Pattern Recognition 47 (2014) 3405–3413

Contents lists available at ScienceDirect

Pattern Recognition

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

# Enhancing multi-label classification by modeling dependencies among labels

Shangfei Wang [a,*], Jun Wang [a], Zhaoyu Wang [a], Qiang Ji [b]

[a] Key Lab of Computing and Communication Software of Anhui Province, School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, PR China
[b] Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

*Article history:*
Received 2 July 2013
Received in revised form
4 April 2014
Accepted 8 April 2014
Available online 16 April 2014

*Keywords:*
Multi-label classification
Bayesian network
Structure learning
Incomplete label assignments
Maximum likelihood estimation

ABSTRACT

In this paper, we propose a novel framework for multi-label classification, which directly models the dependencies among labels using a Bayesian network. Each node of the Bayesian network represents a label, and the links and conditional probabilities capture the probabilistic dependencies among multiple labels. We employ our Bayesian network structure learning method, which guarantees to find the global optimum structure, independent of the initial structure. After structure learning, maximum likelihood estimation is used to learn the conditional probabilities among nodes. Any current multi-label classifier can be employed to obtain the measurements of labels. Then, using the learned Bayesian network, the true labels are inferred by combining the relationship among labels with the labels' estimates obtained from a current multi-labeling method. We further extend the proposed multi-label classification method to deal with incomplete label assignments. Structural Expectation-Maximization algorithm is adopted for both structure and parameter learning. Experimental results on two benchmark multi-label databases show that our approach can effectively capture the co-occurrent and the mutual exclusive relation among labels. The relation modeled by our approach is more flexible than the pairwise or fixed subset labels captured by current multi-label learning methods. Thus, our approach improves the performance over current multi-label classifiers. Furthermore, our approach demonstrates its robustness to incomplete multi-label classification.



## 1. Introduction

Multi-label classification is a classification problem where one sample can be assigned with more than one target labels simultaneously. There are many multi-label learning applications. For example, a piece of music may be characterized by both dreamy and cheerful [1]. An image may include grass, cow and sky [2]. Hence, a data sample (image or music) may simultaneously contain multiple different labels that characterize different properties of the data.

Usually the labels are dependent on each other. Take music emotional tagging for example, some emotions may appear together frequently, while others may not. A piece of music may induce the feelings of relaxing, comfortable and happy, but it rarely induces disgust at the same time. Such dependencies among labels are one of the key issues in multi-label learning. Current research can be divided into three groups: ignoring dependencies, exploring dependencies directly only from labels, and exploring label dependencies indirectly with the help of features or hypotheses. The first group takes no account of the relation among labels, it therefore suffers from unstable performance. The second group considers pairwise relation, or the fixed label combinations present in training data directly from labels without considering features or hypotheses. However, the dependencies among multiple labels are more complex and flexible, beyond pairwise and fixed label combinations. In fact, there are two kinds of relationship: co-existence and mutual exclusion. For example, in music emotional tagging, a piece of sad music may elicit both sadness and anger but rarely happiness, which reflects the co-existent relation between sadness and anger, and the mutual exclusive relation between sadness and happiness. Thus, the second group cannot fully explore the feasible dependencies among labels. The third group can model more feasible label relation with the aid of features and hypotheses. However, its computation cost is much higher than the first two groups.

Furthermore, since annotating labels is time confusing and require expertise, labels may be missing for some applications. For example, because of difficulty with annotating certain labels,

* Corresponding author. Tel.: +86 551 63602824.
*E-mail addresses:* sfwang@ustc.edu.cn (S. Wang), junwong@mail.ustc.edu.cn (J. Wang), wazhy@mail.ustc.edu.cn (Z. Wang), qji@ecse.rpi.edu (Q. Ji).

http://dx.doi.org/10.1016/j.patcog.2014.04.009
0031-3203/

annotators may only provide the main emotion of a piece of music, and users may tag an image just with several main objects. Therefore, learning from incomplete labels is another key issue for multi-label classification. However, current multi-label classification research rarely addresses learning from incomplete labels.

In this paper, we propose a Bayesian Network (BN) to systematically capture the dependencies among different labels directly. The nodes of the BN represent the labels. The links and their parameters capture the probabilistic relation among labels. Our structure learning algorithm [3] is employed to learn the BN structure. By exploiting the decomposable property of the Bayesian Information Criterion (BIC) score function, the algorithm significantly reduces the search space of possible structures and guarantees the global optimality. After structure learning, the conditional probabilities are directly learned on the training data. Then, we can infer the true labels by instantiating the measurement nodes with the labels' estimates obtained from a traditional multi-labeling method. The experimental results on two multi-label datasets show that both the co-occurrent and the mutual exclusive relation among labels can be effectively captured by our structure learning algorithm. The relation modeled by our approach is more flexible than pairwise or fixed subset labels captured by current multi-label learning methods, it improves the performance of current multi-label classifiers which model label dependence directly. Furthermore, we extend our approach to deal with incomplete labels by using structural Expectation-Maximization (EM) algorithm. The experimental results on the same two multi-label datasets show the advantage of our method.

## 2. Related work

Multi-label classification methods can be categorized into two groups: problem transformation methods and algorithm adaptation methods. The former includes Binary Relevance (BR) [4], Label Powerset (LP) [4], and Random k labelsets (RAkEL) [5]. They transform the multi-label classification task into one or more single-label classification tasks, and then any traditional classification algorithms can be used. The latter consists of Binary Relevance k Nearest Neighbors (BRkNN) [6], Multi-Label k Nearest Neighbors (MLkNN) [7], AdaBoost.MH [8], etc. They extend specific learning algorithms to handle multi-label data directly. A comprehensive overview of current research in multi-label classification can be found in [9,10].

Due to the large number of possible label sets, multi-label classification is rather challenging. Successfully exploiting the dependencies inherent in multiple labels is the key to facilitate the learning process. Considering dependencies among labels, most present multi-label learning strategies can be categorized into three groups: methods ignoring label correlation, methods considering label correlation directly, and methods considering label correlation indirectly. The first group (i.e., BR [4]) decomposes multi-label problem into multiple independent binary classification problems (one per category). By ignoring the correlation among labels, the generalization ability of such method may be weak. The second group addresses the pairwise relation between labels (such as Calibrated Label Ranking (CLR)), or the fixed label combinations existing in training data (such as LP), or a random subset of the combinations (such as RAkEL)). However, the relation among labels may be beyond pairwise, and cannot be expressed by a fixed subset of labels existing in training data. Besides, the number of the pairwise subsets increases exponentially when the number of the labels is quite large. Meanwhile, there may not be sufficient training data when there are few instances for the combined labels. Thus, the second group may not capture the label relation effectively. The third group considers label dependencies with the help of features or hypothesis. Godbole and Sarawagi [11] stacked the outputs of BR along with the full original feature space into a separate meta classifier, creating a two-stage classification process. Read et al. [12] proposed the classifier chain model to link $n$ classifier into a chain. The feature space of each classifier in the chain is extended with the label associations of all previous classifiers. Ghamrawi and McCallum [13] adopted conditional random field to capture the impact of an individual feature on the co-occurrence probability of a pair of labels. Sun et al. [14] proposed to construct a hyperedge for each label, and include all instances annotated with a common label into one hyperedge, thus capturing their joint similarity. Zhangs [15] proposed Bayesian Network to model the dependencies among label errors, and then a binary classifier was constructed for each label combining the features and the parental labels, which were regarded as additional features. Huang et al. [16] modeled the label relation by a hypothesis reuse process. When the classifier of a certain label is learned, all trained hypotheses generated for other labels are taken into account via weighted combinations. These methods can model the flexible dependencies among labels to some extent, but their computation costs are usually much higher compared with the second group.

Among the above, Zhangs' work is the most similar one to ours. They proposed to use a Bayesian Network (BN) structure to encode the conditional dependencies of labels as well as the feature set: $P(\lambda_1, \lambda_2, \ldots, \lambda_n|x)$, where $x$ is the features and $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the multiple target labels, $n$ is the number of labels. Since they thought directly modeling $P(\lambda_1, \lambda_2, \ldots, \lambda_n|x)$ by Bayesian approach was intractable, they adopted an approximate method to model the dependencies among label errors, which was independent of features $x$. Based on the learned BN structure of errors, a binary classifier was constructed for each label $\lambda_i$ combining the features $x$ and the parental labels $pa(\lambda_i)$, which were regarded as additional features.

Unlike Zhangs' method, we propose a Bayesian Network to systematically capture the dependencies among different labels, $P(\lambda_1, \ldots, \lambda_n)$, directly. The nodes of the BN represent the labels. The links and their parameters capture the probabilistic relation among labels. Our BN structure learning algorithm [3] is adopted. After structure learning, maximum likelihood estimation is used to learn the conditional probabilities. Then, we can infer the true labels by instantiating the measurement nodes with the labels' estimates obtained from a traditional multi-labeling method.

Compared to Zhangs' method, we first directly capture the dependencies among labels. Then we obtain label measurements, $M\lambda_i$, using any multi-label classifier. After that, we infer an instance's multiple labels simultaneously using Most Probable Explanation (MPE) inference: $P(\lambda_1, \ldots, \lambda_n|M\lambda_1, \ldots, M\lambda_n) = P(M\lambda_1, \ldots, M\lambda_n|\lambda_1, \ldots, \lambda_n)P(\lambda_1, \ldots, \lambda_n)/P(M\lambda_1, \ldots, M\lambda_n)$. Thus, our approach can explicitly model the co-existent and the mutual exclusive relation among labels, instead of the errors of the labels. Besides, our approach can infer the multiple labels of an instance simultaneously, not recognize each label separately. Our approach can be easily combined with any multi-label classifier to enhance its performance.

Current multi-label classification methods require complete label assignments. However, multi-label classification with incomplete label assignments is frequently encountered in realistic scenario, especially when the number of labels is very large. Till now, little research [17] has addressed the challenge of multi-label classification with incomplete labels [18].

In this paper, we propose a BN to systematically capture the dependencies among labels directly. Furthermore, we extend our approach to address multi-label classification with incomplete labels using a structural EM algorithm.

Compared with related work, our contributions are as follows:

1. We are the first to directly model the inherent probabilistic dependencies among labels using Bayesian network. The relation modeled by our approach is more feasible than those of current direct approaches. Since features and hypotheses are not considered during label dependency modeling, the computation cost of our approach is less than those of current indirect approaches.
2. We are among the first to address multi-label learning from incomplete labels based on the co-occurrent and the mutual exclusive relationship among labels.

## 3. Methods

The framework of our approach is shown in Fig. 1, consisting of two modules: measurement extraction and labels' relation modeling by BN. The training phase of our approach includes training the traditional multi-label classifier for measurement extraction and training the BN to capture the semantic relationship among labels. For measurement extraction, current multi-label algorithms are used. Given the measurements, we infer the final labels of samples through the most probable explanation (MPE) inference with the BN model. The details are provided as follows.

### 3.1. Measurement extraction

Let $TD=\{x_l,(\lambda_{1l},\ldots,\lambda_{nl})\}_{l=1}^{m}$ are the training data, where $x_l\in R^d$ is the feature, $(\lambda_{1l},\ldots,\lambda_{nl})$ is the multiple target labels, $n$ is the number of labels, and $m$ is the number of training samples. The measurements $M\lambda$ are the preliminary estimations of the class labels using an existing multi-label classification algorithm $h_i$ based on training data $TD$. For a new instance $x_l$, the algorithm $h_i$ outputs a binary vector $M\lambda_{il}=h_i(x_l)$, which is regarded as measurements to be further refined by the BN.

### 3.2. Label relationship modeling by Bayesian network

In order to model the semantic relationship among categories, a BN model is utilized in this work. Strictly speaking, the relationship among labels is mutual. Their relationship can hence be better captured by undirected links. In this sense, undirected graphical model such as Markov Network can better capture relationship among labels. However, structure learning of a Markov Network is very challenging and remains an open problem. We therefore use directed graphical model, i.e., BN to approximately capture the relationship among labels. As a probabilistic graphical model, BN can effectively capture the dependencies among variables in data. In our work, each node of the BN is a variable, representing one of the multiple labels, and the links and their conditional probabilities capture the probabilistic dependencies among labels.

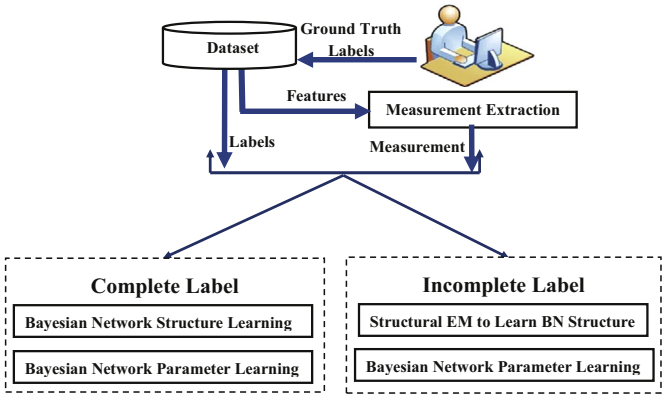


**Fig. 1.** The framework of our proposed multi-label classification approach.

#### 3.2.1. BN structure and parameters with complete label

A BN is a directed acyclic graph (DAG) $G=(\Lambda,E)$, where $\Lambda=\{\lambda_i\}_{i=1}^{n}$ represents a collection of $n$ nodes and $E$ denotes a collection of arcs. Given the complete dataset of multiple target labels $D=\{\lambda_{il}\}$ of the training set, where $i=1,2,\ldots,n$ is an index to the number of nodes, and $l=1,2,\ldots,m$ is an index to the number of samples. Score-based BN structure learning aims to find a DAG $G$ that maximizes the score function $S(G)$. In this work, we employ the Bayesian Information Criterion (BIC) [19] score function as defined in

$$S(G)=\max_{G} L_D(G)-\frac{Dim_G}{2}\log\ m \tag{1}$$

where the first term is the log-likelihood function of structure $G$ with respect to the training data $D$, representing how well $G$ fits to the data and the second term is a penalty term that limits the complexity of $G$, where $Dim_G$ is the number of independent parameters. The score function hence represents a trade-off between the model fitness to the data and the model complexity. By exploiting the decomposable property of the BIC score as well as the conditional independencies embedded in the BN, the score function $S$ for $G$ can be written as the sum of the scores for each node $\lambda_i$, i.e.,

$$S(G)=\sum_{i=1}^{n}S_i(\lambda_i)=\sum_{i=1}^{n}\left(\max_{\theta_i}L_{pa(\lambda_i)}(\theta_i)-Dim_{pa(\lambda_i)}\cdot\frac{\log\ m}{2}\right) \tag{2}$$

where $pa(\lambda_i)$ represents the parents of node $\lambda_i$. Given Eq. (2), the structure learning can be formulated as identifying the optimal parent set $pa(\lambda_i)$ for each node $\lambda_i$ and combining the optimal structures for each node subject to the constraint that the combined structure $G$ is a DAG. Since the number of possible parent sets for each node is exponential with respect to the number of nodes $n$, to effectively explore the space, we introduced a branch-and-bound ($B\&B$) with cache approach [3] to efficiently learn the optimal BN structure. In this algorithm, the cache is used to store the possible parent sets for each node along with their scores. By exploiting the properties with the BIC score, we proved in [3] that for an optimal BN, each node has at most $log_2 n$ parents. We further provided additional theorems [3] that can significantly reduce the number of parent sets needed to store in the cache for each node. Finally, structural constraints such as the degree of each node and the presence or the absence of a link are allowed to further reduce the parent space for each node. Given the reduced cache space, we then introduced a constrained $B\&B$ algorithm to find the best BN structure as summarized in Algorithm 1, where $C:(\lambda_i,pa(\lambda_i))\rightarrow R$ is the cache that stores, for each node, the possible parent configurations complying with the structural constraints and the theorems and their scores, $G$ is the graph created by combining the best parent set for each node without checking for acyclicity, and $s$ is the score of $G$; $H$ is an initially empty matrix containing, for each possible arc between nodes, a mark stating that the arc must be present, or is prohibited, or is free (may be present or not). $H$ is used to record the branch and bound operations; $Q$ is a priority queue of triples $(G;H;s)$, ordered by $s$, and $(G_{best};s_{best})$ is the best DAG and score found so far ($s_{best}$ is initialized to be $-\infty$).

**Algorithm 1.** BN structure learning [3].

1: **while** $Q$ is not empty **do**
2: Remove the top $(G_{cur};H_{cur};s_{cur})$ of $Q$. If $s_{cur}\le s_{best}$ (worse than an already known solution), then discard the current element and start the loop again.
3: If $G_{cur}$ is a DAG and satisfies all structural constraints, update $(G_{best};S_{best})$ with $(G_{cur};S_{cur})$ and start the loop again.

4: For $G_{cur}$ with cycles, take a directed cycle $v=(\lambda_{a_1}\rightarrow\lambda_{a_2}\rightarrow\cdots\rightarrow\lambda_{a_{q+1}})$ with $a_1=a_{q+1}$. (start of the branch and bound operation)
5: **for** $y=1,\ldots,q$ **do**
6: Mark in $H_{cur}$ that the link $\lambda_{a_y}\rightarrow\lambda_{a_{y+1}}$ is prohibited
7: Recompute $(G;S)$ from $(G_{cur};S_{cur})$ such that the parents of $\lambda_{a_{y+1}}$ in $G$ comply with this restriction and the subgraph of $G$ contains arcs marked as required in $H_{cur}$ and must comply with the structural constraints. Use the values in the cache $C(\lambda_{a_{y+1}},pa(\lambda_{a_{y+1}}))$ to avoid recomputing scores.
8: Add the triple $(G,;H_{cur};S)$ into $Q$.
9: Mark on $H_{cur}$ that the arc $\lambda_{a_y}\rightarrow\lambda_{a_{y+1}}$ must be present, and that the sibling arc $\lambda_{a_{y+1}}\rightarrow\lambda_{a_y}$ is prohibited, and continue.
10: **end for**
11: **end while**

At each step, a graph is picked up from a priority queue, and it is verified if it is a DAG. If so, it is a feasible structure for the network and we compare its score against the best score so far. Otherwise, there must be a directed cycle in the graph, which is then broken into subcases by forcing some arcs to be absent/present systematically through the branch and bound operation. Each subcase is put in the queue to be processed. The procedure stops when the queue is empty. More details of this algorithm can be found in [3] and its software can be downloaded from this link[1]

Given the learned structure $G$, the conditional probabilities $\theta$ among nodes can be re-learned by Maximum Likelihood Estimation (MLE) method using the same training data, which can be described as a constrained optimization problem as shown in

$$\theta^*=\max_{\theta} L(\theta)$$
$$\text{s.t.}\quad g_{ij}(\theta)=\sum_{k=1}^{s_i}\theta_{ijk}-1=0 \tag{3}$$

where

$$L(\theta)=\log\prod_{i=1}^{n}\prod_{j=1}^{|pa(\lambda_i)|}\prod_{k=1}^{|\lambda_i|}\theta_{ijk}^{n_{ijk}}, \tag{4}$$

and $g_{ij}$ imposes the constraint that the parameters of each node sums to 1 over all the states of that node, $\theta_{ijk}$ is the conditional probability of node $i$ is equal to $k$, given the $j^{th}$ parent configuration, and $n_{ijk}$ is the number of elements in $D$ with node $i$ having a value of $k$ and its parent having $j^{th}$ configuration. Solving the above equations, we can get $\theta_{ijk}=n_{ijk}/\sum_k n_{ijk}$.

### 3.2.2. BN structure and parameter learning with incomplete labels

When labels are incomplete, the above BN learning method cannot be used directly. Hence, the structural EM algorithm [20] is adopted to address incomplete labels. In structural EM, we iterate over a pair of steps: the E-step and the M-step. In the E-step, we use our current model to generate a completed data set, based on which we compute the expected sufficient statistics. In the M-step, we use these expected sufficient statistics to improve our model, including both parameters and structure. The structure and parameter learning procedures in the M-step are the same as Algorithm 1. The detailed learning algorithm is summarized in Algorithm 2.

**Algorithm 2.** Structural EM algorithm.

1: Initialize the structure $G^0$ and the parameter set $\theta^0=\{P(\lambda_i|pa(\lambda_i)),P(\lambda_i)\}$ for the BN model based on the complete portion of the training data. This structure is initialized based on Algorithm 1 using only the completely labeled portion of the data.
2: **repeat**
3: E-step
Using the new BN structure and parameters to infer the missing labels of the incomplete samples and update the training set $D$ by completing the missing labels.
4: M-step
Using the updated training set $D$ and Algorithm 1 to learn a new BN structure and parameters.
5: **until** convergence

### 3.2.3. BN inference

After BN structure and parameter learning, the relation among multi-labels is captured by the links and conditional probabilities of the learned BN. Then, the measurement nodes are linked to the corresponding labels, as shown in Figs. 2 and 3. During recognition, the posterior probability of category labels can be estimated via BN inference by combining the likelihood from measurement with the BN model. Let $\lambda_i$ and $M\lambda_i$, $i\in\{1,\ldots n\}$, denote the label variable and the corresponding measurement obtained from a multi-label learning method respectively. Instead of performing MAP inference for

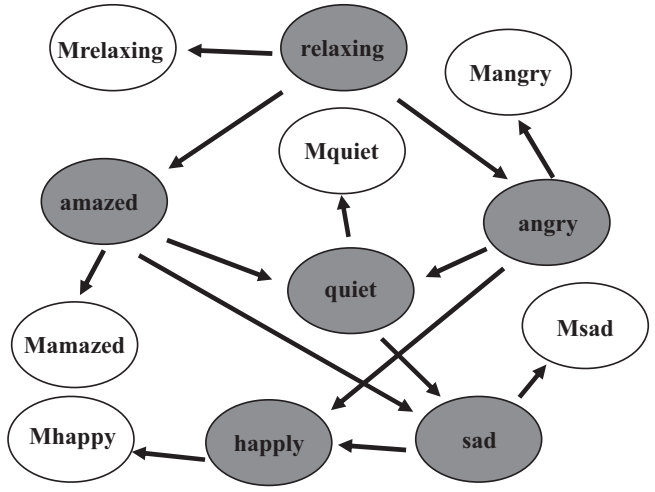


**Fig. 2.** The learned BN structure from emotions dataset. The shaded nodes are the hidden nodes we want to infer and the unshaded nodes are the corresponding measurement nodes.

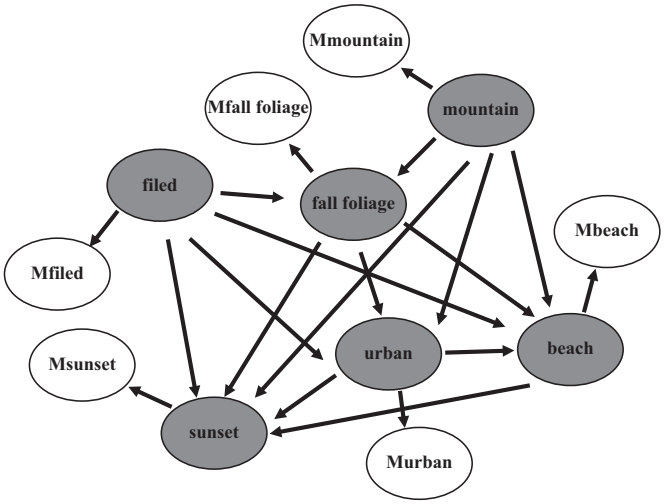


**Fig. 3.** The learned BN structure from scene dataset. The shaded nodes are the hidden nodes we want to infer and the unshaded nodes are the corresponding measurement nodes.

[1] The software is available at http://www.ecse.rpi.edu/cvrl/structlearning.html.

**Table 1**
Information of the datasets.

| Dataset | Examples | Features | Labels | DLS | LC | LD |
|---|---|---|---|---|---|---|
| emotions | 593 | 72 | 6 | 27 | 1.868 | 0.311 |
| scene | 2712 | 294 | 6 | 15 | 1.074 | 0.179 |

'DLS' represents the number of distinct label subsets; 'LC' (label cardinality) denotes the average number of labels per example; 'LD' (label cardinality) is the average number of labels of the examples divided by the number of labels [4].

each node individually, we perform the Most Probable Explanation (MPE) inference [21]. MPE computes the joint posterior probabilities of all nodes $\lambda_i$, given their measurements $M\lambda_i$. Through the MPE inference, we can identify a joint configuration of all nodes that best explains the given measurements, i.e.,

$$\begin{aligned}\Lambda^{\star} &= \arg\max_{k_1,\ldots,k_n} P(\lambda_1^{k_1},\ldots,\lambda_n^{k_n})|M\lambda_1,\ldots,M\lambda_n)\\ &= \arg\max_{k_1,\ldots,k_n} \frac{P(\lambda_1^{k_1},\ldots,\lambda_n^{k_n},M\lambda_1,\ldots,M\lambda_n)}{P(M\lambda_1,\ldots,M\lambda_n)}\\ &\propto \arg\max_{k_1,\ldots,k_n} \prod_{i=1}^{n} P(M\lambda_i|\lambda_i^{k_i}) \prod_{i=1}^{n} P(\lambda_i^{k_i}|pa(\lambda_i)) \end{aligned} \quad (5)$$

where the condition probabilities, i.e., $P(\lambda_i^{k_i}|pa(\lambda_i))$ and $P(M\lambda_i|\lambda_i^{k_i})$, are learned from training set using MLE algorithm in Eq. (3). An existing exact BN inference method such as the variable elimination or the junction tree method [22] may be employed to perform the MPE inference.

## 4. Experiments and analyses

### 4.1. Experimental conditions

#### 4.1.1. Datasets

In this paper, two datasets showed in Table 1 are used to test our proposed method: emotions dataset [1] and scene dataset [2]. The emotions dataset is concerned with the classification of music clips. It contains 593 songs categorized into one or more out of six emotional categories. For each music clip, 8 rhythmic features and 64 timbre features are extracted. The scene dataset is created to address the problem of emerging demand for semantic image classification. Each instance in scene dataset is an image which is assigned with multiple classes. The dataset contains 2407 images associated with up to 6 labels. The first and second moments are extracted from image blocks. Totally, there are 294 features for each image.

#### 4.1.2. Label measurements

Four commonly used multi-label classification methods are adopted to obtain the measurements of the labels for complete multi-label classification experiments. They are Binary Relevance (BR) [4], Calibrated Label Ranking (CLR) [23], LP [4] and Random k label sets (RAkEL) [5].

BR considers each label independently. First, it changes original data set to $n$ data sets, each data set $D_i$ for one label $\lambda_i$. Then, any traditional classification algorithm can be used as the classifier $h_i$ using $D_i$. For a new instance, each classifier $h_i$ outputs a binary label $M\lambda_i = h_i(x)$, where $x$ is the feature vector of the sample. Then, the combination of the labels predicted by $n$ classifiers ($\bigcup_{i=1}^{n} M\lambda_i$) is adopted as the final output. BR assumes the labels are independent, ignoring the correlation among those labels.

CLR is an extension of Ranking by Pairwise Comparison (RPC), which considers the pairwise label relation. RPC transforms the multi-label dataset into $(n\times(n-1))/2$ binary label datasets, one for each pair of labels. A binary classifier is then trained on each of these datasets. Given a new instance, its ranking of labels is obtained by counting the votes of all these binary classifiers for each label. Ranking does not have a natural "zero-point", and therefore cannot distinguish between the relevant and non-relevant labels of a new instance. CLR is proposed by introducing an additional label (i.e., calibration label) to the original label set. The calibration label can be thought as a split point between relevant and irrelevant labels.

LP considers each distinct label combination existing in the training set as a different class of a single-label classification task. Any traditional classification algorithm can be used to obtain the single-label classifier. A possible drawback of LP method is that some labels are associated with very few training samples which makes the learning difficult. To deal with the problem, RAkEL is proposed by breaking the initial set of labels into $l$ random subsets, each subset has $k$ labels, and then employing LP to train $l$ corresponding classifiers. For a new instance, its labels are the combination of all the LP classifiers, which calculates the mean of these predictions for each label and outputs a final positive decision. RAkEL considers the randomly selected combinations of labels, but it does not capture the probabilistic relation among labels, and it cannot represent their coexistent and mutual exclusive relationship.

The SVM with a linear kernel is adopted as the classifier for the above four methods, and cross validation is adopted to tune the parameters of the SVM. The input of these methods is the feature vector extracted from the instances. The outputs are binary vectors, indicating whether an instance has a certain label or not. The outputs are regarded as the input to the BN to infer the final labels. 10-fold cross-validation is adopted.

For experiments of incomplete labels, BR is adopted. Similarly, the SVM with a linear kernel and 10-fold cross validation are adopted.

#### 4.1.3. Evaluation metric

The evaluation metric of multi-label classification is different from that of single label classification, since for each instance there are multiple labels which may be classified partly correctly or partly incorrectly. Thus, there are two kinds of commonly used metrics, example-based and label-based measures [24], evaluating the multi-label classification performance from the view of instance and label respectively. We adopt both measures in this work. Let $\Lambda_i$ denotes the true labels for instance $i$, which is a binary vector, and $Z_i$ is the predicted labels for instances $i$, $m$ represents the number of the instances and $n$ is the number of labels. The example-based measures: accuracy, recall, F1-measure and subset accuracy are defined in Eqs. (6), (7), (8) and (9) [24], and the label-based measures: recall and F1-measure are defined in Eqs. (10) and (11) [24] respectively:

$$Accuracy = \frac{1}{m}\sum_{i=1}^{n}\left|\frac{\Lambda_i\bigcap Z_i}{\Lambda_i\bigcup Z_i}\right| \quad (6)$$

$$Recall = \frac{1}{m}\sum_{i=1}^{n}\frac{|\Lambda_i\bigcap Z_i|}{|\Lambda_i|} \quad (7)$$

$$F_1 = \frac{1}{m}\sum_{i=1}^{n}\frac{2|\Lambda_i\bigcap Z_i|}{|\Lambda_i|+|Z_i|} \quad (8)$$

$$Subset\ Accuracy = \frac{1}{m}\sum_{i=1}^{n} I(\Lambda_i = Z_i) \quad (9)$$

$$Recall, R_{micro} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\Lambda_i^j Z_i^j}{\sum_{j=1}^{n}\sum_{i=1}^{m}\Lambda_i^j} \quad (10)$$

$$F_{1-micro} = \frac{2\sum_{j=1}^{n}\sum_{i=1}^{m}\Lambda_i^j Z_i^j}{\sum_{j=1}^{n}\sum_{i=1}^{m}\Lambda_i^j + \sum_{j=1}^{n}\sum_{i=1}^{m}Z_i^j} \quad (11)$$

**Table 2**
Dependencies among emotional labels for emotions dataset.

| $P(\lambda_j\|\lambda_i)$ | $\lambda_j$ | | | | | |
|---|---|---|---|---|---|---|
| | amazed | happy | relaxing | quiet | sad | angry |
| $\lambda_i$ | | | | | | |
| amazed | 1 | 0.3237 | 0.0751 | 0 | 0.0578 | 0.5318 |
| happy | 0.3373 | 1 | 0.5482 | 0.0422 | 0.006 | 0.0723 |
| relaxing | 0.0492 | 0.3447 | 1 | 0.3939 | 0.3598 | 0.0265 |
| quiet | 0 | 0.0473 | 0.7027 | 1 | 0.7095 | 0.0135 |
| sad | 0.0595 | 0.0060 | 0.5655 | 0.6250 | 1 | 0.1190 |
| angry | 0.4868 | 0.0635 | 0.0370 | 0.0106 | 0.1058 | 1 |

**Table 3**
Dependencies among labels for scene dataset.

| $P(\lambda_j\|\lambda_i)$ | $\lambda_j$ | | | | | |
|---|---|---|---|---|---|---|
| | beach | sunset | fall foliage | field | mountain | urban |
| $\lambda_i$ | | | | | | |
| beach | 1.0000 | 0.0000 | 0.0000 | 0.0023 | 0.0890 | 0.0445 |
| sunset | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| fall foliage | 0.0000 | 0.0000 | 1.0000 | 0.0605 | 0.0353 | 0.0000 |
| field | 0.0023 | 0.0000 | 0.0554 | 1.0000 | 0.1755 | 0.0139 |
| mountain | 0.0713 | 0.0000 | 0.0263 | 0.1426 | 1.0000 | 0.0019 |
| urban | 0.0441 | 0.0000 | 0.0000 | 0.0139 | 0.0023 | 1.0000 |

### 4.2. Experimental results and analyses for complete labels

#### 4.2.1. Results and analyses of multiple label's dependencies modeling by BN

We quantify the co-occurrence among different labels using a conditional probability of $P(\lambda_j|\lambda_i)$, which measures the probability of label $\lambda_j$, given label $\lambda_i$.

Tables 2 and 3 show the conditional probabilities between different labels for the emotions dataset and the scene dataset respectively. From Table 2, we can find that each music piece can display multiple emotions. For instance, quiet is often accompanied by relaxing and sad with high probability. There exist two kinds of relationship among emotions: co-occurrence and mutual exclusion. For example, $P(happy|angry)$ and $P(happy|sad)$ are respectively 0.0635 and 0.006, which show happiness rarely coexists with sad and anger. Quiet is always coexistent with relaxing as indicated by a high $P(relaxing|quiet)$ of 0.7027. From Table 3, we can find that although one image may have multiple labels, their co-existent probability is rare low. For example, $P(fallfoliage|mountain)$ is 0.0263, which means that there is few samples with these two labels.

To systematically capture relationship among labels, we learned a BN on each dataset. Figs. 2 and 3 show the learned BN from the emotions dataset and the scene dataset respectively.

The links in the structure represent the dependencies among labels. For example, in Fig. 2, the links from relaxing to angry and amazed demonstrate that there are strong dependencies between the two pairs. From Table 2, we can see that the probabilities of $P(angry|relaxing)$ and $P(amazed|relaxing)$ are 0.0265 and 0.0492 respectively, indicating mutual exclusive relation. Meanwhile, the link from quiet to sad shows the co-occurrent relationship because the probability of $P(sad|quiet)$ is 0.7095 in Table 2.

Comparing two learned BNs with the two dependency tables, we find that the label pairs whose conditional probabilities are top ranked or bottom ranked are all linked in the BNs for emotions and scene datasets. It demonstrates the effectiveness of the BN structure learning method which can effectively capture the mutual exclusive and the co-existent relationship among multiple labels. For example, in emotions dataset, the probabilities reflected by the links showing mutual exclusive relationship are $P(quiet|amazed)$ (0), $P(happy|sad)$ (0.006), $P(quiet|angry)$ (0.0106), $P(amazed|relaxing)$ (0.0492), $P(sad|amazed)$ (0.0578) and $P(happy|angry)$ (0.0635). The probability reflected by the link showing co-existent relationship is $P(sad|quiet)$ (0.7095) in emotions dataset. These kinds of relation among labels captured by our BN are beyond the scope of those captured by commonly used multi-label learning methods.

**Table 4**
Comparative experimental results of our model with commonly used multi-label classifiers for complete labels in emotions dataset.

| Method | Example-based | | | | Label-based | |
|---|---|---|---|---|---|---|
| | acc. | rec. | F1 | subAcc. | micRec. | micF1. |
| LP | 0.555 | 0.664 | 0.634 | 0.329 | 0.665 | 0.656 |
| LP+BN | 0.554 | **0.673** | 0.634 | 0.320 | **0.675** | 0.656 |
| CLR | 0.516 | 0.608 | 0.595 | 0.275 | 0.608 | 0.652 |
| CLR+BN | **0.547** | **0.677** | **0.621** | **0.325** | **0.680** | **0.654** |
| BR | 0.514 | 0.598 | 0.593 | 0.268 | 0.596 | 0.650 |
| BR+BN | **0.552** | **0.684** | **0.629** | **0.322** | **0.687** | **0.660** |
| RAkEL | 0.572 | 0.670 | 0.654 | 0.324 | 0.665 | 0.684 |
| RAkEL+BN | **0.575** | **0.703** | **0.658** | **0.327** | **0.701** | 0.681 |

"acc." refers to "accuracy", "rec." refers to "recall", "subAcc." refers to "subsetAccuracy", "micRec." refers to "micro recall", "micF1." refers to "micro F1".

**Table 5**
Comparative experimental results of our model with commonly used multi-label classifiers for complete labels in scene database.

| Method | Example-based | | | | Label-based | |
|---|---|---|---|---|---|---|
| | acc. | rec. | F1 | subAcc. | micRec. | micF1. |
| LP | 0.736 | 0.748 | 0.748 | 0.698 | 0.733 | 0.744 |
| LP+BN | **0.739** | 0.747 | **0.750** | **0.704** | 0.732 | **0.747** |
| CLR | 0.609 | 0.675 | 0.636 | 0.532 | 0.664 | 0.697 |
| CLR+BN | **0.673** | **0.688** | **0.686** | **0.634** | **0.677** | 0.686 |
| BR | 0.606 | 0.655 | 0.629 | 0.541 | 0.644 | 0.691 |
| BR+BN | **0.668** | **0.679** | **0.680** | **0.632** | **0.668** | 0.680 |
| RAkEL | 0.681 | 0.715 | 0.700 | 0.624 | 0.704 | 0.735 |
| RAkEL+BN | **0.707** | **0.718** | **0.719** | **0.671** | **0.706** | 0.718 |

"acc." refers to "accuracy", "rec." refers to "recall", "subAcc." refers to "subsetAccuracy", "micRec." refers to "micro recall", "micF1." refers to "micro F1".

#### 4.2.2. Results and analyses of multi-label recognition

The multi-label classification results for the two datasets are summarized in Tables 4 and 5. From the two tables, we can obtain the following observations:

1. BR performs the worst among the four commonly used multi-label classifiers for both datasets. The reason may be that BR ignores the relation among labels, while other three methods consider pairwise relation or the randomly selected sub-combinations existing in the training label sets. It proves the importance of label relation for multi-label classification.
2. Our approach outperforms the four commonly used multi-label classifiers, since both example-based and label-based measures of our approach are better than those of four commonly used multi-label classifiers in most cases for both two datasets. It demonstrates the effectiveness of our approach, since it can more effectively capture the dependencies among labels. Furthermore, our method increases the example-based accuracy, example-based F1, and example-based subset accuracy in most cases. It indicates that our method increases the completely correctly predicted examples, and thus not only improves the recognition

**Table 6**
The results obtained using Zhang's work and our method.

| Dataset | Method | Example-based | | | | Label-based | | computational costs(seconds) |
|---|---|---|---|---|---|---|---|---|
| | | acc. | rec. | F1 | subAcc. | micRec. | micF1. | |
| emotions | zhang | 0.510 | 0.578 | 0.581 | 0.287 | 0.571 | 0.643 | **16** |
| | BR+BN | **0.552** | **0.684** | **0.629** | **0.322** | **0.687** | **0.660** | 27 |
| scene | zhang | 0.566 | 0.596 | 0.582 | 0.517 | 0.585 | 0.671 | 583 |
| | BR+BN | **0.668** | **0.679** | **0.680** | **0.632** | **0.668** | **0.680** | **334** |
| image | zhang | 0.462 | 0.489 | 0.491 | 0.378 | 0.462 | 0.568 | 444 |
| | BR+BN | **0.555** | **0.577** | **0.588** | **0.458** | **0.549** | **0.587** | **218** |
| yeast | zhang | 0.502 | 0.579 | 0.612 | 0.151 | 0.573 | 0.636 | 730 |
| | BR+BN | **0.521** | **0.629** | **0.617** | **0.230** | **0.623** | **0.639** | **631** |

"acc." refers to "accuracy","rec." refers to "recall", "subAcc." refers to "subsetAccuracy", "micRec." refers to "micro recall", "micF1." refers to "micro F1". the number in parentheses below the dataset name represents the scale of the dataset, which is calculated as the production of number of the instances and number of the multi labels.

accuracy, but also makes the recognition results more balanced from the view of instance.

#### 4.2.3. Comparison with related work

We compare our method with the most related work, Zhangs' work [15]. Zhangs' method used an existing structure learning algorithm to learn the BN to capture the dependencies among label errors. Because of limitation with our BN structure learning algorithm, we compare our method with Zhangs' only on four image datasets: emotions [1], scene [2], image [25] and yeast [26] datasets. The yeast gene dataset contains 2417 genes, each of them is represented by a 103-dimensional feature vector. There are 14 possible class labels. The image dataset consists of 2000 natural scene images associated with up to 5 labels. For each image, 294 features as same as those of scene dataset are provided. Table 6 lists the experimental results.

From the table, we can find for all the four datasets, our method outperforms Zhangs' method in terms of most parameters, which demonstrates the superiority of our method to Zhangs' method.

We further compare the computational cost of our method with Zhangs's. The experimental conditions are as follows: the CPU is 2.53 GHz, the RAM is 12G and the version of matlab is R2012a with windows server 2008 system. The 10-cross validation running times of our methods and Zhangs' are listed in the last column of Table 6. From Table 6, we can find that on three datasets, i.e., image, yeast and scene, the computational cost of our method is less than Zhangs'. It demonstrates that our method is more effective than Zhangs' in most cases. For emotions dataset, Zhangs' is lower than ours. We further analyze the learned structure for emotions dataset, and find that the learned structure of label errors is simpler than the learned structure of labels. This explains why Zhangs' computational cost is lower than that of ours on this dataset.

### 4.3. Experimental results and analyses for incomplete labels

We investigate robustness of the proposed method with respect to missing labels, and compare it with BR and BR with BN. We simulate missing labels by randomly removing $X\%$ of the labels where $X=10$, 20, 30, 40 from the training data. Our proposed method handles incomplete labels by structural EM, while BR and BR with BN are trained with only the complete portion of the data. 10-fold cross validation is adopted.

Tables 7 and 8 show the results on emotions and scene datasets. From the two tables, we can obtain the following observations:

1. The performance of BR monotonically decreases when the portion of incomplete labels increases. It indicates the negative effect of the incomplete labels on the multi-label classifier that ignores the relation among labels.

**Table 7**
Results of incomplete labels in emotions dataset.

| Proportion (%) | Method | Example-based | | | | Label-based | |
|---|---|---|---|---|---|---|---|
| | | acc. | rec. | F1 | subAcc. | micRec. | micF1. |
| 10 | BR | 0.508 | 0.576 | 0.585 | 0.268 | 0.573 | 0.647 |
| | BR+BN | 0.561 | 0.693 | 0.644 | 0.314 | 0.689 | 0.671 |
| | Ours | **0.571** | **0.708** | **0.654** | **0.322** | **0.703** | **0.679** |
| 20 | BR | 0.507 | 0.576 | 0.584 | 0.272 | 0.577 | 0.645 |
| | BR+BN | 0.554 | 0.681 | 0.633 | 0.320 | 0.683 | 0.669 |
| | Ours | **0.567** | **0.702** | **0.650** | **0.320** | **0.699** | **0.676** |
| 30 | BR | 0.504 | 0.579 | 0.582 | 0.265 | 0.576 | 0.644 |
| | BR+BN | 0.564 | 0.706 | 0.647 | 0.320 | 0.706 | 0.672 |
| | Ours | **0.579** | **0.720** | **0.663** | **0.332** | **0.718** | **0.688** |
| 40 | BR | 0.503 | 0.574 | 0.582 | 0.258 | 0.571 | 0.646 |
| | BR+BN | 0.560 | 0.691 | 0.643 | 0.309 | 0.685 | 0.667 |
| | Ours | **0.571** | **0.707** | **0.657** | **0.320** | **0.705** | **0.680** |

"acc." refers to "accuracy", "rec." refers to "recall", "subAcc." refers to "subsetAccuracy", "micRec." refers to "micro recall", "micF1." refers to "micro F1".

**Table 8**
Results of incomplete labels in scene dataset.

| Proportion (%) | Method | Example-based | | | | Label-based | |
|---|---|---|---|---|---|---|---|
| | | acc. | rec. | F1 | subAcc. | micPre. | micF1. |
| 10 | BR | 0.603 | 0.653 | 0.626 | 0.536 | 0.642 | 0.689 |
| | BR+BN | 0.659 | 0.670 | 0.671 | 0.624 | 0.659 | 0.671 |
| | Ours | **0.667** | **0.678** | **0.679** | **0.632** | **0.667** | **0.679** |
| 20 | BR | 0.602 | 0.653 | 0.625 | 0.535 | 0.642 | 0.688 |
| | BR+BN | 0.663 | 0.674 | 0.675 | 0.627 | 0.663 | 0.674 |
| | Ours | **0.669** | **0.679** | **0.681** | **0.633** | **0.668** | **0.680** |
| 30 | BR | 0.598 | 0.648 | 0.621 | 0.532 | 0.638 | 0.687 |
| | BR+BN | 0.652 | 0.661 | 0.662 | 0.620 | 0.648 | 0.661 |
| | Ours | **0.667** | **0.677** | **0.679** | **0.631** | **0.667** | **0.679** |
| 40 | BR | 0.600 | 0.647 | 0.622 | 0.535 | 0.637 | 0.687 |
| | BR+BN | 0.655 | 0.665 | 0.666 | 0.622 | 0.653 | 0.665 |
| | Ours | **0.662** | **0.672** | **0.674** | **0.625** | **0.662** | **0.674** |

"acc." refers to "accuracy", "rec." refers to "recall", "subAcc." refers to "subsetAccuracy", "micRec." refers to "micro recall", "micF1." refers to "micro F1".

2. The performance of BR with BN fluctuates when the portion of incomplete labels increases. The reason may be that the modified relation among labels makes the classifier robust to incomplete labels to certain extent.
3. The performance of our proposed method is better than both BR and BR with BN, since most of the example-based and label-based measures of our method are better than those of BR and BR with BN. It validates the effectiveness of our algorithm for

incomplete labels. Furthermore, the improvements of the example-based accuracy, example-based F1, and the label-based F1 indicate that our method not only improves the recognition accuracy, but also makes the recognition results more balanced from the view of both example and label. Our method can predict more completely correct samples, since the subset accuracy increases in most cases.

## 5. Conclusion

In this work, we propose a method to enhance multi-label classification by modeling dependencies among labels using BN. First, the label measurements are obtained using traditional multi-label classification methods. Second, BN is used to model the dependencies among different labels. It is different from the traditional direct methods which ignore the dependencies among labels or just consider pairwise or fixed label combinations. The experimental results on two multi-label databases show that our approach can effectively capture the co-occurrent and the mutual exclusive relation among labels, and thus, our approach outperforms other methods. The relation modeled by our approach is more flexible than that of pairwise or fixed subset labels captured by current direct multi-label learning methods. Furthermore, we extend our approach to deal with incomplete labels by structural EM algorithm and relationship among labels. Experiments on two datasets show the effectiveness of our multi-label learning algorithm with incomplete labels.

Our method has two limitations. First, when the number of labels is large, it is difficult to effectively learn the structure of the BN using our software. More effective structure learning method should be considered in our future work. Second, the BN only approximates the label dependencies since they can be better captured by the undirected graphical models. However, structure learning of undirected graphical models is very challenging. An effective and accurate label dependence model should be studied in the near future.

## Conflict of interest statement

None declared.

## Acknowledgments

This work has been supported by National Program 863 (2008AA01Z122), the National Science Foundation of China (Grant No. 61175037, 61228304), project from Anhui Science and Technology Agency (1106c0805008) and the Fundamental Research Funds for the Central Universities.

## References

[1] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, Ioannis Vlahavas, Multi-label classification of music into emotions. in: ISMIR 2008: Proceedings of the 9th International Conference of Music Information Retrieval, Lulu.com, 2008, pp. 325–330.

[2] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, Christopher M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.

[3] Cassio P. de Campos, Qiang Ji, Efficient structure learning of Bayesian networks using constraints, J. Mach. Learn. Res. 12 (2011) 663–689.

[4] Grigorios Tsoumakas, Ioannis Katakis, Ioannis Vlahavas, Mining Multi-Label data, Data Mining and Knowledge Discovery Handbook, 2010, pp. 667–685.

[5] Grigorios Tsoumakas, Ioannis Vlahavas, Random *k*-labelsets: an ensemble method for multilabel classification, in: Machine Learning: ECML 2007, 2007, pp. 406–417.

[6] E. Spyromitros, G. Tsoumakas, I. Vlahavas, An empirical study of lazy multi-label classification algorithms, in: Artificial Intelligence: Theories, Models and Applications, 2008, pp 401–406.

[7] Minling Zhang, Zhihua Zhou, Ml-knn: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[8] Robert E. Schapire, Yoram Singer, Boostexter: a boosting-based system for text categorization, Mach. Learn. 39 (2) (2000) 135–168.

[9] Grigorios Tsoumakas, Ioannis Katakis, Multi label classification: an overview, Int. J. Data Warehous. Min. 3 (3) (2007) 1–13.

[10] André C.P.L.F. de Carvalho, Alex A. Freitas, A tutorial on multi-label classification techniques, Foundations of Computational Intelligence vol. 5, Springer, 2009, pp. 177–195.

[11] Shantanu Godbole, Sunita Sarawagi, Discriminative methods for multi-labeled classification, in: Proceedings of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2004, pp. 22–30.

[12] Geoff Holmes Jesse Read, Bernhard Pfahringer, Eibe Frank, Classifier chains for multi-label classification, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, ECML PKDD '09, Berlin, Heidelberg, Springer-Verlag, 2009, pp. 254–269.

[13] Nadia Ghamrawi, Andrew McCallum, Collective multi-label classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 195–200.

[14] Liang Sun, Shuiwang Ji, Jieping Ye, Hypergraph spectral learning for multi-label classification, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 668–676.

[15] Minling Zhang, Kun Zhang, Multi-label learning by exploiting label dependency, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 999–1008.

[16] Sheng-Jun Huang, Yang Yu, Zhi-Hua Zhou, Multi-label hypothesis reuse, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 525–533.

[17] Serhat Selcuk Bucak, Rong Jin, Anil K Jain, Multi-label learning with incomplete class assignments, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 2801–2808.

[18] Janardhan Rao Doppa, Shahed Sorower, Mohammad Nasresfahani, Walker Orr, Thomas G. Dietterich, Xiaoli Fern, Prasad Tadepalli, Jed Irvine, Learning rules from incomplete examples via implicit mention models, J. Mach. Learn. Res.: Proc. Track 20 (2011) 197–212.

[19] Gideon Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.

[20] Nir Friedman, The bayesian structural em algorithm, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 129–138.

[21] Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco, CA, USA, 1988.

[22] Daphne Koller, Nir Friedman, Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning Series), The MIT Press, Cambridge, Massachusetts, USA, 2009.

[23] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, Klaus Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.

[24] Mohammad S Sorower, A Literature Survey on Algorithms for Multi-Label Learning. Technical Report, Oregon State University, 2010.

[25] Min-Ling Zhang, Zhi-Hua Zhou, ML-KNN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[26] André Elisseeff, Jason Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, 2001, pp. 681–687.

**Shangfei Wang** received the M.S. degree in circuits and systems, and the Ph.D. degree in signal and information processing from University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1999 and 2002 respectively. From 2004 to 2005, she was a postdoctoral research fellow in Kyushu University, Japan. She is currently an Associate Professor of School of Computer Science and Technology, USTC. She is an IEEE member. Her research interests cover computation intelligence, affective computing, multimedia computing, information retrieval and artificial environment design. She has authored or co-authored over 50 publications.

**Jun Wang** received the B.S. degree in computer science and technology from China University of Geosciences, Wuhan, China, in 2012. She is currently pursuing the M.S. degree in computer science in the University of Science and Technology of China, Hefei, China. Her research interest is affective computing.

**Zhaoyu Wang** received the B.S. degree in School of Mathematics and Information Science from Anhui University of Technology, Ma Anshan, Anhui, China, in 2010. And he received the M.S. degree in computer science in the University of Science and Technology of China, Hefei, China, in 2013. His research interest is affective computing.

**Qiang Ji** received his Ph.D degree in Electrical Engineering from the University of Washington. He is currently a Professor with the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute (RPI). He recently served as a program director at the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute at University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at University of Nevada at Reno, and the US Air Force Research Laboratory. He currently serves as the director of the Intelligent Systems Laboratory (ISL) at RPI.

His research interests are in computer vision, probabilistic graphical models, information fusion, and their applications in various fields. He has published over 160 papers in peer-reviewed journals and conferences. His research has been supported by major governmental agencies including NSF, NIH, DARPA, ONR, ARO, and AFOSR as well as by major companies including Honda and Boeing. He is an editor on several related IEEE and international journals and he has served as a general chair, program chair, technical area chair, and program committee member in numerous international conferences/workshops. He is a fellow of IAPR.