A Post Randomization Framework for Privacy-Preserving Bayesian Network Parameter Learning

JIANJIE MA K.SIVAKUMAR
School of Electrical Engineering and Computer Science,
Washington State University
Pullman, WA. 99164-2752
{jma, siva}@eecs.wsu.edu

Abstract: - Post Randomization technique has been successfully used in statistical disclosure limitation. The application of Post Randomization technique to Privacy-Preserving data mining is explored in this paper. The problem of privacy-preserving Bayesian network parameter learning is considered as a specific example. We propose to use post randomization technique to randomize the privacy-sensitive variables in learning Bayesian network parameters from distributed heterogeneous databases. The only required information from the data set is a set of sufficient statistics for learning Bayesian network parameters. The proposed method estimates the sufficient statistics from the randomized data. We show both theoretically and experimentally that this method learns a set of accurate parameters, even under large levels of randomization. We also illustrate the trade off between privacy and accuracy by simulations.

Key-Words: - Bayesian Network, Privacy-Preserving Data Mining, Distributed Heterogeneous Databases, Post Randomization

1. 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 [1,2]. The other 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 [4]. Privacy-preserving Bayesian network (BN) learning is a more recent topic. Wright and Yang [10] discuss privacy-preserving BN structure computation on distributed heterogeneous databases while Meng *et al*. [8] have considered the privacy-sensitive BN 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. The number of secure computation operations increases exponentially with the possible configurations of the problem variables. The current work on privacy-preserving BN learning focuses on the multiparty models, which requires that every party have some computational capability. Besides this model, our work considers a model where there is a data miner who actually does all the computations for the participating parties. SMC method has the following two drawbacks: (1) it assumes a semi-honest model, which is often unrealistic in the real world (2) it requires large volumes of synchronized computations among

participating parties. Most of the synchronized computations are overheads due to privacy requirement. Post randomization overcomes the drawbacks of SMC method by a trade off between accuracy and privacy. A malicious party who does not obey the protocol in SMC method can easily get some private information of other parties which he will not be able to if post randomizations are implemented to individual data records.

2. Problem Formulation

Privacy-Preserving BN learning involves distributed databases, where the database is owned by several parties. If the database is homogeneously distributed, privacy-preserving BN Learning is relatively easy since every party can send data miner (or other parties) a set of sufficient statistics from his part of the database. Privacy of individual records will not be breached by sending sufficient statistics to other parties or data miner. The problem of privacy-preserving BN learning from heterogeneous database is that several parties who each own a vertical portion of the database want to learn a global BN for their mutual benefits but they are concerned about the privacy of their sensitive variables. In this paper, we consider the problem of BN parameter learning for the case of discrete variables. We consider the following two models.

Model I: There is no data miner; every party has to do some portion of the learning computations. Every party sends their randomized data to those parties who need those data.

Model II: There is a data miner who is does all computations for the participating parties. Every party simply sends all their randomized data to the data miner.

3. Privacy Analysis for Post Randomization

Consider a database D with n variables $\{X_1 \cdots X_n\}$, where X_i takes discrete values from the set S_i. The post randomization for variable X_i is a (random) mapping $R_i : S_i \to S_i$, based on a set of transition probabilities $p^i_{lm} = p(\tilde{X}_i = k_m \mid X_i = k_l)$, 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 X_i with original value k_l is randomized to the value k_m. Post Randomization is so named because the randomization happens after data have been collected. Let $P^i = \{p^i_{lm}\}$ denote the $K_i \times K_i$ matrix that has p^i_{lm} as its $(l,m)th$ entry, where K_i is the cardinality of the set S_i. The condition that P^i is nonsingular has to be imposed if we want to estimate the frequency distribution of variable X_i from the randomized variables. In the following, we give out some simple but effective post randomization schemes on which our experiments are based. If variable X_i takes binary values, we can use Binary Randomization as shown in Fig. 1(a). If the variable is ternary, ternary symmetric channel as shown in Fig.1(b) can be used.

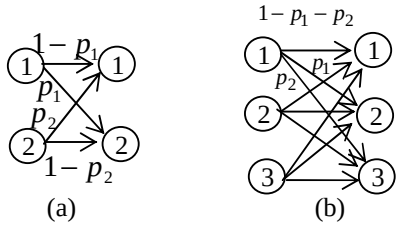


Fig. 1: Randomization Schemes

We can apply the same randomization schemes independently to all of the variables: uniform randomization to the data set. Alternatively, we can use a non-uniform randomization where different post randomization schemes are applied to different variables independently. The non-uniform randomization is effective when different variables have different sensitivity levels. 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. 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 estimators for frequency counts will also be lager under the same sample size. One solution for this problem is to partition the K_i categories 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. The problem of how many groups should the K_i values be partitioned into is a matter of design choice.

The post randomization can also be implemented to several variables simultaneously. For example, the variables X_i and X_j can be randomized simultaneously according to transition probability $p(\tilde{X}_i = l_1, \tilde{X}_j = l_2 \mid X_i = k_1, X_j = k_2)$. Randomizing variables simultaneously can avoid the possible inconsistency of the database caused by randomization.

We consider the notion of privacy introduced by Evfimievski *et al*. [5] in terms of an amplification factor γ. The amplification γ in [5] is proposed in the framework where every data record should be randomized with a factor greater than γ, before the data are sent to the data miner, to limit privacy breach. However, in this paper, we use the amplification γ purely as a worst-case quantification of privacy for a designed post randomization scheme. It is proved in [5] that if the randomization operator is at most γ amplifying, revealing $\tilde{X}_i = k$ will cause neither an upward ρ_1 -to- ρ_2 privacy breach nor a downward ρ_2 -to- ρ_1 privacy breach if $\frac{\rho_2}{\rho_1}\frac{1-\rho_1}{1-\rho_2} > \gamma$. Clearly, the smaller the value of γ, the better is the worst case privacy. Ideally we would like to have $\gamma = 1$. The at most γ amplification provides a worst case quantification of privacy. However, it does not provide any information about privacy in general. Besides γ, we use $K = \min_{k'} \#\{k \mid P(\tilde{X}_i = k' \mid X_i = k) > 0\}$, which is the minimum number of possible categories that can be randomized to category k' in a designed post randomization, as another quantification of privacy. This *K* indicates the privacy preserved in general. It is similar to the *K* defined in *K*-anonymity in [9] but in probabilistic sense. If we group the categories of a variable into several group, then *K* become smaller in general

4. Post Randomization Framework for Parameter Learning

For parameter learning, we assume the structure *G* is fixed and known to every participating party. For Model I, we use the definition of cross variable and cross parents defined in [3]. N_{ijk} is the number of records such that X_i is in k th category while its parents are in j th category.

For each party a_i

(1) Randomize cross parents at same site according to their respective privacy requirements using post randomization described in Section 3. 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 in 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 using local data and randomized parent data from other parties.

(5) Compute the parameters for cross variables using the estimated sufficient statistics $\hat{N}_{ijk}$ s.

(6) Share the parameters with all other parties.

Local variables at each site are not randomized for local calculations.

Steps of learning parameters for model II:

For each party a_i:

(1) Randomize all sensitive variables according to their respective privacy requirements using post randomization described in Section 3. Randomizations are done independently for each (combined) variable and each record.

(2) Send randomized data and their corresponding probability transition matrices to the data miner.

For the data miner:

(1) Estimate the sufficient statistics N_{ijk} for each node X_i using the randomized data from participating parties.

(2) Estimate the parameters using the estimated sufficient statistics $\hat{N}_{ijk}$.

(3) Broadcast the parameters to all parties.

The details of estimation of sufficient statistics and parameter (step 4 and 5 for Model I, Step 1 and 2 for data miner in Model II) from randomized data are described in Section 5.

5. Estimation of Sufficient Statistics & Parameters from Randomized Data

The problem of privacy-preserving BN Parameter learning can be decomposed into a series of estimation of N_{ijk} s for each node X_i and a given fixed structure G from the randomized data $\tilde{D}$. Consider the following general case: Variable X_i with cardinality K_i has Q parent nodes $Pa_i(1),\cdots,Pa_i(Q)$. The cardinality of $Pa_i(q)$ is $K_{Pa_i(q)}$. These variables can be arbitrary vertically partitioned to different parties in both models. The randomization of each (combined) variable can also be done by grouping the categories of the variables into groups. We have the following different cases for estimating N_{ijk} s from the randomized data $\tilde{D}$ due to simultaneous randomization.

(a) X_i and its parents are all randomized independently each other.

(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 non-parent variables.

For (b) and (c) above, we can consider the simultaneously randomized variables as combined variables in estimating the sufficient statistics. For example, if variable X_i is randomized simultaneously with one of its parents $Pa_i(1)$, N_{ijk} is equal to the number of records such that $(X_i;Pa_i(1))=(k,j^1)$, $Pa_i(2)=j^2$,…, $Pa_i(Q)=j^Q$, where $(X_i;Pa_i(1))$ is a combined variable. Thus, we can estimate the N_{ijk} s from the randomized data by considering $(X_i;Pa_i(1))$ as a single variable with cardinality $|K_i|\times|K_{Pa_i}(1)|$. For case (d), since the current N_{ijk} doesn't involve the variable randomized simultaneously with X_i, the data miner can get the marginal transition probability matrix from the given transition matrix of the combined variable.

From the above arguments, we conclude that the cases (b), (c), and (d) above can effectively be considered to be equivalent to case (a). Hence, without loss of generality, we can discuss case (a) only. We denote by $Pa(X_i)$ as a compound variable for all the parents of Variable X_i. Hence $Pa(X_i)$ takes $J_i=\prod_{q=1}^{Q}K_{Pa_i(q)}$ different values.

$N_{ij} = \sum_{k=1}^{K_i} N_{ijk}$ and N_i is $J_i K_i$ dimensional vector of N_{ijk} values, that is $N_i = (N_{i11}, N_{i12}, \cdots, N_{i1K_i}, N_{i21}, \cdots N_{iJ_iK_i})^t$, where superscript *t* denotes transpose. $N_i(l)$ is an element of N_i. $\tilde{N}_{ijk}$, $\tilde{N}_{ij}$ and $\tilde{N}_i$ are defined similarly as N_{ijk}, N_{ij} and N_i respectively but for the randomized data $\tilde{D}$. $\hat{N}_{ijk}$, $\hat{N}_{ij}$ and $\hat{N}_i$ are estimators of N_{ijk}, N_{ij} and N_i respectively. Given the training data *D* with *N* records of variables X_i and its *Q* parents in the above general case, if they are post-randomized with transition probability matrices P^i, $P^{Pa^i(1)}$, …, $P^{Pa^i(Q)}$, respectively, we have the following theorem.

Theorem 1: $E[\tilde{N}_i \mid D] = P^t N_i$, where $P = P^i \otimes P^{pa^i}$ and $P^{pa^i} = P^{pa^i(1)} \otimes P^{pa^i(2)} \otimes \cdots \otimes P^{pa^i(Q)}$, $\otimes$ denotes Kronecker matrix product. Moreover, $Cov[\tilde{N}_i \mid D] = \sum_{l=1}^{J_i K_i} N_i(l) V_l$ where V_l is a $K_i J_i \times K_i J_i$ covariance matrix such that its $(l_1, l_2)th$ element is

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

Proofs are omitted here due to the page limitations. Interested readers can refer to a longer version of this paper for details [7]. The following theorem establishes the bias and variance of the estimator $\hat{N}_i = (P^t)^{-1}\tilde{N}_i$. Its proof is straight-forward and is omitted.

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

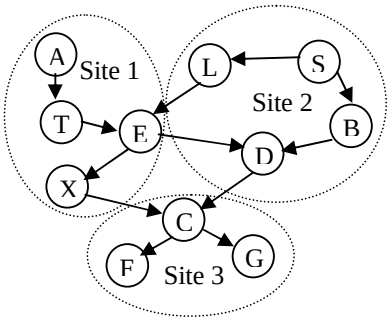
We can use the estimated sufficient statistics to get ML estimate of the parameters as $\hat{\theta}_{ijk} = \frac{\hat{N}_{ijk}}{\hat{N}_{ij}} = \frac{\hat{N}_{ijk}}{\sum_{k=1}^{K_i} \hat{N}_{ijk}}$ and the MAP estimate of the parameters as $\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + \hat{N}_{ijk}}{\alpha_{ij} + \hat{N}_{ij}}$, where the prior distribution of θ_{ij} is assumed to be Dirichlet with parameter $\{\alpha_{ij1}, \alpha_{ij2}, \cdots, \alpha_{ijr_i}\}$. The distribution of estimator $\hat{\theta}_{ijk}$ is discussed in [7]. One important result from [7] is that the distribution of the estimator $\hat{\theta}_{ijk}$ can be approximated as a normal distribution with mean θ_{ijk} and with a variance of the order of $\frac{1}{N}$, where N is the training sample size.

6. Experimental Results

6.1 Non-uniform Randomization

In this experiment, we use the Bayesian Network shown in Fig. 2, where the variables are distributed over three sites. 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 dataset *D*. This data was then randomized according to the scheme described in Table 1, where variables *T*, *S*, and *G* were considered not sensitive and hence not randomized. The corresponding at most γ amplification is also shown in Table 1. *K*=2 for Binary randomization whereas *K*=3 for ternary randomization. Table 2 shows a part of parameters learnt from the randomized data using the algorithm described in Section 4 for Model II. Less randomization occurs in Model I, so the results for Model I are better than

those for Model II. The remaining part can be calculated by one minus the given part. All the values in the Table are average over 5 independent runs, with the corresponding standard deviation indicated in parenthesis. It is clear from the Table that the proposed algorithms can accurately learn the BN parameters for both scenarios, even for moderate levels of randomization.



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: A Bayesian Network for experiment 6.1

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

Table 1: Randomization performed

6.2 Trade off between Privacy and Accuracy

In this experiment, we use the Bayesian network shown in Fig. 3, where variables are distributed over two sites. All Variables are binary. We generated 10,000 samples from this Bayesian Network. In order to see the trade off between privacy and accuracy, we randomize the samples using binary symmetric randomization with different levels of $p = p_1 = p_2$ and learn parameters from randomized samples using the method discussed in Section 4. As in previous experiment, we only present the results using Model II. In the experiment using Model II, every variable is randomized using the Symmetric Binary Randomization with the same randomization level *p*. Since parameters associated with a node is nothing but the conditional probability given its parents, the accuracy of parameters associated with a node can be measured by conditional Kullback-Leibler (CKL) distance between the parameters learnt from randomized data and those learnt from non-randomized data. The CKL distance for node *i* in our case is

$$D(X_i, p) = \sum_{j=1}^{J_i} P(pa_i = j) D_{KL}(P^{(0)}(X_i \mid pa_i = j), P^{(p)}(X_i \mid pa_i = j)),$$

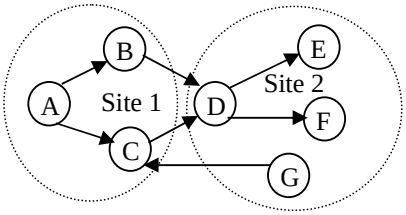
where $P^{(0)}(X_i \mid pa_i = j)$ and $P^{(p)}(X_i \mid pa_i = j)$ are the parameters learnt from non-randomized data and

A	0.70(0.70)	*T*	0.10(0.50) 0.90(0.77)
S	0.50(0.00)	*X*	0.20(0.80)0.60(1.1)
L	0.30(0.49)0.71(0.57)0.39(0.64)0.14(0.55)		
B	0.80(0.77)0.16(0.41)0.094(0.71)0.49(0.73)		
E	0.25(0.20)0.81(0.90)0.14(2.7)0.51(1.2) 0.31(2.6)0.41(2.34)		
D	0.69(2.2)0.65(1.3)0.11(3.3)0.38(0.77)0.79(1.7) 0.39(5.65)		
C	0.90(2.0)0.38(1.6)0.61(2.6)0.25(2.1)		
F	0.24(0.73)0.91(1.1)		
G	0.20(0.30) 0.40(0.29)		

Table 2: Mean and standard deviation ($\times 10^{-2}$) over 5 runs of parameters learnt from the randomized data**.**

those learnt from randomized data with randomization level *p* respectively and D_{KL} denotes the ordinary KL distance between two distributions. We present those distances associated with node *C*, node *D* and node *F* in Fig. 4. Those nodes are typical nodes for the given Bayesian network. The averages are over 10 independent runs. Average plus one standard deviation of 10 runs is also depicted (with dotted line). From the Fig. 4, we can clearly see the trade off between accuracy and privacy. Since we use the symmetric binary randomization, more privacy is preserved with bigger p when $p < 0.5$. With 10,000 training samples, the method still gets

good accuracy when p=0.3.



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: A Bayesian Network for experiment 6.2

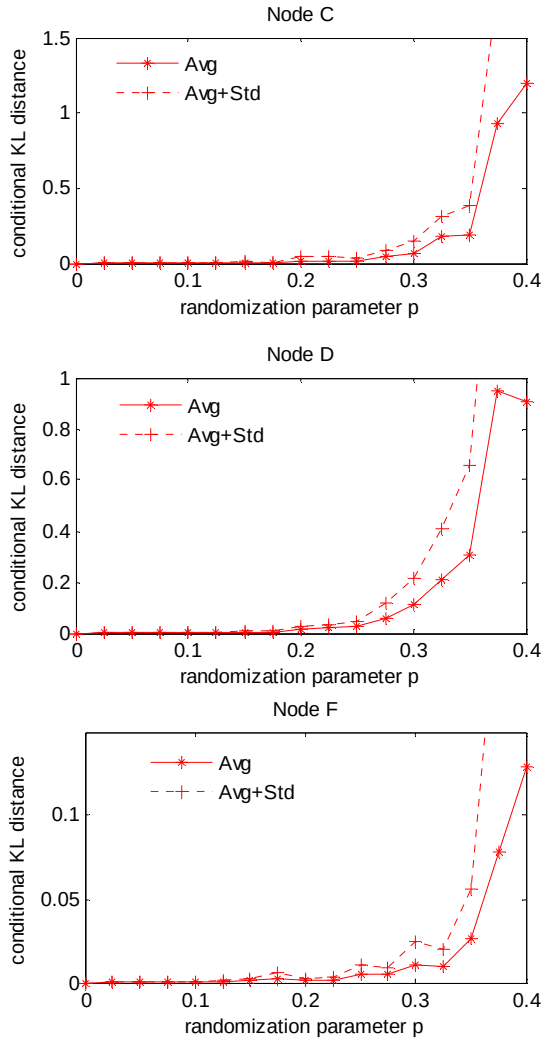


Fig. 4: CKL distance vs. randomization level p

6.3 Training Sample size

As pointed out in Section 4, the variance of the estimator of parameter θ_{ijk} is of the order of one over the sample size N. Thus, under the same accuracy requirement, more privacy can be preserved if there are more training samples. This experiment is performed to illustrate the effect of training sample size. We generated 2500×2^8 training samples using Bayesian network in Fig. 3. The proposed method in Section 4 is used to learn the Bayesian parameters from randomized data with randomization levels $p=0.1$, $p=0.2$, $p=0.3$, and $p=0.4$ with training sample size 2500×2^k $(k=1\cdots 8$) respectively. The experiment results are shown in Fig. 5. As in experiment 6.2, the average is over 10 independent runs and the average plus one standard deviation is also shown. The experiment results for randomization level $p=0.4$ are shown separately. Those Conditional distances out of the scale of vertical axis are not shown in the Figure. From this experiment, we can clearly see that training sample size play a key role in the trade off between accuracy and privacy. We can see that when the training sample size is very large, we can have both good privacy and good accuracy.

7. Conclusion

We have proposed a post randomization technique to learn 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 parameters. Our experiments show that post randomization is an efficient, flexible, and easy-to-use method to learn Bayesian network parameters from privacy sensitive data. Currently, we are exploring the extension of post randomization techniques to learn BN structure from sensitive data. The idea of estimating sufficient statistics from randomized data can be used to learn other data mining models like decision trees. We plan to report these extensions and applications in a future publication.

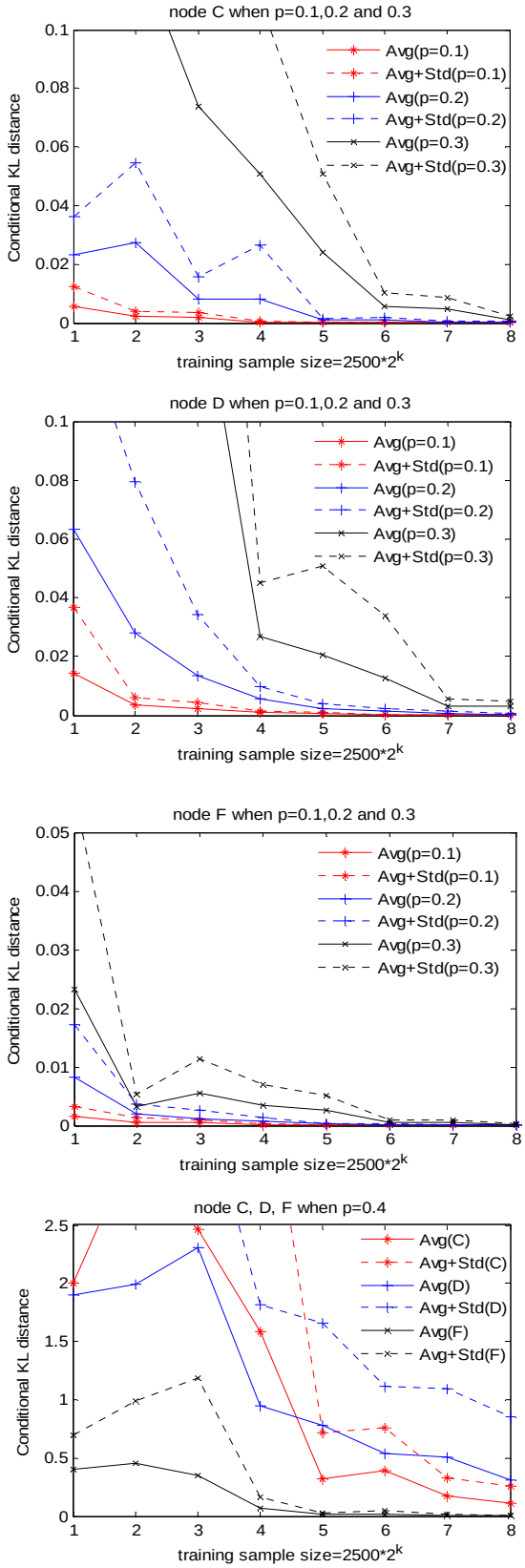


Fig. 5: CKL distance vs. training sample size

References:

[1] D. Agrawal and C. C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithm, SIGMOD 2001

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

[3] R. Chen, K. Sivakumar, and H. Kargupta, "Collective Mining of Bayesian Networks from Distributed Heterogeneous Data," *Knowledge and Information Systems Journal*, vol. 6, 2004.

[4] 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.

[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] J. Ma and K. Sivakumar, Privacy-Preserving Bayesian Network Learning Using Post Randomization, (in preparation), 2005.

[8] 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.

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

[10] 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.