# Randomization in Privacy Preserving Data Mining

Alexandre Evfimievski

Cornell University
Ithaca, NY 14853, USA
aevf@cs.cornell.edu

## ABSTRACT

Suppose there are many clients, each having some personal information, and one server, which is interested only in aggregate, statistically significant, properties of this information. The clients can protect privacy of their data by perturbing it with a randomization algorithm and then submitting the randomized version. The randomization algorithm is chosen so that aggregate properties of the data can be recovered with sufficient precision, while individual entries are significantly distorted. How much distortion is needed to protect privacy can be determined using a privacy measure. Several possible privacy measures are known; finding the best measure is an open question. This paper presents some methods and results in randomization for numerical and categorical data, and discusses the issue of measuring privacy.

## 1. INTRODUCTION

Suppose that some company needs to construct an aggregate model of its customers' personal data. For example, a retail store wants to know the age and income of its customers who are more likely to buy DVD players or mountain ski equipment; a movie recommendation system would like to learn users' movie preferences in order to make advertisements more targeted; or an on-line business arranges its webpages according to an aggregate model of its website visitors. In all these cases, there is one central server (the company), and many clients (the customers), each having a piece of information. The server collects this information and builds its aggregate model using, for example, a classification algorithm or an algorithm for mining association rules. Often the resulting model no longer contains personally identifiable information, but contains only averages over large groups of clients.

The usual solution to the above problem consists in having all clients send their personal information to the server. However, many people are becoming increasingly concerned about the privacy of their personal data. They would like to avoid giving out much more about themselves than is required to run their business with the company. If all the company needs is the aggregate model, a solution is preferred that reduces the disclosure of private data while still allowing the server to build the model. One possibility is as follows: before sending its piece of data, each client perturbs it so that some true information is taken away and some false information is introduced. This approach is called *randomization*. Another possibility is to decrease precision of the transmitted data by rounding, suppressing certain values, replacing values with intervals, or replacing categorical values by more general categories up the taxonomical hierarchy, see [8; 14; 23; 24].

The usage of randomization for preserving privacy has been studied extensively in the framework of statistical databases [9; 10; 12; 13; 20]. In that case, the server has a complete and precise database with the information from its clients, and it has to make a version of this database public, for others to work with. One important example is census data: the government of a country collects private information about its inhabitants, and then has to turn this data into a tool for research and economic planning. However, it is assumed that private records of any given person should not be released nor be recoverable from what is released. In particular, a company should not be able to match up records in the publicly released database with the corresponding records in the company's own database of its customers.

In the case of statistical databases, however, the database is randomized when it is already fully known. This is different from our problem, where the randomization procedure is run on the client's side, and must be decided upon before the data is collected. A randomization for a statistical database is usually chosen so that it preserves certain aggregate characteristics (averages and covariance matrices for numerical data, or marginal totals in contingency tables for categorical data), or changes them in a predetermined way [12; 15]. Besides randomization, other privacy preserving transformations are used such as sampling and swapping values among records [15; 27]. Our choice is more limited due to the nature of our problem.

## 2. NUMERICAL RANDOMIZATION

Let each client $C_i$, $i = 1, 2, \ldots, N$, have a numerical attribute $x_i$. Assume that each $x_i$ is an instance of random variable $X_i$, where all $X_i$ are independent and identically distributed. The cumulative distribution function (the same for every $X_i$) is denoted by $F_X$. The server wants to learn the function $F_X$, or its close approximation; this is the aggregate model which the server is allowed to know. The server can know anything about the clients that is derivable from the model, but we would like to limit what the server knows about the actual instances $x_i$.

The paper [4] proposes the following solution. Each client randomizes its $x_i$ by adding to it a random shift $y_i$. The shift values $y_i$ are independent identically distributed random variables with cumulative distribution function $F_Y$; their distribution is chosen in advance and is known to the server. Thus, client $C_i$ sends randomized value $z_i = x_i + y_i$ to the server, and the server's task is to approximate function $F_X$ given $F_Y$ and values $z_1, z_2, \ldots, z_N$. Also, it is necessary to understand how to choose $F_Y$ so that

- the server can approximate $F_X$ reasonably well, and
- the value of $z_i$ does not disclose too much about $x_i$.

The amount of disclosure is measured in [4] in terms of confidence intervals. Given confidence $c\%$, for each randomized value $z$ we can define an interval $[z - w_1, z + w_2]$ such that for all nonrandomized values $x$ we have

$$\mathbf{P}\left[Z - w_1 \leqslant x \leqslant Z + w_2 \;\mid\; Z = x + Y,\; Y \sim F_Y\right] \;\geqslant\; c\%.$$

In other words, here we consider an "attack" where the server computes a $c\%$-likely interval for the private value $x$ given the randomized value $z$ that it sees. The shortest width $w = w_1 + w_2$ for a confidence interval is used as the *amount of privacy* at $c\%$ confidence level.

Once the distribution function $F_Y$ is determined and the data is randomized, the server faces the *reconstruction problem*: Given $F_Y$ and the realizations of $N$ i.i.d. random samples $Z_1, Z_2, \ldots, Z_N$, where $Z_i = X_i + Y_i$, estimate $F_X$. In [4] this problem is solved by an iterative algorithm based on Bayes' rule. Denote the density of $X_i$ (the derivative of $F_X$) by $f_X$, and the density of $Y_i$ (the derivative of $F_Y$) by $f_Y$; then the reconstruction algorithm is as follows:

1. $f_X^0$ := Uniform distribution;
2. $j := 0$ // Iteration number;
3. `repeat`

   (a) $f_X^{j+1}(a) := \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \dfrac{f_Y(z_i - a)\, f_X^j(a)}{\int_{-\infty}^{\infty} f_Y(z_i - z)\, f_X^j(z)\, dz}$;

   (b) $j := j + 1$;

   `until` (stopping criterion met).

For efficiency, the density functions $f_X^j$ are approximated by piecewise constant functions over a partition of the attribute domain into $k$ intervals $I_1, I_2, \ldots, I_k$. The formula in the algorithm above is approximated by ($m(I_t)$ is the midpoint of $I_t$):

$$f_X^{j+1}(I_p) := \frac{1}{N} \sum_{i=1}^{N} \frac{f_Y(m(z_i) - m(I_p))\, f_X^j(I_p)}{\sum_{t=1}^{k} f_Y(m(z_i) - m(I_t))\, f_X^j(I_t)\, |I_t|} \qquad (1)$$

It can also be written in terms of cumulative distribution functions, where $\Delta F_X((a, b]) = F_X(b) - F_X(a) = \mathbf{P}[a < X \leqslant b]$ and $N(I_s)$ is the number of randomized values $z_i$ inside interval $I_s$:

$$\Delta F_X^{j+1}(I_p) := \sum_{s=1}^{k} \frac{N(I_s)}{N} \frac{f_Y(m(I_s) - m(I_p))\, \Delta F_X^j(I_p)}{\sum_{t=1}^{k} f_Y(m(I_s) - m(I_t))\, \Delta F_X^j(I_t)}$$

In paper [1] it has been shown that if the formula (1) is replaced with formula

$$f_X^{j+1}(I_p) := \frac{1}{|I_p|\, N} \sum_{i=1}^{N} \frac{\Delta F_Y(z_i - I_p)\, f_X^j(I_p)}{\sum_{t=1}^{k} \Delta F_Y(z_i - I_t)\, f_X^j(I_t)},$$

where $z - (a, b] = [z - b, z - a)$, then the quality of approximation is somewhat better. This formula is derived using the framework of Expectation-Maximization (EM) algorithms [7; 19], which allows to consider a more general randomization setting. Suppose the server wants to approximate density $f_X$ of nonrandomized (original) attribute distribution by some density from a parametric family $\{f_{X;\Theta}\}$. If the server knew $x_1, x_2, \ldots, x_N$, it could find a maximum likelihood parameter value $\hat{\Theta}$ by computing

$$\hat{\Theta} = \operatorname*{argmax}_{\Theta} \log \prod_{i=1}^{N} f_{X;\Theta}(x_i)$$

However, the server knows only $z_1, z_2, \ldots, z_N$; so this formula is replaced by an iterative EM procedure:

$$\Theta^{k+1} = \operatorname*{argmax}_{\Theta} \mathbf{E}\left[\log \prod_{i=1}^{N} f_{X;\Theta}(X_i) \;\middle|\; \begin{array}{c} f_X = f_{X;\Theta^k}, \\ \forall i:\; X_i + Y_i = z_i \end{array}\right]$$

A similar approach has been described earlier for statistical databases [18]. Both papers [4; 1] consider two specific examples of additive randomization: with a uniform (on a segment) and a Gaussian density $f_Y$ for the shift distribution.

In [4] it is shown how to use randomized numerical data in classification, namely in building a decision tree [5]. The main problem in decision tree construction is finding the right split point at each node. The quality of a split point depends on the frequency of records from each class in the subsets to the left and to the right of the split point. So, the tree building algorithm has to estimate the frequency distributions for points of each class, split by different split points for all numerical attributes. Several estimating strategies are considered, which differ in how many times the iterative reconstruction algorithm is applied:

- **Global**: once for each attribute, at root;
- **ByClass**: once for each class and attribute, at root;
- **Local**: once for each class and attribute at each node.

Since the reconstruction algorithm requires partitioning the attribute domains into intervals, the only split points considered are the interval boundaries. To estimate the class frequencies for a split at a non-root node, the records are associated with attribute intervals as follows. The records are sorted by each (randomized) attribute, and then, given the reconstruction of distribution $F_X$, they are associated with an interval according to their order and so as to observe the distribution. Experimental results show that the class prediction accuracy for decision trees constructed over randomized data (using **ByClass** or **Local**) is reasonably close (within 5%–15%) to the trees constructed over original data, even with heavy enough randomization to have 95%-confidence intervals as wide as the whole range of an attribute. The training set had 100,000 records.

## 3. ITEMSET RANDOMIZATION

Papers [22; 11] consider randomization of categorical data, in the context of association rules. Suppose that each client $C_i$ has a *transaction* $t_i$, which is a subset of a given finite set of items $\mathcal{I}$, $|\mathcal{I}| = n$. For any subset $A \subset \mathcal{I}$, its *support* in the dataset of transactions $T = \{t_i\}_{i=1}^N$ is defined as the fraction of transactions containing $A$ as their subset:

$$\mathrm{supp}^T(A) \;:=\; |\{t_i \mid A \subseteq t_i, i = 1 \ldots N\}| \;/\; N;$$

an itemset $A$ is *frequent* if its support is at least a certain threshold $s_{\min}$. An *association rule* $A \Rightarrow B$ is a pair of disjoint itemsets $A$ and $B$; its support is the support of $A \cup B$, and its *confidence* is the fraction of transactions containing $A$ that also contain $B$:

$$\mathrm{conf}^T(A \Rightarrow B) \;:=\; \mathrm{supp}^T(A \cup B) \;/\, \mathrm{supp}^T(A)\,.$$

An association rule *holds* for $T$ if its support is at least $s_{\min}$ and its confidence is at least $c_{\min}$, which is another threshold. Association rules were introduced in [2], and [3] presents efficient algorithm Apriori for mining association rules that hold for a given dataset. The idea of Apriori is to make use of *antimonotonicity property*:

$$\forall A \subseteq B :\; \mathrm{supp}^T(A) \geqslant \mathrm{supp}^T(B)\,.$$

Conceptually, it first finds frequent 1-item sets, then checks the support of all 2-item sets whose 1-subsets are frequent, then checks all 3-item sets whose 2-subsets are frequent, etc. It stops when no candidate itemsets (with frequent subsets) can be formed. It is easy to see that the problem of finding association rules can be reduced to finding frequent itemsets.

A natural way to randomize a set of items is by deleting some items and inserting some new items. Paper [11] considers a family of randomization operators called *select-a-size*. A select-a-size randomization operator is defined for a fixed transaction size $|t| = m$ and has two parameters: a randomization level $0 \leqslant \rho \leqslant 1$ and a probability distribution $(p[0], p[1], \ldots, p[m])$ over set $\{0, 1, \ldots, m\}$. Given a transaction $t$ of size $m$, the operator generates a randomized transaction $t'$ as follows:

1. The operator selects an integer $j$ at random from the set $\{0, 1, \ldots, m\}$ so that $\mathbf{P}\,[j \text{ is selected}] = p[j]$.

2. It selects $j$ items from $t$, uniformly at random (without replacement). These items, and no other items of $t$, are placed into $t'$.

3. It considers each item $a \notin t$ in turn and tosses a coin with probability $\rho$ of "heads" and $1 - \rho$ of "tails". All those items for which the coin faces "heads" are added to $t'$.

If different clients have transactions of different sizes, then select-a-size parameters have to be chosen for each transaction size. So, this (nonrandomized) size has to be transmitted to the server with the randomized transaction. The randomization operator used in [22] does not have this drawback; it has only one parameter $0 \leqslant p \leqslant 1$ which determines, for each item independently, the probability of the item not being "flipped" (discarded if present, or inserted if absent) in the transaction. For any fixed transaction size $m$, this operator becomes a special case of select-a-size, with $\rho = 1 - p$ and $p[j] = \binom{m}{j} p^j (1-p)^{m-j}$.

In the set $T'$ of randomized transactions available to the server, itemsets have supports very different from their supports in the nonrandomized dataset $T$. Therefore, techniques were developed that allow to estimate original supports given randomized supports. It is important to note that randomized support of an itemset $A$ is a random variable that depends on the original supports of all subsets of this itemset. Indeed, a transaction that contains all but one item of $A$ has a very different probability to contain $A$ after randomization than a transaction that contains no items of $A$. So, in [11] the behavior of itemset $A$, $|A| = k$, w.r.t. randomization is characterized by the $(k+1)$-vector of its *partial supports* $\vec{s} = (s_0, s_1, \ldots, s_k)^T$, where

$$s_l \;:=\; |\{t_i \,:\, |A \cap t_i| = l,\;\; i = 1 \ldots N\}| \;/\; N.$$

It is shown that the vector $\vec{s}\,'$ of randomized partial supports is distributed as $1/N$ times a sum of multinomial distributions, with its expectation and covariance matrix being

$$\mathbf{E}\;\vec{s}\,' \;=\; P \cdot \vec{s}, \quad \mathbf{Cov}\;\vec{s}\,' \;=\; \frac{1}{N} \cdot \sum_{l=0}^{k} s_l\, D[l],$$

for $(k+1) \times (k+1)$ matrices $P$ and $D[0], D[1], \ldots, D[k]$ that depend on the parameters of the randomization operator. Matrix $P$ is defined as

$$P_{l\,l'} \;\;=\;\; \mathbf{P}\,[\,|R(t) \cap A| = l \;\mid\; |t \cap A| = l'\,]$$

where $R$ is the randomization operator. Computing the inverse matrix $Q = P^{-1}$ gives an unbiased estimator $\vec{s}_{\mathrm{est}}$ for $\vec{s}$, as well as the estimator's covariance matrix and its unbiased estimator:

$$\vec{s}_{\mathrm{est}} = Q \cdot \vec{s}\,'; \quad (\mathbf{Cov}\;\vec{s}_{\mathrm{est}})_{\mathrm{est}} = \frac{1}{N} \cdot \sum_{l=0}^{k} (\vec{s}_{\mathrm{est}})_l\, Q\, D[l]\, Q^T.$$

In particular, it lets us estimate the nonrandomized support $s$ of $A$ and its variance:

$$s_{\mathrm{est}} = \sum_{l=0}^{k} s'_l\, Q_{k\,l}\,; \quad (\mathbf{Var}\;s_{\mathrm{est}})_{\mathrm{est}} = \frac{1}{N} \cdot \sum_{l=0}^{k} s'_l (Q_{k\,l}^2 - Q_{k\,l}).$$

The support estimator formula can be used inside Apriori algorithm for mining frequent itemsets, so that the algorithm works over randomized dataset. However, since the estimator is a random variable, it may violate the antimonotonicity property. This may cause an itemset to be discarded even though its estimated support, as well as its true support, is above threshold. This effect can be reduced by lowering the threshold by an amount proportional to the standard deviation of the estimator.

## 4. LIMITING PRIVACY BREACHES

Consider the following simple randomization $R$: given a transaction $t$, we consider each item in turn, and with probability 80% replace it with a new random item; with probability 20% we leave the item unchanged. Since most of the items get replaced, we may suppose that this randomization preserves privacy well. However, it is not so, at least not all the time. Indeed, let $A = \{x, y, z\}$ be a 3-item set with partial supports

$$s_3 = \mathrm{supp}^T(A) = 1\%; \quad s_2 = 5\%; \quad s_1 + s_0 = 94\%.$$

Assume that overall there are 10,000 items and 10 million transactions, all of size 10. Then 100,000 transactions contain $A$, and 500,000 more transactions contain all but one items of $A$. How many of these transactions contain $A$ after they are randomized? The following is a rough average estimate:

$$A \subset t \text{ and } A \subset R(t) : 100{,}000 \cdot 0.2^3 = 800$$

$$|A \cap t| = 2 \text{ and } A \subset R(t) : 500{,}000 \cdot 0.2^2 \cdot \frac{8 \cdot 0.8}{10{,}000} = 12.8$$

$$|A \cap t| \leqslant 1 \text{ and } A \subset R(t) : < 10^7 \cdot 0.2 \cdot \left(\frac{9 \cdot 0.8}{10{,}000}\right)^2 \approx 1.04$$

So, there will be about 814 randomized transactions containing $A$, out of which about 800, or 98%, contained $A$ before randomization as well. Now, suppose that the server receives from client $C_i$ a randomized transaction $R(t)$ that contains $A$. The server now knows that the actual, nonrandomized transaction $t$ at $C_i$ contains $A$ with probability about 98%. On the other hand, the prior probability of $A \subset t$ is just 1%. The disclosure of $A \subset R(t)$ has caused a probability jump from 1% to 98%. Paper [11] calls this situation a *privacy breach*.

Intuitively, a privacy breach with respect to some property $P(t)$ occurs when, for some possible outcome of randomization (= some possible view of the server), the posterior probability of $P(t)$ is higher than a given threshold called *the privacy breach level*. Of course, there are always some properties that are likely; so, we have to only look at "interesting" properties, such as the presence of a given item in $t$. Statistical database literature considers a similar notion (*pessimistic risk*) for record identification [17]. In [11], the following special case of breaches is considered: Itemset $A$ *causes a privacy breach of level* $p_{\mathrm{b}}$ if for some item $a \in A$ and some transaction $t$ we have $\mathbf{P}\left[a \in t \mid A \subseteq R(t)\right] \geqslant p_{\mathrm{b}}$. It is assumed here that the transaction at each client is an independent instance of a distribution over transactions.

In order to prevent privacy breaches from happening, paper [11] suggests to randomize transactions by inserting many "false" items, as well as deleting some "true" items. So many "false" items should be inserted into a transaction that one is as likely to see a "false" itemset as a "true" one. In select-a-size randomization operator, it is the randomization level $\rho$ that determines the probability of a "false" item to be inserted. The other parameters, namely the distribution $(p[0], p[1], \ldots, p[m])$, are set in [11] so that, for a certain "cutoff" integer $K$, any number of items from 0 to $K$ is retained from the original transaction with probability $1/(K+1)$, while the rest of the items are inserted independently with probability $\rho$. The question of optimizing all select-a-size parameters to achieve maximum recoverability for a given breach level is left open.

The parameters of randomization are checked for privacy as follows. It is assumed that the server knows the maximum possible support of an itemset for each itemset size, among transactions of each transaction size, or their upper bounds. Based on this knowledge, the server computes partial supports for (imaginary) *privacy-challenging itemsets*, and tests randomization parameters by computing posterior probabilities $\mathbf{P}\left[a \in t \mid A \subseteq R(t)\right]$ from the definition of privacy breaches. The randomization parameters are selected to keep variance low while preventing privacy breaches for privacy-challenging itemsets.

Graphs and experiments with real-life datasets show that, given several million transactions, it is possible to find randomization parameters so that the majority of 1-item, 2-item, and 3-item sets with support at least 0.2% can be recovered from randomized data, for privacy breach level of 50%. However, long transactions (longer than about 10 items) have to be discarded, because the privacy-preserving randomization parameters for them must be "too randomizing," saving too little for support recovery; in both real-life datasets used in [11] most transactions had 5 items or less. Those itemsets that were recovered incorrectly ("false drops" and "false positives") were usually close to the support threshold, i.e. there were few outliers. The standard deviation for 3-itemset support estimator was at most 0.07% for one dataset and less than 0.05% for the other; for 1-item and 2-item sets it is smaller still.

## 5. MEASURES OF PRIVACY

Each of the papers [4; 1; 22; 11] suggests its own way of measuring privacy, and there are other suggestions in the literature. In [4] (see Section 2) privacy is measured in terms of confidence intervals. The nonrandomized numerical attribute $x_i$ is treated as an unknown parameter of the distribution of the randomized value $Z_i = x_i + Y_i$. Given an instance $z_i$ of the randomized value $Z_i$, the server can compute an interval $I(z_i) = [x_-(z_i), x_+(z_i)]$ such that $x_i \in I(z_i)$ with at least certain probability $c\%$; this should be true for all $x_i$. The length $|I(z_i)|$ of this confidence interval is treated as a privacy measure of the randomization.

Unfortunately, as pointed out in [1], this measure can be misleading. One problem is that the domain of the nonrandomized value and its distribution are not specified. Consider an attribute $X$ with the following density function:

$$f_X(x) \;=\; \begin{cases} 0.5 & \text{if } 0 \leqslant x \leqslant 1 \text{ or } 4 \leqslant x \leqslant 5 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Assume that the perturbing additive $Y$ is distributed uniformly in $[-1, 1]$; then, according to the confidence interval measure, the amount of privacy is 2 at confidence level 100%. However, if we take into account the fact that $X$ must be within $[0, 1] \cup [4, 5]$, we can always compute a confidence interval of size 1 (not 2). The interval is computed as follows:

$$I(z) \;=\; \begin{cases} [0, 1] & \text{if } -1 \leqslant z \leqslant 2 \\ [4, 5] & \text{if } 3 \leqslant z \leqslant 6 \end{cases}$$

Moreover, in many cases the confidence interval can be even shorter: for example, for $z = -0.5$ we can give interval $[0, 0.5]$ of size 0.5.

Paper [1] suggests to measure privacy using Shannon's information theory [26; 25]. The average amount of information in the nonrandomized attribute $X$ depends on its distribution and is measured by its differential entropy

$$h(X) \;=\; \mathop{\mathbf{E}}_{x \sim X} \left(-\log_2 f_X(x)\right) \;=\; -\int_{\Omega_X} f_X(x) \, \log_2 f_X(x) \, dx.$$

The average amount of information that remains in $X$ after the randomized attribute $Z$ is disclosed can be measured by the conditional differential entropy

$$\begin{aligned} h(X|Z) \;&=\; \mathop{\mathbf{E}}_{(x,z) \sim (X,Z)} \left(-\log_2 f_{X|Z=z}(x)\right) \\ &=\; -\int_{\Omega_{X,Z}} f_{X,Z}(x,z) \, \log_2 f_{X|Z=z}(x) \, dx \, dz. \end{aligned}$$

The average information *loss* for $X$ that occurs by disclosing $Z$ can be measured in terms of the difference between the two entropies:

$$I(X;Z) = h(X) - h(X|Z) = \mathop{\mathbf{E}}_{(x,z)\sim(X,Z)} \log_2 \frac{f_{X|Z=z}(x)}{f_X(x)}.$$

This quantity is also known as *mutual information* between random variables $X$ and $Z$. It is proposed in [1] to use the following functions to measure amount of privacy ($\Pi(X)$) and amount of privacy loss ($\mathcal{P}(X|Z)$):

$$\Pi(X) := 2^{h(X)}; \quad \mathcal{P}(X|Z) := 1 - 2^{-I(X;Z)}.$$

In the example above (see (2)) we have

$$\Pi(X) = 2; \;\; \Pi(X|Z) = 2^{h(X|Z)} \approx 0.84; \;\; \mathcal{P}(X|Z) \approx 0.58.$$

A possible interpretation of these numbers is that, without knowing $Z$, we can localize $X$ within a set of size 2; when $Z$ is revealed, we can (on average) localize $X$ within a set of size 0.84, which is less than 1.

However, even this information-theoretic measure of privacy is not without some difficulties. To see why, we have to turn to the notion of privacy breaches from [11]. In the example above, suppose that clients would not like to disclose the property "$X \leqslant 0.01$." The prior probability of this property is 0.5%; however, if the randomized value $Z$ happens to be in $[-1, -0.99]$, the posterior probability $\mathbf{P}\,[X \leqslant 0.01 \mid Z = z]$ becomes 100%. Of course, $Z \in [-1, -0.99]$ is unlikely:

$$\begin{aligned}\mathbf{P}\,[-1 \leqslant Z \leqslant -0.99] &= \int_{-1}^{-0.99} dz \int_{-\infty}^{+\infty} f_X(x)\, f_Y(z-x)\, dx \\ &= \int_{-1}^{-0.99} dz \int_{0}^{z+1} 0.5 \cdot 0.5\, dx \;=\; 0.0000125\end{aligned}$$

Therefore, $Z \in [-1, -0.99]$ occurs for about 1 in 100,000 records. But every time it occurs the property "$X \leqslant 0.01$" is *fully disclosed*, becomes 100% certain. The mutual information, being an *average* measure, does not notice this rare disclosure. Nor does it alert us to the fact that whether $X \in [0, 1]$ or $X \in [4, 5]$ is fully disclosed for *every* record; this time it is because the prior probability of each of these properties is high (50%).

The notion of privacy breaches, on the other hand, captures these disclosures. Indeed, for any privacy breach level $\rho < 100\%$ and for some randomization outcome (namely, for $Z \leqslant -0.99$) the posterior probability of property "$X \leqslant 0.01$" is above the breach level. The problem with the definition of privacy breaches from [11] is that we have to specify which properties are privacy-sensitive, whose probabilities must be kept below breach level. Specifying too many privacy-sensitive properties may require too destructive a randomization, leading to a very imprecise aggregate model at the server. Thus, the question of the right privacy measure is still open.

A completely different approach to measuring privacy is suggested in [16]. This paper measures private information in terms of its monetary value, as a form of intellectual property. The cost of each piece of information must be determined in a "fair" way, so as to reflect the contribution of this piece in the overall profit. Two notions of fairness are analysed in [16], both coming from the theory of coalitional games: the *core* and the *Shapley value*. Let $S$ be the set of potential participants in a business, and for every subset $S' \subseteq S$ we know the payoff $v(S')$ that occurs if only the participants in $S'$ actually cooperate. Then the core consists of all possible ways to divide the total payoff $v(S)$ between all participants so that, for all $S' \subseteq S$, the share given to $S'$ is at least $v(S')$. In other words, the core contains all "fair" ways of dividing the total payoff, in the sense that no group of participants has an incentive to secede. The Shapley value is a way to divide the payoff so that each participating agent is awarded an amount equal to the average contribution of this agent to the payoff of the group at the time of his or her arrival, where the average is taken over all arrival orders of the agents. In our case, participating clients disclose certain private information to the server and then benefit from the data model built by the server. The paper analyses the core and the Shapley value for several simplified scenarios involving private information.

# 6. CONCLUDING REMARKS

The research in using randomization for preserving privacy has shown promise and has already led to interesting and practically useful results. It gives an impression of being a part of some deeper *statistical* approach to security and privacy, providing a connection to the groundbreaking work of Claude Shannon on secrecy systems [25], and allows us to look at privacy under a different angle than the conventional cryptographic approach [6; 21]. It raises an important question of measuring privacy, which should be addressed in the purely cryptographic setting as well since the disclosure through legitimate query answers must also be measured. Randomization does not rely on intractability hypotheses from algebra or number theory, and does not require costly cryptographic operations or sophisticated protocols. It is possible that future studies will combine statistical approach to privacy with cryptography and secure multiparty computation, to the mutual benefit of all of them.

# 7. REFERENCES


[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th Symposium on Principles of Database Systems*, Santa Barbara, California, USA, May 2001.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington DC, USA, May 26–28 1993.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, September 1994.

[4] R. Agrawal and R. Srikant. Privacy preserving data mining. In *Proceedings of the 19th ACM SIGMOD Conference on Management of Data*, Dallas, Texas, USA, May 2000.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. CRC Press, Boca Raton, Florida, USA, 1984.

[6] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2003.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society, Series B*, 39:1–38, 1977.

[8] G. T. Duncan, R. Krishnan, R. Padman, P. Reuther, and S. Roehrig. Cell suppression to limit content-based disclosure. In *Proceedings of the 30th Hawaii International Conference on System Sciences*, volume 3. IEEE Computer Society Press, 1997.

[9] G. T. Duncan and S. Mukherjee. Optimal disclosure limitation strategy in statistical databases: Deterring tracker attacks through additive noise. *Journal of the American Statistical Association*, 95(451):720–729, 2000.

[10] T. Evans, L. Zayatz, and J. Slanta. Using noise for disclosure limitation of establishment tabular data. *Journal of Official Statistics*, 14(4):537–551, 1998.

[11] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining*, pages 217–228, Edmonton, Alberta, Canada, July 23–26 2002.

[12] S. E. Fienberg, U. E. Makov, and R. J. Steele. Disclosure limitation using perturbation and related methods for categorical data. *Journal of Official Statistics*, 14(4):485–502, 1998.

[13] J. M. Gouweleeuw, P. Kooiman, L. C. R. J. Willenborg, and P.-P. de Wolf. Post randomisation for statistical disclosure control: Theory and implementation. *Journal of Official Statistics*, 14(4):463–478, 1998.

[14] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining*, pages 279–288, Edmonton, Alberta, Canada, July 23–26 2002.

[15] J. J. Kim and W. E. Winkler. Masking microdata files, 1995.

[16] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan. On the value of private information. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, 2001.

[17] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9(2):313–331, 1993.

[18] R. J. A. Little. Statistical analysis of masked data. *Journal of Official Statistics*, 9(2):407–426, 1993.

[19] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, November 1996. 304 pp.

[20] S. Mukherjee and G. T. Duncan. Disclosure limitation through additive noise data masking: Analysis of skewed sensitive data. In *Proceedings of the 30th Hawaii International Conference on System Sciences*, volume 3, pages 581–586. IEEE Computer Society Press, 1997.

[21] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2003.

[22] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.

[23] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, Seattle, Washington, USA, June 1–3 1998.

[24] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: $k$-anonymity and its enforcement through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, California, USA, May 1998.

[25] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28-4:656–715, 1949.

[26] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[27] L. Zayatz, R. Moore, and B. T. Evans. New directions in disclosure limitation at the census bureau. In *Proceedings of the Government Statistics Section, Joint Statistical Meetings*. American Statistical Association, 1996.