International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 38296-38301

Perturbing Sensitive Data using Additive Noise

A.V.Sriharsha
PhD Research Scholar,
Department of Information Technology,
SCSVMV University,
Kancheepuram, INDIA
avsreeharsha@gmail.com

Dr. C. Parthasarathy,
Assistant Professor,
Department of Information Technology,
SCSVMV University,
Kancheepuram, INDIA
drsarathy45@gmail.com

Abstract- Privacy Preserving Data Mining (PPDM) addresses the problem of developing accurate models about aggregated data without access to precise information in individual data record. A widely studied perturbation-based PPDM approach introduces random perturbation to individual values to preserve privacy before data are published. Previous solutions of this approach are limited in their tacit assumption of trust on data miners without sensitivity issue. In this work, we overcome the problem of distrust and set, the more trusted a data miner is the less perturbed copy of the data it can access. Under this setting, a malicious data miner may have access to differently perturbed copies of the same data through various means, and may combine these diverse copies to jointly infer additional information about the original data that the data owner does not intend to release. Preventing such diversity attacks is the key challenge of our work. Perturbing costs on data sets are further optimized based on the sensitivity of the attributes required for perturbation. Our solution allows a data owner to generate perturbed copies of its data for arbitrary sensitivity levels on demand. This feature offers data owners' a maximum flexibility.

Keywords: Data Mining, Privacy Preserving Data Mining, Perturbation, Additive Noise.

Introduction

The convergence of pervasive forms of data collection, widespread deployment of cheap digital sensors, and economics of infinite storage is apparently leading us into an age of perfect remembering where "*everyone is on the record all the time*." This convergence of pervasive forms of data collection, widespread digitization of analog records, economics of infinite storage, and subsumption of all media into the digital format is, according to the majority of commentators, inexorably leading us into an age of "*perfect remembering*," enabling individuals to "*google their past*," recalling at will individual events in full multimedia richness, identifying trends in personal health, work activities, and lifestyle. Indeed, if one factors in the gradual elimination of paper in favor of digital forms for commercial transactions, communication, documentation, etc., and the continually plummeting costs of digital storage, the picture of a world where "*everyone is on the record all the time*" does not seem far-fetched. With regard to both health and learning, similar patterns will obtain. Instead of relying on patients' vague account of their ailments, doctors will finally have access to "*minutely detailed chronicles of vital signs, behavior, diet, and exercise, along with physician' diagnoses, prescriptions, advice, and test results*." While each of these approaches does contribute something to restoring a certain balance, each has also significant drawbacks. Practitioners of digital abstinence must systematically forego the various benefits service providers offer in exchange of release of personal information; privacy rights have historically enjoyed limited successes in the US, and policies for automatic negotiation of privacy settings between information sharing devices are notoriously complicated for dedicated experts, let alone for casual users.

What is PPDM?

Privacy in data for data mining is ensured by many methods since decades. Data Transformation in the KDD process ensures transforming the data into cryptic codes and some abbreviated forms, yet the details of the data are guessable to the data miners. Protecting privacy for the data which is believed to confidential such as individual data, has been a great challenge for the data miners during the KDD process.

The main consideration in privacy preserving data mining is the sensitive nature of raw data. The data miner, while mining for aggregate statistical information about the data, should not be able to access data in its original form with all the sensitive information. Simple techniques like deleting the unique personal identifier like name or social security number from a dataset containing personal information does not help always. It is not safe enough since re-identification attacks have emerged which can link different public data sets to re-identify the original subjects.

Perturbation

One approach to privacy-preserving data mining is based on perturbing the original data, then providing the perturbed dataset as input to the data mining algorithm. The privacy-preserving properties are a result of the perturbation: Data values for individual entities are distorted, and thus individually identifiable (private) values are not revealed. This work shows that random additive perturbation fails to preserve privacy in the data mining. As in the erstwhile research, random matrices have 'predictable' structures in the spectral domain and it develops a random matrix-based '*Spectral Filtering Technique*' (SPF) to retrieve original data from the dataset distorted by adding random values. The proposed method works by comparing the spectrum generated from the observed data with that of random matrices. This work presents the theoretical foundation and extensive experimental results to demonstrate that in many cases random data distortion preserves very little data privacy. It presents some direct comparison with previously suggested privacy preserving data mining techniques based

on additive random perturbation as well to show the serious breach of privacy. It also explores the possibility of proposed spectral filtering technique on different data types and perturbation methods e.g. discrete data and exclusive or noise. The analytical framework presented in this work points out several possible avenues for the development of new privacy-preserving data mining techniques.

Background

Data anonymization is the process of conditioning a dataset such that no sensitive information can be learned about any specific individual, yet valid scientific conclusions can nevertheless be drawn. Deidentification, or removing explicit identifiers like names and phone numbers, is necessary but insufficient to protect individual privacy. We must also remove enough additional information so that an attacker cannot infer an identity based on what remains (a reidentification disclosure) or otherwise infer sensitive information about an individual (a prediction disclosure). These kinds of disclosures could be made by examining the data for combinations of variables that might uniquely identify someone, or for patterns of values that unintentionally reveal sensitive information. This is exactly what happened when a reporter reidentified an Internet user in released, deidentified search queries - the combination of several queries was enough to narrow the searcher's identity to one particular person [2].

The only known way to prevent these disclosures is to remove additional information from the dataset. Most existing methods work by perturbing or suppressing variable values, causing uncertainty in identity inference or sensitive–value estimation. This has been an area of active research for three decades, yet nearly every aspect of it remains an open question: How do we measure privacy protection, and what amount of protection do we want? What is the optimal method of perturbing the data to achieve this protection? How do we measure the impact of the perturbation on scientific analysis, and what is an acceptable impact?

Related Work

Disclosure limitation in a public data release involves some degree of modification of the data to be released. Instead of publishing the original data ***D***, a masked version ***D′*** is published. The masking improves privacy but reduces the utility of the published data, in comparison to the original data. This tension between privacy and utility is unavoidable: privacy and utility are two different views of the same thing, the amount of information published. By reducing the amount of information published, privacy improves but utility decreases; and the other way round. Two extreme cases are: publish the original data, which offers the greatest utility but the least privacy; and publish encrypted or random data, which incurs no disclosure risk at all, but offers no utility.

Disclosure limitation technologies seek equilibrium between privacy and utility: the disclosure risk must be limited, but the data need to remain useful. Sometimes the required equilibrium between privacy and utility does not exist; for instance, when access to very accurate and sensitive data is required by some data recipient. As the publication of such a data set is not feasible, data providers must rely on other mechanisms such as data access restriction and non-disclosure agreements.

Most disclosure limitation mechanisms are specifically designed to avoid releasing information that is known to be disclosive. Such mechanisms are instructed with the kind of data releases that may lead to a privacy breach, and are designed to avoid them. To determine the data releases that may lead to a privacy breach, a guess on the amount of side information available to the intruders is usually made. As long as this guess is accurate, the disclosure limitation mechanism accomplishes its duty, but a privacy breach may happen if there are intruders with greater amounts information.

Perlin Noise

A good random number generator produces numbers that have no relationship and show no discernible pattern. As we are beginning to see, a little bit of randomness can be a good thing when programming organic, lifelike behaviors. However, randomness as the single guiding principle is not necessarily natural. An algorithm known as "*Perlin noise*," named for its inventor Ken Perlin, takes this concept into account. Perlin developed the noise function while working on the original Tron movie in the early 1980s; it was designed to create procedural textures for computer-generated effects. In 1997 Perlin won an Academy Award in technical achievement for this work. Perlin noise can be used to generate various effects with natural qualities, such as clouds, landscapes, and patterned textures like marble. The applications of Perlin Noise are most considered in the field of graphics, where a smooth noise is added to an image, video or audio in order to conceal certain facts. The similar attempt is made on the databases, where the result of a query could not reveal the original status of the data. Perlin noise has a more organic appearance because it produces a naturally ordered ("*smooth*") sequence of pseudo-random numbers. The Perlin Noise depends on Octaves, Colors and Textures, which are the properties detected from the data, when Perlin Noise is applied on the graphics. When comes to databases, they are the Sensitivity Levels of the Attribute, Database schemas that attributes are present, Semantic Inference of the attributes for the queries posed on the databases.

Symbols	Components to Generate Perlin Noise for Graphics	Equivalent Assumptions of Components in a database for Noise generation.
α	Octaves	Sensitivity Levels of the Attribute;
β	Colors	Database schemas that attributes are present;
δ	Textures	Semantic Inference of the attributes in the query results returned from the databases;

Table 1: Symbols used for describing the dimensions in Perlin Noise.

Therefore a Three-dimensional PN (3dPN) is an ideal for will be generated for the data set. For One-dimensional

application of Perlin Noise, the noise is only the random number of the integer.

$$x = \text{random}(0, \textbf{width})$$

In the above assumption, **width** is the difference between the lower limit and the upper limit of the data values of a given attribute and zero is the starting flag of the random number generation. The random values are dependent on time t. Time gap persists in generating the random numbers series, from one random value to the next. As one of the computing overhead of a microprocessor is time that varies for each instruction to generate the random numbers in a series, they appear to be incremental for short time gaps, a considerably large time gap can generate even a lower random number than in the progressive series of random numbers.

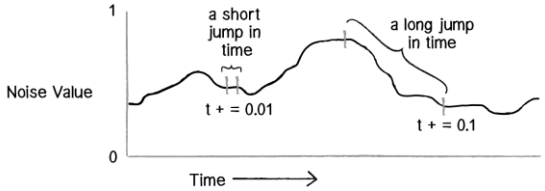

Fig 1: Illustrating Noise value with respect to time variance.

However the time gap cannot be kept algorithmically as a constant. The group of time gaps is always chosen.

Influence of Octaves or/are the Sensitivity Levels of the Attribute: Octaves are measured as low and high in the CG application of PN. In the database application, the attribute's sensitivity level is assumed. Less the sensitivity, coarse the noise generated, more the sensitivity smooth noise generated. If the degree of sensitivity is high, obviously the attribute is very sensitive that it can project more information or the just the presence of such attribute in the query result gives much crucial information to the adversary (query intruder/poser/attacker), where measures to conceal the ability of projecting facts need to be taken care. If the degree of sensitivity is low, the presence of attribute in the query result is very auxiliary, even measures to conceal the ability of projecting facts is taken care but not more than that of an attribute with higher degree of sensitivity.

Influence of Colors or/are the Database schemas that attributes are present: Colors are notional in the Perlin Noise. Variations of the data with respect to colors are a basic representation of the differences in the data. Attributes during the design of schemas in the databases, have similar applications though they differ in their nomenclature. Such attributes having similar range, domain and entropy need to be identified as the data with different colors. A sum of a kind of similar attributes with respect to their applications is measured as colors of attributes.

Influence of Textures or/are the Semantic Inference of the attributes in the query results returned from the databases: Some of the attributes that pose prominence in the query result, same set of attributes form as a supporting group to form a composition of information.

Kronecker product

In this problem, the covariance matrix of Relaxed Perlin Noise can be written as the Kronecker product of two matrices. Where the data set is have a $1 \times n$ and noise as $p \times q$. We explore the properties of the Kronecker product for efficient computation.

The Kronecker product is a binary matrix operator that maps two matrices of arbitrary dimensions into a larger matrix with a special block structure. The example is shown below.

Let us take in this example, If A is an $\boldsymbol{m} \times \boldsymbol{n}$ matrix and B is a $\boldsymbol{p} \times \boldsymbol{q}$ matrix, then the Kronecker product A ⊗ B is the $\boldsymbol{mp} \times \boldsymbol{nq}$ block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \dots & \dots & \dots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \rightarrow [1]$$

Problem Formulation

In this section, we present the problem setting and describe our threat model, state our privacy goal and identify the design space.

Following are the symbols and names used in the description of problem formulation:

Notation	Definition
$\boldsymbol{X}$	Original data set
$\boldsymbol{N}$	Number of tuples addressed in data set
$\boldsymbol{Y}$	Perturbed copies of data set
$\boldsymbol{Z}$	Noise generated using Relaxed Perlin Noise
$\boldsymbol{H}$	Noise Inducer
$\boldsymbol{I}$	Set of Noise values (in a matrix)
$\boldsymbol{R}$	Relation in the data set
$\boldsymbol{s}$	Perturbation magnitude
$\boldsymbol{S}$	A set of sensitivity levels
$\boldsymbol{k}$	Clearance constant
$\boldsymbol{j}$	A particular sensitivity level

Problem Settings

In PPDM problem, we consider in this paper, a data owner trusts data miners at different levels and generates a series of perturbed copies of its data for different trust levels. This is done by adding varying amount of noise to the data. Under this setting, data miners at higher trust levels can access less perturbed copies. Such less perturbed copies are not accessible by data miners at lower trust levels. In certain scenarios, data miners at higher trust levels may also have access to the perturbed copies at more than one trust levels. Data miners at different trust levels may also collude to share the perturbed copies among them. As such, it is common that data miners can have access to more than one perturbed copies.

Specifically, we assume that the data owner wants to release M perturbed copies of its data $\boldsymbol{X}$, which is an $N \times 1$ vector with calculated mean and covariance. These M copies can be generated in various fashions. They can be jointly generated all at once. Alternatively, they can be generated at different times upon receiving new requests from data miners, in an on-demand fashion. The latter case gives data owners' a maximum flexibility.

It is true that the data owner may consider releasing only the mean and covariance of the original data. We remark that simply releasing the mean and covariance does not provide the same utility as the perturbed data. For many real applications, knowing only the mean and covariance may not be sufficient to apply data mining techniques, such as clustering, principal component analysis, and classification. By using random perturbation to release the data set, the data owner allows the data miner to exploit more statistical information without releasing the exact values of sensitive attributes.

Based on the above scenario the about the sensitive attributes, where there is also a need for dispersal of sensitive attributes, the existing methods pose a problem face to present such attributes to the believed to be data miners also. In this problem the sensitivity of the attributes also are assessed and the guided noise using the Relaxed Perlin Noise generation method is applied, to enable the comfort of the data miners. This method can also be applied to the other attributes where the privacy can be ensured with relative data. This method prevents the guessability attacks and other adversarial attacks even in the distributed scenario.

Basic Setting

Let $\boldsymbol{Y} = [Y_1 \ldots Y_M]$ be the set of all perturbed copies Y_i where $1 \leq \boldsymbol{i} \leq M$ and M is maximum number of perturbed copies. Let $\boldsymbol{Z} = [Z_1 \ldots Z_M]$ be the set of noise generated using Relaxed Perlin Noise generator. Let $\boldsymbol{H}$ be an inducer with $(N \cdot M) \times N$ and the matrix follows:

$$H = \begin{bmatrix} I_N \\ \ldots \\ I_N \end{bmatrix} \qquad \rightarrow [2]$$

where I_N represents an $N \times N$ identity matrix. Now, we have the relationship between $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{Z}$ as $\boldsymbol{Y} = \boldsymbol{HX} + \boldsymbol{Z}$.

To be more robust against advanced filtering attacks, individual noise terms in $\boldsymbol{Z_i}$ added to different attributes in $\boldsymbol{X}$ should have the same correlations as the attributes themselves; otherwise $\boldsymbol{Z_i}$ can be easily filtered out. That explains the covariance of the individual noise terms and the covariance of the data set should be in correlation with respect to a perturbation magnitude ($\boldsymbol{s}$). The perturbation magnitude is chosen by the data owner a value, according to the trust level associated with the target perturbed copy $\boldsymbol{Y_i}$.

$$\mathrm{COV}(\boldsymbol{Z_i}) = \boldsymbol{s}.\mathrm{COV}(\boldsymbol{X}) \text{ and } \mathrm{COV}(\boldsymbol{Y_i}) = (1 + \boldsymbol{s}).\mathrm{COV}(\boldsymbol{X}) \qquad \rightarrow [3]$$

Setting with Sensitivity of Attribute

This is further expanded for the above setting, with explicitly sensitivity identification and generating the Relaxed Perlin Noise for sensitive attributes.

Let $\boldsymbol{X}$ be the data set that contains various attributes which among them some are high in sensitivity and some are low in sensitivity. The level or degree of sensitivity is determined by Information Gain methods where the attribute is said to have a set of values which can determine the primary class. Typically in data mining functionalities, classification of datasets in data mining is a group of attributes contribute to define a class, using which a classification tree or decision tree is generated. Usually ID3 algorithm and its variants are used to determine the tree. Information Gain calculation methods are widely used to determine the attribute participation into a class and connect as node in some level of the tree. The top level of the tree has attributes with high degree of information gain, the degree of information gain reduces to the sub levels. However, the fundamental Information Gain method is used to group the attributes into a class. Similar method is used to determine the Information Gain ratio for each attribute of the data set and determined to be sensitive if the ratio satisfies the assumed threshold. The data set $\boldsymbol{X}$ is said to have attributes of $\boldsymbol{A}_1^{\,1} \ldots \boldsymbol{A}_{\boldsymbol{n}}^{\,S}$. Where there are $\boldsymbol{n}$ attributes and $\boldsymbol{S}$ degrees of sensitivity levels.

DEFINITION 1:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.

DEFINITION 2:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.
Where $\boldsymbol{n}$ is not equal to $\boldsymbol{S}$.

DEFINITION 3:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.
Where all $\boldsymbol{n}$ attributes does not have all sensitivity levels $\boldsymbol{S}$.

DEFINITION 4:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.
Where a group of attributes among $\boldsymbol{n}$ have a particular discrete sensitivity levels in $\boldsymbol{S}$.

DEFINITION 5:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.
Where a group of or all the attributes among $\boldsymbol{n}$ does not have any particular discrete sensitivity levels in $\boldsymbol{S}$.

DEFINITION 6:
A relation $\boldsymbol{R}$ in the data set $\boldsymbol{X}$ with $\boldsymbol{n}$ number of attributes with $\boldsymbol{S}$ levels of degrees of sensitivity.
Where a group of or a few attributes among $\boldsymbol{n}$ have any particular sensitivity level only in $\boldsymbol{S}$.

As stated from the above definitions, various assumptions can be made of the data sets which containing various characteristics. In this formulation let us assume the ideal version of the data set which contains a sufficient number of attributes ($\boldsymbol{n}$) with selected sensitivity levels in $\boldsymbol{S}$. The sensitivity levels are described in a spectrum consisting from high to low. The maximum of sensitivity levels in the

group of attributes or the individual selection of attributes is considered for the determining the perturbation magnitude.

The perturbation magnitude (***s***) is even judged according to the sensitivity level of a group or individual attributes (***A***). In a group of attributes, for each attribute information gain ratio is calculated and where the maximum of information gain ratio among the attributes in the group is selected as the perturbation magnitude. This perturbation magnitude is applied to all the attributes commonly and the distortion is applied with randomization.

$$\boldsymbol{s} \equiv \text{k.max}(\boldsymbol{S}) \rightarrow [4]$$

The value of ***s*** is identically equal to the maximum sensitivity of the attributes with a clearance constant. A clearance constant is used to rectify approximation error in the sensitivity level in order to calculate the integer perturbation magnitude.

The data miner who requests for the data set from the data owner does not always have the need of all attributes. A set of attributes significantly or randomly selected would be chosen and they need to publish. A particular perturbed copy for the convenience of the data miner is not possible to develop from the original data set, several perturbed copies are developed and distributed among the group of data miners and they are shared according to their requirements.

For a data set ***X*** with ***A*** attributes say there are ***n*** attributes, ***n*** × ***n*** number of gain groups are generated and their Information Gain ratio is calculated. From the data miner end the attribute information required to publish is taken, however all the attributes requested by the data miner for publishing are not perturbed and not issued to public use. The attributes for publishing are available in the gain groups but in many combinations of other attributes. A particular gain group is selected with the less information gain ratio where it contains maximum of the attributes to publish. Based on the Information Gain ratio of that group as the seed value for the Relaxed Perlin Noise generates noise for the group of attributes and their values. The perturbation of the data with the noise is implemented by Kronecker product of the data set and the Relaxed Perlin Noise values.

Let $\boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}(\boldsymbol{X})$ Gain groups of the data set, Let $\boldsymbol{A}_{\mathbf{P}}$ is the set of attributes required be published to the data miner and they are members of all the groups of attributes $\boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}(\boldsymbol{X})$ in the data set,

$$\boldsymbol{A}_{\mathbf{P}} \in \boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}(\boldsymbol{X}) \text{ and}$$

$$\boldsymbol{A}_{\mathbf{P}} \subset \boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}(\boldsymbol{X})$$

Let Information Gain ratios of all the attribute groups of the data set ***X*** be $\boldsymbol{G}(\boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]})$.

$\boldsymbol{s} = \max(\boldsymbol{G}(\boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}))$ is the perturbation magnitude.

Let Seed value for the Relaxed Perlin Noise generator is

$$\boldsymbol{G}(\boldsymbol{A}_{\mathbf{P}}) \leq \boldsymbol{G}(\boldsymbol{A}_{[(\mathbf{1}\ldots\boldsymbol{n})\times(\mathbf{1}\ldots\boldsymbol{n})]}).$$

$\boldsymbol{G}_{\mathbf{ratio}}$ $(\boldsymbol{G}(\boldsymbol{A}_{\mathbf{P}}))$ is the seed value for the Relaxed Perlin Noise generator.

Let ***P*** be the Relaxed Perlin Noise that is generated as a vector for all the attributes and their values of the selected data set of an attribute group to publish to the data miner.

$$\text{So, } P = \begin{matrix} P_{11} & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & P_{mn} \end{matrix}$$

is the matrix of Relaxed Perlin Noise values each column generated for the attribute in the group and the rows for the values of the attributes and hence, the perturbed copies ***Y*** is determined as

$$\boldsymbol{Y} = \boldsymbol{X}(\boldsymbol{A}_{\mathbf{P}}) \otimes \boldsymbol{P} \rightarrow [5]$$

Let $\boldsymbol{Y} = [Y_1 \ldots Y_M]$ be the set of all perturbed copies Y_i where $1 \leq \boldsymbol{i} \leq M$ and M is maximum number of perturbed copies. Let $\boldsymbol{P} = [P_1 \ldots P_M]$ be the set of noise generated using Relaxed Perlin Noise generator with the sensitivity consciousness. Let ***H*** be an inducer with $(N \cdot M) \times N$ and the matrix follows:

$$H = \begin{bmatrix} I_N \\ \ldots \\ I_N \end{bmatrix} \rightarrow [6]$$

where I_N represents an $N \times N$ identity matrix. Now, we have the relationship between ***Y***, ***X***, and ***P*** as $\boldsymbol{Y} = \boldsymbol{HX} + \boldsymbol{P}$ or the $\boldsymbol{Y}_{\mathbf{P}}$.

To be more robust against advanced filtering attacks, individual noise terms in $\boldsymbol{P}_i$ added to different attributes in ***X*** should have the same correlations as the attributes themselves, otherwise $\boldsymbol{P}_i$ can be easily filtered out. That explains, the covariance of the individual noise terms and the covariance of the data set should be in correlation with respect to a perturbation magnitude (***s***) . The perturbation magnitude is chosen by the data owner a value, according to the trust level associated with the target perturbed copy $\boldsymbol{Y}_i$.

$$\text{COV}(\boldsymbol{Z}_i) = \boldsymbol{s}.\text{COV}(\boldsymbol{X}) \text{ and } \text{COV}(\boldsymbol{Y}_i) = (1 + \boldsymbol{s}).\text{COV}(\boldsymbol{X}) \rightarrow [7]$$

Threat Model

We always think malicious data miners who always attempt to attack on publishing, reconstruct a more accurate estimate of the original data given in perturbed copies. The reconstruction accuracy depends heavily on the adversaries' knowledge about the domain. Reinforcing the assumption that adversaries have the knowledge of the statistics of the original data ***X*** and the noise ***Z*** , i.e., mean of ***X***, and covariance matrices of ***X*** and ***Z*** and that the adversaries with less knowledge are weaker. We assume adversaries only perform linear estimation attacks, where estimates can only be linear functions of the perturbed data ***Y***. It is known that if

$\boldsymbol{X}$ follows a jointly Gaussian distribution, then Linear Least Square estimation achieves the minimum estimation error among both linear and nonlinear estimation methods. For $\boldsymbol{X}$ with general distribution, Linear Least Square estimation has the minimum estimation error among all linear estimation methods. In our setting the perturbed copies with Relaxed Perlin Noise generation $\boldsymbol{Y_P}$. From the derivation of $\boldsymbol{Y} = \boldsymbol{HX} + \boldsymbol{P}$ has very least estimation that an adversary can possibly make.

Design Space and Privacy goal

In this setting, a data owner releases distinctly perturbed copies of its data to multiple data miners. One key goal of the data owner is to control the amount of information about its data that adversaries may derive.

We assume that the data owner wants to distribute a total of $\boldsymbol{M}$ different perturbed copies of its data, i.e., $\boldsymbol{Y_i}$ $(1 \leq \boldsymbol{i} \leq \boldsymbol{M})$, each for a trust level $\boldsymbol{i}$. The assumption of $\boldsymbol{M}$ is for ease of analysis. It will become clear later that our solution of the on-demand generation allows a data owner to generate as many different copies as it wishes. The data owner can easily control the amount of the information about its data an attacker may infer from a single perturbed copy. The data owner can easily control the privacy of an individual copy $\boldsymbol{Y_i}$ by setting **s** (perturbation magnitude) according to trust level $\boldsymbol{i}$ through a one-to-one mapping.

Conclusions

Many interesting and important directions are worth exploring. For example, it is not clear how to expand the scope of other approaches in the area of partial information hiding, such as random rotation-based data perturbation, *k*-anonymity, and retention replacement, to multiple levels of sensitivity. It is also of great interest to extend our approach to handle evolving data streams.

Last but not the least, our solution allows data owners to generate perturbed copies of its data at arbitrary sensitivity levels on-demand. This property offers the data owner maximum flexibility.

The key challenge lies in preventing the data miners from combining copies at different trust levels to jointly reconstruct the original data more accurate than what is allowed by the data owner. We address this challenge by properly correlating noise across copies at different sensitivity levels. We prove that if we design the noise covariance matrix to have corner-wave property, then data miners will have no diversity gain in their joint reconstruction of the original data.

References

[1] Richard H. Rand, Dieter Armbruster, "*Perturbation Methods, Bifurcation Theory and Computer Algebra*", © 1987 by Springer-Verlag New York Inc.

[2] Jaideep Vaidya, Chris Clifton, Michael Zhu, "*Privacy Preserving Data Mining*", © 2006 Springer Science+Business Media, Inc.

[3] Charu C. Aggarwal, Philip S. Yu, "Privacy-Preserving Data Mining - Models and Algorithms", © 2008 Springer Science+Business Media, LLC.

[4] Christos Dimitrakakis, Aris Gkoulalas-Divanis, Aikaterini Mitrokotsa Vassilios S. Verykios, Yücel Saygin (Eds.), "*Privacy and Security Issues in Data Mining and Machine Learning*", © Springer-Verlag Berlin Heidelberg, September 2010.

[5] Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu, "*Introduction to Privacy-Preserving Data Publishing*", Chapman & Hall/CRC, © 2011 by Taylor and Francis Group, LLC.

[6] Aritra Dasgupta and Robert Kosara, "*Adaptive Privacy-Preserving Visualization Using Parallel Coordinates*", IEEE Transactions On Visualization And Computer Graphics, Vol. 17, No. 12, Pp. 2241, © December 2011.

[7] Pui K. Fong and Jens H. Weber-Jahnke, "*Privacy Preserving Decision Tree Learning Using Unrealized Data Sets*", IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 2, Pp. 353 © February 2012.

[8] Yaping Li, Minghua Chen, Qiwei Li, and Wei Zhang, "*Enabling Multilevel Trust in Privacy Preserving Data Mining*", IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 9, Pp. 1598, © September 2012.

[9] Daniel Shiffman, "The Nature of Code", http://natureofcode.com/ © 2012.

[10] Fosca Giannotti, Laks V. S. Lakshmanan, Anna Monreale, Dino Pedreschi, and Hui (Wendy) Wang, "*Privacy-Preserving Mining of Association Rules From Outsourced Transaction Databases*", IEEE Systems Journal, Vol. 7, No. 3, Pp. 385 © September 2013.

[11] Noman Mohammed, Dima Alhadidi, Benjamin C. M. Fung, and Mourad Debbabi, "*Secure Two-Party Differentially Private Data Release for Vertically-Partitioned Data*", IEEE Transactions On Depedable And Secure Computing, Vol. 11, No. 1, Pp. 59 © January/February 2014.

[12] Perlin Noise, Wikipedia.