Received 15 October 2014; revised 17 November 2014; accepted 21 November 2014.
Date of publication 3 December, 2014; date of current version 10 June, 2015.

Digital Object Identifier 10.1109/TETC.2014.2377559

Label Correlation Mixture Model: A Supervised Generative Approach to Multilabel Spoken Document Categorization

ZHIYANG HE[1], JI WU[1], AND TAO LI[2]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]School of Computer Science, Florida International University, Miami, FL 33199 USA

CORRESPONDING AUTHOR: J. Wu (wuji_ee@mail.tsinghua.edu.cn) and T. Li (taoli@cs.fiu.edu)

The work of Z. He and J. Wu was supported in part by the Planned Science and Technology Project through Tsinghua University, Beijing, China, under Grant 20111081023, in part by the National High-Tech Research and Development Program (863 Program) of China under Grant 2012AA011004, in part by the National Natural Science Foundation of China under Grant 61170197, and in part by the Electronic Information Industry Development Fund through the Project entitled Research and Development and Industrialization on Information Retrieval System Based on Man-Machine Interaction with Natural Speech. The work of T. Li was supported by the U.S. National Science Foundation under Grant DBI-0850203, Grant HRD-0833093, Grant CNS-1126619, and Grant IIS-121302.

ABSTRACT Multilabel categorization, which is more difficult but practical than the conventional binary and multiclass categorization, has received a great deal of attention in recent years. This paper proposes a novel probabilistic generative model, label correlation mixture model (LCMM), to depict the multiply labeled documents, which can be used for multilabel spoken document categorization as well as multilabel text categorization. In LCMM, labels and topics have the one-to-one correspondences. The LCMM consists of two important components: 1) a label correlation model and 2) a multilabel conditioned document model. The label correlation model formulates the generating process of labels where the dependences between the labels are taken into account. We also propose an efficient algorithm for calculating the probability of generating an arbitrary subset of labels. The multilabel conditioned document model can be regarded as a supervised label mixture model, in which labels for a document are known. Each label is characterized by distributions over words. For the parameter learning of the multilabel conditioned document model, in addition to maximum-likelihood estimation, a discriminative approach based on the minimum classification error rate training is proposed. To evaluate LCMM, extensive multilabel categorization experiments are conducted on a spoken document data set and three standard text data sets. The experimental results in comparison with other competitive methods demonstrate the effectiveness of LCMM.

INDEX TERMS Label correlation mixture model, probabilistic generative model, multi-label spoken document categorization, multi-label text categorization, Bayesian decision theory, minimum classification error rate method.

I. INTRODUCTION

Spoken document categorization can be considered as a special text categorization problem. The general text categorization is an important and basic problem in the natural language processing field. Suppose that $\mathbf{D}$ is an observed data set and $\mathbf{Y}$ is a label set with K labels. The text categorization problem is how to build an optimal text classifier based on a certain learning criterion:

$$F : \mathbf{d} \rightarrow \mathbf{y}. \quad \mathbf{d} \in \mathbf{D}, \mathbf{y} \in \mathbf{Y}.$$

The traditional text categorization is a single label classification problem, in which the label $\mathbf{y}$ is a single discrete value. When $K = 2$, which means $\mathbf{y}$ is a binary value, such as 0 or 1, it is the conventional binary classification; when $K > 2$, it is the multi-class classification. However, in the real world, a text may belong to more than one category. For instance, a financial news article possibly not only belongs to the "economy" category but also belongs to the "politics" category; and a spoken document relevant to a conversation between peoples may contain more than one topic. This is

a multi-label text categorization problem, in which $\mathbf{y}$ is set-valued. In recent years, many research efforts have been focused on this more difficult but practical text categorization problem, which has been indispensable for many recent applications.

In order to solve the multi-label text classification problem, many approaches have been proposed. They can be broadly grouped into two main categories [1], [14]: *problem transformation methods* and *algorithm adaptation methods*. The problem transformation methods, such as classifier chain [2], label power-set [3], and maximal figure-of-merit [4], [5], transform the multi-label classification problem into one or more single-label classification problems. The algorithm adaptation methods modify and extend the existing algorithm to directly solve the multi-label problem. Typical algorithm adaptation methods include Boostexter [6], multi-label KNN [7], and multi-label neural networks [8].

According to the Bayesian decision theory of minimum error rate case, given a document $\mathbf{d}_{test}$, the goal of multi-label classification is to find the optimal subset of labels which has the maximum posterior probability. Based on the Bayes' rule, we have

$$\begin{aligned}\hat{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{d}_{test}) \\ &= \operatorname*{argmax}_{\mathbf{y}} \frac{P(\mathbf{d}_{test}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{d}_{test})} \\ &= \operatorname*{argmax}_{\mathbf{y}} P(\mathbf{d}_{test}|\mathbf{y})P(\mathbf{y}). \end{aligned} \tag{1}$$

According to Eq. (1), the probabilistic generative model can be adopted and the key points are how to calculate the conditional probability $P(\mathbf{d}_{test}|\mathbf{y})$ and the prior $P(\mathbf{y})$. Due to the inherent ability of modeling document data, it is a natural idea to utilize the methodology of topic model to deal with the text categorization problem. The traditional topic models were proposed to model text and other discrete collective data. The popular models include probabilistic latent semantic analysis (PLSA) [10], Latent Dirichlet allocation (LDA) [11] and their variations, in which each document is modeled as a mixture over a set of topics. However, these unsupervised topic models can not be directly used for classification task.

In recent years, many supervised approaches of topic models have been proposed for the multi-label text categorization problem. A mixture model approach was proposed in [9] for multi-label text classification, in which each label was regarded as a class and modeled by a word distribution. The class mixture model was used to calculate $P(\mathbf{d}_{test}|\mathbf{y})$. This model is similar to PLSA except for the manners of updating parameters. The parameters of this mixture model can be trained by the maximum likelihood estimation. However, the approach directly used the frequency of each label to estimate the corresponding probability $P(\mathbf{y})$, which is not appropriate for many real applications since there are not enough data to estimate the priors of all the different classes. Labeled LDA was proposed in [12] as a direct way to train the parameters of the LDA model with supervised label information. Compared with the LDA model, labeled LDA has an additional Bernoulli sampling process to depict the fact that only a subset of labels associate with one document. One weakness of labeled LDA is that the correlations between labels were not considered. CoL-model [13] was also a supervised LDA version which focused on formulating the correlations between labels. The formulation was carried out through sampling labels according to a multinomial distribution whose parameters were drawn from a Normal distribution with full covariance matrix. However, in the classification phase, neither labeled LDA nor CoL-model can effectively evaluate an arbitrary subset of labels for a testing document. The reason is that these models can not provide a prior for a subset to calculate the joint probability in Eq. (1). Instead, they directly provided a subset using a threshold in terms of the relative probability of a single label. This strategy was easy for implementation but often limited the classification performance. Dependency LDA was another supervised LDA approach proposed in [15]. Dependency LDA obtains a subset of labels for a document by first sampling from the topic distributions over labels, and then determining the label distributions by using the label frequency. The dependencies between labels are not explicitly modeled. In addition, for a testing document after training, the stochastic sampling procedure must be performed to obtain appropriate labels. As a result, the approach is not efficient.

In order to model the multi-labeled document data, there are three primary questions need to be addressed:

- How to incorporate the supervised label set into the learning procedure?
- How to formulate the correlations of labels?
- How to effectively evaluate an arbitrary subset of labels for a testing document?

In this paper, we propose a probabilistic generative model, *label correlation mixture model* (LCMM), to address the aforementioned problems. In LCMM, we define the one-to-one correspondences between *labels* and *topics*, so the two terms are interchangeable in the rest of this paper. According to LCMM, the generating process of a labeled document consists of two phases, which correspond to two models: a label correlation model and a multi-label conditioned document model (or a document model for short hereafter). In the first phase, labels are generated based on a stochastic process where the correlations between labels are taken into account. In the second phase, the documents are generated based on the generated labels, which are characterized by distributions over words. The second phase can be seen as a *supervised* label mixture model, in which the dependencies between labels are also involved. In the classification stage, given a testing document $\mathbf{d}_{test}$, the prior $P(\mathbf{y})$ of an arbitrary subset $\mathbf{y}$ and the conditional probability $P(\mathbf{d}_{test}|\mathbf{y})$ can be calculated by the label correlation model and the document model, respectively.

The details of our approach are presented in the rest of the paper. Section II describes the LCMM framework which

consists of a label correlation model and a document model. The parameter estimation and inference for these models are discussed in Section III and Section IV, respectively. The multi-label classification strategy is described in Section V. Section VI presents the experimental results, and Section VII concludes the paper.

II. THE GENERAL FORM OF LABEL CORRELATION MIXTURE MODEL

Let the text document corpus be represented by $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M\}$, with words from a vocabulary $\mathbf{W} = \{w_1, w_2, \ldots, w_N\}$. Suppose the label set $\mathbf{Y}$ has K labels, $\mathbf{Y} = \{y_1, y_2, \ldots, y_K\}$. λ denotes K dimensional parameter of a multinomial distribution for the size of a subset. φ is also a K dimensional parameter of a multinomial distribution, which depicts the probabilities of randomly selecting a certain single label. Ξ denotes a set of parameters for formulating the correlations between labels.

LCMM assumes the following generating process for each document $\mathbf{d}$ in the corpus $\mathbf{D}$:

S1. Generating a label subset size $L \sim Multinomial(\lambda)$.

S2. Generating a label subset $\mathbf{y}$ with L labels for the document:

a) Sample the first label $\bar{y}_1 \sim Multinomial(\varphi), \bar{y}_1 \in \mathbf{Y}$, $\mathbf{y} = \{\bar{y}_1\}$.

b) For each label $\bar{y}_l, l \in [2, L]$: Sample $\bar{y}_l$ from $\mathbf{Y} - \mathbf{y}$ with probability $P(\bar{y}_l|\mathbf{y}, \Xi)$, then, $\mathbf{y} = \mathbf{y} \bigcup \{\bar{y}_l\}$.

S3. Generating all the words in the document $\mathbf{d}$ based on the subset $\mathbf{y}$. Suppose there are V words in the document, and for each word:

a) Sample a label z_k with probability $P(z_k|\mathbf{y}), z_k \in \mathbf{y}$.

b) Sample a word w_n from $P(w_n|z_k)$, which is a label-conditional multinomial probability of the word w_n.

The LCMM model is represented as a probabilistic graphical model in Figure 1.

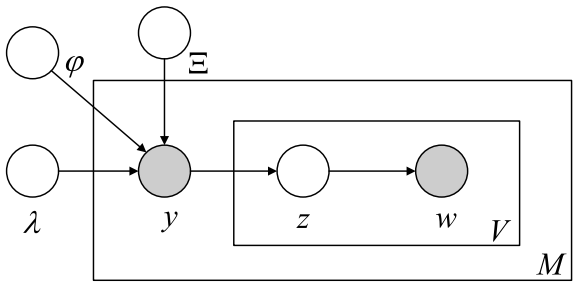


FIGURE 1. Graphical model representation for label correlation mixture model, in which the labels and words are both observed variables.

Given the parameters λ, φ and Ξ, the generating process of document $\mathbf{d}$ and its labels can be translated into a joint probability, which has the expression as follows:

$$P(\mathbf{y}, \mathbf{d}|\lambda, \varphi, \Xi) = P(\mathbf{d}|\mathbf{y})P(\mathbf{y}|\lambda, \varphi, \Xi). \tag{2}$$

In the above equation, $P(\mathbf{y}|\lambda, \varphi, \Xi)$ is called the label correlation model and $P(\mathbf{d}|\mathbf{y})$ is called the document model.

The label correlation model depicts the stochastic generating process of labels for a document, which are described in S1 and S2. Because there exist $L!$ different orders (or sequences) of the labels to generate a subset $\mathbf{y}$, all these orders should be summed over to calculate the probability $P(\mathbf{y}|\lambda, \varphi, \Xi)$, that is

$$P(\mathbf{y}|\lambda, \varphi, \Xi) = P_\lambda(L) \cdot \sum_{S_L \in \Phi_\mathbf{y}} P(S_L|\varphi, \Xi), \tag{3}$$

where $\Phi_\mathbf{y}$ denotes all the possible orders of labels in $\mathbf{y}$:

$$\Phi_\mathbf{y} = \{(\bar{y}_{\pi(1)}, \bar{y}_{\pi(2)}, \ldots, \bar{y}_{\pi(L)}) | (\pi(1), \pi(2), \ldots, \pi(L)) \in \{\text{all the permutations of the integers from 1 to L}\}\}. \tag{4}$$

Suppose $S_L = (\bar{y}'_1, \bar{y}'_2, \ldots, \bar{y}'_L)$, we have

$$P(S_L|\varphi, \Xi) = P_\varphi(\bar{y}'_1) \cdot P(\bar{y}'_2|\{\bar{y}'_1\}, \Xi) \cdot \ldots \cdot P(\bar{y}'_L|\{\bar{y}'_1, \bar{y}'_2, \ldots, \bar{y}'_{L-1}\}, \Xi). \tag{5}$$

After generating the label subset $\mathbf{y}$, the words are sampled in S3, which corresponds to the document model. In this step, the approach we used is similar to PLSA except for the restriction of only using the selected labels in S2. The conditional probability of all the words in document $\mathbf{d}$ can be calculated by

$$P(\mathbf{d}|\mathbf{y}) = \prod_{n=1}^{N} (\sum_{z_k \in \mathbf{y}} P(w_n|z_k) \cdot P(z_k|\mathbf{y}))^{\bar{n}(\mathbf{d}, w_n)}, \tag{6}$$

where $\bar{n}(\mathbf{d}, w_n)$ is the number of w_n in document $\mathbf{d}$.

The above process can be interpreted as follows. For generating a document, we first make a decision that L labels (or topics) will be involved. Then, we select L labels in turn and each label is chosen based on the selected labels. After that, the words of the document are generated and each word is probably related to any selected label.

III. LABEL CORRELATION MODEL

An accurate estimation of the prior $P(\mathbf{y}|\lambda, \varphi, \Xi)$ is important but difficult for multi-label classification. One important reason is the difficulty of formulating the relationships between labels. The label correlation model provides an effective strategy to solve the problem as described below.

A. PARAMETER ESTIMATION: λ AND φ

Two multinomial distributions with parameters λ and φ are adopted in S1 and S2. λ is K dimensional vector $(\lambda_1, \lambda_2, \ldots, \lambda_K)$, in which $\lambda_k (k \in [1, K])$ represents the probability of containing k labels for a subset and $\sum_1^K \lambda_k = 1$. λ_k can be estimated by m_k/M, where m_k is the count of subsets that contain k labels in the training data set and $M = \sum_1^K m_k$. φ is a K dimensional vector $(\varphi_1, \varphi_2, \ldots, \varphi_K)$, where $\varphi_k (k \in [1, K])$ is the probability

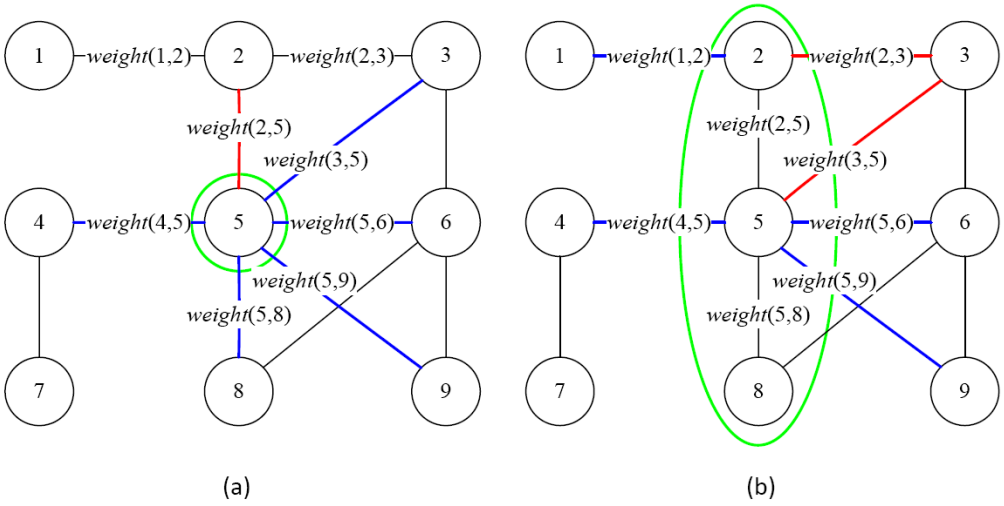


FIGURE 2. Examples of label correlation networks and the calculation of $P(\mathbf{y} \to \bar{y})$. In subgraph (a), the conditional probability $P(\{5\} \to 2)$ can be calculated by $P(\{5\} \to 2) = \frac{\kappa(\{5\},2)}{\sum_{\bar{y}' \in \{2,3,4,6,8,9\}} \kappa(\{5\},\bar{y}')} = \frac{\xi(5,2)}{\sum_{\bar{y}' \in \{2,3,4,6,8,9\}} \xi(5,\bar{y}')}$. As for subgraph (b), $P(\{2, 5, 8\} \to 3) = \frac{\kappa(\{2,5,8\},3)}{\sum_{\bar{y}' \in \{1,3,4,6,9\}} \kappa(\{2,5,8\},\bar{y}')} = \frac{\xi(2,3)+\xi(5,3)}{\sum_{\bar{y}' \in \{1,3\}} \xi(2,\bar{y}')+\sum_{\bar{y}' \in \{3,4,6,9\}} \xi(5,\bar{y}')+\sum_{\bar{y}' \in \{6\}} \xi(8,\bar{y}')}$.

of hitting the k^{th} label and $\sum_1^K \varphi_k = 1$. φ_k in this paper is estimated by s_k/S, where s_k is the count of the k^{th} single label in the training data set and $S = \sum_1^K s_k$.

B. LABEL CORRELATION NETWORK

The key of LCMM is the S2-b in the generating process of Section II. The label selection involves the distribution $P(\bar{y}_l|\mathbf{y}, \Xi)$. However, because the number of power set of $\mathbf{Y}$ is an exponential function of K in theory, it is impractical to estimate the distributions for all the possible subsets. Here we propose a novel approach to estimate these conditional probabilities, by which the probability of generating an arbitrary subset can be effectively calculated.

We first define a *label correlation network* using an undirected graph. In a label correlation network, each vertex represents a single label, each edge between two vertexes indicates the correlation of the two corresponding labels, and there is a weight on each edge representing a certain measure of the correlation. Co-occurrence is a typical kind of representation for the correlation between two labels. In this paper the frequency of co-occurrence of two labels is chosen as the weight on the edge between the corresponding two vertexes. We can then build a label correlation network from all the observed multiple labels of the training data.

The basic idea is that the labels are generated based on a label correlation network. That is to say, the conditional probability in S2-b of the generating process can be calculated based on the correlations between the relevant labels, which are all included in the label correlation network. Therefore, in this case, Ξ represents all the weights in the label correlation network, $\Xi = \{\xi(k,k')|k \neq k', k \in [1,K], k' \in [1,K]\}$, where $\xi(k,k')$ is the weight on the edge between the two vertexes which correspond to label y_k and label $y_{k'}$. $\widetilde{\mathbf{y}}$ denotes the complementary set of a label subset $\mathbf{y}$, $\widetilde{\mathbf{y}} = \mathbf{Y} - \mathbf{y}$. $\mathbf{y}^*$ denotes the candidate set of $\mathbf{y}$, $\mathbf{y}^* = \{\bar{y}|\exists e(y,\bar{y}), y \in \mathbf{y}, \bar{y} \in \widetilde{\mathbf{y}}\}$, $e(y,\bar{y})$ is the edge between vertexes which correspond to label y and label $\bar{y}$, respectively. In other words, $\mathbf{y}^*$ includes all the labels that connects to at least one label in $\mathbf{y}$. Given a subset $\mathbf{y}$, suppose that we will select another label $\bar{y}$. Our basic assumption is that the label $\bar{y}$ should be chosen from $\mathbf{y}^*$, i.e. $\bar{y} \in \mathbf{y}^*$. Every label, which belongs to $\mathbf{y}^*$, has a probability to be chosen. This probability is relevant to both $\mathbf{y}$ and $\mathbf{y}^*$. Given $\mathbf{y}$, $P(\mathbf{y} \to \bar{y})$ denotes the conditional probability of choosing a label $\bar{y}$ in $\mathbf{y}^*$ given $\mathbf{y}$ and can be calculated by

$$P(\mathbf{y} \to \bar{y}) = \frac{\kappa(\mathbf{y},\bar{y})}{\sum_{\bar{y}' \in \mathbf{y}^*} \kappa(\mathbf{y},\bar{y}')} = \frac{\sum_{y \in \mathbf{y}} \xi(y,\bar{y})}{\sum_{\bar{y}' \in \mathbf{y}^*} \sum_{y \in \mathbf{y}} \xi(y,\bar{y}')}, \bar{y} \in \mathbf{y}^* \quad (7)$$

where $\kappa(\mathbf{y},\bar{y}) = \sum_{y \in \mathbf{y}} \xi(y,\bar{y})$. We also have $\sum_{\bar{y} \in \mathbf{y}^*} P(\mathbf{y} \to \bar{y}) = 1$. It should be pointed out that in the generating process of S2-b each label is sampled from a multinomial distribution. Figure 2 shows two examples of label correlation networks and how to calculate the conditional probability of sampling a new label according to Eq. (7).

C. INFERENCE FOR A LABEL SET

According to Eq. (7), the conditional probabilities in Eq. (5) can be rewriten as

$$P(S_L|\varphi, \Xi) = P_\varphi(\bar{y}'_1) \cdot P(\{\bar{y}'_1\} \to \bar{y}'_2) \cdot \ldots \cdot P(\{\bar{y}'_1, \bar{y}'_2, \ldots, \bar{y}'_{L-1}\} \to \bar{y}'_L). \quad (8)$$

It can be calculated based on the label correlation network. We define the *sampling probability* $\widehat{P}(\mathbf{y})$ of a label subset $\mathbf{y}$ as the second term of the RHS of Eq. (3), i.e., $\widehat{P}(\mathbf{y}) = \sum_{S_L \in \Phi_\mathbf{y}} P(S_L|\varphi, \Xi)$. In theory, $\widehat{P}(\mathbf{y})$ can be calculated by summing over all the $L!$ possible label

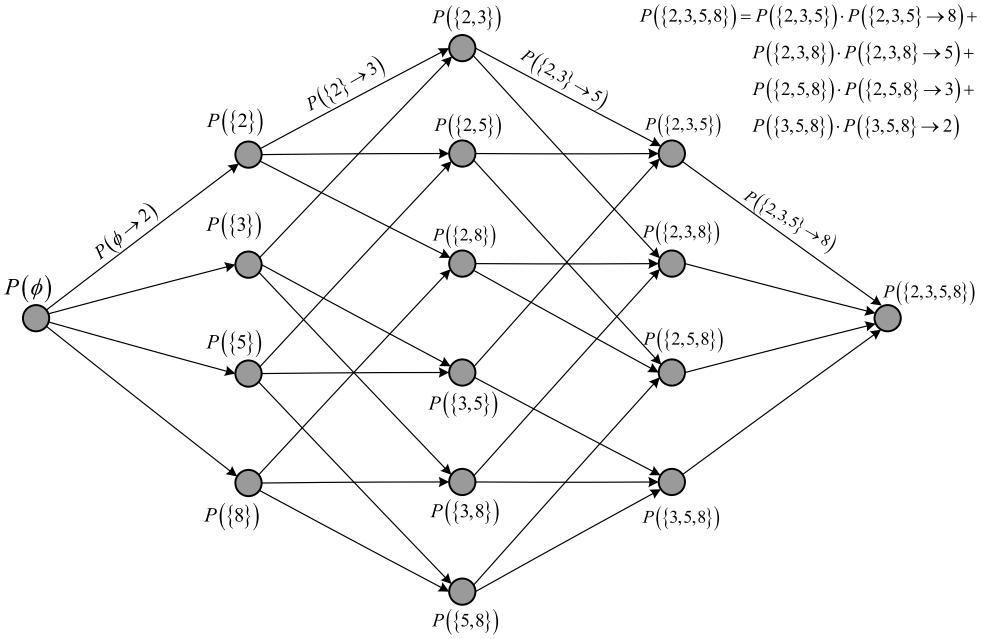


FIGURE 3. An example for the process of calculating sampling probability. This can be displayed by a finite state machine with trellis structure. In this example, $\mathbf{y} = \{2, 3, 5, 8\}$.

sequences' probabilities. However, according to the Stirling's approximation, $L! \approx \sqrt{2\pi L} \cdot (L/e)^L$, the total computational requirements are on the order of $L \cdot \sqrt{2\pi L} \cdot (L/e)^L$ operations. This complexity greatly influences the computational efficiency. Fortunately, there is an equivalent algorithm for calculating $\widehat{P}(\mathbf{y})$, which is similar with the well known *forward algorithm* [16], [17] and can easily alleviate the expensive computational requirement.

We use a finite state machine to illustrate this algorithm. Suppose that the sampling probability of subset $\mathbf{y}$ will be calculated. As for the finite state machine, the states have one-to-one correspondence to the subsets of $\mathbf{y}$. There are two special states: one corresponds to the empty set and is used as the initial state; and the other one corresponds to $\mathbf{y}$ itself and is used as the final state. We use $\mathbf{s_e}$ to represent the state which corresponds to the subset $\mathbf{e}$, $\mathbf{e} \subset \mathbf{y}$. For two subsets $\mathbf{e}'$ and $\mathbf{e}$, if

$$|\mathbf{e}'| + 1 = |\mathbf{e}|, \quad \mathbf{e}' \subset \mathbf{e},$$

then, we call $\mathbf{e}'$ the *precursor set* of $\mathbf{e}$. In the finite state machine, there exists a transition from each precursor set of $\mathbf{e}$ to $\mathbf{s_e}$. We define the transition score as the conditional probability of generating $\mathbf{e}$ when $\mathbf{e}'$ is given, which can be calculated according to the Eq. (7). This finite state machine can be displayed by a directed acyclic graph (DAG). In this DAG, a complete path from the initial state $\mathbf{s}_\phi$ to the final state $\mathbf{s_y}$ corresponds to a label sequence of $\mathbf{y}$ and the path score which is the product of all the transition scores is equal to the probability of generating the corresponding label sequence, i.e. Eq. (8). Therefore, $\widehat{P}(\mathbf{y})$ can be calculated by summing over all the path scores, which is just from the definition of the sampling probability without complexity constraints. Figure 3 displays an example of DAG for a finite state machine, which has a trellis structure. We furthermore define the *state score* as the sampling probability of the corresponding subset. From Figure 3 we can see that the state score can be calculated inductively by

$$\widehat{P}(\mathbf{e}) = \sum_{\mathbf{e}'} \widehat{P}(\mathbf{e}') \cdot P(\mathbf{e}' \to (\mathbf{e} - \mathbf{e}')). \tag{9}$$

In other words, the sampling probability of a subset can be obtained through the sampling probabilities of its precursor set. This is also true for $\mathbf{y}$, and the equivalent formula is

$$\widehat{P}(\mathbf{y}) = \sum_{y \in \mathbf{y}} \widehat{P}(\mathbf{y} \setminus y) \cdot P(\mathbf{y} \setminus y \to y). \tag{10}$$

The above equation ignores the conditions λ, φ and Ξ for brevity. The calculation of $\widehat{P}(\mathbf{y})$ does not require summing over all the path scores. This divide-and-conquer machinery significantly alleviates the computational complexity.

The upper bound of the operation requirement can be reduced to $L/\sqrt{2\pi L} \cdot 2^L$. Typically the actual number of labels for a document in practice is not very large. So the size L of each subset $\mathbf{y}$ can also be controlled, which is relevant to the multi-label classification strategy discussed in Section V. Therefore, the exponential upper bound of the proposed algorithm can be acceptable in most cases.

IV. MULTI-LABEL CONDITIONED DOCUMENT MODEL

A. MAXIMUM LIKELIHOOD ESTIMATION

The multi-label conditioned document model is similar to PLSA. However, an obvious difference between the two models is that the labels of a document are restricted to belonging

to the document's label set. Therefore, the document model can be regarded as a supervised label mixture model. The re-estimation equations for parameters in EM algorithm are:

$$P(z_k|w_n, \mathbf{y}_i) = \frac{P(w_n|z_k)P(z_k|\mathbf{y}_i)}{\sum_{z_l \in \mathbf{y}_i} P(w_n|z_l)P(z_l|\mathbf{y}_i)}, z_k \in \mathbf{y}_i. \quad (11)$$

$$P(z_k|\mathbf{y}_i) = \frac{\sum_n \overline{n}(\mathbf{d}_i, w_n)P(z_k|w_n, \mathbf{y}_i)}{\overline{n}(\mathbf{d}_i)}, z_k \in \mathbf{y}_i. \quad (12)$$

$$P(w_n|z_k) = \frac{\sum_i \overline{n}(\mathbf{d}_i, w_n)P(z_k|w_n, \mathbf{y}_i)}{\sum_n \sum_i \overline{n}(\mathbf{d}_i, w_n)P(z_k|w_n, \mathbf{y}_i)}. \quad (13)$$

$\overline{n}(\mathbf{d}_i, w_n)$ is the number of w_n in document $\mathbf{d}_i$, $\overline{n}(\mathbf{d}_i)$ is the number of all the words in document $\mathbf{d}_i$. The equations above are exactly the same as the PLSA except that label z_k of a document $\mathbf{d}_i$ must belong to the document's label set $\mathbf{y}_i$.

B. MCE CLASSIFIER DESIGN

The maximum likelihood estimation approach may not guarantee a minimum classification error rate performance. There are two key reasons: (1) the mismatch between the chosen model hypothesis and the actual data distribution; (2) the inadequacy of the training data. In order to solve these problems, many criteria of discriminative training approaches have been proposed over the past few years. Multi-label categorization is a typical classification problem, therefore, we propose the training approach based on minimum classification error (MCE) method [18]–[20] to learn the parameters of document model discussed in Section IV-A. It should be pointed out that the MLE model is used as the initial model for our learning approach. During the learning approach, only the word distributions conditioned on labels (Eq. (13)) will be updated, and meanwhile, the document-conditional probabilities (Eq. (12)) for each training document will be the same as the final probabilities of MLE training. We denote these parameters to be updated by $\Lambda = \{P(w_n|z_k)|n \in [1, N], k \in [1, K]\}$.

We first define two kinds of *discriminant functions* for document $\mathbf{d}_i$ in the training data set as follows.

$$g_{\mathbf{y}_i}(\mathbf{d}_i; \Lambda) = \sum_n \overline{n}(\mathbf{d}_i, w_n) \cdot \log[\sum_{z_k \in \mathbf{y}_i} P(z_k|\mathbf{y}_i)P(w_n|z_k)] \quad (14)$$

$$g_j(\mathbf{d}_i; \Lambda) = \sum_n \overline{n}(\mathbf{d}_i, w_n) \cdot \log P(w_n|z_j), z_j \in \widetilde{\mathbf{y}}'_i \subseteq \widetilde{\mathbf{y}}_i \quad (15)$$

Eq. (14) calculates the *log-likelihood* for document model with the label set $\mathbf{y}_i$ of document $\mathbf{d}_i$. This is different from the traditional classification. Eq. (15) calculates the *log-likelihood* for the parameters of the single label that does not belong to the label set $\mathbf{y}_i$. $\widetilde{\mathbf{y}}_i$ is complementary set of $\mathbf{y}_i$, i.e. $\widetilde{\mathbf{y}}_i = \mathbf{Y} - \mathbf{y}_i$. $\widetilde{\mathbf{y}}'_i$ is a subset of $\widetilde{\mathbf{y}}_i$ and can be regarded as the competing categories.

Based on these discriminant functions, the *misclassification measure* takes the following form:

$$d(\mathbf{d}_i) = -g_{\mathbf{y}_i}(\mathbf{d}_i; \Lambda) + \log[\frac{1}{|\widetilde{\mathbf{y}}'_i|} \sum_j \exp\{g_j(\mathbf{d}_i; \Lambda) \cdot \eta\}]^{1/\eta}, \quad z_j \in \widetilde{\mathbf{y}}'_i \quad (16)$$

where $|\widetilde{\mathbf{y}}'_i|$ represents the size of the subset $\widetilde{\mathbf{y}}'_i$, and η is a positive number. $d(\mathbf{d}_i)$ is a continuous measure of Λ and approximately quantifies the separation between the correct label subset and the competing labels.

We also need to choose a loss function, which is smooth and approximates the 0-1 step function. The misclassification measure will then be embedded in this function. The general form of the *loss function* can be defined as:

$$\ell(\mathbf{d}_i; \Lambda) = \ell(d(\mathbf{d}_i)) \quad (17)$$

where $\ell(\cdot)$ is a sigmoid function:

$$\ell(d) = \frac{1}{1 + e^{-\gamma d + \theta}} \quad (18)$$

In Eq. (18), γ is a constant and influences the learning rating of parameters. θ is also a constant which can be seen as an offset of d from 0. Finally, based on the loss function, the overall *objective function* of the multi-label text categorization problem can be defined as:

$$L(\Lambda) = \sum_{i=1}^{M} \ell(\mathbf{d}_i; \Lambda). \quad (19)$$

C. MCE OPTIMIZATION

The objective function of Eq. (19) can be optimized to find a suitable set of parameters. As for traditional MCE approaches, the generalized probabilistic descent (GPD) [18], [21], [22] algorithm has been proved to be powerful. However, for document model, all the parameters are conditional probabilities, which maintain the constraints: $P(w_n|z_k) \geq 0$, $\sum_n P(w_n|z_k) = 1$, $n \in [1, N]$, $k \in [1, K]$. The GPD algorithm is an unconstrained approach, which is not appropriate to be directly adopted to optimize the parameters. In our approach, we use another optimization method, growth transformation [23]–[26], which can naturally solve the above problem. Based on growth transformation approach, the parameter set Λ, can be updated with the following equation:

$$\overrightarrow{P}(w_n|z_r) = \frac{P(w_n|z_r)(\frac{\partial L(\Lambda)}{\partial P(w_n|z_r)} + T)}{\sum_{n'} P(w_{n'}|z_r)(\frac{\partial L(\Lambda)}{\partial P(w_{n'}|z_r)} + T)}, \quad n \in [1, N], n' \in [1, N], r \in [1, K] \quad (20)$$

where

$$\frac{\partial L(\Lambda)}{\partial P(w_n|z_r)} = \sum_{i=1}^{M} \frac{\partial \ell(\mathbf{d}_i; \Lambda)}{\partial P(w_n|z_r)} = \sum_{i=1}^{M} \frac{\partial \ell(\mathbf{d}_i; \Lambda)}{\partial d(\mathbf{d}_i)} \cdot \frac{\partial d(\mathbf{d}_i)}{\partial P(w_n|z_r)} \quad (21)$$

$$\frac{\partial \ell(\mathbf{d}_i; \Lambda)}{\partial d(\mathbf{d}_i)} = \gamma \cdot \ell(\mathbf{d}_i; \Lambda)[1 - \ell(\mathbf{d}_i; \Lambda)] \quad (22)$$

$$\frac{\partial d(\mathbf{d}_i)}{\partial P(w_n|z_r)} = \begin{cases} \frac{-\overline{n}(\mathbf{d}_i, w_n)P(z_r|\mathbf{y}_i)}{\sum_{z_k \in \mathbf{y}_i} P(z_k|\mathbf{y}_i)P(w_n|z_k)}, & z_r \in \mathbf{y}_i. \\ \frac{\exp\{g_r(\mathbf{d}_i; \Lambda) \cdot \eta\}}{\sum_j \exp\{g_j(\mathbf{d}_i; \Lambda) \cdot \eta\}} \cdot \frac{\overline{n}(\mathbf{d}_i, w_n)}{P(w_n|z_r)}, & z_r \in \widetilde{\mathbf{y}}'_i. \end{cases} \quad (23)$$

In Eq. (20), T is a constant which controls the convergence speed of the objective function $L(\Lambda)$ and $\overrightarrow{P}(w_n|z_r)$ is the updated parameter corresponding to $P(w_n|z_r)$. There exists a value T^* such that $T \geq T^*$ will guarantee the growth of $L(\Lambda)$ [23], therefore, γ in Eq. (22) is a negative constant in the sigmoid function of Eq. (18).

D. FOLD-IN PROCESS

Given a new document and a label subset, the fold-in process can be performed, which is the same as that of PLSA. The EM algorithm can be adopted again to obtain the conditional probability (Eq. (12)) of each given label, while the word distributions conditioned on labels (Eq. (13)) are not changed during the fold-in process.

V. MULTI-LABEL CLASSIFICATION STRATEGY

We present the framework of our multi-label classification approach based on label correlation model and document model as follows.

Given a new document, in order to find the best subset of $\mathbf{Y}$, all the possible label subsets should be assessed. However, the number of all the subsets is $2^K - 1$, which makes it impractical to implement, even when K is not very large. We use a simple greedy strategy with the following three steps to solve this problem:

- Step-a: Carry out the fold-in process to calculate the conditional probability $P(z|\mathbf{y})$ for each label in the whole label set (i.e. $\mathbf{y} = \mathbf{Y}$) according to Eq. (11) and (12).
- Step-b: Discard the labels whose conditional probability is lower than a threshold.
- Step-c: Compare all the possible subsets that are limited to being chosen from the remainder labels to explore the best subsets according to Eq. (1), Eq. (2), Eq. (3), and Eq. (6).

This strategy is reasonable because when a document is forced to associate with all the labels, the true labels belong to the document should have relatively higher conditional probabilities. Our experimental results verify that this greedy strategy works well.

VI. EXPERIMENTS

A. DATA SETS

We use four data sets to evaluate LCMM. The details of data sets are presented in Table 1.

The first one is called MSD-TCS, which is a Mandarin spoken document data set. Each document corresponds to a telephone conversation between a caller and a call-center staff in a telecom customer service system. Each document contains one or more topics, such as "3G service", "music", "SMS", "cancel", "download", etc. Our task is to assign one or more topics to a given document, which can be formulated into a multi-label text categorization problem. Both the training and testing data are recognition text, which are from a large vocabulary continuous speech recognition (LVCSR) system with an average word accuracy 73.4%. The dictionary

TABLE 1. Data sets collection and statistics. #trn is the number of training documents. #tst indicates the number of testing documents. #lbl is the number of labels. l-card indicates label cardinality [1], which is the average number of labels relevant to each document.

data set	*#trn*	*#tst*	*#lbl*	*l-card*
MSD-TCS	9747	1083	109	2.75
Reuters-top10	6490	2544	10	1.10
Reuters-top36	7543	2906	36	1.19
Reuters-all90	7768	3019	90	1.24
TMC2007	21519	7077	22	2.16
OHSUMED	6965	6964	23	1.67

including 15969 words used in the LVCSR system is also utilized as the original vocabulary for text categorization. Besides the training set and testing set presented in Table 1, there exists a validation set with 812 documents for tuning the constant parameters in our approach which will be described in Section IV-C.

We also use three standard text classification data sets downloaded from Web to evaluate the performance of LCMM: Reuters-21578 [27], TMC2007 [28] and OHSUMED [29]. Reuters-21578 is a collection of articles appeared in the Reuters newswire in 1987. We use the Mod-Apte split as descried in the "README" file accompanying the original set. This data set is named Reuters-all90 in this paper, which contains 90 classes. Two subsets of Reuters-all90 are also used for experiments. They only use the 10 and 36 largest classes [5], [9], [13], which are called Reuters-top10 and Reuters-top36 respectively. TMC2007 is a data set about aviation safety reports which record the problems occurred during flights. OHSUMED includes medical abstracts from the MeSH categories of the year 1991.

B. COMPARISON APPROACHES AND EVALUATION MEASURES

LCMM is compared with other reported multi-label classification approaches: Mixture Model (Mix-Model) [9], CoL-Model [13], naive Bayes (NB) [1], [13], ML-KNN [7], [13], [34], ML-SVM [13], [32]–[34], Multinomial Model (Mul-Model) [33], Boostexter [6], [13], MFoM-Bin [4], MFoM-MC [5], Rocchio method (Roc-Method) [32], KNN [32], fuzzy similarity KNN (FS-KNN) [34], Ensemble-CC [2], Ensemble-BM [2], Ensemble-PS [2], [3], RAKEL [2], [35].

As for LCMM, the MCE training procedure for the document model is adopted. For illustrating the performance improvement that stems from the prior based on the label correlation model, the performances of LCMM without prior (LCMM-NoPr) are also evaluated. LCMM-NoPr ignores the calculation of the prior $P(\mathbf{y})$ or treats the prior $P(\mathbf{y})$ as a constant in the classification process. In addition, the approaches with MLE training procedure (LCMM-MLE and LCMM-MLE-NoPr) for the document model is also evaluated on the MSD-TCS data set for comparison.

Various kinds of evaluation measures are adopted in this paper: *accuracy*, *precision*, *recall*, *F-Score*, *micro/macro*

precision (Micro/Macro P), *micro/macro recall* (Micro/Macro R), *micro/macro F-Score* (Micro/Macro F). Tsoumakas and Vlahavas [1], Chai et al. [36] described the details of these evaluation measures.

C. EXPERIMENTAL SETUP AND RESULT ANALYSIS

Simple preprocessing is carried out on each data set. The meaningless tags in each document are removed, and all the words are lowercased and the words on a standard stop word list of about 400 words are removed.

The competing labels in Eq. (15) are obtained through Step-a and Step-b described in Section V. After step-b, the subset of the remaining labels, except for the manual labels, are used as the competing set, i.e., $\widetilde{\mathbf{y}}_i'$ in Eq. (15). η in Eq. (16) is fixed at 2.0, γ in Eq. (18) is -1.0, θ is -1.0. The iteration number of MCE training is set to be 50. As for the stage of classification, the strategy described in Section V is adopted. The thresholds of Step-b in the learning and classification stages are set to be 0.12. These parameters result in the best performance on the validation set of MSD-TCS and are chosen for the following experiments on all the testing sets.

We first compare LCMM with LCMM-MLE on the MSD-TCS data set and the experimental results are shown in Tables 2 and 3. The tables illustrate that the performance measures of MCE training approach is about 4.6% better on average than those of the MLE training approach for all the evaluation measures, demonstrating the effectiveness of the MCE training approach. In the following experiments, we only provide the performance measures of LCMM approach compared with other reported methods. Since these approaches were evaluated by different measures and the data sets were also different, we list the corresponding performance values in Tables 4–7 and the complete experimental results for all the evaluation measures are included in Appendix A.

TABLE 2. Performance comparison on MSD-TCS.

Approach	Accuracy	Precision	Recall	F-Score
LCMM-MLE-NoPr	0.617	0.740	0.697	0.707
LCMM-MLE	0.712	0.813	0.766	0.780
LCMM-NoPr	0.678	0.794	0.747	0.759
LCMM	0.734	0.840	0.778	0.799

TABLE 3. Performance comparison on MSD-TCS.

Approach	Micro P	Micro R	Micro F
LCMM-MLE-NoPr	0.674	0.778	0.722
LCMM-MLE	0.783	0.805	0.794
LCMM-NoPr	0.744	0.812	0.777
LCMM	0.839	0.769	0.803
Approach	**Macro P**	**Macro R**	**Macro F**
LCMM-MLE-NoPr	0.742	0.661	0.699
LCMM-MLE	0.815	0.733	0.772
LCMM-NoPr	0.791	0.708	0.747
LCMM	0.840	0.737	0.785

With regard to the performance measures on Reuters (shown in Tables 4 and 5), LCMM outperforms most of the reported approaches on precision, recall and F-Score measures. According to micro F-Score and macro F-Score measures, the comparison shows that the performance values

TABLE 4. Performance comparison on Reuters.

Approach	Precision	Recall	F-Score
Reuters-top10			
Mix-Model	0.839	-	-
CoL-Model	0.901	0.923	0.898
LCMM-NoPr	0.800	**0.973**	0.848
LCMM	**0.940**	0.949	**0.932**
Reuters-top36			
NB	0.751	0.892	0.803
ML-KNN	0.795	0.797	0.791
ML-SVM	0.878	0.814	0.848
CoL-Model	0.872	0.875	0.876
LCMM-NoPr	0.835	0.864	0.829
LCMM	**0.882**	**0.911**	**0.883**
Reuters-all90			
Mul-Model	0.852	0.720	-
Boostexter	-	-	0.851
CoL-Model	0.867	**0.873**	0.866
LCMM-NoPr	0.818	0.830	0.805
LCMM	**0.878**	0.872	**0.870**

TABLE 5. Performance comparison on Reuters.

Approach	Micro F	Macro F
Reuters-top10		
Roc-Method	0.839	0.681
KNN	0.852	0.855
ML-SVM	0.926	0.857
MFoM-Bin	0.933	0.883
MFoM-MC	**0.937**	**0.884**
LCMM-NoPr	0.793	0.686
LCMM	0.921	0.840
Reuters-all90		
Roc-Method	0.765	0.550
KNN	0.793	0.529
ML-KNN	0.751	-
FS-KNN	0.762	-
ML-SVM	0.869	0.445
MFoM-Bin	0.884	0.556
MFoM-MC	**0.888**	0.630
LCMM-NoPr	0.751	0.551
LCMM	0.816	**0.635**

TABLE 6. Performance comparison on TMC2007.

Approach	Accuracy	Macro F
Ensemble-CC	0.530	0.551
Ensemble-BM	0.527	0.548
Ensemble-PS	0.523	0.561
RAKEL	0.529	0.557
LCMM-NoPr	0.405	0.524
LCMM	**0.533**	**0.595**

TABLE 7. Performance comparison on OHSUMED.

Approach	Accuracy	Macro F
Ensemble-CC	0.411	0.378
Ensemble-BM	0.414	0.379
Ensemble-PS	0.420	0.376
RAKEL	0.416	0.398
LCMM-NoPr	0.409	0.478
LCMM	**0.489**	**0.537**

of LCMM are better than that of Roc-Method and the series of KNN approaches, but are worse than those of ML-SVM and MFoM approaches. We further analyze these results in more detail. From the perspective of model training, the parameter learning procedure of LCMM tries to find the parameters that optimize the local performance of each document, which is coincident with the document-based evaluation measures, such as precision, recall and, F-Score. The ML-SVM and MFoM approaches learn the parameters to obtain the best performances for each label, which is consistent with the micro and macro measures. Note that MFoM, whose objective function was directly designed for optimizing micro F-Score, achieved the best performances in Table 5. On the other hand, LCMM is a probabilistic generative model, which can be used for classification based on its modeling ability. For Reuters-top10, the number of labels is not large and the data is sufficient for parameter estimation. The performances on micro F-Score and macro F-Score of LCMM are slightly worse than those of ML-SVM and MFoM methods. However, from Reuters-top10 to Reuters-all90, the number of labels increases by 80 while the number of documents increases by only 1278, and the top 10 labels still accounts for a large proportion (74.5%) of all the label occurrences. As a result, there are not sufficient data for the parameter learning for the other 80 labels. Therefore, the performances on Reuters-all90 of LCMM decrease clearly. ML-SVM and MFoM methods are essentially discriminative models whose primary targets are learning different surfaces between labels, which are relatively less sensitive to the amount of training data compared with LCMM.

With regard to the other two data sets TMC2007 and OHSUMED, LCMM performs much better than the reported methods in terms of accuracy and macro F-Score. In addition, we can observe from the tables (including the performances presented in Appendix A) that the overall performance values of LCMM-NoPr are significantly worse than those of LCMM. LCMM-NoPr tends to assign more labels to a document, which can result in the improvement on the recall or micro/macro recall scores on several data sets. However, this improvement does not occur on all the data sets because the relevant recall or micro/macro recall performance values are determined not only by the number of labels but also by the accuracy of the document model itself. All these results demonstrate the effectiveness of the prior based on the label correlation network.

As for time efficiency, we measure the time consumption for LCMM on each data set and the results are presented in Table 8. For all experiments, we used an Intel Xeon E5405 2.00GHz processor running Windows Server2003-32bits(OS). All the programs for LCMM are written in C++. From the results in Table 8, we can give a qualitative conclusion that LCMM is efficient compared to other methods [2], whose results are not presented in the table because of different experimental conditions. The strategy demonstrated in Section V is reasonable and can effectively control the time complexity of LCMM. However, from results about the average prediction time for a testing document, we find strong positive correlation between the time consumption and the number of labels in the data set.

TABLE 8. Testing time for a whole set and average time for a testing example (in seconds).

data set	testing time	average time
MSD-TCS	39.6	3.65×10^{-2}
Reuters-top10	3.2	1.26×10^{-3}
Reuters-top36	17.1	5.87×10^{-3}
Reuters-top90	75.2	2.49×10^{-2}
TMC2007	47.4	6.69×10^{-3}
OHSUMED	51.3	7.36×10^{-3}

D. DISCUSSION

From the perspective of Bayesian decision theory, both the likelihood and the prior should be calculated for classification. In order to calculate the likelihood, the document model is adopted in the LCMM approach and similar models [9], [12] are also used in other approaches. These supervised probabilistic models are the basis of the classifiers. However, multi-label categorization performance is also significantly influenced by the prior, which can be illustrated through the comparison of the experimental results of LCMM-NoPr and LCMM. Compared with other approaches, one primary advantage of LCMM is that it can provide an reasonable estimation for the prior probability of an arbitrary subset of labels, which facilitates the classification within the Bayesian decision framework. Therefore, LCMM can have further performance improvement compared with LCMM-NoPr.

According to Eq. (3), if every multi-label class only has one single label, then the multi-label classification problem reduces to a single label classification problem and Eq. (3) is equivalent to direct estimating the prior for each single label. However, for general multi-label classification problems, it is difficult to directly calculate the probability of a label subset, because the number of multi-label classes may be significantly larger than the number of observations. LCMM does not directly calculate the prior of a label subset. Instead, LCMM estimates the probability of a label subset based on the correlations between pairs of labels, which are encoded on a label correlation network. The approximate computation can be implemented through a recursive strategy according to Eq. (10), in which large complicated problems are decomposed to a set of small simple problems. On the other hand, the correlation between labels is involved in the process of estimating a prior, which is reasonable and necessary for the multi-label case. Furthermore, each multi-label class can be represented by a subgraph of the label correlation network. So LCMM can estimate an unseen multi-label class in the training data set, as long as it corresponds to a complete subgraph in the label correlation network. This can also be considered as the LCMM's smoothing effect for the unseen data.

TABLE 9. Performances of LCMM-NoPr and LCMM according to accuracy, precision, recall, F-score.

Approach	Accuracy	Precision	Recall	F-Score
Reuters-top10				
LCMM-NoPr	0.787	0.800	0.973	0.848
LCMM	0.922	0.940	0.949	0.932
Reuters-top36				
LCMM-NoPr	0.791	0.835	0.864	0.829
LCMM	0.855	0.882	0.911	0.883
Reuters-all90				
LCMM-NoPr	0.771	0.818	0.830	0.805
LCMM	0.839	0.878	0.872	0.870
TMC2007				
LCMM-NoPr	0.405	0.530	0.631	0.524
LCMM	0.533	0.671	0.668	0.626
OHSUMED				
LCMM-NoPr	0.409	0.479	0.673	0.524
LCMM	0.489	0.610	0.640	0.584

TABLE 10. Performances of LCMM-NoPr and LCMM according to micro precision, micro recall, micro F-score.

Approach	Micro P	Micro R	Micro F
Reuters-top10			
LCMM-NoPr	0.678	0.955	0.793
LCMM	0.912	0.930	0.921
Reuters-top36			
LCMM-NoPr	0.781	0.787	0.784
LCMM	0.821	0.861	0.841
Reuters-all90			
LCMM-NoPr	0.772	0.731	0.751
LCMM	0.850	0.785	0.816
TMC2007			
LCMM-NoPr	0.466	0.565	0.511
LCMM	0.649	0.618	0.633
OHSUMED			
LCMM-NoPr	0.438	0.619	0.513
LCMM	0.575	0.584	0.579

TABLE 11. Performances of LCMM-NoPr and LCMM according to macro precision, macro recall, macro F-score.

Approach	Macro P	Macro R	Macro F
Reuters-top10			
LCMM-NoPr	0.551	0.910	0.686
LCMM	0.843	0.837	0.840
Reuters-top36			
LCMM-NoPr	0.618	0.590	0.604
LCMM	0.716	0.662	0.688
Reuters-all90			
LCMM-NoPr	0.597	0.512	0.551
LCMM	0.722	0.567	0.635
TMC2007			
LCMM-NoPr	0.420	0.697	0.524
LCMM	0.641	0.556	0.595
OHSUMED			
LCMM-NoPr	0.386	0.628	0.478
LCMM	0.571	0.507	0.537

In addition, we can see from Section II that the label correlation model and the document model are relatively independent. In our experiments, we train the two models on the same training data set. However, this is not necessary in practice and the models can be trained on different data sets. For the document model, both the documents and the corresponding multi-label classes are needed, while for the prior model, the data set can only contain the multi-label classes.

VII. CONCLUSION

This paper presents a label correlation mixture model for multiple labeled document data, which is a probabilistic generative model and can be used for multi-label spoken document categorization as well as multi-label text categorization. LCMM models the generating process of both multiple labels and words of a given document in two phases, which correspond to a label correlation model and a document model. The label correlation network is defined and constructed for formulating the correlation between labels and estimating the prior of an arbitrary subset of labels. The words are generated based on labels, which are depicted by the document label, of which the parameters can be learned through the MCE criterion. The experimental results on a spoken document data set and three standard text data set illustrate LCMM's effectiveness.

For multiply labeled document data, LCMM provides a general framework for generative model and has advantages in terms of the concise depiction of data generating process and the ability of addressing the correlations between labels. In many applications, there may be a certain order or a hierarchical structure underlying the labels [37]. These structures can be regarded as a special case of the label correlation network. Therefore, LCMM is also applicable to these cases. Moreover, LCMM can also be used for other multiple labeled collections of discrete data sets.

APPENDIX A
COMPLETE PERFORMANCES

In this appendix, the complete results of LCMM-NoPr and LCMM are presented in Tables 9–11. The relevant evaluation measures are described in Section IV-B.

REFERENCES

[1] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[2] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, 2011.

[3] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2008, pp. 995–1000.

[4] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A maximal figure-of-merit (MFoM)-learning approach to robust classifier design for text categorization," *ACM Trans. Inf. Syst.*, vol. 24, no. 2, pp. 190–218, 2006.

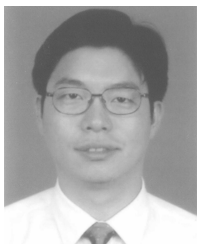
[5] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A MFoM learning approach to robust multiclass multi-label text categorization," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 42.

[6] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 135–168, 2000.

[7] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[8] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
[9] A. K. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Proc. AAAI Workshop Text Learn.*, 1999, pp. 1–7.
[10] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 50–57.
[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.
[12] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, vol. 1. 2009, pp. 248–256.
[13] H. Wang, M. Huang, and X. Zhu, "A generative probabilistic model for multi-label classification," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2008, pp. 628–637.
[14] T. Li, C. Zhang, and S. Zhu, "Empirical studies on multilabel classification," in *Proc. 18th IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, 2006, pp. 86–92.
[15] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Mach. Learn.*, vol. 88, nos. 1–2, pp. 157–208, 2012.
[16] L. E. Baum, "An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.
[17] B. H. Juang and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.
[18] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification [pattern recognition]," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.
[19] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.
[20] W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proc. IEEE*, vol. 88, no. 8, pp. 1201–1223, Aug. 2000.
[21] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, Nov. 1998.
[22] W. Chou, B. H. Juang, and C. H. Lee, "Segmental GPD training of HMM based speech recognizer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 1. Mar. 1992, pp. 473–476.
[23] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 107–113, Jan. 1991.
[24] R. Schluter and W. Macherey, "Comparison of discriminative training criteria," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 1. May 1988, pp. 493–496.
[25] D. Kanevsky, "A generalization of the Baum algorithm to functions on non-linear manifolds," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 1. May 1995, pp. 473–476.
[26] Y. Normandin, R. Cardin, and R. de Mori, "High-performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 299–311, Apr. 1994.
[27] D. D. Lewis. *Reuters-21578 Text Categorization Test Collection, Distribution 1.0*. [Online]. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578, accessed 1997.
[28] A. N. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *Proc. IEEE Aerosp. Conf.*, Mar. 2005, pp. 3853–3862.
[29] [Online]. Available: http://disi.unitn.it/moschitti/corpora.htm
[30] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.
[31] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell, MA, USA: Kluwer, 2002.
[32] F. Debole and F. Sebastiani, "An analysis of the relative hardness of Reuters-21578 subsets," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 56, no. 6, pp. 584–596, 2005.
[33] D. Vilar, M. J. Castro, and E. Sanchis, "Multi-label text classification using multinomial models," in *Advances in Natural Language Processing*. Berlin, Germany: Springer-Verlag, 2004, pp. 220–230.
[34] J.-Y. Jiang, S.-C. Tsai, and S.-J. Lee, "FSKNN: Multi-label text categorization based on fuzzy similarity and k nearest neighbors," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2813–2821, 2012.
[35] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Machine Learning*. Berlin, Germany: Springer-Verlag, 2007, pp. 406–417.
[36] K. M. A. Chai, H. L. Chieu, and H. T. Ng, "Bayesian online classifiers for text classification and filtering," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2002, pp. 97–104.
[37] C. Zeng, T. Li, L. Shwartz, and G. Y. Graharnik, "Hierarchical multi-label classification over ticket data using contextual loss," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.

ZHIYANG HE received the B.S. degree in communication engineering from People's Liberation Army Information Engineering University, Zhengzhou, China, in 2003, and the M.S. degree in signal and information processing from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006. He was with iFLYTEK Corporation, Hefei, China, as a Researcher, from 2006 to 2012. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China. His research interests include spoken language understanding, natural language understanding, machine learning, and pattern recognition.



JI WU is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University, Beijing, China, and a Visiting Scholar with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. He received the B.S. and Ph.D. degrees in electronic engineering from Tsinghua University, in 1996 and 2001, respectively, where he is the Head of the Multimedia Signal and Intelligence Information Processing Laboratory. Since 2006, he has been the Director of Tsinghua-iFlyTek Joint Laboratory for Speech Technologies, Beijing. He is also the Leader of Technical Work Group with the Speech Industry Alliance of China. His research interests include speech recognition, natural language processing, pattern recognition, machine learning, and data mining.

TAO LI received the Ph.D. degree in computer science from the Department of Computer Science, University of Rochester, Rochester, NY, USA, in 2004. He is currently a Full Professor with the School of Computing and Information Sciences, Florida International University, Miami, FL, USA. His research interests are data mining, computing system management, information retrieval, and machine learning. He was a recipient of the NSF CAREER Award and the multiple IBM Faculty Research Awards.