# Privacy-preserving Distributed Clustering using Generative Models

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

## Abstract

*We present a framework for clustering distributed data in unsupervised and semi-supervised scenarios, taking into account privacy requirements and communication costs. Rather than sharing parts of the original or perturbed data, we instead transmit the parameters of suitable generative models built at each local data site to a central location. We mathematically show that the best representative of all the data 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. We provide empirical results on different data types to highlight the generality of our framework. The results show that high quality distributed clustering can be achieved with little privacy loss and low communication cost.*

## 1. Introduction

Extracting useful knowledge from large, distributed data repositories can be a very difficult task when such data cannot be directly centralized or unified as a single file or database either due to legal, proprietary or technical restrictions. This has led to the emergence of distributed data mining techniques that try to obtain high quality information from distributed sources with minimal interactions among the data sources. Most of the techniques developed so far have focused on classification or on association rules [1, 2, 8, 13]. 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) [12, 18], 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) [7]. These techniques, however, do not specifically address privacy issues.

In this paper, we focus on the little explored problems of clustering horizontally distributed data in unsupervised and semi-supervised settings, 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 *via* clustering or classification, the entire distributed data, without actually 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.

In this paper we make three main contributions. First, we introduce a privacy preserving framework for distributed clustering in unsupervised and semi-supervised scenarios that is applicable to a wide variety of data types and learning algorithms, so long as they can provide a generative model [11]. In this framework, the parties owning the individual data sources independently train generative models on the local data and send the model parameters to a central combiner that integrates the models. This limits the amount of interactions between the data sources and the combiner and enables us to formulate the distributed clustering problem in a general as well as tractable form. Second, we present the idea that it is possible to obtain efficient solutions to optimization problems based on generative models by formulating approximate versions of the problems using sampling techniques, which can then be solved using existing learning algorithms. We apply this idea to the specific problem of distributed clustering in unsupervised and semi-supervised sce-

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE

narios to develop EM based algorithms that are guaranteed to asymptotically converge to a global model that is locally optimal as the sample size used to obtain the global model goes to $\infty$. Finally, we propose a measure for quantifying privacy based on ideas from information theory. This allows us to formalize the problem of obtaining a local model given the privacy constraints and demonstrate that there is an asymptotic relation between the average logarithm of privacy of the local models and the KL-divergence quality cost of the optimal model.

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 $\lambda$ is denoted by $p_\lambda$. 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 $x$.

## 2. Problem definition

Consider a situation wherein there are multiple data sources containing unlabeled or partially labeled data and our aim is to obtain a combined global clustering or classification model subject to privacy and communication restrictions. We will approach this distributed clustering problem by first dividing it into two sub-problems — (i) choosing local models based on privacy and communication 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 and communication 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 clustering, privacy and communication costs. This approach also allows the individual parties to use proprietary algorithms and domain knowledge, and enables reuse of legacy clusterings [18].

Let $\{\mathcal{X}_i\}_{i=1}^n$ be $n$ horizontally partitioned data sources generated by a common underlying model, $\lambda^0$ 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 and communication costs are satisfied, i.e., $\forall i,\ 1 \leq i \leq n,\ \mathcal{P}(\lambda_i) \geq \rho_i$ and $\mathcal{C}(\lambda_i) \leq c_i$, where $\mathcal{P}(\cdot)$ and $\mathcal{C}(\cdot)$ are the privacy and communication cost functions discussed later in section 5, and $\{\rho_i\}_{i=1}^n$ and $\{c_i\}_{i=1}^n$ are the lowest allowed privacy and highest allowed communication 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 Clustering** (DMC) 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 DMC problem is to obtain the optimal global clustering model $\lambda_c^*$ belonging to a given family of models $\mathcal{F}$, i.e.,

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

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

### 2.1. 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 combining classification and clustering models. In our scheme, a **classification model**, i.e., a generative model $\lambda$, 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 $\lambda$, 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.

### 2.2. Model quality

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

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 try to optimize the quality cost function based only on the KL-divergence measure 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.,

$$\begin{aligned} D_{KL}^{\text{clus}}(\lambda_1, \lambda_2) &= KL(p_{\lambda_1}(x) \| p_{\lambda_2}(x)) \\ &= \int_{\Omega_x} p_{\lambda_1}(x) \log\left(\frac{p_{\lambda_1}(x)}{p_{\lambda_2}(x)}\right) dx, \end{aligned}$$

where $\Omega_x$ is the domain of $x$, 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., $D_{KL}^{\text{class}}(\lambda_1, \lambda_2) = KL(p_{\lambda_1}(x, y) \| p_{\lambda_2}(x, y))$.

## 3. Unsupervised distributed clustering

In this section, we pose the DMC problem for an unsupervised scenario as an optimization problem and propose a practical algorithm that asymptotically converges to a locally optimal solution. The objective of the DMC problem for an unsupervised scenario is to obtain a global model $\lambda_c$ 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.,

$$\lambda_c^* = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmin}}\, \mathcal{Q}(\lambda_c) = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} \nu_i D_{KL}^{\text{clus}}(\lambda_i, \lambda_c), \quad (1)$$

where $\{\lambda_i\}_{i=1}^n$ are the local clustering models based on different unlabeled data sources with weights $\{\nu_i\}_{i=1}^n$ summing to 1. 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 $\lambda_c$,*

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

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

[1] This result is true for a class of functions called Bregman divergences [3] of which KL-divergence and squared $L_2$ distance are particular cases.

Applying the above theorem for clustering models, we can see that the cost function in (1) is equal to $\sum_{i=1}^{n} \nu_i D_{KL}^{\text{clus}}(\lambda_i, \bar{\lambda}) + D_{KL}^{\text{clus}}(\bar{\lambda}, \lambda_c)$. The first term is independent of $\lambda_c$ and hence, optimizing the cost function in (1) is equivalent to minimizing KL-divergence with respect to the mean model $\bar{\lambda}$. In the absence of constraints, the optimal solution is just the mean model $\bar{\lambda}$, as KL-divergence is always non-negative and 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 $\lambda^0$,*

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

*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 $\lambda^0$ 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 lemma, 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 improvement 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 DMC problem (1).

$$\lambda_c^* = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmin}}\, D_{KL}^{\text{clus}}(\bar{\lambda}, \lambda_c) \quad (2)$$

The new optimization problem (2) is difficult to solve directly using gradient descent techniques. Therefore, we pose an approximate version of the above problem and solve it *via* Expectation-Maximization [6]. Let $\bar{\mathcal{X}} = \{x_j\}_{j=1}^m$ be a dataset 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 $\bar{\mathcal{X}}$, i.e.,

$$\max_{\lambda_c \in \mathcal{F}} L(\bar{\mathcal{X}}, \lambda_c) = \max_{\lambda_c \in \mathcal{F}} \frac{1}{m} \sum_{j=1}^{m} \log(p_{\lambda_c}(x_j)), \quad (3)$$

[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.

**Algorithm 1** Unsupervised Distributed Clustering

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

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

**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, $\lambda_c^a$, such that

$$\lambda_c^a = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmax}}\, L(\bar{\mathcal{X}}, \lambda_c) = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmax}} \frac{1}{m} \sum_{j=1}^{m} \log(p_{\lambda_c}(x_j)).$$

where $L(\bar{\mathcal{X}}, \lambda_c)$ is the average log-likelihood of $\bar{\mathcal{X}}$ with respect to $\lambda_c$. As the size of the dataset $\bar{\mathcal{X}}$ goes to $\infty$, the average log-likelihood converges to the cross entropy between the densities $p_{\bar{\lambda}}$ and $p_{\lambda_c}$, i.e., $\underset{m\to\infty}{\text{Lt}}\, L(\bar{\mathcal{X}}, \lambda_c) = \underset{m\to\infty}{\text{Lt}}\, \mathbf{E}_{x\in\bar{\mathcal{X}}}[\log(p_{\lambda_c}(x))] = \mathbf{E}_{x\sim p_{\bar{\lambda}}}[\log(p_{\lambda_c}(x))]$. Now, the cross entropy between any two densities is linearly related to the KL-divergence between them, i.e., $\mathbf{E}_{x\sim p_{\bar{\lambda}}}[\log(p_{\lambda_c}(x))] = \mathbf{E}_{x\sim p_{\bar{\lambda}}}[\log(p_{\bar{\lambda}}(x)) - \log\left(\frac{p_{\bar{\lambda}}(x)}{p_{\lambda_c}(x)}\right)] = H(\bar{\lambda}) - D_{KL}^{\text{clus}}(\bar{\lambda}, \lambda_c)$, where $H(\bar{\lambda})$ is the entropy of the mean model and is independent of $\lambda_c$. 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 (3), therefore converges to the unsupervised DMC problem (2) as the size of $\bar{\mathcal{X}}$ goes to $\infty$.

Viewing (3) as a maximum-likelihood parameter estimation problem leads to Algorithm 1. The main idea is to first generate a dataset $\bar{\mathcal{X}}$ following the mean model $\bar{\lambda}$, using Markov Chain Monte Carlo (MCMC) sampling techniques [14] 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 $\lambda_c^a$ is a local minimizer of the approximate problem and not necessarily the same as the solution $\lambda_c^*$ of the original unsupervised DMC problem (1). However, it is guaranteed to asymptotically converge to a locally optimal solution as the size of $\bar{\mathcal{X}}$ goes to $\infty$. 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.

## 4. Semi-supervised distributed clustering

In this section, we consider the DMC problem for a semi-supervised setting of which the unsupervised and completely supervised scenarios are special cases. Then, as in the unsupervised case, we pose it as an optimization problem and present an efficient EM based algorithm to solve it.

Consider a situation where only some of the data sources have labeled data. In this case, the objective is to use the local classification models $\{\lambda_{Ai}\}_{i=1}^{n_A}$ based on labeled sources and local clustering models $\{\lambda_{Bi}\}_{i=1}^{n_B}$ based on the unlabeled data sources to obtain a global model whose components correspond to the different classes. As in the previous case, we minimize the KL-divergence of the global model from the local models leading to the optimization problem,

$$\min_{\lambda_c \in \mathcal{F}} \left\{ \sum_{i=1}^{n_A} \nu_{Ai} D_{KL}^{\text{class}}(\lambda_{Ai}, \lambda_c) + \sum_{i=1}^{n_B} \nu_{Bi} D_{KL}^{\text{clus}}(\lambda_{Bi}, \lambda_c) \right\}, \tag{4}$$

where $\mathcal{F}$ is a mixture model family and $\{\nu_{Ai}\}_{i=1}^{n_A}, \{\nu_{Bi}\}_{i=1}^{n_B}$ are the weights of the classification and clustering models respectively that together sum to 1. Applying Theorem 1 for the clustering and classification models, it is easy to see that the semi-supervised DMC problem (4) is exactly equivalent to a simpler problem,

$$\min_{\lambda_c \in \mathcal{F}} \{\nu_A D_{KL}^{\text{class}}(\bar{\lambda}_A, \lambda_c) + \nu_B D_{KL}^{\text{clus}}(\bar{\lambda}_B, \lambda_c)\}, \tag{5}$$

where $\nu_A = \sum_{i=1}^{n_A} \nu_{Ai}$, $\nu_B = \sum_{i=1}^{n_B} \nu_{Bi}$ and the models $\bar{\lambda}_A$ and $\bar{\lambda}_B$ are such that $p_{\bar{\lambda}_A}(x,y) = \frac{1}{\nu_A} \sum_{i=1}^{n_A} \nu_{Ai} p_{\lambda_{Ai}}(x,y)$ and $p_{\bar{\lambda}_B}(x) = \frac{1}{\nu_B} \sum_{i=1}^{n_B} \nu_{Bi} p_{\lambda_{Bi}}(x)$. When $\nu_A = 0$, i.e., there are no classification models, this problem reduces to the unsupervised DMC problem (2) and when $\nu_B = 0$, i.e., there are no clustering models, it reduces to a supervised distributed classification problem. For the supervised case, this formulation is different from the usual formulation based on the misclassification error. However, it turns out that empirically, the most effective solution [4] 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 exactly the same as the mean classification model $\bar{\lambda}_A$ 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.

We now address the simplified semi-supervised DMC problem (5) using the following approximate version. Let $\bar{\mathcal{X}}_A = \{(x_{Aj}, y_{Aj})\}_{j=1}^{m_A}$ be a labeled dataset sampled from the mean classification model $\bar{\lambda}_A$ and $\bar{\mathcal{X}}_B = \{x_{Bj}\}_{j=1}^{m_B}$ be an unlabeled dataset sampled from the mean clustering

**Algorithm 2** Semi-supervised Distributed Clustering

**Input:** Set of classification models $\{\lambda_{Ai}\}_{i=1}^{n_A}$ and clustering models $\{\lambda_{Bi}\}_{i=1}^{n_B}$ with weights $\{\nu_{Ai}\}_{i=1}^{n_A}$ and $\{\nu_{Bi}\}_{i=1}^{n_B}$ respectively that together sum to 1, Mixture model family $\mathcal{F}$.

**Output:** $\lambda_c^a \simeq \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmin}} \{\sum_{i=1}^{n_A} \nu_{Ai} D_{KL}^{\text{class}}(\lambda_{Ai}, \lambda_c) + \sum_{i=1}^{n_B} \nu_{Bi} D_{KL}^{\text{clus}}(\lambda_{Bi}, \lambda_c)\}$

**Method:**

1. Obtain mean classification model $\bar{\lambda}_A$ and mean clustering model $\bar{\lambda}_B$ such that

$$p_{\bar{\lambda}_A}(x,y) = \frac{1}{\nu_A}\sum_{i=1}^{n_A} \nu_{Ai} p_{\lambda_{Ai}}(x,y)$$

and

$$p_{\bar{\lambda}_B}(x) = \frac{1}{\nu_B}\sum_{i=1}^{n_B} \nu_{Bi} p_{\lambda_{Bi}}(x),$$

where $\sum_{i=1}^{n_A} \nu_{Ai} = \nu_A$ and $\sum_{i=1}^{n_B} \nu_{Bi} = \nu_B$.

2. Generate $\bar{\mathcal{X}}_A = \{(x_{Aj}, y_{Aj})\}_{j=1}^{m_A}$ and $\bar{\mathcal{X}}_B = \{x_{Bj}\}_{j=1}^{m_B}$ from the mean models, $\bar{\lambda}_A$ and $\bar{\lambda}_B$ respectively so that $\frac{m_A}{m_B} = \frac{\nu_A}{\nu_B}$ using MCMC sampling.

3. Apply the modified EM algorithm to obtain the optimal model, $\lambda_c^a$ that is the solution of

$$\underset{\lambda_c \in \mathcal{F}}{\operatorname{argmax}}\, L(\bar{\mathcal{X}}, \lambda_c) = \underset{\lambda_c \in \mathcal{F}}{\operatorname{argmax}} \{\nu_A L(\bar{\mathcal{X}}_A, \lambda_c) + \nu_B L(\bar{\mathcal{X}}_B, \lambda_c)\}.$$

model $\bar{\lambda}_B$ such that the sizes of the datasets, $m_A$ and $m_B$ are proportional to the weights $\nu_A$ and $\nu_B$ and $m_A + m_B = m$. Now consider the problem of finding a mixture model $\lambda_c^a \in \mathcal{F}$ that maximizes the average log-likelihood of the combined dataset $\bar{\mathcal{X}} = \bar{\mathcal{X}}_A \cup \bar{\mathcal{X}}_B$, i.e.,

$$\max_{\lambda_c \in \mathcal{F}} L(\bar{\mathcal{X}}, \lambda_c) = \max_{\lambda_c \in \mathcal{F}} \left\{\nu_A L(\bar{\mathcal{X}}_A, \lambda_c) + \nu_B L(\bar{\mathcal{X}}_B, \lambda_c)\right\} \tag{6}$$

where $L(\cdot, \lambda_c)$ is the average log-likelihood function with respect to $\lambda_c$. Using the same relations between the log-likelihood, cross entropy and the KL-divergence as for unsupervised DMC, it is easy to show that the solution to the approximate problem converges to the solution of the original problem (5) as the size of the dataset $\bar{\mathcal{X}}$ goes to $\infty$.

The approximate problem is again a maximum likelihood parameter estimation problem where we need to learn the parameters for the mixture model that maximizes the likelihood of the combined dataset $\bar{X} = \bar{\mathcal{X}}_A \cup \bar{\mathcal{X}}_B$. This can be easily solved using the EM framework, by assuming that the missing data is the posterior probabilities of the mixture components for only the objects in $\bar{\mathcal{X}}_B$, i.e., the unlabeled data objects [15]. Because of this, we only need to update the posterior probabilities of the unlabeled data objects in the expectation step. The maximization step remains unchanged. This results in a modified EM algorithm that can be used as part of the overall semi-supervised distributed clustering algorithm (Algorithm 2).

## 5. Privacy and communication costs

In this section, we quantify the privacy and communication costs 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.

**Privacy.** In order to quantify privacy, we need a measure that indicates the uncertainty in predicting the original dataset from the model. The work in [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 $\Omega_z$ is the domain of $z$. This quantity indicates the uncertainty in predicting the data given the model $\lambda$ [5], 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 $\lambda$ 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 [5], we have $\mathcal{P}(z, \lambda) = (p_\lambda(z))^{-1}$.

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 $\infty$, 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 with high variance has a low likelihood of generating the data and hence, has a high privacy.

Intuitively, if the local models are more detailed, the combined model can be improved at the cost of decreased privacy. In particular, using the weak law of large numbers and Chebyshev inequality [16], it can be shown that the average log-privacy of the local models converges to their average cross-entropy with a high probability when the sizes

of the individual datasets are large enough. Since the average cross entropy is linearly related to the KL-divergence between the mean model and the true model, there exists an asymptotic linear relation between the average log-privacy and ideal quality cost of the mean model, i.e., $\sum_{i=1}^{n} \nu_i \log(\mathcal{P}(\mathcal{Z}_i, \lambda_i)) + H(\lambda^0) \simeq KL(p_{\lambda^0}(z)||p_{\bar{\lambda}}(z)) = \mathcal{Q}_I(\bar{\lambda})$, 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.

**Communication cost.** To quantify the communication cost $\mathcal{C}(\lambda)$, we consider the number of bits or words required to unambiguously specify the model to the central combiner. When the generative model family is already known to the central combiner, then one needs to only consider the cost of specifying the values of the parameters. A more formal definition would be to consider the Kolmogorov complexity [5] or the minimum description length of the local model, i.e., $\mathcal{C}(\lambda) = K_v(\lambda)$.

## 6 Experimental evaluation

In this section, we provide empirical evidence that for a reasonable global sample size and privacy level and a few runs of the EM algorithm, 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, communication and quality costs vary with the resolution of local models.

We performed experiments on the four different types of data shown in Table 1. 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 and models can be downloaded from `http://www.lans.ece.utexas.edu/~srujana/gencl/data`.

We empirically found that our approach is more beneficial when the number of clusters as well as the learning algorithms applied to the individual data sites are different, as this creates diversity in the models. However, since in this work our emphasis 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 distributions 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 in order to reduce the probability of getting stuck in local minima.

**Table 1. Details of generative models and datasets for different data types.**

| Data Type | Model Type | #Dim/Seq. Length | Total Data Size (N) | #Sites | #Runs |
|---|---|---|---|---|---|
| Vector | Gaussian Full-covariance | 8 | 5000 | 5 | 10 |
| Directional | von Mises-Fisher | 100 | 5000 | 5 | 10 |
| Discrete sequence | Discrete HMM 5 states 4 symbols | 30 | 1000 | 5 | 5 |
| Continuous sequence | Cont. HMM 5 states 4 mixtures | 30 | 600 | 3 | 5 |

For each setting, we computed the privacy and communication costs of the local models and the ideal quality cost functions based on the various distance measures listed 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.

### 6.1 Results and discussion

We applied our algorithm to different types of data in both unsupervised and semi-supervised settings choosing the global MCMC sample size to be equal to the combined size of all the data sources and the local model resolution to be the same as that of the true model. We also studied how the quality of the global model varies with the global sample size, the resolution of the local models and the percentage of labeled data by performing experiments on the Euclidean vector datasets.

**Quality of global model.** Figure 1 shows the quality of the different models for all four data types, in a *fully unsupervised* setting. 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. 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. Figure 2 shows the quality of the different models in a *semi-supervised* setting.

The mean model in this setting is the mean classification model obtained by combining only the local classification models. Once again, the global model provides better quality than any of the local classification models. Sometimes, it is even better than that of the mean classification model,

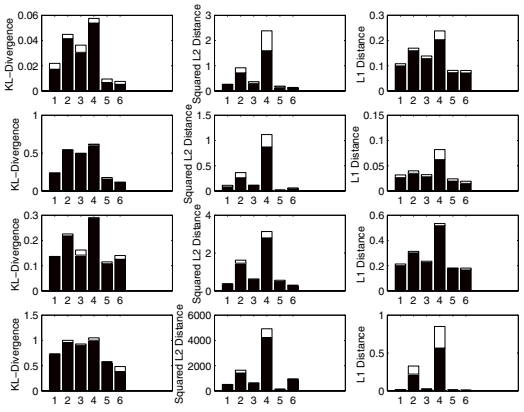


**Figure 1. Global model quality for different types of data in an unsupervised setting.**

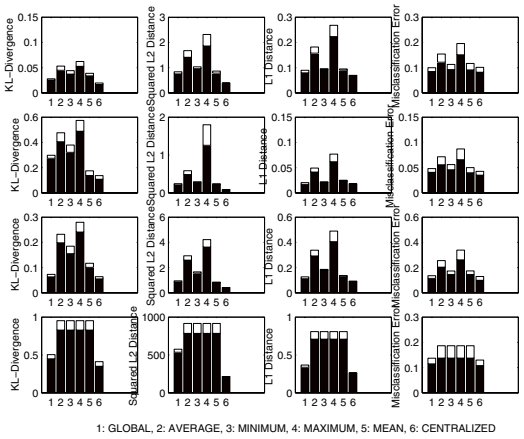


**Figure 2. Global model quality for different types of data in a semi-supervised setting.**

underscoring the effectiveness of using unlabeled data for improving the performance of classification models.

**Variation of global model quality with sample size.** For a fair comparison, we chose the global sample size to be equal to the combined size of all the data sources for the previous experiments. However, theoretical results indicate that we can obtain a better quality model with a higher sample size. In order to test this hypothesis, we ran our algorithm multiple times on the Euclidean vector datasets changing only the global sample size. Figure 3 shows how the quality of the different models vary with the sample size in an unsupervised setting. As one may expect, the quality of the global model improves with the number of artificially generated samples, with diminishing returns after a point. The behavior is similar for semi-supervised settings as well.

**Variation of privacy, communication and quality cost with model resolution.** An important aspect of the our clustering framework is the trade-off between privacy, communication 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 4 shows the variation of the average log-privacy, communication and quality cost with the number of clusters in the local models for Euclidean vector datasets. The behavior is similar for semi-supervised 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, while the communication cost goes up. 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.

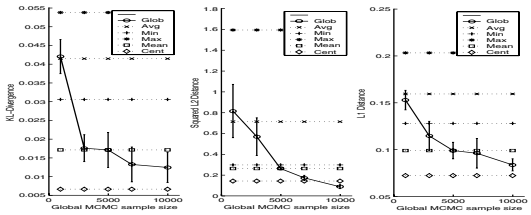


**Figure 3. Variation of global model quality with sample size.**

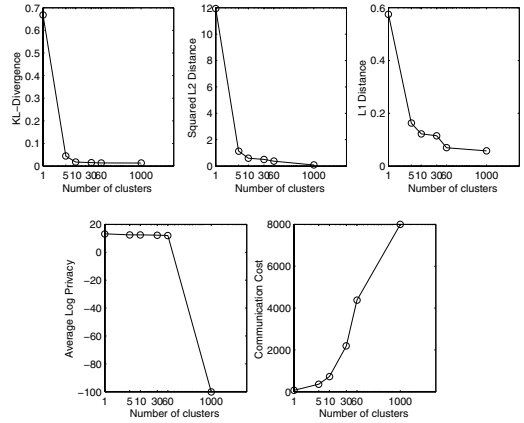


**Figure 4. Variation of privacy, cluster quality and communication cost with respect to base model resolution.**

**Variation in model quality with percentage of labeled data.** Figure 5 shows the quality of the models obtained using different number of local classification models on Euclidean vector data, i.e., different percentages of labeled data. From the figure, we note that the quality costs of the mean classification model as well as global model decrease as the number of classification models increases. Another interesting trend is that the global model performs better than the mean classification model when the percentage of labeled data is less but becomes relatively worse as the per-

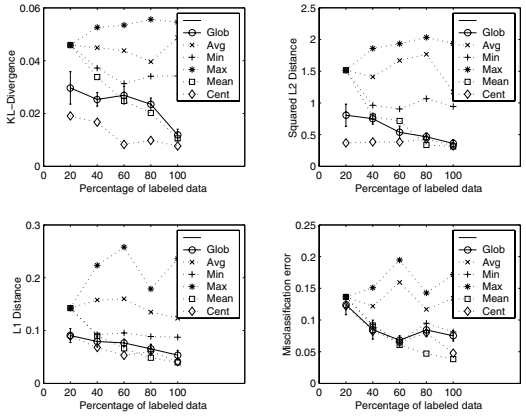


**Figure 5. Variation in the model quality with percentage of labeled data.**

centage of labeled data increases. This shows that it might be beneficial to use unlabeled data for improving classification models when there is very little labeled data. On the other hand, there is little utility in using unlabeled data when there is significant amount of labeled data.

## 7 Related work

Our distributed clustering technique relies on combining multiple parametric models. Other works of similar flavor applied to different settings include stacking for density estimation [17], distributed cooperative Bayesian learning [19]. However, in all these cases the emphasis is on quality and robustness rather than interpretability or privacy.

A simple example of integrating multiple generative models for clustering is the combining of the sets of means obtained through multiple $k$-means solutions. This has been studied in a variety of settings [9, 10], all of which are restricted to vector data. In contrast, our framework applies to arbitrary generative clustering models, hence covering a wide range of complex data types encountered in data mining.

In works that focus on privacy-preserving data mining, often individual records or attributes are subjected to a "privacy preserving" transformation and the goal is to obtain useful information from such transformed data. Classification and association rule techniques for this scenario have been proposed in [1, 2, 8]. These approaches are also restricted to vector data because of an add operator requirement. Another setting is an inter-enterprise data mining scenario such as the one considered in this paper, where multiple parties with confidential databases want to apply data mining algorithms to the union of their databases. There is very little literature in this area, a notable exception being the cryptographic method for enabling a secure two party computation for performing the ID3 decision tree algorithm in [13].

**Acknowledgments** This work was supported in part by NSF grants IDM-0307792 and ITR-0312471. We would also like to thank Arindam Banerjee and Ravi Koku for their helpful suggestions.

## References

[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Symposium on Principles of Database Systems*, 2001.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM SIGMOD*, pages 439–450, 2000.

[3] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

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

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

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[7] I. S. Dhillon and D. S. Modha. A data-clustering algorithm on distributed memory multiprocessors. In *ACM SIGKDD*, 1999.

[8] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *KDD*, 2002.

[9] U. M. Fayyad, C. Reina, and P. S. Bradley. Initialization of iterative refinement clustering algorithms. In *ICML*, pages 194–198, 1998.

[10] A. L. N. Fred and A. K. Jain. Data clustering using evidence accumulation. In *ICPR*, pages IV:276–280, 2002.

[11] J. Ghosh. Scalable clustering methods for data mining. In N. Ye, editor, *Handbook of Data Mining*, pages 247–277. Lawrence Erlbaum, 2003.

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

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

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

[15] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

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

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

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

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