2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

A Privacy-Aware Bayesian Approach for Combining Classifier and Cluster Ensembles

Ayan Acharya[1], Eduardo R. Hruschka[1,2], and Joydeep Ghosh[1]

[1]University of Texas (UT) at Austin, USA
[2]University of Sao Paulo (USP) at Sao Carlos, Brazil

***Abstract*—This paper introduces a privacy-aware Bayesian approach that combines ensembles of classifiers and clusterers to perform semi-supervised and transductive learning. We consider scenarios where instances and their classification/clustering results are distributed across different data sites and have sharing restrictions. As a special case, the privacy aware computation of the model when instances of the target data are distributed across different data sites, is also discussed. Experimental results show that the proposed approach can provide good classification accuracies while adhering to the data/model sharing constraints.**

I. 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 due to a variety of constraints. Recently, there has been an emphasis on how to obtain high quality information from distributed sources via statistical modeling while simultaneously adhering to restrictions on the *nature* of the data or models to be shared, due to data ownership or privacy issues. Much of this work has appeared under the moniker of "privacy-preserving data mining".

Three of the most popular approaches to privacy-preserving data mining techniques are: (i) query restriction to solve the inference problem in databases [10] (ii) subjecting individual records or attributes to a "privacy preserving" randomization operation and subsequent recovery of the original data [3], (iii) using cryptographic techniques for secure two-party or multi-party communications [17]. Meanwhile, the notion of privacy has expanded substantially over the years. Approaches such as k-anonymity and l-diversity [14] focused on privacy in terms of indistinguishableness of one record from others under allowable queries. More recent approaches such as differential privacy [8] tie the notion of privacy to its impact on a statistical model.

The larger body of distributed data mining techniques developed so far have focused on simple classification/clustering algorithms or on mining association rules [2], [5], [9], [13]. Allowable data partitioning is also limited, typically to *vertically partitioned* or *horizontally partitioned* data [7]. These techniques typically do not specifically address privacy issues, other than through encryption [19]. This is also true of earlier, data-parallel methods [7] that are susceptible to privacy breaches, and also need a central planner that dictates what algorithm runs on each site. In this paper, we introduce a privacy-aware Bayesian approach that combines ensembles of classifiers and clusterers and is effective for both semi-supervised and transductive learning. As far as we know, this topic has not been addressed in the literature.

The combination of multiple classifiers to generate an ensemble has been proven to be more useful compared to the use of individual classifiers [16]. Analogously, several research efforts have shown that cluster ensembles can improve the quality of results as compared to a single clusterer — *e.g.*, see [20] and references therein. Most of the motivations for combining ensembles of classifiers and clusterers are similar to those that hold for the standalone use of either classifier or cluster ensembles. However, some additional nice properties can emerge from such a combination. For instance, unsupervised models can provide supplementary constraints for classifying new data and thereby improve the generalization capability of the resulting classifier. Having this motivation in mind, a Bayesian approach to combine cluster and classifier ensembles in a privacy-aware setting is presented. We consider that a collection of instances and their clustering/classification algorithms reside in different data sites.

The idea of combining classification and clustering models has been introduced in the algorithms described in [11], [1]. However, these algorithms do not deal with privacy issues. Our probabilistic framework provides an alternative approach to combining class labels with cluster labels under conditions where sharing of individual records across data sites is not permitted. This soft probabilistic notion of privacy, based on a quantifiable information-theoretic formulation, has been discussed in detail in [15].

II. BC^{3}E Framework

A. Overview

Consider that a classifier ensemble previously induced from training data is employed to generate a set of class labels for every instance in the target data. Also, a cluster ensemble is applied to the target data to provide sets of cluster labels. These class/cluster labels provide the inputs to Bayesian Combination of Classifier and Cluster Ensembles (**BC^{3}E**) algorithm.

B. Generative Model

Consider a target set $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ formed by N unlabeled instances. Suppose that a classifier ensemble composed of r_1

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

IEEE computer society

classification models has produced r_1 class labels (not necessarily different) for every instance $\mathbf{x}_n \in \mathcal{X}$. Similarly, consider that a cluster ensemble comprised of r_2 clustering algorithms has generated cluster labels for every instance in the target set. Note that the cluster labeled as *1* in a given data partition may not align with the cluster numbered *1* in another partition, and none of these clusters may correspond to class *1*. Given the class and cluster labels, the objective is to come up with refined class probability distributions $\{\boldsymbol{\theta}_n\}_{n=1}^N$ of the target set instances. To that end, assume that there are k classes, which are denoted by $C = \{C_i\}_{i=1}^k$. The observed class and cluster labels are denoted by $\boldsymbol{X} = \{\{w_{1nl}\}, \{w_{2nm}\}\}$ where w_{1nl} is the class label of the n^{th} instance for the l^{th} classifier and w_{2nm} is the cluster label assigned to the n^{th} instance by the m^{th} clusterer. A generative model is proposed to explain the observations $\boldsymbol{X}$, where each instance $\mathbf{x}_n$ has an underlying mixed-membership to the k different classes. Let $\boldsymbol{\theta}_n$ denote the latent mixed-membership vector for $\mathbf{x}_n$. It is assumed that $\boldsymbol{\theta}_n$ – a discrete probability distribution over the k classes – is sampled from a Dirichlet distribution, with parameter $\boldsymbol{\alpha}$. Also, for the k classes (indexed by i) and r_2 different base clusterings (indexed by m), we assume a multinomial distribution $\boldsymbol{\beta}_{mi}$ over the cluster labels. If the m^{th} base clustering has $k^{(m)}$ clusters, $\boldsymbol{\beta}_{mi}$ is of dimension $k^{(m)}$ and $\sum_{j=1}^{k^{(m)}} \boldsymbol{\beta}_{mij} = 1$. The generative model can be summarized as follows. For each $\mathbf{x}_n \in \mathcal{X}$:

1) Choose $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha})$.
2) $\forall l \in \{1, 2, \cdots, r_1\}$, choose $w_{1nl} \sim \text{multinomial}(\boldsymbol{\theta}_n)$.
3) $\forall m \in \{1, 2, \cdots, r_2\}$.
   a) Choose $\boldsymbol{z}_{nm} \sim \text{multinomial}(\boldsymbol{\theta}_n)$ where $\boldsymbol{z}_{nm}$ is a vector of dimension k with only one component being unity and others being zero.
   b) Choose $w_{2nm} \sim \text{multinomial}(\boldsymbol{\beta}_{r\boldsymbol{z}_{nm}})$.

If the n^{th} instance is sampled from the i^{th} class in the m^{th} base clustering (implying $z_{nmi} = 1$), then its cluster label will be sampled from the multinomial distribution $\boldsymbol{\beta}_{mi}$. Modeling of the classification results from r_1 different classifiers for the n^{th} instance is straightforward: the observed class labels ($\{w_{1nl}\}$) are assumed to be sampled from the latent mixed-membership vector $\boldsymbol{\theta}_n$. In essence, the posteriors of $\{\boldsymbol{\theta}_n\}$ are expected to get more accurate in an effort to explain both classification and clustering results (*i.e.* $\boldsymbol{X}$) in the same framework. **BC^{3}E** derives its inspiration from the mixed-membership naïve Bayes model [18].

To address the log-likelihood function of **BC^{3}E**, let us denote the set of hidden variables by $\boldsymbol{Z} = \{\{\boldsymbol{z}_{nm}\}, \{\boldsymbol{\theta}_n\}\}$. The model parameters can conveniently be represented by $\boldsymbol{\zeta}_0 = \{\boldsymbol{\alpha}, \{\boldsymbol{\beta}_{mi}\}\}$. Therefore, the joint distribution of the hidden and observed variables can be written as:

$$p(\mathbf{X}, \boldsymbol{Z}|\boldsymbol{\zeta}_0) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_n|\boldsymbol{\alpha}) \prod_{l=1}^{r_1} p(w_{1nl}|\boldsymbol{\theta}_n) \prod_{m=1}^{r_2} p(\boldsymbol{z}_{nm}|\boldsymbol{\theta}_n) p(w_{2nm}|\boldsymbol{\beta}, \boldsymbol{z}_{nm}) \quad (1)$$

In theory, inference and estimation with the proposed model could be performed by maximizing the log-likelihood in Eq. (1) – using the *Expectation Maximization* family of algorithms [6]. However, the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ makes the exact computation in the summation over the classes intractable in general [4]. Therefore, inference and estimation is performed using Variational Expectation Maximization (**VEM**) [12].

C. Approximate Inference and Estimation

1) Inference: To obtain a tractable lower bound on the observed log-likelihood, we specify a fully factorized distribution to approximate the true posterior of the hidden variables:

$$q(\boldsymbol{Z}|\{\boldsymbol{\zeta}_n\}_{n=1}^N) = \prod_{n=1}^{N} q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n) \prod_{m=1}^{r_2} q(\boldsymbol{z}_{nm}|\boldsymbol{\phi}_{nm}) \quad (2)$$

where $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\gamma}_n)$ $\forall n \in \{1, 2, \cdots, N\}$, $\boldsymbol{z}_{nm} \sim \text{multinomial}(\boldsymbol{\phi}_{nm})$ $\forall n \in \{1, 2, \cdots, N\}$ and $\forall m \in \{1, 2, \cdots, r_2\}$, and $\boldsymbol{\zeta}_n = \{\boldsymbol{\gamma}_n, \{\boldsymbol{\phi}_{nm}\}\}$, which is the set of variational parameters corresponding to the n^{th} instance. Further, $\boldsymbol{\alpha} = (\alpha_i)_{i=1}^k$, $\boldsymbol{\gamma}_n = (\gamma_{ni})_{i=1}^k$ $\forall n$, and $\boldsymbol{\phi}_{nm} = (\phi_{nmi})_{i=1}^k$ $\forall n, m$; where the components of the corresponding vectors are made explicit. Using Jensen's inequality, a lower bound on the observed log-likelihood can be derived:

$$\begin{aligned} \log[p(\boldsymbol{X}|\boldsymbol{\zeta}_0)] &\geq \mathbf{E}_{q(\boldsymbol{Z})}[\log[p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\zeta}_0)]] + H(q(\boldsymbol{Z})) \\ &= \mathcal{L}(q(\boldsymbol{Z})) \end{aligned} \quad (3)$$

where $H(q(\boldsymbol{Z})) = -\mathbf{E}_{q(\boldsymbol{Z})}[\log[q(\boldsymbol{Z})]]$ is the entropy of the variational distribution $q(\boldsymbol{Z})$, and $\mathbf{E}_{q(\boldsymbol{Z})}[.]$ is the expectation w.r.t $q(\boldsymbol{Z})$. It turns out that the inequality in (3) is due to the non-negative KL divergence between $q(\boldsymbol{Z}|\{\boldsymbol{\zeta}_n\})$ and $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\zeta}_0)$ – the true posterior of the hidden variables. Let $\mathcal{Q}$ be the set of all distributions having a fully factorized form as given in (2). The optimal distribution that produces the tightest possible lower bound $\mathcal{L}$ is thus given by:

$$q^* = \arg\min_{q \in \mathcal{Q}} \text{KL}(p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\zeta}_0)||q(\boldsymbol{Z})). \quad (4)$$

The optimal value of ϕ_{nmi} that satisfies (4) is given by

$$\phi_{nmi}^* \propto \exp(\psi(\gamma_{ni})) \prod_{j=1}^{k^{(m)}} \beta_{mij}^{w_{2nmj}} \ \forall n, m, i, \quad (5)$$

where, $w_{2nmj} = 1$ if the cluster label of the n^{th} instance in the m^{th} clustering is j and $w_{2nmj} = 0$ otherwise. Since $\boldsymbol{\phi}_{nm}$ is a multinomial distribution, the updated values of the k components should be normalized to unity. Similarly, the optimal value of $\{\gamma_{ni}\}$ that satisfies (4) is given by:

$$\gamma_{ni}^* = \alpha_i + \sum_{l=1}^{r_1} w_{1nli} + \sum_{m=1}^{r_2} \phi_{nmi} \quad (6)$$

Note that the optimal values of $\boldsymbol{\phi}_{nm}$ depend on $\boldsymbol{\gamma_n}$ and vice-versa. Therefore, iterative optimization is adopted to minimize the lower bound till convergence is achieved.

2) Estimation: For estimation, we maximize the optimized lower bound obtained from the variational inference w.r.t the free model parameters $\boldsymbol{\zeta}_0$ (by keeping the variational parameters fixed). Taking the partial derivative of the lower bound w.r.t $\boldsymbol{\beta}_{mi}$ we have:

$$\beta^*_{mij} \propto \sum_{n=1}^{N} \phi_{nmi} w_{2nmj} \ \forall j \in 1, 2, \cdots, k \tag{7}$$

Again, since $\boldsymbol{\beta}_{mi}$ is a multinomial distribution, the updated values of $k^{(m)}$ components should be normalized to unity. However, no direct analytic form of update exists for $\boldsymbol{\alpha}$, and a numeric optimization method has to be resorted to. The part of the objective function that depends on $\boldsymbol{\alpha}$ is given by:

$$\begin{aligned} \mathcal{L}_{[\alpha]} &= N \left[\sum_{i=1}^{k} \log(\Gamma(\alpha_i)) - \log(\Gamma(\sum_{i=1}^{k} \alpha_i)) \right] \\ &+ \sum_{n=1}^{N} \sum_{i=1}^{k} \left[\psi(\gamma_{ni}) - \psi(\sum_{i=1}^{k} \gamma_{ni}) \right] (\alpha_i - 1) \end{aligned} \tag{8}$$

Note that the optimization has to be performed with the constraint $\boldsymbol{\alpha} \geq \mathbf{0}$. Once the optimization in M-step is done, E-step starts and the iterative update is continued till convergence.

III. Privacy Aware Computation

Inference and estimation using **VEM** allows performing computation without explicitly revealing the class/cluster labels. One can visualize instances, along with their class/cluster labels, arranged in a matrix form so that each data site contains a subset of the matrix entries. Depending on how the matrix entries are distributed across different sites, three scenarios can arise – i) *Row Distributed Ensemble*, ii) *Column Distributed Ensemble*, and iii) *Arbitrarily Distributed Ensemble*.

A. Row Distributed Ensemble

In the row distributed ensemble framework, the target set $\mathcal{X}$ is partitioned into D different subsets, which are assumed to be at different locations. The instances from subset d are denoted by $\mathcal{X}_d$, so that $\mathcal{X} = \cup_{d=1}^{D} \mathcal{X}_d$. It is assumed that class and cluster labels are available – *i.e.*, they have already been generated by some classification and clustering algorithms. The objective is to refine the class probability distributions (obtained from the classifiers) of the instances from $\mathcal{X}$ without sharing the class/cluster labels across the data sites.

A careful look at the E-step – Equations (5) and (6) – reveals that the update of the variational parameters corresponding to each instance in a given iteration is independent of those of other instances given the model parameters from the previous iteration. This suggests that we can maintain a client-server based framework, where the server only updates the model parameters (in the M-step) and the clients (corresponding to individual data sites) update the variational parameters of the instances in the E-step. For instance, consider a situation (shown in Fig. 1) where a target dataset $\mathcal{X}$ is partitioned into two subsets, $\mathcal{X}_1$ and $\mathcal{X}_2$, and that these subsets are located in two different data sites. The data site 1 has access to $\mathcal{X}_1$ and accordingly, to the respective class and cluster labels of their instances. Similarly, the data site 2 has access to the instances of $\mathcal{X}_2$ and their class/cluster labels.

Now, data site 1 can update the variational parameters $\{\boldsymbol{\zeta}_n\}$ $\forall \mathbf{x}_n \in \mathcal{X}_1$. Similarly, data site 2 can update the variational parameters $\{\boldsymbol{\zeta}_n\}$ $\forall \mathbf{x}_n \in \mathcal{X}_2$. Once the variational parameters are updated in the E-step, the server gathers information from the two sites and updates the model parameters. Here, the primary requirement is that the class and cluster labels of instances from different data sites should not be available to the server. Now, Eq. (7) can be broken as follows:

$$\beta_{mij}{}^* \propto \sum_{x_n \in \mathcal{X}_1} \phi_{nmi} w_{2nmj} + \sum_{x_n \in \mathcal{X}_2} \phi_{nmi} w_{2nmj} \tag{9}$$

The first and second terms can be calculated in data sites 1 and 2, separately, and then sent to the server, where the two terms can be added and β_{mij} can get updated $\forall m, i, j$. The variational parameters $\{\phi_{nmj}\}$ are not available to the sever and thus only some aggregated information about the values of $\{w_{2nm}\}$ for some $\mathbf{x}_n \in \mathcal{X}$ is sent to the server. We also observe that more the number of instances in a given data site, more difficult it becomes to retrieve the cluster labels (*i.e.* $\{w_{2nm}\}$) from individual clients. Also, in practice, the server does not get to know how many instances are present per data site which only makes the recovery of cluster labels even more difficult. Also note that the approach adopted only splits a central computation in multiple tasks based on how the data is distributed. Therefore, the performance of the proposed model with all data in a single place should always be the same as the performance with distributed data assuming there is no information loss in data transmission from one node to another.

In summary, the server, after updating $\boldsymbol{\zeta}_0$ in the M-step, sends them out to the individual clients. The clients, after updating the variational parameters in the E-step, send some partial summation results in the form shown in Eq. (9) to the server. The server node is helpful for the conceptual understanding of the parameter update and sharing procedures. In practice, however, there is no real need for a server. Any of the client nodes can itself take the place of server, provided that the computations are carried out in separate time windows and in proper order.

B. Column and Arbitrarily Distributed Ensemble

The column and arbitrarily distributed ensembles are illustrated in Figs. 2 and 3 respectively. Analogous distributed inference and estimation frameworks can be derived in these two cases without sharing the cluster/class labels among different data sites. However, detailed discussion is avoided due to space constraints.

IV. Experimental Evaluation

We have already shown, theoretically, that the classification results obtained by the privacy-aware **BC^{3}E** are precisely the same as those we would have gotten if all the information originally distributed across different data sites were available at a single data site. Therefore, we assess the learning

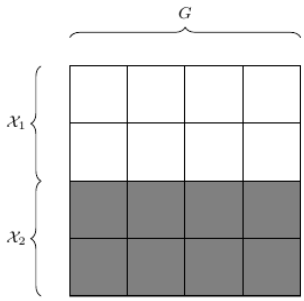

Fig. 1. Row Distributed Ensemble

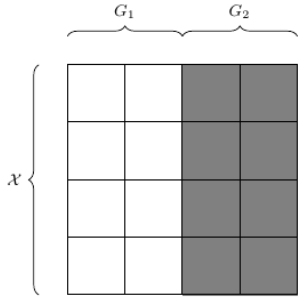

Fig. 2. Column Distributed Ensemble

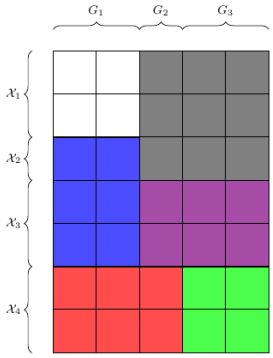

Fig. 3. Arbitrarily Distributed Ensemble

Dataset	Best Component	Classifier Ensemble	C^{3}E	BGCM	BC^{3}E
Halfmoon (2%)	93.02 ± 0.81	92.53 ± 1.83	$\mathbf{99.64 \pm 0.08}$	99.58 ± 0.08	99.37 ± 1.57
German Numer (10%)	68.20 ± 1.06	67.70 ± 1.34	$\mathbf{70.77 \pm 0.26}$	67.83 ± 1.74	70.30 ± 1.17
Heart (7%)	72.91 ± 4.60	71.26 ± 6.18	83.54 ± 3.79	$\mathbf{84.54 \pm 2.63}$	82.96 ± 6.35
Indian Pima (5%)	70.65 ± 2.77	69.95 ± 2.48	74.75 ± 2.66	$\mathbf{75.86 \pm 1.25}$	75.29 ± 2.23
Wine (10%)	84.32 ± 5.62	83.84 ± 6.29	$\mathbf{88.79 \pm 3.52}$	85.43 ± 5.23	88.60 ± 3.82

TABLE I
EXPERIMENTAL EVALUATION OF BC^{3}E

capabilities of **BC^{3}E** using five benchmark datasets (*Heart*, *German Numer*, *Halfmoon*, *Wine*, and *Pima Indians Diabetes*) — all stored in a single location. Semi-supervised approaches are most useful when labeled data is limited, while these benchmarks were created for evaluating supervised methods. Therefore, we use only small portions (from 2% to 10%) of the training data to build classifier ensembles. The remaining data is used as a target set — with the labels removed. We adopt 3 classifiers (Decision Tree, Generalized Logistic Regression, and Linear Discriminant). For clustering, we use hierarchical single-link and k-means algorithms. The achieved results are presented in Table I, where *Best Component* indicates the accuracy of the best classifier of the ensemble. We also compare **BC^{3}E** with two related algorithms (**C^{3}E** [1] and **BGCM** [11]) that do not deal with privacy issues. One can observe that, besides having the privacy-preserving property, **BC^{3}E** presents competitive accuracies with respect to their counterparts. Indeed, the Friedman test, followed by the Nemenyi post-hoc test for pairwise comparisons between algorithms, shows that there is no significant statistical difference ($\alpha = 10\%$) among the accuracies of **BC^{3}E**, **C^{3}E**, and **BGCM**.

V. EXTENSION AND FUTURE WORK

The results achieved so far motivate us to employ soft classification and clustering. Applications of **BC^{3}E** to real-world transfer learning problems are also in order.

ACKNOWLEDGMENTS

This work was supported by NSF (IIS-0713142 and IIS-1016614) and by the Brazilian Agencies FAPESP and CNPq.

REFERENCES

[1] A. Acharya, E. R. Hruschka, J. Ghosh, and S. Acharyya. C^{3}E: A Framework for Combining Ensembles of Classifiers and Clusterers. In *10th Int. Workshop on MCS, Vol. 6713, Springer*, pages 86–95, 2011.

[2] 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.

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

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] P. Chan, S. Stolfo, and D. Wolpert (Organizers). Integrating multiple learned models. *Workshop with AAAI'96*, 1996.

[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 *Proc. Large-scale Parallel KDD Systems Workshop, ACM SIGKDD*, August 1999.

[8] C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC*, pages 371–380, 2009.

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

[10] C. Farkas and S. Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2002.

[11] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Proc. of NIPS*, pages 1–9, 2009.

[12] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

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

[14] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[15] S. Merugu and J. Ghosh. Privacy perserving distributed clustering using generative models. In *Proc. of ICDM*, pages 211–218, Nov, 2003.

[16] N. C. Oza and K. Tumer. Classifier ensembles: Select real-world applications. *Inf. Fusion*, 9:4–20, January 2008.

[17] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explorations*, 4(2):12–19, 2002.

[18] H. Shan and A. Banerjee. Mixed-membership naive bayes models. *Data Min. Knowl. Discov.*, 23:1–62, July 2011.

[19] J. Vaidya and C. Clifton. Privacy-perserving k-means clustering over vertically patitioned data. In *KDD*, pages 206–215, 2003.

[20] H. Wang, H. Shan, and A. Banerjee. Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, 1:1–17, January 2011.