Privacy-Sensitive Bayesian Network Parameter Learning

D. Meng and K. Sivakumar
School of EECS, Washington State University
Pullman, WA 99164-2752, USA
{dmeng, siva}@eecs.wsu.edu

H. Kargupta *
Department of CSEE, UMBC
Baltimore, MD 21250, USA
hillol@cs.umbc.edu

Abstract

This paper considers the problem of learning the parameters of a Bayesian Network, assuming the structure of the network is given, from a privacy-sensitive dataset that is distributed between multiple parties. This work belongs to the growing field of privacy-preserving data mining that deals with the problem of building accurate data mining models, while preserving the privacy of individual records. For a binary-valued dataset, we show that the count information required to estimate the conditional probabilities (model parameters) in a Bayesian network can be obtained as a solution to a set of linear equations involving some inner product between the relevant different feature vectors. Therefore, any privacy-sensitive method for computing inner product between vectors can be used to solve the Bayesian network parameter learning problem. We consider a random projection-based method that was proposed elsewhere to securely compute the inner product (with a modified implementation of that method).

1. Introduction

Advances in networking, storage, and computing technologies have resulted in an unprecedented increase in the amount of data that is collected and available to the public at large. If anything, this trend is expected to continue (and likley increase) for the forseeable future. This explosive growth in digital data has brought increased concerns about the privacy of personal information [1–3]. Analytic tools from data mining have the ability to efficiently discover valuable and non-trivial relationships and information present in the data, which exacerbates the privacy concerns [1, 13, 21, 39, 40]. Financial transactions, medical records, and network communication traffic are a few examples. Privacy is also an important issue in applications realted to counter-terrorism and homeland security. Some security related applications would require creating profiles, constructing social network models, and detecting terrorist communications from privacy-sensitive data. For example, mining healthcare data for the detection of bio-terrorism may require mining clinical records and pharmaceutical purchases of certain specific drugs. However, combining such diverse datasets belongling to different parties may violate privacy laws. Therefore, it is important to be able to extract desired data mining models from the data, without accessing the raw data in its original form.

Privacy-preserving data mining is an evolving area within the broad field of data mining that has emerged in response to these concerns [5, 12, 35]. In the following, we briefly review some of the important approaches proposed in the literature.

1.1. Related Work

There exists a growing body of literature on privacy-sensitive data mining. These algorithms can be divided into two broad groups: (a) approaches based on data perturbation or randomization and (b) approaches based on secure computation.

The first approach to privacy-sensitive data mining starts by first perturbing the data using randomized techniques. The perturbed data is then used to extract the patterns and models. The randomized value distortion technique for learning decision trees [5] and association rule learning [24] are examples of this approach. Evfimievski et al. [22, 23] and Rizvi [38] have also considered the approach in [5] in the context of association rule mining and suggest techniques for limiting privacy breaches.

Secure Multi-Party Computation (SMC) is the problem of evaluating a function of two or more parties' secret inputs, such that each party finally learns their

*Also affiliated with AGNIK, LLC, USA.

specified function output and nothing else is revealed, except what is implied by the party's own inputs and outputs. SMC problem was first introduced by Yao [45] and extended by Goldreich et al. [28]. These works use a similar methodology: the function f to be computed is represented as a boolean circuit, and then the parties run a protocol for every gate in the circuit. Every participant gets shares of the input wires and the output wires for every gate. Since determining which share goes to which party is done randomly, a party's own share tells it nothing. Upon completion, the parties exchange their shares, enabling each to compute the final result. The circuit evaluation protocol [27, 28], 1-out-of-k oblivious transfer [8], homomorphic encryption (secure sum), commutative encryption [7, 15], Yao's millionaire problem (secure comparison) [44] and some other cryptographical techniques serve as the building blocks of SMC. It has been proved in [27] that any functionality, which is expressed by an arithmetic circuit over GF(2) is privately computable. However, although appealing in their generality and simplicity, the traditional SMC is difficult to achieve efficiently. The complexity of the protocol depends on the size of the circuit, which depends on the size of the input. This is not practical for large database discovery, and calls for new ideas about algorithm and system design. Du and Atallah [18] have presented a collection of new secure multi-party computation applications such as privacy-preserving information retrieval [17], [9–11, 14, 17, 26, 31, 34], privacy-preserving statistical analysis [19], privacy-preserving intrusion detection, privacy-preserving geometric computation [6] etc. It also proposed a transformation framework that allows us to transform normal distributed computations to secure multi-party computation systematically. Agrawal [4] proposed a paradigm of information sharing across private databases based on cryptographic protocols. Clifton [12] has described several secure multi-party computation based algorithms that can support privacy-preserving data mining, e.g., secure sum, secure set union, secure size of set intersection and secure scalar product. This secure scalar computation has been directly applied in [42] for association rule mining from vertically partitioned data. Similar approaches to compute scalar product securely have been proposed elsewhere [17, 30, 36]. The Secure Multi-Party Computation idea has also been applied for association rule mining over horizontally partitioned data [32], distributed decision tree induction [20], K-Means clustering over vertically partitioned data [41], naive Bayes classification for horizontally partitioned data [33] and multivariate statistical analysis [16, 19]. Feigenbaum et al. have addressed the problem of computing approximations using SMC [25]. More recently, Wright and Yang [43] have proposed a privacy-preserving Bayesian Network structure learning algorithm.

1.2. Our Contribution

In this paper, we consider the problem of learning the parameters of a Bayesian Network (BN), assuming the structure of the network is given, from a privacy-sensitive dataset that is distributed between multiple parties. We make two important contributions towards that end. For a binary-valued dataset, we show that the count information required to estimate the conditional probabilities (model parameters) in a Bayesian network can be obtained as a solution to a set of linear equations involving some inner product between the relevant different feature vectors. Therefore, any privacy-sensitive method for computing inner product between vectors can be used to solve the Bayesian network parameter learning problem. Specifically, we consider a random projection-based method (to compute the inner product) that was proposed elsewhere, with a modified implementation of that method. This modified implementation requires considerably less exchange of perturbed data and produces probability estimates that are almost the same as that obtained using complete exchange of raw data. As such, the modified implementation of secure inner product computation can be used is several other privacy-sensitive data mining problems like clustering, correlation computation etc.

The rest of the paper is organized as follows. Section 2 provides a brief overview of Bayesian Networks (BN) followed by a description of the problem statement. In Section 3, we describe our proposed algorithm. A modified implementation of the inner product computation using random projections is described in Section 3.2. Experimental results are presented in Section 4. Finally, Section 5 provides some discussion and concluding remarks.

2. Problem Description

In this section, we first provide a brief overview of Bayesian Networks (BN). We then describe the privacy-sensitive BN parameter learning problem.

A BN is a probabilistic graph model, which is an important tool in data mining. It can be defined as a pair $(\mathcal{G}, p)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph (DAG). Here, $\mathcal{V}$ is the node set which represents variables in the problem domain and $\mathcal{E}$ is the edge set which denotes probabilistic relationships among the variables.

For a variable $X \in \mathcal{V}$, a parent of X is a node from which there exists a directed link to X. Figure 1 is a BN called the ASIA model. All the variables in this model are binary. Let $pa(X)$ denote the set of parents of X, then the conditional independence property can be used to factor the joint probability as follows:

$$P(\mathcal{V}) = \prod_{X \in \mathcal{V}} P(X \mid pa(X)).$$

The set of conditional distributions $\{P(X \mid pa(X)), X \in \mathcal{V}\}$ are called the parameters of a Bayesian network.

Learning a BN involves learning the structure of the network (the directed graph), and obtaining the conditional probabilities (parameters) associated with the network.

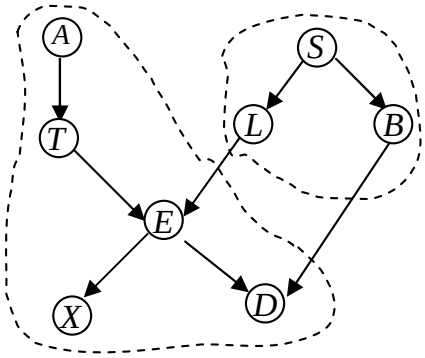

Figure 1. ASIA Model

We consider a set-up where the data corresponding to the different nodes are distributed among two or more parties. For example, in the ASIA model of Figure 1, party I contains observations for features (nodes) A, T, E, X, and D, whereas part II contains observations for features S, L, and B. We assume that there is a common "key" variable (perhaps timestamp, or spatial location, or serial no.) that is shared between the the sites, so that observations can be linked across sites. This is usually referred to vertical partitioning of the data or a heterogeneous data distribution. The dataset is privacy-sensitive in the sense that each party does not wish to share its raw data with the other parties. However, they wish to mine the entire combined dataset to obtain a global BN. We assume that the structure of the global BN is known to all the parties and focus on the problem of estimating the parameters (conditional probabilities) of the network. Our proposed solution to this problem is presented in the following section.

3. Algorithm

In the following, we assume that the features of the BN are binary, taking values in the set $\{-1, 1\}$. Extensions to multi-variate (discrete) case is conceptually similar, except for the algebra. In Section 3.1, we describe a system of linear equations, whose solution yields the desired conditional probabilities. The coefficient matrix for the linear equations can be obtained from the BN structure, which is assumed to be known. The inner product between certain feature vectors are needed to obtain the "right-hand-side vector" of the linear equations. Any secure inner product computation module can be used for this purpose. This is discussed in Section 3.2. Finally, Section 3.3 provides a privacy analysis of the proposed method.

3.1. Equations for BN parameter learning

In this section we build a set of linear equations whose solution yields all the conditional probabilities for a BN. We assume that all the data are binary with values 1 or -1 and the structure of BN is given.

For simplicity, first consider a node z with two parent nodes x and y. We need to obtain the values of all the conditional probabilities for z, given the values of nodes x and y. As shown in Table 1, there are eight $(2^3 = 8)$ different count values — $\{a, b, \ldots, h\}$ — to be determined. For example, b represents the number of observations with $x = -1$, $y = 1$ and $z = -1$. The corresponding probabilities can be obtained simply by normalizing the count values with respcet to the total number of observations N.

Let ${N_{ijk}}^{xyz}$ denote the number of obsevations for which $x = i$, $y = j$, and $z = k$, for $i, j, k \in \{-1, 1\}$. We then have

$$P(z = k \mid x = i, y = j) = \frac{{N_{ijk}}^{xyz}}{{N_{ij}}^{xy}}, \qquad (1)$$

$i, j, k \in \{-1, 1\}$, where ${N_{ij}}^{xy}$ denotes the number of obsevations for which $x = i$, and $y = j$.

Definition 3.1 *(Pseudo inner product) Given $n \geq 1$ vectors $x_1, x_2, \ldots, x_n$, each of dimension k, we define their pseudo-inner product (pip)*

$$pip(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{k} \prod_{i=1}^{n} x_{ij},$$

where $x_i = [x_{i1}, x_{i2}, \ldots, x_{ik}]$, $i = 1, 2, \ldots, n$ are the components of vector x_i. Note that for $n = 1$, we simply have

$$pip(x_1) = \sum_{j=1}^{k} x_{1j};$$

Table 1. Three-node example

	x, y			
	-1, -1	-1, 1	1, -1	1, 1
$z=-1$	a	b	c	d
$z=1$	e	f	g	h

i.e., $pip(x)$ is simply the sum of the components of vector x.

Let N be the total number of observations and X, Y, Z denote the data vector (column vector) for nodes x, y, z, respectively. Since there are three data vectors, we can compute $2^3 - 1 = 7$ different pseudo inner products. Observe that each pseudo inner product can be expressed uniquely by count variables $a, b, \ldots, h$. For example, $pip(Z)$ equals the sum of the entries in vector Z, which is precisely the number of observations with $z = 1$ minus the number of observations with $z = -1$. Indeed, we can write $(e+f+g+h) - (a+b+c+d) = pip(Z)$.

Similarly, we can verify that:

$$\begin{aligned}
-a-b-c-d+e+f+g+h &= pip(Z)\\
-a-b+c+d-e-f+g+h &= pip(X)\\
-a+b-c+d-e+f-g+h &= pip(Y)\\
a+b-c-d-e-f+g+h &= pip(Z,X)\\
a-b+c-d-e+f-g+h &= pip(Z,Y)\\
-a+b+c-d+e-f-g+h &= pip(Z,X,Y)\\
a-b-c+d+e-f-g+h &= pip(X,Y)
\end{aligned}$$

Another obvious condition is:

$$(e+f+g+h) + (a+b+c+d) = N$$

We can rewrite the above eight equations in matrix form as follows:

$$Ax = b, \tag{2}$$

where

$$A = \begin{pmatrix}
-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1\\
-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1\\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
-1 & 1 & 1 & -1 & 1 & -1 & -1 & 1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix},$$

$x = [a, b, c, d, e, f, g, h]^T$, and $b = [pip(Z),\ pip(X),\ pip(Y),\ pip(Z,X),\ pip(Z,Y),\ pip(Z,X,Y),\ pip(X,Y),\ N]^T$. It is easy to verify that matrix A is nonsingular. So we can solve the linear equations to get all the required conditional probabilities.

Table 2. A node with $m-1$ parent nodes

$x_1, x_2, \ldots, x_m$				
0	1	2	$\ldots$	$2^m - 1$
v_0	v_1	v_2	$\ldots$	v_{2^m-1}

This simple idea can be easily generalized to the case of arbitrary number of parent nodes.

Suppose a node x_1 has $m-1$ parent nodes $x_2, x_3, \ldots, x_m$, each variable taking values in the set $\{-1, 1\}$, as shown in Table 2. Clearly, there 2^m possible configurations for the variables $x_1, \ldots, x_m$, which correspond to the 2^m different conditional probability values. Let us use an integer index $i = 0, 1, \ldots, 2^m - 1$ to denote these 2^m configurations and let v_i denote the number of occurences of the configuration denoted by the index i in the dataset. For each subset of $\{x_1, \ldots, x_m\}$, we can compute the corresponding pseudo inner product, which would give one equation in the variables $v_0, \ldots, v_{2^m-1}$. Since the nodes $x_1, \ldots, x_m$ take values in $\{-1, 1\}$, the coefficient for each v_i in all those equations is either 1 or -1.

Let $V(m) = [v_0, v_1, \ldots, v_{2^m-1}]^T$ denote the vector of count variables. Given a subset $\{i_1, \ldots, i_k\}$ of $\{1, \ldots, m\}$, consider the pseudo inner product $\alpha_i = pip(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, where index $i = 0, 1, \ldots, 2^m - 1$ runs over the different subsets of $\{1, \ldots, m\}$. The equation corresonding to the pseudo inner product α_i can then be written as

$$Q^T(m,i)V(m) = \alpha_i = pip(x_{i_1}, x_{i_2}, \ldots, x_{i_k}),$$

$i = 0, 1, \ldots, 2^m - 1$, where $Q^T(m,i)$ is a $\{-1,1\}$-valued vector of dimension 2^m. We can then write

$$A(m)V(m) = \alpha, \tag{3}$$

where

$$A(m) = \begin{bmatrix} Q^T(m,0) \\ Q^T(m,1) \\ \vdots \\ Q^T(m, 2^m-1) \end{bmatrix}$$

and $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_{2^m-1}]^T$. In (3), m denotes the number of variables. In order to show that matrix $A(m)$ is nonsingular, we use induction on m. We have already shown that $A(3)$ is nonsingular and it is trivial to verify that $A(1)$, $A(2)$ are also nonsingular. In the following, we establish the fact that $A(m+1)$ is nonsingular, assuming that $A(m)$ is nonsingular.

Suppose we add an extra parent node x_{m+1} to node x_1. The ordering of the variables is as shown in Tables 2 and 3. Suppose $V(m+1) = [v_0, v_1, \ldots, v_{2^m-1}, v_{2^m}, \ldots, v_{2^{m+1}-1}]^T$ is the new vector of variables. As before, for $i = 0, 1, \ldots, 2^m - 1$,

Table 3. A node with m parent nodes

	$x_1, \ldots, x_m$				
	0	1	2	...	$2^m - 1$
$x_{m+1} = -1$	v_0	v_1	v_2	...	v_{2^m-1}
$x_{m+1} = 1$	v_{2^m}	v_{2^m+1}	v_{2^m+2}	...	$v_{2^{m+1}-1}$

let $\alpha_i = pip(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, where $\{i_1, \ldots, i_k\}$ is a subset of $\{1, 2, \ldots, m\}$. In other words, α_i denotes a pseudo inner product not involving variable x_{m+1}. Let $Q^T(m+1, i)V(m+1) = \alpha_i$, $i = 0, 1, \ldots, 2^m - 1$ denote the corresponding 2^m equations. It is then easy to see from Table 3 that

$$Q(m+1, i) = [Q(m, i) \quad Q(m, i)], i = 0, 1, \ldots, 2^m - 1. \tag{4}$$

Similarly, for $i = 0, 1, \ldots, 2^m - 1$, let $\beta_i = pip(x_{i_1}, x_{i_2}, \ldots, x_{i_k, x_{m+1}})$, where $\{i_1, \ldots, i_k\}$ is a subset of $\{1, 2, \ldots, m\}$. In other words, β_i denotes a pseudo inner product involving variable x_{m+1}. Let $Q^T(m+1, 2^m + i)V(m+1) = \beta_i$, $i = 0, 1, \ldots, 2^m - 1$, denote the corresponding 2^m equations. It is then easy to see from Table 3 that

$$Q(m+1, 2^m + i) = [-Q(m, i) \quad Q(m, i)], \tag{5}$$

$i = 0, 1, \ldots, 2^m - 1$.

We can now put the 2^m equations each corresponding to the α_i and β_i into a single set of 2^{m+1} equations as follows:

$$A(m+1)V(m+1) = [\alpha^T \quad \beta^T]^T,$$

where

$$A(m+1) = \begin{pmatrix} A(m) & A(m) \\ -A(m) & A(m) \end{pmatrix}$$

It is now clear that coefficient matrix $A(m+1)$ is nonsingular, provided $A(m)$ is nonsingular, which is true by the induction hypothesis.

3.2. Secure Inner Product Computation

From the previous subsection, we know if a BN structure is given, in other words the parent nodes of a node are given, the coefficient matrix A is uniquely determined. Therefore, if we can compute the pseudo inner products α_i in (3), the BN parameters can be obtained by solving the linear equations in (3). If the variables corresponding to the parent node(s) of a given node belong to a different party than the variable of the node itself, then computing the pseudo inner product would require exchange of raw data between the parties. Therefore, we need a privacy-sensitive method to compute inner products inorder to accomplish this step.

In general two types of methods are available for this purpose. The first one is based on a secure multi-party computation (SMC) scheme and the second one uses random projections or multiplicative noise. SMC based methods have been discussed by Du et al. [19, 20] and have the advantage that data privacy is guaranteed. However, practical implementation of these SMC protocals involve considerable amount of synchronization and communication between the parties. In our experiments, we used a random projection based method proposed in [37]. The important equations are reproduced below:

Let U be an $m \times n$ data matrix, with m observations and n features. Suppose R is an $m \times m$ orthogonal matrix; i.e., $R^T R = RR^T = I$. Consider the (multiplicatively) perturbed matrix

$$U_1 = RU.$$

Note that we use a single projection as opposed to the proposed double projection in [37]. It is easy to see that

$$U_1^T U_1 = (U^T R^T)(RU) = U^T (R^T R)U = U^T U. \tag{6}$$

There the inner products between the columns of U can be computed using the perturbed matrix U_1. So the owner of the data set U computes U_1 and hands over that to the other party (or a third party who does the data mining), who can then compute the required inner products required in the right-hand-side of (3). In practice, perturbation matrix R is chosen to be a random orthogonal matrix. This can be accomplished by starting with a random matrix with independent identically distributed (i.i.d.) entries W and orthogonalizing it. Standard orthogonalization techniques like Gram-Schmidt or QR decomposition can be used here [29].

3.3. Communication, Error, and Privacy Analysis

We now present a brief analysis of the communication cost and privacy of the proposed scheme.

First observe that those nodes, all of whose parents are in the same site, there is no privacy or communication problem and those parameters (conditional probabilities) can be locally estimated and communicated to the other parties. Since the communicating of model parameters is usually orders of magnitude less expensive than that of communicating raw data, we can ignore this cost.

Suppose, node i has $n_a - 1$ parents at the same site and n_b parents at a different site. Therefore, roughly

$2^{n_a}2^{n_b} = 2^{n_a+n_b}$ pseudo inner products have to be computed securely. This would require communication of $O(m2^{n_i})$ bits, where $n_i = n_a + n_b$ is one more than the number of parents of node i. Therefore, the total communication cost is $O(m\sum_i 2^{n_i})$. Note that in typical BN applications $n_i << n$.

The pseudo inner product computation is the only step that requires some exchange of data between the parties. Therefore, any privacy breach would have to occur in that step. Theorem 1 in [37] discusses the privacy-preserving properties of the random projection method. In particular, the $m \times m$ random orthogonal matrix R has $m(m-1)/2$ independent random entries (the rest of the $m(m+1)/2$ entries being determined by orthogonality constraints). As such, there are infinitely many solutions U, in general, to $U_1 = RU$, if R is unknown. By using a single random orthogonal matrix R in the projection instead of two random matrices R_1, R_2 as in [37], we do not have to "average" over results over multiple trials. Moreover, inner products computed using a single random orthogonal matrix R are virtually error-free as opposed to the case with double projection using random matrices, where the error goes to zero as the number of independent trials goes to infinity.

4. Experimental Results

In this section, we present results of our experiments with the proposed privacy-sensitive BN parameter learning for the ASIA model (see Figure 1). The conditional probability of a variable is a multi-dimensional array, where the dimensions are arranged in the same order as ordering of the variables, viz. $\{A, S, T, L, B, E, X, D\}$. Table 4 depicts the conditional probability of node E. It is laid out such that the first dimension toggles fastest. From Table 4, we can write the conditional probability of node E as a single vector as follows: $[0.9, 0.1, 0.1, 0.01, 0.1, 0.9, 0.9, 0.99]$. The true conditional probabilities (parameters) of the ASIA model are given in Table 5 following this ordering scheme. A data set with 2000 samples was generated from this ASIA model.

Table 4. The conditional probability of node E

No.	T	L	E	Probability
1	−1	−1	−1	0.9
2	1	−1	−1	0.1
3	−1	1	−1	0.1
4	1	1	−1	0.01
5	−1	−1	1	0.1
6	1	−1	1	0.9
7	−1	1	1	0.9
8	1	1	1	0.99

Table 5. All conditional probabilities for the ASIA model

A	0.9	0.1						
T	0.9	0.3	0.1	0.7				
S	0.5	0.5						
L	0.9	0.1	0.1	0.9				
E	0.9	0.1	0.1	0.01	0.1	0.9	0.9	0.99
X	0.95	0.02	0.05	0.98				
B	0.8	0.9	0.2	0.1				
D	0.9	0.2	0.8	0.9	0.1	0.8	0.2	0.1

4.1. Computing pseudo inner products using random orthogonal matrix

We generated a random matrix R_1 whose entries were i.i.d. Gaussian with zero mean and unit variance. This matrix was then orthogonalized using a QR decomposition to obtain a random orthogonal matrix R. The estimated parameters using our proposed algorithm in Section 3 are tabulated in Table 6. As expected, the estimated parameters are almost identical to the true values.

A	0.91	0.09						
T	0.91	0.31	0.09	0.69				
S	0.51	0.49						
L	0.91	0.09	0.1	0.9				
E	0.9	0.1	0.1	0.01	0.1	0.9	0.9	0.99
X	0.96	0.02	0.04	0.98				
B	0.8	0.91	0.2	0.09				
D	0.89	0.21	0.81	0.84	0.11	0.79	0.19	0.16

Table 6. Simulation results

We now compare the results of our single projection based inner product computation with that using a double projection [37].

4.2 Computing pseudo inner product using the (double) random matrix projection

In this case, we genrated two random matrices R_1, R_2 whose elements are i.i.d. Gaussian with zero mean and unit variance. We used the scheme proposed by Liu et al. in Figure 7 of [37]. In this case, since the pseudo inner products are not computed exactly, the solution to our linear equations in (3) may

produce values outside the range $[0, 1]$[1]. In this case, we simply rounded off the solution to either 0 or 1, as appropriate. Note that we need to run several independent trials (and average them) to compute the inner products.

Tables 7 and 8 are the estimated probabilities using 25 independent trials and 100 independent trials, respectively.

Table 7. The mean for 25 runs

A	0.9	0.1						
T	0.9	0.35	0.09	0.65				
S	0.53	0.48						
L	0.91	0.1	0.09	0.9				
E	0.56	0	0.46	0.27	0.44	1	0.54	0.73
X	0.95	0.03	0.05	0.97				
B	0.77	0.89	0.21	0.11				
D	0.52	0.47	0.8	0.83	0.48	0.53	0.21	0.17

Table 8. The mean for 100 runs

A	0.9	0.1						
T	0.9	0.35	0.09	0.65				
S	0.53	0.48						
L	0.91	0.1	0.09	0.9				
E	0.56	1	0.38	0.18	0.4	0	0.62	0.82
X	0.95	0.03	0.05	0.97				
B	0.77	0.89	0.21	0.11				
D	0.57	0.41	0	0.58	0.43	0.59	1	0.42

Clearly, the double projection method has large estimation errors, even after 100 independent runs.

5. Discussion and Conclusions

We considered the problem of learning the parameters of a Bayesian Network, assuming the structure of the network is given, from a privacy-sensitive dataset that is distributed between multiple parties. We considered the case of vertical (or heterogeneous) partitioning, where different parties hold values corresponding to a different subset of the variables. For a binary-valued dataset, we showed that the count information required to estimate the conditional probabilities (model parameters) of a Bayesian network can be obtained as a solution to a set of linear equations involving some inner product (more precisely, the pseudo inner product) between the relevant different feature vectors. As such, any privacy-sensitive method for computing inner product between vectors can be used to solve the Bayesian network parameter learning problem. In particular, inner product between vectors can be computed using secure protocols from the secure multi-party computation literature or using a random projection based approach. In our experiments, we considered a random projection-based method with a single projection using a random orthogonal matrix. This implementation requires considerably less exchange of perturbed data and produces almost error-free results as compared with that using double projection using random matrices.

[1]Note that the variables are all probabilities and hence values outside the range $[0, 1]$ are not meaningful.

Acknowledgements

The authors acknowledge supports from the United States National Science Foundation grants IIS-0329143 and IIS-0350533.

References

[1] Data mining: Staking claim on your privacy. Office of the Information and Privacy Commissioner, Ontario, January 1998.

[2] Directive on privacy protection. European Union, October 1998.

[3] The end of privacy. The Economist, May 1999.

[4] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *ACM International Conference on Management of Data*, San Diego, CA, June 2003.

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

[6] M. J. Atallah and W. Du. Secure multi-party computational geometry. In *WADS2001: Seventh International Workshop on Algorithms and Data Structures*, pages 165–179, Providence, Rhode Island, August 2001.

[7] J. C. Benaloh and M. D. Mare. One-way accumulators: A decentralized alternative to digital signatures. *Advances in Cryptology – EUROCRYPT'93. Workshop on the Theory and Application of Cryptographic Techniques. Lecture Notes in Computer Science.*, 765:274–285, May 1993.

[8] G. Brassard, C. Crépeau, and J. Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - Crypto86*, volume 234–238. Lecture Notes in Computer Science, 1987.

[9] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM Conferecne on Computer and Communications Security*, pages 120–127, Singapore, Novermber 1999.

[10] B. Chor and N. Gilboa. Computationally private information retrieval (extended abstract). In *Proceedings of the twenty-ninth annual ACM symposium on Theory of Computing*, El Paso, TX, May 1997.

[11] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, Milwaukee, Wi, October 1995.

[12] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for Privacy Preserving Distributed Data Mining. *ACM SIGKDD Explorations*, 4(2):28–34, 2003.

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

[14] G. Di-Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for database private information retrieval. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, 1998.

[15] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[16] W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In *17th Annual Computer Security Applications Conference*, New Orleans, Louisianna, 2001.

[17] W. Du and M. J. Atallah. Protocols for secure remote database access with approximate matching. In *7th ACM Conference on Computer and Communications Security(ACMCCS 2000). The first workshop on Security of Privacy in E-Commerce*, Athens, Greece, November 2000.

[18] W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In *New Security Paradigms Workshop*, pages 11–20, Cloudcroft, New Mexico, USA, September 11-13 2001.

[19] W. Du, Y. S. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of 2004 SIAM International Conference on Data Mining (SDM04)*, pages 222–233, Lake Buena Vista, FL, April 2004.

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

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

[22] A. Evfimevski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the ACM SIGMOD/PODS Conference*, pages 211–222, San Diego, CA, June 2003.

[23] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, July 2002.

[24] S. Evfimievski. Randomization techniques for privacy preserving association rule mining. *SIGKDD Explorations*, 4(2), December 2002.

[25] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. Wright. Secure multiparty computation of approximations. In *Proceedings of the 28th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 2076 of *Lecture Notes in Computer Science*, pages 927–938, Berlin, 2001. Springer.

[26] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval. In *The 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'98)*, 1998.

[27] O. Goldreich. *Secure Multi-Party Computation (Working Draft)*. Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, June 1998.

[28] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th annual ACM symposium on Theory of Computing*, pages 218–229, 1987.

[29] G. H. Golub and C. F. Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[30] M. A. Ioannis Ioannidis, Ananth Grama. A secure protocol for computing dot-products in clustered and distributed environments. pages 79–84, Vancouver, British Columbia, August 2002. Proceedings of International Conference on Parallel Processing (ICPP). CERIAS TR 2003-02.

[31] Y. Ishai and E. Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *Proceedings of the thirty-first annual ACM symposium on Theory of Computing*, Atlanta, GA, May 1999.

[32] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, June 2002.

[33] M. Kantarcoglu and J. Vaidya. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM Workshop on Privacy Preserving Data Mining*, pages 3–9, Melbourne, FL, November 2003.

[34] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings of the 38th annual IEEE computer society conference on Foundations of Computer Science*, Miami Beach, FL, October 1997.

[35] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in Cryptology - CRYPTO*, pages 36–54, 2000.

[36] K. Liu, H. Kargupta, and J. Ryan. Multiplicative noise, random projection, and privacy preserving data mining from distributed multi-party data. Technical

Report TR-CS-03-24, Computer Science and Electrical Engineering Department, University of Maryland, Baltimore County, 2003.

[37] K. Liu, H. Kargupta, and J. Ryan. Multiplicative noise, random projection, and privacy preserving data mining from distributed multi-party data. Technical report, UMBC, 2003.

[38] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[39] K. Thearling. Data mining and privacy: A conflict in the making. DS*, March 1998. `http://www.thearling.com/text/dsstar/privacy.htm`.

[40] B. Thuraisingham. Data mining, national security, privacy and civil liberties. *ACM SIGKDD Explorations Newsletter*, 4(2), 2002.

[41] J. Vaidya and C. Clifton. Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. In *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, D.C., August 2003.

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

[43] R. Wright and Z. Yang. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data. In *Proceedings of the tenth ACM SIGKDD Conference*, Seattle, WA, August 2004.

[44] A. C. Yao. Protocols for secure computation. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982.

[45] A. C. Yao. How to generate and exchange secrets. In *Proceedings 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.