Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining -2

Prof. Ja-Ling Wu
Dept. CSIE & GINM
National Taiwan University
1 RANDOM PROJECTION-BASED MULTIPLICATIVE PERTURBATION

1.1 Basic Mechanism

Random projection refers to the technique of projecting a set of data points from a high-dimensional space to a randomly chosen lower-dimensional subspace. The key idea of random projection arises from the Johnson-Lindenstrauss Lemma [2] as follows:
Lemma 5.1 (JOHNSON-LINDENSTRAUSS LEMMA). For any $0 < \epsilon < 1$ and any integer $s$, let $k$ be a positive integer such that $k \geq \frac{4\ln s}{\epsilon^2/2 - \epsilon^3/3}$. Then, for any set $S$ of $s = |S|$ data points in $\mathbb{R}^m$, there is a map $f : \mathbb{R}^m \rightarrow \mathbb{R}^k$ such that, for all $x, y \in S$,

$$(1 - \epsilon)\|x - y\|^2 \leq \|f(x) - f(y)\|^2 \leq (1 + \epsilon)\|x - y\|^2,$$

where $\|\cdot\|$ denotes the vector 2-norm.
This lemma shows that any set of $s$ points in $m$-dimensional Euclidean space can be embedded into an $O\left(\frac{\log s}{\epsilon^2}\right)$-dimensional space such that the pair-wise distance of any two points are maintained within an arbitrarily small factor.
This beautiful property implies that it is possible to change the data’s original form by reducing its dimensionality but still maintains its statistical characteristics.

In this lecture, we shall demonstrate how random matrices can be used for this kind of map. To give the reader a general idea of how the random projection technique perturbs the data, we did both row-wise and column-wise projections of the sample data given in Fig. 2a.
The results are shown in Figs. 6a and 6b.

It can be seen that the original structure of the data has been dramatically obscured.

In the following part of this lecture, we discuss some interesting properties of the random matrix and random projection, which are good for maintaining the data utility.
Fig. 6. (a) The perturbed data after a row-wise random projection which reduces 50 percent of the data points. (b) The perturbed data after a column-wise random projection which maps the data from 3D space onto 2D space. The random matrix is chosen from $N(0, 1)$ and the original data is given in Fig. 2a.
Lemma 5.2. Let $R$ be a $p \times q$ random matrix such that each entry $r_{i,j}$ of $R$ is independent and identically chosen from some unknown distribution with mean zero and variance $\sigma_r^2$, then

$$E[R^T R] = p\sigma_r^2 I \quad \text{and} \quad E[RR^T] = q\sigma_r^2 I.$$ 

Proof. Let $r_{i,j}$ and $\epsilon_{i,j}$ be the $i$, $ith$ entries of matrix $R$ and $R^T R$, respectively,

$$\epsilon_{i,j} = \sum_{t=1}^{p} r_{t,i}r_{t,j}$$

$$E[\epsilon_{i,j}] = E \left[ \sum_{t=1}^{p} r_{t,i}r_{t,j} \right] = \sum_{t=1}^{p} E[r_{t,i}r_{t,j}].$$
Since the entries of random matrix are independent and identically distributed (i.i.d.),

\[
E[\epsilon_{i,j}] = \begin{cases} 
\sum_{t=1}^{p} E[r_{t,i}] E[r_{t,j}] & \text{if } i \neq j; \\
\sum_{t=1}^{p} E[r_{t,i}^2] & \text{if } i = j.
\end{cases}
\]

Now, note that \(E[r_{i,j}] = 0\) and \(E[r_{i,j}^2] = \sigma_r^2\), therefore,

\[
E[\epsilon_{i,j}] = \begin{cases} 
0 & \text{if } i \neq j; \\
p \sigma_r^2 & \text{if } i = j.
\end{cases} \implies E[R^T R] = p \sigma_r^2 I.
\]

Similarly, we have \(E[RR^T] = q \sigma_r^2 I\).
Lemma 5.3 (ROW-WISE PROJECTION). Let $X$ and $Y$ be two data sets owned by Alice and Bob, respectively. $X$ is an $m \times n_1$ matrix, and $Y$ is an $m \times n_2$ matrix. Let $R$ be a $k \times m (k < m)$ random matrix such that each entry $r_{i,j}$ of $R$ is independent and identically chosen from some unknown distribution with mean zero and variance $\sigma_r^2$. Further, let

$$U = \frac{1}{\sqrt{k\sigma_r}} RX, \quad \text{and} \quad V = \frac{1}{\sqrt{k\sigma_r}} RY; \quad \text{then}$$

$$E[U^TV] = X^TY. \quad (4)$$
Lemma 5.4 (COLUMN-WISE PROJECTION). Let $X$ and $Y$ be two data sets owned by Alice and Bob, respectively. $X$ is an $m_1 \times n$ matrix and $Y$ is an $m_2 \times n$ matrix. Let $R$ be an $n \times k (k < n)$ random matrix such that each entry $r_{i,j}$ of $R$ is independent and identically chosen from some unknown distribution with mean zero and variance $\sigma_r^2$. Further, let

$$U = \frac{1}{\sqrt{k}\sigma_r} XR, \quad \text{and} \quad V = \frac{1}{\sqrt{k}\sigma_r} YR; \quad \text{then}$$

$$E[UV^T] = XY^T. \quad (5)$$
The above results show that the row-wise projection preserves the column-wise inner product and the column-wise projection preserves the row-wise inner product.

The beauty of this property is that inner product is directly related to many other distance-related metrics.

To be more specific:
The Euclidean distance of $x$ and $y$ is

$$
\|x - y\| = \sqrt{(x - y)^T(x - y)}.
$$
If the data vectors have been normalized to unity, then the cosine angle of $x$ and $y$ is

$$\cos \theta = \frac{x^T y}{\|x\| \cdot \|y\|} = x^T y.$$
If the data vectors have been normalized to unity with zero mean, the sample correlation coefficient of $x$ and $y$ is

$$\rho_{x,y} = \frac{\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{\left(\sum x_i^2 - \left(\frac{\sum x_i}{m}\right)^2\right) \left(\sum y_i^2 - \left(\frac{\sum y_i}{m}\right)^2\right)}} = x^T y.$$
Thus, if the data owner reduces the number of attributes of the data by projection, the statistical dependencies among the observations will be maintained; if the data owner compresses the observations, the relationship between the attributes will be preserved.
On the one hand, given only the perturbed data $U$ or $V$, one cannot determine the values of the original data $X$ or $Y$, which is based on the premise that the possible solutions are infinite when the number of equations is less than the number of unknowns. On the other hand, we can directly apply common data-mining algorithms on the perturbed data without accessing the original sensitive information.
1.2 Error Analysis

In practice, due to the cost of communication and security concerns, we always use one specific realization of the random matrix $R$. Therefore, we need to know more about the distribution of $R^TR$ (similarly, for $RR^T$) in order to quantify the utility of the random projection-based perturbation technique.
Assume entries of the $k \times m$ random matrix $R$ are i.i.d. and chosen from Gaussian distribution with mean zero and variance $\sigma_r^2$, we can study the statistical properties of the estimation of the inner product.
Let $\epsilon_{i,j}$ be the $i, j$th entry of matrix $R^T R$. It can be proved that $\epsilon_{i,j}$ is approximately Gaussian, $E[\epsilon_{i,i}] = k\sigma_r^2$, $Var[\epsilon_{i,i}] = 2k\sigma_r^4$, $\forall i$ and $E[\epsilon_{i,j}] = 0$, $Var[\epsilon_{i,j}] = k\sigma_r^4$, $\forall i, j, i \neq j$ (please see Appendix I for the proof which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm). The following lemma gives the mean and variance of the projection error.
Lemma 5.5. Let $x, y$ be two data vectors in $\mathbb{R}^m$. Let $R$ be a $k \times m$ random matrix. Each entry of $R$ is independent and identically chosen from Gaussian distribution with mean zero and variance $\sigma_r^2$. Further, let

$$u = \frac{1}{\sqrt{k}\sigma_r} Rx, \quad \text{and} \quad v = \frac{1}{\sqrt{k}\sigma_r} Ry; \quad \text{then}$$

$$E[u^T v - x^T y] = 0,$$

$$\text{Var}[u^T v - x^T y] = \frac{1}{k} \left( \sum_i x_i^2 \sum_i y_i^2 + \left( \sum_i x_i y_i \right)^2 \right).$$
In particular, if both $x$ and $y$ are normalized to unity, 
$\sum_i x_i^2 \sum_i y_i^2 = 1$ and $(\sum_i x_i y_i)^2 \leq 1$. We have the upper bound of the variance as follows:

$$Var[u^T v - x^T y] \leq \frac{2}{k}.$$  

**Proof.** Please see Appendix II which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm. □
Lemma 5.5 shows that the error \((u^T v - x^T y)\) of the inner product matrix produced by random projection-based perturbation technique is zero, on average, and the variance is at most the inverse of the dimensionality of the reduced space multiplied by 2 if the original data vectors are normalized to unity.

Actually, since \(\epsilon_{i,j}\) is approximately Gaussian, the error also has an approximate Gaussian distribution, namely, \(N(0, \sqrt{2/k})\).
To validate the above claim, we choose two randomly generated data sets from a uniform distribution in $[0, 1]$, each with 10,000 observations and 100 attributes. We normalize all the attributes to unity and compare the column-wise inner product of these two data sets before and after row-wise random projection.

Fig. 7a gives the results and it depicts that, even under 50 percent data projection rate (when $k=5,000$), the inner product still preserves very well after perturbation, and the error indeed approximates Gaussian distribution with mean zero and variance less than $2/k$. 
Fig. 7b shows the Root Mean Squared Error (RMSE) of the estimated inner product matrix with respect to the dimensionality of the reduced subspace. It can be seen that, as k increases, the error goes down exponentially, which means that the higher the dimensionality of the data, the better this technique works!
Fig. 7. (a) **Distribution of the error** of the estimated inner product matrix over two distributed data sets. Each data set contains 10,000 records and 100 attributes. $k = 50\% \times 10,000 = 5,000$ (50 percent row-wise projection). The random matrix is chosen from $N(0, 1)$. Note that the variance of the error is even smaller than the variance of distribution $N(0, \sqrt{2/k})$.

(b) **Root Mean Squared Error (RMSE)** of the estimated inner product matrix with respect to the dimensionality of the reduced subspace.
By applying Lemma 5.5 to the vector $x - y$, we have

$$E[||u - v||^2 - ||x - y||^2] = 0.$$  

If $x$ and $y$ are normalized to unity,

$$Var[||u - v||^2 - ||x - y||^2] \leq \frac{32}{k},$$

where $||x - y||^2 = (x - y)^T(x - y)$ is the square of the Euclidean distance of $x$ and $y$. Note that this bound defines the maximum variance of the distortion. As a generalization of [39, Theorem 2], we also have the probability bound of the Euclidean distance as follows:
Lemma 5.6. Let $x, y$ be two data vectors in $\mathbb{R}^m$. Let $R$ be a $k \times m$-dimensional random matrix. Each entry of the random matrix is independent and identically chosen from Gaussian distribution with mean zero and variance $\sigma_r^2$. Further, let

$$u = \frac{1}{\sqrt{k}\sigma_r} Rx, \quad \text{and} \quad v = \frac{1}{\sqrt{k}\sigma_r} Ry;$$

then

$$\Pr\{(1 - \epsilon) \|x - y\|^2 \leq \|u - v\|^2 \leq (1 + \epsilon) \|x - y\|^2\} \geq 1 - 2e^{-\frac{(\epsilon^2 - \epsilon^3)k}{4}}$$

for any $0 < \epsilon < 1$. 
Proof. Directly follows the proof of [39, Theorem 2] with the exception that random matrix is chosen independently according to $N(0, \sigma_r)$. 

This result also shows that as the reduced dimensionality $k$ increases, the distortion drops exponentially, which echoes the above observations that the higher the dimensionality of the data, the better the random projection works. Many applications of random projection can be found in the literature, e.g., image and text clustering [19] and distributed decision tree construction [20].


2. PRIVACY ANALYSIS

- Generally speaking, the **random projection-based multiplicative perturbation** technique guarantees that both the **dimensionality** and the **exact value of each element** of the original data are **kept confidential**.

- These properties are based on the assumptions that both **data and random noise** are from the **continuous real domain** and all the **participating parties** are **semi-honest**.
Consider the model $U = RX$, where $R \in \mathbb{R}^{k \times m}$ with $k < m$, and $X \in \mathbb{R}^{m \times n}$. This model can be viewed as a set of underdetermined systems of linear equations (more unknowns than equations), each with the form $u = Rx$, where $x$ is an $m \times 1$ column vector from $X$ and $u$ is the corresponding column vector from $U$. For each linear system, assume both $R$ and $u$ are known, so the solution is never unique. In practice, the system can be analyzed by the QR factorization [42] of $R^T$ such that

$$R^T = Q \left( \begin{array}{c} \overline{R} \\ 0 \end{array} \right),$$
\[ R^T = Q \begin{pmatrix} \overline{R} \\ 0 \end{pmatrix}, \]

where \( Q \) is an \( m \times m \) orthogonal matrix and \( \overline{R} \) is a \( k \times k \) upper triangular matrix. If \( R \) has full row rank, i.e., \( \text{rank}(R) = k \), there is a unique solution \( x_{\text{min-norm}} \) that minimizes \( \|x\|_2^2 \):

\[
x_{\text{min-norm}} = Q \begin{pmatrix} \overline{R}^{-1} \\ 0 \end{pmatrix} = Q \begin{pmatrix} \overline{R} \\ 0 \end{pmatrix} (\overline{R}^T \overline{R})^{-1} u
\]

\[
= R^T (RR^T)^{-1} u = R^\dagger u,
\]

where \( R^\dagger \) is nothing but the pseudoinverse of \( R \). This
solution $x_{\text{min-norm}}$ serves as a starting point to the under-determined system $u = Rx$. The complete solution set can be characterized by adding an arbitrary vector from the null space of $R$, which can be constructed by the rational basis for the null space of $R$, denoted by $N$. It can be confirmed that $RN = 0$ and that any vector $x$, where

$$x = x_{\text{min-norm}} + N\nu$$

for an arbitrary vector $\nu$ satisfies $u = Rx$. 
These results prove that, even if the random matrix $R$ is known to the adversary, it is impossible to find the exact values of all the elements in vector $x$ of each underdetermined system of linear equations. The best we can do is to find the minimum norm solution.

However, one may ask whether it is possible to completely identify some elements in the vector $x$. Obviously, if we can find as many linearly independent equations as some unknown elements, we can partially solve the system. In the following, we will discuss this possibility by using the “l-secure” definition introduced in [43, Definition 4.1].

3. This problem is referred to as finding a minimum norm solution to an underdetermined system of linear equations.

A coefficient matrix $R$ is said to be $l$-secure if, by removing any $l$ columns from $R$, the remaining sub-matrix still has full row rank, which guarantees that any nonzero linear combination of the row vectors of $R$ contains at least $l + 1$ nonzero elements.

Otherwise, assume there are at most $l$ nonzero elements. Then, if we remove these $l$ corresponding columns from $R$ and apply the same linear combination on all the row vectors of this remaining sub-matrix, we will get a zero vector, which means the row vectors of this sub-matrix are linearly dependent and the rank of this sub-matrix is not of full row rank, which contradicts the $l$-secure definition.
So, if a coefficient matrix is $l$-secure, each unknown variable in a linear equation is disguised (shaded) by at least $l$ other unknown variables no matter what kind of nonzero linear combination produces this equation. Now, the question is whether we can find $l + 1$ linearly independent equations that just involve these $l + 1$ unknowns?
The answer is No. It can be proved that any \( l + 1 \) nonzero linear combinations of the equations contains at least \( 2l + 1 \) unknown variables if these \( l + 1 \) vectors are linearly independent. The following theorem formalizes this property (which can be viewed as a generalization of [43, Theorem 4.3]).
Theorem 6.1. Let $\Upsilon$ be an $(l + 1) \times m$ matrix, where each row of $\Upsilon$ is a nonzero linear combination of row vectors in $R$. If $R$ is $l$-secure, the linear equations system $u = \Upsilon x$ involves at least $2l + 1$ unknown variables if these $l + 1$ vectors are linearly independent.\(^4\)

---

4. If these $l + 1$ vectors are not linearly independent, the $l + 1$ equations contain $\Gamma + l$ unknown variables. Here, $\Gamma$ denotes the rank of the matrix formed by these $l + 1$ vectors.
Proof. Since row vectors of $\Upsilon$ are all linearly independent, $u = \Upsilon x$ can be transformed into $u = (I : \Upsilon') x$ through a proper Gaussian elimination, where $I$ is the $(l + 1) \times (l + 1)$ identity matrix, $\Upsilon'$ is a $(l + 1) \times (m - (l + 1))$ matrix, and $(I : \Upsilon')$ is a vertical concatenation of $I$ and $\Upsilon'$. Since $R$ is $l$-secure, each row of $(I : \Upsilon')$ contains at least $l + 1$ nonzero entries, which corresponds to $l + 1$ unknowns. Because in each row of $(I : \Upsilon')$, there is a single 1 from $I$, there are at least $l$ nonzero entries in $\Upsilon'$. Thus, the whole system contains at least $2l + 1$ unknowns, with $l + 1$ unknowns being contributed by $I$, and at least $l$ unknowns from $\Upsilon'$. $\square$
In summary, if a coefficient matrix is $l$-secure, any linear combinations of the equations contains at least $l + 1$ variables and it is not possible to find $l + 1$ linearly independent equations that just involve the same $l + 1$ variables, thus the solutions to any partial unknown variables are infinite.

Now, consider the $k \times m$ random projection matrix and the restrictions of ICA we discussed in the previous sections. When $m = 2k - 1$, after removing any $k - 1$ columns from mixing matrix $R$, according to the proof of Theorem 4.4, the remaining square matrix has full row rank with probability 1.
That means **the system is (k-1)-secure with probability 1** when the mixing matrix \( R \) is known to the adversary, i.e., theoretically, each unknown variable is disguised by at least \( k-1 \) variables, and we cannot find \( k-1 \) linearly independent equations that just involve these variables, so the solutions are infinite.

When \( m > 2k-1 \), the security level is even higher because we can remove more columns while keeping the sub-matrix full row rank (however, the accuracy of the random projection will probably be compromised if \( k \) is too small).
Remark:

This result shows that, even if the random matrix $R$ is known to the adversary, if $R$ is $(k - 1)$-secure, each unknown variable is masked by at least $k - 1$ other unknown variables no matter how the equations are linear combined. So, it is impossible to find the exact value of any element in the original data.
Since the exact values of the original data cannot be identified, let us change gears and see how well can we estimate them if both the perturbed data and the specific random matrix are known (however, we assume the adversary does not know the true variance of the random entries, and, in practice, an estimated one may be used instead.).
Recall the projection model described in Section 5. If entries of the $k \times m$ random matrix $R$ are independent and identically chosen from Gaussian distribution with mean zero and variance $\sigma_r^2$, given $u = \frac{1}{\sqrt{k\sigma_r}} Rx$, we can estimate $x$ by multiplying on the left by $\frac{1}{\sqrt{k\hat{\sigma}_r}} R^T$, where $\hat{\sigma}_r$ is the estimated variance of the random entries. Note that, in practice, since the specific realization of $R$ is disclosed, an adversary can compute $\hat{\sigma}_r$ by computing the sample variance of $r_{i,j}$. Therefore, in the following equations, we view $\hat{\sigma}_r$ as a constant. We have

$$\frac{1}{\sqrt{k\hat{\sigma}_r}} R^T u = \frac{1}{k\hat{\sigma}_r \sigma_r} R^T R x.$$
The estimation for the $i$th data element of vector $x$, denoted by $\hat{x}_i$, can be expressed as

$$\hat{x}_i = \frac{1}{k\hat{\sigma}_r \sigma_r} \sum_t \epsilon_{i,t} x_t,$$

where $\epsilon_{i,j}$ is the $i,j$th entry of $R^TR$. With simple mathematical derivation, we have the expectation and variance of the estimation as follows:
\[ E[\hat{x}_i] = \frac{\sigma_r}{\hat{\sigma}_r} x_i, \]
\[ Var[\hat{x}_i] = \frac{1}{k^2 \sigma_r^2 \sigma_r^2} \left( (2k + k^2)\sigma_r^4 x_i^2 + k\sigma_r^4 \sum_{t, t \neq i} x_t^2 \right) - \left( \frac{\sigma_r}{\hat{\sigma}_r} x_i \right)^2. \]

When the estimated variance \( \hat{\sigma}_r^2 \approx \sigma_r^2 \), we have
\[ E[x_i - \hat{x}_i] \approx 0, \]
\[ Var[x_i - \hat{x}_i] \approx \frac{2}{k} x_i^2 + \frac{1}{k} \sum_{t, t \neq i} x_t^2. \]
In summary, when the random matrix is completely disclosed, one cannot find the exact value of any element of the original data. However, by exploring the properties of the random matrix $R$, we can find an approximation of the original data. The distortion is zero on average, and its variance is approximately $\frac{2}{k}x_i^2 + \frac{1}{k} \sum_{t,t\neq i} x_t^2$.

We view this variance as a privacy measure in the worst case. By controlling the magnitude of the vector $x$ (which can be done by simply multiplying a scalar to each element of the vector), we can adjust the variance of the distortion of the estimation, which, in turn, changes the privacy level.
2.2 The Dimensionality and the Distribution of the Random Matrix Are Disclosed

This section studies whether an adversary can get a good estimation of the original data through a random guess of the random matrix if he or she knows the probability density function (PDF) of R and its dimensionality m.
Assume the adversary generated a random matrix $\hat{R}$ according to the PDF. Given $u = Rx$, the adversary can estimate $x$ by multiplying on the left of $u$ by $\frac{1}{\sqrt{k\sigma_r}} \hat{R}^T$

$$\frac{1}{\sqrt{k\sigma_r}} \hat{R}^T u = \frac{1}{\sqrt{k\sigma_r}} \hat{R}^T \frac{1}{\sqrt{k\sigma_r}} Rx.$$ 

Let $\hat{e}_{i,j}$ denote the $i,j$th entry of $\hat{R}^T R$ such that $\hat{e}_{i,j} = \sum_t \hat{r}_{t,i} r_{t,j} \forall i,j$. Let $\hat{x}_i$ denote the estimation of $x_i$, we have

$$\hat{x}_i = \frac{1}{k\sigma_r^2} \sum_t \hat{e}_{i,t} x_t.$$
The expectation and variance of $\hat{x}_i$ are

$$E[\hat{x}_i] = E \left[ \frac{1}{k\sigma_r^2} \sum_t \hat{e}_{i,t} x_t \right] = 0,$$

$$\text{Var}[\hat{x}_i] = E \left[ \frac{1}{k^2 \sigma_r^4} \left( \sum_t \hat{e}_{i,t} x_t \right)^2 \right] = \frac{1}{k} \sum_t x_t^2.$$ 

Here, we use the fact that $E[\hat{e}_{i,j}] = 0$, $E_{p \neq q}[\hat{e}_{i,p} \hat{e}_{i,q}] = 0$ and $E[\hat{e}_{i,t}^2] = k\sigma_r^4$.

This fact indicates that the adversary cannot identify the original data by a random guess of the random matrix, all she or he can get is approximately a null matrix with all entries being around 0.
2.3 The Data Inputs are Restricted to Boolean

In the discussion of Section 2.1, we do not assume any prior knowledge of the original data with the exception that it is from the continuous real domain. However, when the data inputs are restricted to Boolean, our protocol will be at a high disclosure risk. For example, suppose the adversary knows the random matrix is (0.1, 0.3, 0.5) and the perturbation equation is $0.1d_1 + 0.3d_2 + 0.5d_3 = 0.9$, where $(d_1, d_2, d_3)$ is the original data. Then, even though there is just one equation, the adversary will know that $d_1 = d_2 = d_3 = 1$. 
Actually, if the system of linear equations has a unique solution (either for all the unknowns or for partial unknowns), the adversary could try all possible combinations of 1 and 0 for all the data elements to obtain the correct solution. Similar results will occur if the data is discrete and the adversary knows exactly all the possible candidates.
However, we need to note that, in practice, both the dimensionality of the data and the random matrix are kept secret, so the adversary does not know the equation “$0.1d_1 + 0.3d_2 + 0.5d_3 = 0.9,$” but only a single number 0.9. Therefore, the random projection-based perturbation offers a reasonable protection for Boolean and other discrete data.
2.4 The Distribution of the Data is Revealed

- Recall that, if all the sources are non-Gaussian and statistically independent, it is possible for over-complete ICA to identify the mixing matrix up to scaling and permutation ambiguities.

- If the adversary also happens to know the distribution of the original data sources under this situation, over-complete ICA could possibly reconstruct the sources in a probabilistic sense.
However, in the literature, over-complete ICA has only been treated in particular cases, and an exact recovery is still impossible.

Actually, in practice, the data sets usually have more than one Gaussians and correlated components, ICA can only find the “real” hidden independent factors behind the original data, but not the data itself.
2.5 The Trouble with Malicious Parties

The perturbation technique we discussed assumes a **semi-honest model**, which means all the parties follow the protocol properly and there is no collusion. However, it is possible that the **data miner and one of the data owners are malicious** and they want to **cooperatively extract the sensitive information from the other party**. For example, to probe Bob’s private data, Alice may reveal the secret random matrix to the data miner or the data miner may send Bob’s perturbed data back to Alice. **These behaviors are actually the same as disclosing the specific realization of the random matrix**, which is well studied in Section 2.1.
3. APPLICATIONS

3.1 Inner Product/Euclidean Distance Estimation from Heterogeneously Distributed Data

Problem. Let $X$ be an $m \times n_1$ data matrix owned by Alice and $Y$ be an $m \times n_2$ matrix owned by Bob. Compute the column-wise inner product and Euclidean distance matrices of the data ($X : Y$) without directly accessing it.
Algorithm:

1. Alice and Bob cooperatively generate a secret random seed and use this seed to generate a \( k \times m \) random matrix \( R \).
2. Alice and Bob project their data onto \( \mathbb{R}^k \) using \( R \) and release the perturbed version \( U = \frac{1}{\sqrt{k} \sigma_r} RX \) and \( V = \frac{1}{\sqrt{k} \sigma_r} RY \) to a third party.
3. The third party computes the inner product matrix using the perturbed data \( U \) and \( V \) and gets
\[
\begin{pmatrix}
U^T U & U^T V \\
V^T U & V^T V
\end{pmatrix}
\approx
\begin{pmatrix}
X^T X & X^T Y \\
Y^T X & Y^T Y
\end{pmatrix}.
\]

**Discussions:** Similarly, the third party can compute the Euclidean distance on the perturbed data. When the data is properly normalized, the inner product matrix is nothing but the cosine angle matrix or the correlation coefficient matrix of \((X : Y)\).
3.2 K-Means Clustering from Homogeneously Distributed Data

Problem. Let $X$ be an $m_1 \times n$ data matrix owned by Alice and $Y$ be an $m_2 \times n$ matrix owned by Bob. Cluster the union of these two data sets $\left( \frac{X}{Y} \right)$ without directly accessing the raw data.

Algorithm:

1. Alice and Bob cooperatively generate a secret random seed and use this seed to generate an $n \times k$ random matrix $R$.
2. Alice and Bob project their data onto $\mathbb{R}^k$ using $R$ and release the perturbed version $U = \frac{1}{\sqrt{k\sigma_r}}XR$ and $V = \frac{1}{\sqrt{k\sigma_r}}YR$.
3. The third party does K-Means clustering over the data set $\left( \frac{U}{V} \right)$. 
Discussions: The above algorithm is based on the fact that column-wise projection preserves the distance of row vectors. Actually, random projection maps the data to a lower-dimensional random space while maintaining much of its variance just like PCA. However, random projection only requires $O(mnk)(k << n)$ computations to project an $m \times n$ data matrix into $k \times n$ dimensions, while the computation complexity of estimating the PCA is $O(n^2m) + O(n^3)$. 
3.3 Linear Classification

**Problem.** Given a collection of sensitive data points $x_i (i = 1, 2, \ldots)$ in $\mathbb{R}^n$, each labeled as positive or negative, find a weight vector $w$ such that $wx_i^T > 0$ for all positive points $x_i$ and $wx_i^T < 0$ for all negative points $x$. Here, we assume $x_i (i = 1, 2, \ldots)$ is a row vector.

**Algorithm:**

1. The data owner generates an $n \times k$ random matrix $R$ and projects the data to $\mathbb{R}^k$ using $R$ such that $x_i' = \frac{1}{\sqrt{k}\sigma_r} x_i R$, \( \forall i \), and releases the perturbed data.
2. Run the perceptron algorithm in $\mathbb{IR}^k$:
   
a. Let $w' = 0$. Do until all the examples are correctly classified
   
   - Pick an arbitrary misclassified example $x'_i$ and let
     
     $$w' \leftarrow w' + \eta \cdot \text{classlabel}(x'_i) \cdot x'_i.$$  

     Here, $\eta$ is the learning rate.
Discussions: Note that, in this algorithm, the class labels are not perturbed. Future example $x$ is labeled positive if $w'\left(\frac{1}{\sqrt{k}\sigma_r}xR\right)^T > 0$ and negative otherwise. This is actually the same as checking whether $(w' \frac{1}{\sqrt{k}\sigma_r} R^T)x^T > 0$, namely, a linear separator in the original n-dimensional space. This also implies that $w'$ is nothing but the projection of $w$ such that $w' = \frac{1}{\sqrt{k}\sigma_r} w R$ and, therefore,

$$w'x_i^T = \frac{1}{\sqrt{k}\sigma_r} w R \frac{1}{\sqrt{k}\sigma_r} R^T x_i^T \approx wx_i^T.$$
This algorithm can be easily generalized for **Support Vector Machine (SVM)** because, in the Lagrangian dual problem of the SVM task, the relationship of the original data points is completely quantified by **inner product**.

The random projection-based technique may be even more powerful when used with some other geometric transformation techniques like **scaling**, **translation**, and **rotation**. Combining this with SMC-based techniques offers another interesting direction.