Analysis Of Privacy Preserving Random Perturbation Techniques: Further Explorations *

Haimonti Dutta
Department of CSEE
University of Maryland
Baltimore County
Baltimore, Maryland 21250
hdutta1@cs.umbc.edu

Hillol Kargupta
Department of CSEE
University of Maryland
Baltimore County
Baltimore, Maryland 21250
hillol@cs.umbc.edu

Souptik Datta
Department of CSEE
University of Maryland
Baltimore County
Baltimore, Maryland 21250
souptik1@cs.umbc.edu

Krishnamoorthy Sivakumar
School of EECS
Washington State University
Pullman, Washington, USA
siva@eecs.wsu.edu

ABSTRACT

Privacy is becoming an increasingly important issue in many data mining applications, particularly in the security and defense area. This has triggered the development of many privacy-preserving data mining techniques. A large fraction of them uses randomized data distortion techniques to mask the data for preserving the privacy. They attempt to hide the sensitive data by randomly modifying the data values using additive noise. This paper questions the utility of such randomized data distortion technique for preserving privacy in many cases and urges caution.It notes that random objects (particularly random matrices) have "predictable" structures in the spectral domain and then offers a random matrix-based spectral filtering technique to retrieve original data from the data-set distorted by adding random values. It extends our earlier work questioning the efficacy of random perturbation techniques using additive noise for privacy-preserving data mining in continuous valued domain and presents new results in the discrete domain. It shows that the growing collection of random perturbation-based "privacy-preserving" data mining techniques may need a careful scrutiny in order to prevent privacy breaches through linear transformations. The paper also presents extensive experimental results in order to support this claim.

*This is an extension of the paper "Random Data Perturbation Techniques and Privacy Preserving Data Mining", accepted for publication in the proceedings of the IEEE International Conference on Data Mining, 2003. Therefore, it contains some common material.

WPES'03, October 30, 2003, Washington, DC, USA.

Categories and Subject Descriptors

E.m [**Data**]: Miscellaneous

General Terms

Security

Keywords

Privacy, security, random-perturbation

1. INTRODUCTION

Many security and counter-terrorism-related decision support applications need data mining techniques for identifying emerging behavior, link analysis, building predictive models, and extracting social networks. They often deal with multi-party databases/data-streams where the data are privacy sensitive. Financial transactions, health-care records, and network communication traffic are a few examples. Figure 1 depicts the data sources of a typical security screening application where the data may be privacy sensitive. Mining the data in such applications requires algorithms that are sensitive to privacy issues.

There is a growing body of literature on data mining techniques [1, 12, 15] that try to protect the data privacy with varying degrees of success [14]. These algorithms try to extract the data patterns without directly accessing the original data and attempt to guarantee that the mining process does not get sufficient information to reconstruct the original data. Some of these techniques are related to the general framework of secure multi-party computation introduced elsewhere [22].

This paper considers the problem of mining multi-party privacy-sensitive data using random perturbation-based techniques. It presents a negative result that may in fact help the field in a positive way—leading toward a new class of more robust privacy-preserving data mining algorithms. It considers random additive perturbations used by many existing privacy-preserving data mining techniques (e.g. [1,

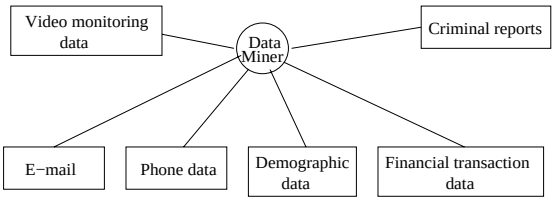

Figure 1: Data sources for a typical security screening application. Many of these sources deal with privacy sensitive data.

7, 8]) that try to preserve data privacy by adding random noise while making sure that the underlying distribution is still accurately preserved. It points out that in many cases, the original data can be easily filtered out from the perturbed data using a spectral decomposition technique. This paper argues that these additive random perturbation-based techniques may compromise data privacy under linear transformations of the perturbed data in many cases. This paper extends our earlier work [14] and offers new results for discrete data.

Section 2 briefly reviews the different types of data considered in this paper for privacy protection—continuous valued data, discrete valued transaction data, and graph structured data. This section also reviews some of the existing privacy-preserving data mining techniques for these data types. Section 3 presents an overview of the random matrix-based filtering technique used in this paper for filtering out the random additive noise from the perturbed data. Section 4 presents several experimental results. Finally, Section 5 concludes this paper.

2. PRIVACY PRESERVATION OF CONTINUOUS AND DISCRETE DATA

Most security applications deal with heterogeneous data from different sources. This section considers some of the common data types that these applications usually deal with and discusses some of the existing random perturbation-based privacy-preserving data mining algorithms for each of these domains.

It first considers continuous valued data and a random data perturbation technique for privacy preservation of this type of data. Next it considers discrete valued graph structured and transaction data for privacy-preserving applications. This paper argues that the privacy protection of these randomized perturbation-based techniques may be compromised by a spectral filtering technique discussed later in this paper.

2.1 Continuous Valued Data

Continuous valued data are widely prevalent among different data mining applications and security applications are no exceptions. Several randomized techniques have been proposed for privacy preserving data mining of continuous data. Random additive perturbation [1] is one of them that is directly relevant to the work presented in this paper. This section presents a brief review of this technique. It works by adding "randomly" generated noise from a given distribution to the values of sensitive attributes.The following sections discuss the data perturbation technique and the estimation of density functions from the perturbed data set.

2.1.1 Perturbing the Data

The random additive perturbation method attempts to preserve privacy of the data by modifying values of the sensitive attributes using a randomized process. The authors of [1] explore two possible approaches — Value-Class Membership and Value Distortion — and emphasize the Value Distortion approach. In this approach, the owner of a dataset returns a value $u_i + v$, where u_i is the original data, and v is a random value drawn from a certain distribution. The n original data values $u_1, u_2, \ldots, u_n$ are viewed as realizations of n independent and identically distributed (i.i.d.) random variables U_i, $i = 1, 2, \ldots, n$, each with the same distribution as that of a random variable U. In order to perturb the data, n independent samples $v_1, v_2, \ldots, v_n$, are drawn from a distribution V. The owner of the data provides the perturbed values $u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n$ and the cumulative distribution function $F_V(r)$ of V. The reconstruction problem is to estimate the distribution $F_U(x)$ of the original data, from the perturbed data.

2.1.2 Estimation of Density Function from the Perturbed Dataset

Estimating the density function is a common problem in data mining and security applications are not an exception. The density information can be used for clustering, classification, and other related problems. Perturbed data using additive noise allows estimating the underlying density function reasonably well.

The authors [1] suggest the following method to estimate the distribution $F_U(u)$ of U, given n independent samples $w_i = u_i + v_i$, $i = 1, 2, \ldots, n$ and $F_V(v)$. Using Bayes' rule, the posterior density function $f'_U(u)$ of U, given that $U+V = w$, can be written as

$$f'_U(u) = \frac{f_V(w-u) f_U(u)}{\int_{-\infty}^{\infty} f_V(w-z) f_U(z) dz},$$

where $f_U(.)$, $f_V(.)$ denote the probability density function of U and V respectively. If we have n independent samples $u_i + v_i = w_i$, $i = 1, 2, \ldots, n$, the corresponding posterior density can be obtained by averaging:

$$f'_U(u) = \frac{1}{n} \sum_{i=1}^{n} \frac{f_V(w_i-u) f_U(u)}{\int_{-\infty}^{\infty} f_V(w_i-z) f_U(z) dz}. \quad (1)$$

For sufficiently large number of samples n, we expect the above density function to be close to the real density function $f_U(u)$. In practice, since the true density $f_U(u)$ is unknown, we need to modify the right-hand side of Equation 1. The authors suggest an iterative procedure where at each step $j = 1, 2, \ldots$, the posterior density $f_U^{j-1}(u)$ estimated at step $j-1$ is used in the right-hand side of Equation 1. Detailed description of this approach can be found elsewhere [1]. A related approach to estimate the density function and a discussion on quantifying privacy can be found in [2]. The following section considers discrete data types.

2.2 Discrete Valued Transaction Data

Association rule learning is a widely popular technique for link analysis in data mining applications. This section

considers market basket transaction data in Boolean representations and a recently proposed randomized perturbation technique for privacy preserving association rule learning.

Market basket data is usually a collection of transactions, where each transaction contains some product ids that are sold, and quantity sold [6]. The transactions can be represented in a tabular form, where each column represents one product id, and each row represents one transaction. For the sake of simplicity, let us consider transactions that only keep track of whether or not an item was purchased, not the quantity sold. In that case, we can represent a transaction using an ℓ-dimensional Boolean string where ℓ is the maximum number of different items that are available to a customer. The i-th bit will be set to 0 if the corresponding item is not sold in that transaction; it will be set to 1 otherwise. Therefore, one can represent a collection of m transactions using an $m \times \ell$ dimensional Boolean matrix. Table 1 shows an example.

	1	2	3	4	5
100	0	0	1	0	0
200	1	1	0	0	1
300	0	1	0	1	0
400	1	0	0	0	0

Table 1: Boolean matrix representation of market basket transaction data.

Association rule learning techniques are frequently applied for mining such transaction data. When the data is privacy sensitive we must restrict the access to the raw data for mining purposes. Randomized techniques have been proposed elsewhere [7] that work by randomly adding and deleting items from a transaction with a probability that depends on the number of items sold in a transaction. In the Boolean representation this is equivalent to flipping the bit values in the binary string representing a transaction. Although this bit-flipping probability varies one should be able to simplify the scenario in most real-life applications. In most market basket transaction data set the number of items sold stays within a relatively narrow regime. For example, even if a drug store has $10,000$ items in the inventory, most customers are likely to buy only a small fraction of them. Therefore, we may be able to divide the transaction data set among a set of subsets where each of them contains all the transactions that contain the same number of sold items per transaction. In each of these subsets, the bit-flipping probability remains constant across different transactions. So the item addition/deletion-based technique with varying bit-flipping probability can be reduced to a set of problems for each of the subsets with constant bit-flipping probability. Therefore, the fundamental privacy-preserving technique is reduced to random perturbation of the Boolean data representing the transactions. The work presented in this paper points out that such privacy of such Boolean data may not be adequately preserved using random perturbation of the bits.

2.3 Graph Structured Data

Data in the form of graph structures shows up in many link analysis applications. Telephone communication networks, intelligence sources usually generate this types of

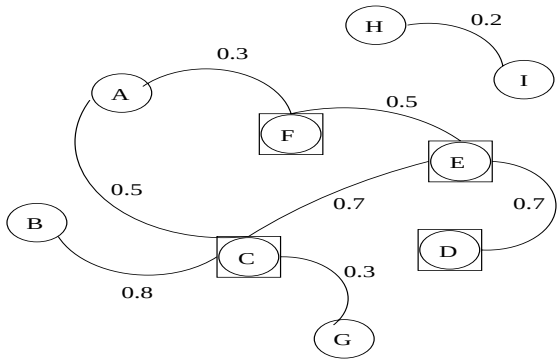

Figure 2: Graph data for link analysis.

data. These applications usually involve analysis of weighted directed or undirected graphs for detecting different characteristics like social groups, outliers behavior, and instance of target sub-graphs. Although the graphs themselves are not in tabular forms, they can be represented in that form. Adjacency matrix is one possible way to do that. For example, consider the graph shown in Figure 2. Let us assume that links C-E, F-E, and D-E are privacy sensitive. These links may correspond to properties that deal with sensitive features and therefore the exact link weights cannot be disclosed to the third party interested in mining the data.

	C	D	E	F
C	0	0	0.7	0
D	0	0	0.7	0
E	0.7	0	0	0.5
F	0	0	0.5	0

Table 2: The privacy sensitive links are represented using adjacency matrix-based representation.

One possible solution to this problem is to perturb the sensitive information in a secured fashion so that specific underlying data patterns remain invariant but the data itself appears very different from its original form. This problem can be posed in the following abstract form. Given the sensitive component of the graph shown in Table 2, find a representation of the data that preserves both privacy and the target types of data pattern.

When the graph is not weighted, the adjacency matrix is a Boolean matrix. Adding or deleting a link is therefore equivalent to flipping a bit value in the adjacency matrix. This paper considers this simpler version of the problem. It shows that the original graph structure can be accurately estimated even after perturbing the adjacency matrix by random noise (i.e. random arch addition and deletion). Although this randomized perturbation-based technique appear to be a natural extension of the other random data perturbation schemes, this may not offer sufficient protection from privacy breaches.

3. BREACHING THE PRIVACY: A RANDOM MATRIX-BASED FILTERING APPROACH

This section points out that although the data may look apparently different after the random additive perturbation, it is possible to extract the original data by using spectral filtering techniques. Detailed description of the material discussed in this section can be found elsewhere [14] of which this paper is an extension.

Consider an $m \times n$ data matrix U and a noise matrix V with same dimensions. The random value perturbation technique generates a modified (or perturbed) data matrix $U_p = U + V$. Our objective is to extract U from U_p. Although the noise matrix V may introduce seemingly significant difference between U and U_p, it may not be successful in hiding the data. Random noise has well defined probabilistic properties that may be used to identify the noise component of the perturbed data matrix U_p in an appropriate representation. The rest of this section argues that the spectral representation of the data allows us to do exactly that.

Consider the covariance matrix of U_p:

$$\begin{aligned} U_p^T U_p &= (U+V)^T (U+V) \\ &= U^T U + V^T U + U^T V + V^T V. \end{aligned} \quad (2)$$

Note that when the signal vector (columns of U) and random noise vector (columns of V) are uncorrelated, we have $E[U^T V] = E[V^T U] = 0$. This assumption is valid in practice since the noise V that is added to the data U is generated by a statistically independent process. If the number of observations is sufficiently large, we have that $U^T V \approx 0$. Equation 2 can now be simplified as follows:

$$U_p^T U_p = U^T U + V^T V \quad (3)$$

Since the correlation matrices $U^T U$, $U_p^T U_p$, and $V^T V$ are symmetric and positive semi-definite, let

$$U^T U = Q_u \Lambda_u Q_u^T, \; U_p^T U_p = Q_p \Lambda_p Q_p^T, \text{ and} \quad (4)$$

$$V^T V = Q_v \Lambda_v Q_v^T, \quad (5)$$

where Q_u, Q_p, Q_v are orthogonal matrices whose column vectors are eigenvectors of $U^T U$, $U_p^T U_p$, $V^T V$, respectively, and Λ_u, Λ_p, Λ_v are diagonal matrices with the corresponding eigenvalues on their diagonals.

It has been shown elsewhere [14] that for "reasonable" signal-to-noise ratio,

$$\Lambda_p \approx \Lambda_u + \Lambda_v. \quad (6)$$

Suppose the signal covariance matrix has only a few dominant eigenvalues, say $\lambda_{1,(u)} \geq \cdots \geq \lambda_{k,(u)}$, with $\lambda_{i,(u)} \leq \epsilon$ for some small value ϵ and $i = k+1, \ldots, n$. This condition is true for many real-world signals. Suppose $\lambda_{k,(u)} > \lambda_{1,(v)}$, the largest eigenvalue of the noise covariance matrix. It is then clear that we can separate the signal and noise eigenvalues Λ_u, Λ_v from the eigenvalues Λ_p of the observed data by a simple thresholding at $\lambda_{1,(v)}$. Note that equation 6 is only an approximation. However, in practice, one can design a filter based on this approximation to filter out [14] the perturbation from the data. This filtering approach first separates the signal eigenstates from those belonging to the noisy eigenstates and then use the signal eigenstates to construct an approximation of the original data by projecting the perturbed data on to the subspace spanned by the signal eigenvectors. In other words, $\hat{U} = U_p A_u {A_u}^T$, where A_u is the matrix whose columns are the eigenvectors corresponding to the signal eigenvalues.

It is obvious that the above theory can be extended to discrete data and this paper makes an attempt to present some results obtained by the authors in this regard.

4. RESULTS

This section presents several experimental results documenting the performance of the spectral filtering technique in reconstructing the continuous and discrete data.

4.1 Experiments with continuous data

We have performed experiments with artificial dataset having specific trend in its value as well as real world dataset containing random component. The results show that for dataset with specific trend like one shown in Figure 3, due to absence of any random component in actual data, Equation 6 holds closely, giving a close estimation of the actual data. Extensive experimental results, including comparison with other filtering techniques like moving average, Weiner filter, presented elsewhere [14] also support the observation.

The accuracy of the suggested method depends upon different factors. One is the relative amount of noise added to the actual data. The method works well as long as the relative noise content remain within a specific limit. In fact if that is not the case then the data mining algorithm will also have trouble extracting accurate patterns from the data. We define the term "Signal-to-Noise Ratio" (SNR) to quantify the relative amount of noise added to actual data to perturb it.

$$\text{SNR} = \frac{\text{Value of Actual Data}}{\text{Value of Noise Added to the Data}}$$

As the noise added to the actual value increases, the SNR decreases. Our experiments show that this method predicts the actual data reasonably well up to a SNR value of 1.0 (i.e. 100% noise). Figure 4 shows the difference in estimation accuracy as the SNR increases from 1. The dataset used has square trend in its values. The upper figure shows the estimation corresponding to 24% noise(mean SNR = 4.2), and the lower figure shows estimation corresponding to 90% noise (mean SNR = 1.1).

The second important factor is the inherent noise in the original dataset before we add noise explicitly for preserving privacy. The spectral filtering technique will remove the random noise regardless of its source. Therefore, if the data set contains some noisy eigenstates it will be removed since we do not have to identify whether this noise component originated from the original data set or from the privacy-preserving data transformation. As a result, sometimes the filtered data may look quite different from the original data set.

4.2 Experiments with adjacency matrices of graphs

This section presents experimental results for filtering out randomly perturbed graph structured data. First, note that the additive noise, in itself, is required to be an adjacency matrix, rather than any ordinary boolean matrix. This is required since, if this is not the case, (i.e additive noise is any

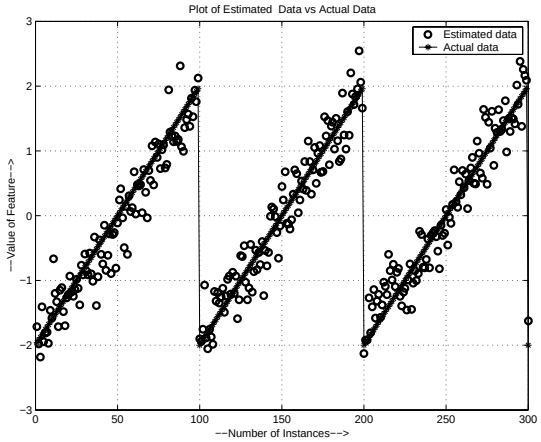

Figure 3: Estimation of triangular data using the spectral filtering technique.

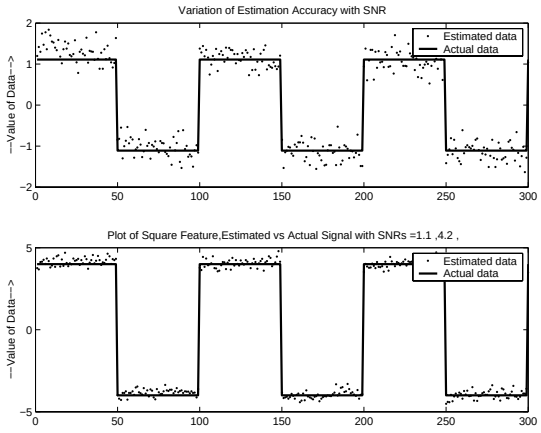

Figure 4: A higher noise content (low SNR) leads to less accurate estimation. SNR in upper figure is 1.1, while that for lower figure is 4.2.

boolean matrix without adjacency properties) then distortion in the data becomes obvious and the data miner comes to know at least to a small degree that the data has been tampered with. Consider the following example:

Let A be a 4×4 adjacency matrix of a graph as shown in Table 3. Let N be the 4×4 noise matrix that has been added to the adjacency matrix A. Note that in this example N (Table4) is any boolean matrix and does not necessarily have the properties of an adjacency matrix. The perturbed data(that is to be revealed to the data miner) is shown in the matrix P in Table 5. Obviously this is no longer an adjacency matrix and the miner immediately realizes that there has been some tampering with the data and thus may make attempts to obtain the original data (for example at the very least he can try and remove the ones from the diagonal of the perturbed matrix). But this should not be allowed in practise and hence for avoiding such a scenario in our experiments we assume that the noise matrix is also an adjacency matrix.

It must be noted, that in the case of market basket analysis, we may use the noise as an ordinary boolean matrix and the signal can be reconstructed in the same manner.

	1	2	3	4
1	0	0	1	0
2	0	0	1	0
3	1	1	0	0
4	0	0	0	0

Table 3: The privacy sensitive links in a boolean adjacency matrix-based representation.

	1	2	3	4
1	1	0	0	1
2	0	0	0	1
3	0	0	1	0
4	1	0	0	0

Table 4: A noise matrix which does not have properties of adjacency matrices.

Another important concern here is in addition of two boolean matrices. We use the OR function to perform the addition of the original adjacency matrix and the noise matrix. The rationale behind the same is as follows: If the noise matrix OR the original matrix has a 1 at a certain position, then the perturbed data also has a 1. However, the question arises when both the original and the noise matrix have 1's at the same position. Intuitively this means that there is a link in the original graph as well as in the noise graph. Hence in the perturbed graph, there should also exist a link.

There is also a difference in the way the **SNR** is estimated, considering that this is a boolean adjacency matrix.The SNR, for the special case of adjacency matrices of graphs is defined as follows:

$$\text{SNR} = \frac{\text{No of similar bits in the org data and pert data}}{\text{No of dissimilar bits in the org data and pert data}}$$

We perform experimentation on artificially generated data graphs. In these experiments, we assume the existence of deterministic rules that vertex i has links with, say n other vertices. For example, suppose that vertex 5 has links with 8 other vertices 6, 7, 8, and 9 and also has links with vertices 1, 2, 3, and 4 and so on. Such artificially generated rules can be supposed to mimic the real life scenario since vertices are linked to one another following certain rules. Figures 5, 6, and 7 show the original graph with 100 vertices, the perturbed graph and the reconstructed graph respectively. It is interesting to note that the error in reconstruction reduces (Figure 8) considerably as the number of vertices in the graph increases.

4.3 Experiments with Transaction Data

This section presents results for experiments with transaction data. In market basket data, it is common to observe that, some items are sold together while others are not. For example in supermarkets, bread and butter are usually sold together in one transaction.In our experiments, we synthetically generate boolean transaction matrices,containing spe-

	1	2	3	4
1	1	0	1	1
2	0	0	1	1
3	1	1	1	0
4	1	0	0	0

Table 5: The perturbed matrix released to the data miner.

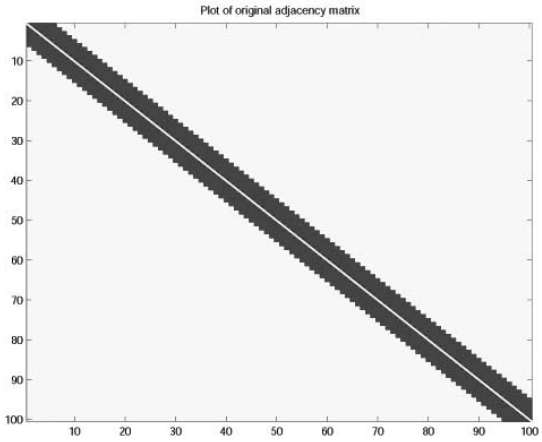

Figure 5: Original Adjacency Matrix of a graph with 100 vertices.

cific rules determining, which products are sold together.Thus each transaction matrix has an underlying distribution, which is of importance. A relatively small transaction matrix, generated with 10 products and 20 transactions has been used here for illustration. In Figure 9 the black portion indicates items sold together in a particular transaction and is based on artificially generated "trends" in transaction. Since transaction data can be privacy sensitive, the question is, can addition of random boolean noise to this transaction matrix, really distort the data or can it be easily reconstructed ? Addition of a boolean noise matrix produces a perturbed data set, given to the data miner as shown in Figure 10. The reconstructed matrix, which closes resembles the original transaction data is shown is Figure 11. In real life, transaction data often contains several noise components. When this happens, our methodology would not be able to filter out the already incorporated noise and hence the accuracy of reconstruction becomes subject to the amount of noise initially present in the data.

5. CONCLUSIONS

Preserving privacy in data mining activities is a very important issue in many applications. Randomization-based techniques are likely to play an important role in this domain. However, this paper illustrates some of the challenges that these techniques face in preserving the data privacy. It showed that under certain conditions it is relatively easy to breach the privacy protection offered by the random perturbation based techniques. It provided extensive experimental results with different types of data and showed that this is really a concern that we must address. This paper also presented results for discrete graph structured data represented

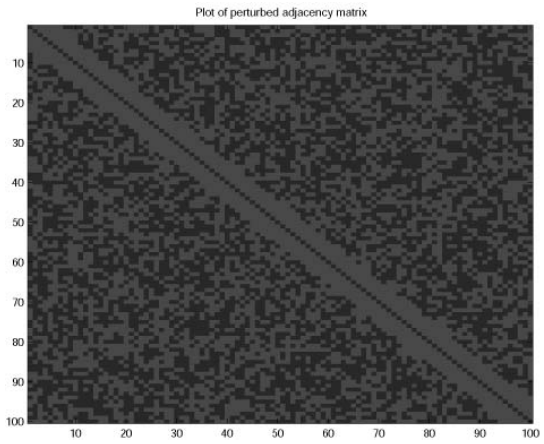

Figure 6: Perturbed Adjacency Matrix.

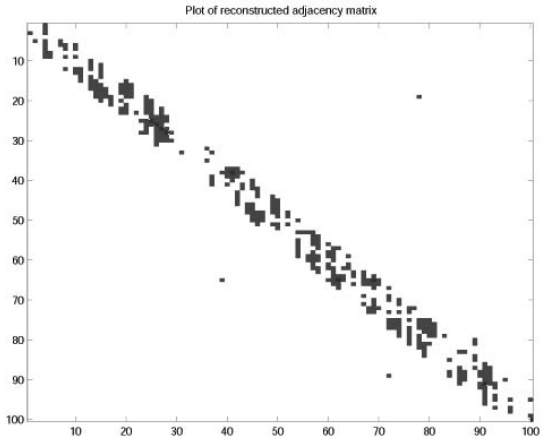

Figure 7: Reconstructed Adjacency Matrix.

using Boolean Adjacency Matrices and Boolean Transaction Matrices.

The paper offers a random-matrix based data filtering technique that may find wider application in developing a new perspective toward developing better privacy-preserving data mining algorithms. For example, we may be able to use this framework to develop algorithms that explicitly guard against potential compromise on privacy through linear transformations. The current privacy-preserving data mining algorithms do not pay adequate attention to this issue. Since the problem mainly originates from the usage of additive, independent "white" noise for privacy preservation, we should explore "colored" noise for this application.

6. ACKNOWLEDGEMENTS

The authors acknowledge supports from the NASA (NRA) NAS2-37143 and the United States National Science Foundation CAREER award IIS-0093353.

7. REFERENCES

[1] R. Agrawal and S. Ramakrishnan. Privacy-preserving data mining. In *Proceedings of SIGMOD Conference*, pages 439–450, 2000.

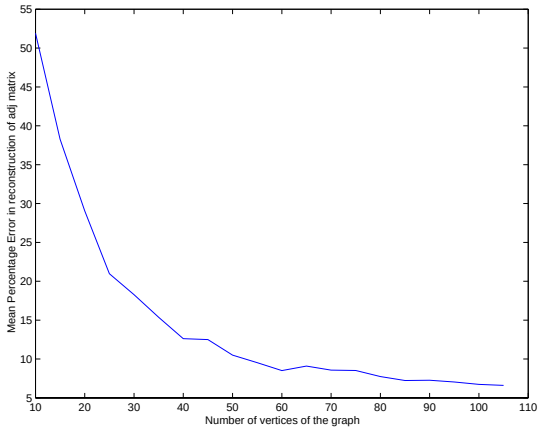

Figure 8: Plot of error in reconstruction versus the number of vertices in the graph.

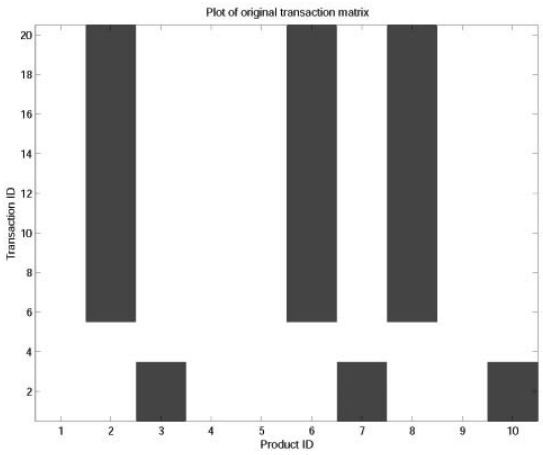

Figure 9: Original Transaction Matrix

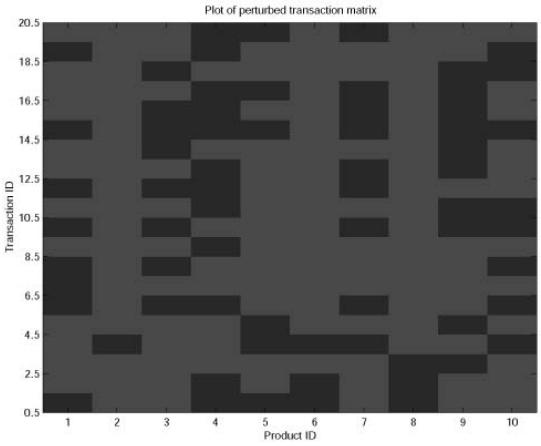

Figure 10: Perturbed Transaction Matrix.

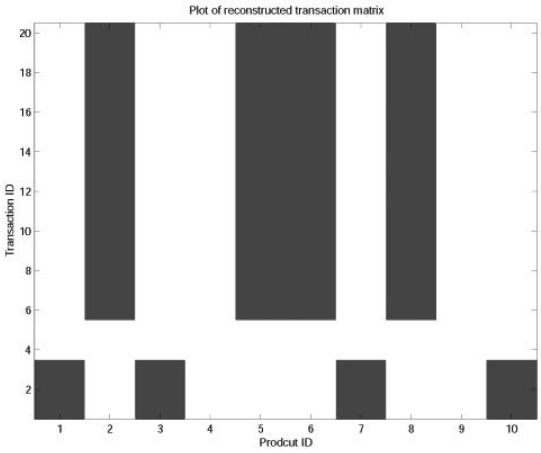

Figure 11: Reconstructed Transaction Matrix

[2] D. Agrawal and C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In *Proceedings of Symposium on Principles of Database Systems*, pages 247–255, 2001.

[3] R. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proc. of the 40th Foundations of Computer Science*, New York, New York, 1999.

[4] D. J. Cook and L. B. Holder. Graph Based Data mining. IEEE Intelligent Systems, 15(2), pages 32-41, 2000.

[5] W. Du and Z. Zhan. Building decision tree classifier on private data. In *Workshop on Privacy, Security, and Data Mining at The 2002 IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December 9 2002.

[6] J.S. Deogun and V.V. Raghavan and H. Sever Association Mining and Formal Concept Analysis,1998.

[7] A. Evfimievski and R. Srikant and R. Agrawal and J. Gehrke. Privacy Preserving Mining of Association Rules. Proc. of 8th ACM SIGKDD First International Conference on Know ledge Discovery and Data Mining, 2002.

[8] A. Evfimievski. Randomization in Privacy-Preserving Data Mining. ACM SIGKDD Explorations, Volume 4, issue 2, pages 43–48, 2003.

[9] R. Falk and A. Well. Many faces of the correlation coefficient. *Journal of Statistics Education*, 5(3), 1997.

[10] H. Goldberg and T. Senator. Restructuring databases for knowledge discovery by consolidation and link formation. First International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1995.

[11] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence: Imitating Life*, pages 43–56, 1994.

[12] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data, 2002.

[13] H. Kargupta, B. Park, D. Hershberger, and E. Johnson. Collective data mining: A new perspective towards distributed data mining. In *Advances in Distributed and Parallel Knowledge Discovery, Eds: Kargupta, Hillol and Chan, Philip.* AAAI/MIT Press, 2000.

[14] H. Kargupta, S. Datta, and K. Sivakumar. (2003). On the Privacy Preserving Properties of Random Data Perturbation Techniques. Accepted for publication in the Proceedings of the IEEE International Conference on Data Mining. Melbourne, USA.

[15] H. Kargupta, K. Liu, and J. Ryan. (2003). Random projection and privacy preserving correlation computation from distributed data. Technical Report TR-CS-03-24, Computer Science and Electrical Engineering Department, University of Maryland, Baltimore County.

[16] L. C. Parra. An Introduction to Independent Component Analysis and Blind Source Separation. Sarnoff Corporation, CN-5300, Princeton, NJ 08543, 1999.

[17] M. K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks*, volume 13, pp. 251-274.

[18] R. Srikant, R. Agrawal Mining Quantitative Association Rules in Large Relational Tables. In *ACM SIGMOD International Conference on Management of Data,1996* .

[19] J. S. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *In The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.

[20] S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.

[21] K. L. Weldon. A simplified introduction to correlation and regression. *Journal of Statistics Education*, 8(3), 2000.

[22] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Symposium on Foundations of Computer Science (FOCS)*, pages 160–164. IEEE Computer Society Press, 1982.

[23] Y. Zue and D. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[24] UCI Machine Learning Repository. http://www.ics.uci.edu/ mlearn/MLSummary.html.