Mining Local Data Sources For Learning Global Cluster Models Via Local Model Exchange

Xiao-Feng Zhang, Chak-Man Lam, William K. Cheung, *Member, IEEE*

***Abstract*— Distributed data mining has recently caught a lot of attention as there are many cases where pooling distributed data for mining is probibited, due to either huge data volume or data privacy. In this paper, we addressed the issue of learning a global cluster model, known as the latent class model, by mining distributed data sources. Most of the existing model learning algorithms (e.g., EM) require access to all the available training data. Instead, we studied a methodology based on periodic model exchange and merge, and applied it to Web structure modeling. In addition, we have tested a number of variations of the basic idea, including confining the exchange to some privacy friendly parameters and varying the number of distributed sources. Experimental results show that the proposed distributed learning scheme is effective with accuracy close to the case with all the data physically shared for the learning. Also, our results show empirically that sharing less model parameters as a further mechanism for privacy control does not result in significant performance degradation for our application.**

***Index Terms*— Distributed data mining, model-based learning, latent class model, privacy preservation**

I. Introduction

Most of the machine learning and data mining algorithms work with a rather basic assumption that all the training data can be pooled together in a centralized data repository. Recently, there exist a growing number of cases that the data have to be physically distributed due to some constraints. Examples include the data privacy concern in commercial enterprises where customers' private information are supposed not to be disclosed to other parties without their consent. Another example is mining individuals' incoming e-mails for some global patterns of junk mails, and sharing personal emails with others is a scenario which is almost impossible. Additional relevant examples including distributed medical data analysis, intrusion detection, data fusion in sensor networks, etc.[9] This calls for a lot of recent research interest on distributed machine learning and data mining [7].

A common methodology for distributed machine learning and data mining is of two-stage type — first performing local data analysis and then combining the local results forming the global one. For example, in [10], a meta-learning process was proposed as an additional learning process for combining a set of locally learned classifiers (decision trees in particular) for a global classifier. A related implementation has been realized under a Grid platform known as the Knowledge Grid [11]. In [9], Kargupta *et al.* proposed what they called collective data mining and the distributed data are assumed to possess different sets of features, each being considered as an orthogonal basis. The orthogonal bases are then combined to give the overall result. They have applied it to learning Bayesian Networks for Web log analysis [12], [8].

Regarding incorporation of local data privacy control in distributed data mining, Clifton *et al.* [13], [14], [15] and Du *et al.* [16], [17], [18] have proposed solutions to distributed association rules mining with privacy preserving capability. Under the premise that parties prefer to share the local data mining results instead of the original local data, each party site learns and disclose only their local patterns, which will eventually be aggregated together to form some global patterns. Other than taking associated rule mining, Merugu *et al.* [19], [20], [21], [22], [23] works on mining global clusters (in the form of Gaussian mixture model) of high dimension feature vectors which are distributed in different sites. Their proposed method starts with creating local cluster models and then resampling from the combined models "virtual" global samples for training the global model. A quantitative data privacy measure was proposed and they pointed out that some trade-off between the global model accuracy and local data privacy has to be made.

All the aforementoned methods adopt the two-stage methodology for distributed data mining. The instrinsic limitation is that patterns which emerge only when the local data are aggregated cannot be discovered at all. In this paper, instead of taking the two-stage methodology, we propose to allow the local data mining stage and the result combining stage to interleave. In particular, we choose the latent class model as an example, where the iterative expectation and minimization algorithm is typically used for estimating the model parameters based on some training data. We learn local latent class models based on the local data but allow the immediately learned model parameters to be exchanged. For merging the exchange models which are supposed to be heterogeneous, relative entropy is used as the measure for aligning, and thus merging, of the local latent classes. The main rationale of the proposed methodology lies on the conjecture that periodic sharing of intermediate local analysis results can reduce the biases due to the local data and thus help learn a more accurate global model. For performance evaluation, experiments on applying the proposed methodology to Web cluster analysis using both Web contents and links have been conducted where the WebKB dataset is used for benchmarking. A few variations of the proposed methodology have also been proposed by considering the situation that a higher level of privacy is required as well as that the degree of data distribution is different. We found that the proposed periodic model exchange methodoloy can achieve an global model accuracy higher than the case using the two-stage methodology, and sometimes can even

[1] William K. Cheung is with Hong Kong Baptist University, Hong Kong.

outperform the situation with all the data physically pooled together for the model learning. While the gain is due to the additional communication effort, we also provide the computational complexity and the communication cost analysis for comparing different model exchange settings.

The remaining of the paper is organized as follow. Section 2 describes a particular latent class model for modeling hyperlinked Web pages. Section 3 explains how the proposed periodic model-exchange methodology can be applied to the distributed model learning. Also, the computational complexity as well as the communication overhead involved are analyzed. Details about the experimental setup for evaluating the different variations of the basic idea as well as the corresponding results can be found in Section 4. Section 5 concludes the paper and proposes some possible future research directions.

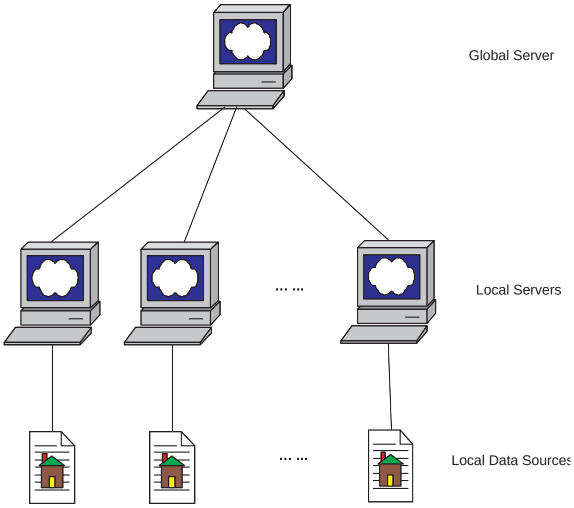

Fig. 1. A senario with a single global server mediating multiple physically distributed local servers.

II. Latent class models and Web structure analysis

The latent class model (LCM) is a statistical model under the family of mixture models. It has been adopted for modeling the co-occurence of multiple random variables with applications to a number of areas. A particular latent class model for analyzing Web contents and Web links was proposed in [2], which can be considered as a joint model of two related latent class models called PLSA [5] (for Web contents) and PHITS [1] (for Web links).

Let t_i denote the i^{th} term, d_j the j^{th} document, c_l the document being cited (or linked), N_{ij} the observed frequency that t_i exists in d_j, A_{lj} the observed frequency that c_l is being linked by d_j.

By assuming that given an underlying latent factor z_k, t_i and c_l are independent of d_j and are independent of each other, the log likelihood $\mathcal{L}$ of the observed data (Web pages) can be given as

$$\begin{aligned}\mathcal{L} &= \sum_j \Bigg[\alpha \sum_i N_{ij} \log \sum_k P(t_i|z_k)P(z_k|d_j) \\ &\quad +(1-\alpha)\sum_l A_{lj} \log \sum_k P(c_l|z_k)P(z_k|d_j)\Bigg]\end{aligned} \tag{1}$$

where α determines the relative importance between observed terms (used in PLSA) and observed links (used in PHITS). Data normalization is adopted as in [2] to reduce the bias due to different document sizes. Model parameters $\{P(t_i|z_k), P(c_l|z_k), P(z_k|d$ are estimated using the tempered Expectation and Maximization (EM) algorithm [2] so as to avoid the local minimum problem of the standard EM algorithm.

III. Model exchange methodology for LCM learning

As mentioned in Section 1, the main focus of this paper is to explore how well physically separately datasets can be used to learn a global cluster model (LCM in our case) through periodic model exchange. The traditional methodolody of distributed learning is to do it in a two-stage manner — finishing local analysis and then merging the local results. For LCM learning, it corresponds to learning the local LCMs $\{LCM^{lm}\}$ first based on terms and hyperlinks information observed at each distributed site, and then performing the model merging subsequently to form the global model LCM^{gm}. In this paper, we view this methodology as an *one-shot* model exchange scheme. Based on this scheme, only the standard LCM learning process is needed at each site and the accuracy of the global estimate is determined only by how well the local models are merged.

Instead of only exchanging models at the final stage, we here propose a *multiple* model exchange scheme, where the two stages of learning interleave to perform some *cross learning*. Other than accessing its local set of data, each local data source will, now, receive from time to time models of the other data sources to help the model estimation task. The EM step implementation needed at each local site for LCM learning will be affected as parameters of local and non-local models are needed to be merged for each exchange before the sequent EM steps can be proceeded. After all the models in the distributed sites converge, the finally merged LCM is denoted as LCM^{gm}.

In the following, details of the one-shot and multiple model exchange schemes are explained. Also, the computational complexity as well as the communication overhead of the proposed schemes will be discussed as both are important for serious applications.

A. One-shot model exchange scheme

In this model exchange scheme, we perform only two main steps, namely *local model learning* and *model merging*. Figure 2 shows the overview of the one-shot model exchange scheme.

1) Local model learning: The local model learning step first estimates the parameters of LCM_p^{lm} using the local term-document matrix N_{ij}^p and link-document matrix A_{lj}^p observed at the p^th site.[1] One can follow the computation as described in Section II to estimate the model parameters' values. For setting the value of α, it is believed that different sites, possessing different data, may require a different value of for optimal performance. In this paper, we learn multiple LCM^{lm}s within a site by varying α from zero to one, with lower and upper extremes corresponding to PHITS and PLSA, as explained in [2]. To find the optimal one, we first use a factored nearest neighbor approach for measuring the factoring accuracy. In particular, for a learned LCM corresponding to a given value of α, a Web page d_j is considered to be correctly factored by that LCM if it belongs to the same class[2] of its neighbors. To define the neighborhood, we compute the cosine value of the Web pages' projections on the factor space, given as

$$sim(\vec{P}(z|d_i), \vec{P}(z|d_j)) = \frac{\vec{P}(z|d_i) \cdot \vec{P}(z|d_j)}{\|\vec{P}(z|d_i)\| \cdot \|\vec{P}(z|d_j)\|}. \tag{2}$$

The model associated to an α which gives the highest overall accuracy will be chosen for the subsequent merging.

2) Model merging: It is common that distributed data sources are heterogeneous. For example, in our case, the data at different Web sites are best described by different parameter sets, involving different terms, links as well as different latent classes (hidden patterns) captured by z. In order to combine different local models $\{LCM_p^{lm}\}$ to form a global one, we first need to assume that the unique identity of each data item can be identified to the extent that repeated appearance of them in different sites can be found. Thus, those repeated data items, after merging, can be re-indexed to aggregate their effect in the learning process. After reindexing, the latent parts of the local models whose identities can never be pre-defined have to be aligned before they can be merged.

Re-indexing: For each local model, we first enlarge and re-index the set of model parameters $\{P(z|d), P(t|z), P(c|z)\}$ by noting the difference between the local model and the other non-local models received from the other data sources. The parameters of the unseen variables are first initialized to zero.

Latent variables matching: As the latent part of each local LCM is induced from their corresponding training datasets, it is hard to have a pre-agreed way to know how they should be matched. Here, we propose to use the relative entropy between the probability distributions of the latent variables for a pair of local LCMs to align their latent variables.

For our application domain, two cases are to be considered: a) Web pages in different sites are non-overlapping, and b) some Web pages are shared in different sites. For the former case, we merely need to consider $P(t_i|z_k)$ and the relative entropy of a pair of latent variables z_k and $z_{k'}$ corresponding to two local models LCM_p^{lm} and $LCM_{p'}^{lm}$ is given as

$$H1_{p,p'}(z_k, z_{k'}) = \sum_i P_p(t_i|z_k) \log \frac{P_p(t_i|z_k)}{P_{p'}(t_i|z_{k'})}. \tag{3}$$

For the latter case, we use $P(t_i, c_l|z_k)$ for computing the relative entropy, given as

$$H2_{p,p'}(z_k, z_{k'}) = \sum_i \sum_l P_p(t_i, c_l|z_k) \log \frac{P_p(t_i, c_l|z_k)}{P_{p'}(t_i, c_l|z_{k'})}. \tag{4}$$

Two latent classes are considered to be closely matched if the value of their relative entropy is close to zero. The best one-to-one matching between the two sets of latent class models are computed based on the matrix $\{H1_{p,p'}(z_k, z_{k'})\}$ or $\{H2_{p,p'}(z_k, z_{k'})\}$. In this paper, we only consider the case where the LCMs have identical numbers of latent variables and assume that their latent variables possess the one-to-one correspondence property. In general, these assumptions should be relaxed.

Parameter merging: After the latent variables are matched, we can readily combine the local and non-local model parameters. For simplicity, we use simple averaging for the merge. A weighted sum based on some accuracy or uncertainty measures of the local models may worth further research effort.

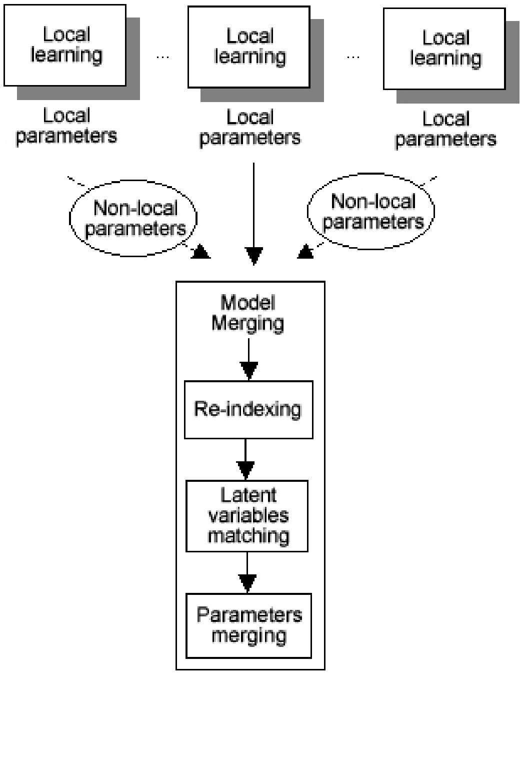

Fig. 2. Overview of one-shot model exchange scheme.

B. Multiple model exchange scheme

Under the multiple model exchange scheme, the local learning and model merging steps for one-shot model exchange

[1] Note that cross-site links are not considered in this pilot study, which however is an important part to be included in our future work.

[2] The class labels are available in the training set.

interleave *during* the learning process, which we call it *cross learning*. Cross learning is here defined as learning a local model with the use of non-local information *during* the learning process. Local model parameterss are exchanged at the intermediate stages, instead of the final stage. Similar to the one-shot model exchange scheme, such a cross learning process involves four steps, namely re-indexing, latent variables matching, parameter merging and local model parameter estimation. Most of them are identical to those for the one-shot model exchange, except for some minor implementation details. However, as the model exchange happens multiple times, the exchanging and merging steps could have much more influence on the overall performance. The main rationale is that periodic sharing of intermediate local analysis results can reduce the biases due to the local data and thus help learn a more accurate overall global model. Figure 3 shown the overview of the periodic model exchange scheme.

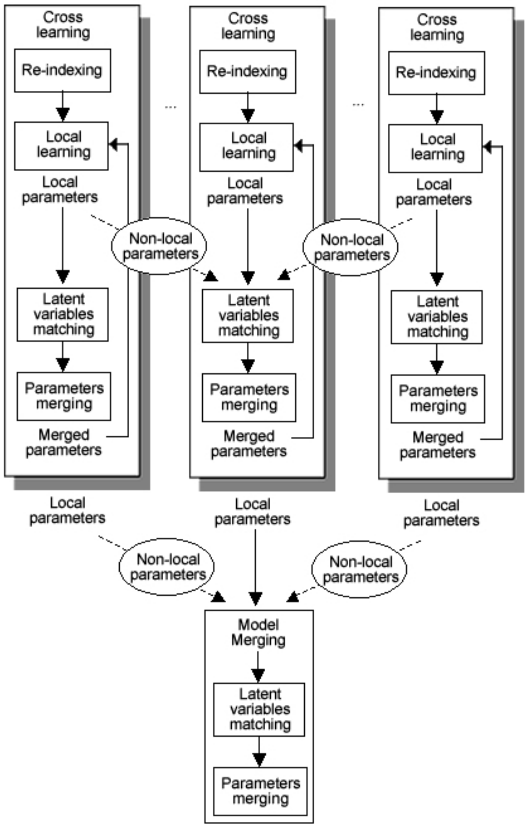

Fig. 3. Overview of multiple model exchange scheme.

C. Communication overhead and computational complexity

In this section, the asymptotic communication overhead and computational complexity of the two model exchange schemes are discussed in detail. Table I shows the notations used. Here, the communication overhead (CO) per model exchange includes parameters transmission. Related overheads for the two schemes are basically the same, given as

TABLE I

NOTATIONS

Notation	Definition
M/M_g	Number of local/global Web pages
N/N_g	Number of local/global hyperlinks
P/P_g	Number of local /global terms
Q	Number of latent variables
R	Number of distributed sources
I_{ter}	Number of EM iterations
I_{ex}	Number of non-local parameters exchanges
CO	Overhead of parameters transmission per model exchange

$$\begin{aligned} CO &= O(Q(M+N+P)/bandwidth) \\ &= O(rQ(M+N+P)) \end{aligned}$$

For the computational complexity, we compare the performance of the two exchange schemes ($O_{one}, O_{multiple}$) as well as the case with a single centralized server hosting all the data ($O_{central}$). They are given as

$$\begin{aligned} O_{central} &= O(I_{ter}M_gQ(N_g+P_g)) \\ O_{one} &= O(I_{ter}MQ(N+P)) \\ O_{multiple} &= O(I_{ter}MQ(N+P) \end{aligned}$$

For the overall complexity ($O^{overall}$), we add up the communication overheads and the computational ones, given as

$$\begin{aligned} O^{all}_{central} &= O(I_{ter}M_gQ(N_g+P_g)) \\ O^{all}_{one} &= O(I_{ter}Q((N+P)(M+r)+rM)) \\ O^{all}_{multiple} &= O(I_{ter}Q((N+P)(M+rI_{ex})+rM)) \\ &= O^{all}_{one}+O(rI_{ter}I_{ex}Q(N+P)). \end{aligned}$$

Thus, it is noted that the communication overhead (CO) becomes insignificant when the size of the dataset (of the order $M(N+P)$) is much larger than that of the models (of the order $(M+N+P)$). The overall computational complexity will still be dominated by the local learning processes. Furthermore, M is much smaller than M_g in general as R increases (that is the data are more distributed). Therefore, the parallellism gained by the independent learning of the local models $\{LCM^{lm}\}$ should result in a shorter overall learning time when compared with that of the global model LCM^{gm}.

D. Model exchange scheme with additional privacy preserving cabability

One of the important motivations for sharing models instead of data is related to data privacy. For the aforementioned application on Web structure analysis, sharing local LCM models assumes the knowledge of a set of unique identifiers for all the Web pages at different data sources, no matter they are for public access or within the intranets. While each site can re-label all the identifiers based on the URLs of the Web pages, sharing those identifiers may still cause privacy and security concern of internal users of different sites. The situation will become even more obvious if we replace each Web page by a customer and each hyperlink by a product that a customer has purchased [24]. There will not be a company willing to share with others their transaction records. One effective way to alleviate the aforementioned privacy issue is to share only aggregated information. In our case, the model parameters belonging to this category (which we refer as the most *privacy-friendly* parameters as illustrated in Figure 4) is $P(t|z)$.

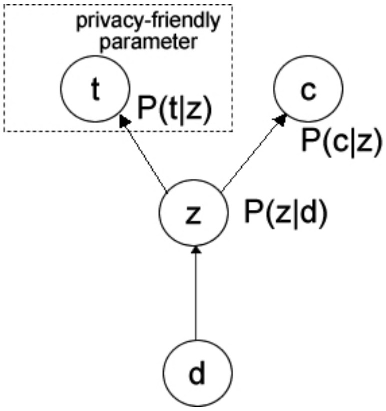

Fig. 4. Privacy-friendly parameters of LCM.

By sharing only $P(t|z)$, we gain additional advantage due to the reduced requirement of communication cost as well as computational complexity. Note that as the value of N and M increases, the increase in P will soon be saturated if the vocabulary under a particular domain is exhausted. The corresponding overall complexity can be reduced to:

$$O_{multiple}^{all} = O_{one}^{all} + O(rQPI_{ter}I_{ex}).$$

IV. Performance Evaluation

We have applied the proposed model exchange approach to the WebKB dataset [6]. As this study is novel and there are no directly related works in the literature, we conduct our experiments to compare the commonly adopted two-stage approach and the proposed multiple model exchange approach. While our experiment focuses only on learning LCM in a distributed manner, we believe that the approach should also apply to distributed learning of other statistical models.

For the training and testing dataset, a total of 546 web pages, which are pre-classified into 3 categories: *course*, *department* and *student* in WebKB, have been used and each class contains 182 pages. In the following, we describe the data pre-processing steps adopted and how the experiments were designed and conducted.

A. Web page preprocessing

As mentioned in Section II, the term-document matrix N_{ij} and hyperlink-document matrix A_{lj} are required for the LCM learning. Hyperlinks between Web pages can easily be identified based on the anchor tags for computing A_{lj}. For Web page contents, we removed all the html tags as well as the contents between the `<SCRIPT>` tags. Also, stopwords removal and stemming [4] were applied subsequently. The remaining terms were all changed to be of lower case. We then extracted only terms with their document frequencies bigger than a threshold value [3]. We have tested the threshold of 5, 10, and 20 (denoted as DF05, DF10, DF20), resulting in datasets with their numbers of distinct words equal 1629, 957 and 550 respectively. The factored nearest neighbor approach (1-nn and 3-nn), as described in Section III-A.1, is used for comparing their accuracy and the corresponding results are shown in Table II. We found that DF10 and DF20 outperform DF5 and the performance of DF10 and DF20 are comparable. As a smaller number of terms implies lower computational complexity as shown in the previous section, DF20 was used in the subsequent experiments.

TABLE II
CLASSIFICATION ACCURACY (%) FOR D05, D10 AND D20.

	1-nn (%)	3-nn (%)
D05	81.46	82.71
D10	86.30	85.20
D20	86.41	86.81

B. Experiment setups for different model exchange schemes

We performed a number of experiments for learning latent class models using the one-shot and multiple model exchange schemes with 1) different parameter exchange periods to indicate different degrees of non-local data availability 2) different numbers of distributed data sources to indicate different degrees of data distribution. In particular, we have tried different exchange periods, 2, 5, 10, 15, 20 and ∞ (which degenerates to one-shot model exchange case) and performed the experiments with 2 to 6 distributed data sources. For preparing the distributed data sources, we partitioned the WebKB dataset so that part of Web pages in one partition also appear in some others. Classification accuracy (as described in Section III-A.1) and the training time are the performance measures we adopted. As the local models learned at the distributed sites have to synchronize at each model exchange stage, in our experiment, we recorded the maximum computational time among those needed by the distributed servers. To contrast the additional privacy concern mentioned in Section III-D, we deliberately learned an LCM by exchanging all three sets of model parameters, i.e., $P(t|z)$, $P(c|z)$ and $P(z|d)$, and another only the privacy-friendly parameters, i.e., $P(t|z)$ for performance comparison. Lastly, as the EM algorithm only gives sub-optimal solutions, for each LCM training, we have tried ten different random initializations and reported the average performance of the ten cases.

C. Experimental Results

1) Performance comparison for exchange different sets of model parameters: The classification accuracy and the training time associated to the distributed LCM learning with all parameters exchanged and with only the privacy-friendly parameters exchanged for two distributed sets are tabulated in Table III and IV, respectively.

TABLE III
CLASSIFICATION ACCURACY (%) AND TRAINING TIME (SEC) BASED ON EXCHANGING THE FULL SET OF PARAMETERS.

Exchange period	**1-nn (%)**	**3-nn (%)**	**Time (mm:ss)**
∞	81.85	80.90	0:54
20	77.93	78.64	1:47
15	78.04	77.91	1:41
10	78.59	79.38	2:11
5	80.73	82.11	2:17
2	**83.26**	**84.71**	2:29

TABLE IV
CLASSIFICATION ACCURACY (%) AND TRAINING TIME (SEC) BASED ON EXCHANGING ONLY THE PRIVACY-FRIENDLY PARAMETERS.

	1-nn (%)	**3-nn (%)**	**Time (mm:ss)**
∞	83.11	82.75	0:56
20	84.45	81.45	0:55
15	82.03	81.47	0:57
10	80.93	81.32	1:03
5	83.11	83.10	1:06
2	**87.51**	**87.55**	1:43

According to Table III and IV, it is observed that the performance of exchanging only the privacy-friendly parameters is always better than that of exchanging the full set of model parameters. In addition, as expected, the computational time for exchanging only the privacy-friendly parameters is significantly less than that for exchanging all model parameters because of the reduced communication overhead. This effect is especially obvious for cases with higher exchange frequencies. Therefore, in the following experiments, we only adopt the scheme of exchanging $\{P(t|z)\}$.

TABLE V
CLASSIFICATION ACCURACY (%) EVALUATED BY 1-NN BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION AND DIFFERENT MODEL EXCHANGE PERIODS.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	83.11	81.45	82.03	80.93	83.11	**87.51**
3 sets	**87.23**	86.74	86.52	86.43	85.55	85.82
4 sets	84.93	85.53	**87.57**	87.14	79.43	83.86
5 sets	77.01	79.38	79.93	**84.40**	79.74	83.22
6 sets	76.85	79.62	82.05	83.46	81.52	**83.55**

TABLE VI
CLASSIFICATION ACCURACY (%) EVALUATED BY 3-NN BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION AND DIFFERENT MODEL EXCHANGE PERIODS.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	82.75	81.45	81.47	81.32	83.10	**87.55**
3 sets	87.45	87.40	**87.73**	87.42	85.55	86.65
4 sets	84.30	85.27	87.29	**87.78**	85.95	85.79
5 sets	78.11	79.41	80.29	84.07	84.45	**85.11**
6 sets	75.66	78.74	82.05	83.04	84.18	**85.55**

TABLE VII
TRAINING TIME (SEC) BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	0:56	0:55	0:57	1:03	1:06	1:43
3 sets	0:50	0:52	0:55	0:50	1:37	1:34
4 sets	0:40	0:41	0:41	0:41	1:33	1:30
5 sets	0:37	0:33	0:41	0:42	1:29	1:28
6 sets	0:30	0:33	0:33	0:35	1:24	1:24

2) Performance sensitivity on different degrees of data distribution and different model exchange periods: In Table V to VII, the classification accuracy evaluated based on the 1-nn and 3-nn factoring approaches as well as the training time for learning the LCM based on different experiment settings are reported. According to Table V and VI, the accuracy decreases monotonically as the number of distributed sources increases. It is possibly due to the fact that when data are distributed to different sites, the amount of available information for each source decreases. Therefore, the overall performance is reduced. The gain, as shown in Table VII, is that the training time is reduced (due to the parallellism). By allowing model exchange, as observed in Table V and VI, we found that the accuracy can be significantly increases, the trend is not especially clear though. In general, allowing parameters exchange more frequently can result in better overall performance of the global model.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a methodology for learning a global latent model from distributed data sources by multiple model exchanges with promising results. With the option to exchange only the privacy-friendly parameters, our empirical results show that the overall model gives acceptable and sometimes even better accuracy. In addition, we observed that while the increase in the number of distributed sites can lower the overall accuracy of the global model, the interpolating effect caused by the model exchange can improve the accuracy to some extent.

While this work provides us some interesting and encouraging results for exploring distributed data mining through model (or generally speaking knowledge) exchange, there still exist a number of areas worth further research effort. What we have proposed in this paper is a model-specific methodology for distributed data mining. Ways for generalizing the proposed methodology so as to be applied to different types of models is one of the worth-pursuing research directions.

In addition, the way we exchange all local model parameters or privacy-friendly parameters is based on some fixed time periods. one can go one step further to derive adaptive on-demand model exchange strategies for minimizing the communication cost while still maintaining the desired accuracy. Furthermore, there still exist no guarantee for the convergence of the global model. We are currently investigating different discounting strategies for addressing the model convergence issue.

REFERENCES

[1] D.Cohn, H.Chang, "Learning to probabilistically identify authoritative documents," *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[2] D.Cohn, T.Hofmann, "The missing link - A probabilistic model of document content and hypertext connectivity," *Advances in Neural Information Processing Systems*, 2001.

[3] E. Bingham, J. Kuusisto and K. Lagus, "ICA and SOM in text document analysis," *Proceedings of SIGIR'02*, 361 - 362 2002

[4] G.Salton and M.J.McGill, *Introduction to modern information retrieval*, McGraw-Hill, New York, 1983.

[5] T. Hofmann, "Probabilistic latent semantic analysis," *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[6] Web-KB. Available electronically at *http://www.cs.cmu.edu/˜WebKB/*.

[7] Distributed Data Mining Bibliography. Available at *URL: http://www.csee.umbc.edu/˜hillol/DDMBIB/ ddmbib_html/index.html*.

[8] R. Chen and S. Krishnamoorthy, "A new algorithm for learning parameters of a Bayesian network from distributed data," Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), pages 585-588, Maebashi City, Japan, December 2002.

[9] H. Kargupta, B. Park, D. Hershberger, and E. Johnson, "Collective data mining: A new perspective towards distributed data mining," *Advances in Distributed and Parallel Knowledge Discovery*, pages 133-184. MIT/AAAI Press, 2000.

[10] A. Prodromidis and P. Chan, "Meta-learning in distributed data mining systems: Issues and approaches," *Advances of Distributed Data Mining*, MIT/AAAI Press, 2000.

[11] M. Cannataro and D. Talia, "The Knowledge Grid," *Communications of the ACM*, 46(1):89-93, January 2003.

[12] R. Chen, S. Krishnamoorthy, and H. Kargupta, "Distributed Web mining using Bayesian networks from multiple data streams," *Proceedings of the IEEE International Conference on Data Mining*, pages 281-288, IEEE Press, November 2001.

[13] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 639-644, July 2002.

[14] J. Vaidya and C. Clifton, "Privacy preserving K-means clustering over vertically partitioned data," *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206-215, August 2003.

[15] C. Clifton, M. Kantarcioglu, J. Vaidya, X.D. Lin and M.Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations Newsletter*, 4(2):28 - 34, 2002.

[16] W.L. Du, Yunghsiang S. Han and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 222-233, 2004.

[17] H. Polat and W.L. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," *Proceedings of The Third IEEE International Conference on Data Mining*, pages 625-628, 2003.

[18] W.L. Du and Z.J. Zhan, "Using randomized response techniques for privacy-preserving data mining," *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 505-510, 2003.

[19] S. Merugu and J. Ghosh, "A probabilistic approach to privacy-sensitive distributed data mining," *Proceedings of the Sixth International Conference on Information Technology (CIT)*, 2003.

[20] S. Merugu, A. Banerjee, I. Dhillon and J. Ghosh, "Clustering with Bregman Divergences," *Proceedings of the Fourth IEEE International Conference on Data Mining*, 2004.

[21] S. Merugu and J. Ghosh, "Distributed data mining with limited knowledge sharing," *Proceedings of the Fifth International Conference on Advances in Pattern Recognition (ICAPR)*, 2003.

[22] S. Merugu and J. Ghosh., "Privacy-preserving distributed clustering using generative models," *Proceedings of the Third IEEE International Conference on Data Mining*, 2003.

[23] S. Merugu, J. Ghosh and A. Strehl, "A Consensus Framework for Integrating Distributed Clusterings under Limited Knowledge Sharing," *Proceedings of the National Science Foundation (NSF) Workshop on Next Generation Data Mining*, 2002.

[24] K. Cheung, K.C. Tsui and J. Liu, "Extended Latent Class Model for Collaborative Recommendation," IEEE Transactions on Systems, Man and Cybernetics - Part A, Vol.34, No.1, pp. 143-147, January, 2004.