Front.Comput.Sci.
DOI

REVIEW ARTICLE

Binary Relevance for Multi-Label Learning: An Overview

Min-Ling ZHANG (✉), Yu-Kun LI, Xu-Ying LIU, Xin GENG

[1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3] Collaborative Innovation Center for Wireless Communications Technology, China

Abstract Multi-label learning deals with the problem where each example is represented by a single instance while associated with multiple class labels simultaneously. Binary relevance is arguably the most intuitive solution to learn from multi-label examples, which works by decomposing the multi-label learning task into a number of *independent* binary learning tasks (one per class label). In view of its potential weakness of ignoring correlations among labels, many correlation-enabling extensions to binary relevance have been proposed in the past decade or so. In this paper, we aim to review the state-of-the-art of binary relevance from three aspects. Firstly, basic settings of multi-label learning and the binary relevance solution are briefly summarized. Secondly, representative strategies to endow binary relevance with the ability of label correlation exploitation are discussed. Thirdly, some of our recent studies on binary relevance towards issues other than label correlation exploitation are also introduced. As a conclusion, suggestions on future research directions are outlined.

Keywords Machine learning, multi-label learning, binary relevance, label correlation, class-imbalance, relative labeling-importance

1 Introduction

Multi-label learning is one of the popular learning frameworks to model real-world objects with multiple semantic meanings [1, 2]. For instance, in text categorization, a news document on government reform can cover multiple topics such as *politics*, *economics*, and *society* [3]; in image classification, a natural scene image can depict multiple sceneries such as *sky*, *sand*, *sea* and *yacht* [4]. Generally, multi-label objects widely exist in real-world applications including information retrieval [5], bioinformatics [6], multimedia content annotation [7], web mining [8], etc.

The goal of multi-label learning is to induce a multi-label predictor which can assign a set of relevant labels for the unseen instance. To achieve this, the most intuitive solution is to learn one binary classifier for each class label, where the relevancy of each class label for the unseen instance is determined by the prediction yielded by the corresponding binary classifier [9]. Specifically, this *binary relevance* procedure works in an *independent* manner where the binary classifier for each class label is learned by ignoring the co-existence of other class labels. Due to its conceptual simplicity, binary relevance has attracted considerable attentions in multi-label learning researches.[1)]

Nonetheless, a consensus assumption on multi-label learning lies in that the correlations among labels should be well exploited in order to build multi-label prediction models with strong generalization performance [1, 2, 10, 11]. The decomposition nature of binary relevance leads to its incapability in exploiting label correlations, and therefore many correlation-enabling extensions to binary relevance have been proposed in the past decade or so [12–29]. Generally, representative strategies to endow binary relevance with the ability of label correlation exploitation include the *chaining* structure assuming random label correlations, the *stacking* structure assum-

Received month dd, yyyy; accepted month dd, yyyy

E-mail: zhangml@seu.edu.cn

[1)] According to Google Scholar (by June 2017), the seminal work on binary relevance [9] has received more than 1100 citations.

ing full-order label correlations, and the *controlling* structure assuming pruned label correlations.

Although label correlation plays an essential role to induce effective multi-label learning models, recent studies show that some inherent properties of multi-label learning should be investigated as well in order to achieve strong generalization performance. On one hand, class labels in the label space usually have *imbalanced distributions*, i.e. the number of positive instances w.r.t. each class label is far less than its negative counterparts [30–39]. On the other hand, class labels in the label space usually have *different labeling-importance*, i.e. the importance degree of each class label in characterizing the semantics of multi-label example is relative to each other [40–45]. Therefore, to enhance the generalization performance of binary relevance models, it is also beneficial to take those inherent properties into account along with label correlation exploitation in the learning procedure.

In this paper, we aim to give an overview on the state-of-the-art of binary relevance for multi-label learning. In Section 2, formal definitions on multi-label learning as well as the canonical binary relevance solution are briefly summarized. In Section 3, representative strategies to enable label correlation exploitation for binary relevance are discussed. In Section 4, some of our recent studies on related issues of binary relevance are also introduced. Finally, Section 5 concludes by suggesting several future research directions on binary relevance.

2 Binary Relevance

Let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional instance space, and $\mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ denote the label space consisting of q class labels. Then, the goal of multi-label learning is to induce a multi-label predictor $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from the multi-label training set $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$. Here, for each multi-label training example $(\boldsymbol{x}^i, \boldsymbol{y}^i)$, $\boldsymbol{x}^i \in \mathcal{X}$ is a d-dimensional feature vector $[x_1^i, x_2^i, \ldots, x_d^i]^\top$ and $\boldsymbol{y}^i \in \{-1, +1\}^q$ is a q-bits binary vector $[y_1^i, y_2^i, \ldots, y_q^i]^\top$ with $y_j^i = +1$ (-1) indicating y_j^i is a relevant (irrelevant) label for $\boldsymbol{x}^i$.[2)] Equivalently, the set of relevant labels $Y^i \subseteq \mathcal{Y}$ for $\boldsymbol{x}^i$ corresponds to $Y^i = \{\lambda_j \mid y_j^i = +1,\ 1 \le j \le q\}$. Given an unseen instance $\boldsymbol{x}^* \in \mathcal{X}$, its relevant label set Y^* is predicted as $Y^* = f(\boldsymbol{x}^*) \subseteq \mathcal{Y}$.

Binary relevance is arguably the most intuitive solution to learn from multi-label training examples [1, 2], which decomposes the multi-label learning problem into q independent binary learning problems. Here, each binary classification problem corresponds to one class label in the label space $\mathcal{Y}$ [9]. Specifically, for each class label λ_j, binary relevance derives a binary training set $\mathcal{D}_j$ from the original multi-label training set $\mathcal{D}$ in the following way:

$$\mathcal{D}_j = \{(\boldsymbol{x}^i, y_j^i) \mid 1 \le i \le m\} \tag{1}$$

In other words, each multi-label training example $(\boldsymbol{x}^i, \boldsymbol{y}^i)$ is transformed into a binary training example based on its *relevancy* to λ_j.

After that, a binary classifier $g_j : \mathcal{X} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_j$ by applying some binary learning algorithm $\mathcal{B}$, i.e. $g_j \leftarrow \mathcal{B}(\mathcal{D}_j)$. Therefore, the multi-label training example $(\boldsymbol{x}^i, \boldsymbol{y}^i)$ will contribute to the learning process of all binary classifiers g_j $(1 \le j \le q)$, where $\boldsymbol{x}^i$ is utilized as positive (negative) training example in inducing g_j based on its relevancy (irrelevancy) to λ_j.[3)]

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by querying the outputs of each binary classifier:

$$Y^* = \{\lambda_j \mid g_j(\boldsymbol{x}^*) > 0,\ 1 \le j \le q\} \tag{2}$$

As shown in Eq.(2), the predicted label set Y^* would be empty when all binary classifiers yield negative outputs on $\boldsymbol{x}^*$. In this case, one might choose the so-called *T-Criterion* [9] to predict the class label with *greatest* (least negative) output. Other criteria to aggregate the outputs of binary classifiers can be found in [9].

Table 1 summarizes the pseudo-code of binary relevance. As shown in Table 1, there are several properties which are noteworthy for binary relevance:

- Firstly, the prominent property of binary relevance lies in its conceptual simplicity. Specifically, binary relevance is a *first-order* approach which builds the classification model in a label-by-label manner and thus ignores the co-existence of other class labels. The modeling complexity of binary relevance is linear to the number of class labels (i.e. q) in the label space;
- Secondly, binary relevance falls into the category of *problem transformation* approaches, which solve multi-label learning problem by transforming it into other well-established learning scenarios (binary classification in this case) [1, 2]. Accordingly, binary relevance is not restricted to particular learning techniques and can be instantiated with any binary learning algorithm with diverse characteristics;

[2)] Without loss of generality, binary assignment of each class label is represented by +1 and -1 (other than 1 and 0) in this paper.

[3)] In the seminal literature on binary relevance [9], this training procedure is also termed as *cross-training*.

Table 1 The pseudo-code of binary relevance [9].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ $(\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1,+1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\})$

$\mathcal{B}$: the binary learning algorithm

$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j$ according to Eq.(1);
  Induce the binary classifier $g_j :\leftarrow \mathcal{B}(\mathcal{D}_j)$;
end for
return $Y^* = \{\lambda_j \mid g_j(\boldsymbol{x}^*) > 0,\ 1 \le j \le q\}$

- Thirdly, binary relevance optimizes the macro-averaged *label-based* multi-label evaluation metrics, which evaluate the learning system's performance on each class label separately and then return the mean value across all class labels. Therefore, the actual multi-label metric being optimized depends on the binary loss which is minimized by the binary learning algorithm $\mathcal{B}$ [46,47];
- Fourthly, binary relevance can be easily adapted to learn from multi-label examples with missing labels, where the labeling information for training examples are incomplete due to factors such as high labeling cost, carelessness of human labelers, etc. [48–50]. To accommodate this situation, binary relevance can derive the binary training set in Eq.(1) by simply excluding those examples whose labeling information y^i_j is not available.

3 Correlation-Enabling Extensions

As discussed in Section 2, binary relevance has been widely-used for multi-label modeling due to its simplicity and other attractive properties. Nonetheless, one potential weakness of binary relevance lies in its ignorance of exploiting label correlations to improve the learning system's generalization performance [1, 2]. Therefore, a natural consideration is trying to endow binary relevance with the ability of label correlation exploitation while at the same time still retain its linear modeling complexity w.r.t. the number of class labels.

In light of the above consideration, significant number of correlation-enabling extensions have been proposed following the seminal work on binary relevance. In the following, three representative extension strategies are discussed respectively, including the chaining structure assuming *random* label correlations [12–18], the stacking structure assuming *full-order* label correlations [19–23], and the controlling structure assuming *pruned* label correlations [24–29].

3.1 Binary Relevance with Chaining Structure

In the chaining structure, a total of q binary classifiers are induced according to a chaining order specified over the class labels. Specifically, one binary classifier is built for each class label based on the predictions of preceding classifiers in the chain [12, 14].

Given the label space $\mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$, let $\pi : \{1, 2, \ldots, q\} \mapsto \{1, 2, \ldots, q\}$ be the permutation used to specify a chaining order over all the class labels, i.e. $\lambda_{\pi(1)} \succ \lambda_{\pi(2)} \succ \cdots \succ \lambda_{\pi(q)}$. Thereafter, for the j-th class label $\lambda_{\pi(j)}$ in the ordered list, the *classifier chain* approach [12, 14] works by deriving a corresponding binary training set $\mathcal{D}_{\pi(j)}$ from $\mathcal{D}$ in the following way:

$$\mathcal{D}_{\pi(j)} = \left\{ \left(\left[\boldsymbol{x}^i, y^i_{\pi(1)}, \ldots, y^i_{\pi(j-1)} \right], y^i_{\pi(j)} \right) \,\middle|\, 1 \le i \le m \right\} \tag{3}$$

Here, the binary assignments of preceding class labels in the chain, i.e. $\left[y^i_{\pi(1)}, \ldots, y^i_{\pi(j-1)} \right]$, are treated as extra features to append the original instance $\boldsymbol{x}^i$.

After that, a binary classifier $g_{\pi(j)} : \mathcal{X} \times \{-1,+1\}^{j-1} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_{\pi(j)}$ by applying some binary learning algorithm $\mathcal{B}$, i.e. $g_{\pi(j)} \leftarrow \mathcal{B}(\mathcal{D}_{\pi(j)})$. In other words, $g_{\pi(j)}$ determines the relevancy of $\lambda_{\pi(j)}$ through exploiting its correlations with preceding labels $\lambda_{\pi(1)}, \ldots, \lambda_{\pi(j-1)}$ in the chain.

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by iteratively querying the outputs of each binary classifier along the chaining order. Let $\eta^{\boldsymbol{x}^*}_{\pi(j)} \in \{-1,+1\}$ denote the predicted binary assignment of $\lambda_{\pi(j)}$ on $\boldsymbol{x}^*$, which are recursively determined as follows:

$$\begin{aligned} \eta^{\boldsymbol{x}^*}_{\pi(1)} &= \mathrm{sign}\left[g_{\pi(1)}(\boldsymbol{x}^*) \right] \\ \eta^{\boldsymbol{x}^*}_{\pi(j)} &= \mathrm{sign}\left[g_{\pi(j)}\left(\left[\boldsymbol{x}^*, \eta^{\boldsymbol{x}^*}_{\pi(1)}, \ldots, \eta^{\boldsymbol{x}^*}_{\pi(j-1)} \right] \right) \right] \end{aligned} \tag{4}$$

Here, sign$[\cdot]$ represents the signed function. Accordingly, the relevant label set Y^* turns out to be:

$$Y^* = \left\{ \lambda_{\pi(j)} \mid \eta^{\boldsymbol{x}^*}_{\pi(j)} = +1,\ 1 \le j \le q \right\} \tag{5}$$

Table 2 summarizes the pseudo-code of classifier chain. As shown in Table 2, classifier chain is a *high-order* approach which considers correlations among labels in a random manner specified by the permutation π. To account for

Table 2 The pseudo-code of classifier chain [12, 14].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ ($\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1,+1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$)

π: the permutation used to specify chaining order

$\mathcal{B}$: the binary learning algorithm

$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_{\pi(j)}$ according to Eq.(3);
  Induce the binary classifier $g_{\pi(j)} :\hookleftarrow \mathcal{B}(\mathcal{D}_{\pi(j)})$;
end for
Determine the binary assignments $\eta^{\boldsymbol{x}^*}_{\pi(j)}$ ($1 \le j \le q$) according to Eq.(4);
return $Y^* = \left\{\lambda_{\pi(j)} \mid \eta^{\boldsymbol{x}^*}_{\pi(j)} = +1,\ 1 \le j \le q\right\}$ w.r.t. Eq.(4)

the randomness introduced by permutation ordering, an effective choice is to build an *ensemble* of classifier chains with n random permutations $\{\pi^r \mid 1 \le r \le n\}$. One classifier chain can be learned according to each random permutation and the outputs from all classifier chains are aggregated to yield the final prediction [12, 14, 16].

It is also worth noting that predictive errors incurred in preceding classifiers would be propagated to subsequent classifiers along the chain, and these undesirable influences become more pronounced if error-prone class labels happen to be placed at the starting chaining positions [12, 14, 28, 51]. Furthermore, in training phase the extra features used to append input space $\mathcal{X}$ correspond to the ground-truth labeling assignments (i.e. Eq.(3)), while in testing phase the extra features used to append $\mathcal{X}$ correspond to predicted labeling assignments (i.e. Eq.(4)). One way to rectify this discrepancy is to replace the extra features $\left[y^i_{\pi(1)}, \ldots, y^i_{\pi(j-1)}\right]$ in Eq.(3) with $\left[\eta^{\boldsymbol{x}^i}_{\pi(1)}, \ldots, \eta^{\boldsymbol{x}^i}_{\pi(j-1)}\right]$, so that the predicted labeling assignments are used to append $\mathcal{X}$ in both the training and testing phases [17, 51].

From statistical point of view, the task of multi-label learning is equivalent to learn the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ with $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \{-1,+1\}^q$. Accordingly, $p(\boldsymbol{y} \mid \boldsymbol{x})$ can be factorized w.r.t. the chaining order specified by π as follows:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{q} p\left(y_{\pi(j)} \mid \boldsymbol{x}, y_{\pi(1)}, \ldots, y_{\pi(j-1)}\right) \tag{6}$$

Here, each term in the RHS of Eq.(6) represents the conditional probability of observing $y_{\pi(j)}$ given $\boldsymbol{x}$ and its preceding labels in the chain. Specifically, this term can be estimated by utilizing binary learning algorithm $\mathcal{B}$ which is capable of yielding probabilistic outputs (e.g. Naive Bayes). Thereafter, relevant label set for the unseen instance is predicted by performing exact inference [13] or approximate inference (when q is large) over the probabilistic classifier chain [15, 18].

3.2 Binary Relevance with Stacking Structure

In the stacking structure, a total of $2q$ binary classifiers are induced by stacking a set of q meta-level binary relevance models over another set of q base-level binary relevance models. Specifically, each meta-level binary classifier is built upon the predictions of all base-level binary classifiers [19].

Following the notations in Section 2, let g_j ($1 \le j \le q$) denote the set of base-level classifiers learned by invoking the standard binary relevance procedure on the multi-label training set, i.e. $g_j \hookleftarrow \mathcal{B}(\mathcal{D}_j)$. Thereafter, for each class label λ_j, the *stacking aggregation* approach [1, 19] derives a meta-level binary training set $\mathcal{D}^M_j$ in the following way:

$$\mathcal{D}^M_j = \left\{\left(\left[\boldsymbol{x}^i, \mathrm{sign}[g_1(\boldsymbol{x}^i)], \ldots, \mathrm{sign}[g_q(\boldsymbol{x}^i)]\right], y^i_j\right) \middle| \ 1 \le i \le m\right\} \tag{7}$$

Here, the signed prediction of base-level classifiers, i.e. $\left[\mathrm{sign}[g_1(\boldsymbol{x}^i)], \ldots, \mathrm{sign}[g_q(\boldsymbol{x}^i)]\right]$, are treated as extra features to append the original instance $\boldsymbol{x}^i$ in the meta-level.

After that, a meta-level classifier $g^M_j : \mathcal{X} \times \{-1,+1\}^q \mapsto \mathbb{R}$ can be induced from $\mathcal{D}^M_j$ by applying some binary learning algorithm $\mathcal{B}$, i.e. $g^M_j \hookleftarrow \mathcal{B}(\mathcal{D}^M_j)$. In other words, g^M_j determines the relevancy of λ_j through exploiting its correlations with all the class labels.

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by feeding the outputs of base-level classifiers as extra inputs to the meta-level classifiers:

$$Y^* = \left\{\lambda_j \mid g^M_j(\tau^{\boldsymbol{x}^*}) > 0,\ 1 \le j \le q\right\}$$
$$\text{where } \tau^{\boldsymbol{x}^*} = \left[\boldsymbol{x}^*, \mathrm{sign}[g_1(\boldsymbol{x}^*)], \ldots, \mathrm{sign}[g_q(\boldsymbol{x}^*)]\right] \tag{8}$$

Table 3 summarizes the pseudo-code of stacking aggregation. As shown in Table 3, stacking aggregation is a *full-order* approach which assumes that each class label has correlations with all the other class labels. It is worth noting that stacking aggregation employs ensemble learning [52] to combine two sets of binary relevance models with deterministic label correlation exploitation, while ensemble learning can also be

Table 3 The pseudo-code of stacking aggregation [19].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ $(\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1,+1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\})$

$\mathcal{B}$: the binary learning algorithm

$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j$ according to Eq.(1);
  Induce the base-level binary classifier $g_j :\hookleftarrow \mathcal{B}(\mathcal{D}_j)$;
end for
for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j^M$ according to Eq.(7);
  Induce the meta-level binary classifier $g_j^M :\hookleftarrow \mathcal{B}(\mathcal{D}_j^M)$;
end for
return $Y^* = \left\{\lambda_j \mid g_j^M(\tau^{x^*}) > 0,\ 1 \le j \le q\right\}$ w.r.t Eq.(8)

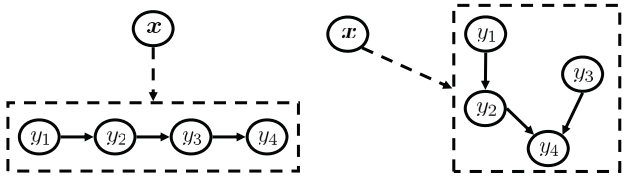
applied to classifier chain to account for its randomness of label correlation exploitation.

Other than employing the outputs of base-level classifiers $\left[\text{sign}[g_1(\boldsymbol{x}^i)], \ldots, \text{sign}[g_q(\boldsymbol{x}^i)]\right]$ to append the inputs of meta-level classifiers, it is also feasible to employ the ground-truth labeling assignments $\left[y_1^i, \ldots, y_q^i\right]$ to instantiate the meta-level binary training set (i.e. Eq.(7)) [21]. Nonetheless, similar to the standard classifier chain approach, this practice would also lead to the discrepancy issue regarding the extra features used to append the input space $\mathcal{X}$ in the training and testing phases.

There are also other ways to make use of the stacking strategy to induce multi-label prediction model. Given the base-level classifiers g_j ($1 \le j \le q$) and the meta-level classifiers g_j^M ($1 \le j \le q$), other than only relying on meta-level classifiers to yield final predictions (i.e. Eq(8)), one can also aggregate the outputs of both base-level and meta-level classifiers to accomplish the task [20]. Furthermore, other than using the binary labeling assignments as extra features for stacking, one can also adapt specific techniques to generate tailored features for stacking such as discriminant analysis [22] or rule learning [23].

3.3 Binary Relevance with Controlling Structure

In the controlling structure, a total of $2q$ binary classifiers are induced based on some dependency structure specified over the class labels. Specifically, one binary classifier is built for each class label by exploiting pruned predictions of q binary relevance models [25].



(a) Examplar DAG structure I (b) Examplar DAG structure II

Fig. 1 Examples of two Bayesian network (DAG) structures with $\boldsymbol{x}$ serving as the common parent. The conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ factorizes according to either structure as: (a) $p(\boldsymbol{y} \mid \boldsymbol{x}) = p(y_1 \mid \boldsymbol{x}) \cdot p(y_2 \mid y_1, \boldsymbol{x}) \cdot p(y_3 \mid y_2, \boldsymbol{x}) \cdot p(y_4 \mid y_3, \boldsymbol{x})$; (b) $p(\boldsymbol{y} \mid \boldsymbol{x}) = p(y_1 \mid \boldsymbol{x}) \cdot p(y_2 \mid y_1, \boldsymbol{x}) \cdot p(y_3 \mid \boldsymbol{x}) \cdot p(y_4 \mid y_2, y_3, \boldsymbol{x})$.

Bayesian network (or *directed acyclic graph*, DAG) is a convenient tool to explicity characterize correlations among class labels in a compact way [25–27]. As mentioned in Subsection 3.1, a statistical equivalence to multi-label learning corresponds to model the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ with $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} = \{-1,+1\}^q$. Given the Bayesian network structure $\mathcal{G}$ specified over $(\boldsymbol{x}, \boldsymbol{y})$, the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ can be factorized according to $\mathcal{G}$ as follows:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{q} p(y_j \mid \mathbf{pa}_j, \boldsymbol{x}) \tag{9}$$

Here, $\boldsymbol{x}$ serves as the common parent for each y_j ($1 \le j \le q$) as all class labels inherently depend on the feature space $\mathcal{X}$. In addition, $\mathbf{pa}_j$ represents the set of parent class labels of y_j implied by $\mathcal{G}$. Figure 1 illustrates two examples of how the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ can be factorized according to the given Bayesian network structure.

To learn the Bayesian network structure $\mathcal{G}$ from multi-label training set $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ is difficult, where existing Bayesian network learning techniques [53] are not directly applicable due to two major reasons. On one hand, variables in the Bayesian network have mixed types with $\boldsymbol{y}$ (class labels) being discrete and $\boldsymbol{x}$ (feature vector) being continuous. On the other hand, computational complexity would be prohibitively high when the input dimensionality (i.e. number of features) is too large.

Obviously, the above two issues are brought by the involvement of feature vector $\boldsymbol{x}$ in learning the Bayesian network structure. In light of this, the *LEAD* approach [25] chooses to eliminate the effect of features to simplify the Bayesian network generation procedure. Following the notations in Section 2, let g_j ($1 \le j \le q$) denote the binary classifiers induced by standard binary relevance procedure, i.e. $g_j \hookleftarrow \mathcal{B}(\mathcal{D}_j)$. Accordingly, a set of *error* random vari-

ables are derived so as to decouple the influences of $\boldsymbol{x}$ from all class labels:

$$e_j = y_j - \mathrm{sign}(g_j(\boldsymbol{x})) \ \ (1 \le j \le q) \tag{10}$$

Thereafter, the Bayesian network structure $\mathcal{G}$ among all class labels (conditioned on $\boldsymbol{x}$) can be learned from e_j ($1 \le j \le q$) with off-the-shelf packages [54–56].

Based on the DAG structure implied by $\mathcal{G}$, for each class label λ_j, the LEAD approach derives a binary training set $\mathcal{D}_j^{\mathcal{G}}$ from $\mathcal{D}$ in the following way:

$$\mathcal{D}_j^{\mathcal{G}} = \left\{ \left(\left[\boldsymbol{x}^i, \mathbf{pa}_j^i \right], y_j^i \right) \;\middle|\; 1 \le i \le m \right\} \tag{11}$$

Here, the binary assignments of parent class labels, i.e. $\mathbf{pa}_j^i$, are treated as extra features to append the original instance $\boldsymbol{x}^i$.

After that, a binary classifier $g_j^{\mathcal{G}} : \mathcal{X} \times \{-1,+1\}^{|\mathbf{pa}_j|} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_j^{\mathcal{G}}$ by applying some binary learning algorithm $\mathcal{B}$, i.e. $g_j^{\mathcal{G}} \leftarrowtail \mathcal{B}(\mathcal{D}_j^{\mathcal{G}})$. In other words, $g_j^{\mathcal{G}}$ determines the relevancy of λ_j through exploiting its correlations with the parent class labels $\mathbf{pa}_j$ implied by $\mathcal{G}$.

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by iteratively querying the outputs of each binary classifier w.r.t. the Bayesian network structure. Let $\pi^{\mathcal{G}} : \{1,2,\ldots,q\} \mapsto \{1,2,\ldots,q\}$ be the causal order implied by $\mathcal{G}$ over all class labels, i.e. $\lambda_{\pi^{\mathcal{G}}(1)} \succ \lambda_{\pi^{\mathcal{G}}(2)} \succ \cdots \lambda_{\pi^{\mathcal{G}}(q)}$. Furthermore, let $\eta_{\pi^{\mathcal{G}}(j)}^{\boldsymbol{x}^*} \in \{-1,+1\}$ denote the predicted binary assignment of $\lambda_{\pi^{\mathcal{G}}(j)}$ on $\boldsymbol{x}^*$, which are recursively determined as follows:

$$\begin{aligned} \eta_{\pi^{\mathcal{G}}(1)}^{\boldsymbol{x}^*} &= \mathrm{sign}\left[g_{\pi^{\mathcal{G}}(1)}^{\mathcal{G}}(\boldsymbol{x}^*) \right] \\ \eta_{\pi^{\mathcal{G}}(j)}^{\boldsymbol{x}^*} &= \mathrm{sign}\left[g_{\pi^{\mathcal{G}}(j)}^{\mathcal{G}}\left(\left[\boldsymbol{x}^*, \langle \eta_a^{\boldsymbol{x}^*} \rangle_{y_a \in \mathbf{pa}_{\pi^{\mathcal{G}}(j)}} \right] \right) \right] \end{aligned} \tag{12}$$

Accordingly, the relevant label set Y^* turns out to be:

$$Y^* = \left\{ \lambda_{\pi^{\mathcal{G}}(j)} \mid \eta_{\pi^{\mathcal{G}}(j)}^{\boldsymbol{x}^*} = +1, \ 1 \le j \le q \right\} \tag{13}$$

Table 4 summarizes the pseudo-code of LEAD. As shown in Table 4, LEAD is a *high-order* approach which controls the order of correlations by the number of parents of each class label implied by $\mathcal{G}$. Similar to stacking aggregation, LEAD also employs ensemble learning to combine two sets of binary classifiers g_j ($1 \le j \le q$) and $g_j^{\mathcal{G}}$ ($1 \le j \le q$) to yield the multi-label prediction model. Specifically, predictions of the q binary classifiers g_j are *pruned* w.r.t. the parents for label correlation exploitation.

There are also other ways to consider pruned label correlations with specific controlling structure. Firstly, tree-based Bayesian network can be utilized as a simplified DAG structure where second-order label correlations are considered by pruning each class label with (up to) one parent [26,27]. Secondly, the stacking structure can be adapted to fulfill controlled label correlation exploitation by pruning uncorrelated outputs of base-level classifiers for stacking meta-level classifiers [24, 29]. Thirdly, class labels with error-prone predictions can even be filtered out from the pool of class labels for correlation exploitation [28].

Table 4 The pseudo-code of LEAD [25].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ ($\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1,+1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$)

$\mathcal{B}$: the binary learning algorithm

$\mathcal{L}$: the Bayesian network structure learning algorithm

$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j$ according to Eq.(1);
  Induce the binary classifier $g_j :\leftarrowtail \mathcal{B}(\mathcal{D}_j)$;
end for
Derive the *error* random variables e_j ($1 \le j \le q$) according to Eq.(10);
Learn the Bayesian network structure $\mathcal{G} \leftarrowtail \mathcal{L}(e_1, \ldots, e_q)$;
for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j^{\mathcal{G}}$ according to Eq.(11);
  Induce the binary classifier $g_j^{\mathcal{G}} :\leftarrowtail \mathcal{B}(\mathcal{D}_j^{\mathcal{G}})$;
end for
Specify the causal order $\pi^{\mathcal{G}}$ over all class labels w.r.t. $\mathcal{G}$;
return $Y^* = \left\{ \lambda_{\pi^{\mathcal{G}}(j)} \mid \eta_{\pi^{\mathcal{G}}(j)}^{\boldsymbol{x}^*} = +1, \ 1 \le j \le q \right\}$ w.r.t. Eq.(12)

4 Related Issues

As discussed in Section 3, to enhance binary relevance it is necessary to enable label correlation exploitation in the learning process. Nevertheless, it is also noteworthy that some inherent properties of multi-label learning should be investigated as well to further enhance the generalization performance of binary relevance. Specifically, recent studies on the issue of *class-imbalance*, i.e. the number of posi-

tive instances and negative instances w.r.t. each class label are imbalanced distributed [30–39], and the issue of *relative labeling-importance*, i.e. each class label has different labeling-importance [40–45], are introduced respectively.

4.1 Class-Imbalance

The issue of class-imbalance widely exist in multi-label learning tasks, especially when the label space consists of significant number of class labels. For each class label $\lambda_j \in \mathcal{Y}$, let $\mathcal{D}_j^+ = \{(\boldsymbol{x}^i, +1) \mid y_j^i = +1, 1 \le i \le m\}$ and $\mathcal{D}_j^- = \{(\boldsymbol{x}^i, -1) \mid y_j^i = -1, 1 \le i \le m\}$ denote the set of *positive* and *negative* training examples w.r.t. λ_j. Correspondingly, the level of class-imbalance can be characterized by the imbalance ratio:

$$ImR_j = \frac{\max\left(|\mathcal{D}_j^+|, |\mathcal{D}_j^-|\right)}{\min\left(|\mathcal{D}_j^+|, |\mathcal{D}_j^-|\right)} \tag{14}$$

Here, $|\cdot|$ returns the cardinality of a set and in most cases $|\mathcal{D}_j^+| < |\mathcal{D}_j^-|$ holds. Generally, the imbalance ratio is high for most benchmark multi-label data sets [1, 57]. For instance, among the 42 class labels of the `rcv1` benchmark data set, the average imbalance ratio (i.e. $\frac{1}{q}\sum_{j=1}^{q} ImR_j$) is greater than 15 and the maximum imbalance ratio (i.e. $\max_{1\le j\le q} ImR_j$) is greater than 50 [38].

To deal with the issue of class-imbalance in multi-label learning, existing approaches employ binary relevance as an intermediate step in the learning procedure. Specifically, by decomposing the multi-label learning task into q independent binary learning tasks, each of them will be addressed by prevalent binary imbalance learning techniques such as over-/under-sampling [32, 36, 37], thresholding the decision boundary [31, 33, 34], or optimizing imbalance-specific metric [30, 35, 39]. Obviously, as standard binary relevance has been applied prior to subsequent modeling, existing approaches deal with class-imbalance multi-label learning at the expense of ignoring exploitation of label correlations.

Therefore, a favorable solution to class-imbalance multi-label learning is to consider the exploitation of label correlations and the exploration of class-imbalance simultaneously. In light of this, the COCOA approach is proposed based on a specific strategy named *cross-coupling aggregation* [38]. For each class label λ_j, a binary classifier g_j^I is induced from $\mathcal{D}_j$ (i.e. Eq.(1)) by applying some binary imbalance learning algorithm $\mathcal{B}^I$ [58], i.e. $g_j^I \leftarrow\!\shortmid \mathcal{B}^I(\mathcal{D}_j)$. In addition, a random subset of K class labels $J_K \subset \mathcal{Y} \setminus \{\lambda_j\}$ are drawn for pairwise cross-coupling with λ_j. For each coupling label $\lambda_k \in J_K$, COCOA derives a tri-class training set $\mathcal{D}_{jk}^{\mathbf{tri}}$ for label pair (λ_j, λ_k) from $\mathcal{D}$ in the following way:

$$\mathcal{D}_{jk}^{\mathbf{tri}} = \{(\boldsymbol{x}^i, \psi^{\mathbf{tri}}(\boldsymbol{y}^i, \lambda_j, \lambda_k)) \mid 1 \le i \le m\}$$
$$\text{where } \psi^{\mathbf{tri}}(\boldsymbol{y}^i, \lambda_j, \lambda_k) = \begin{cases} 0, & \text{if } y_j^i = -1 \text{ and } y_k^i = -1 \\ +1, & \text{if } y_j^i = -1 \text{ and } y_k^i = +1 \\ +2, & \text{if } y_j^i = +1 \end{cases} \tag{15}$$

Among the three derived class labels, first two of them (i.e. 0 and +1) exploit label correlations by considering the joint labeling assignments of λ_j and λ_k w.r.t. $\boldsymbol{y}^i$, and the third class label (i.e. +2) corresponds to the interested case of λ_j being a relevant label.

After that, a tri-class classifier $g_{jk}^I : \mathcal{X} \times \{0, +1, +2\} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_{jk}^{\mathbf{tri}}$ by applying some multi-class imbalance learning algorithm $\mathcal{M}^I$ [59–61], i.e. $g_{jk}^I \leftarrow\!\shortmid \mathcal{M}^I(\mathcal{D}_{jk}^{\mathbf{tri}})$. In other words, a total of $K + 1$ classifiers including g_j^I and g_{jk}^I ($\lambda_k \in J_K$) have been induced for class label λ_j.

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by aggregating the predictions of the above classifiers induced by binary and multi-class imbalanced learning

Table 5 The pseudo-code of COCOA [38].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ ($\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1, +1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$)
$\mathcal{B}^I$: the binary imbalance learning algorithm
$\mathcal{M}^I$: the multi-class imbalance learning algorithm
K: the number of coupling class labels
$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

for $j = 1$ **to** q **do**
  Derive the binary training set $\mathcal{D}_j$ according to Eq.(1);
  Induce the binary classifier $g_j^I :\leftarrow\!\shortmid \mathcal{B}^I(\mathcal{D}_j)$;
  Draw a random subset $J_K \subset \mathcal{Y} \setminus \{\lambda_j\}$ with K coupling class labels;
  for each $\lambda_k \in J_K$ **do**
    Derive the tri-class training set $\mathcal{D}_{jk}^{\mathbf{tri}}$ according to Eq.(15);
    Induce the tri-class classifier $g_{jk}^I :\leftarrow\!\shortmid \mathcal{M}^I(\mathcal{D}_{jk}^{\mathbf{tri}})$;
  end for
end for
Return $Y^* = \{\lambda_j \mid f_j(\boldsymbol{x}^*) > t_j,\ 1 \le j \le q\}$ w.r.t. Eq.(16)

algorithms:

$$Y^* = \{\lambda_j \mid f_j(\boldsymbol{x}^*) > t_j,\ 1 \le j \le q\}$$
$$\text{where}\ \ f_j(\boldsymbol{x}^*) = g_j^I(\boldsymbol{x}^*) + \sum\nolimits_{\lambda_k \in J_K} g_{jk}^I(\boldsymbol{x}^*, +2) \qquad (16)$$

Here, t_j is the bipartition threshold which is set by optimizing certain empirical metric (e.g. F-measure) over $\mathcal{D}_j$.

Table 5 summarizes the pseudo-code of COCOA. As shown in Table 5, COCOA is a *high-order* approach which considers correlations among labels in a random manner via the K coupling class labels in J_K. Specifically, during the training phase, label correlation exploitation is enabled via an ensemble of pairwise cross-coupling between class labels. During the testing phase, class-imbalance exploration is enabled via aggregating classifiers induced from class-imbalance learning algorithms.

4.2 Relative Labeling-Importance

Existing approaches to multi-label learning, including binary relevance, take the common assumption of equal labeling-importance. Here, class labels associated with the training example are regarded to be relevant while their relative importance in characterizing the example's semantics are not differentiated [1, 2]. Nevertheless, the degree of labeling importance for each associated class label is generally different while not directly accessible from multi-label training examples. Figure 2 shows an exemplar multi-label natural scene image with descending relative labeling-importance: *sky* ≻ *water* ≻ *cloud* ≻ *building* ≻ *pedestrian*. Similar situations hold for other types of multi-label objects such as multi-category documents with different topical importance, multi-functionality gene with different expression levels, etc.

It is worth noting that there have been works on multi-label learning which aim to make use of *auxiliary* labeling-importance information. Different forms of auxiliary information exist such as *ordinal scale* over each class label [40], *full ranking* over relevant class labels [41], *importance distribution* over all class labels [43, 44], and *oracle feedbacks* over queried labels of unlabeled examples [45]. However, in standard multi-label learning, those auxiliary information are not assumed to be available and the only accessible labeling information are the relevancy/irrelevancy of each class label.

Intuitively, by leveraging the *implicit* relative labeling-importance information, further improvement on generalization performance of the multi-label learning system can be expected. In light of this, the RELIAB approach is proposed to incorporating relative labeling-importance information in the learning process [42]. Formally, for each instance

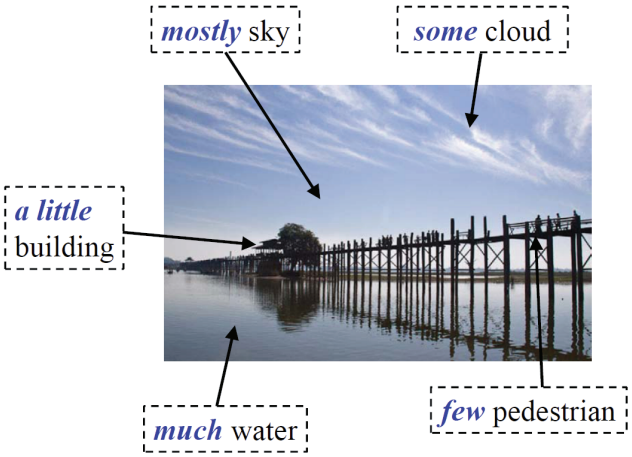


Fig. 2 An exemplar natural scene image annotated with multiple class labels. The relative labeling-importance of each annotation is also illustrated in this figure, which however is not explicitly provided by the annotator [42].

$\boldsymbol{x}$ and class label λ_j, the relative labeling-importance of λ_j in characterizing $\boldsymbol{x}$ is denoted as $\mu_{\boldsymbol{x}}^{\lambda_j}$. Specifically, these terms $\mu_{\boldsymbol{x}}^{\lambda_j}$ $(1 \le j \le q)$ satisfy the non-negativity constraint $\mu_{\boldsymbol{x}}^{\lambda_j} \ge 0$ and the normalization constraint $\sum_{j=1}^{q} \mu_{\boldsymbol{x}}^{\lambda_j} = 1$.

In the first stage, RELIAB estimates the implicit relative labeling-importance information $\mathcal{U} = \{\mu_{\boldsymbol{x}_i}^{\lambda_j} \mid 1 \le i \le m,\ 1 \le j \le q\}$ via iterative label propagation. Let $G = (V, E)$ be the fully-connected graph constructed over all the training examples with $V = \{\boldsymbol{x}^i \mid 1 \le i \le m\}$. Furthermore, a similarity matrix $\mathbf{W} = [w_{ik}]_{m \times m}$ is specified for G as follows:

$$\forall_{i,k=1}^{m}:\ \ w_{ik} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}^i - \boldsymbol{x}^k\|_2^2}{2\sigma^2}\right), & \text{if } i \ne k \\ 0, & \text{if } i = k \end{cases} \qquad (17)$$

Here, $\sigma > 0$ is the width constant for similarity calculation. Correspondingly, the label propagation matrix $\mathbf{P}$ is set as:

$$\mathbf{P} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$$
$$\text{where}\ \ \mathbf{D} = \text{diag}[d_1, d_2, \ldots, d_m]\ \ \text{with}\ \ d_i = \sum\nolimits_{k=1}^{m} w_{ik} \qquad (18)$$

In addition, the labeling-importance matrix $\mathbf{R} = [r_{ij}]_{m \times q}$ is initialized with $\mathbf{R}^{(0)} = \mathbf{\Phi} = [\phi_{ij}]_{m \times q}$ as follows:

$$\forall 1 \le i \le m,\ \forall 1 \le j \le q:\ \ \phi_{ij} = \begin{cases} 1, & \text{if } y_j^i = +1 \\ 0, & \text{if } y_j^i = -1 \end{cases} \qquad (19)$$

Then, the label propagation procedure works by iteratively updating $\mathbf{R}$ as: $\mathbf{R}^{(t)} = \alpha\mathbf{P}\mathbf{R}^{(t-1)} + (1-\alpha)\mathbf{\Phi}$. Actually, $\mathbf{R}^{(t)}$ will converge to $\mathbf{R}^*$ as t grows to infinity [42, 62, 63]:

$$\mathbf{R}^* = (1-\alpha)(\mathbf{I} - \alpha\mathbf{P})^{-1}\mathbf{\Phi} \qquad (20)$$

Here, $\alpha \in (0, 1)$ is the trade-off parameter balancing the information flow from label propagation and initial labeling. After

that, the implicit relative labeling-importance information $\mathcal{U}$ is obtained by normalizing $\mathbf{R}^*$ by each row:

$$\forall 1 \le i \le m,\ \forall 1 \le j \le q:\ \mu_{\boldsymbol{x}_i}^{\lambda_j} = \frac{r_{ij}^*}{\sum_{j=1}^{q} r_{ij}^*} \tag{21}$$

In the second stage, in order to make use of the information conveyed by $\mathcal{U}$, RELIAB chooses the maximum entropy model [64] to parametrize the multi-label predictor:

$$f_j(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\boldsymbol{\theta}_j^\top \boldsymbol{x}\right) \quad (1 \le j \le q)$$
$$\text{where } Z(\boldsymbol{x}) = \sum\nolimits_{j=1}^{q} \exp\left(\boldsymbol{\theta}_j^\top \boldsymbol{x}\right) \tag{22}$$

To induce the prediction model $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_q]$, RELIAB chooses to minimize the following objective function:

$$V(\boldsymbol{\Theta}, \mathcal{U}, \mathcal{D}) = V_{dis}(\boldsymbol{\Theta}, \mathcal{U}) + \beta \cdot V_{emp}(\boldsymbol{\Theta}, \mathcal{D}) \tag{23}$$

Here, the first term $V_{dis}(\boldsymbol{\Theta}, \mathcal{U})$ evaluates how well the prediction model $\boldsymbol{\Theta}$ fits with the estimated relative labeling-importance information $\mathcal{U}$ (e.g. by Kullback-Leibler divergence), the second term evaluates how well the prediction model $\boldsymbol{\Theta}$ classifies the training examples in $\mathcal{D}$ (e.g. by empirical ranking loss). Furthermore, β is the regularization parameter balancing the two terms of objective function.

Given an unseen instance $\boldsymbol{x}^*$, its relevant label set Y^* is determined by thresholding the parametrized prediction model:

$$Y^* = \{\lambda_j \mid f_j(\boldsymbol{x}^*) > t(\boldsymbol{x}^*),\ 1 \le j \le q\} \tag{24}$$

Here, $t(\boldsymbol{x}^*)$ is the thresholding function which can be learned from the training examples as well [1, 34, 42].

Table 6 summarizes the pseudo-code of RELIAB. As shown in Table 6, RELIAB employs a two-stage procedure to learn from multi-label examples, where the relative labeling-importance information estimated in the first stage contribute to model induction in the second stage. Furthermore, the order of label correlations considered by RELIAB depends on the empirical loss chosen to instantiate $V_{emp}(\boldsymbol{\Theta}, \mathcal{D})$.

Table 6 The pseudo-code of RELIAB [42].

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}^i, \boldsymbol{y}^i) \mid 1 \le i \le m\}$ ($\boldsymbol{x}^i \in \mathcal{X}, \boldsymbol{y}^i \in \{-1, +1\}^q, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$)

α: the trade-off parameter in (0,1)

β: the regularization parameter

$\boldsymbol{x}^*$: the unseen instance ($\boldsymbol{x}^* \in \mathcal{X}$)

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$ ($Y^* \subseteq \mathcal{Y}$)

Process:

1: Construct the fully-connected graph $G = (V, E)$ with $V = \{\boldsymbol{x}^i \mid 1 \le i \le m\}$;
2: Specify the weight matrix $\mathbf{W}$ according to Eq.(17);
3: Set the label propagation matrix $\mathbf{P}$ according to Eq.(18);
4: Initialize the labeling-importance matrix $\mathbf{R}$ according to Eq.(19), and then yield the converged solution $\mathbf{R}^*$ according to Eq.(20);
5: Obtain the relative labeling-importance information $\mathcal{U}$ according to Eq.(21);
6: Learn the parametrized prediction model $\boldsymbol{\Theta}$ by minimizing the objective function specified in Eq.(23);
7: Return $Y^* = \{\lambda_j \mid f_j(\boldsymbol{x}^*) > t(\boldsymbol{x}^*),\ 1 \le j \le q\}$ w.r.t. Eq.(24)

5 Conclusion

In this paper, the state-of-the-art of binary relevance, which is one of the most important solutions to multi-label learning, is reviewed. Particularly, the basic setting of binary relevance, some representative correlation-enabling extensions, and related issues on class-imbalance and relative labeling-importance have been discussed. Code packages for the learning algorithms introduced in this paper are also publicly-available.[4)]

For binary relevance, there are several research issues which are worth further investigation. Firstly, performance evaluation in multi-label learning is more complicated than single-label learning where a number of popular multi-label evaluation metrics have been proposed [1, 2, 10, 11]. It is desirable to design correlation-enabling extensions to binary relevance which are tailored to optimize designated multi-label metric suitable for the multi-label learning task at hand. Secondly, in binary relevance the same set of features are used to induce the classification models for all class labels. It is appropriate to develop binary relevance style learning algorithms which are capable of utilizing label-specific features to characterize distinct properties of each class label [65–67]. Thirdly, the modeling complexities of binary relevance as well as its extensions are linear to the number of class labels in the label space. It is necessary to adapt binary relevance to accommodate the extreme multi-label learning scenario with huge (e.g. millions) number of class

[4)] `http://mulan.sourceforge.net/` (binary relevance [9], classifier chain [12, 14], stacking aggregation [19])
`http://cse.seu.edu.cn/PersonalPage/zhangml/Resources.htm` (LEAD [25], COCOA [38], RELIAB [42])

labels [68–72].

References

1. M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.
2. Z.-H. Zhou and M.-L. Zhang. Multi-label learning. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1–8. Springer, Berlin, 2016.
3. R. E. Schapire and Y. Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
4. R. S. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Matrix completion for multi-label image classification. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 190–198. MIT Press, Cambridge, MA, 2011.
5. C. Sanden and J. Z. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 705–714, Beijing, China, 2011.
6. Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
7. G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*, pages 17–26, Augsburg, Germany, 2007.
8. L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the 19th International Conference on World Wide Web*, pages 211–220, Madrid, Spain, 2009.
9. M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.
10. G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–686. Springer, Berlin, 2010.
11. E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):Article 52, 2015.
12. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In W. Buntine, M. Grobelnik, and J. Shawe-Taylor, editors, *Lecture Notes in Artificial Intelligence 5782*, pages 254–269. Springer, Berlin, 2009.
13. K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning*, pages 279–286, Haifa, Israel, 2010.
14. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.
15. A. Kumar, S. Vembu, A. K. Menon, and C. Elkan. Learning and inference in probabilistic classifier chains with beam search. In P. A. Flach, T. De Bie, and N. Cristianini, editors, *Lecture Notes in Computer Science 7523*, pages 665–680. Springer, Berlin, 2012.
16. N. Li and Z.-H. Zhou. Selective ensemble of classifier chains. In Z.-H. Zhou, F. Roli, and J. Kittler, editors, *Lecture Notes in Computer Science 7872*, pages 146–156. Springer, Berlin, 2013.
17. R. Senge, J. J. del Coz, and E. Hüllermeier. Rectifying classifier chains for multi-label classification. In *Proceedings of the 15th German Workshop on Learning, Knowledge, and Adaptation*, pages 162–169, Bamberg, Germany, 2013.
18. D. Mena, E. Montañés, J. R. Quevedo, and J. J. del Coz. A family of admissible heuristics for A* to perform inference in probabilistic classifier chains. *Machine Learning*, in press.
19. S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In H. Dai, R. Srikant, and C. Zhang, editors, *Lecture Notes in Artificial Intelligence 3056*, pages 22–30. Springer, Berlin, 2004.
20. E. Montañés, J. R. Quevedo, and J. J. del Coz. Aggregating independent and dependent models to learn multi-label classifiers. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Lecture Notes in Artificial Intelligence 6912*, pages 484–500. Springer, Berlin, 2011.
21. E. Montañés, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier. Dependent binary relevance models for multi-label classification. *Pattern Recognition*, 47(3):1494–1508, 2014.
22. M. A. Tahir, J. Kittler, and A. Bouridane. Multi-label classification using stacked spectral kernel discriminant analysis. *Neurocomputing*, 171(C):127–137, 2016.
23. E. Loza Mencía and F. Janssen. Learning rules for multi-label classification: A stacking and a separate-and-conquer approach. *Machine Learning*, 105(1):77–126, 2016.
24. G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Working Notes of the First International Workshop on Learning from Multi-Label Data*, pages 101–116, Bled, Slovenia, 2009.
25. M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 999–1007, Washington D. C., 2010.
26. A. Alessandro, G. Corani, D. Mauá, and S. Gabaglio. An ensemble of Bayesian networks for multilabel classification. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1220–1225, Beijing, China, 2013.
27. L. E. Sucar, C. Bielza, E. F. Morales, P. Hernandez-Leal, J. H. Zaragoza, and P. Larrañaga. Multi-label classification with bayesian network-based chain classifiers. *Pattern Recognition Letters*, 41:14–22, 2014.
28. Y.-K. Li and M.-L. Zhang. Enhancing binary relevance for multi-label learning with controlled label correlations exploitation. In D.-N. Pham and S.-B. Park, editors, *Lecture Notes in Artificial Intelligence 8862*, pages 91–103. Springer, Berlin, 2014.
29. A. Alali and M. Kubat. Prudent: A pruned and confident stacking ap-

proach for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2480–2493, 2015.

30. J. Petterson and T. Caetano. Reverse multi-label learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1912–1920. MIT Press, Cambridge, MA, 2010.
31. E. Spyromitros-Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1583–1588, Barcelona, Spain, 2011.
32. M. A. Tahir, J. Kittler, and F. Yan. Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition*, 45(10):3738–3750, 2012.
33. J. R. Quevedo, O. Luaces, and A. Bahamonde. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883, 2012.
34. I. Pillai, G. Fumera, and F. Roli. Threshold optimisation for multi-label classifiers. *Pattern Recognition*, 46(7):2055–2065, 2013.
35. K. Dembczyński, A. Jachnik, W. Kotłowski, W. Waegeman, and E. Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1130–1138, Atlanta, GA, 2013.
36. F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163:3–16, 2015.
37. F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera. Mlsmote: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, 89:385–397, 2015.
38. M.-L. Zhang, Y.-K. Li, and X.-Y. Liu. Towards class-imbalance aware multi-label learning. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 4041–4047, Buenos Aires, Argentina, 2015.
39. B. Wu, S. Lyu, and B. Ghanem. Constrained submodular minimization for missing labels and class imbalance in multi-label learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2229–2236, Phoenix, AZ, 2016.
40. W. Cheng, K. Dembczyński, and E. Hüllermeier. Graded multilabel classification: The ordinal case. In *Proceedings of the 27th International Conference on Machine Learning*, pages 223–230, Haifa, Israel, 2010.
41. M. Xu, Y.-F. Li, and Z.-H. Zhou. Multi-label learning with PRO loss. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, pages 998–1004, Bellevue, WA, 2013.
42. Y.-K. Li, M.-L. Zhang, and X. Geng. Leveraging implicit relative labeling-importance information for effective multi-label learning. In *Proceedings of the 15th IEEE International Conference on Data Mining*, pages 251–260, Atlantic City, NJ, 2015.
43. X. Geng, C. Yin, and Z.-H. Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.
44. X. Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.
45. N. Gao, S.-J. Huang, and S. Chen. Multi-label active learning by model guided distribution matching. *Frontiers of Computer Science*, 10(5):845–855, 2016.
46. K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.
47. W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44, 2013.
48. Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou. Multi-label learning with weak label. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 593–598, Atlanta, GA, 2010.
49. M. Xu, R. Jin, and Z.-H. Zhou. Speedup matrix completion with side information: Application to multi-label learning. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2301–2309. MIT Press, Cambridge, MA, 2013.
50. R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Matrix completion for weakly-supervised multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):121–135, 2015.
51. R. Senge, J. J. del Coz, and E. Hüllermeier. On the problem of error propagation in classifier chains for multi-label classification. In M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, editors, *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 163–170. Springer, Berlin, 2014.
52. Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, Boca Raton, FL, 2012.
53. D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.
54. M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 241–248, Menlo Park, CA, 2006.
55. V. Smith, J. Yu, T. Smulders, A. Hartemink, and E. Jarvis. Computational inference of neural information flow networks. *PLoS Computational Biology*, 2:1436–1449, 2006.
56. K. Murphy. Software for graphical models: A review. *ISBA Bulletin*, 14(4):13–15, 2007.
57. G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. MULAN: A java library for multi-label learning. *Journal of Machine Learning Research*, 12(Jul):2411–2414, 2011.
58. H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
59. S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 42(4):1119–1130, 2012.
60. X.-Y. Liu, Q.-Q. Li, and Z.-H. Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In *Proceedings of the 13th IEEE International Conference on Data Mining*, pages 478–487, Dallas, TX, 2013.

61. L. Abdi and S. Hashemi. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):238–251, 2016.
62. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 284–291. MIT Press, Cambridge, MA, 2004.
63. X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. In R. J. Brachman and T. G. Dietterich, editors, *Synthesis Lectures to Artificial Intelligence and Machine Learning*, pages 1–130. Morgan & Claypool Publishers, San Francisco, CA, 2009.
64. S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.
65. M.-L. Zhang and L. Wu. LIFT: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):107–120, 2015.
66. S. Xu, X. Yang, H. Yu, D.-J. Yu, J. Yang, and E. C. C. Tsang. Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems*, 104:52–61, 2016.
67. J. Huang, G. Li, Q. Huang, and X. Wu. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3309–3323, 2016.
68. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2764–2770, Barcelona, Spain, 2011.
69. R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 13–24, Rio de Janeiro, Brazil, 2013.
70. C. Xu, D. Tao, and C. Xu. Robust extreme multi-label learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1275–1284, San Francisco, CA, 2016.
71. H. Jain, Y. Prabhu, and M. Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944, San Francisco, CA, 2016.
72. W.-J. Zhou, Y. Yu, and M.-L. Zhang. Binary linear compression for multi-label classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.