

On Mining Local Data Sources For Learning Global Cluster Models - An Empirical Study

Chak-Man Lam, Xiao-Feng Zhang, Kwok-Wai Cheung
Hong Kong Baptist University
Computer Science Dept.
Kolwoon Tong, Hong Kong
johanna, xfzhang, william@comp.hkbu.edu.hk

Abstract

Distributed data mining has been a topic getting more important nowadays as there are many cases where physically sharing of data is prohibited, e.g., due to huge data volume or data privacy. In this paper, we are interested in learning a global cluster model by exploring data in distributed sources. A methodology based on periodic model exchange and merge is proposed and applied to hyperlinked Web pages analysis. In addition, we have tested a number of variations of the basic idea, including putting more emphasis on the privacy concern and testing the effect of having different numbers 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.

1. Introduction

Most of the machine learning and data mining algorithms work with a rather basic assumption that all the training data have been pooled in a centralized data repository. Recently, there exist a growing number of cases that the data have to be physically distributed, due to either their huge volumes or privacy concern. Relevant examples include distributed medical data analysis, intrusion detection, data fusion in sensor networks, customer record analysis, 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 paradigm is a two-stage one — first performing local data analysis and then combining the local results forming a global model. 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). A related implementation has been realized under a Grid platform known as the

Knowledge Grid [11]. In [9], Kargupta *et al.* proposed the so-called collective data mining where the distributed data possess different sets of features, each being considered as an orthogonal basis. The orthogonal bases are then combined to give the overall result. The scheme have been applied to learning Bayesian Networks for Web log analysis [12, 8].

In [13], instead of having the two-stage setting, a methodology having the local data analysis stage and results combining stage interleaved was proposed. The main rationale of the proposed methodology lies on the conjecture that periodic sharing of intermediate local analysis results can reduce the local biases and thus help learning a more accurate global model. In this paper, a few variations of the proposed methodology have been proposed, including the situation that a higher level of privacy is required as well as that with an increasing number of distributed sources. Similar to [13], a particular latent class model for hyperlinked Web pages is chosen as the global cluster model where the expectation and maximization algorithm is adopted as usual for the local model training. Sharing of intermediate results is done via model exchanges among the distributed data sources, and relative entropy is used as the measure for aligning, and thus merging, of the local latent class models. Experiments based on WebKB have been performed and the results corresponding to the different variations of the model-exchange methodology have been compared. We found that some settings can achieve an accuracy which is even higher than the case with all the data physically shared for the model learning.

The remaining of the paper is organized as follow. Section 2 describes a particular latent class model for modeling hyperlinked Web pages. Section 3 describes 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 experiment setup

for evaluation different variations of the basic idea as well as the corresponding results can be found in Section 4. Section 5 concludes the paper with possible future directions.

2. 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-occurrence of multiple random variables with applications to a number of areas. In [2], a latent class model for analyzing Web contents and Web links was proposed, which can be considered as a joint model of two related latent class models PLSA (for Web contents) and PHITS (for Web links) [2].

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} given the observed data can be written as

$$\mathcal{L} = \sum_j \left[\alpha \sum_i N_{ij} \log \sum_k P(t_i|z_k)P(z_k|d_j) + (1 - \alpha) \sum_l A_{lj} \log \sum_k P(c_l|z_k)P(z_k|d_j) \right] \quad (1)$$

where α determines the relative importance between an observed term (used in PLSA) and an observed link (used in PHITS). Data normalization, as used in [2], is adopted to reduce the bias due to document size and the tempered Expectation and Maximization algorithm is used for estimating the model parameters $\{P(t_i|z_k), P(c_l|z_k), P(z_k|d_j)\}$.

3. 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) via periodic model exchange. The traditional methodology of distributed learning is to do 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_{loc}) 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_{mm}). In this paper, we call this methodology *one-shot* model exchange. 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.

The newly proposed methodology we are testing here is named *multiple* model exchange, where the two stages of learning are interleaved and *cross learning* is performed instead. Other than accessing its local set of data, each local data source will, now, receive from time to time models of *cross-site* data to help the estimation task. The EM steps involved in the LCM learning will be affected as parameters of local and non-local models are needed to be merged as the EM steps proceed. After all the models in the distributed sites converge, the finally merged LCM is denoted as LCM_{emm} . In the following, details of 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.

3.1. One-shot model exchange scheme

In this model exchange scheme, we perform two main steps, namely *local learning* and *model merging*. Figure 1 shown the overview of one-shot model exchange scheme.

3.1.1. Local learning The local learning step learn the parameters of LCM_{loc} using the local term-document matrix N_{ij} and link-document matrix A_{lj} observed at each site. While one can follow the description in Section 2, the way to set the value α is still unknown and it is believed that different sites, possessing different datasets, may work best at different α . In this paper, we learn multiple LCM_{loc} s within a site by varying α from zero to one, with lower and upper extremes corresponding to PHITS and PLSA. To find the optimal one, a factored nearest neighbor approach is used. In particular, a Web page d_j is considered to be correctly factored by an LCM if it belongs to the same class¹ of its neighbors, which is defined by the pages' projections in the factor space to identify the nearest neighbor. That is, we compute the cosine between the $\{P(z_k|d_j)\}$ of two pages as the neighborhood measure. The model associated to an α which gives the highest overall accuracy will be chosen for the subsequent merging.

3.1.2. Model Merging It is common that the distributed data sources are heterogeneous and contain data with different sets of parameters. In our case, this implies that the sets of $\{d_j, c_l, t_i\}$ in different sites could be different. In order to combine all the LCM_{loc} s to form a global one, we first need to re-index the parameters of different LCM_{loc} . Here, it is assumed that there is a way to uniquely determine the identity of a particular parameter. After reindexing, the

¹ The class labels are available in the training set

latent parts of the local models (with the enlarged set of parameters) 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 received non-local models. The parameters of the unseen variables are all initiated to zero.

Latent variables matching As the latent part of each local LCM is induced from training data, it is hard to have a pre-agreed way to know how they should be matched. Instead, we propose to use relative entropy between the probability distributions that the observed pages and links are generated given the latent variables to align the latent variables between different LCM_{loc} s. Let $\{z_1, z_2, z_3, \dots\}$ and $\{z_a, z_b, z_c, \dots\}$ denote the latent variables of LCM_{loc}^1 and LCM_{loc}^2 respectively. For our application, two cases are to be considered: a) Web pages in different sites being non-overlapping, and b) some pages are shared in different sites. For the former case, we use merely $P(t_i|z_k)$ for computing the relative entropy for a latent class pair, given as

$$H_{k,k'}^1(LCM_{loc}^1, LCM_{loc}^2) = \sum_i P(t_i|z_k) \log \frac{P(t_i|z_k)}{P(t_i|z_{k'})}. \quad (2)$$

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

$$H_{k,k'}^2(LCM_{loc}^1, LCM_{loc}^2) = \sum_i \sum_l P(t_i, c_l|z_k) \log \frac{P(t_i, c_l|z_k)}{P(t_i, c_l|z_{k'})}. \quad (3)$$

Two latent classes are considered to be closely matched if the value of the relative entropy measure is close to zero. The best one-to-one matching between the two sets of latent class models are computed based on the matrix $H_{k,k'}^1(LCM_{loc}, LCM_{cro})$ or $H_{k,k'}^2(LCM_{loc}, LCM_{cro})$. In this paper, we only consider the case of LCMs with identical numbers of latent classes. In general, this assumption 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.

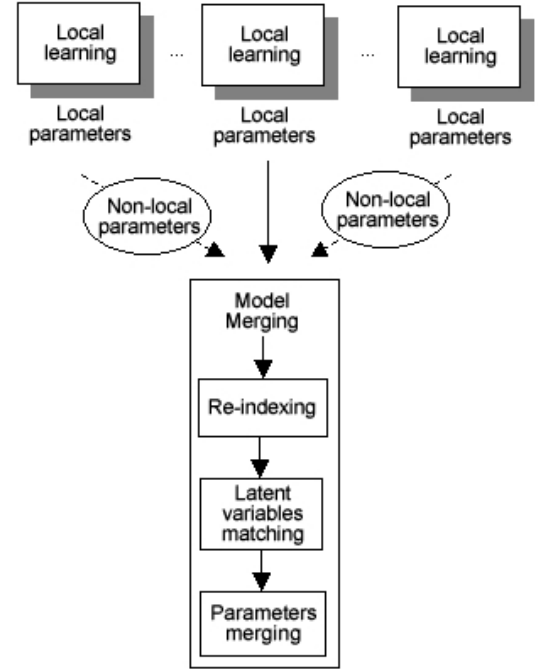


Figure 1. Overview of One-shot Model Exchange Scheme.

3.2. Multiple model exchange scheme

For this model exchange scheme, the local learning and model merging steps for one-shot model exchange are interleaved *during* the learning, 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. In this paper, local models are exchanged at the intermediate stages, instead of the final stage. Similar to the one-shot model exchange, cross learning involves four processes, namely re-indexing, latent variable matching, parameter merging and local learning. Most of them are identical to those for the one-shot model exchange with only minor implementation details. The research issues include how to minimize the communication overhead in exchanging model parameters, adaptive fusion of the local models, matching models with uneven number of latent classes, exchanging models in a periodic manner or in an on-demand manner. Figure 2 shown the overview of the periodic model exchange scheme.

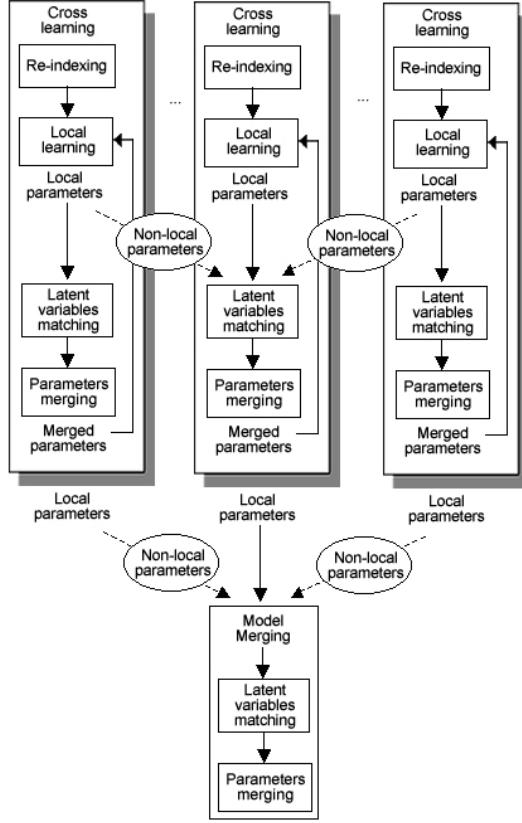


Figure 2. Overview of Multiple Model Exchange Scheme.

3.3. 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 1 shows the notations used. Here, the communication overhead (CO) per model exchange includes parameters transmission and merging. Related overheads for both schemes are basically the same, given as

$$H_t = (M'Q + N'Q + P'Q)/\text{bandwidth}$$

$$H_m \propto H_t$$

$$\text{CO per model exchange} = H_t + H_m$$

For the computational complexity (CC), we compare the performance of the two exchange schemes (CC_{mm} , CC_{emm}) as well as the case with a single centralized server hosting all the data (CC_{cen}). They are given as

$$CC_{cen} = I_{ter}(M'_g N'_g Q + M'_g P'_g Q)$$

Table 1. Notations

Notation	Definition
M/M_g	Local/Global number of Web pages
N/N_g	Local/Global number of hyperlinked pages
P/P_g	Local/Global number of 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
H_t	Parameters transmission overhead per model exchange
H_m	Parameters merging overhead per model exchange

Table 1. Notations

$$CC_{mm} = I_{ter}(M'N'Q + M'P'Q)$$

$$CC_{emm} = I_{ter}(M'_g N'_g Q + M'_g P'_g Q)$$

For the overall complexity (TT), we add up the communication overheads and the computational ones, given as

$$TT_{cen} = I_{ter}(M'_g N'_g Q + M'_g P'_g Q)$$

$$TT_{mm} = I_{ter}(M'N'Q + M'P'Q) + (H_t + H_m)R$$

$$TT_{emm} = I_{ter}(M'_g N'_g Q + M'_g P'_g Q) + (H_t + H_m)RI_{ex}$$

Thus, it is noted that the communication overhead becomes insignificant when the size of the datasets are much bigger than that of the models. The overall computational time will still be dominated by the local learning processes. Furthermore, M is much smaller than M_g in general. Therefore, we expect that LCM_{mm} needs less time for learning when compared with LCM_{emm} .

3.4. Model exchange scheme with additional privacy concern

One of the important motivations for sharing models instead of data is related to data privacy. Sharing all the model parameters, sometimes, may not be feasible too. For the aforementioned application on Web structure analysis, sharing local LCM models assume the knowledge of a set of unique identifiers for all the Web pages in the different data sources, no matter they are for public access or within the intranet. While each site can re-label all identifiers, sharing those identifiers may still cause privacy concern of internal users of different sites. One effective way to protect privacy is to share only aggregated information and the only model parameter without the need of Web page unique identifiers (i.e., the most privacy-friendly parameter of LCM which is illustrated in Figure 3) is $P(t|z)$.

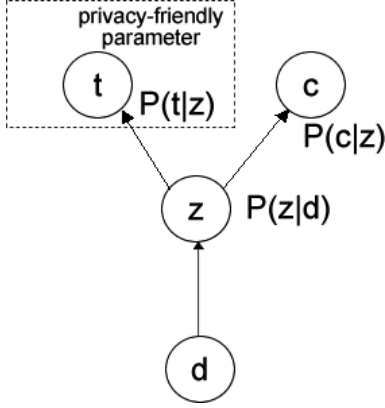


Figure 3. The Most Privacy-Friendly Parameter of LCM.

By sharing only $P(t|z)$, we gain additional advantage due to the reduced requirement of communication cost as well as computational complexity. The corresponding overall complexity can be reduced to

$$TT_{emm}^{private} = I_{ter}(M'N'Q + M'P_g'Q) + (H_t^{private} + H_m^{private})RI_{ex}$$

where $H_t^{private}$ and $H_m^{private}$ are significantly smaller than H_t and H_m correspondingly.

4. Experiments

For performance evaluation, we have applied the proposed scheme to the WebKB dataset [6]. As a preliminary experiment, a total of 546 web pages have been used, which are pre-classified into 3 categories: course, department and student. Each class contains 182 pages. In the following, we describe how the data preprocessing steps that we have performed and how the experiments are designed and performed.

4.1. Web pages preprocessing

In the joint model proposed by [2], the term-document matrix N_{ij} and hyperlink-document matrix A_{lj} are needed 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 first removed all the html tags as well as the contents between the <SCRIPT> tags. Also, stop words removal and stemming [4] were performed subsequently. The remaining terms will all be changed to be of lower case. We then extracted only those with their document frequencies bigger than a threshold value [3]. We have

tested the threshold (the minimum number of document occurrence) of 5, 10, and 20 (denoted as DF5, DF10, DF20) and DF20 was used in the following experiments.

4.2. Experiment setups for different model exchange schemes

In our experiments, we performed LCM learning of multiple model exchange scheme with different number of distributed data sites. For LCM_{emm} , we have tried different model exchange periods, 2, 5, 10, 15, 20 and ∞ (which degenerates to one-shot model exchange case) and performed the experiments with 2 - 6 distributed data sources. Regarding the data partition issue, we assume that part of Web pages at different sites is overlapping to each other. In particular, we replicated the whole dataset and evenly distributed to different sites. To contrast the additional privacy concern mentioned in Section 3.4, we deliberately learned a related model, which exchange all the three model parameters instead of only the privacy-friendly parameters for performance comparison. As the EM algorithm can only give sub-optimal solution, for each LCM training, we have tried ten different random initializations reported the average performance of the ten cases.

4.3. Performance comparison

The classification accuracy and the training time associated to LCM_{emm} with all parameters exchange and only privacy-friendly parameter exchange for two distributed sets are tabulated in Table 2 and 3 respectively.

	1-nn (%)	3-nn (%)	Time (mm:ss)
$LCM_{emm}(\infty)$	81.85	80.90	0:54
$LCM_{emm}(20)$	77.93	78.64	1:47
$LCM_{emm}(15)$	78.04	77.91	1:41
$LCM_{emm}(10)$	78.59	79.38	2:11
$LCM_{emm}(5)$	80.73	82.11	2:17
$LCM_{emm}(2)$	83.26	84.71	2:29

Table 2. Classification Accuracy and Training Time for LCM_{emm} with All Model Parameters Exchange.

As the learning process taken place at distributed sites has to synchronize at each model exchange, we recorded

the maximum computational time among those need by the distributed servers.

	1-nn (%)	3-nn (%)	Time (mm:ss)
$LCM_{emm}(\infty)$	83.11	82.75	0:56
$LCM_{emm}(20)$	84.45	81.45	0:55
$LCM_{emm}(15)$	82.03	81.47	0:57
$LCM_{emm}(10)$	80.93	81.32	1:03
$LCM_{emm}(5)$	83.11	83.10	1:06
$LCM_{emm}(2)$	87.51	87.55	1:43

Table 3. Classification Accuracy and Training Time for LCM_{emm} with only Privacy-Friendly Parameters Exchange.

By observing both Table 2 and 3, we can found that for any period of parameter exchange, the performance of only exchange the privacy-friendly parameters (i.e. in our case is $P(t|z)$) always outweighs that of exchange all model parameters. This observation shows that by only exchanging the privacy-friendly parameters, the performance of the overall model still maintains or even having better classification accuracy than exchange all the model parameters. In addition, the computational time of only exchange the privacy-friendly parameters are significantly less than exchanging all model parameters because of the less communication overhead. This effect is obvious especially for higher exchange frequency. These empirical findings evidence our discussion expressed in Section 3.4. Therefore, in the following experiments, we only concern LCM_{emm} with only privacy-friendly parameter exchange.

	∞	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 4. Classification Accuracy (1-nn) of LCM_{emm} using Different Overlapped Datasets.

In Table 4 to 6, the classification accuracy with 1-nn and 3-nn and the training time of LCMemmm using differ-

	∞	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 5. Classification Accuracy (3-nn) of LCM_{emm} using Different Overlapped Datasets.

	∞	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

Table 6. Training Time of LCM_{emm} using Different Overlapped Datasets.

ent number of distributed overlapped datasets are shown respectively. According to Table 4 and 5, the accuracy decrease monotonically as the number of distributed sites increase. It is due to the fact that when data are distributed to different sites, the amount of available information for each sites decreases. Therefore, the overall performance is reduced. Besides, as shown in Table 6, the training time also decrease as the number of data sites increase.

Finally, by observing Table 4 and 5, we can still found that the accuracy increase if parameters are allowed to exchange during the learning stage. In generally, allowing parameters exchange frequently, can result in better overall performance of LCM learning.

5. Conclusion

In this paper, we have presented the LCM learning using multiple model exchange scheme in addition with the privacy concern from distributed data sources. With the option to exchange only the privacy-friendly parameters, the overall model still gives acceptable accuracy. Also the performance outperforms significantly to the one that exchange all the model parameters. Further, when the number of distributed sites increases, the accuracy of the global model will become lower. However, interpolating unseen data from different local models can result in a more ac-

curate global model. From the above empirical study, it is possible to learn global cluster model on mining local data sources. Currently, we are investigating an adaptive exchange scheme such that the communication cost can be minimized at the same time keeping the classification accuracy. Also, a detailed theoretical study on whether using the multiple model exchange scheme can arrive to global model convergence is under our discussion.

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*, 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*. In 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. Hersherberger, and E. Johnson. *Collective Data Mining: A New Perspective Towards Distributed Data Mining*. In Hillol Kargupta and Philip Chan, editors, 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*. In Hillol Kargupta and Philip Chan, editors, 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*. In Proceedings of the IEEE International Conference on Data Mining, pages 281-288. IEEE Press, November 2001.
- [13] C. Lam, X. Zhang and K. Cheung. *Learning Latent Class Models From Distributed Data Sources*. Submitted, 2004.