Privacy-Preserving Bayesian Network Learning From Heterogeneous Distributed Data

Jianjie Ma and Krishnamoorthy Sivakumar
School of EECS, Washington State University, Pullman, WA 99164-2752, USA
Telephone: 1-509-335-4969, FAX: 1-509-335-3818, Email: {jma, siva}@eecs.wsu.edu

***Abstract*— In this paper, we propose a post randomization technique to learn a Bayesian network (BN) from distributed heterogeneous data, in a privacy sensitive fashion. In this case, two or more parties own sensitive data but want to learn a Bayesian network from the combined data. We consider both structure and parameter learning for the BN. The only required information from the data set is a set of sufficient statistics for learning both network structure and parameters. The proposed method estimates the sufficient statistics from the randomized data. The estimated sufficient statistics are then used to learn a BN. For structure learning, we face the familiar extra-link problem since estimation errors tend to break the conditional independence among the variables. We propose modifications of score functions used for BN learning, to solve this problem. We show both theoretically and experimentally that post randomization is an efficient, flexible, and easy-to-use method to learn Bayesian network from privacy sensitive data.**

***Index Terms*— Privacy Preserving Data Mining, Bayesian Network, Post Randomization**

I. Introduction

Privacy-preserving data mining deals with the problem of building accurate data mining models over aggregate data, while protecting privacy at the level of individual records. There are two main approaches to privacy preserving data mining. One approach is to perturb or randomize the data before sending it to the data miner. The perturbed or randomized data are then used to learn or mine the models and patterns [1]. Evfimieski *et al*. [5] proposed a select-a-size randomization technique for privacy-preserving mining of association rules. Du *et. al*. [4] suggested using randomized response techniques for privacy-preserving data mining and constructed decision trees from randomized data. Another approach is to use secure multiparty computation (SMC) to enable two or more parties to build data models without every party learning anything about the other party's data [9]. Though the SMC approach is appealing in its generality and simplicity, specific and efficient protocols have to be developed for data mining purpose since it is apparently inefficient for data mining applications [3]. All the current available techniques using SMC are based on a semi-honest model.

A. Related Work

Privacy-preserving Bayesian network learning is a more recent topic. Wright and Yang [15] discuss privacy-preserving Bayesian network structure computation on distributed heterogeneous data while Meng *et al*. [12] have considered the privacy-sensitive Bayesian Network parameter learning problem. The underlying method used in both works is to convert the computations required for BN learning into a series of inner product computations and then to use a secure inner product computation method proposed elsewhere. An SMC based method for inner product computation is used in [15] whereas [12] uses a method based on random projection proposed in [10]. The number of secure computation operations increases exponentially with the possible configurations of the problem variables. Wright *et al*. [15] proposed a privacy-preserving Bayesian network structure computation method. The accuracy of the learned structure using [15] is still not clear since their method is based on an approximated score function which might cause error links. Our experiments show that extra-link problem is severe even with small estimation errors. Existing literature on privacy preserving Bayesian network learning focuses on multiparty models. In addition to this model, our paper also considers a model where there is a data miner who actually does all the computations and learning for the participating parties. SMC based method has the following two drawbacks: (a) unrealistic semi-honest model assumption (b) large volumes of cooperative or synchronized computations among the parties involved. Most of the synchronized computations are the overheads due to privacy requirement. Other related works include [4], [5], [14], all of which consider the case where there is a data miner who does all the learning. In [4], [5], the focus is on association rules mining whereas [5] uses a select-a-size randomization. Rizi and Haritsa [14] proposed a randomization scheme called MASK which is based on a simple probabilistic distortion of user data. Post randomization gives a general framework for randomization of categorical data after data are collected. Select-a-size randomization can be considered as a special and intelligently designed post randomization. MASK [14] is a post randomization technique for binary variables. In [4], [5] the data is randomized by record, which actually implements randomization to all variables simultaneously. Randomizing all the variables simultaneously introduces unnecessary randomness. The proposed Post randomization method provides a more flexible way to cope with situations when different variables have different privacy requirements.

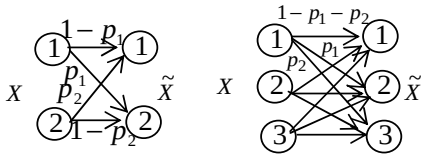


Fig. 1. (Left) Binary Randomization (Right) Ternary Symmetric Randomization

B. Our Contribution

This paper uses post randomization techniques to preserve privacy during Bayesian network learning. Gouweleeuw *et al*. [6] introduced post randomization for statistical databases for information disclosure control. Post randomization technique has proved to be effective in disclosure control. We explored the possibility of using post randomization in privacy-preserving data mining taking the privacy preserving Bayesian network learning as an example. We consider two Privacy-Preserving Bayesian network learning setups on distributed heterogeneous data, where different sets of variables are collected at the different sites. We develop estimators for frequency counters used in the learning of Bayesian network (both structure and parameters) and expressions for their covariance, based on the randomized data. Our experiments show that post randomization is an efficient, flexible and easy-to-use method to learn Bayesian network from privacy sensitive data. Using post randomization in privacy-preserving data mining overcomes the inherent drawbacks of SMC method and provides a reasonable privacy and accuracy. It is possible for a malicious party in SMC to get private information of other parties while he can only get randomized data if post randomization has been implemented to the data. Using Post randomization for privacy-preserving data mining provides a general framework for randomization of categorical data in privacy-preserving data mining.

II. Post Randomization and its Privacy Analysis

A. Post Randomization

Consider a data set D with a set of variables $X_1, X_2, \ldots, X_n$, where X_i takes discrete values from a set S_i whose cardinality is K_i. Post randomization for variable X_i is a (random) mapping $R_i : S_i \rightarrow S_i$, based on a set of transition probabilities $p^i_{lm} = p(\tilde{X}_i = k_m | X_i = k_l)$, where $k_m, k_l \in S_i$ and $\tilde{X}_i$ denotes the (randomized) variable value corresponding to variable X_i. The transition probability p^i_{lm} is the probability that a variable with original value k_l is randomized to the value k_m. Let $P^i = \{p^i_{lm}\}$ denote the $K_i \times K_i$ dimensional matrix that has p^i_{lm} as its (l, m)th entry. The randomized data set is $\tilde{D} = \{\tilde{y}_1, \ldots, \tilde{y}_N\}$, where $\tilde{y}_i$ is an instance of the randomized variables $\{\tilde{X}_1, \ldots, \tilde{X}_n\}$. For example, Binary Randomization can be used if the variable is binary. Ternary Symmetric Randomization is a choice if the variable is ternary. Binary and Ternary Symmetric Randomization are as shown in Fig. 1. We can apply the same randomization scheme independently to all of the variables—uniform randomization of the data set. Alternatively, we can use a non-uniform randomization, where different post randomization schemes are applied to different variables, independently. For example, we can choose different randomization parameters p_1 and p_2 to different binary variables for non-uniform randomization if the privacy requirement of the two variables are different. Non-uniform randomization is effective when different variables require different levels of privacy. The non-uniform randomization includes the special case when there is no privacy requirement for some of the variables. From the above, we can see that if variable X_i takes K_i values (or categories), the dimension of P^i will be $K_i \times K_i$. With larger K_i, more randomization is introduced into variable X_i in general. This is good from a privacy point of view. However, the variances of the estimator of frequency counters will also be larger for a given size of training samples. One solution to this problem is to partition the K_i categories of variable X_i into several groups such that a value in one group can only be randomized to a value in the same group. In this case, Matrix P^i becomes a block diagonal matrix. Post randomization can also be implemented on several variables simultaneously. For example, the variables X_i and X_j can be randomized simultaneously according to transition probability $p(\tilde{X}_i = l_1, \tilde{X}_j = l_2 \mid X_i = k_1, X_j = k_2)$. We can consider those variables randomized simultaneously as a combined variable when estimating the frequency counters. Proper simultaneous randomization can avoid the possible inconsistency of the randomized data set which independent randomization might cause.

B. Privacy Analysis of Post Randomization

We consider the notion of privacy introduced by Evfimievski *et al*. [5] in terms of an amplification factor γ. The γ-amplification in [5] is proposed in the framework where every data record should be randomized with a factor less than γ to limit the privacy breach before the data are sent to the data miner. However, in this paper we use the amplification γ purely as a worst-case quantification of privacy for a designed post randomization scheme. We shall first briefly review the notion of γ-amplification in our context of post randomization. A post randomization operator for variable X_i with transition probability P^i is at most γ-amplifying for $\tilde{X}_i = k$ if $\forall k_1, k_2 \ \frac{P^i(k_1,k)}{P^i(k_2,k)} \leq \gamma$, where $k, k_1, k_2 \in S_i$ and $|S_i| = K_i$. A post randomization operator is at most γ-amplifying for variable X_i if it is at most γ-amplifying for any $k \in S_i$. An upward ρ_1-to-ρ_2 privacy breach occurs when the posterior belief $p(X_i = k' \mid \tilde{X}_i = k) \geq \rho_2$, while the prior belief $p(X_i = k') \leq \rho_1$. A downward ρ_2-to-ρ_1 privacy breach occurs when the posterior belief $p(X_i \neq k' \mid \tilde{X}_i = k) \geq \rho_1$, while the prior belief $p(X_i \neq k') \leq 1 - \rho_2$. If the randomization operator is at most γ-amplifying for X_i, revealing $\tilde{X}_i$ will cause neither an upward ρ_1-to-ρ_2 privacy breach nor a downward ρ_2-to-ρ_1 privacy breach if $\frac{\rho_2(1-\rho_1)}{\rho_1(1-\rho_2)} > \gamma$. Clearly, smaller the value of γ, better is the worst case privacy. Ideally we would like to have $\gamma = 1$. Interested reader can refer to [5] for a detailed discussion about γ-amplification. For binary

symmetric randomization (Fig. 1), if $0 \leq p = p_1 = p_2 \leq 0.5$, then it is easy to see that it is at most γ-amplifying for $\gamma = \frac{1-p}{p}$. For the ternary symmetric randomization, if $1 - p_1 - p_2 \geq p_1 \geq p_2$, then amplification is at most $\gamma = \frac{1-p_1-p_2}{p_2}$. The at most γ-amplification provides a worst case quantification of privacy. For a given γ and prior belief ρ_1, we can get a ρ_2^* such that $\frac{\rho_2^*(1-\rho 1)}{\rho_1(1-\rho_2^*)} = \gamma$ and we will not have a privacy breach with posterior belief $\rho_2 > \rho_2^*$. However, the at most γ amplification does not provide any information of privacy preserved in general. Besides γ, we use $K^* = \min k\#\{k'|p(\tilde{X}_i = k|X_i = k') > 0\}$, minimum number of possible categories that can be randomized to category k for a designed post randomization, where minimum is taken over all categories of X_i. This K^* indicates the privacy preserved in general. It is similar to the K defined in K-anonymity in [13] but in a probabilistic sense.

III. Framework of Privacy-preserving BN Learning Using Post Randomization

The problem of Bayesian network learning is to find a network G and corresponding parameters that best matches the given training data set $D = \{y_1, y_2, \ldots, y_N\}$, where each record y_i is an instance of variables $\{X_1, X_2, \ldots, X_n\}$. Each party in our case observes a subset of the variables $\{X_1, X_2, \ldots, X_n\}$. We note that all the required information for the parameter and structure learning are sufficient statistics N_{ijk}s of each candidate structure (the fixed structure G only for parameter learning) from the training data D, where N_{ijk} is the number of records such that variable X_i is in its kth configuration and its parents $Pa(X_i)$ are in the jth configuration. Therefore, the problem of (privacy-sensitive) Bayesian network learning is equivalent to the problem of calculating the sufficient statistics (in a privacy-sensitive manner). In this paper, we estimate those sufficient statistics from the randomized data.

We consider the following two setups
(I) Several parties want to learn a global Bayesian network but are concerned about the privacy of their individual data. This corresponds to the Multiparty model of SMC.
(II) All parties send their randomized data to a data miner who does the learning.

A. Parameter Learning

For parameter learning, the structure G is assumed fixed and known to every party. For setup I, we used the definitions of cross variable and cross parent from [2]. Sensitive cross parents are the only variables that need to be randomized in setup I. Learning parameters for setup I above can be done as follows: For each party a_i,
(1) Randomize sensitive cross parents belonging to its own party according to their respective privacy requirements using post randomization as described in Section II. Randomizations are done independently for each (combined) variable and each record.
(2) Send randomized cross parents of party a_i for party a_j to party a_j together with the probability transition matrix used.
(3) Learn parameters for local variables of party a_i. This step does not involve randomized data.
(4) Estimate the sufficient statistics N_{ijk}s for each cross variable at same site a_i using the local data and randomized parent data from other parties.
(5) Estimate the parameters for cross variables using the estimated sufficient statistics.
(6) Share the parameters with all other parties.
In setup II: Every party randomizes all its sensitive variables according to their respective privacy requirements using post randomization (similar to randomization done to cross parents in setup I). Randomized data and their corresponding probability transition matrices are then sent to data miner. Data Miner then estimates the sufficient statistics N_{ijk}s and parameters for each node X_i using the randomized data.
The details of estimation of sufficient statistics N_{ijk} and parameter learning using estimated sufficient statistics are described in Sections IV and V respectively.

B. Structure Learning

During the search of a BN Structure (Directed Acyclic Graph — DAG) that best fit the data, perform randomization and estimation of sufficient statistics as described in Section III-A for each candidate structure as a fixed structure G. Use those estimated sufficient statistics to calculate the score of the candidate structure G. Then, we can choose a structure with maximum score. We use K-2 algorithm to search for a DAG that approximately has the maximum score. The details of structure learning from randomized data using K-2 algorithm are presented in Section VI.

IV. Estimation Of Sufficient Statistics From Randomized Data

From Section III, we can see that the problem of privacy-preserving Bayesian network learning can be decomposed into a series of estimation of N_{ijk}s for each node X_i and each candidate structure. The parents $Pa(X_i)$ of Node X_i are given by the (candidate) structure G.

Consider the following general case: The cardinality of Node X_i is K_i and it has Q parent nodes $Pa(X_i) = \{Pa_i(1), Pa_i(2), \ldots, Pa_i(Q)\}$ in the candidate structure. The cardinality of each parent $Pa_i(q)$ is $K_{Pa_i(q)}$. These variables can be arbitrarily vertically partitioned to different parties in both setups. The randomization of each (combined) variable can also be done by grouping the categories of the variable into groups. Our discussion below is independent of the specific partitioning and grouping of the variables.

We have the following different cases for estimating N_{ijk}s from the randomized data $\tilde{D}$ due to simultaneous randomization. Note that only the variables belonging to the same party can be randomized simultaneously in both of our setups.
(a) X_i and all of its parents are randomized independently.
(b) Some parents of X_i are randomized simultaneously.
(c) X_i is randomized simultaneously with some of its parents.
(d) X_i is randomized simultaneously with other variables (not its parent).
For cases (b) and (c), we can consider the simultaneously randomized variables as a combined variable. For example, if

node X_i is randomized simultaneously with one of its parents $Pa_i(1)$, N_{ijk} is equal to the number of records such that $(X_i; Pa_i(1)) = (k; j^1), Pa_i(2) = j^2, \ldots, Pa_i(Q) = j^Q$, where $(X_i; Pa_i(1))$ is a combined variable. Thus, we can estimate the N_{ijk}s from the randomized data by treating $(X_i; Pa_i(1))$ as a single variable with cardinality $|K_i| \times \left|K_{Pa_i(1)}\right|$. For case (d), since the current N_{ijk} does not involve the variable randomized simultaneously with X_i, we can get the marginal transition probability matrix from the given transition probability matrix, which is for the combined variable. Hence, without loss of generality, we can consider case (a) only.

Suppose the transition probability matrices of X_i and its parents are $P^i, P^{Pa_i(1)}, \ldots, P^{Pa_i(Q)}$, respectively. The problem here is to estimate the sufficient statistics N_{ijk} from the randomized data. We denote by $Pa(X_i)$ as a compound variable for all the parents of Node X_i. Hence $Pa(X_i)$ takes $J_i = \prod_{q=1}^{Q} K_{Pa_i(q)}$different values. The following are some notations used in the sequel: Superscript $\sim$ denotes the variables after randomization and superscript $\wedge$ denotes an estimate of the corresponding variable. N_{ijk} is as defined in Section III and $N_{ij} = \sum_{k=1}^{K_i} N_{ijk}$, where K_i is the cardinality of variable X_i. N_i is the J_iK_i dimensional vector of N_{ijk} values, that is $N_i = (N_{i11}, N_{i12}, \ldots, N_{i1K_i}, N_{i21}, \ldots, N_{iJ_iK_i})^t$, where superscript t denotes matrix transpose. $N_i(l)$ for $1 \leq l \leq J_iK_i$ is the number of records that $\{X_i, Pa(X_i)\}$ have the lth configuration, where $l = K_i(j-1) + k$ for some j and k such that X_i is in kth configuration while $Pa(X_i)$ in jth configuration. $\tilde{N}_{ijk}$, $\tilde{N}_{ij}$, and $\tilde{N}_i$ are defined similarly as N_{ijk}, N_{ij} and N_i but in the randomized data $\tilde{D}$. $\hat{N}_{ijk}$, $\hat{N}_{ij}$, and $\hat{N}_i$ are the estimations of N_{ijk}, N_{ij}, and N_i, respectively.

Given the training Data set D with N records, a candidate structure G and a randomization scheme characterized by probability transition matrices $P^i, P^{Pa_i(1)}, \ldots, P^{Pa_i(Q)}$, we have the following theorems.

Theorem 1.

(a) $E[\tilde{N}_i|D] = P^t N_i$, where $P = P^i \otimes P^{Pa_i}$ and $P^{Pa_i} = P^{Pa_i(1)} \otimes P^{Pa_i(2)} \ldots \otimes P^{Pa_i(Q)}$, $\otimes$ denotes Kronecker matrix product.

(b) Denote Y^i_{ml} as a binomial random variable that gives the number of records such that $\{\tilde{X}_i, \tilde{P}a(X_i)\}$ is in the lth configuration while $\{X_i, Pa(X_i)\}$ is in the mth configuration. We have $Y^i_{ml} \sim \mathcal{B}(N_i(m), \pi)$, where $\pi = P(m,l)$, the (m,l)th element of the probability matrix P defined in (a) and $\mathcal{B}$ denotes Binomial probability distribution. Moreover, $\text{Cov}\{Y^i_{ml_1}, Y^i_{nl_2}\} =$

$$\begin{cases} \text{Var}\{Y^i_{ml}\} = N_i(m)P(m,l_1)(1-P(m,l_1)) \\ \quad \text{if } n = m, l_1 = l_2 \\ -N_i(m)P(m,l_1)P(m,l_2) \quad \text{if } n = m, l_1 \neq l_2 \\ 0 \quad \text{if } n \neq m \end{cases}$$

(c) For $l = 1, 2, \ldots, J_iK_i$, $\tilde{N}_i(l) = \sum_{m=1}^{J_iK_i} Y^i_{ml}$. Moreover, $\text{Cov}\{\tilde{N}_i \mid D\} = \sum_{l=1}^{J_i} N_i(l)V_l$ where V_l is a $K_iJ_i \times K_iJ_i$ covariance matrix such that its (l_1, l_2)th element $V_l(l_1, l_2) =$

$$\begin{cases} P(l,l_1)(1-P(l,l_1)) & \text{if } l_1 = l_2, \\ -P(l,l_1)P(l,l_2) & \text{if } l_1 \neq l_2. \end{cases}$$

Proofs are omitted here due to the page limitations. Interested reader can refer to a longer version of this paper [11] for details.

The following theorem establishes the bias and variance of the estimator $\hat{N}_i = (P^t)^{-1}\tilde{N}_i$.

Theorem 2. $\hat{N}_i = (P^t)^{-1}\tilde{N}_i$ is an unbiased estimator for N_i and $\text{Cov}\{\hat{N}_i|D\} = (P^{-1})^t Cov\{\tilde{N}_i|D\}P^{-1}$, where P and $\text{Cov}\{\tilde{N}_i|D\}$ are given in Theorem 1.

A Binomial distribution $\mathcal{B}(n,p)$ can be approximated by a normal distribution $\mathcal{N}(np, np(1-p))$, when n is large. Since in Bayesian Network learning we usually have a relatively large sample size, the distribution of $\tilde{N}_i(l)$ can be well approximated by a normal distribution by Theorem 1(c). $\tilde{N}_i$ and $\hat{N}_i$ can be approximated by a J_iK_i dimensional joint normal random variable since by Theorem 2. In particular, $\hat{N}_i \sim \mathcal{N}(N_i, Cov\{\hat{N}_i \mid D\})$, where $\text{Cov}\{\hat{N}_i \mid D\}$ is given by Theorem 2.

V. Parameter Estimator and its Distribution

The Maximum Likelihood (ML) estimate of the BN parameter using the estimated sufficient statistics N_{ijk} is $\hat{\theta}^{ML}_{ijk} = \frac{\hat{N}_{ijk}}{\hat{N}_{ij}} = \frac{\hat{N}_{ijk}}{\sum_{k=1}^{K_i}\hat{N}_{ijk}}$ and the Maximum a posteriori (MAP) estimate of the parameter using the estimated sufficient statistics is $\hat{\theta}_{ijk} = \frac{\alpha_{ijk}+\hat{N}_{ijk}}{\alpha_ij+\hat{N}_{ij}}$, where $\alpha_{ijk}$is from the assumption of prior Dirichlet distribution for θ_{ij}, that is $P(\theta_{ij}|G) = Dirichlet(\alpha_{ij1}, \ldots, \alpha_{ijK_i})$. Here, we use the ML estimator and analyze its performance. Results for the MAP estimator can be obtained in a similar fashion.

The estimated parameter is a ratio of two dependent (well approximated) normal random variables with non-zero mean. The exact distribution of the ratio of two normal variables $W = \frac{X_1}{X_2}$ with $(X_1, X_2) \sim \mathcal{N}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$ for arbitrary mean, variance, and correlation is well-known [7]. However, the general distribution is quite complicated. An approximation to the exact distribution, when the probability $P(X_2 > 0) \to 1$ or when $\frac{\mu_2}{\sigma_2}$ is large is also given in [7]. Furthermore, $Z = \frac{\mu_2 w - \mu_1}{\sigma_1\sigma_2 a(w)}$ is approximately a standard normal distribution if $\frac{\mu_2}{\sigma_2}$ is large. A Taylor series expansion of Z around $\frac{\mu_1}{\mu_2}$ shows that $Z \approx \frac{\mu_2}{\sigma_1\sigma_2\sqrt{\frac{\mu_1^2}{\sigma_1^2\mu_2^2} - \frac{2\rho\mu_1}{\sigma_1\sigma_2\mu_2} + \frac{1}{\sigma_2^2}}}(w - \frac{\mu_1}{\mu_2})$. It follows that the distribution of W can be approximated by a normal distribution with mean $\frac{\mu_1}{\mu_2}$ and variance $\frac{\sigma_1^2\sigma_2^2}{\mu_2^2}(\frac{\mu_1^2}{\sigma_1^2\mu_2^2} - \frac{2\rho\mu_1}{\sigma_1\sigma_2\mu_2} + \frac{1}{\sigma_2^2})$. We have $\hat{\theta}^{ML}_{ijk} = \frac{\hat{N}_{ijk}}{\hat{N}_{ij}} = \frac{\hat{N}_{ijk}}{\sum_{k=1}^{K_i}\hat{N}_{ijk}}$. The jointly normal variable $(\hat{N}_{ijk}, \hat{N}_{ij})$ has (approximate) distribution $\mathcal{N}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$, where $\mu_1 = N_{ijk}$, $\mu_2 = N_{ij}$ and $\sigma_1^2, \sigma_2^2, \rho$ can be obtained from Theorem 2. From Theorems 1 and 2, we can see $\frac{\mu_2}{\sigma_2}$ is of the order of $\sqrt{N_{ij}}$ for a given data set and a transition probability matrix P. So $\frac{\mu_2}{\sigma_2}$ is large even for relatively small sample sizes. Hence, the approximation of the distribution of ratio of two normal random variables with the simpler form works well for us. On the other hand, the normal approximation using Taylor expansion is also feasible in our case since $0 \leq \frac{N_{ijk}}{N_{ij}} \leq 1$. Hence, for a given Data set D and a probability transition matrix P,

$\hat{\theta}_{ijk}^{ML} = \frac{\hat{N}_{ijk}}{\hat{N}_{ij}}$ can be approximated by a normal distribution with mean $\theta_{ijk} = \frac{N_{ijk}}{N_{ij}}$ and variance $\frac{\sigma_1^2\sigma_2^2}{N_{ijk}^2}(\frac{N_{ijk}^2}{\sigma_1^2 N_{ij}^2} - \frac{2\rho N_{ijk}}{\sigma_1\sigma_2 N_{ij}} + \frac{1}{\sigma_2^2}) = \frac{\sigma_1^2\sigma_2^2}{N_{ijk}^2}(\frac{\theta_{ijk}^2}{\sigma_1^2} - \frac{2\rho\theta_{ijk}}{\sigma_1\sigma_2} + \frac{1}{\sigma_2^2})$. We can see the variance is of the order $\frac{1}{N}$ since σ^2s are of the order N from Theorem 2.

VI. Structure Learning from Randomized Data

The problem of learning Bayesian network structure from sample data D is to find a network structure G' that best matches the sample data D. Our problem here is to learn the network structure from the randomized data $\tilde{D}$. We use K-2 algorithm for structure learning. K-2 is a greedy search algorithm that searches for a DAG G' that (approximately) maximizes a score function $Score(D, G)$. The two score functions we discuss in this paper are Bayesian score and BIC/MDL score. The Bayesian score for a node is $Score(X_i, Pa_i(X_i)) = \prod_{j=1}^{J_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \prod_{k=1}^{K_i} \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ij})}$. The BIC/MDL score for a node is given by $Score(X_i, Pa(X_i)) = \sum_{k=1}^{K_i}\sum_{j=1}^{J_i} N_{ijk} log(\theta_{ijk}) - \frac{N}{2}\#(X_i, Pa(X_i))$,where $\#(X_i, Pa(X_i))$ is the number of parameters we need to represent $p(X_i|Pa(X_i))$. The decomposability property of those score function makes a single operation in K-2 algorithm as the addition of a parent to a variable. The addition of a parent corresponds to two different candidate structures G_1 and G_2. By comparing $P_{old} = score(D, X_i, Pa(X_i))$ and $P_{new} = score(D, X_i, Pa(X_i) \cup \{Z\})$, where Z is a new candidate parent for variable X_i, K-2 algorithm decides if there is a link between candidate parent Z and node X_i. We use the two sets of estimated sufficient statistics $\hat{N}_{ijk}$s to calculate P_{old} and P_{new}. The estimation of sufficient statistics is done as described in Section IV for each variable and the difference between P_{old} and P_{new} is caused only by the sufficient statistics associated with node X_i. The estimation of sufficient statistics is done as described in Section IV for structures G_1 and G_2. The framework of structure learning for the two setups were discussed in Section III-B.

One problem with structure learning is that estimation errors of the sufficient statistics tend to cause extra links, which are links that appears in the structure learned from randomized data $\tilde{D}$ but not in the structure learned from original data D. Missing links happen only when the randomization is relatively large. Missing links are those links that are learnt from the original data D but are not learnt from the randomized data $\tilde{D}$. The extra-link problem is not difficult to understand from the statistical definition of independence. For example, if we have a sample data from two independent discrete random variables A and B, we can conclude statistically that A and B are independent if we have $p(A = i, B = j) = p(A = i)p(B = j) \pm \varepsilon \ \forall i, j$, where $p(A = i, B = j)$, $p(A = i)$, and $p(B = j)$ are estimated from their respective relative frequency in the sample set and ε depends on the specific independence testing method used. If the data samples of A and B are post-randomized, the relative frequencies can only be estimated using the available randomized samples. The estimation errors tend to cause $p(A = i, B = j) > p(A = i)p(B = j) + \varepsilon$ or $p(A = i, B = j) < p(A = i)p(B = j) - \varepsilon$ for some i and j. If the same independence testing is used, we tend to conclude that A and B are dependent. Similar argument holds for conditional independence which is encoded by the Bayesian network structure.

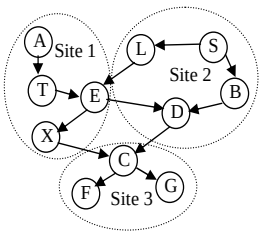
Our experiments show that the extra links exist even for relatively small estimation errors. This kind of effect of estimation error on indpendence testing usually causes $Score(C, \{AB\}) > Score(C, \{A\})$ although C is actually independent of B given A. Thus an extra link $B \rightarrow C$ will usually result. The above discussion suggests that if we want to learn correct structures from the randomized data, we should penalize complex structures. For Bayesian score, we propose adding a parent only when $P_{new} > \eta P_{old}$, where η is a suitable threshold. For BIC/MDL score, we propose modifying the score function by increasing the penalty term (description length); that is, $Score_B(X_i, Pa(X_i)) = \sum_{k=1}^{K_i}\sum_{j=1}^{J_i} N_{ijk} log(\theta_{ijk}) - C^* \frac{N}{2}\#(X_i, Pa(X_i))$ for some $C^* > 1$. Experiments show that the threshold η and C^* depend on the level of randomization, More the randomization, larger the threshold η and C^* should be. The underlying relationship between the threshold η (or C^*) and randomization under the available samples can be a further research topic. We believe there exists optimal choices for threshold η and C^* for a designed randomization scheme.

VII. Experimental Results

We now present some experimental results that demonstrate the accuracy of the BN learning algorithm for different levels of randomization. In Setup I, less number of variables are randomized than Setup II. Therefore, the experimental results of Setup I are always better than those for Setup II. Hence, we present only the results from Setup II here.

A. Parameter Learning: Non-uniform Randomization

In this experiment, we use the Bayesian Network shown in Fig. 2, where the variables are distributed over three parties. All variables are binary except variables L and B, which are ternary. The conditional probabilities of the different nodes are also shown. $20,000$ samples were generated from this Bayesian Network to form the data set D. This data set was



A	0.7,0.3	T	0.1,0.9,0.9,0.1
S	0.5,0.5	L	0.3,0.7,0.4,0.15,0.3,0.15
X	0.2,0.6,0.8 ,0.4	F	0.25,0.9,0.75,0.1
E	0.25,0.8,0.15,0.5,0.3,0.4,0.75,0.2,0.85,0.5,0.7,0.6		
D	0.7,0.65,0.1,0.4,0.8,0.35,0.3,0.35,0.9,0.6,0.2 ,0.65		
C	0.9,0.4,0.6,0.25,0.1,0.6,0.4,0.75		
B	0.8 ,0.15,0.1,0.5,0.1,0.35		
G	0.2,0.4,0.8,0.6		

Fig. 2. Bayesian Network for Experiment 1

then randomized according to the scheme described in Table

I, where variables T, S, and G were considered not sensitive and hence not randomized. Note that we use a non-uniform randomization with different levels of randomization for different variables. The corresponding at most γ amplification are also shown in Table I. $K^* = 2$ for Binary randomization while $K^* = 3$ for ternary randomization. Table II shows the parameter of all nodes learnt from the randomized data using the algorithm described in Section III for setup II. All the values in the Table are average over 5 runs, with the corresponding standard deviation indicated in parenthesis. It is clear from Table II that the proposed algorithms can accurately learn the BN parameters for both Setups.

TABLE I

RANDOMIZATION PERFORMED TO THE VARIABLES

A,D	Binary sym	$p = 0.25$	$\gamma = 3$
L,B	Ternary sym	$p = 0.15$	$\gamma = 4.67$
E	Binary sym	$p = 0.2$	$\gamma = 4$
X	Binary sym	$p = 0.4$	$\gamma = 3$
C,F	Binary non_sym	$p_1 = 0.1, p_2 = 0.25$	$\gamma = 9$

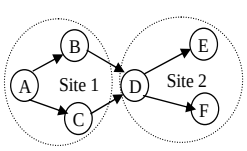
B. Parameter Learning: Uniform Randomization

In this experiment, we use a uniform randomization and test the accuracy of the BN parameters as a function of the randomization parameter p. We used the Bayesian Network shown in Fig. 3. All nodes are binary. We consider the parameters of node D, which has parents in a different site. 5,000 samples were generated from the above Bayesian Network. Binary Symmetric randomization with parameter p was used to randomize the variables. Fig. 4 (top) shows a graph of the parameters as a function of p (mean of the estimated parameters over 10 runs is plotted; plot of mean plus one standard deviation is also included). It is clear from the figure that for randomization parameter $p \leq 0.25$, we can estimate the BN parameters with almost no error. Even for p values up to 0.3, we get reasonably good parameter estimates. From a privacy perspective, $p = 0.25$ corresponds

TABLE II

MEAN AND STANDARD DEVIATION ($\times 10^{-2}$) OVER 5 RUNS OF PARAMETERS LEARNT FROM THE RANDOMIZED DATA

A	0.70(0.70)	0.30(0.70)		
T	0.10(0.50)	0.90(0.77)	0.90(0.50)	0.097(0.77)
S	0.50(0.00)	0.49(0.00)		
L	0.30(0.49) 0.31(0.93)	0.71(0.57) 0.15(0.45)	0.39(0.64)	0.14(0.55)
B	0.80(0.77) 0.10(0.11)	0.16(0.41) 0.36(0.65)	0.094(0.71)	0.49(0.73)
E	0.25(0.20) 0.31 (2.6) 0.86(2.7)	0.81(0.90) 0.41(2.34) 0.50(1.2)	0.14(2.7) 0.75(2.0) 0.69(2.64)	0.51 (1.2) 0.19(0.90) 0.59(2.3)
D	0.69 (2.2) 0.79(1.7) 0.89(3.34)	0.65(1.3) 0.39(5.65) 0.62(0.77)	0.11(3.3) 0.31(2.3) 0.21(1.7)	0.38(0.77) 0.35(1.3) 0.61(5.7)
X	0.20(0.80)	0.60(1.1)	0.81(0.80)	0.40(1.1)
C	0.90(2.0) 0.10(2.0)	0.38(1.6) 0.62(1.6)	0.61(2.6) 0.39(2.6)	0.25(2.1) 0.75(2.1)
F	0.24(0.73)	0.91(1.1)	0.77(0.73)	0.092(1.1)
G	0.20(0.30)	0.40(0.29)	0.80(0.30)	0.60(0.29)



A	0.5 0.5	G	0.3 0.7
B	0.8 0.2 0.2 0.8	E	0.8 0.2 0.2 0.8
C	0.5 0.9 0.7 0.35 0.5 0.1 0.3 0.65		
D	0.95 0.15 0.75 0.1 0.05 0.85 0.25 0.9		
F	0.8 0.2 0.2 0.8		

Fig. 3. Bayesian Network for Experiment 2

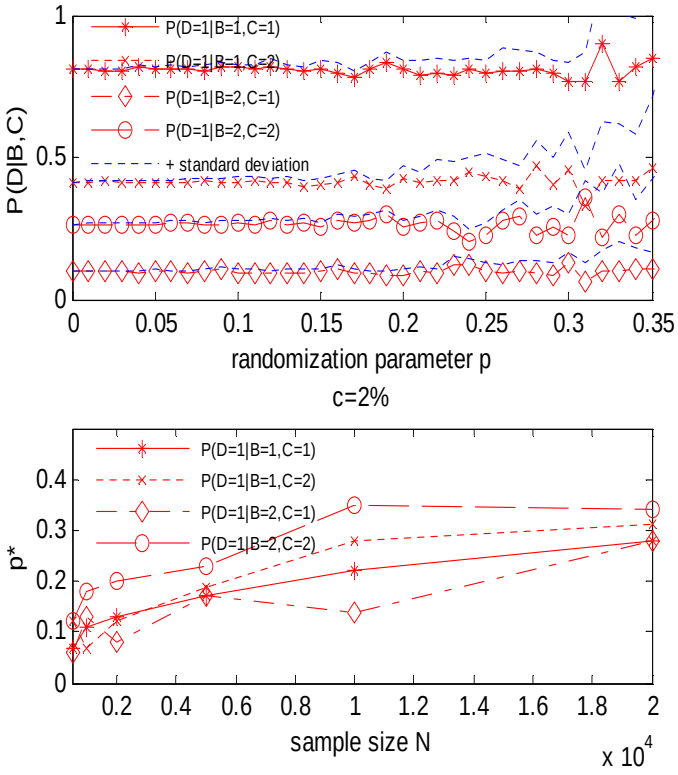


Fig. 4. (Top) Estimated parameters $\hat{P}(D|B,C)$ vs. p; (Bottom) p^* vs. N

to an amplification factor of $\gamma = \frac{1-0.25}{0.25} = 3$ with $K^* = 2$. We have to point out that this is done with 5,000 samples. Under the same accuracy requirement, it is intuitive that the p value can be closer to 0.5 if more samples are available. The closer p is to 0.5, the closer γ is to 1. Another way to assess the performance of the algorithms is to determine the maximum level of randomization p that we can use for a given accuracy. Towards that end, for a given level of required estimation accuracy, defined in terms of an absolute parameter estimation error threshold c, let p^* be the smallest value of p (obtained by averaging over ten runs) for which the absolute value of the estimation error exceeds c. Fig. 4(bottom) shows the variation of p^* as a function of the sample size N, for values of $c = 2\%$. Note that $c = 2\%$ corresponds to very small parameter estimation error.

C. Structure Learning

In this experiment, we test the accuracy of BN structure learning from randomized data. 10,000 samples from the BN in Fig. 3 was used. All variables were randomized using a binary symmetric randomization with parameter p. The K-2 algorithm with threshold η for Bayesian score or penalty

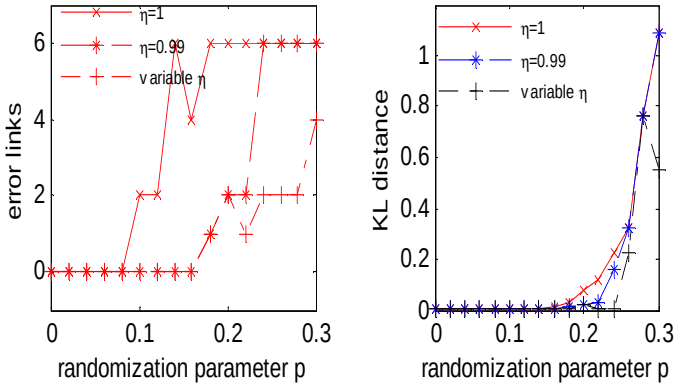


Fig. 5. Structure learning using Bayesian Score: (Left) No. of links in error; (Right) KL distance

term C^* for BIC/MDL score was used to learn the BN structure from the randomized data. We quantify the error in structure learning with two different error measures: (a) Sum of missing links and extra links and (b) KL-distance between the joint probability of the learnt BN and the true BN. The latter actually incorporates errors in the structure as well as the parameters and might be better. Fig. 5 (left) shows the number of links in error (sum of missing links and extra links) as a function of the randomization parameter p, for the case of Bayesian scores. Fig. 5 (right) depicts a similar graph for KL-distance. Three different choices of the threshold were considered: $\eta = 1, \eta = 0.99$, and a variable η value depending on the randomization parameter value p chosen as follows $\eta = 0.99$ if $0 < p \leq 0.15$, $\eta = 0.98$ if $0.15 < p \leq 0.25$ and $\eta = 0.97$ if $0.25 < p \leq 0.30$. We have similar results using BIC/MDL score with three different values for the penalty term $C^* = 1, C^* = 4$, and $C^* = 8$. Due to page limitation, the results using BIC/MDL are not presented here. We would like to add that the structure error was always contributed by extra links, with just one exception (when Bayesian score with variable η is used) where we had one missing link. It can be seen from the graphs that a variable value of η gives better results than a fixed value. For the case of Bayesian score, we can have a randomization level of $p = 0.2$, with little error in structure learning. From a privacy perspective, this corresponds to a value of $\gamma = 4$ with $K^* = 2$. The performance with BIC/MDL score is even better, where we can have a randomization level of $p = 0.25$, with little error in structure learning. This corresponds to a value of $\gamma = 3$. We also have to point out here that this is done using a sample size of 10,000 points. With more samples, it is intuitive that we can still get small errors in structure learning with more randomization; i.e., p closer to 0.5.

VIII. Discussion and Conclusions

We have proposed a post randomization technique to learn the structure and parameters of a Bayesian network from distributed heterogeneous data. Our method estimates the sufficient statistics from the randomized data, which are subsequently used to learn the BN. For structure learning, we used a modified score function to deal with the familiar extra-link problem. Experimental results with different levels of randomization and different sample sizes show that our method is capable of accurately estimating the BN. We quantified the privacy of our randomization scheme using the concept of γ-amplification and K^* similar to the concept of K-anonimity. We showed that we obtain a fairly good level of privacy. We believe the post randomization can be easily extended to many other Privacy-preserving data mining applications whose computation also only depend on a set of sufficient statistics such as the decision tree learning. The randomization in our experiment is implemented to the individual variables. However, the randomization can also be implemented to combined variables, which might include all the variables in one party. Combining all the variables in one party as one combined variable might prevent privacy breach between the variables in the same party.

IX. Acknowledgements

This work was supported by the United States National Science Foundation Grant IIS-0350533. We would like to thank Dr. Kargupta for many fruitful discussions and ideas.

References

[1] R. Agrawal and R. Srikant, *Privacy-preserving data mining*, In Proceedings of SIGMOD Conference on Management of Data, pages 439-450, May 2000.

[2] R. Chen, K. Sivakumar, and H. Kargupta, *Collective Mining of Bayesian Networks from Distributed Heterogeneous Data* (accepted) Knowledge and Information Systems Journal.

[3] 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.

[4] W. Du and Z.Zhan, *Using Randomized Response Techniques for Privacy-Preserving Data Mining*, In Proceedings of the 9th ACM SIGKDD, Washington, DC, USA. August 2003. Page 505-510.

[5] A. Evfimievski, J. Gehrke, and R. Srikant, *Limiting privacy breaches in privacy preserving data mining* In proceedings of the ACM SIGMOD/POD Conference, pages 211-222, San Diego, CA, June 2003.

[6] 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, Vol.14 1998 pages 463-478.

[7] D. V. Hinkley, *On the ratio of two correlated normal random variables*, Biometrika (1969), 56, 3, pages 635-639.

[8] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, *On the privacy Preserving Properties of Random Data Perturbation Techniques*, In Proceedings of the IEEE International Conference on Data Mining, Pages 99-106, Melbourne, FL. November 2003.

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

[10] K. Liu, H. Kargupta, and J. Ryan, *Multiplicative Noise, Random Projection, and Privacy Preserving Data Mining from Distributed Multi-Party Data*, (In Communication), 2003.

[11] J. Ma and K. Sivakumar, *Privacy-Preserving Bayesian Network Learning Using Post Randomization*, (in preparation), 2006.

[12] D. Meng, K. Sivakumar and H. Kargupta, *Privacy-Sensitive Bayesian Network Parameter Learning*, In the Fourth IEEE International Conference on Data Mining. Brighton, UK. November 2004.

[13] L.Sweeney, *k-anonymity: a model for protecting privacy*, International Journal on uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557-570,2002.

[14] S. Rizi and J. R . Haritsa, *Maintaining data privacy in association rule mining*, In the proceedings of the 28th VLDB Conference , Hongkong, China, 2002.

[15] R. Wright and Z. Yang, *Privacy Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data*, In Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining.