A Probabilistic Approach to Privacy-sensitive Distributed Data Mining

Srujana Merugu and Joydeep Ghosh
Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78712
{merugu, ghosh }@ece.utexas.edu

***Abstract*— We introduce a general framework for inter-enterprise distributed data mining that takes into account privacy requirements. It is based on building probabilistic or generative models of the data at each local site. The parameters of these models are then transmitted to a central location instead of the original or perturbed data. We mathematically show that the best representative of all the local models is a certain " mean" model, and empirically show that this model can be approximated quite well by generating artificial samples from the underlying distributions using Markov Chain Monte Carlo techniques, and then fitting a combined global model with a chosen parametric form to these samples. We also propose a new measure that quantifies privacy based on information theoretic concepts, and show that decreasing privacy leads to a higher quality of the combined model and vice versa. Empirical results on (distributed) clustering and classification - two of the most fundamental data mining procedures - are provided on different kinds of datasets consisting of vectors, variable length sequences and high dimensional directional data, to highlight the generality of our framework. The results show that high quality global models can be achieved with little loss of privacy.**

I. INTRODUCTION

While data mining and pattern recognition algorithms invariably operate on centralized data, usually in the form of a single flat file, in practice, related information is often acquired and stored at geographically distributed locations due to organizational or operational constraints. Centralization of such data before analysis may not be desirable because of computational or bandwidth costs. In some cases, it may not even be possible due to variety of real-life constraints including security, privacy, proprietary nature of data/software and the accompanying ownership and legal issues. The relevance of such constraints has become very evident of late as several agencies attempt to integrate their databases and analytical techniques, prompting much interest in distributed data mining (DDM).

Most of the DDM techniques developed so far have focused on classification or on association rules [1], [3], [6], [8]. There has also been some work on distributed clustering for *vertically partitioned data* (different sites contain different attributes/features of a common set of records/objects) [7], [12], and on parallelizing clustering algorithms for *horizontally partitioned data* (i.e. the objects are distributed amongst the sites, which record the same set of features for each object) [5]. These techniques, however, do not specifically address privacy issues. In contrast, privacy-preserving data mining techniques typically involve (i) query restriction, which is highly manual, or (ii) subjecting individual records or attributes to a "privacy preserving" perturbation and subsequently try to recover the original data. Most of these approaches are restricted to vector data and are applicable only in settings where a central party is collecting individual records that need to be protected.

In this paper we focus on a distributed inter-enterprise data mining setting, taking into account various privacy restrictions. The prototypical application scenario is one in which there are multiple parties with confidential databases of the same schema. The goal is to characterize (for example, to perform clustering or classification) the entire distributed data, without actually first pooling this data. For example, the parties can be a group of banks, with their own sets of customers, who would like to have a better insight into the behavior of the entire customer population without compromising the privacy of their individual customers. A fundamental assumption is that there is an (unknown) underlying distribution that represents the different datasets and it is possible to learn this unknown distribution by combining high-level information from the different sources instead of sharing individual records.

The approach that we propose is very generic, applicable to a wide range of data types and data mining procedures. It is based on building generative models on each of the local data sources and combining them centrally using only the model parameters and a minimum amount of supplementary data if need be. Instances of this approach are found in a few recent works including stacking for density estimation [11], where the combined model was empirically shown to be more accurate than the base models, and distributed cooperative Bayesian learning approach [13], where different Bayesian agents estimate the parameters of the target distribution and a meta-learner combines the outputs of these agents to obtain the final parameters. In our work, the combined global model is obtained from the "virtual samples" generated by the local models, using Monte Carlo Markov Chain sampling techniques. We prefer to use generative models for representing the local data sources as they provide a better understanding of the data distribution and are also more suited for a privacy-preserving setting.

A word about the notation: Sets such as $\{z_1, \cdots, z_n\}$ are enumerated as $\{z_i\}_{i=1}^n$. Probability density functions of a model λ is denoted by p_λ. Expectation of functions of a random variable z following a distribution p are denoted by $\mathbf{E}_{z\sim p}[\cdot]$. x is used to denote objects and takes values over the domain of data while y is used to denote class labels and z is used when a statement holds for both (x, y) and x.

II. PROBLEM DEFINITION

There are two broad approaches to distributed learning. The first approach employs data-parallel methods [5] that

are susceptible to privacy breaches. Besides, it is difficult to quantify the privacy provided by these parallel algorithms. The second approach involves building models locally and then combining them at a central location to obtain a more accurate model [3], [13]. This approach enables easy analysis of privacy costs in terms of the local model that is shared with the central location. Moreover, it allows the individual parties to use proprietary algorithms and domain knowledge, and enables reuse of legacy models [12].

In this paper, we adopt the second approach. We divide the distributed learning problem into two sub-problems — (i) choosing local models based on privacy restrictions, and (ii) combining the local models effectively to obtain a "good" global model. In our current work, we formalize the first problem by quantifying privacy costs and mainly focus on solving the second problem, assuming that the first problem is solved. This separation of concerns obviates the need for optimizing a complicated objective function that simultaneously captures the quality of model, privacy costs.

Let $\{\mathcal{X}_i\}_{i=1}^n$ be n horizontally partitioned data sources generated by a common underlying model, λ^0 and let $\{\lambda_i\}_{i=1}^n$ be the local models obtained by applying clustering or classification algorithms to these data sources. Then, the objective of the first sub-problem is to obtain the local models $\{\lambda_i\}_{i=1}^n$, such that the constraints on the privacy costs are satisfied, i.e., $\forall i,\ 1 \leq i \leq n,\ \mathcal{P}(\lambda_i) \geq \rho_i$ where $\mathcal{P}(\cdot)$ is the privacy cost function discussed in section 4, and $\{\rho_i\}_{i=1}^n$ are the lowest allowed privacy costs for the local models.

For the second sub-problem, the aim is to obtain a high quality global model that is also highly interpretable. Quality can be easily quantified in terms of how representative the model is of the true distribution, while interpretability, i.e., ease of understanding or describing the model, is difficult to quantify. Hence, to make the problem tractable, we require that the global model be specified as a mixture model based on a given parametric family (e.g., mixture of Gaussians). We call the resulting search problem of finding the highest quality global model within this family of models the **Distributed Model-based Learning** (DML) problem and state it more formally below.

Let $\{\nu_i\}_{i=1}^n$ be non-negative weights associated with the local models based on their importance or on the size of the corresponding data sources. The objective of the DML problem is to obtain the optimal global model λ_c^* belonging to a given family of models $\mathcal{F}$, i.e.,

$$\lambda_c^* = \operatorname*{argmin}_{\lambda_c \in \mathcal{F}} \mathcal{Q}(\lambda_c), \tag{1}$$

where $\mathcal{Q}(\cdot)$ is the model quality cost defined in terms of the local models and their weights.

A. Model Representation

We represent both classification and clustering models in terms of density functions. This common representation enables us to define cost functions for both types of models in a uniform manner and also leads to a systematic approach for solving both the distributed clustering and classification problems. In our scheme, a **classification model**, i.e., a generative model λ, produced by a classification algorithm is specified in terms of the joint density on the data objects x and the class labels y, $p_\lambda(x, y) = \sum_{h=1}^k I[y = h]\pi_\lambda^h p_\lambda(x|h)$, where $\{\pi_\lambda^h\}_{h=1}^k$ are the class priors, $\{p_\lambda(x|h)\}_{h=1}^k$ are the class conditional densities, k is the number of classes and $I[\cdot]$ is the indicator function. On the other hand, a **clustering** model, i.e., a generative model λ, produced by a clustering algorithm is specified in terms of probability density $p_\lambda(x)$ on the data objects x alone and is given by, $p_\lambda(x) = \sum_{h=1}^k \pi_\lambda^h p_\lambda(x|h)$, where $\{\pi_\lambda^h\}_{h=1}^k$ are the cluster priors, $\{p_\lambda(x|h)\}_{h=1}^k$ are the cluster densities and k is the number of clusters.

B. Model Quality

A natural definition for the quality cost, $\mathcal{Q}_I(\cdot)$, for a global model, is simply the " distance" from the underlying true model λ^0, i.e., $\mathcal{Q}_I(\lambda_c) = D(\lambda^0, \lambda_c)$, where $D(\cdot, \cdot)$ is a suitable distance measure for models. Since λ^0 is not known, we instead, consider the different local models $\{\lambda_i\}_{i=1}^n$ as estimators of λ^0 with weights $\{\nu_i\}_{i=1}^n$ and define the quality cost function in terms of the average distance from the local models, i.e., $\mathcal{Q}(\lambda_c) = \sum_{i=1}^n \nu_i D(\lambda_i, \lambda_c)$.

Metrics based on the norms of density functions such as the L_1 distance and the squared L_2 distance and KL-divergence are the commonly used distance measures for comparing a pair of generative models. For classification models, another suitable measure is the mismatch in the labelings, which reduces to the misclassification error when one of the models being compared is the true model. Of all these, KL-divergence is the most natural comparison measure since it is linearly related to the average log-likelihood of the data generated by one model with respect to the other. It is also a well-behaved differentiable function of the model parameters unlike the other measures. Hence, we optimize the quality cost function based on the KL-divergence and use other measures only for secondary evaluation of the experimental results. For clustering models, we consider the KL-divergence between the density functions of just the data values, i.e., $D_{KL}^{\text{clus}}(\lambda_1, \lambda_2) = KL(p_{\lambda_1}(x) \| p_{\lambda_2}(x))$ and for classification models, we consider the KL-divergence between the joint densities $p_{\lambda_1}(x, y)$ and $p_{\lambda_2}(x, y)$, i.e., $D_{KL}^{\text{class}}(\lambda_1, \lambda_2) = KL(p_{\lambda_1}(x, y) \| p_{\lambda_2}(x, y))$.

III. DISTRIBUTED MODEL-BASED LEARNING

In this section, we first pose the DML problem as an optimization problem and present an approximation using sampling techniques. Then, we propose practical algorithms to efficiently address this approximate problem for the distributed clustering and classification scenarios.

The objective of the DML problem is to obtain a global model λ_c belonging to a particular parametric family $\mathcal{F}$ such that the quality cost function $\mathcal{Q}(\cdot)$ based on KL-divergence is minimized, i.e.,

$$\lambda_c^* = \operatorname*{argmin}_{\lambda_c \in \mathcal{F}} \mathcal{Q}(\lambda_c) = \operatorname*{argmin}_{\lambda_c \in \mathcal{F}} \sum_{i=1}^n \nu_i D_{KL}(\lambda_i, \lambda_c), \tag{2}$$

where $\{\lambda_i\}_{i=1}^n$ are either the local clustering models or local classification models based on different data sources with weights $\{\nu_i\}_{i=1}^n$ summing to 1 and D_{KL} is either D_{KL}^{clus} or

D_{KL}^{class} depending on whether it is a clustering or classification scenario. This problem can be simplified using the following result.

Theorem 1 [1] *Given a set of models $\{\lambda_i\}_{i=1}^n$ with weights $\{\nu_i\}_{i=1}^n$ summing to 1, then for any model λ_c,*

$$\sum_{i=1}^{n} \nu_i KL(p_{\lambda_i}(z)\|p_{\lambda_c}(z)) = \sum_{i=1}^{n} \nu_i KL(p_{\lambda_i}(z)\|p_{\bar{\lambda}}(z)) + KL(p_{\bar{\lambda}}(z)\|p_{\lambda_c}(z)),$$

where $\bar{\lambda}$ is such that $p_{\bar{\lambda}}(z) = \sum_{i=1}^n \nu_i p_{\lambda_i}(z)$.

Applying the above theorem, we can see that the cost function in (2) can be written as

$$\sum_{i=1}^{n} \nu_i D_{KL}(\lambda_i, \lambda_c) = \sum_{i=1}^{n} \nu_i D_{KL}(\lambda_i, \bar{\lambda}) + D_{KL}(\bar{\lambda}, \lambda_c).$$

The first term on the right is independent of λ_c and hence, optimizing the cost function in (2) is equivalent to minimizing KL-divergence with respect to the mean model $\bar{\lambda}$. In the absence of any constraints, the optimal solution is just the mean model $\bar{\lambda}$, as KL-divergence is always positive and equal to zero only when both the arguments are equal. The mean model also has the following nice property, which follows from Jensen's inequality.

Theorem 2 *Given a set of models $\{\lambda_i\}_{i=1}^n$ with weights $\{\nu_i\}_{i=1}^n$ summing to 1 and the true model λ^0,*

$$D(\lambda^0, \bar{\lambda}) \leq \sum_{i=1}^{n} \nu_i D(\lambda^0, \lambda_i),$$

where $\bar{\lambda}$ is such that $p_{\bar{\lambda}}(z) = \sum_{i=1}^n \nu_i p_{\lambda_i}(z)$ and $D(\cdot,\cdot)$ is any distance function [2] that is convex in the density function of the second model.

Since the true model λ^0 is unknown, it is not possible to find out which of the models $\{\lambda_i\}_{i=1}^n$ is more accurate in terms of the ideal quality cost function $\mathcal{Q}_I(\cdot)$. However, from the above theorem, one can guarantee that the mean model will always provides an improvement over the average quality of the available models. When the individual models have independent errors, the expected gain can be considerably higher. The mean model is thus a good choice in terms of both $\mathcal{Q}(\cdot)$ and $\mathcal{Q}_I(\cdot)$, but it might not be a very interpretable model as it will, in general, have a large number of overlapping components. Instead, it is desirable to require the combined model to belong to a specified parametric family $\mathcal{F}$. Therefore, we find the model in $\mathcal{F}$ that is closest to the mean model in terms of KL-divergence. From Theorem 1, this is also the exact solution to the DML problem (2), i.e.,

$$\lambda_c^* = \operatorname*{argmin}_{\lambda_c \in \mathcal{F}} D_{KL}(\bar{\lambda}, \lambda_c) \tag{3}$$

The new optimization problem (3) is difficult to solve directly using gradient descent techniques. Therefore, we pose an approximate version of the above problem and solve it efficiently *via* maximum likelihood estimation methods. Let $\bar{\mathcal{Z}} = \{z_j\}_{j=1}^m$ be a dataset, either labeled ($z_j = (x_j, y_j)$) or unlabeled ($z_j = x_j$), obtained by sampling from the mean model. Consider the problem of finding the model $\lambda_c^a \in \mathcal{F}$ that maximizes the average log-likelihood of the dataset $\bar{\mathcal{Z}}$, i.e.,

$$\lambda_c^a = \operatorname*{argmax}_{\lambda_c \in \mathcal{F}} L(\bar{\mathcal{Z}}, \lambda_c) = \operatorname*{argmax}_{\lambda_c \in \mathcal{F}} \frac{1}{m} \sum_{j=1}^{m} \log(p_{\lambda_c}(z_j)), \tag{4}$$

where $L(\bar{\mathcal{Z}}, \lambda_c)$ is the average log-likelihood of $\bar{\mathcal{Z}}$ with respect to λ_c. As the size of the dataset $\bar{\mathcal{Z}}$ goes to ∞, the average log-likelihood converges to the cross entropy between the densities $p_{\bar{\lambda}}$ and p_{λ_c}, i.e., $\operatorname*{Lt}_{m\to\infty} L(\bar{\mathcal{Z}}, \lambda_c) = \operatorname*{Lt}_{m\to\infty} \mathbf{E}_{z\in\bar{\mathcal{Z}}}[\log(p_{\lambda_c}(z))] = \mathbf{E}_{z\sim p_{\bar{\lambda}}}[\log(p_{\lambda_c}(z))]$. Now, the cross entropy between any two densities is linearly related to the KL-divergence between them, i.e., $\mathbf{E}_{z\sim p_{\bar{\lambda}}}[\log(p_{\lambda_c}(z))] = \mathbf{E}_{z\sim p_{\bar{\lambda}}}[\log(p_{\bar{\lambda}}(z)) - \log\left(\frac{p_{\bar{\lambda}}(z)}{p_{\lambda_c}(z)}\right)] = H(\bar{\lambda}) - D_{KL}(\bar{\lambda}, \lambda_c)$, where $H(\bar{\lambda})$ is the entropy of the mean model and is independent of λ_c. Hence, maximizing the cross entropy with respect to the mean model is equivalent to minimizing the KL-divergence with respect to the mean model. The approximate problem (4), therefore converges to the original DML problem (3) as the size of $\bar{\mathcal{Z}}$ goes to ∞. Viewing (4) as a maximum-likelihood parameter estimation problem leads to efficient algorithms for addressing the distributed clustering and classification problems.

Algorithm 1 Distributed Clustering

Input: Set of clustering models $\{\lambda_i\}_{i=1}^n$ with weights $\{\nu_i\}_{i=1}^n$ summing to 1, Mixture model family $\mathcal{F}$.

Output: $\lambda_c^a \simeq \operatorname*{argmin}_{\lambda_c \in \mathcal{F}} \sum_{i=1}^n \nu_i D_{KL}^{\text{clus}}(\lambda_i, \lambda_c)$

Method:

1. Obtain mean model $\bar{\lambda}$ such that
$$p_{\bar{\lambda}}(x) = \sum_{i=1}^{n} \nu_i p_{\lambda_i}(x).$$
2. Generate $\bar{\mathcal{X}} = \{x_j\}_{j=1}^m$ from mean model, $\bar{\lambda}$ using MCMC sampling.
3. Apply EM algorithm to obtain the optimal model, λ_c^a, such that
$$\lambda_c^a = \operatorname*{argmax}_{\lambda_c \in \mathcal{F}} L(\bar{\mathcal{X}}, \lambda_c) = \operatorname*{argmax}_{\lambda_c \in \mathcal{F}} \frac{1}{m} \sum_{j=1}^{m} \log(p_{\lambda_c}(x_j)).$$

A. Distributed Clustering

For the clustering scenario, we address the approximate DML problem (4) using the Expectation-Maximization (EM) framework. The main idea is to first generate an unlabeled dataset $\bar{\mathcal{X}}$ following the mean model $\bar{\lambda}$, using Markov Chain Monte Carlo (MCMC) sampling techniques [9] and then, apply the EM algorithm to this dataset to obtain the clustering model $\lambda_c^a \in \mathcal{F}$ that maximizes its likelihood of being observed. The resulting model λ_c^a is a local minimizer of the approximate problem and not necessarily the same as the

[1] This result is true for a class of functions called Bregman divergences of which KL-divergence and squared L_2 distance are particular cases.

[2] Examples of distance functions that are convex in the density function of the second argument include KL-divergence, L_1 distance and squared L_2 distance.

solution λ_c^* of the original DML problem (2). However, it is guaranteed to asymptotically converge to a locally optimal solution as the size of $\bar{\mathcal{X}}$ goes to ∞. In practice, one can use multiple runs of the EM algorithm and pick the best solution among these so that the obtained model is reasonably close to the globally optimal model.

B. Distributed Classification

For the classification scenario, we obtain a similar algorithm (Algorithm 2). The only difference being that it is now possible to directly obtain the maximum likelihood estimates (MLE), without using the EM algorithm as we now have access to labeled data. As before, we generate a labeled dataset $\bar{\mathcal{X}}$ from the mean model $\bar{\lambda}$ using MCMC sampling and then, estimate the parameters of the classification model $\lambda_c^a \in \mathcal{F}$ that maximizes the likelihood of observing $\bar{\mathcal{X}}$. Usually, the approximate distributed classification problem is convex in nature, ensuring that the resulting model λ_c^a is the global minimizer, which is guaranteed to asymptotically converge to the optimal solution of the original DML problem as the size of $\bar{\mathcal{X}}$ goes to ∞.

Note that our formulation of the distributed classification problem is different from the usual formulation based on the misclassification error. However, it turns out that empirically, the most effective solution [2] for minimizing the misclassification error given a set of classification models is to obtain a combined classifier based on the mean posterior probabilities, which is the same as the mean model $\bar{\lambda}$, i.e., the unconstrained optimal solution, under the assumption that the data densities $p_{\lambda_i}(x)$ for the different classification models are the same. This assumption is not restrictive and is in fact usually true for distributed classification scenarios, e.g., bagged predictors, for which the mean posterior classifier performs well.

IV. PRIVACY COSTS

In this section, we quantify the privacy cost using ideas from information theory and also show that there is an inverse relation between the privacy of the local models and the quality of the mean model.

In order to quantify privacy, we need a measure that indicates the uncertainty in predicting the original dataset from the model. The work [1] proposes a privacy measure based on the differential entropy of the generating distribution given by $h(\lambda) = -\int_{\Omega_z} p_\lambda(z)\log_2(p_\lambda(z))dz$, where Ω_z is the domain of z. This quantity indicates the uncertainty [4] in the distribution of the model λ, but does not consider the privacy of a particular dataset with respect to a model. For example, a model with an extremely peaked distribution will have very low entropy, but if the peaks do not correspond to the actual objects in the dataset, then there is not much privacy lost. This motivates us to define a slightly different measure that considers the privacy of the model with respect to the actual objects in the dataset. We propose that the privacy, $\mathcal{P}(z,\lambda)$ of an object z given a model λ be defined in terms of the probability of generating the data object from the model. The higher the probability, the lower the privacy. More specifically, noting that the reciprocal of the probability is related to uncertainty [4], we have $\mathcal{P}(z,\lambda) = (p_\lambda(z))^{-1}$.

Algorithm 2 Distributed Classification

Input: Set of classification models $\{\lambda_i\}_{i=1}^n$ with weights $\{\nu_i\}_{i=1}^n$ summing to 1, Mixture model family $\mathcal{F}$.

Output: $\lambda_c^a \simeq \underset{\lambda_c\in\mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^n \nu_i D_{KL}^{\text{class}}(\lambda_i,\lambda_c)$

Method:

1. Obtain mean model $\bar{\lambda}$ such that
$$p_{\bar{\lambda}}(x,y) = \sum_{i=1}^{n} \nu_i p_{\lambda_i}(x,y).$$
2. Generate a labeled set $\bar{\mathcal{X}} = \{(x_j,y_j)\}_{j=1}^m$ from mean model, $\bar{\lambda}$ using MCMC sampling.
3. Apply MLE methods to obtain the optimal model, λ_c^a, such that
$$\lambda_c^a = \underset{\lambda_c\in\mathcal{F}}{\operatorname{argmax}}\ L(\bar{\mathcal{X}},\lambda_c) = \underset{\lambda_c\in\mathcal{F}}{\operatorname{argmax}}\ \frac{1}{m}\sum_{j=1}^{m}\log(p_{\lambda_c}(x_j,y_j)).$$

For vector data, $\mathcal{P}(z,\lambda) = 1$ implies that z can be predicted with the same accuracy as a random variable with a uniform distribution on a ball of unit volume. We can now define the privacy, $\mathcal{P}(\mathcal{Z},\lambda)$ of a dataset $\mathcal{Z}$ with respect to the model as some function of the privacy of the individual data objects. The geometric mean has a nice interpretation as the reciprocal of the average likelihood of the dataset being generated by the model, assuming that the individual samples are i.i.d., i.e.,

$$\mathcal{P}(\mathcal{Z},\lambda) = \left(\prod_{z\in\mathcal{Z}} p_\lambda(z)\right)^{\frac{-1}{|\mathcal{Z}|}} = 2^{\left(-\frac{1}{|\mathcal{Z}|}\sum_{z\in\mathcal{Z}}\log_2 p_\lambda(z)\right)}.$$

A higher likelihood of generating the dataset from the model implies a lower amount of privacy. For example, let us consider vector space data being modeled by a mixture of Gaussians. A highly detailed model with Gaussians of vanishing variance, centered at each of the data objects gives away the entire dataset and has no privacy. This is to be expected as the probability density $p_\lambda(z)$ goes to ∞, for all data objects $z \in \mathcal{Z}$ making the privacy measure go to 0^+. On the other hand, a very coarse model, say with a single Gaussian of high variance has a low likelihood of generating the data and hence, has a high privacy.

A. Relationship with Model Quality

Intuitively, if the local models are more detailed, the combined model can be improved at the cost of decreased privacy. In particular, there is a linear relation between the average logarithm of privacy (log-privacy) of the local models and the quality of the optimal mean model. Using the weak law of large numbers and Chebyshev inequality [10], it can be shown that the log-privacy of the local models, λ_i converges to the cross-entropy between p_{λ_i} and the true distribution, p_{λ^0}, when the size of the individual data sources tend to ∞. As a result, the average log-privacy of the local models converges to their average cross-entropy, which is linearly related to the KL-divergence between the mean model and the true model, i.e., when the sizes of the individual datasets

TABLE I

DETAILS OF GENERATIVE MODELS AND DATASETS.

Data Type	Model Type	#Dim/Seq. Length	Total Data Size (N)	#Sites
Vector	Gaussian Full-covariance	8	5000	5
Directional	von Mises-Fisher	100	5000	5
Discrete sequence	Discrete HMM 5 states 4 symbols	30	1000	5
Continuous sequence	Cont. HMM 5 states 4 mixtures	30	600	3

are large enough, then with a high probability,

$$\sum_{i=1}^{n} \nu_i \log(\mathcal{P}(\mathcal{Z}_i, \lambda_i)) + H(\lambda^0) \simeq KL(p_{\lambda^0}(z) \| p_{\bar{\lambda}}(z)) = \mathcal{Q}_I(\bar{\lambda}),$$

where $\bar{\lambda}$ is the mean model. As the privacy of the local models increases, the ideal quality cost of the mean model, which is the optimal model with no constraints, also goes up. On the other hand, when the privacy of the local models decreases, the mean model tends to be more accurate.

V. EXPERIMENTAL EVALUATION

In this section, we provide empirical evidence that for a reasonable global sample size and privacy level, the global model obtained through our approach is as good as or better than the best local model for different types of data not only in terms of KL-divergence but also for other distance measures. We also present results that show how the privacy and quality costs vary with the resolution of local models.

A. Datasets and Learning Algorithms

We performed experiments on four different types of data shown in Table V-A. Artificial data was preferred since the true generative models is known, unlike in the case of real data, and one can perform controlled experiments to better understand algorithmic properties. In order to generate the data, we chose, for each run of the experiment, a mixture model with a fixed number (=5) of components and used it to create a collection of datasets of equal size by sampling independently using MCMC techniques. These datasets were then used as the distributed data sources for training the local clustering or classification models.

We empirically found that our approach is more beneficial when the learning algorithms applied to the individual data sites are different, as this creates diversity in the models. However, since our emphasis here is not on the model selection problem, we present results obtained by applying the same learning algorithm to all the sites. For the unlabeled datasets, we used EM algorithms based on mixture models of the appropriate type. For the labeled datasets, we estimated the parameters of the class conditional densities using maximum likelihood estimation (MLE) methods. The EM algorithms at both the local and global level were run multiple times and the best solution was chosen so as to reduce the probability of getting stuck in local minima.

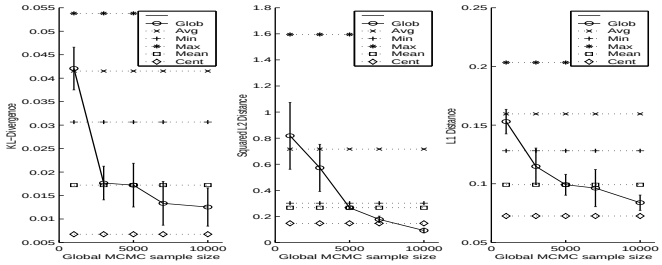

Fig. 1. Variation of global model quality with sample size in a clustering scenario.

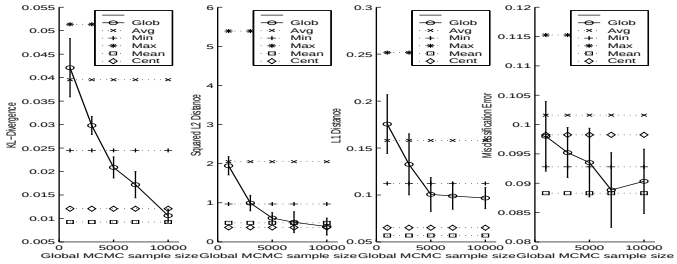

Fig. 2. Variation of global model quality with sample size in a classifi cation scenario.

B. Performance Metrics

For each setting, we computed the privacy costs of the local models and the ideal quality costs based on the various distance measures mentioned in section 2. Distance measures that are integrals were estimated by averaging over 10,000 samples drawn from the appropriate distributions. The centralized model obtained using the union of all the datasets was used as the reference for each experiment.

C. Results and Discussion

We first studied the performance of our distributed learning algorithms on the Euclidean vector datasets for different choices of global MCMC sample size and local model resolution. Based on these experiments, we chose good values for the global sample size and model resolution and applied our algorithms to different data types in both clustering and classification settings.

1) Variation of Global Model Quality with MCMC Sample size: An important step in our model-based learning approach is choosing the global MCMC sample size. Theoretical results indicate that the quality of model tends to improve as the sample size increases to ∞. In order to test this hypothesis, we ran our algorithm multiple times on the Euclidean vector datasets changing only the global sample size. Figures 1 and 2 shows how the quality of the different models vary with the sample size for in a clustering and classification scenario respectively. In both the cases, the quality of the global model improves with the number of artificially generated samples, with diminishing returns after a point. The global model quality tends to be about the same as that of the average model when the global sample size is equal to that of the individual data sources and steadily becomes better. When the sample size increases to that of the combined size of all the data sources, the global model is better than even the best of the local models.

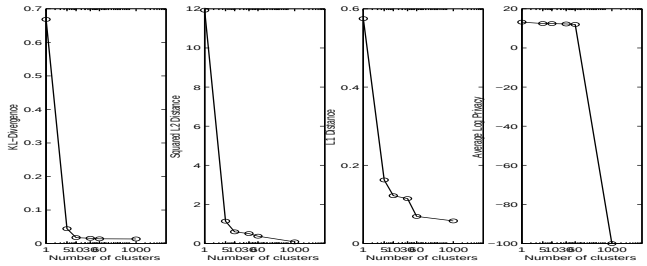

Fig. 3. Variation of privacy and cluster quality w.r.t base model resolution.

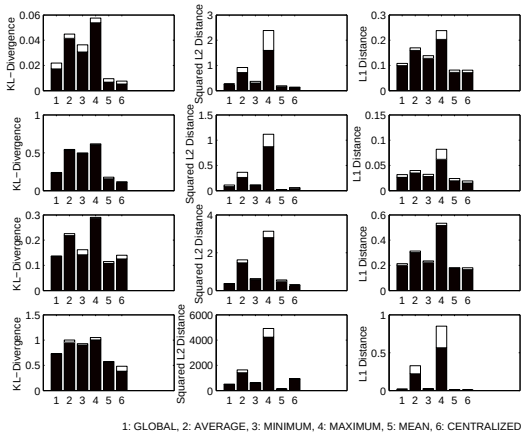

Fig. 4. Global model quality for different types of data in a clustering scenario. The rows 1-4 correspond to the results on Gaussian, directional, discrete and continuous sequence data respectively. The black bar represents the average value and the white bar represents the standard deviation.

2) Variation of Privacy and Quality cost with Model Resolution: Another significant aspect of our framework is the trade-off between privacy restrictions and the quality of the combined model obtained. This trade-off can be controlled by picking a suitable model resolution, e.g., number of clusters/classes. Figure 3 shows the variation of the average log-privacy and quality cost with the number of clusters in the local models for Euclidean vector datasets. The behavior is similar for classification settings as well. From the plots, we note that the average log-privacy as well as the quality costs decrease as the number of clusters increases. At a thousand clusters/location (i.e. one cluster per point) there is maximum loss of privacy, but because of the natural clusters in the data, comparable cluster quality can be obtained much before this limiting value, i.e., at a much lesser privacy cost.

3) Quality of Global Model for different data types: We also applied our learning algorithms to different data types to illustrate the generality of our approach. For a fair comparison, we chose the global sample size to be equal to the combined size of all the data sources and the model resolution of the local models to be the same as that of the true model. Figure 4 shows the quality of the different models for all four data types, in a clustering setting. In all the cases, the global model performs better than the best local model. Moreover, the global model quality is in general closer to the quality of the centralized model than the average quality of the local models. Similar results were obtained for the classification settings.

VI. CONCLUSION

We present a privacy preserving framework for inter-enterprise distributed data mining that is applicable to a wide variety of data types and learning algorithms, so long as they can provide a generative model. Our approach is based on obtaining a global model from "virtual samples' generated from the local models using MCMC sampling techniques. We also propose practical algorithms for distributed clustering and classification based on this approach. Theoretical results indicate that the algorithms asymptotically converge to an optimal solution while empirical results show that it is possible to obtain a high quality global model with a reasonable sample size and very little loss of privacy. Finally, we quantify privacy based on ideas from information theory and provide results that illustrate the trade-off between the privacy and the global model quality.

VII. REFERENCES

[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *PODS*, pages 247–255, 2001.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] P. Chan, S. Stolfo, and D. Wolpert. Integrating multiple learned models for improving and scaling machine learning algorithms. *Machine Learning*, 36(1-2), 1996.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[5] I. S. Dhillon and D. S. Modha. A data-clustering algorithm on distributed memory multiprocessors. In *KDD*, pages 245–260, 1999.

[6] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *KDD*, pages 217–228, 2002.

[7] E. Johnson and H. Kargupta. Collective, hierarchical clustering from distributed, heterogeneous data. In M. Zaki and C. Ho, editors, *Large-Scale Parallel KDD Systems*, pages 221–244. LNCS Vol. 1759, Springer, 1999.

[8] Y. Lindell and B. Pinkas. Privacy preserving data mining. *LNCS*, 1880:36–77, 2000.

[9] R. M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, Univ. of Toronto, 1993.

[10] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1984.

[11] P. Smyth and D. Wolpert. An evaluation of linearly combining density estimators via stacking. *Machine Learning*, 36(1/2):53–89, July 1999.

[12] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. *JMLR*, 3(3):583–617, 2002.

[13] K. Yamanishi. Distributed cooperative Bayesian learning strategies. *Information and Computation*, 150:22–56, 1998.