Neurocomputing 92 (2012) 116–123

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Capturing correlations of multiple labels: A generative probabilistic model for multi-label learning

Haiping Ma [a], Enhong Chen [a,*], Linli Xu [a], Hui Xiong [b]

[a] Computer Science Department, University of Science and Technology of China (USTC), China
[b] Management Science and Information Systems Department, Rutgers University, United States

ARTICLE INFO

Available online 12 March 2012

Keywords:
Multi-label learning
Ranking
Label correlation
Generative model

ABSTRACT

Recent years have witnessed a considerable surge of interest in the multi-label learning problem. It has been shown that a key factor for a successful multi-label learning algorithm is to effectively exploit relations between labels. However, most of the previous work exploiting label relations focuses on pairwise relations. To handle the situations where there are intrinsic correlations among multiple labels, in this paper, we propose a generative model, Labeled Four-Level Pachinko Allocation Model (L-F-L-PAM), to capture correlations among multiple labels. In our approach of multi-label learning on text data, we apply the proposed model for inferring the training data and the standard Four-Level Pachinko Allocation Model for the test data. Furthermore, we propose a pruned Gibbs Sampling algorithm in the test stage to reduce the inference time. Finally, extensive experiments have been performed to validate the effectiveness and efficiency of our new approach. The results demonstrate significant improvements of our model over Labeled LDA (L-LDA) and superiority in terms of both effectiveness and computational efficiency over other high-performing multi-label learning methods.

1. Introduction

Text data has become the major information source in our daily life [1]. In the meantime, with the explosive growth of the digitally stored text documents, content-based document management tasks that help users to quickly find their desired information have gained a prominent status in the data mining field [2]. One of these tasks is that of automatically organizing a text document into predefined categories, which is known as text classification. This problem has attracted more and more attention from researchers due to its wide applications, such as information retrieval (IR), information extraction and filtering, text mining, and natural language processing. In many real-world problems, text data, including documents and web pages, are frequently annotated with more than a single label. For example, a newspaper article talking about the reactions of Christian churches to the release of the Da Vinci Code film might belong to the following two categories: Society/Religion and Arts/Movies. Therefore, multi-label document classification has become a challenging research theme in the data mining field, and has found successful applications in various domains, not limited to traditional document classification. For instance, Katakis et al. [3] models the tag suggestion as a multi-label text classification task, where each tag associated with a document may be treated as a label. Since the task of multi-label document classification can be naturally modeled as a multi-label learning problem, its accuracy and efficiency can be improved through multi-label learning algorithms.

A variety of multi-label learning approaches have been proposed in the literature. Most of them involve learning a number of different binary classifiers [4,5] and using the outputs to determine the label or labels of a new sample. The two main deficiencies of these methods are (1) the rough separation strategy ignores the correlation between the classes; (2) these approaches toward multi-label learning suffer severely from unbalanced data, especially when the number of labels is large.

The second group of studies on multi-label learning considers the pairwise relations between class labels. These methods involve modeling the correlations between any two class labels for multi-label learning in a generative way [6,7], or discriminatively extending specific single-label learning algorithms to incorporate pairwise label correlations [8–13].

However, these pairwise approaches may suffer from the fact that the correlations between different labels would possibly go beyond second-order [14]. There exists some previous work in the literature which exploits higher order relations between class labels, most of which imposes all the other labels' influence on each label [15–17], or addresses correlations among a random

* Corresponding author. Tel.: +86 13956957326.
E-mail address: cheneh@ustc.edu.cn (E. Chen).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.08.039

subset of labels [18,19]. However, such assumptions are likely to be violated in many real-world applications, where certain structures often exist among labels. Table 1 shows an example of association rules among labels in a real data set. Association rule learning has been considered for the task of label correlation learning in [20]. We notice that not only pairwise correlations between labels such as the label "M14" and "MCAT" but also correlations among multiple labels such as the label "CCAT", "C151" and "C15" exist. To handle these types of label structures, Zhang et al. [14] exploits label dependency for multi-label learning by incorporating the correlated labels as additional features to construct classifiers for each label. The resulting Bayesian networks encoding the conditional dependencies of the labels is approximately learned.

In this paper, we present a generative model for learning from multi-label text data. We have several motivations for using a generative model to capture correlations of multiple labels. First, it would be easy to postulate complex latent structures responsible for a set of observations; Second, the correlations between the different factors could be easily exploited by introducing latent variables [6]. Fig. 1 shows the DAG structure of the model. There is one root, s nodes at the second level, s' labels at the third level and n words at the bottom. If multiple label nodes at the third level are related, they will have a parent node in the second level. To simplify the model, we learn the parameters instead of learning structure of connectivity between the nodes in the second level and nodes in the third level. The root is connected to all nodes at the second level, nodes at the second level are fully connected to labels and labels are fully connected to words. In this paper, the nodes in the second level are called super-labels. As we can see, the model structure is similar to the Four-Level Pachinko Allocation Model [21]. The major difference is that the third level consists of observed labels instead of latent sub-topics. Thus, in this paper, we call the proposed generative model Labeled Four-Level Pachinko Allocation Model (L-F-L-PAM for short).

Finally, we also show how to use this model for multi-label learning. Additionally, in order to reduce the inference time in the test stage, we present Pruned Gibbs Sampling for inferring the unlabeled test documents. We conduct extensive experiments on real news articles and web pages to compare our proposed model with the other state-of-the-art baselines. The experiments show that our proposed model that considers relations of multiple labels can greatly improve the performance in the label ranking task on multi-label text data.

Overview: The remainder of the paper is organized as follows. In Section 2, we briefly introduce the statistical generative models applied in learning multi-label text data. Section 3 shows the L-F-L-PAM and presents the generative learning process on multi-label documents, followed by the introduction of the inference algorithm and the parameter estimation method. In Section 4, we conduct experiments to evaluate L-F-L-PAM. Finally, Section 5 concludes the paper.

Table 1
An example of association rules among labels in a real data set. The data set is a subset of Rcv1_v2 with documents IDs from 2286 to 5479. Here, the support threshold is set to 10% and the confidence threshold is 90%.

Association rule	Support (%)	Confidence (%)
{C15→CCAT}	19.6	100
{CCAT,C151→C15}	11.7	100
{M14→MCAT}	10.2	100

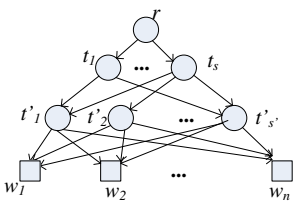


Fig. 1. The DAG structure of L-F-L-PAM.

2. Generative topic models for learning multi-label text data

Statistical generative models have been successfully used to capture the semantic characteristics in text documents. Latent Dirichlet Allocation (LDA) [22] is a widely used topic model, a completely unsupervised algorithm that models each document as a mixture of topics. Some research shows that unsupervised LDA does not perform very well in supervised settings [23]. To address this issue, several modifications of LDA are successively proposed, such as Supervised LDA [24] and DiscLDA [25]. However, these models are inappropriate for multi-label learning because they only allow a single label for each document [26].

For the task of multi-label text classification, several generative topic models have been recently proposed. For example, McCallum [27] proposed a mixture model trained by EM, assuming that a multi-label document is produced by a mixture of the word distributions of its labels, where each label generates different words. Given a new document the most probable set of labels is then selected from the power set of possible classes with Bayes rule. Based on an assumption that multi-labeled text has a mixture of characteristic words appearing in single-labeled text, Parametric Mixture Models (PMM1 and PMM2) [7] are presented, where PMM2 is a more flexible version of PMM1 and explicitly incorporates the pairwise correlation between any two class labels. Ramage et al. [26] proposed Labeled LDA (L-LDA) based on the idea that each word in a document is associated with the most appropriate labels. As a multi-label classifier, L-LDA can trade off label-specific word distributions with document-specific label distributions in quite the same way. It is also shown to be competitive with a strong baseline (multiple one vs-rest SVMs) for multi-label text classification. Another related piece of work is the ColModel proposed by Wang et al. [6], which is a generative probabilistic model employing a multivariate normal distribution to capture the correlation between two labels. Unfortunately, the work discussed above can not capture correlations of multiple labels. In this paper, we present Labeled Four-Level Pachinko Allocation Model which adds one additional latent correlations level based on the Labeled LDA model. The model can capture relations among multiple labels to improve the learning performance on multi-label text data.

3. Labeled Four-Level Pachinko Allocation Model

In this section, we introduce several notations and then give a description of the generative process, inference algorithm and parameter estimation method for the L-F-L-PAM.

First, we define a multi-label training set D consisting of $|D|$ documents, K unique labels and V words, and the model consists of s super-labels in the second level, K labels in the third level and V words at the bottom. α_r is an s-dimensional Dirichlet parameter to characterize the super-label Dirichlet prior distribution under the root r; $\{\alpha_i\}_{i=1}^{s}$ are s K-dimensional Dirichlet parameters to represent the label distributions under super-labels; $\{\beta_j\}_{j=1}^{K}$ are K V-dimensional Dirichlet parameters to express the word distributions under labels. A given labeled document d is represented by a tuple consisting of a list of word indices $\mathbf{w}^{(d)}=\{w_1,\ldots,w_{N_d}\}$ and a list of binary label presence/absence indicators $\mathbf{\Lambda}^{(d)}=\{l_1^{(d)},\ldots,l_K^{(d)}\}$, where $w_i\in\{v_1,\ldots,v_V\}$ and $l_k^{(d)}\in\{0,1\}$. Let $\mathbf{z}^{(d)}=\{z_1,\ldots,z_{N_d}\}$ and

$\mathbf{z}'^{(d)} = \{z'_1, \ldots, z'_{N_d}\}$ be the vectors of all words' super-label assignment and label assignment of the document d, where $z_i \in \{t_1, \ldots, t_s\}$ and $z'_j \in \{t'_1, \ldots, t'_K\}$. In addition, for each label t'_k in the corpus, we define $D_k = \{d | l_k^{(d)} = 1\}$ as the collection of documents that contain the label t'_k. Besides, we assume that the document's labels $\mathbf{\Lambda}^{(d)}$ are generated using a Bernoulli coin toss for each label t'_k, with the labeling prior probability Φ_k that indicates whether the corresponding label is in the given document or not [26].

Conditioned on the model parameters (α,β,$\mathbf{\Phi}$), the graphical representation of the L-F-L-PAM is depicted in Fig. 2. Following the standard probabilistic graphical model formalism [28], each node represents a random variable, and the links express probabilistic relationships between these variables and the number s in a box means the unit in the box is repeated s times. Shaded nodes are observed random variables, unshaded nodes are latent random variables.

The L-F-L-PAM can be viewed in terms of a generative process as shown in Table 2, the main difference from traditional Four-Level Pachinko Allocation Model is reflected in Step 2. For each labeled document d, the label proportions $\theta_{t_i}^{(d)}$, under each super-label t_i, are drawn from $g_i(\alpha_i|\mathbf{\Lambda}^{(d)})$ rather than $g_i(\boldsymbol{\alpha}_i)$. Similarly, in Fig. 2, θ_i is dependent not only on the label prior parameter α_i but also the observed labels $\mathbf{\Lambda}$. Therefore, the label assignment z'_t, in step 3b in Table 2, is restricted to the document's labels $\mathbf{\Lambda}^{(d)}$.

Following this process, we can write the joint distribution of all known and hidden variables given the Dirichlet parameters as follows:

$$p(d,\mathbf{z}^{(d)},\mathbf{z}'^{(d)},\theta^{(d)},\varphi|\alpha,\beta) = p(\mathbf{\Lambda}^{(d)}|\mathbf{\Phi})p(\theta_r^{(d)}|\alpha_r)\prod_{j=1}^{K} p(\varphi_{t'_j}|\beta_j)$$

$$\times \prod_{i=1}^{s} p(\theta_{t_i}^{(d)}|\alpha_i,\mathbf{\Lambda}^{(d)}) \prod_{t=1}^{N_d} (p(z_t|\theta_r^{(d)})p(z'_t|\theta_{z_t}^{(d)})p(w_t|\varphi_{z'_t}))$$

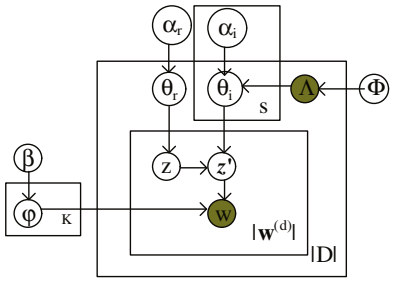


Fig. 2. The graphical model of L-F-L-PAM.

Table 2
Generation process for Labeled Four-Level Pachinko Allocation Model.

1. For each label t'_j, $j \in \{1, \ldots, K\}$
   (a) Generate multinomial distribution over words $\varphi_{t'_j} \sim Dir(.|\beta_j)$
2. For each document d
   (a) For each label t'_k, $k \in \{1, \ldots, K\}$, sample $l_k^{(d)} \in \{0, 1\} \sim Bernoulli(.|\Phi_k)$
   (b) Sample $\theta_r^{(d)}$ from the root $g_r(\alpha_r)$, where $\theta_r^{(d)}$ is a multinomial distribution over super-labels
   (c) For each super-label t_i, sample $\theta_{t_i}^{(d)}$ from $g_i(\alpha_i|\Lambda^{(d)})$, where $\theta_{t_i}^{(d)}$ is a multinomial distribution over labels that appear in the document d
3. For each token word w_t in the document d
   (a) Sample a super-label z_t from $\theta_r^{(d)}$
   (b) Sample a label z'_t from $\theta_{z_t}^{(d)}$
   (c) Sample word w_t from $\varphi_{z'_t}$

And the likelihood of the document d is obtained by integrating over $\theta^{(d)}$, φ and summing over $\mathbf{z}^{(d)}$, $\mathbf{z}'^{(d)}$ as follows:

$$p(d|\alpha,\beta) = \iint p(\mathbf{\Lambda}^{(d)}|\mathbf{\Phi})p(\theta_r^{(d)}|\alpha_r)\prod_{j=1}^{K} p(\varphi_{t'_j}|\beta_j)\prod_{i=1}^{s} p(\theta_{t_i}^{(d)}|\alpha_i,\mathbf{\Lambda}^{(d)})$$

$$\times \prod_{t=1}^{N_d}\sum_{z_t,z'_t}(p(z_t|\theta_r^{(d)}) \times p(z'_t|\theta_{z_t}^{(d)})p(w_t|\varphi_{z'_t}))\, d\theta^{(d)}\, d\varphi$$

Finally, the probability of generating a whole corpus is the product of the probability for every document:

$$p(\mathbf{D}|\alpha,\beta) = \prod_d p(d|\alpha,\beta) \tag{1}$$

In the rest of the section, we will discuss the inference algorithm and parameter estimation method for L-F-L-PAM.

3.1. Inference

Inferring the multinomial distribution parameters θ and φ by directly and exactly maximizing the likelihood of the whole data collection in Eq. (1) is intractable. Fortunately, we find that the Bernoulli prior parameter Φ is d-separated from the rest of the model given $\mathbf{\Lambda}^{(d)}$ since the labels $\mathbf{\Lambda}^{(d)}$ of the document are observed. As a result, the L-F-L-PAM is the same as Four-Level Pachinko Allocation Model, except for the constraint that the label proportions $\boldsymbol{\theta}_{t_i}^{(d)}$ are limited to the labels of the document d. We can also use collapsed Gibbs sampling [29] to perform inference. For each token word in each document d, we just need to jointly sample the super-labels and label assignments because of the same root in all assignment paths [21]. For each position p in the labeled document d, $w_p = v_k$, the sampling probability for each super-label and label pair is as follows:

$$P(z_p = t_i, z'_p = t'_j | D, \mathbf{z}_{-p}, \mathbf{z}'_{-p}, \alpha, \beta) \propto P(w, z_p, z'_p | D_{-p}, \mathbf{z}_{-p}, \mathbf{z}'_{-p}; \alpha, \beta)$$

$$= \frac{n_{ri}^{(d)} + \alpha_{ri}}{n_r^{(d)} + \sum_{i=1}^{s}\alpha_{ri}} \cdot \frac{n_{ij}^{(d)} + \alpha_{ij}}{n_i^{(d)} + \sum_{j=1}^{K}\alpha_{ij}} \cdot \frac{n_{jk} + \beta_{jk}}{n_j + \sum_{t=1}^{V}\beta_{jt}}$$

where the subscript $-p$ indicates all the observations or assignments except the current position p. Excluding the current token, $n_{ij}^{(d)}$ contains the number of times the label t'_j is assigned to some word tokens under super-label t_i in document d; n_i^d is the number of occurrences of the super label t_i in document d; n_j is the number of occurrences of label t'_j in the whole corpus; n_{jk} is the count of word v_k in the label t'_j. The three parts at the right hand side of the equation affect the super-label and label pair assignment for a particular word token in each document. Among them, the left two parts are the probability of super-label t_i and the probability of the label t'_j under super-label t_i in the document d; the right part is the probability of word w_p under label t'_j in the whole corpus. It is worth noting that the sampled labels t'_j for each multi-labeled document d is restricted to be d's (observed) label set.

The only problem left for the inference procedure is, when we are in the testing phase, we may not know exactly which labels are assigned to the given document in advance. So we adopt standard Four-Level Pachinko Allocation Model for unlabeled test set, and also perform Gibbs Sampling. In this paper, we propose pruned Gibbs Sampling algorithm for inferring unlabeled test data. It is based on the idea that if a label is not assigned to any tokens of a word in training set, then it will have low probability to be considered for any token of this word in test documents. To reduce the inference and learning time in the test stage, for each word in the test set, we just consider the labels that have been assigned to this word during training. The entire process of the pruned Gibbs Sampling algorithm is summarized in Table 3.

Table 3
Pruned Gibbs sampling algorithm.

Input: Super labels $T=\{t_1,\ldots,t_s\}$
V label sets $S_k, k\in\{1,\ldots,V\}$. $S_k=\{t'_j|n_{jk}>0\}$, where n_{jk} contains the number of times the word v_k is assigned to the label t'_j in the training set

For every document $\underline{d}$ from the test documents
For every word $\underline{w}_p=v_k$ in document $\underline{d}$
(1) Remove the current super-label and label pair assignment for $\underline{w}_p$
(2) Let $X=T\times S_k$
(3) Sample the new super-label and label pair assignment for $\underline{w}_p$ from X

Given the test set $\underline{D}$, let $\underline{\mathbf{w}}^{(\underline{\mathbf{d}})}$, $\underline{\mathbf{z}}^{(\underline{\mathbf{d}})}$ and $\underline{\mathbf{z}}'^{(\underline{\mathbf{d}})}$ be the vectors of all words, their super-label assignments and label assignments of the test document $\underline{d}$. For each super-label and label pair assignment for each token word $\underline{w}_p=v_k$ in test document $\underline{d}$, the sampling probability depends on the current assignment of all the other words in the test set and the assignments of all words in the training set as follows:

$$P(\underline{z}_p=t_i,\underline{z}'_{-p}=t'_j|\underline{D},\underline{\mathbf{z}}_{-p},\underline{\mathbf{z}}'_{-p},D,\mathbf{z},\alpha,\beta)$$
$$=\frac{n_{ri}^{(\underline{d})}+\alpha_{ri}}{n_r^{(\underline{d})}+\sum_{i=1}^{s}\alpha_{ri}}\times\frac{n_{ij}^{(\underline{d})}+\alpha_{ij}}{n_i^{(\underline{d})}+\sum_{j=1}^{K}\alpha_{ij}}\times\frac{\underline{n}_{jk}+n_{jk}+\beta_{jk}}{\underline{n}_j+n_j+\sum_{t=1}^{V}\beta_{jt}}$$

where the new notation $\underline{n}_{jk}$ indicates the number of times that the word v_k is assigned to the label t'_j and $\underline{n}_j$ is the number of times the label t'_j is sampled in the test documents $\underline{D}$.

After sampling, the label distribution of the document $\underline{d}$ is $\vartheta^{(\underline{d})}=\{\vartheta^{(\underline{d})}_{t'_1},\ldots,\vartheta^{(\underline{d})}_{t'_K}\}$, each distribution component of which denotes the confidence score of assigning the corresponding label to document $\underline{d}$, and is computed as follows:

$$\vartheta^{(\underline{d})}_{t'_j}=\sum_{i=1}^{s}\frac{n_{ri}^{(\underline{d})}+\alpha_{ri}}{n_r^{(\underline{d})}+\sum_{i=1}^{s}\alpha_{ri}}\cdot\frac{n_{ij}^{(\underline{d})}+\alpha_{ij}}{n_i^{(\underline{d})}+\sum_{j=1}^{K}\alpha_{ij}} \quad (2)$$

3.2. Parameter estimation

In L-F-L-PAM, the Dirichlet parameter α_{ri} can be interpreted as a prior observation count for the number of times super-label t_i is sampled for a document. The hyperparameter β_{jk} can be interpreted as the prior observation count on the number of times the word v_k is sampled from the label t'_j before any word from the document collection is observed. In this paper, we assume a fixed symmetric Dirichlet distribution for the root such that $\alpha_{r1}=\cdots=\alpha_{rs}$, and similarly for $\beta_{jk}(j=1,\ldots,K;\ k=1,\ldots,V)$.

The parameter α_i can be interpreted as the number of times one label is sampled for a document under the super-label t_i. It captures different correlations among labels, so it is necessary to estimate the Dirichlet parameter $\boldsymbol{\alpha}_i$ for each super-label t_i. Following the choice of the Four-Level Pachinko Allocation Model, smoothing moment matching [30] is exploited to learn the super-label Dirichlet parameters. In each iteration of Gibbs sampling, α_{ij}, the jth component in $\boldsymbol{\alpha}_i$, is updated according to the following rules:

$$mean_{ij}=\frac{1}{N_i}\cdot\sum_{d\in D_j}\frac{n_{ij}^{(d)}}{n_i^{(d)}}$$

$$var_{ij}=\frac{1}{N_i}\cdot\sum_{d\in D_j}\left(\frac{n_{ij}^{(d)}}{n_i^{(d)}}-mean_{ij}\right)^2$$

$$m_{ij}=\frac{mean_{ij}\cdot(1-mean_{ij})}{var_{ij}}-1$$

$$\alpha_{ij}=\frac{mean_{ij}}{\exp\left(\frac{\sum_{j=1}^{K}\log(m_{ij})}{K-1}\right)}$$

where d is from the collection D_j instead of D, since just word tokens from document collection D_j can be assigned to label t'_j.

4. Experiment evaluation

In this section, we will apply our proposed approach for multi-label learning problem on the Rcv1_v2 as well as 8 Yahoo! data sets and evaluate it based on evaluation metrics of class ranking, comparing with several state-of-the-art multi-label learning methods, including Ranking by pairwise comparison (PRC) [31], BP-MLL [10], and L-LDA. The only factor that causes difference between our model and L-LDA is that our model has an additional level that can formulate the correlation of several labels. The comparison based on these two models can illustrate the significance of incorporating the correlation of multiple labels in multi-label learning.

4.1. Data preparation

Rcv1_v2: Rcv1_v2 [32] text data set[1] is news stories collected by Reuters and organized by three different category sets: Topics, Industries, and Regions. It has been widely used as a benchmark data set to evaluate text classification [1]. We choose the first 6000 documents with document IDs from 2286 to 8584 and consider the Topics category set in our experiments. In this paper, the first 3000 documents with their labels are used as the training set and the rest as the test set.

Eight Yahoo data sets: Each data set contains web pages collected from one of Yahoo!'s top-level categories, and each page is labeled with one or more second level sub-categories. More details about the Yahoo! multi-label web page categorization data set[2] are given in [7,33,34]. The eight data sets used in our experiment are Computers & Internet, Education, Entertainment, Health, Recreation, Reference, Science, Society. In our experiments, we use the original training and test subsets provided in the releases of the eight data sets.

The details of the nine data sets are given in Table 4. Note that labels that appear in the test set but not in the training set are ignored. "LC" (Label cardinality) is the average number of labels of the examples in training set and is used to quantify the number of alternative labels that characterize the examples of a multi-label training data set.

4.2. Evaluation metrics

Since our approach only produces a ranked list of class labels for a test document, four ranking measures are employed for all the algorithms in this paper. They are specially designed for multi-label learning and proposed in [9]. Given a multi-label test set $\underline{D}$, the details of the four metrics are as below. Here, the set of labels $P_{\underline{d}}$ are called relevant for the given instance $\underline{d}$, the set $N_{\underline{d}}$ are the irrelevant labels; $\tau(\underline{d},\lambda)$ denotes the rank of the label λ in the predicted ranking for a given instance $\underline{d}$, $\tau^{-1}(\underline{d},k)$ is a label that is assigned to rank k.

(1) Average precision (avgprec) evaluates the average fraction of labels ranked above a particular label $\lambda\in P_{\underline{d}}$ which actually is

[1] URL: http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm.
[2] URL: http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz.

in $P_{\underline{d}}$; the bigger the value, the better the performance:

$$avgprec(\tau)=\frac{1}{|\underline{D}|}\sum_{\underline{d}\in\underline{D}}\frac{1}{|P_{\underline{d}}|}\sum_{\lambda\in P_{\underline{d}}}\frac{|\{\lambda'\in P_{\underline{d}}\,|\,\tau(\underline{d},\lambda')\le\tau(\underline{d},\lambda)\}|}{\tau(\underline{d},\lambda)} \tag{3}$$

(2) Ranking loss (rloss) evaluates the average fraction of pairs of relevant label and irrelevant label that are not correctly ordered; the smaller the value, the better the performance:

$$rloss(\tau)=\frac{1}{|\underline{D}|}\sum_{\underline{d}\in\underline{D}}\frac{|\{(\lambda,\lambda')\in P_{\underline{d}}\times N_{\underline{d}}:\tau(\underline{d},\lambda)>\tau(\underline{d},\lambda')\}|}{|P_{\underline{d}}|\cdot|N_{\underline{d}}|} \tag{4}$$

(3) One-error computes how many times the top-ranked label is not relevant for the instance; the smaller the value, the better the performance.

$$one\text{-}error(\tau)=\frac{1}{|\underline{D}|}\sum_{\underline{d}\in\underline{D}}\left\|\tau^{-1}(\underline{d},1)\notin P_{\underline{d}}\right\| \tag{5}$$

Here $\|\pi\|$ equals 1 if π holds and 0 otherwise.

(4) Coverage evaluates how many steps are need, on average, to move down the label list in order to cover all the relevant labels of the instance; the smaller the value, the better the performance:

$$coverage(\tau)=\frac{1}{|\underline{D}|}\sum_{\underline{d}\in\underline{D}}\max_{\lambda\in P_{\underline{d}}}\tau(\underline{d},\lambda)-1 \tag{6}$$

Table 4
Characteristics of data sets. "#Training set" indicates the number of instances in the training set; "#Test set" indicates the number of instances of the test set; "#Label" indicates the number of labels in the training set.

Data set	#Training set	#Test set	#Label	LC
Rcv1_v2	3000	3000	95	3.27
Computers	6270	6170	32	1.52
Education	6030	6000	33	1.46
Entertainment	6356	6374	21	1.41
Health	4557	4648	29	1.64
Recreation	6471	6357	22	1.43
Reference	4027	3999	32	1.66
Science	3214	3214	39	1.47
Society	7273	7239	26	1.68

Here the coverage metric is normalized according to [14] so that all the four metrics vary between [0, 1].

The four evaluation metrics defined above measure the ranking performance of multi-label learning approach from different aspects. In general, it is difficult for one multi-label learning algorithm to outperform another algorithm in terms of all the four measures.

4.3. Experimental results and discussion

4.3.1. The parameters setting

We use the same settings of Dirichlet parameters for the root and label as discussed in [21]. For the root, each component of the Dirichlet parameter vector is 0.01. The multinomial distributions for labels are sampled once for the whole corpus from a given Dirichlet with parameter 0.01. The estimation process for the super-label Dirichlet parameters is depicted in Section 3.2.

We employ the class perplexity to estimate the effect of the number of super-label factors. Class perplexity is used by convention in language modeling and can be thought as the inverse of the geometric mean per-class likelihood, a better generalization capability is indicated by a lower class perplexity over the held-out testing samples [6]. For a test set $\underline{D}$, the perplexity is as follows:

$$perplexity(\underline{D})=\exp\left\{\frac{-\sum_{\underline{d}\in\underline{D}}\sum_{t_j^{\underline{d}}\in P_{\underline{d}}}\log p(t_j^{\underline{d}}|\mathbf{w}^{(\underline{d})})}{\sum_{\underline{d}\in\underline{D}}P_{\underline{d}}}\right\} \tag{7}$$

As is well known, the common method to evaluate perplexity in topic models is to hold out test data from the corpus to be trained and then test the estimated model on the held-out data. Here, for each data set, we first ordered the original training set by document ID, then held out the last 25% for test purposes and trained the model on the remaining 75%. In the cases where there are some labels not present in the remaining 75%, for each of these labels, we move one sample containing the label from the held-out test set to the training set. Fig. 3 shows the class perplexity with different number of super-labels in all data sets. In the following experiments, we will pick the number of super-labels that produces the best result.

In order to make the comparison more meaningful, we also learn the hyperparameters in L-LDA. Minka's fixed-point iteration

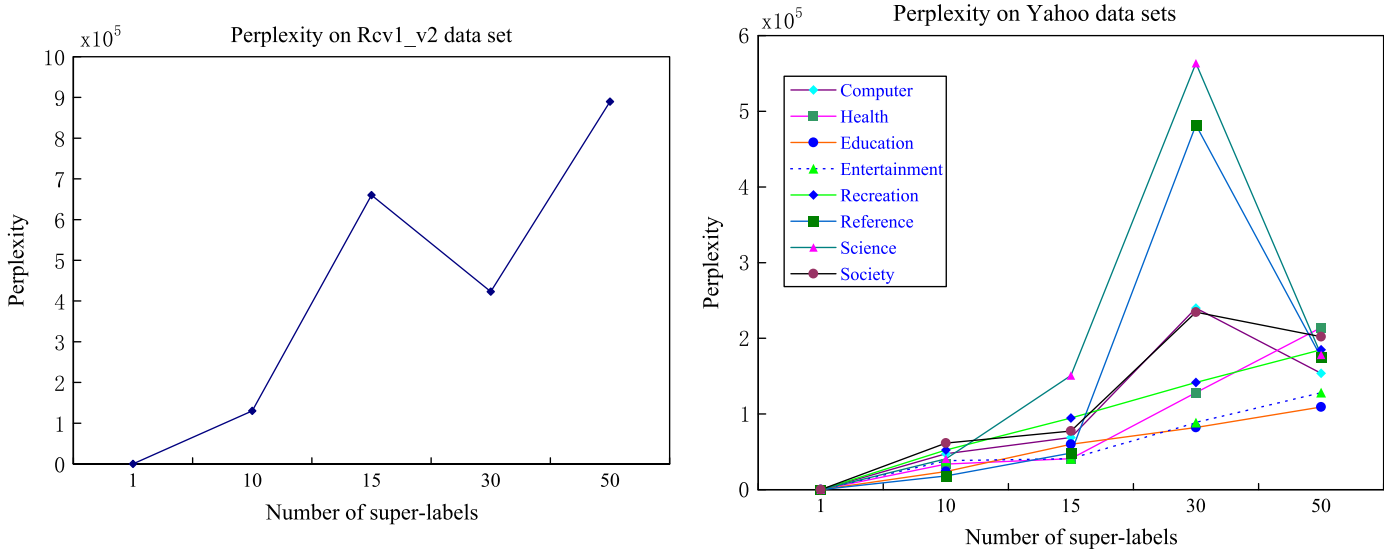


Fig. 3. Class Perplexity on the number of super-labels. The left panel illustrates the perplexity on the RCV1_v2 data set and the right panel illustrates the perplexity on the Yahoo! data sets.

technique, which is most widely used for learning hyperparameters in LDA, is applied here. In this paper, each step of fixed-point iteration is formalized as follows:

$$\alpha^* \leftarrow \frac{\alpha \sum_{k=1}^{K} \sum_{d \in D_k} [\Psi(n_k^{(d)} + \alpha) - \Psi(\alpha)]}{K \sum_{d \in D_k} [\Psi(\sum_{k=1}^{K} n_k^{(d)} + K\alpha) - \Psi(K\alpha)]}$$

$$\beta^* \leftarrow \frac{\beta \sum_{k=1}^{K} \sum_{n=1}^{V} [\Psi(n_{kn} + \beta) - \Psi(\beta)]}{V \sum_{k=1}^{K} [\Psi(\sum_{n=1}^{V} n_{kn} + V\beta) - \Psi(V\beta)]}$$

It is slightly different from that used in LDA [35]. Since in Labeled LDA each label k is just sampled in document collection D_k. We initialize the hyperparameters as $\alpha = 1.0$, $\beta = 0.01$ and turn on Minka's updates after 20 loops. The best hyperparameter settings based on the training set are in Table 5.

In Gibbs sampling for L-F-LPAM and L-LDA, we use 800 burn-in iterations, then draw five samples in the following 200 iterations during training, and 100 iterations and one sample during test. Each Gibbs sampler is initialized randomly. For BP-MLL, the number of hidden neurons is set to 20% of the dimensionality and the number of training epoches is set to 100 [10]. Libsvm (with linear kernel) [36] are used as the base classifier for RPC.

4.3.2. A comparison of results

The experimental results in terms of different metrics are reported in Tables 6–9, where the rank of each method in each data set is shown in bold face in the parentheses, the average rank of each method in terms of each metric is shown in the last line of each table. In the case of a tie, ranks are added together and divided by the number of ties. We observe that our method improves substantially over L-LDA on the all data sets in terms of each metric and it is not directly clear how to compare our method with the other baselines. Following the suggestions in [37], we compare the different methods according to their average rank.

Table 5
Parameters selection results of L-LDA.

Data set	Best α	Best β
Rcv1_v2	0.002	0.011
Computer	0.001	0.088
Education	0.001	0.103
Entertainment	0.002	0.138
Health	0.002	0.082
Recreation	0.002	0.121
Reference	0.001	0.104
Science	0.001	0.078
Society	0.002	0.130

Table 6
Performance of each algorithm in terms of average precision on the all data sets.

Data set	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Rcv1_v2	0.667(**2**)	0.387(**4**)	0.783(**1**)	0.577(**3**)
Computers	0.689(**1**)	0.440(**4**)	0.667(**2**)	0.645(**3**)
Education	0.586(**1**)	0.380(**4**)	0.582(**2**)	0.562(**3**)
Entertainment	0.713(**1.5**)	0.576(**4**)	0.705(**3**)	0.713(**1.5**)
Health	0.761(**2**)	0.582(**4**)	0.766(**1**)	0.695(**3**)
Recreation	0.674(**1**)	0.593(**4**)	0.637(**3**)	0.657(**2**)
Reference	0.707(**2**)	0.559(**4**)	0.755(**1**)	0.637(**3**)
Science	0.634(**1**)	0.502(**4**)	0.625(**2**)	0.535(**3**)
Society	0.605(**3**)	0.484(**4**)	0.607(**2**)	0.618(**1**)
Average rank	1.611	4.000	1.889	2.500

Table 7
Performance of each algorithm in terms of ranking loss on the all data sets.

Data set	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Rcv1_v2	0.087(**2**)	0.795(**4**)	0.305(**3**)	0.057(**1**)
Computers	0.084(**2**)	0.330(**4**)	0.168(**3**)	0.072(**1**)
Education	0.082(**2**)	0.354(**4**)	0.169(**3**)	0.074(**1**)
Entertainment	0.060(**1**)	0.061(**2**)	0.063(**3**)	0.082(**4**)
Health	0.054(**1**)	0.285(**4**)	0.089(**3**)	0.056(**2**)
Recreation	0.079(**1**)	0.169(**4**)	0.121(**3**)	0.111(**2**)
Reference	0.068(**2**)	0.270(**4**)	0.075(**3**)	0.067(**1**)
Science	0.117(**2**)	0.350(**4**)	0.177(**3**)	0.098(**1**)
Society	0.107(**1**)	0.213(**4**)	0.186(**3**)	0.115(**2**)
Average rank	1.556	3.778	3.000	1.667

Table 8
Performance of each algorithm in terms of one-error on all the data sets.

Data set	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Rcv1_v2	0.116(**1.5**)	0.392(**3**)	0.116(**1.5**)	0.434(**4**)
Computers	0.374(**2**)	0.523(**4**)	0.368(**1**)	0.473(**3**)
Education	0.528(**2**)	0.612(**3**)	0.457(**1**)	0.672(**4**)
Entertainment	0.375(**2**)	0.466(**4**)	0.372(**1**)	0.384(**3**)
Health	0.255(**2**)	0.384(**3**)	0.241(**1**)	0.478(**4**)
Recreation	0.405(**1**)	0.446(**3**)	0.412(**2**)	0.450(**4**)
Reference	0.390(**2**)	0.508(**3**)	0.303(**1**)	0.529(**4**)
Science	0.453(**2**)	0.554(**3**)	0.425(**1**)	0.631(**4**)
Society	0.470(**2**)	0.609(**4**)	0.386(**1**)	0.482(**3**)
Average rank	1.833	3.333	1.167	3.667

Table 9
Performance of each algorithm in terms of coverage on all the data sets.

Data set	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Rcv1_v2	0.226(**3**)	0.479(**4**)	0.178(**2**)	0.147(**1**)
Computers	0.127(**2**)	0.408(**4**)	0.193(**3**)	0.113(**1**)
Education	0.107(**2**)	0.418(**4**)	0.183(**3**)	0.101(**1**)
Entertainment	0.130(**3**)	0.329(**4**)	0.124(**2**)	0.116(**1**)
Health	0.107(**2**)	0.426(**4**)	0.138(**3**)	0.100(**1**)
Recreation	0.154(**2**)	0.294(**4**)	0.206(**3**)	0.152(**1**)
Reference	0.078(**1**)	0.274(**4**)	0.085(**3**)	0.083(**2**)
Science	0.126(**1**)	0.319(**4**)	0.171(**3**)	0.135(**2**)
Society	0.194(**2**)	0.339(**4**)	0.284(**3**)	0.181(**1**)
Average rank	2.000	4.000	2.778	1.222

The average rank on all the comparing algorithms in terms of different evaluation criterions is summarized in Table 10, along with the average rank of each method. We observe that L-F-L-PAM exhibits the highest average rank in the average precision and rank loss measures, the second best average rank in terms of one-error and coverage. Over all metrics, L-F-L-PAM has the highest average rank.

In addition, the Wilcoxon signed-rank test (two-tailed at $p=5\%$) was applied in order to examine if L-F-L-PAM is statistically significant over the rest of the methods in terms of different metrics. Over all data sets, whenever L-F-L-PAM achieves significantly better/similar/worse performance than the competing algorithm, a win/tie/loss is counted. The resulting win/tie/loss counts for L-F-L-PAM against the competing algorithms are summarized in Table 11. We can observe that L-F-L-PAM performs

Table 10
The average rank of each method in each evaluation metric, along with the average rank of each method.

Metric	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Avgprec	1.611	4.000	1.778	2.611
Rank loss	1.556	3.778	3.000	1.667
One-error	1.833	3.333	1.167	3.667
Coverage	2.000	4.000	2.778	1.222
Average rank	1.750	3.778	2.208	2.264

Table 11
The win/tie/loss results for L-F-L-PAM against the compared algorithms in terms of different evaluation metrics.

Metric	L-F-L-PAM against		
	L-LDA	RPC	BP-MLL
Avgprec	1/0/0	0/1/0	1/0/0
Rank loss	1/0/0	1/0/0	0/1/0
One-error	1/0/0	0/0/1	1/0/0
Coverage	1/0/0	1/0/0	0/1/0
In total	4/0/0	2/1/1	2/2/0

Table 12
The computational time including the training phase and test phase of each multi-label learning algorithm on all the data sets, measured in hours.

Data set	Algorithm			
	L-F-L-PAM	L-LDA	RPC	BP-MLL
Rcv1_v2	1.01	0.14	2.35	13.54
Computers	1.79	0.45	9.12	26.24
Education	1.57	0.31	7.96	25.21
Entertainment	1.66	0.50	5.91	26.29
Health	0.98	0.30	3.62	19.05
Recreation	1.41	0.59	8.53	30.23
Reference	1.44	0.54	3.44	17.61
Science	1.43	0.48	4.59	13.47
Society	2.46	0.85	13.43	30.39

significantly better than BP-MLL and L-LDA in terms of average rank and one-error, significantly better than RPC and L-LDA in terms of rank loss and coverage, and just significantly worse than RPC in terms of one-error.

Table 12 reports the computational costs consumed by each multi-label learning algorithm on the all data sets. As shown in Table 12, L-F-L-PAM consumes much less time than BP-MLL and RPC on all data sets while being slower than L-LDA.

5. Conclusion

In this paper, we try to address data mining for document classification with a special treatment from multi-label learning perspective. To this end, we propose a generative model, called Labeled Four-Level Pachinko Allocation Model (L-F-L-PAM), which has a latent correlation level to formulate the latent relations of multiple labels. Furthermore, we propose a pruned Gibbs Sampling algorithm in the test stage to reduce the inference time. The empirical results show that the proposed method outperforms other state-of-the-art baselines in terms of effectiveness. This shows the evidences that it is necessary to consider the relations of multiple labels in the multi-label learning problem. In addition, our method is superior in terms of computational efficiency compared with some other high-performing multi-label learning methods.

Indeed, our model can be generalized to other multi-label learning problems, not narrowly restricted to text data. For example, scene categorization in image processing and gene function annotation in bioinformatics, where the objects of scenes or genes can be treated as documents in multi-label document classification, the features representing the objects correspond to the words at the bottom, and class labels in images or functional classes in genes work as labels at the third level, respectively. In this way, the correlations among multiple labels can be captured by the super-labels in the second level.

Finally, real-world applications of multi-label learning often feature a large number of classes and a relatively small size of training data. All the multi-label learning approaches exploiting relations between labels may suffer from the problem that the learned relations among labels are not consistent with the actual situation. Thus, we will also try to use large scale external data sources for learning the structures of multi-label relations offline.

Acknowledgments

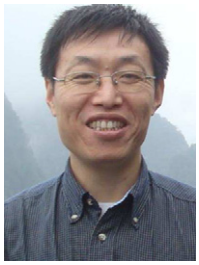
Many thanks to Qi Liu, Tengfei Bao, and Biao Xiang for their valuable discussion and suggestions. The work described in this paper was supported by grants from Natural Science Foundation of China (Grant No. 61073110, 60775037, 61003135), the Key Program of National Natural Science Foundation of China (Grant No. 60933013), the National Major Special Science & Technology Project (Grant No. 2011ZX04016-071), and Research Fund for the Doctoral Program of Higher Education of China (20093402110017).

References

[1] B. Yang, J.-T. Sun, T. Wang, Z. Chen, Effective multi-label active learning for text classification, in: Proceedings of the KDD '09, Paris, France, June 28–July 1, 2009, pp. 917–926.
[2] X.-B. Xue, Z.-H. Zhou, Distributional features for text categorization, IEEE Trans. Knowl. Data Eng. 21 (3) (2009) 428–442.
[3] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of label, in: Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08).
[4] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the ECML, Chemnitz, Germany, April 21–24, 1998, pp. 137–142.
[5] Y. Yang, An evaluation of statistical approaches to text categorization, Inf. Retr. 1 (1–2) (1999) 69–90.
[6] H. Wang, M. Huang, X. Zhu, A generative probabilistic model for multi-label classification, in: Proceedings of the ICDM, Pisa, Italy, December 15–19, 2008, pp. 628–637.
[7] N. Ueda, K. Saito, Parametric mixture models for multi-labeled text, in: Proceedings of the NIPS, 2002, pp. 721–728.
[8] S. Zhu, X. Ji, W. Xu, Y. Gong, Multi-labelled classification using maximum entropy method, in: Proceedings of the SIGIR, Salvador, Brazil, August 15–19, 2005, pp. 274–281.
[9] R.E. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, IEEE Trans. Knowl. Mach. Learn. 39 (2–3) (2000) 135–168.
[10] M.-L. Zhang, Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.
[11] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Proceedings of the NIPS, 2001, pp. 681–687.
[12] J. Furnkranz, E. Hullermeier, E. Loza Mencia, Multi-label classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.
[13] N. Ghamrawi, A. McCallum, Collective multi-label classification, in: Proceedings of the CIKM'05, pp. 195–200.
[14] M.-L. Zhang, K. Zhang, Multi-label learning by exploiting label dependency, in: Proceedings of the KDD, Washington, DC, USA, July 25–28, 2010, pp. 999–1008.
[15] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, Adv. Knowl. Discovery Data Mining 3056 (2004) 22–30.

[16] S. Ji, L. Tang, S. Yu, J. Ye, Extracting shared subspace for multi-label classification, in: Proceedings of the KDD, Las Vegas, Nevada, USA, August 24–27, 2008, pp. 381–389.
[17] R. Yan, J. Tesic, J.R. Smith, Model-shared subspace boosting for multi-label classification, in: Proceedings of the KDD, San Jose, California, USA, August 12–15, 2007, pp. 834–843.
[18] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Proceedings of the ECML/PKDD (2), 2009, pp. 254–269.
[19] G. Tsoumakas, I. Vlahavas, Random k-Labelsets: an ensemble method for multilabel classification, in: Proceedings of the ECML '07, Warsaw, Poland, September 17–21, 2007, pp. 406–417.
[20] S.-h. Park, J. Frnkranz, Multi-label classification with label constraints, in: Proceedings of the Proceedings of the ECML/PKDD-08 Workshop on Preference Learning (PL-08), 2008, pp. 157–171.
[21] W. Li, A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations, in: Proceedings of the ICML '06, Pennsylvania, USA, June 25–29, 2006, pp. 577–584.
[22] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[23] V. Krishnan, Shortcomings of latent models in supervised settings, in: Proceedings of the SIGIR05, Salvador, Brazil, August 15–19, 2005, pp. 625–626.
[24] D. Blei, J. McAuliffe, Supervised topic models, in: Proceedings of the NIPS, 2007.
[25] S. Lacoste-Julien, F. Sha, M.I. Jordan, DiscLDA: discriminative learning for dimensionality reduction and classification, in: Proceedings of the NIPS, 2008, pp. 897–904.
[26] D. Ramage, D. Hall, R. Nallapati, C.D. Manning, Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora, in: Proceedings of the EMNLP '09, 2009, pp. 248–256.
[27] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: Proceedings of the AAAI'99 Workshop on Text Learning, Orlando, FL, 1999.
[28] W.L. Buntine, Operations for learning with graphical models, J. Artif. Intell. Res. 2 (1994) 159–225.
[29] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. USA 101 (1) (2004) 5228–5235.
[30] G. Casella, R. Berger. Statistical Inference. Duxbury Resource Center, 2001.
[31] E. Hüllermeier, J. Fürnkranz, W. Cheng, K. Brinker, Label ranking by learning pairwise preferences, Artif. Intell. 172 (16–17) (2008) 1897–1916.
[32] D.D. Lewis, Y. Yang, T.G. Rose, G. Dietterich, F. Li, RCV1: a new benchmark collection for text categorization research, J. Mach. Learn. Res. 5 (2004) 361–397.
[33] N. Ueda, K. Saito, Single-shot detection of multiple categories of text using parametric mixture models, in: Proceedings of the KDD, 2002, pp. 626–631.
[34] H. Kazawa, H. Taira, T. Izumitani, E. Maeda, Maximal margin labeling for multi-topic text categorization, Joho Shori Gakkai Kenkyu Hokoku 93 (2004) 53–60.
[35] Q. Liu, E. Chen, H. Xiong, C.H.Q. Ding, Exploiting user interests for collaborative filtering: interests expansion via personalized ranking, in: Proceedings of the ACM CIKM'10, 2010, pp. 1697–1700.
[36] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001, Software available at 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉.
[37] S. Garca, F. Herrera, An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2009) 2677–2694.

Haiping Ma received her B.Sc. degree in computer science from Anhui University, Hefei, China, in 2008. Currently she is a Ph.D. candidate in the Department of Computer Science & Technology at university of Science and Technology of China and a member of Data Mining group. Her main research interests include machine learning and data mining, especially in multi-label learning and Topic Model.



Enhong Chen is a professor and a vice dean at the School of Computer Science and Technology, University of Science and Technology of China, IEEE Senior Member. He received his doctor degree in USTC in 1996. He was selected in Program for New Century Excellent Talents in University (supported by Chinese Ministry of Education) in 2005. He is the vice-chair of Multimedia Computing and Communication Ministry of Education – Microsoft key Laboratory. He serves as a councilor of Machine Learning Society of Chinese Association for Artificial Intelligence, and a councilor of Artificial Intelligence and Pattern Recognition Society of the Database Society of Chinese Computer Federation. Prof. Chen has been actively involved in the research community by serving as a PC member for more than 30 conferences, such as ICTAI 2006, ICTAI 2007, AIRS2009, AIRS2010, KDD2010. One of his paper received Best Application Paper Award in KDD2008. His research interests include semantic web, machine learning and data mining, web information processing, constraint satisfaction problem.

Linli Xu received her Ph.D. degree in Computer Science from the University of Waterloo in 2007, B.E. degree in Computer Science from the University of Science and Technology of China (USTC). She is currently an Associate Professor at the School of Computer Science and Technology, University of Science and Technology of China. Her main research areas are machine learning and data mining, more specifically in unsupervised learning and semi-supervised learning, large margin approaches, optimization, and convex programming.

Hui Xiong is currently an Associate Professor in the Management Science and Information Systems department at Rutgers University. He received B.E. degree from the University of Science and Technology of China, China, the M.S. degree from the National University of Singapore, Singapore, and the Ph.D. degree from the University of Minnesota, USA. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published over 90 technical papers in peer-reviewed journals and conference proceedings. He is a co-editor of Clustering and Information Retrieval (Kluwer Academic Publishers, 2003) and a co-Editor-in-Chief of Encyclopedia of GIS (Springer, 2008). He is an Associate Editor of the Knowledge and Information Systems journal and has served regularly in the organization committees and the program committees of a number of international conferences and workshops. He is a senior member of the IEEE, and a member of the ACM.