the original graph patterns. Zheleva et al [11]. used a model in which nodes are not labeled but edges are labeled which are sensitive and should be hidden. They hid and removed some edges based on edge clustering techniques. These methods all focus on preserving either node or edge privacy.

Based on these theoretical analysis, researchers developed various algorithms to add/delete some edges to break the chances of differentiating the given nodes and/or edges from deidentified social networks. They placed emphases on the protection of social entitys identification via de-identification k-anonymity and variants. For example, Backstrom et al [17]. described a framework to distinguish the possibility of a certain edge existed in a social network. It shows that the identification of almost any node is easy to be leaked based on the implantation. Korolova et al [2]. developed a breach analysis on the nodes identification just based on a part of background information regarding the neighborhood. Wang et al. used a logic function to quantify the node anonymity in [16]. Hay et al. , Zhou et al , and Liu et al [4]. presented an essentially similar scheme to add

and/or delete some unweighted edges in social networks to keep malicious users from accurately re-identifying target nodes based on auxiliary information about the number of neighbors. Interested readers can refer to for a comprehensive discussion about privacy preserving social networks against the disclosure of confidential nodes and links. For insight about privacy preserving social networks look [6].

2.3 Gantt Chart

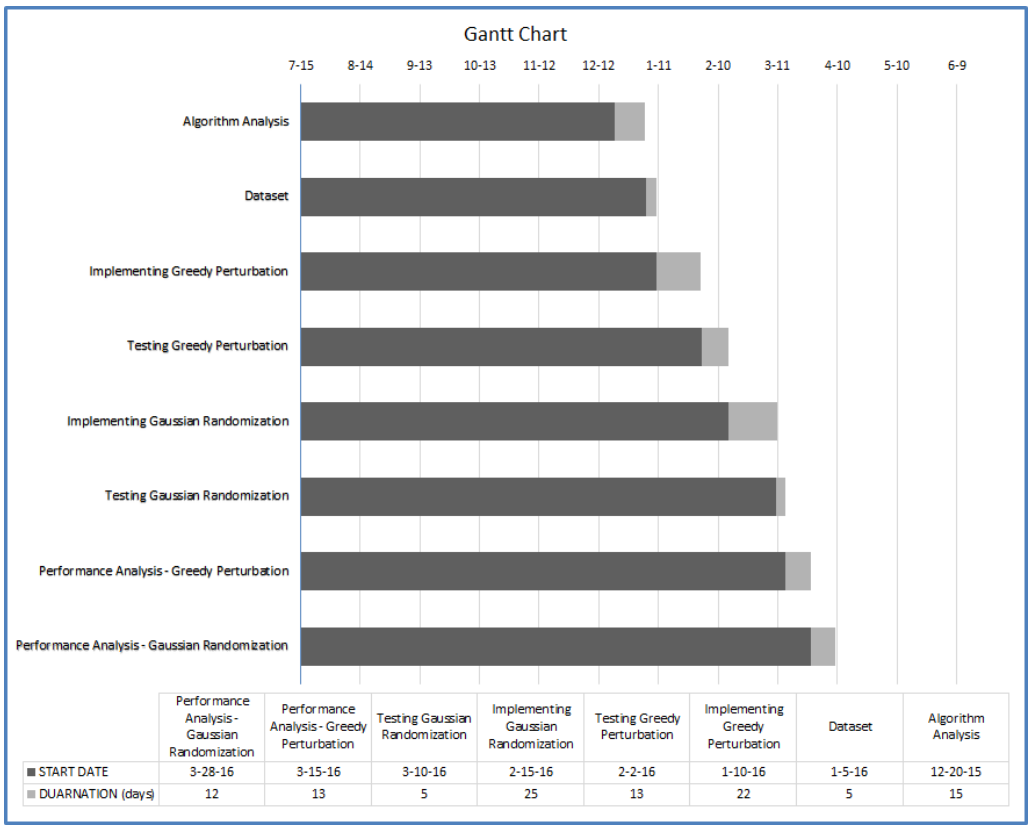

Figure 2.1: Gantt Chart

Chapter 3

System Design

3.1 Use Case Diagram

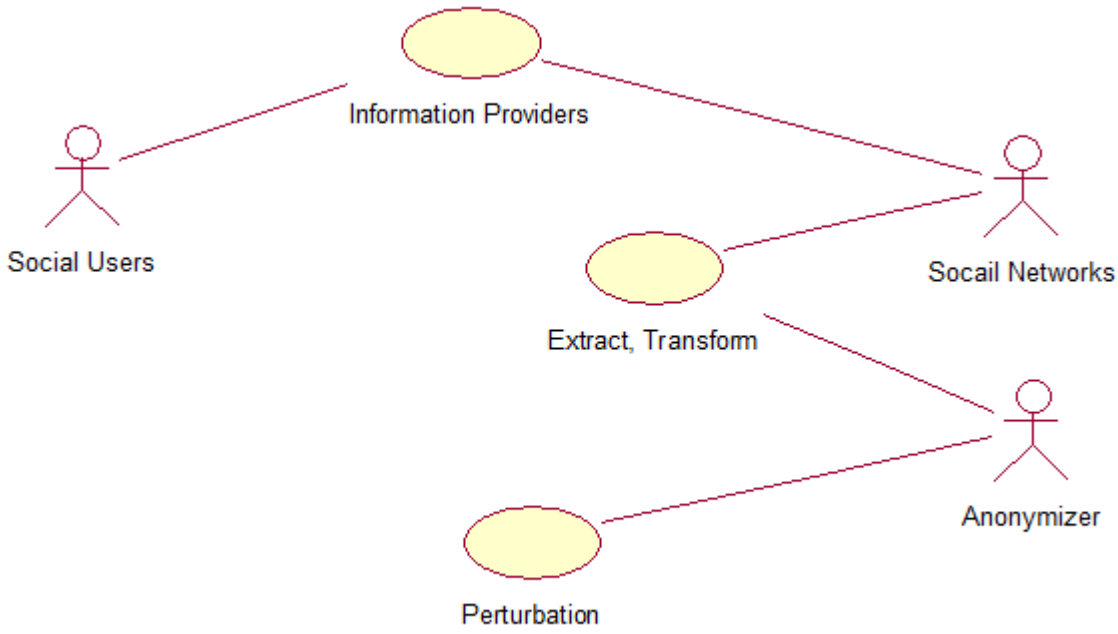

Figure 3.1: Use Case Diagram

3.2 Sequence Diagram

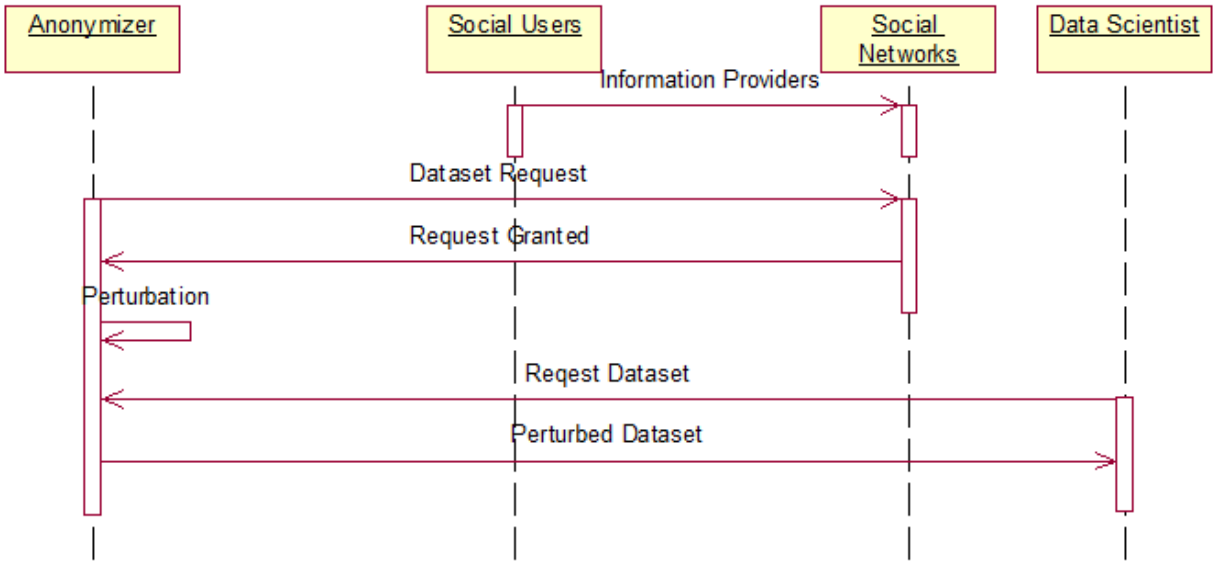

Figure 3.2: Sequence Diagram

3.3 Collaboration Diagram

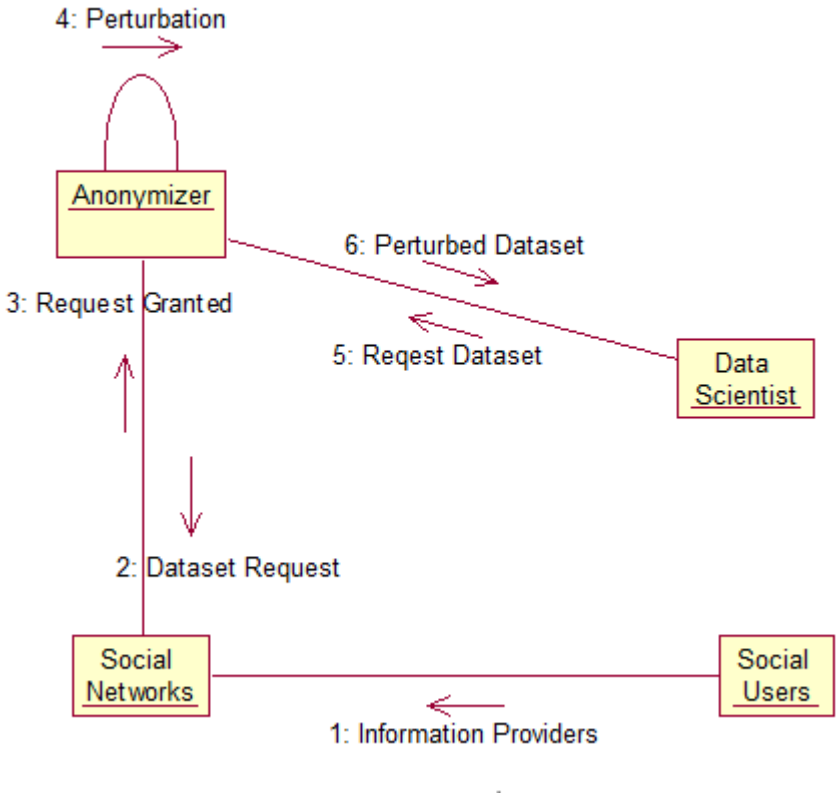

Figure 3.3: Collaboration Diagram

3.4 Data Flow Diagram

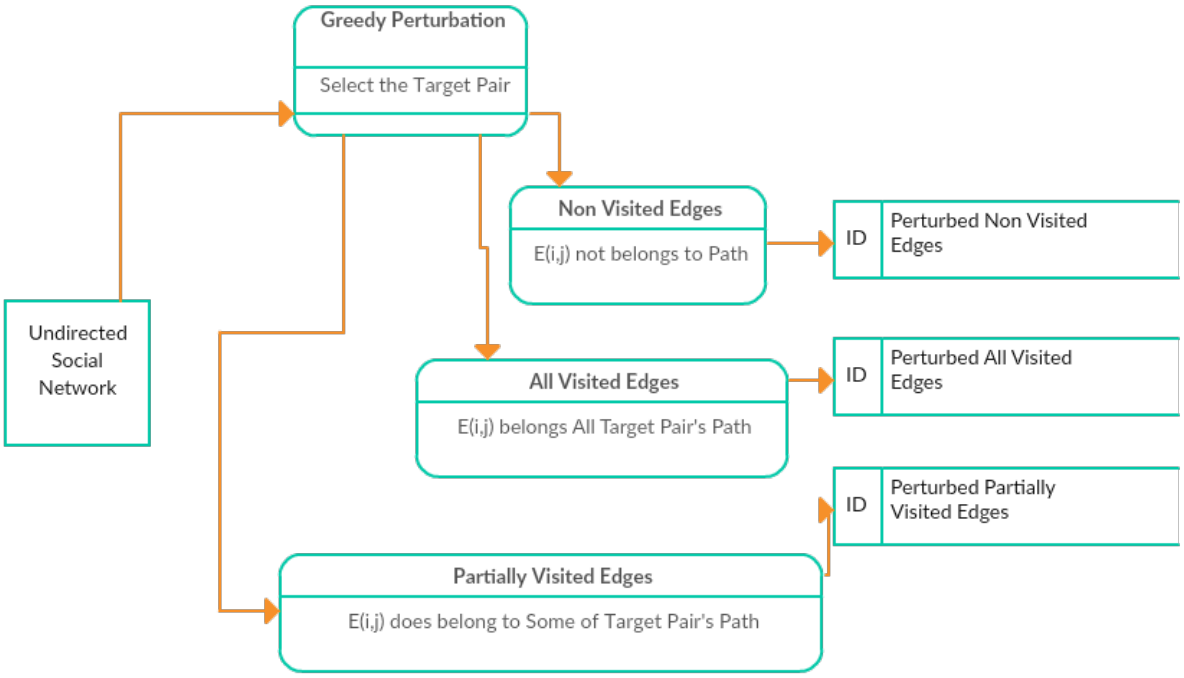

Figure 3.4: Data Flow Diagram

Chapter 4

Methodology

4.1 Edge Weight Perturbation

There exist a variety of social networks. Some of them are dynamic in which a social network will develop continuously and its structure may become very large and unpredictable. The others are static which may not change dramatically in a short period time. Due to the difficulty of collecting global information about the social networks in the first category, a Gaussian randomization multiplication technique [20] is implemented which does not need any network information in advance. On the other hand, a static social network is the one that useful structural information such as the existing shortest paths and the corresponding path lengths are easily obtained in advance. With this information, a useful edge weight perturbation strategy is developed based on a greedy perturbation algorithm.

4.2 Gaussian randomization multiplication strategy.

The basic idea behind this algorithm is that every two linked nodes cooperate with the generation of a random number which is consistent with a Gaussian distribution. The weight of the edge connecting these two entities is multiplied by the random number and the individual perturbed weight is released to the public. Because each edges random number and the edges perturbation process is only related to these two linked entities, the random number generation and weight perturbation have nothing to do with other edges. In other words, the perturbation of all edges weights can be done in a distributed environment. The maximum increment or decrement of each weight is only dependent on the parameters of this

distribution. So the shortest paths and the corresponding lengths will probably be preserved if the parameters of the Gaussian distribution are chosen appropriately. Assume that the parameters of the Gaussian distribution are predefined and globally known

Assume that W is an n*n matrix whose entries are either weights if two nodes have a link or 1 otherwise. W is called the adjacency weight matrix of the graph G. W* is the perturbed adjacency weight matrix with the same dimension after this schema. $N(0, \sigma^2)$ stands for an n n*n symmetric Gaussian noise matrix with the mean 0 and the standard deviation Define the perturbed weight of each edge as

$$w^*_{i,j} = w(i,j)(1 * x(i,j))i, j = 1, ..., n$$

Here $x_{i,j}$ is a randomly generated number from the Gaussian distribution. If node Vi has a connection with v_j, then vi generates a random number, $x1_{i,j}$, from the Gaussian distribution, and $v_j$also generates a random number, $x2_{i,j}$, from the same distribution. $x_{i,j}$ is the averaged value between $x1_{i,j}$ and $x2_{i,j}$.

Note that the above multiplication is based on undirected graphs. If the weight multiplication is extended to directed graph cases, the cooperation of generating $x_{i,j}$ is not necessary. Instead, if node vi has a directed edge from node i to node j, then node i can directly generate a random number $x_{i,j}$ from the Gaussian distribution without the cooperation with node j. Other procedures are the same as the above undirected graph case.

The reasons why the Gaussian randomization multiplication strategy is chosen are as follows. 1). It is straightforward to implement in practice. 2). Due to the dynamic evolution nature of social networks, collecting all global information in advance is very hard or costly in a huge and dynamic social network. In particular, in an evolutionary environment, some nodes or edges will emerge in the future and be added to the current network, in which the collection of the current state will probably be totally changed after these insertions. So it is impossible or useless to collect comprehensive global information at a given time for later analysis.

The perturbed graph is reconstructed as $G^* = V^*, E*, W*$. It is clear that the above Gaussian randomization multiplication strategy does not change the structure of the original graph. Namely, $V = V^*, E = E^*$. The only difference between G and G^* is the weights.

4.3 Greedy Perturbation Algorithm

In a static social network, some necessary information about this social network for analysis and privacy-preserving purpose is first collected. But a trusted third-party is needed who will absolutely never collude with any network entities. All social network entities submit their original graph structures along with the edges weights to the third-party. Then all analysis and perturbation procedures are done by the third-party, and a global perturbed social network will be published to the public after the perturbation. Because all analysis and perturbation are done by a central third-party, the undirected social network and directed one have a very similar procedure. In detail, only the directed edges (and the corresponding weights) and directed paths (and the corresponding lengths) are chosen to be fed into the following analysis and perturbation in a directed social network. So, the difference is not distinguished between undirected and directed social networks below.

Before applying perturbation strategy, assume that not all shortest paths of node pairs in a social network are considered to be significant. Actually, in the real world, it is not reasonable that all information is considered as confidential. Suppose that only the data owner has the right to select which shortest paths should be preserved or which ones should not be preserved. The tasks are, under data owners restrictions, to maximize the preservation of edge weights privacy and minimize the difference of the shortest paths and the corresponding lengths between the original social networks and perturbed ones as much as possible.

In other words, the assumption that not all shortest paths are confidential keeps the private shortest paths (the starting and ending nodes, (s1,s2), in the shortest paths form a node pair set H, see below) and the corresponding lengths as close to the original ones as possible, while ignoring possible changes to other public paths. Let H be the set of targeted pairs whose shortest paths and the corresponding path lengths should be preserved as much as possible. For example, in the graph $G = V, E, W$. In a real social network, some of the shortest paths are just one-edge length paths, e.g., $p_{1,3} = e_{1,3}$ but it is assumed that these shortest paths are not included in H. In this case, the greedy perturbation algorithm aims to keep the exact shortest paths and the corresponding close path lengths between v1 and v6, v4 and v6, v3 and v6, respectively

Then, in a social network G=V,E,W ($||V||$=n), the shortest path list set P and the corresponding length n*n matrix D are generated. In P, each entry $p_{s1,s2}$ is a linked list representing the shortest path between s1 and s2, (i.e., s1 and s2 are the beginning and ending nodes of the shortest path, respectively). For example,

$p_{1,6} = (v1- > v2- > v5- > v6)$, the shortest path $p_{1,6}$ successively passes through v1, v2, v5 and v6. In the matrix D, each $d_{s1,s2}$ is the length of the shortest path connecting s1 and s2. In the following contents, all node pairs (s1, s2) of $p_{s1,s2}$ and $d_{s1,s2}$ are in the set H unless otherwise stated explicitly.

So, the goal is to generate a perturbed graph $G^* = V^*, E^*, W^*$ which satisfies the

1. V^*= V and E^* = E,
2. maximize the number of $w^*{}_{i,j}$ such that $w^*{}_{i,j} \neq w_{i,j}$,
3. $d^*{}_{s1,s2} \approx d_{s1,s2}$ for every (s1,s2) in H,
4. $p^*{}_{s1,s2}$ = $p^*{}_{s1,s2}$, for every (s1,s2) in H. Here, s1 and s2 are the beginning and ending nodes of the shortest paths in H, respectively.

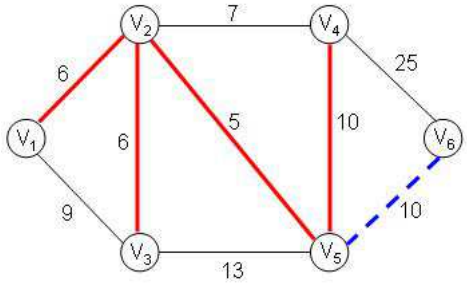

Figure 4.1: Three different categories of edges. The red bold-faced edges are partially visited edges, the black thin edges are non-visited ones, and the blue dashed edge is the all-visited edge.

4.3.1 Non Visited Edges

An edge $e_{i,j}$ is a non-visited edge, if $e_{i,j}$ =2 $p_{s1,s2}$ for every $(s1, s2)$. In other words, none of the shortest path in P passes through the edge $e_{i,j}$. all black thin edges such as edges $e_{1,3}, e_{2,4}, e_{4,6}$ and $e_{3,5}$ are non-visited edges, because the shortest paths of all three targeted pairs in H= (1,6), (4,6), (3,6) do not pass through these edges. In practice, empirically, the non-visited edges are the majority of edges in a social network.

If a non-visited edge $e_{i,j}$ increases its weight by any positive value t (the new perturbed weight is $w^*{}_{i,j} = w_{i,j} + t$), all $d_{s1,s2}$ and $p_{s1,s2}$ in H will not be changed, i.e., $d^*{}_{s1,s2} = d_{s1,s2}$ and $p^*{}_{s1,s2} = p_{s1,s2}$. Because nobody in H passes any non-visited edge, increasing the weights of non visited edges to any value will not change the shortest paths and the corresponding lengths in H. refer figure 4.2.

4.3.2 All Visited Edges

An edge $e_{i,j}$ is called an all-visited edge, if $e_{i,j} \in p_{s1,s2}$ for every (s1,s2) $\in$ H, i.e., all the shortest paths in H pass through the edge $e_{i,j}$ the blue dashed edge $e_{5,6}$ is the all-visited edge since the shortest paths $p_{1,6}, p_{4,6} and p_{3,6}$ in H all go through the edge $e_{5,6}$. Typically, the all-visited edges are very rare in a real social network.

If an all-visited edge $e_{i,j}$ decreases its weight to any positive value (i.e.,$w^*{}_{i,j} = w_{i,j} - t$ and $w^*i, j > 0$), all $p_{s1,s2}$ in H will not be affected, but $d_{s1,s2}$ will be decreased. Actually, $p_{s1,s2} = p_{s1,s2}$ and $d^*{}_{s1,s2} = d_{s1,s2} - t$. refer figure 4.2.

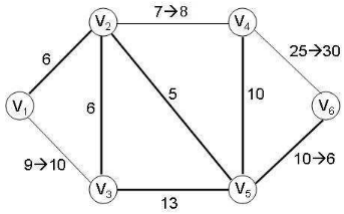

Figure 4.2: Perturbation on the non-visited and all-visited edges.

In a social network, partially-visited edges are prevalent which are major perturbation targets. To minimize the difference between the length of the original shortest path and that of the corresponding perturbed shortest path, two perturbation schemes are developed on partially-visited edges. If the current length of the perturbed shortest path is bigger than the original one, the weight of one edge in this path can be decreased. Otherwise, its weight is increased. So increasing and decreasing are two alternate choices to keep the length of the perturbed shortest path close to the original one.

4.3.3 Partially Visited Edges

An edge $e_{i,j}$ is a partially-visited edge, if $\exists(s1, s2) \in$ H and $\exists(s3, s4) \in$ H such that $e_{i,j} \in p_{s1,s2}$, but $e_{i,j} \notin p_{s3,s4}$. In this case, only some of the shortest paths pass through this edge while this edge does not appear in other the shortest paths. The red bold-faced edges in Figure 4.1 are the partially-visited edges. For example, $e_{2,5}$ is a partially-visited edge since the shortest paths $p_{1,6}$ and $p_{3,6}$ pass through the edge $e_{2,5}$, but $p_{4,6}$ does not go through it. Each edge is perturbed in the graph by four different schemes according to these three different categories.

Partially Visited Egdes, Method A

If a partially-visited edge $e_{i,j}$ increases its weight by t (the new perturbed weight is $w^*_{i,j} = w_{i,j} + t$) and t satisfies the following condition

$$0 < t < min\{d^-_{s1,s2} - d_{s1,s2}\} for\ all\ p_{s1,s2}\ such\ that\ e_{i,j} \in p_{s1,s2}$$

all $p^*_{s1,s2}$ are not changed and $d^*_{s1,s2}$ (the edge $e_{i,j}$ is in $p_{s1,s2}$) will become larger, ($p^*_{s1,s2}$ =$p_{s1,s2}$ and $d^*_{s1,s2}$ = $d_{s1,s2}$ +t), where $d^-_{s1,s2}$ is the length of the conditional shortest path between node s1 and node s2 in a graph $G^- = \{V, E - e_{i,j}, W - w_{i,j}\}$. G^- is the graph in which only the edges $e_{i,j}$ and the corresponding weights from G are deleted. For each node pair (s1,s2), $d_{s1,s2} \leq d^-_{s1,s2}$.

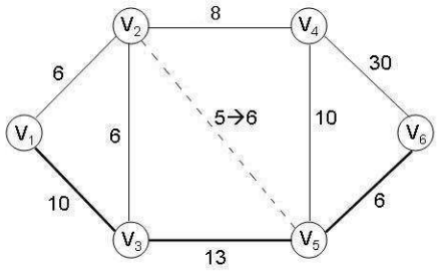

Figure 4.3: Increasing the weight of the partially-visited edge e2,5.

An example of increasing the weight of the partially-visited edge $e_{2,5}$ is shown in Figure 4.3. The shortest paths of two targeted pairs in H, $p_{1,6}$ and $p_{3,6}$, pass through the edge $e_{2,5}$, but the shortest length path $p_{4,6}$ does not go through it. Increasing $w_{2,5}$ will probably affect

the shortest paths $p_{1,6}$ and $p_{3,6}$, but has nothing to do with $p_{4,6}$. Hence, there are totally two constraints to increase $w_{2,5}$ to $w^*2,5 = w_{2,5}$ + t as follows:

$$t < d^-_{1,6} - d_{1,6}, t < d^-_{3,6} - d_{3,6}$$

where $d_{1,6}$ is 17 ($p_{1,6}$= $(v1 \rightarrow v2 \rightarrow v5 \rightarrow v6)$), $d^-_{1,6}$ is 29 ($p^-_{1,6}$=(v1 $\rightarrow$ $v3 \rightarrow v5 \rightarrow v6)$),$d_{3,6}$ is 17 ($p_{3,6}$=(v3 $\rightarrow$ $v2 \rightarrow v5 \rightarrow v6)$), and $d^-_{3,6}$ is 19. Note that these weights are perturbed weights after the perturbation of all non-visited and all-visited edges in Figure 4.3. After solving the inequalities, t should be smaller than 2, and the largest rounded integer number 1 is selected. So $w^*_{2,5} = w_{2,5} + t = 5 + 1 = 6$.

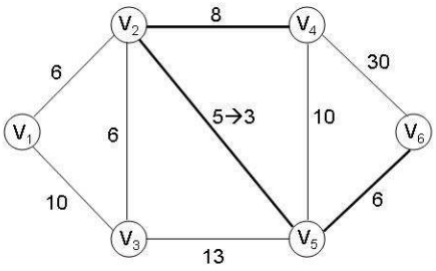

Figure 4.4: Decreasing the weight of a partially-visited edge e(2,5)

Partially Visited Egdes, Method B

For a partially-visited edge $e_{i,j}$, its weight is decreased by t (the new perturbed weight is $w^*_{i,j} = w_{i,j} - t$) and t satisfies the following condition: conditions in

$$0 < t < min\{d_{s1,i} + w_{i,j} + d_{j,s2} - d_{s1,s2} \; for \; all \; p_{s1,s2} \; such \; that \; e_{i,j} \notin p_{s1,s2}\}$$

then all $p^*_{s1,s2}$ is not changed and some $d^*_{s1,s2} = d_{s1,s2} - t$ is decreased ($p^*_{s1,s2} = p_{s1,s2}$). The path which connects $p_{s1,i}, e_{i,j}$ and $p_{j,s2}$ is the conditional shortest path between s1 and s2 through $e_{i,j}$. For example, in Figure 4.4, the conditional shortest path between v4 and v6 through $e_{2,5}$ is $(v4 \rightarrow v2 \rightarrow v5 \rightarrow v6)$, where $(v4 \rightarrow v2)$ is the shortest path $p_{4,2}$, and $(v5 \rightarrow v6)$ is the shortest path $p_{5,6}$. The meaning of Inequality (4.3) is that the length of the conditional shortest path between s1 and s2 through $e_{i,j}$ should still be larger than the length of the perturbed path $p^*_{s1,s2}$

4.3.4 Algorithm

Input

The symmetric adjacency weight matrix W of an original graph G and H (the set of selected shortest paths to be preserved).

Output

The symmetric adjacency weight matrix W^* of the corresponding perturbed graph G^*

1: generate P and D based on W, and assign D to D^*

2: for all non-visited edges $e_{i,j}$, ${w^*}_{i,j} \Leftarrow w_{i,j} + r$(r is any random positive number), and update D^*

3: for all all-visited edges $e_{i,j}$, ${w^*}_{i,j} \Leftarrow w_{i,j} - r$(r is any random positive number which is smaller than $w_{i,j}$), and update D^*

4: sort all partially-visited edges in a descending order with respect to the number of the shortest paths which pass through this partially-visited edge. Such all partially-visited edges form a stack PB

5: while PB !$\neq$ 0 do

6: pop out the top edge $e_{i,j}$ from PB

7: if of cases where ${d^*}_{s1,s2} \leq$the original one is larger than of cases where ${d^*}_{s1,s2}$ ¿ the original one then

8: generate a random value t given the range determined by Method A

9: ${w^*}_{i,j} \Leftarrow w_{i,j} + t$

10: else

11: generate a random value t given the range determined by Method B

12: ${w^*}_{i,j} \Leftarrow w_{i,j} + t$

13: end if

14: update D^*

15: end while

Chapter 5

Results and Discussions

5.1 Dataset

Synthetic Dataset Generated by R-MAT has been used. following table 5.1 is the details of generated graph.

Number of Vertices	1000
Number of Edges	1000
Max Weight	30
Min Weight	10

Table 5.1: Details of RMAT Generated Graph

5.1.1 Target Pairs

In graph $G = \{V, E, W\}$ let H be the number of selected target pairs (Figure 5.1) need to preserved, the target pairs are selected randomly, but the pairs should have path associate with it.

Start Vertex	End Vertex
17	914
1	991
136	722
578	81
126	73
636	273
703	2
800	19
142	54
137	720
594	105
766	32
258	27
823	938
297	464
800	48
195	453
86	540

Figure 5.1: Target Pairs

5.2 Performance Metrics

Performing the Performance of Algorithm and it's Percentage of Privacy Preservation by Plotting the Egde Weight and Shortest Path Length.

In each result below, the x-axis is the difference between the original ones and the corresponding perturbed ones, and the y-axis denotes the percentage of either perturbed weights or perturbed lengths which fall within the x-axis difference to the original ones. In each figure, there are two lines, a dashed line and a solid line. The dashed line represents the perturbed shortest path lengths and the solid line denotes the perturbed edge weights.

Edge Weight

For all Partially Visited Edges in Target Pairs, the difference between Original Edges and Perturbed Edges are identified and the percentage is plotted.

Shortest Path Length

For all Shortest Path Length in Target Pairs, the difference between Original Shorestest Path Cost and Perturbed Shortest Path Cost are identified and the percentage is plotted.

Correlation Coefficient

The correlation coefficient of two variables in a data sample is their co-variance divided by the product of their individual standard deviations. It is a normalized measurement of how the Percentage of Edge Weight and Shortest Path Length are linearly related.

5.3 Gaussian Randomization

Figures 5.4, 5.5 and 5.6 show experimental results with different values of σin Gaussian randomization multiplication. In each figure, the x-axis is the difference between the original ones and the corresponding perturbed ones, and the y-axis denotes the percentage of either perturbed weights or perturbed lengths which fall within the x-axis difference to the original ones. In each figure, there are two lines, a double-dashed line and a solid line. The double-dashed line represents

the perturbed shortest path lengths and the solid line denotes the perturbed edge weights.

Figure 5.3 Shows the comparison of original shortest path **P** & corresponding perturbed shortest path $\mathbf{P}^*$ cost in various Iteration using Gaussian Randomization for all Target Pair with $\sigma = 0.1$ by BellmanFord algorithm.

Type	X	Y
Weight	1.349264	0.609375
Weight	2.698528	0.234375
Weight	4.047792	0.09375
Weight	5.397056	0.046875
Weight	6.746319	0.015625
Length	1.349264	0.555555556
Length	2.698528	0.166666667
Length	4.047792	0.111111111
Length	5.397056	0.055555556
Length	6.746319	0.055555556
Weight	1.082347	0.484848485
Weight	2.164693	0.333333333
Weight	3.24704	0.121212121
Weight	4.329387	0.03030303

Figure 5.2: Sample Result of Gaussian Randomization

Figure 5.2 weight represent percentage of difference between Original edges and Perturbed edges, length represent percentage of difference between Original shortest path cost and Perturbed shortest path cost.

Target Pairs	Shortest Path	Original Cost	Perturbed Cost (1)	Perturbed Cost (2)	Perturbed Cost (3)	Perturbed Cost (4)	Perturbed Cost (5)
(17,914)	(17 : 16), (16 : 35), (35 : 42), (42 : 914)	73	68.45	78.72	67.60	72.23	66.34
(1,991)	(1 : 564), (752 : 564), (752 : 148), (148 : 991)	59	61.13	58.09	62.32	53.76	53.16
(136,722)	(1 : 136), (11 : 1), (417 : 11), (417 : 722)	77	64.89	75.97	66.50	80.70	70.79

Figure 5.3: Comparison of Original & Corresponding perturbed Shortest Path Cost in various Iteration using Gaussian Randomization with $\sigma = 0.1$

The percentage of difference between w^* and w is very close to the percentage of difference between d^* and d, (in these figures 5.4,5.5,5.6 the two lines are similar to each other at all x-axis points). As mentioned earlier, however, the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after the perturbation.

Correlation between edge weight & shortest path length are linearly related for all target pairs with 0.1 σ , 0.15 σ ,0.2 σ is 65% ,75% ,73%. Correlation between edge weight & shortest path length is varying when the σ value is changed.

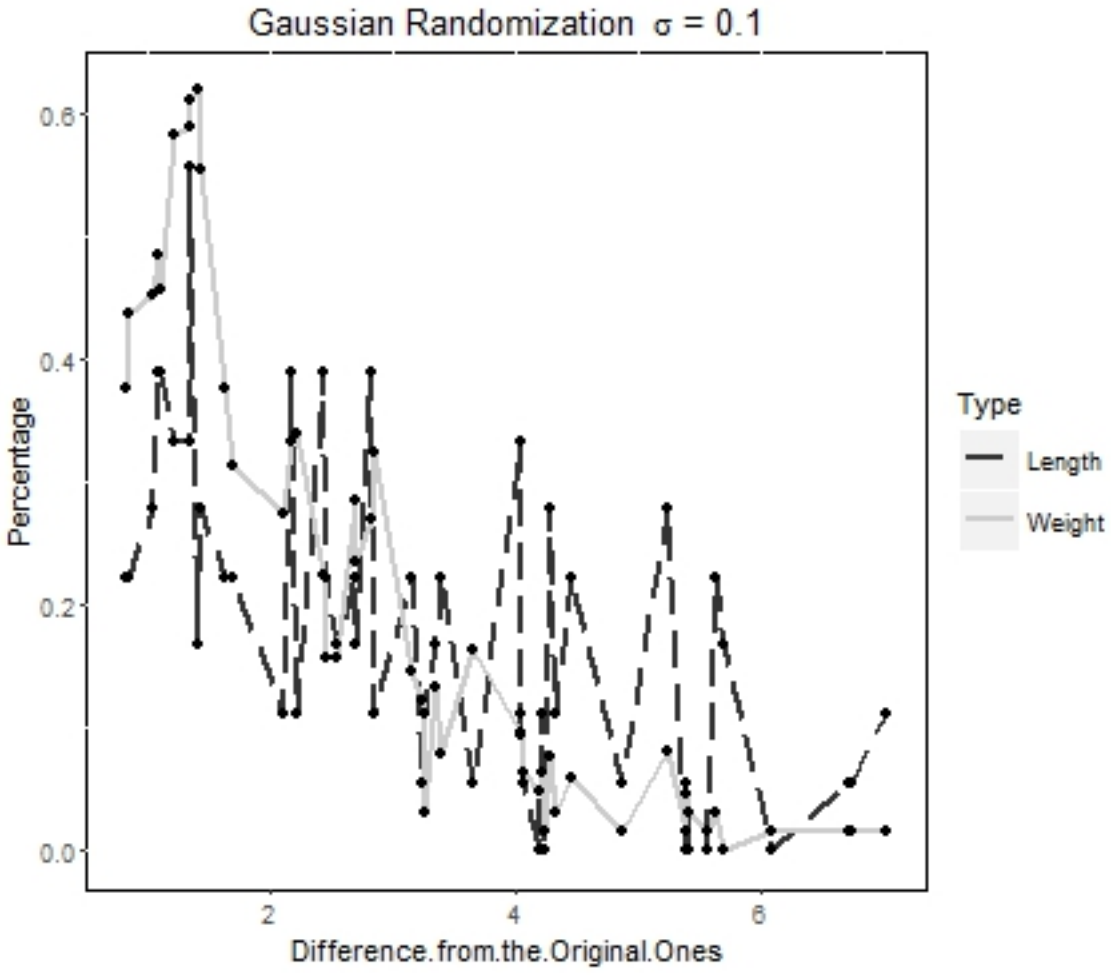

Figure 5.4: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian Randomization with σ=0.1 on Synthetic Data

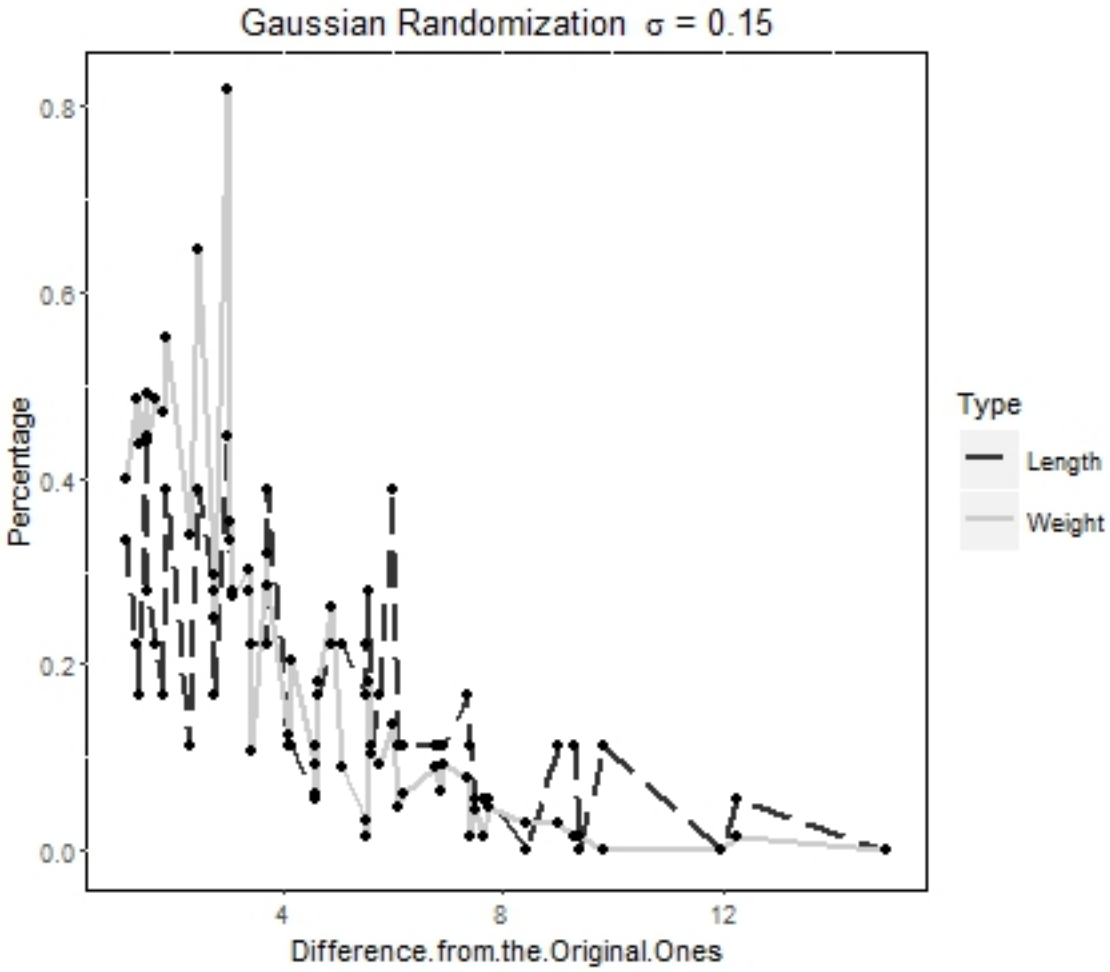

Figure 5.5: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian Randomization with σ=0.15 on Synthetic Data

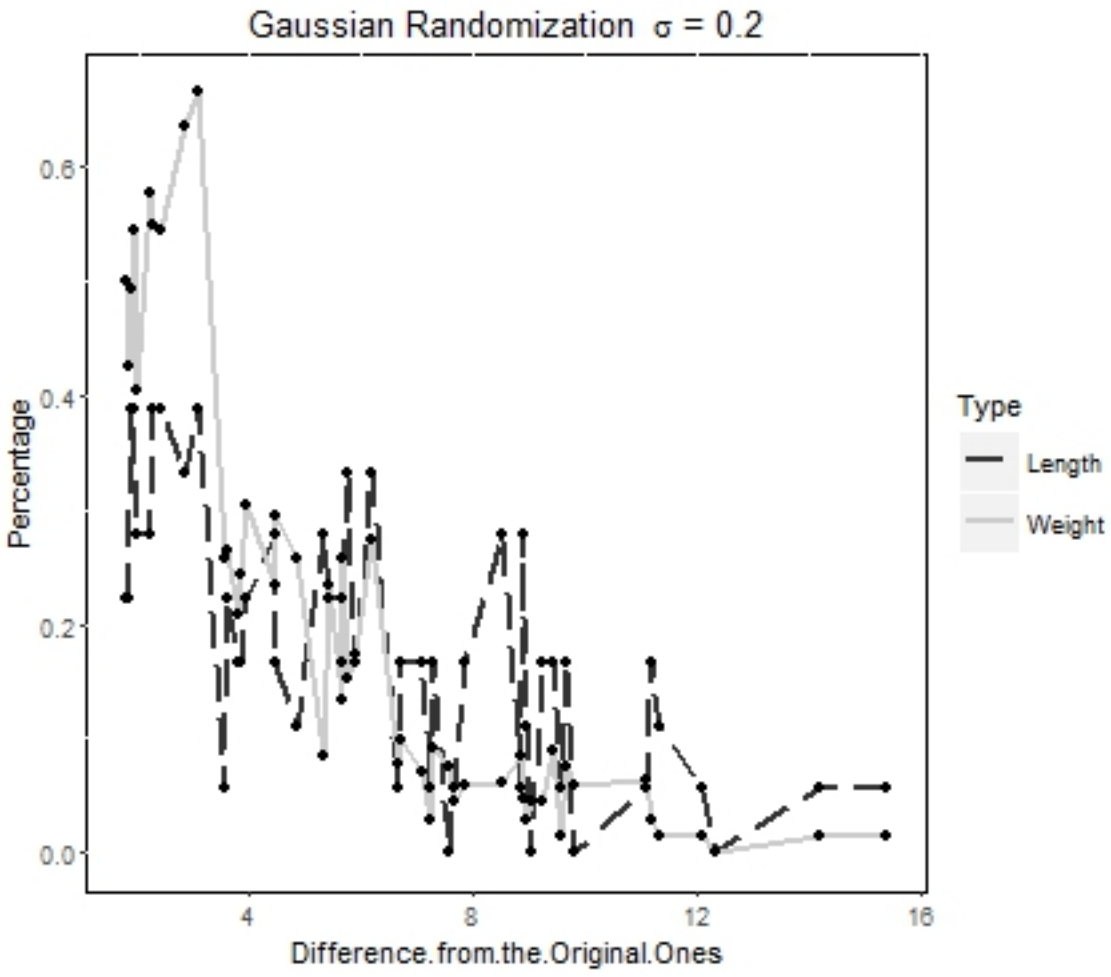

Figure 5.6: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian Randomization with σ=0.2 on Synthetic Data

5.4 Greedy Perturbation Algorithm

From figures 5.7 , 5.8 comparison of the shortest paths **P** and the corresponding lengths **D** by BellmanFord algorithm for Selected Target Pairs **H** on synthetic Dataset.

Target Pairs	Shortest Path	Original Cost
17,914	[(17 : 16), (16 : 35), (35 : 42), (42 : 914)]	73
1,991	[(1 : 564), (752 : 564), (752 : 148), (148 : 991)]	59
136,722	[(1 : 136), (11 : 1), (417 : 11), (417 : 722)]	77
578,81	[(578 : 145), (145 : 644), (644 : 81)]	54
126,73	[(394 : 126), (394 : 449), (73 : 449)]	45
636,273	[(636 : 5), (64 : 5), (64 : 751), (751 : 273)]	69
703,2	[(79 : 703), (5 : 79), (252 : 5), (252 : 2)]	68
800,19	[(532 : 800), (532 : 698), (698 : 19)]	48
142,54	[(142 : 5), (5 : 51), (54 : 51)]	49
137,720	[(384 : 137), (384 : 407), (407 : 720)]	63
594,105	[(535 : 594), (535 : 105)]	23
766,32	[(4 : 766), (604 : 4), (32 : 604)]	55
258,27	[(258 : 23), (320 : 23), (320 : 314), (314 : 266), (266 : 835), (835 : 27)]	91
853,938	[(504 : 823), (504 : 563), (985 : 563), (938 : 985)]	73
297,464	[(407 : 297), (407 : 441), (441 : 464)]	56
800,48	[(532 : 800), (532 : 300), (283 : 300), (283 : 48)]	59
195,453	[(195 : 609), (609 : 1), (453 : 1)]	57
86,540	[(219 : 86), (219 : 157), (157 : 504), (504 : 754), (754 : 540)]	69

Figure 5.7: Original Shortest Paths & Cost using Greedy Perturbation with 100% of targeted pairs

The greedy perturbation algorithm experiment, the weights of non-visited edges and all-visited edges could be changed dramatically without affecting any of the shortest paths in H.

Hence, only the weights of all partially-visited edges are concerned in synthetic data. The experimental results with the greedy perturbation algorithm are shown in Figures 5.9, 5.10, 5.11, 5.12

Target Pairs	Original Cost	Cost	Cost	Cost	Cost	Cost	Average Cost
17,914	73	71	76	74	78	76	75
1,991	59	72	67	65	63	61	65.6
136,722	77	81	143	113	122	92	110.2
578,81	54	53	36	62	59	53	52.6
126,73	45	38	49	52	52	52	48.6
636,273	69	66	58	73	63	73	66.6
703,2	68	68	69	64	79	73	70.6
800,19	48	19	43	49	49	49	41.8
142,54	49	38	56	50	51	53	49.6
137,720	63	62	49	58	53	81	60.6
594,105	23	43	23	28	25	29	29.6
766,32	55	64	39	54	64	75	59.2
258,27	91	100	92	93	76	82	88.6
853,938	73	68	78	74	79	73	74.4
297,464	56	73	57	62	73	76	68.2
800,48	59	47	61	67	66	62	60.6
195,453	57	67	61	62	53	53	59.2
86,540	69	83	101	71	70	91	83.2

Figure 5.8: Comparison of Original & Perturbed Shortest Path Cost in various Iteration using Greedy perturbation with 100% of targeted pairs on on synthetic Dataset

From Figures 5.9, 5.10, 5.11, 5.12, it is obvious that even a large amount of targeted pairs in H which need keep exactly the same shortest paths and the close lengths of the shortest paths, the perturbed shortest path lengths are still very close to the original ones. In addition to this, the shortest paths of all 25%, 50% , 75% and 100% targeted pairs are exactly kept after perturbation, respectively.

Correlation between edge weight and shortest path length are linearly related for 25%, 50%, 75%, 100% of target pairs is 69.1%, 78.8%, 94.4%, 92.2% on synthetic Dataset.

So, whenever percentage of target pairs increase relation between lenght and weight also increases while preserving the privacy in social network data.

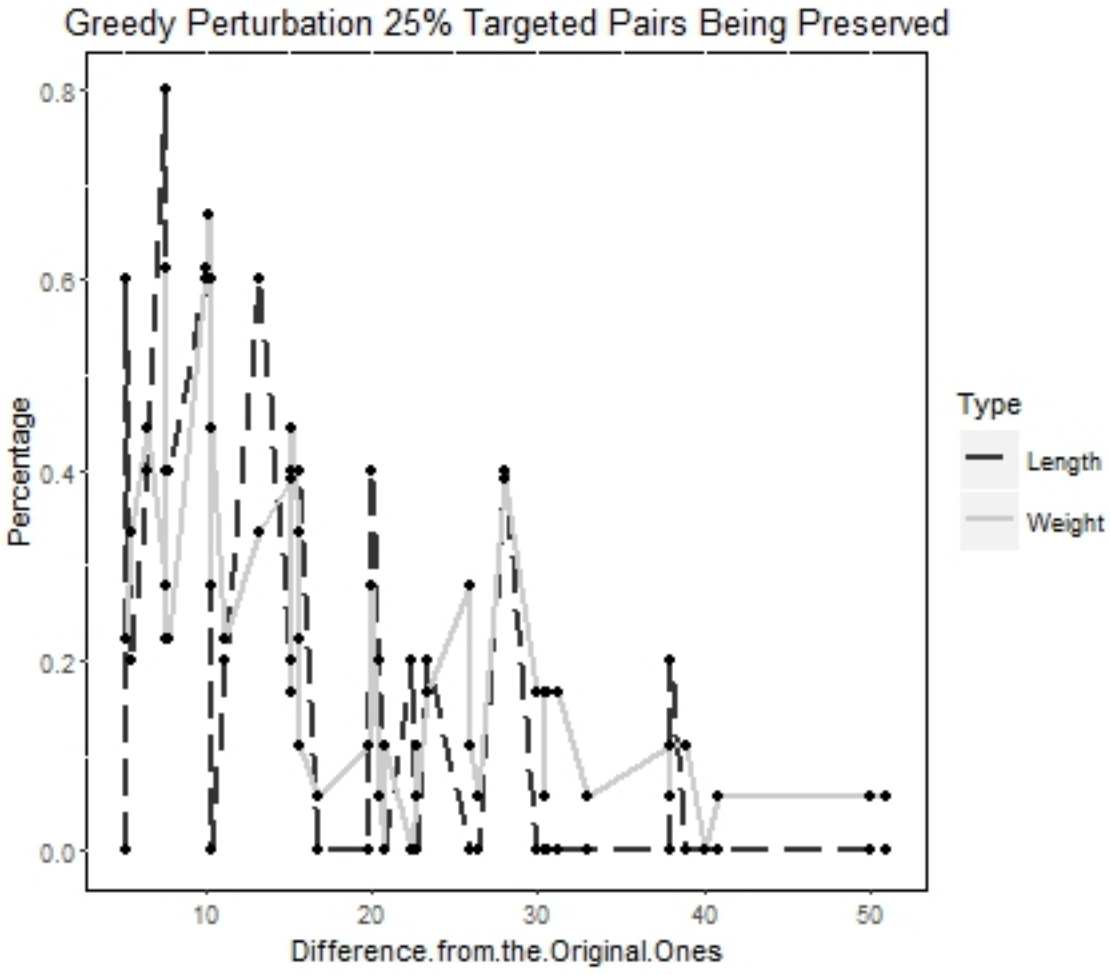

Figure 5.9: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 25% targeted pairs being preserved

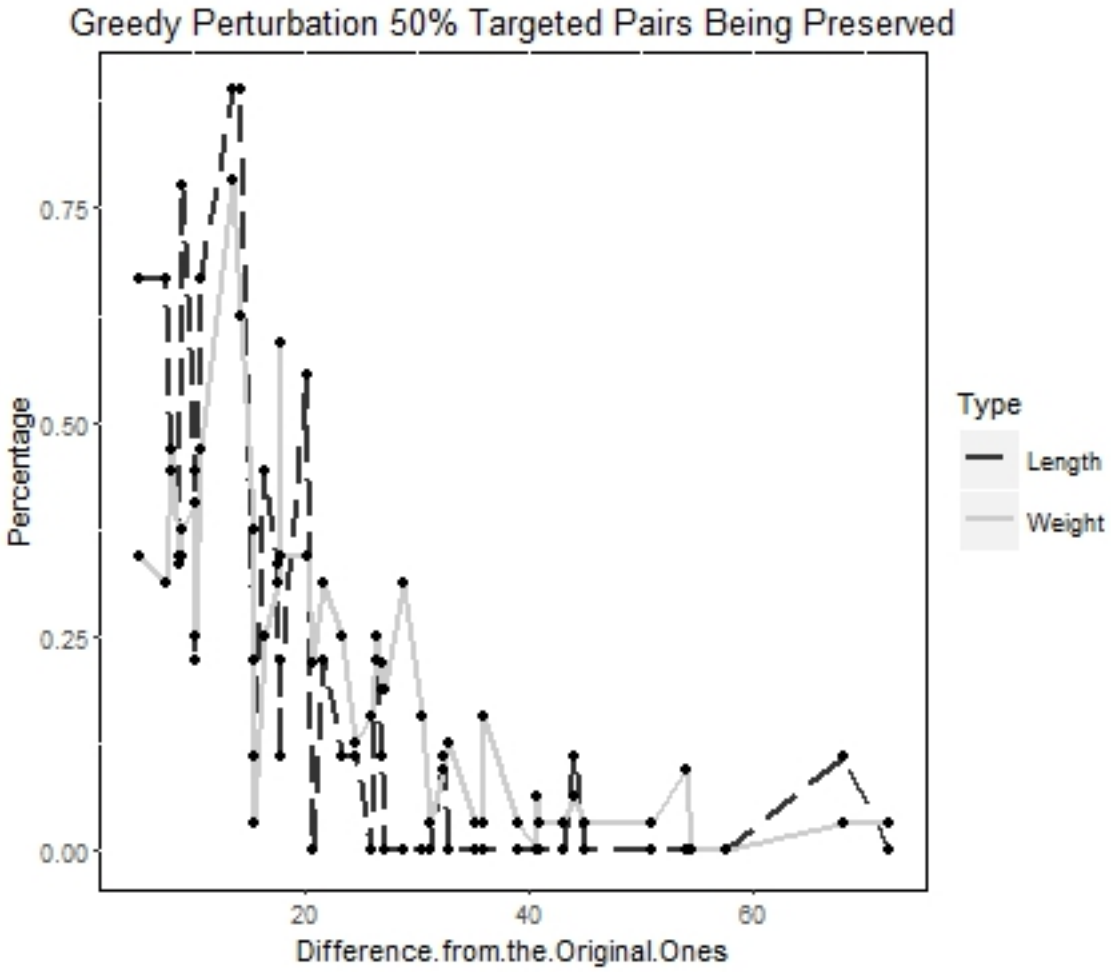

Figure 5.10: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 50% targeted pairs being preserved

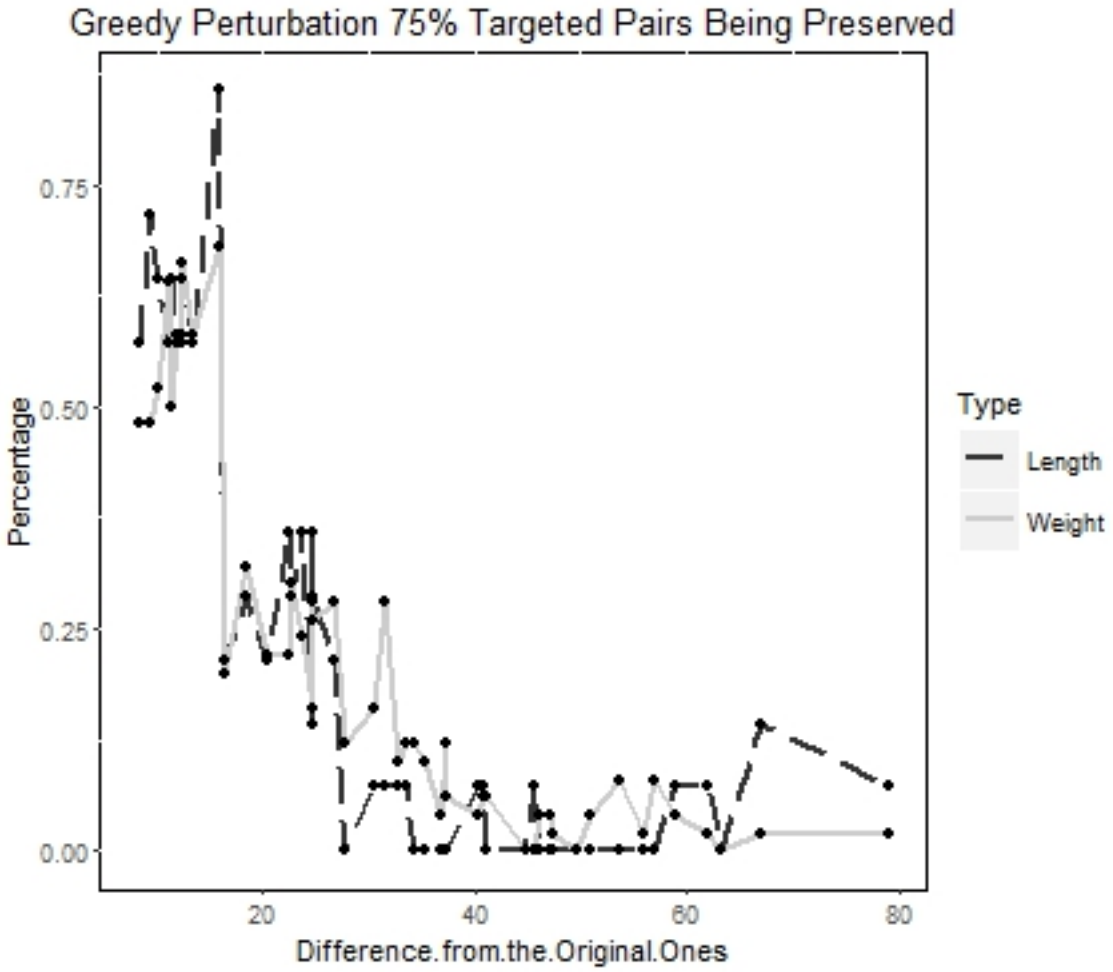

Figure 5.11: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 75% targeted pairs being preserved

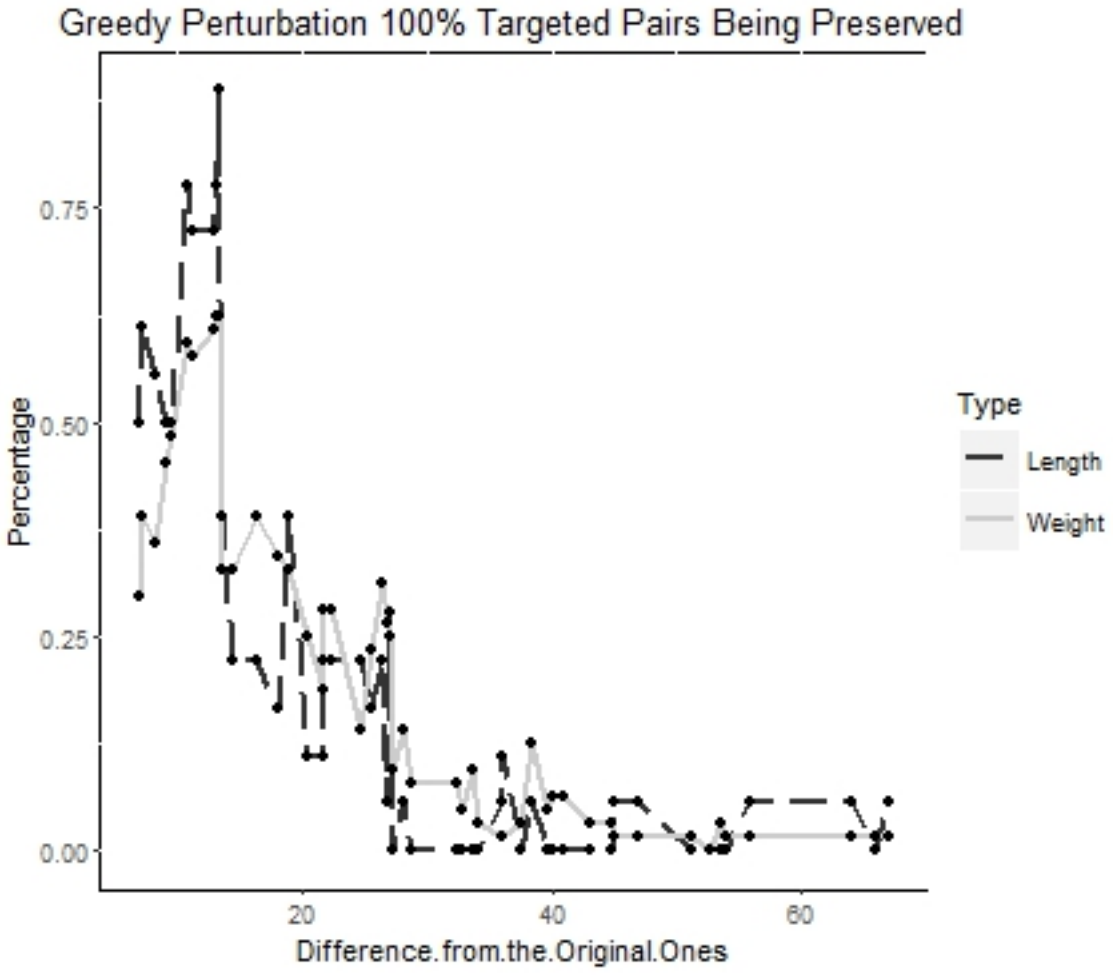

Figure 5.12: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 100% targeted pairs being preserved

Chapter 6

Conclusion and Future Work

In consideration of the privacy issue in social network data mining techniques, the links weights between social network entities are sensitive in some cases such as the business transaction expenses. This work addresses a balance between protection of sensitive weights of network links (edges) and some global structure utilities such as the shortest paths and the corresponding shortest path lengths.

Two perturbation strategies, Gaussian randomization multiplication and greedy perturbation algorithm, are presented to perturb individual (sensitive) edge weights and try to keep exactly the same shortest paths as well as their lengths close to those of the original social network. The experimental results demonstrate that the two perturbation strategies maintains the closeness of the length between the Original and Perturbed Social Network.

Bibliography

[1] A. Evfimievski. *Randomization in Privacy Preserving Data Mining*. ACM SIGKDD Explorations Newsletter, 4(2):43-48, 2002.

[2] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. *Link Privacy in Social Networks*. In Proceedings of IEEE 24th International Conference on Data Engineering (ICDE 2008), pp. 1355-1357, Cancun, Mexico, Apr 7-12, 2008.

[3] B. T. Messmer, and H. Bunke. *Efficient Subgraph Isomorphism Detection: a Decomposition Approach*. IEEE Transactions on Knowledge and Data Engineering, 12(2): 307-323, 2000.

[4] B. Zhou, and J. Pei. *Preserving Privacy in Social Networks Against Neighborhood Attacks*. In Proceedings of the 24th International Conference on Data Engineering (ICDE08), Cancun, Mexico, pp. 506-515, April 2008.

[5] B. Zhou, G. Yan, and B.Wang. *Maximal Planar Networks with Large Clustering Coefficient and Power-Law Degree Distribution*. Physical Review E 71, no. 4, 2005.

[6] B. Zhou, J. Pei, and W. S. Luk. *A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data*. CM SIGKDD Explorations, 10(2): 12-22, December, ACM Press, 2008.

[7] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. *Tools for Privacy Preserving Distributed Data Mining*. ACM SIGKDD Explorations, 4(2):1-7, 2003.

[8] E. M. Rogers. *Diffusion of Innovations, 4th ed*. Simon Shuster, Inc, 1995.

[9] E. M. Rogers. *Diffusion of Innovations, 5th*. Simon Shuster, Inc, 2003.

[10] E. Zheleva, and L. Getoor. *Privacy in Social Networks, Synthesis Lectures on Data Mining Series*. ,Book published by Morgan and Claypool Publishers, 2012.

[11] E. Zheleva, and L. Getoor. *Preserving the Privacy of Sensitive Relationships in Graph Data*. In Proceedings of the 1st ACM SIGKDD International Workshop on Privacy, Security, and Trusting of KDD, San Jose, California, pp. 153-171, Aug 2007.

[12] H. Kargupta, S. Datta, Q.Wang, and K. Sivakumar. *n the Privacy Preserving Properties of Random Data Perturbation Techniques*. In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 99-106, Melbourne, Florida, 2003.

[13] A. Inkpen. *The Japanese Corporate Network Transferred to North America: Implications for North American Firms*. The International Executive, 36(4): 411-433, 1994.

[14] J. Tang, D. Zhang, and L. Yao. *Social Network Extraction of Academic Researchers*. In Proceedings of 2007 IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, Oct, 2007.

[15] K. Chen, G. Sun, and L. Liu. *Towards Attack-Resilient Geometric Data Perturbation*. In Proceedings of the 2007 SIAM International Conference on Data Mining (SDM 2007), pp. 78-89, Minneapolis, MN, 2007.

[16] K. Liu, and E. Terzi. *Towards Identity Anonymization on Graphs*. In Proceedings of SIGMOD 2008, pp. 93-106, Vancouver, BC, Canada, Jun 9-12, 2008.

[17] L. Backstrom, C. Dwork, and J. Kleinberg. *Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography*. In Proceedings of the 16th International Conference on World Wide Web, Alberta, Canada, pp. 181-190, 2007.

[18] L. Sweeney. *Guaranteeing Anonymity When Sharing Medical Data, the Datafly System*. Journal of the American Medical Informatics Association, Suppl. S, pp. 51-55, 1997.

[19] L. W. Young, and R. B. Johnston. *The Role of the Internet in Business-to-Business Network Transformations: a Novel Case and Theoretical Analysis*. Information Systems and e-Business Management, 1(1): 73-91, 2003.

[20] Lian. Liu. *Privacy Preserving Data Mining for Numerical Matrices, Social Networks, and Big Data*. Theses and Dissertations–Computer Science,University of Kentucky, 49-72, 2015.

[21] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. *Anonymizing Social Networks*. University of Massachusetts, Amherst, MA, Tech. Rep. 07-19, 2007.

[22] P. Tendick. *Optimal Noise Addition for Preserving Confidentiality in Multivariate Data*. Journal of Statistical Planning and Inference, 27(2): 341-353, 1991.

[23] R. Agrawal, R. Srikant, and D. Thomas. *Privacy-Preserving Data Mining*. ACM SIGMOD Record, 29(2): 439-450, 2000.

[24] J. M. Read and M. J. Keeling. *Disease Evolution on Networks: the Role of Contact Structure*. In Proceedings of the Royal Society B: Biological Sciences, 270: 699- 708, 2003.

[25] S. Guo, and X. Wu. *On the Use of Spectral Filtering for Privacy Preserving Data Mining*. In Proceedings of the 21st ACM Symposium on Applied Computing, pp. 622-626, Dijon, France, 2006.

[26] S. Guo, X. Wu, and Y. Li. *On the Lower Bound of Reconstruction Error for Spectral Filtering Based Privacy Preserving Data Mining*. Knowledge Discovery in Databases: PKDD 2006, 4213: 520-527, 2006.

[27] S. Xu, J. Zhang, D. Han, and J. Wang. *Data Distortion for Privacy Protection in a Terrorist Analysis System 2005*. In Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics, pp. 459-464, Atlanta, GA, 2005.

[28] S. Xu, J. Zhang, D. Han, and J. Wang. *Singular Value Decomposition Based Data Distortion Strategy for Privacy Protection*. Knowledge and Information Systems, 10(3): 383-397, 2006.

[29] T. Zhou, G. Yan, and B.Wang. *Maximal Planar Networks with Large Clustering Coefficient and Power-Law Degree Distribution*. Physical Review E 71, no. 4, 2005.

[30] Z. Huang, W. Du, and B. Chen. *Deriving Private Information from Randomized Data*. In Proceedings of the 2005 ACM SIGMOD Conference, pp. 37-48, Baltimore, MD, 2005.