Privacy-Preserving Data Mining

Rakesh Agrawal **Ramakrishnan Srikant**
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

A fruitful direction for future data mining research will be the development of techniques that incorporate privacy concerns. Specifically, we address the following question. Since the primary task in data mining is the development of models about aggregated data, can we develop accurate models without access to precise information in individual data records? We consider the concrete case of building a decision-tree classifier from training data in which the values of individual records have been perturbed. The resulting data records look very different from the original records and the distribution of data values is also very different from the original distribution. While it is not possible to accurately estimate original values in individual data records, we propose a novel reconstruction procedure to accurately estimate the distribution of original data values. By using these reconstructed distributions, we are able to build classifiers whose accuracy is comparable to the accuracy of classifiers built with the original data.

1 Introduction

Explosive progress in networking, storage, and processor technologies has led to the creation of ultra large databases that record unprecedented amount of transactional information. In tandem with this dramatic increase in digital data, concerns about informational privacy have emerged globally [Tim97] [Eco99] [eu998] [Off98]. Privacy issues are further exacerbated now that the World Wide Web makes it easy for the new data to be automatically collected and added to databases [HE98] [Wes98a] [Wes98b] [Wes99] [CRA99a] [Cra99b]. The concerns over massive collection of data are naturally extending to analytic tools applied to data. Data mining, with its promise to efficiently discover valuable, non-obvious information from large databases, is particularly vulnerable to misuse [CM96] [The98] [Off98] [ECB99].

A fruitful direction for future research in data mining will be the development of techniques that incorporate privacy concerns [Agr99]. Specifically, we address the following question. Since the primary task in data mining is the development of models about aggregated data, can we develop accurate models without access to precise information in individual data records?

The underlying assumption is that a person will be willing to selectively divulge information in exchange of value such models can provide [Wes99]. Example of the value provided include filtering to weed out unwanted information, better search results with less effort, and automatic triggers [HS99]. A recent survey of web users [CRA99a] classified 17% of respondents as privacy fundamentalists who will not provide data to a web site even if privacy protection measures are in place. However, the concerns of 56% of respondents constituting the pragmatic majority were significantly reduced by the presence of privacy protection measures. The remaining 27% were marginally concerned and generally willing to provide data to web sites, although they often expressed a mild general concern about privacy. Another recent survey of web users [Wes99] found that 86% of respondents believe that participation in information-for-benefits programs is a matter of individual privacy choice. A resounding 82% said that having a privacy policy would matter; only 14% said that was not important as long as they got benefit. Furthermore, people are not equally protective of every field in their data records [Wes99] [CRA99a]. Specifically, a person

- may not divulge at all the values of certain fields;
- may not mind giving true values of certain fields;
- may be willing to give not true values but modified values of certain fields.

Given a population that satisfies the above assumptions, we address the concrete problem of building decision-tree classifiers [BFOS84] [Qui93] and show that that it is possible to develop accurate models while re-

specting users' privacy concerns. Classification is one the most used tasks in data mining. Decision-tree classifiers are relatively fast, yield comprehensible models, and obtain similar and sometimes better accuracy than other classification methods [MST94].

Related Work There has been extensive research in the area of statistical databases motivated by the desire to be able to provide statistical information (sum, count, average, maximum, minimum, pth percentile, etc.) without compromising sensitive information about individuals (see excellent surveys in [AW89] [Sho82].) The proposed techniques can be broadly classified into query restriction and data perturbation. The query restriction family includes restricting the size of query result (e.g. [Fel72] [DDS79]), controlling the overlap amongst successive queries (e.g. [DJL79]), keeping audit trail of all answered queries and constantly checking for possible compromise (e.g. [CO82]), suppression of data cells of small size (e.g. [Cox80]), and clustering entities into mutually exclusive atomic populations (e.g. [YC77]). The perturbation family includes swapping values between records (e.g. [Den82]), replacing the original database by a sample from the same distribution (e.g. [LST83] [LCL85] [Rei84]), adding noise to the values in the database (e.g. [TYW84] [War65]), adding noise to the results of a query (e.g. [Bec80]), and sampling the result of a query (e.g. [Den80]). There are negative results showing that the proposed techniques cannot satisfy the conflicting objectives of providing high quality statistics and at the same time prevent exact or partial disclosure of individual information [AW89]. The statistical quality is measured in terms of bias, precision, and consistency. Bias represents the difference between the unperturbed statistics and the expected value of its perturbed estimate. Precision refers to the variance of the estimators obtained by the users. Consistency represents the lack of contradictions and paradoxes. An exact disclosure occurs if by issuing one or more queries, a user is able to determine the exact value of a confidential attribute of an individual. A partial disclosure occurs if a user is able to obtain an estimator whose variance is below a given threshold.

While we share with the statistical database literature the goal of preventing disclosure of confidential information, obtaining high quality point estimates is not our goal. As we will see, it is sufficient for us to be able to reconstruct with sufficient accuracy the original distributions of the values of the confidential attributes. We adopt from the statistics literature two methods that a person may use in our system to modify the value of a field [CS76]:

- *Value-Class Membership.* Partition the values into a set of disjoint, mutually-exhaustive classes and return the class into which the true value x_i falls.
- *Value Distortion.* Return a value $x_i + r$ instead of x_i where r is a random value drawn from some distribution.

We discuss further these methods and the level of privacy they provide in the next section.

We do not use value dissociation, the third method proposed in [CS76]. In this method, a value returned for a field of a record is a true value, but from the same field in some other record. Interestingly, a recent proposal [ECB99] to construct perturbed training sets is based on this method. Our hesitation with this approach is that it is a global method and requires knowledge of values in other records.

The problem of reconstructing original distribution from a given distribution can be viewed in the general framework of inverse problems [EHN96]. In [FJS97], it was shown that for smooth enough distributions (e.g. slowly varying time signals), it is possible to to fully recover original distribution from non-overlapping, contiguous partial sums. Such partial sums of true values are not available to us. We cannot make a priori assumptions about the original distribution; we only know the distribution used in randomizing values of an attribute. There is rich query optimization literature on estimating attribute distributions from partial information [BDF+97]. In the OLAP literature, there is work on approximating queries on sub-cubes from higher-level aggregations (e.g. [BS97]). However, these works did not have to cope with information that has been intentionally distorted.

Closely related, but orthogonal to our work, is the extensive literature on access control and security (e.g. [Din78] [ST90] [Opp97] [RG98]). Whenever sensitive information is exchanged, it must be transmitted over a secure channel and stored securely. For the purposes of this paper, we assume that appropriate access controls and security procedures are in place and effective in preventing unauthorized access to the system. Other relevant work includes efforts to create tools and standards that provide platform for implementing a system such as ours (e.g. [Wor] [Ben99] [GWB97] [Cra99b] [AC99] [LM99] [LEW99]).

Paper Organization We discuss privacy-preserving methods in Section 2. We also introduce a quantitative measure to evaluate the amount of privacy offered by a method and evaluate the proposed methods against this measure. In Section 3, we present our reconstruction procedure for reconstructing the original data distribution given a perturbed distribution. We also present some empirical evidence of the efficacy of the reconstruction procedure. Section 4 describes techniques for building decision-tree classifiers from perturbed training data using our reconstruction procedure. We present an experimental evaluation of the

accuracy of these techniques in Section 5. We conclude with a summary and directions for future work in Section 6.

We only consider numeric attributes; in Section 6, we briefly describe how we propose to extend this work to include categorical attributes. We focus on attributes for which the users are willing to provide perturbed values. If there is an attribute for which users are not willing to provide even the perturbed value, we simply ignore the attribute. If only some users do not provide the value, the training data is treated as containing records with missing values for which effective techniques exist in the literature [BFOS84] [Qui93].

2 Privacy-Preserving Methods

Our basic approach to preserving privacy is to let users provide a modified value for sensitive attributes. The modified value may be generated using custom code, a browser plug-in, or extensions to products such as Microsoft's Passport (http://www.passport.com) or Novell's DigitalMe (http://www.digitalme.com). We consider two methods for modifying values [CS76]:

Value-Class Membership In this method, the values for an attribute are partitioned into a set of disjoint, mutually-exclusive classes. We consider the special case of **discretization** in which values for an attribute are discretized into intervals. All intervals need not be of equal width. For example, salary may be discretized into 10K intervals for lower values and 50K intervals for higher values. Instead of a true attribute value, the user provides the interval in which the value lies. Discretization is the method used most often for hiding individual values.

Value Distortion Return a value $x_i + r$ instead of x_i where r is a random value drawn from some distribution. We consider two random distributions:

- **Uniform**: The random variable has a uniform distribution, between $[-\alpha, +\alpha]$. The mean of the random variable is 0.
- **Gaussian**: The random variable has a normal distribution, with mean $\mu = 0$ and standard deviation σ [Fis63].

We fix the perturbation of an entity. Thus, it is not possible for snoopers to improve the estimates of the value of a field in a record by repeating queries [AW89].

2.1 Quantifying Privacy

For quantifying privacy provided by a method, we use a measure based on how closely the original values of a modified attribute can be estimated. If it can be estimated with c% confidence that a value x lies in the interval $[x_1, x_2]$, then the interval width $(x_2 - x_1)$ defines the amount of privacy at c% confidence level.

	Confidence		
	50%	95%	99.9%
Discretization	$0.5 \times W$	$0.95 \times W$	$0.999 \times W$
Uniform	$0.5 \times 2\alpha$	$0.95 \times 2\alpha$	$0.999 \times 2\alpha$
Gaussian	$1.34 \times \sigma$	$3.92 \times \sigma$	$6.8 \times \sigma$

Table 1: Privacy Metrics

Table 1 shows the privacy offered by the different methods using this metric. We have assumed that the intervals are of equal width W in Discretization.

Clearly, for $2\alpha = W$, Uniform and Discretization provide the same amount of privacy. As α increases, privacy also increases. To keep up with Uniform, Discretization will have to increase the interval width, and hence reduce the number of intervals. Note that we are interested in very high privacy. (We use 25%, 50%, 100% and 200% of range of values of an attribute in our experiments.) Hence Discretization will lead to poor model accuracy compared to Uniform since all the values in a interval are modified to the same value. Gaussian provides significantly more privacy at higher confidence levels compared to the other two methods. We, therefore, focus on the two value distortion methods in the rest of the paper.

3 Reconstructing The Original Distribution

For the concept of using value distortion to protect privacy to be useful, we need to be able to reconstruct the original data distribution from the randomized data. Note that we reconstruct distributions, not values in individual records.

We view the n original data values $x_1, x_2, \ldots, x_n$ of a one-dimensional distribution as realizations of n independent identically distributed (iid) random variables $X_1, X_2, \ldots, X_n$, each with the same distribution as the random variable X. To hide these data values, n independent random variables $Y_1, Y_2, \ldots, Y_n$ have been used, each with the same distribution as a different random variable Y. Given $x_1+y_1, x_2+y_2, \ldots, x_n+y_n$ (where y_i is the realization of Y_i) and the cumulative distribution function F_Y for Y, we would like to estimate the cumulative distribution function F_X for X.

Reconstruction Problem *Given a cumulative distribution F_Y and the realizations of n iid random samples $X_1+Y_1, X_2+Y_2, \ldots, X_n+Y_n$, estimate F_X.*

Let the value of $X_i + Y_i$ be $w_i (= x_i + y_i)$. Note

that we do not have the individual values x_i and y_i, only their sum. We can use Bayes' rule [Fis63] to estimate the posterior distribution function F'_{X_1} (given that $X_1+Y_1 = w_1$) for X_1, assuming we know the density functions f_X and f_Y for X and Y respectively.

$$
\begin{aligned}
& F'_{X_1}(a) \\
& \quad \equiv \int_{-\infty}^{a} f_{X_1}(z \mid X_1+Y_1 = w_1)\, dz \\
& \quad = \int_{-\infty}^{a} \frac{f_{X_1+Y_1}(w_1 \mid X_1 = z)\, f_{X_1}(z)}{f_{X_1+Y_1}(w_1)}\, dz \\
& \qquad \text{(using Bayes' rule for density functions)} \\
& \quad = \int_{-\infty}^{a} \frac{f_{X_1+Y_1}(w_1 \mid X_1 = z)\, f_{X_1}(z)}{\int_{-\infty}^{\infty} f_{X_1+Y_1}(w_1 \mid X_1 = z')\, f_{X_1}(z')\, dz'}\, dz \\
& \qquad \text{(expanding the denominator)} \\
& \quad = \frac{\int_{-\infty}^{a} f_{X_1+Y_1}(w_1 \mid X_1 = z)\, f_{X_1}(z)\, dz}{\int_{-\infty}^{\infty} f_{X_1+Y_1}(w_1 \mid X_1 = z)\, f_{X_1}(z)\, dz} \\
& \qquad \text{(inner integral is independent of outer)} \\
& \quad = \frac{\int_{-\infty}^{a} f_{Y_1}(w_1 - z)\, f_{X_1}(z)\, dz}{\int_{-\infty}^{\infty} f_{Y_1}(w_1 - z)\, f_{X_1}(z)\, dz} \\
& \qquad \text{(since } Y_1 \text{ is independent of } X_1\text{)} \\
& \quad = \frac{\int_{-\infty}^{a} f_{Y}(w_1 - z)\, f_{X}(z)\, dz}{\int_{-\infty}^{\infty} f_{Y}(w_1 - z)\, f_{X}(z)\, dz} \\
& \qquad \text{(since } f_{X_1} \equiv f_X \text{ and } f_{Y_1} \equiv f_Y\text{)}
\end{aligned}
$$

To estimate the posterior distribution function F'_X given $x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n$, we average the distribution functions for each of the X_i.

$$F'_X(a) = \frac{1}{n}\sum_{i=1}^{n} F'_{X_i} = \frac{1}{n}\sum_{i=1}^{n} \frac{\int_{-\infty}^{a} f_Y(w_i - z)\, f_X(z)\, dz}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f_X(z)\, dz}$$

The corresponding posterior density function, f'_X is obtained by differentiating F'_X:

$$f'_X(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(w_i - a)\, f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f_X(z)\, dz} \qquad (1)$$

Given a sufficiently large number of samples, we expect f'_X in the above equation to be very close to the real density function f_X. However, although we know f_Y,[1] we do not know f_X. Hence we use the uniform distribution as the initial estimate f^0_X, and iteratively refine this estimate by applying Equation 1. This algorithm is sketched out in Figure 1.

Using Partitioning to Speed Computation Assume a partitioning of the domain (of the data values) into intervals. We make two approximations:

[1] For example, if Y is the standard normal, $f_Y(z) = (1/\sqrt{(2\pi)})e^{-z^2/2}$.

(1) $f^0_X :=$ Uniform distribution
(2) $j := 0$ // Iteration number
`repeat`
(3) $\quad f^{j+1}_X(a) := \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(w_i - a)\, f^j_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f^j_X(z) dz}$
(4) $\quad j := j + 1$
`until` (stopping criterion met)

Figure 1: Reconstruction Algorithm

- We approximate the distance between z and w_i (or between a and w_i) with the distance between the mid-points of the intervals in which they lie, and
- We approximate the density function $f_X(a)$ with the average of the density function over the interval in which a lies.

Let $I(x)$ denote the interval in which x lies, $m(I_p)$ the mid-point of interval I_p, and $m(x)$ the mid-point of interval $I(x)$. Let $f_X(I_p)$ be the average value of the density function over the interval I_p, i.e. $f_X(I_p) = \int_{I_p} f_X(z)dz \;/\; \int_{I_p} dz$. By applying these two approximations to Equation 1, we get

$$f'_X(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(m(w_i) - m(a))\, f_X(I(a))}{\int_{-\infty}^{\infty} f_Y(m(w_i) - m(z))\, f_X(I(z))\, dz}$$

Let $I_p, p = 1 \ldots k$ denote the k intervals, and L_p the width of interval I_p. We can replace the integral in the denominator with a sum, since $m(z)$ and $f_X(I(z))$ do not change within an interval:

$$f'_X(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(m(w_i) - m(a))\, f_X(I(a))}{\sum_{t=1}^{k} f_Y(m(w_i) - m(I_t))\, f_X(I_t)\, L_t} \qquad (2)$$

We now compute the average value of the posterior density function over the interval I_p.

$$
\begin{aligned}
& f'_X(I_p) \\
& \quad = \int_{I_p} f'_X(z)dz \;/\; L_p \\
& \quad = \int_{I_p} \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(m(w_i) - m(z))\, f_X(I(z))\, dz}{\sum_{t=1}^{k} f_Y(m(w_i) - m(I_t))\, f_X(I_t)\, L_t} / L_p \\
& \qquad \text{(substituting Equation 2)} \\
& \quad = \int_{I_p} \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(m(w_i) - m(I_p))\, f_X(I_p)\, dz}{\sum_{t=1}^{k} f_Y(m(w_i) - m(I_t))\, f_X(I_t)\, L_t} / L_p \\
& \qquad \text{(since } I(z) = I_p \text{ within } I_p\text{)} \\
& \quad = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(m(w_i) - m(I_p))\, f_X(I_p)}{\sum_{t=1}^{k} f_Y(m(w_i) - m(I_t))\, f_X(I_t)\, L_t} \\
& \qquad \text{(since } \textstyle\int_{I_p} dz = L_p\text{)}
\end{aligned}
$$

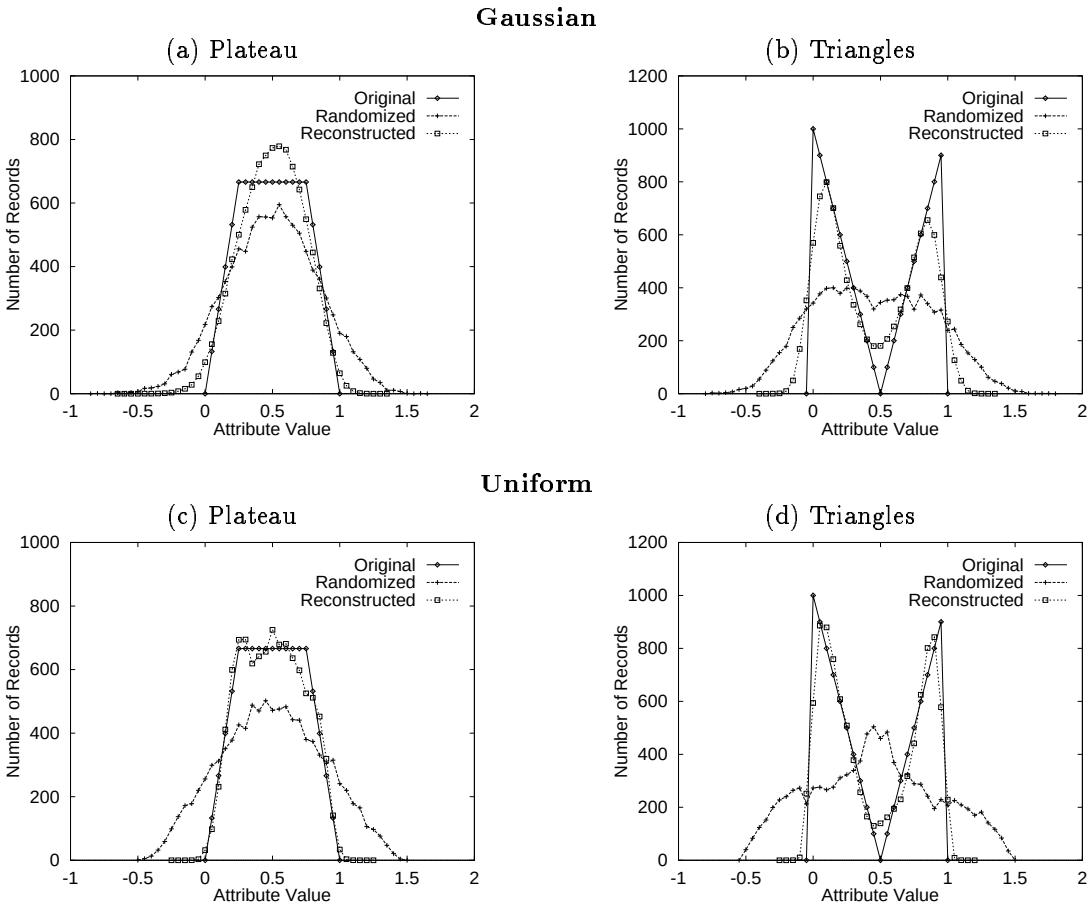


Figure 2: Reconstructing the Original Distribution

Let $N(I_p)$ be the number of points that lie in interval I_p (i.e. number of elements in the set $\{w_i | w_i \in I_p\}$. Since $m(w_i)$ is the same for points that lie within the same interval,

$$f'_X(I_p) = \frac{1}{n} \sum_{s=1}^{k} N(I_s) \times \frac{f_Y(m(I_s) - m(I_p))\, f_X(I_p)}{\sum_{t=1}^{k} f_Y(m(I_s) - m(I_t))\, f_X(I_t)\, L_t}$$

Finally, let $\Pr'(X \in I_p)$ be the posterior probability of X belonging to interval I_p, i.e. $\Pr'(X \in I_p) = f'_X(I_p) \times L_p$. Multiplying both sides of the above equation by L_p, and using $\Pr(X \in I_p) = f_X(I_p) \times L_p$, we get:

$$\Pr'(X \in I_p) = \frac{1}{n} \sum_{s=1}^{k} N(I_s) \times \frac{f_Y(m(I_s) - m(I_p))\, \Pr(X \in I_p)}{\sum_{t=1}^{k} f_Y(m(I_s) - m(I_t))\, \Pr(X \in I_t)} \quad (3)$$

We can now substitute Equation 3 in step 3 of the algorithm (Figure 1), and compute step 3 in $O(m^2)$ time.[2]

[2] A naive implementation of Equation 3 will lead to $O(m^3)$ time. However, since the denominator is independent of I_p, we can re-use the results of that computation to get $O(m^2)$ time.

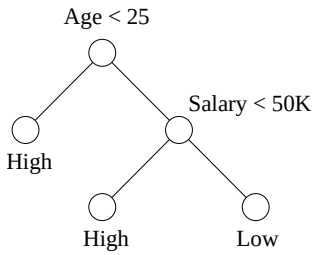
Stopping Criterion With omniscience, we would stop when the reconstructed distribution was statistically the same as the original distribution (using, say, the χ^2 goodness-of-fit test [Cra46]). An alternative is to compare the observed randomized distribution with the result of randomizing the current estimate of the original distribution, and stop when these two distributions are statistically the same. The intuition is that if these two distributions are close to each other, we expect our estimate of the original distribution to also be close to the real distribution. Unfortunately, we found empirically that the difference between the two randomized distributions is not a reliable indicator of the difference between the original and reconstructed distributions.

Instead, we compare successive estimates of the original distribution, and stop when the difference between successive estimates becomes very small (1% of the threshold of the χ^2 test in our implementation).

Empirical Evaluation Two original distributions, "plateau" and "triangles", are shown by the "Original" line in Figures 2(a) and (b) respectively. We add a Gaussian random variable with mean 0 and standard

rid	Age	Salary	Credit Risk
0	23	50K	High
1	17	30K	High
2	43	40K	High
3	68	50K	Low
4	32	70K	Low
5	20	20K	High

(a) Training Set



(b) Decision Tree

Figure 3: Credit Risk Example

deviation of 0.25 to each point in the distribution. Thus a point with value, say, 0.25 has a 95% chance of being mapped to a value between -0.26 and 0.74, and a 99.9% chance of being mapped to a value between -0.6 and 1.1. The effect of this randomization is shown by the "Randomized" line. We apply the algorithm (with partitioning) in Figure 1, with a partition width of 0.05. The results are shown by the "Reconstructed" line. Notice that we are able to pick out the original shape of the distribution even though the randomized version looks nothing like the original.

Figures 2(c) and (d) show that adding an uniform, discrete random variable between 0.5 and -0.5 to each point gives similar results.

4 Decision-Tree Classification over Randomized Data

4.1 Background

We begin with a brief review of decision tree classification, adapted from [MAR96] [SAM96]. A decision tree [BFOS84] [Qui93] is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from the same class. Each non-leaf node of the tree contains a *split point* which is a test on one or more attributes and determines how the data is partitioned. Figure 3(b) shows a sample decision-tree classifier based on the training shown in Figure 3a. $(Age < 25)$ and $(Salary < 50K)$ are two split points that partition the records into High and Low credit risk classes. The decision tree can be used to screen future applicants by classifying them into the *High* or *Low* risk categories.

A decision tree classifier is developed in two phases: a growth phase and a prune phase. In the growth phase, the tree is built by recursively partitioning the data until each partition contains members belonging to the same class. Once the tree has been fully grown, it is pruned in the second phase to generalize the tree by removing dependence on statistical noise or variation that may be particular only to the training data. Figure 4 shows the algorithm for the growth phase.

```
    Partition(Data S)
    begin
(1)     if (most points in S are of the same class) then
(2)         return;
(3)     for each attribute A do
(4)         evaluate splits on attribute A;
(5)     Use best split to partition S into S1 and S2;
(6)     Partition(S1);
(7)     Partition(S2);
    end

    Initial call: Partition(TrainingData)
```

Figure 4: The tree-growth phase

While growing the tree, the goal at each node is to determine the split point that "best" divides the training records belonging to that node. We use the *gini* index [BFOS84] to determine the goodness of a split. For a data set S containing examples from m classes, $gini(S) = 1 - \sum p_j^2$ where p_j is the relative frequency of class j in S. If a split divides S into two subsets S_1 and S_2, the index of the divided data $gini_{split}(S)$ is given by $gini_{split}(S) = \frac{n_1}{n} gini(S_1) + \frac{n_2}{n} gini(S_2)$. Note that calculating this index requires only the distribution of the class values in each of the partitions.

4.2 Training Using Randomized Data

To induce decision trees using perturbed training data, we need to modify two key operations in the tree-growth phase (Figure 4):

- Determining a split point (step 4).
- Partitioning the data (step 5).

We also need to resolve choices with respect to reconstructing original distribution:

- Should we do a global reconstruction using the whole data or should we first partition the data by class and reconstruct separately for each class?
- Should we do reconstruction once at the root node or do reconstruction at every node?

We discuss below each of these issues.

For pruning phase based on the Minimum Description Length principle [MAR96], no modification is needed.

Determining split points Since we partition the domain into intervals while reconstructing the distribution, the candidate split points are the interval boundaries. (In the standard algorithm, every mid-point between any two consecutive attribute values is a candidate split point.) For each candidate split-point, we use the statistics from the reconstructed distribution to compute gini index.

Partitioning the Data The reconstruction procedure gives us an estimate of the number of points in each interval. Let $I_1, ...I_m$ be the m intervals, and $N(I_p)$ be the estimated number of points in interval I_p. We associate each data value with an interval by sorting the values, and assigning the $N(I_1)$ lowest values to interval I_1, and so on.[3] If the split occurs at the boundary of interval I_{p-1} and I_p, then the points associated with intervals $I_1, \ldots, I_{p-1}$ go to S_1, and the points associated with intervals $I_p, \ldots, I_m$ go to S_2. We retain this association between points and intervals in case there is a split on the same attribute (at a different split-point) lower in the tree.

Reconstructing the Original Distribution We consider three different algorithms that differ in when and how distributions are reconstructed:

- **Global**: Reconstruct the distribution for each attribute once at the beginning using the complete perturbed training data. Induce decision tree using the reconstructed data.
- **ByClass**: For each attribute, first split the training data by class, then reconstruct the distributions separately for each class. Induce decision tree using the reconstructed data.
- **Local**: As in ByClass, for each attribute, split the training data by class and reconstruct distributions separately for each class. However, instead of doing reconstruction only once, reconstruction is done at each node (i.e. just before step 4 in Figure 4). To avoid over-fitting, reconstruction is stopped after the number of records belonging to a node become small.

A final detail regarding reconstruction concerns the number of intervals into which the domain of an attribute is partitioned. We use a heuristic to determine the number of intervals, m. We choose m such that there are an average of 100 points per interval. We then bound m to be between 10 and 100 intervals i.e. if $m < 10$, m is set to 10, etc.

Clearly, Local is the most expensive algorithm in terms of execution time. Global is the cheapest algorithm. ByClass falls in between. However, it is closer to Global than Local since reconstruction is done in ByClass only at the root node, whereas it is repeated at each node in Local. We empirically evaluate the classification accuracy characteristics of these algorithms in the next section.

[3] The interval associated with a data value should not be considered an estimator of the original value of that data value.

4.3 Deployment

In many applications, the goal of building a classification model is to develop an understanding of different classes in the target population. The techniques just described directly apply to such applications. In other applications, a classification model is used for predicting the class of a new object without a preassigned class label. For this prediction to be accurate, although we have been able to build an accurate model using randomized data, the application needs access to non-perturbed data which the user is not willing to disclose. The solution to this dilemma is to structure the application such that the classification model is shipped to the user and applied there. For instance, if the classification model is being used to filter information relevant to a user, the classifier may be first applied on the client side over the original data and the information to be presented is filtered using the results of classification.

5 Experimental Results

5.1 Methodology

We compare the classification accuracy of Global, ByClass, and Local algorithms against each other and with respect to the following benchmarks:

- **Original**, the result of inducing the classifier on unperturbed training data without randomization.
- **Randomized**, the result of inducing the classifier on perturbed data but without making any corrections for randomization.

Clearly, we want to come as close to Original in accuracy as possible. The accuracy gain over Randomized reflects the advantage of reconstruction.

We used the synthetic data generator from [AGI+92] for our experiments. We used a training set of 100,000 records and a test set of 5,000 records, equally split between the two classes. Table 2 describes the nine attributes, and Table 3 summarizes the five classification functions. These functions vary from having quite simple decision surface (Function 1) to complex non-linear surfaces (Functions 4 and 5). Functions 2 and 3 may look easy, but are quite difficult. The distribution of values on age are identical for both classes, unless the classifier first splits on salary. Further, the classifier has to exactly find five split-points on salary: 25, 50, 75, 100 and 125 to perfectly classify the data. The width of each of these intervals is less

	Group A	Group B
Function 1	$(\textbf{age} < 40) \vee ((60 \leq \textbf{age})$	otherwise
Function 2	$((\textbf{age} < 40) \wedge (50K \leq \textbf{salary} \leq 100K)) \vee$ $((40 \leq \textbf{age} < 60) \wedge (75K \leq \textbf{salary} \geq 125K)) \vee$ $((\textbf{age} \geq 60) \wedge (25K \leq \textbf{salary} \leq 75K))$	otherwise
Function 3	$((\textbf{age} < 40) \wedge (((\textbf{elevel} \in [0..1]) \wedge (25K \leq \textbf{salary} \leq 75K)) \vee$ $((\textbf{elevel} \in [2..3]) \wedge (50K \leq \textbf{salary} \leq 100K)))) \vee$ $((40 \leq \textbf{age} < 60) \wedge (((\textbf{elevel} \in [1..3]) \wedge (50K \leq \textbf{salary} \leq 100K)) \vee$ $(((\textbf{elevel} = 4)) \wedge (75K \leq \textbf{salary} \leq 125K)))) \vee$ $((\textbf{age} \geq 60) \wedge (((\textbf{elevel} \in [2..4]) \wedge (50K \leq \textbf{salary} \leq 100K)) \vee$ $((\textbf{elevel} = 1)) \wedge (25K \leq \textbf{salary} \leq 75K))))$	otherwise
Function 4	$(0.67 \times (\textbf{salary} + \textbf{commission}) - 0.2 \times \textbf{loan} - 10K) > 0$	otherwise
Function 5	$(0.67 \times (\textbf{salary} + \textbf{commission}) - 0.2 \times \textbf{loan} + 0.2 \times \textbf{equity} - 10K) > 0$ where $\textbf{equity} = 0.1 \times \textbf{hvalue} \times \max(\textbf{hyears} - 20, 0)$	otherwise

Table 3: Description of Functions

Attribute	Description
salary	uniformly distributed from 20K to 150K
commission	**salary** $\geq 75K \Rightarrow$ **commission** $= 0$ else uniformly distributed from 10K to 75K
age	uniformly distributed from 20 to 80
elevel	uniformly chosen from 0 to 4
car	uniformly chosen from 1 to 20
zipcode	uniformly chosen from 9 zipcodes
hvalue	uniformly distributed from $k \times 50K$ to $k \times 150K$, where $k \in \{0 \cdots 9\}$ depends on **zipcode**
hyears	uniformly distributed from 1 to 30
loan	uniformly distributed from 0 to 500K

Table 2: Attribute Descriptions

than 20% of the range of the attribute. Function 2 also contains embedded XORs which are known to be troublesome for decision tree classifiers.

Perturbed training data is generated using both Uniform and Gaussian methods (Section 2). All accuracy results involving randomization were averaged over 10 runs. We experimented with large values for the amount of desired privacy: ranging from 25% to 200% of the range of values of an attribute. The confidence threshold for the privacy level is taken to be 95% in all our experiments. Recall that if it can be estimated with 95% confidence that a value x lies in the interval $[x_1, x_2]$, then the interval width $(x_2 - x_1)$ defines the amount of privacy at 95% confidence level. For example, at 50% privacy, Salary cannot be estimated (with 95% confidence) any closer than an interval of width 65K, which is half the entire range for Salary. Similarly, at 100% privacy, Age cannot be estimated (with 95% confidence) any closer than an interval of width 60, which is the entire range for Age.

5.2 Comparing the Classification Algorithms

Figure 5 shows the accuracy of the algorithms for Uniform and Gaussian perturbations, for privacy levels of 25% and 100%. The x-axis shows the five functions from Table 3, and the y-axis the accuracy.

Overall, the ByClass and Local algorithms do remarkably well at 25% and 50% privacy, with accuracy numbers very close to those on the original data. Even at as high as 100% privacy, the algorithms are within 5% (absolute) of the original accuracy for Functions 1, 4 and 5 and within 15% for Functions 2 and 3. The advantage of reconstruction can be seen from these graphs by comparing the accuracy of these algorithms with Randomized.

Overall, the Global algorithm performs worse than ByClass and Local algorithms. The deficiency of Global is that it uses the same merged distribution for all the classes during reconstruction of the original distribution. It fares well on Functions 4 and 5, but the performance of even Randomized is quite close to Original on these functions. These functions have a diagonal decision surface, with equal number of points on each side of the diagonal surface. Hence addition of noise does not significantly affect the ability of the classifier to approximate this surface by hyper-rectangles.

As we stated in the beginning of this section, though they might look easy, Functions 2 and 3 are quite difficult. The classifier has to find five split-points on salary and the width of each interval is 25K. Observe that the range over which the randomizing function spreads 95% of the values is more than 5 times the width of the splits at 100% privacy. Hence even small errors in reconstruction result in the split points being a little off, and accuracy drops.

The poor accuracy of Original for Function 2 at 25% privacy may appear anomalous. The explanation lies in

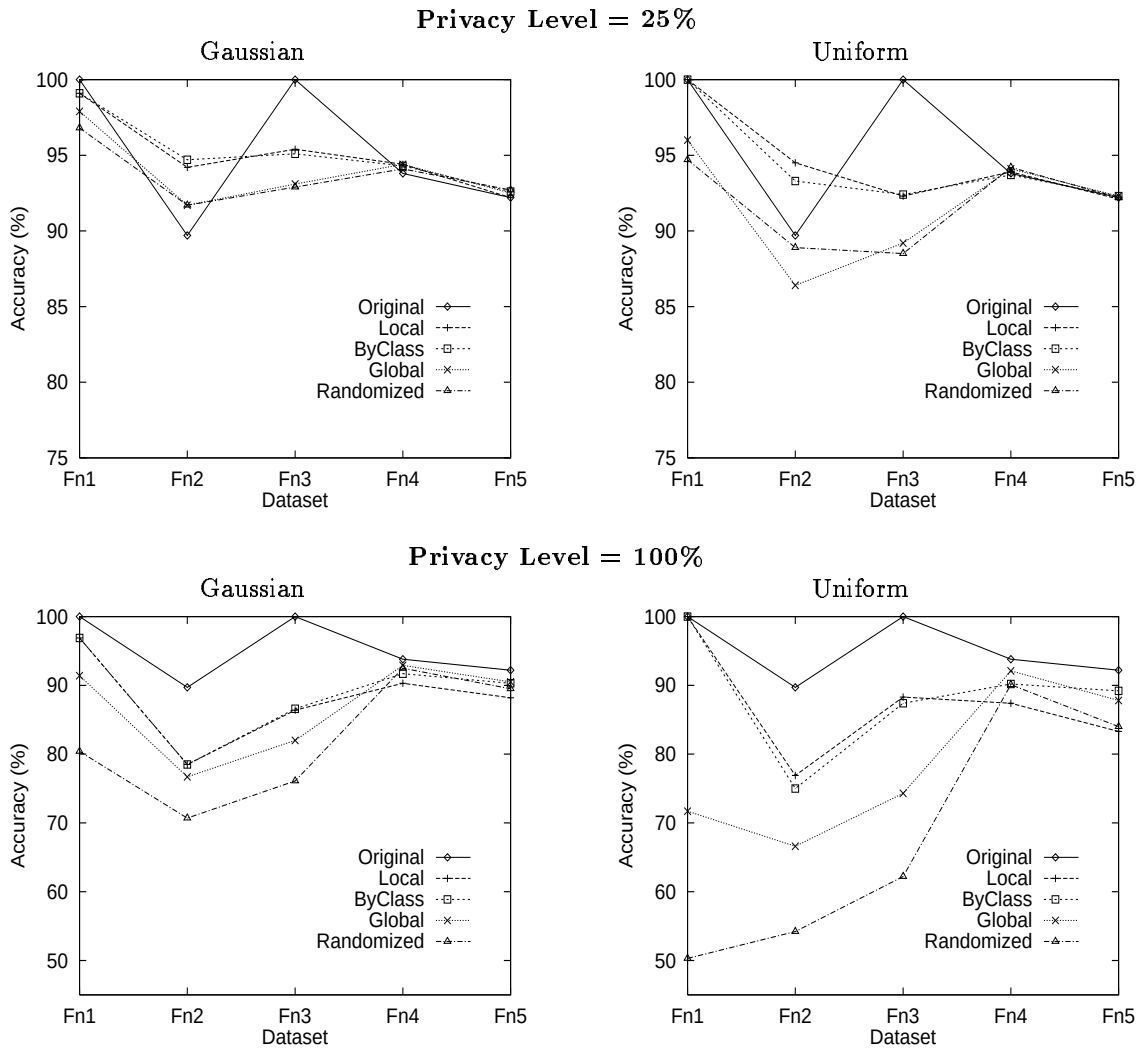


Figure 5: Classification Accuracy

there being a buried XOR in Function 2. When Original reaches the corresponding node, it stops because it does not find any split point that increases gini. However, due to the perturbation of data with randomization, the other algorithms find a "false" split point and proceed further to find the real split.

5.3 Varying Privacy

Figure 6 shows the effect of varying the amount of privacy for the ByClass algorithm. (We omitted the graph for Function 4 since the results were almost identical to those for Function 5.) Similar results were obtained for the Local algorithm. The x-axis shows the privacy level, ranging from 10% to 200%, and the y-axis the accuracy of the algorithms. The legend ByClass(G) refers to ByClass with Gaussian, Random(U) refers to Randomized with Uniform, etc.

Two important conclusions can be drawn from these graphs:

- Although Uniform perturbation of original data results in a much large degradation of accuracy before correction compared to Gaussian, the effect of both distributions is quite comparable after correction.
- The accuracy of the classifier developed using perturbed data, although not identical, comes fairly close to Original (i.e. accuracy obtained from using unperturbed data).

6 Conclusions and Future Work

In this paper, we studied the technical feasibility of realizing privacy-preserving data mining. The basic premise was that the sensitive values in a user's record will be perturbed using a randomizing function so that they cannot be estimated with sufficient precision. Randomization can be done using Gaussian or Uniform perturbations. The question we addressed was whether, given a large number of users who do this perturbation,

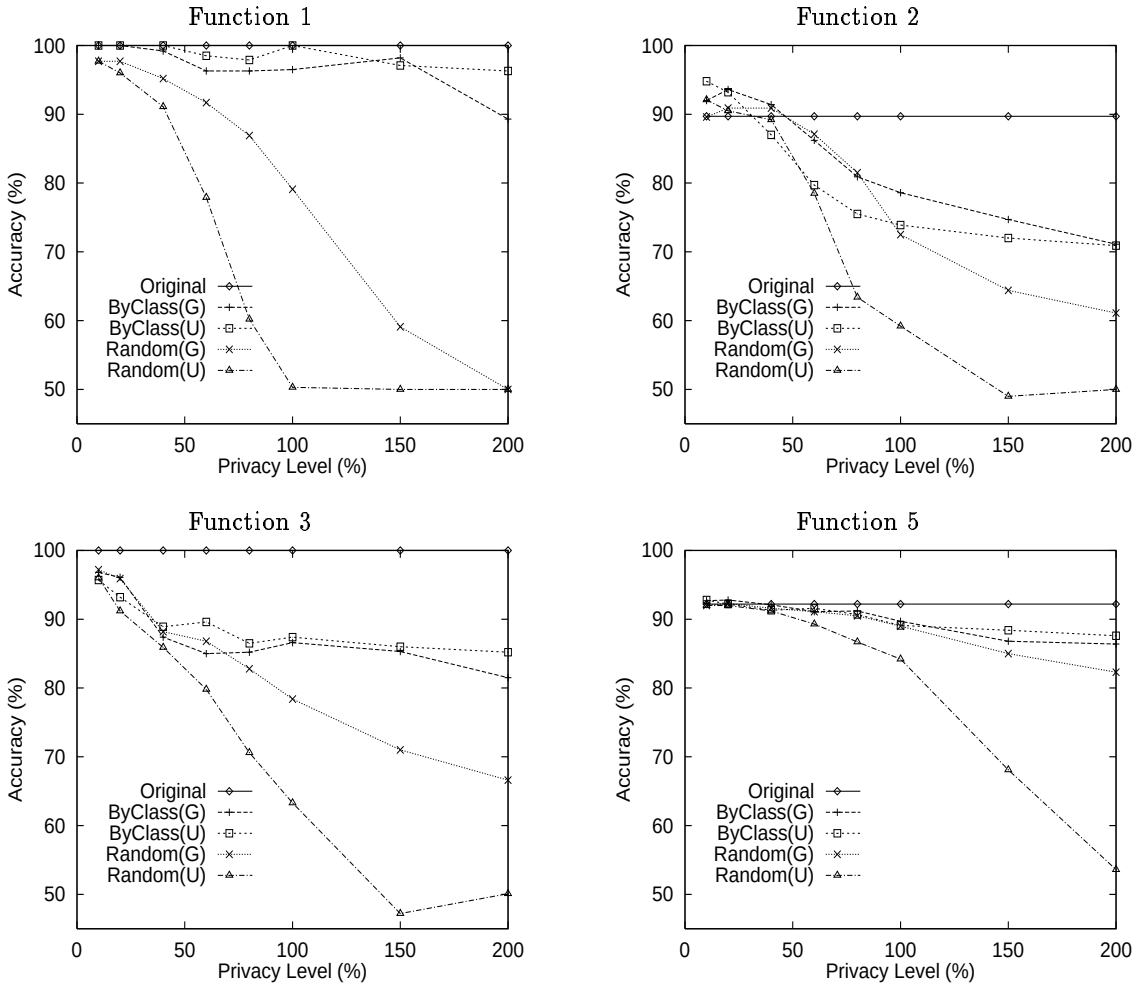


Figure 6: Change in Accuracy with Privacy

can we still construct sufficiently accurate predictive models.

For the specific case of decision-tree classification, we found two effective algorithms, ByClass and Local. The algorithms rely on a Bayesian procedure for correcting perturbed distributions. We emphasize that we reconstruct distributions, not individual records, thus preserving privacy of individual records. As a matter of fact, if the user perturbs a sensitive value once and always return the same perturbed value, the estimate of the true value cannot be improved by successive queries. We found in our empirical evaluation that:

- ByClass and Local are both effective in correcting for the effects of perturbation. At 25% and 50% privacy levels, the accuracy numbers are close to those on the original data. Even at 100% privacy, the algorithms were within 5% to 15% (absolute) of the original accuracy. Recall that if privacy were to be measured with 95% confidence, 100% privacy means that the true value cannot be estimated any closer than an interval of width which is the entire range for the corresponding attribute. We believe that a small drop in accuracy is a desirable trade-off for privacy in many situations.
- Local performed marginally better than ByClass, but required considerably more computation. Investigation of what characteristics might make Local a winner over ByClass (if at all) is an open problem.
- For the same privacy level, Uniform perturbation did significantly worse than Gaussian before correcting for randomization, but only slightly worse after correcting for randomization. Hence the choice between applying the Uniform or Gaussian distributions to preserve privacy should be based on other considerations: Gaussian provides more privacy at higher confidence thresholds, but Uniform may be easier to explain to users.

Future Work We plan to investigate the effectiveness of randomization with reconstruction for categorical attributes. The basic idea is to randomize each categorical value as follows: retain the value with probability p, and choose one of the other values at random with probability $1-p$. We may then derive an equation similar to Equation 1, and iteratively reconstruct the original distribution of values. Alternately, we may be able to extend the analytical approach presented in [War65] for boolean attributes to derive an equation that directly gives estimates of the original distribution.

Acknowledgments A hallway conversation with Robert Morris provided initial impetus for this work. Peter Haas diligently checked the soundness of the reconstruction procedure.

References

[AC99] M.S. Ackerman and L. Cranor. Privacy critics: UI components to safeguard users' privacy. In *ACM Conf. Human Factors in Computing Systems (CHI'99)*, 1999.

[AGI+92] Rakesh Agrawal, Sakti Ghosh, Tomasz Imielinski, Bala Iyer, and Arun Swami. An interval classifier for database mining applications. In *Proc. of the VLDB Conference*, pages 560–573, Vancouver, British Columbia, Canada, August 1992.

[Agr99] Rakesh Agrawal. Data Mining: Crossing the Chasm. In *5th Int'l Conference on Knowledge Discovery in Databases and Data Mining*, San Diego, California, August 1999. Available from `http://www.almaden.ibm.com/cs/quest/papers/kdd99_chasm.ppt`.

[AW89] Nabil R. Adam and John C. Wortman. Security-control methods for statistical databases. *ACM Computing Surveys*, 21(4):515–556, Dec. 1989.

[BDF+97] D. Barbara, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. Ioannidis, H. V. Jagadish, T. Johnson, R.Ng, V. Poosala, and K. Sevcik. The New Jersey Data Reduction Report. *Data Engrg. Bull.*, 20:3–45, Dec. 1997.

[Bec80] Leland L. Beck. A security mechanism for statistical databases. *ACM TODS*, 5(3):316–338, September 1980.

[Ben99] Paola Benassi. Truste: an online privacy seal program. *Comm. ACM*, 42(2):56–59, Feb. 1999.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[BS97] D. Barbara and M. Sullivan. Quasi cubes: Exploiting approximations in multidimensional databases. *SIGMOD Record*, 26(3):12–17, 1997.

[CM96] C. Clifton and D. Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19, May 1996.

[CO82] F.Y. Chin and G. Ozsoyoglu. Auditing and infrence control in statistical databases. *IEEE Trans. Softw. Eng.*, SE-8(6):113–139, April 1982.

[Cox80] L.H. Cox. Suppression methodology and statistical disclosure control. *J. Am. Stat. Assoc.*, 75(370):377–395, April 1980.

[Cra46] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[CRA99a] L.F. Cranor, J. Reagle, and M.S. Ackerman. Beyond concern: Understanding net users' attitudes about online privacy. Technical Report TR 99.4.3, AT&T Labs–Research, April 1999. Available from `http://www.research.att.com/library/trs/TRs/99/99.4/99.4.3/report.htm`.

[Cra99b] Lorrie Faith Cranor, editor. *Special Issue on Internet Privacy*. Comm. ACM, 42(2), Feb. 1999.

[CS76] R. Conway and D. Strip. Selective partial access to a database. In *Proc. ACM Annual Conf.*, pages 85–89, 1976.

[DDS79] D.E. Denning, P.J. Denning, and M.D. Schwartz. The tracker: A threat to statistical database security. *ACM TODS*, 4(1):76–96, March 1979.

[Den80] D.E. Denning. Secure statistical databases with random sample queries. *ACM TODS*, 5(3):291–315, Sept. 1980.

[Den82] D.E. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[Din78] C.T. Dinardo. *Computers and Security*. AFIPS Press, 1978.

[DJL79] D. Dobkin, A.K. Jones, and R.J. Lipton. Secure databases: Protection against user influence. *ACM TODS*, 4(1):97–106, March 1979.

[ECB99] V. Estivill-Castro and L. Brankovic. Data swapping: Balancing privacy against precision in mining for logic rules. In M. Mohania and A.M. Tjoa, editors, *Data Warehousing and Knowledge Discovery DaWaK-99*, pages 389–398. Springer-Verlag Lecture Notes in Computer Science 1676, 1999.

[Eco99] The Economist. *The End of Privacy*, May 1999.

[EHN96] H.W. Engl, M. Hanke, and A. Neubaue. *Regularization of Inverse Problems*. Kluwer, 1996.

[eu998] *The European Union's Directive on Privacy Protection*, October 1998. Available from `http://www.echo.lu/legal/en/dataprot/directiv/directiv.html`.

[Fel72] I.P. Fellegi. On the question of statistical confidentiality. *J. Am. Stat. Assoc.*, 67(337):7–18, March 1972.

[Fis63] Marek Fisz. *Probability Theory and Mathematical Statistics*. Wiley, 1963.

[FJS97] C. Faloutsos, H.V. Jagadish, and N.D. Sidiropoulos. Recovering information from summary data. In *Proc. of the 23rd Int'l Conference on Very Large Databases*, pages 36–45, Athens, Greece, 1997.

[GWB97] Ian Goldberg, David Wagner, and Eric Brewer. Privacy-enhancing technologies for the internet. In *IEEE COMPCON*, February 97.

[HE98] C. Hine and J. Eve. Privacy in the marketplace. *The Information Society*, 42(2):56–59, 1998.

[HS99] John Hagel and Marc Singer. *Net Worth*. Harvard Business School Press, 1999.

[LCL85] Chong K. Liew, Uinam J. Choi, and Chung J. Liew. A data distortion by probability distribution. *ACM TODS*, 10(3):395–411, 1985.

[LEW99] Tessa Lau, Oren Etzioni, and Daniel S. Weld. Privacy interfaces for information management. *Comm. ACM*, 42(10):89–94, October 1999.

[LM99] J.B. Lotspiech and R.J.T. Morris. Method and system for client/server communications with user information revealed as a function of willingness to reveal and whether the information is required. U.S. Patent No. 5913030, June 1999.

[LST83] E. Lefons, A. Silvestri, and F. Tangorra. An analytic approach to statistical databases. In *9th Int. Conf. Very Large Data Bases*, pages 260–274. Morgan Kaufmann, Oct-Nov 1983.

[MAR96] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT)*, Avignon, France, March 1996.

[MST94] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[Off98] Office of the Information and Privacy Commissioner, Ontario. *Data Mining: Staking a Claim on Your Privacy*, January 1998. Available from `http://www.ipc.on.ca/web_site.eng/matters/sum_pap/papers/datamine.htm`.

[Opp97] R. Oppliger. Internet security: Firewalls and beyond. *Comm. ACM*, 40(5):92–102, May 1997.

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[Rei84] Steven P. Reiss. Practical data-swapping: The first steps. *ACM TODS*, 9(1):20–37, 1984.

[RG98] A. Rubin and D. Greer. A survey of the world wide web security. *IEEE Computer*, 31(9):34–41, Sept. 1998.

[SAM96] John Shafer, Rakesh Agrawal, and Manish Mehta. SPRINT: A scalable parallel classifier for data mining. In *Proc. of the 22nd Int'l Conference on Very Large Databases*, Bombay, India, September 1996.

[Sho82] A. Shoshani. Statistical databases: Characteristics, problems and some solutions. In *Proceedings of the Eighth International Conference on Very Large Databases (VLDB)*, pages 208–213, Mexico City, Mexico, September 1982.

[ST90] P.D. Stachour and B.M. Thuraisingham. Design of LDV: A multilevel secure relational database management system. *IEEE Trans. Knowledge and Data Eng.*, 2(2):190–209, 1990.

[The98] Kurt Thearling. Data mining and privacy: A conflict in making. *DS**, March 1998.

[Tim97] Time. *The Death of Privacy*, August 1997.

[TYW84] J.F. Traub, Y. Yemini, and H. Woznaikowski. The statistical security of a statistical database. *ACM TODS*, 9(4):672–679, Dec. 1984.

[War65] S.L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.*, 60(309):63–69, March 1965.

[Wes98a] A.F. Westin. E-commerce and privacy: What net users want. Technical report, Louis Harris & Associates, June 1998. Available from `http://www.privacyexchange.org/iss/surveys/ecommsum.html`.

[Wes98b] A.F. Westin. Privacy concerns & consumer choice. Technical report, Louis Harris & Associates, Dec. 1998. Available from `http://www.privacyexchange.org/iss/surveys/1298toc.html`.

[Wes99] A.F. Westin. Freebies and privacy: What net users think. Technical report, Opinion Research Corporation, July 1999. Available from `http://www.privacyexchange.org/iss/surveys/sr990714.html`.

[Wor] The World Wide Web Consortium. *The Platform for Privacy Preference (P3P)*. Available from `http://www.w3.org/P3P/P3FAQ.html`.

[YC77] C.T. Yu and F.Y. Chin. A study on the protection of statistical databases. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 169–181, 1977.