International Journal of Computer Applications (0975 – 8887)
Volume 130 – No.3, November 2015

Preview and Maximizing the Data Utility in Privacy Preserving Social Network

N. Sridhar
Researcher
GITAM University
Visakhapatnam
India

Y. Srinivas, PhD
Professor
GITAM University
Visakhapatnam
India

ABSTRACT

The field of Preserving Privacy in Data Mining is gaining momentum in the recent times as the data sets are more open towards mining by organizations and academic institutes. Ensuring privacy of the data before publishing it to wider audience is always an open challenge. So far the research is more towards performing perturbation on relational data. The gaining popularity of Social Networks leads to new problems and challenges. Different methodologies have been proposed in an attempt to preserve the privacy of the published datasets for Social Networks. There have been many techniques evolved to exploit the perturbed data and get some sensitive knowledge for Social Network data as well. So on the process of ensuring more privacy, the data perturbation techniques also became complex and more distortive in nature. The Data Utility is level of usefulness of the distorted data. The study of data utility comes into play as the distortion level increases. A model has been proposed in this paper to perform pre perturbation analysis for measuring the data utility and using this as an input to choose the right methodology for achieving the data utility as well as maintaining the privacy of the social network datasets which are graph kind in nature.

General Terms

Privacy Preserving Data Mining in Social Networks

Keywords

Maximizing Data Utility, Privacy Preserving, Social Networks.

1. INTRODUCTION

Data Mining takes key position in today's data world where it has been extensively used in many institutions. The current strategy of mining activities needs to exchange data for mutual benefit. This leads to concern over privacy issues in the recent times. It has also been pointed out that a possible threat of leaking sensitive data of an individual when the data is published to outside word. Several methods came in to picture to deal with privacy in networks. Anonymization and perturbation are the most common methods to preserve the privacy of the datasets. But the natural side effect of privacy preservation is data quality loss. The loss of specific details about certain individuals may affect the data utility and in some cases the data may become completely meaningless. The cryptographic methods also came in to existence which completely anonymize the dataset and which makes the dataset difficult to use. Maintaining the quality of the resultant data is a huge challenge along with privacy protection. The objective of this research is to find an optimum balance between privacy of the dataset and utility while publishing the dataset. [1]

In today's world, social networks are evolving rapidly and have received lot of attention in research and development. Networks which represent friendship, profession, community, disease, and co-authorships are the popular ones. Social networks are denoted using graph data structures. In a Social Network, individuals (or group profiles) in a network represented using vertices. These vertices are called social profiles, while edges represent relationships between these social profiles. [2]

The world of information science is also focusing the analysis on these social networks. Companies like Google, Facebook, Yahoo, and Twitter are doing their research on this and encouraging the public by placing their data to wider audience for further research. The users of the social network are from all kind of age groups. This makes the analysis on this data can be used for the enhancing their customer base for the advertisements. The new potential customers are identified as a result of this analysis. These social networks are becoming as backbones for companies who make their revenue by selling products and services by running their advertisement campaigns. [3]

2. TRENDS OF PRIVACY IN SOCIAL NETWORKS

To preserve privacy in the data sets there are several methods introduced. The methods for data anonymization/perturbation/modification which are originally meant for relational dataset also have been adopted for network data as well. These methods can be classified in to two groups: Methods based on Graph Perturbation and Greedy Graph Modification Method. [4-6]

2.1 Methods based on Graph Perturbation

A graph perturbation method transforms the network graph by modifying graph vertices and edges. The modification could be inserting/deleting the nodes/edges. The modification can be conducted in three ways and correspondingly there are three sub-categories in these types of modifications: Optimization Graph Perturbation Method, Methods based on Clustering, Randomized Graph Perturbation Method.

2.2 Anonymization Methods

One of the popular methods for the relational dataset is k-anonymity. The *k*-degree anonymization is the network variation of existing k-anonymity. A social network graph is said to be k-degree anonymous if every user in the graph is having k-1 friends. So the chance of re identifying the user from the published data is 1/k. To make it more private the values of k- is chosen sufficiently large.

2.3 Methods based on Clustering

A clustering-based method groups users and edges into cluster and anonymizes the respective sub-graph into a vertex group. By grouping nodes this way the profiles can be kept private. The clustering can be further classified as edge clustering, vertex clustering, vertex-attribute mapping clustering, vertex and edge clustering methods.

2.4 Randomized Graph Perturbation Method

This approach will transform the original graph by performing sequence of k edge removal operations followed by k edge additions. Edge insertions and deletions are performed at random portions of the graph. The value of k made public at the time of publishing the dataset. At the end of the operation the edge count remain same. But the operation doesn't affect the nodes in any form.

2.5 Greedy Graph Modification Method

This method can be used in anonymizing social network data primarily to prevent neighborhood attacks. This approach well suited for graphs whose structure remains same, i.e., graph perturbation methods without adding new nodes. It's also suited for relationships which are weighted in nature.

3. GRAPH MEASUREMENT TECHNIQUES

3.1 Degree Distribution

The degree of a vertex in a graph is total count of adjacent nodes to a node or the total number of edges the node has to other nodes. For an undirected graph, the **vertex/node degree** of a vertex *v* is the number of vertices adjacent to v. Self loops are not considered, as the social network will not have any self loops.

The **node degree distribution** gives the number of nodes with degree *i* for $i = 0,1,\ldots$.

The degree distribution $P(d)$ of a graph is defined to be the fraction of nodes in the graph with degree d. Thus if there are n nodes in total in a network and n_d of them have degree d,

$P(d) = n_d/n$. [7-10]

3.2 Network Diameter

Diameter is the maximum length of shortest paths between pair of vertices. If a graph is having disconnected components, then the diameter is calculated for each component. The diameter is the maximum diameter of all diameters of all connected components. The network diameter indicates small-world properties of the analyzed social graph. [7-10]

3.3 Clustering Coefficients

Clustering coefficient is the total count on the number of triangles in an undirected graph. The clustering coefficient C_u of a node $\boldsymbol{u}$ is defined as $C_u = 2e_u/(k_u(k_u\text{-}1))$, where k_u is the number of neighbors of $\boldsymbol{u}$ and e_u is the number of connected pairs between all neighbors of $\boldsymbol{u}$. In other words, the clustering coefficient of a node is the number of triangles (3-loops) that pass through this node, relative to the maximum number of 3-loops that could pass through the node. The clustering coefficient of a node is always a value in the interval 0 and 1. The clustering coefficient of a graph gives the measure the relative frequency of triangles.

The clustering coefficient is a fraction E_a / E_p,

E_a : the actual number of edges between the neighbors of u,

E_p : the maximum number of edges that could possibly exist between the neighbors of u.

$$C = \frac{1}{n} * \sum_{i=1}^{n} C_i$$

The distribution on *average clustering coefficient* gives the clustering coefficient with respective to the neighbors and it is calculated as the average of the clustering coefficients for all vertices *v* with *k* neighbors for $k = 2,\ldots$. The *graph clustering coefficient* is the average of the clustering coefficients for all vertices in the entire graph. [7-10]

3.4 Betweenness Centrality

Betweenness centrality measures the node's centrality in entire network network. This can be computed only for graphs without multiple edges. For a node v, the centrality is the number of shortest paths from all nodes to all others that passes through node v. *Betweenness centrality* gives measure on a particular node about the load and its importance in the network. Betweenness centrality is handy when it comes to deal with complex networks. It is widely applied in the areas of computers, social networks, and scientific. [7-10]

The *betweenness centrality* depicts the amount of control that this vertex poses over the interactions of other nodes in the network. This measure identifies nodes that connect communities (sub graphs which are dense), rather than nodes that present inside a community graph.

The *betweenness centrality* of a node v is calculated by the formula:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

σ_{st} : Total number of shortest paths starting from vertex s to vertex t

$\sigma_{st}(v)$: The total number of shortest paths that pass through vertex v.

Observe that the value of ***g(v)*** of a vertex scales with the number of pairs of vertices. It is rescaled by dividing the number of pairs of nodes not considering v, so that $g \in [0,1]$. Therefore for directed graphs the division is done by $(N-1) \times (N-2)$ and for undirected graphs $(N-1) \times (N-2)/2$, where N is the number of vertices in the connected component.

$$normal(g(v)) = \frac{g(v) - min(g)}{max(g) - min(g)}$$

This results in:

$$\max(normal) = 1$$
$$\min(normal) = 0$$

3.5 Closeness centrality

For a vertex ***v*** the closeness centrality $C_c(v)$ is defined as the inverse of the average length **on** the shortest path**.** Closeness centrality gives a measure on about how the data spreads rapidly from a given vertex to remaining reachable vertices in the graph. The closeness centrality of isolated node is always equal to 0. In general the closeness centrality of each node always falls in the range between 0 and 1. [7-10]

$$C_c(v) = \frac{1}{average(L(p,q))}$$

where $L_{(p,q)}$ is the length of the shortest path between two nodes p and q.

4. PROBLEMS WITH CURRENT METHODS OF PRIVACY

There are numerous data perturbation techniques evolved such as k-anonymity, l-diversity, and random modification techniques. All of these are focused on increasing the level of privacy to counter the different attack scenarios. Most of them suffer from addressing the data quality aspect of the resultant data. There is very little attention on using the data utility as an input to increase or decrease the level of perturbation to the dataset.

5. SOLUTION

The process to finding data utility and fine tuning the amount of noise to be added to the dataset based on data utility measure is proposed here. It is an iterative approach. It has broadly four phases.

5.1 Input Collection

This phase will collect the input from the user. The input is the number of iterations/threshold as to decide when the perturbation process needs to terminated as completed. The metrics to be used for evaluating the graphs, perturbation technique that is going to be used also collected in this phase. The metrics for the graph analysis under consideration are Degree Distribution, Clustering Coefficient, and Diameter. The perturbation techniques considered for evaluation are k-anonymity, random node/edge modification.

5.2 Pre-Analysis

The dataset under consideration is fed to the Graph Miner tool for evaluating the initial set of metrics. These metrics will be used as base for comparing the metrics obtained in the next versions of data set during the process.

5.3 Data Perturbation

This phase will modify the graph data using one of the algorithms that are already available. The algorithms that can be considered are k-anonymity, random node/edge addition.

5.4 Metric Evaluation and Iteration

After applying the data perturbation, the metrics are evaluated on the perturbed graph again. The graph measurement techniques that are mentioned in 3rd section of is paper are used primarily for evaluating the perturbed graph. This phase will compare the resultant metric values with the values that are obtained from the Pre-Analysis. If the resultant metrics exceed the threshold metric then the process will come to an end. If the computed metric is below the threshold then the data perturbation is repeated again and the metrics are computed again. This process of data perturbation and metric evaluation are repeated until the predetermined iterations or until the metric values crosses the user supplied threshold.

For a given graph metric $\mathbf{M_i}$, the Metric Change Rate is computed, which is basically the growth or loss percentage.

This is denoted as $\Delta MCR_i\ (G, G')$

where G : Original Graph
G': Perturbed Graph

Cumulative Metric Change Rate across all Graph Metrics is the average of all metrics change, as

$$CMCR = \frac{1}{|M|} * \sum_{i=1}^{n} \Delta MCR_i\ (G, G')$$

Where $|M|$ = Number of Graph Metrics Considered

ΔMCR_i= i-th graph metric change rate

Three variations of the algorithms are proposed here in this paper. The first one just evaluates the Data Utility based on selected graph metrics. The second one evaluates the Data Utility using the number of used supplied iterations based on selected graph metrics. The third one evaluates the data utility using the user supplied threshold based on selected graph metrics.

Algorithm 1 : Data Utility Measurement

```
Input:
    ds_0 : Dataset Initial
    pAlg : Perturbation Algorithm
    numIter : Number of iterations
    selGraphMetrics : Selected Graph Metrics.
    Ds_new : Dataset Perturbed after balancing the Data Utility.
Output:
    MCRi : Metric Change Rate for Metric located at i from selGraphMetrics array.
    CMCR : Cumulative Metric Change Rate

CMCR := 0;
selGraphMetrics:= {m1, m2, m3, ...};
  for mi := 1 to | selGraphMetricss | do
    metricValue_0 := evalueteMetric(mi, ds_0);
    ds_new := ApplyPerturbarion(ds_0, pAlg);
    metricValue[selGraphMetrics[mi]] := evalueteMetric(mi, ds_new);
    MCRi (metricValue_0, metricValue_new) := computeMetricChangeRate(metricValue_0, metricValue_new);
    cmtr := cmtr + MCRi;
  end;
 /* Cumulative Metric Change Rate Average */
  CMCR := CMCR / | selGraphMetricss |;
  Output: CMCR ;
 end;
```

Algorithm 2: Iteration driven Data Utility

```
Input:
    ds_0 : Dataset Initial
    pAlg : Perturbation Algorithm
    numIter : Number of iterations
    selGraphMetricss : Selected Graph Metrics.
Output:
    ds_iter : Dataset Perturbed after balancing the Data Utility at each stage.
    metricValue_iter : Metric values at every iteration.
    MCRi : Metric Change Rate for Metric located at i from selGraphMetrics array.
    CMCR_iter : Cumulative Metric Change Rate at the Iteration.

 selGraphMetricss:= {m1, m2, m3, ...};
 metricValue_0 := evalueteMetric(mi, ds_0);
 for iter := 1 to numIter do
    begin
    CMCR_iter:= 0;
    for ci := 1 to | selGraphMetricss | do
    begin
        ds_iter := ApplyPerturbarion(ds_iter-1, pAlg);
        metricValue_iter[selGraphMetrics[mi]] :=
evalueteMetric(mi, ds_iter);
        MCRi(metricValue_iter,, metricValue_0)
:=computeMetricChangeRate(metricValue_iter, metricValue_0)
        CMCR_iter  := CMCR_iter  + MCRi (metricValue_iter,
metricValue_0);
     end;
     Output : CMCR_iter/ |selGraphMetricss | ;
   end;
end;
```

<u>Algorithm 3: Threshold driven Data Utility</u>

<u>*Input:*</u>

```
      ds_0      : Dataset Initial
      pAlg      : Perturbation Algorithm
     limitDu : Threshold to stop
     numIter : Number of iterations
     selGraphMetricss : Selected Graph Metrics.
```

<u>*Output:*</u>

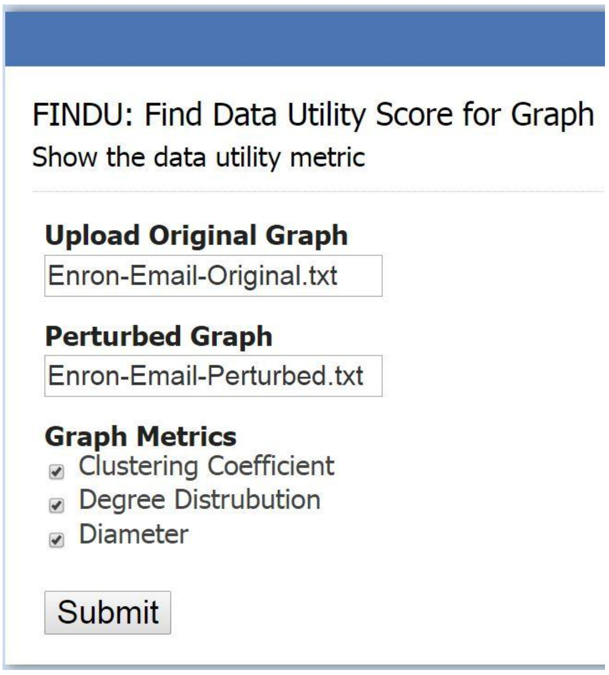
```
      ds_iter     : Dataset Perturbed after balancing the Data
Utility at each stage.
     metricValue_iter : Metric values at every iteration.
     MCRi    : Metric Change Rate for Metric located at i from
selGraphMetrics array.
     CMCR_iter   : Cumulative Metric Change Rate at the
Iteration.

selGraphMetricss:= {m1, m2, m3, …};
metricValue_0 := evalueteMetric(mi, ds_0);

 for iter := 1 to numIter do
   begin
   CMCR_iter  := 0;
   for ci := 1 to | selGraphMetricss |  do
   begin
        ds_iter  := ApplyPerturbarion(ds_iter-1, pAlg);
        metricValue_iter[selGraphMetrics[mi]] :=
evalueteMetric(mi, ds_iter);
        MCRi(metricValue_iter,, metricValue_0)
:=computeMetricChangeRate(metricValue_iter, metricValue_0)
        CMCR_iter  := CMCR_iter  + MCRi (metricValue_iter,
metricValue_0);
     end;
     if ( CMCR_iter  > limitDu )
        begin
                  Output : (CMCR_iter / | selGraphMetricss |  );
                  Output : ds_iter;
                  /* Threshold reached to stop the algorithm */
                 Exit;
         end;
   end;
end;
```

6. IMPLEMENTATION AND DATASET

The Enron email communication dataset has been used for evaluating the algorithm. Two random subsets are taken from this dataset. In this prototype the parameters considered are Clustering Coefficients, Degree Distribution, and Diameter. The perturbation algorithms that are considered are Random Edge/Node modification and k-Anonymity.


FINDU: Find Data Utility Score for Graph
Show the data utility metric

Upload Original Graph
Enron-Email-Original.txt

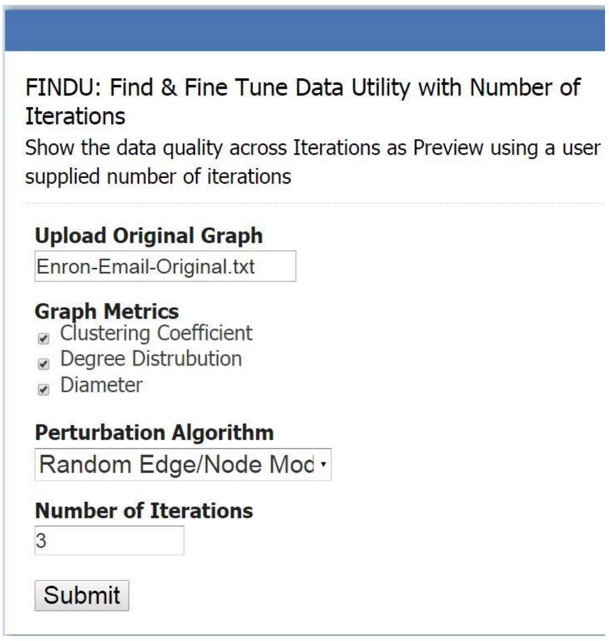
Perturbed Graph
Enron-Email-Perturbed.txt

Graph Metrics
✓ Clustering Coefficient
✓ Degree Distrubution
✓ Diameter

Figure 1: Data Utility computation of dataset

Table 1: Data Utility and graph metrics computation

Graph Metrics	Original Dataset	Perturbed Dataset	Change %
Nodes	36692	39521	92.29
Edges	183831	253726	61.98
Degree Distribution	5.01	6.42	71.86
Diameter	11	12	90.91
Clustering Coefficient	0.49	0.47	104.08
Data Utility is			88.95


FINDU: Find & Fine Tune Data Utility with Number of Iterations
Show the data quality across Iterations as Preview using a user supplied number of iterations

Upload Original Graph
Enron-Email-Original.txt

Graph Metrics
✓ Clustering Coefficient
✓ Degree Distrubution
✓ Diameter

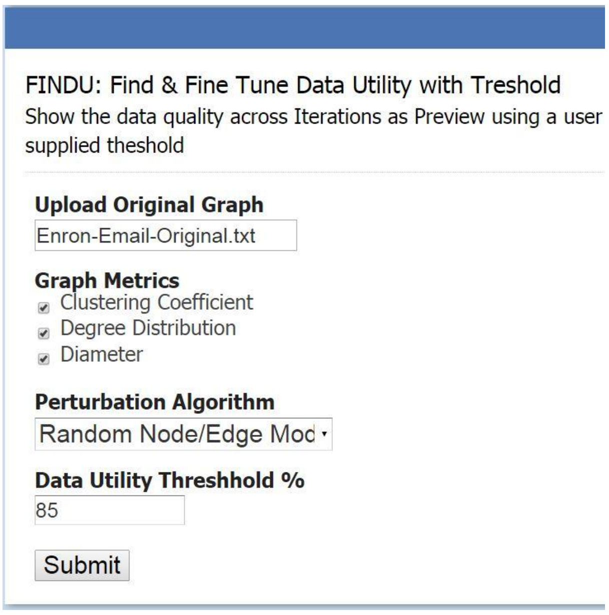
Perturbation Algorithm
Random Edge/Node Mod

Number of Iterations
3

Figure 2: Data Utility computation using 3 iterations

Table 2: Data Utility computation using 3 iterations

Graph Metrics	Pre-Analysis	Iteration-1	Iteration-2	Iteration-3
Nodes	36692	39521	40496	40983
Edges	183831	193726	205819	207883
Degree Distribution	5.01	4.9	5.08	5.07
Diameter	11	12	13	14
Clustering Coefficient	0.49	0.51	0.53	0.56
Data Utility		96.34	90.75	85.75


FINDU: Find & Fine Tune Data Utility with Treshold
Show the data quality across Iterations as Preview using a user supplied theshold

Upload Original Graph
Enron-Email-Original.txt

Graph Metrics
✓ Clustering Coefficient
✓ Degree Distribution
✓ Diameter

Perturbation Algorithm
Random Node/Edge Mod

Data Utility Threshhold %
85

Figure 3: Data Utility computation with user specified threshold

Table 3: Data Utility computation with user specified threshold

Graph Metrics	Pre-Analysis	Iteration-1	Iteration-2	Iteration-3
Nodes	23748	24809	25891	26241
Edges	128423	138653	148929	158323
Degree Distribution	5.41	5.59	5.75	6.03
Diameter	7	8	8	9
Clustering Coefficient	0.54	0.56	0.57	0.58
Data Utility		92.89	91.29	84.19

7. LIMITATIONS

Currently this algorithm works only with non weighted graphs datasets. The algorithm can be extended to use other graph centrality measures. The perturbation techniques used for evaluation can also be extended to l-diversity, random node/edge addition, clustering based modification, greedy modification of node/edges.

8. CONCLUSION

Privacy and Data Utility are quite orthogonal. In order to balance both it is essential that the Data Utility part measured properly on the perturbed dataset. This paper introduced a new Data Utility measurement based on graph statistics and used to evaluate the data utility iteratively. This procedure can be used as benchmark process to decide when to stop further perturbation process. Also it has been shown that the methodology is used as a validation measure to evaluate the data utility of existing perturbation algorithms.

9. REFERENCES

[1] Mohd Izuan Hafez Ninggal and Jemal Abawajy: Privacy Threat Analysis of Social Network Data , Algorithms and Architectures for Parallel Processing Lecture Notes in Computer Science Volume 7017, 2011, pp 165-174.

[2] Bin Zhou, A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data, ACM SIGKDD Explorations Newsletter archive Volume 10 Issue 2, December 2008.

[3] Elena Zheleva, Lise Getoor, “Privacy in Social Networks: A Survey”, In: Social Network Data Analytics, Springer US, pp 277-306, 2011.

[4] Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham, “Preventing Private Information Inference Attacks on Social Networks”, In: IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 8, pp 1849-1862, 2013.

[5] Yuan Cheng, Ravi Sandhu, “Preserving User Privacy from Third-party Applications in Online Social Networks, In Proc. of 22nd international conference on World Wide Web companion, Geneva, Switzerland, pp 723-728, 2013.

[6] K. Liu, E. Terzi, "Towards identity anonymization on graphs," In Proc. of 2008 ACM SIGMOD International conference on Management of data, Vancouver, Canada, 2008

[7] Facebook (2013, Facebook Statistic. Available: http://www.facebook.com/press/info.php/statistics

[8] Yoon, J., Blumer, A., Lee, K.: An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness.

[9] Newman, M.E.J.: A measure of betweenness centrality based on random walks. arXiv (2003) cond-mat/0309045.

[10] Degree, closeness, and betweenness: Application of group centrality measurements to explore macro-disciplinary evolution diachronically. In Proceedings of ISSI 2011, Durban, South Africa