Noname manuscript No.
(will be inserted by the editor)

Multi-Label Classification by Polytree-Augmented Classifier Chains with Label-Dependent Features

Blinded Manuscript

Received: date / Accepted: date

Abstract Multi-label classification faces with several critical challenges, such as modeling label correlations, mitigating label imbalance, removing irrelevant and redundant features, and reducing the complexity for large-scale problems. To address these problems, in this paper, we propose a novel method, polytree-augmented classifier chains with label-dependent features, which models label correlations through flexible polytree structures based on low-dimensional label-dependent feature spaces learned by a two-stage feature selection approach. First a feature weighting approach is applied to efficiently remove irrelevant features for each label and mitigate the effect of label imbalance. Second, a polytree structure is built in the label space using estimated conditional mutual information. Third, an appropriate label-dependent feature subset is found by taking account of label correlations in the polytree. Extensive empirical studies on six synthetic datasets and twelve real-world datasets demonstrate the superior performance of the proposed method. In addition, by incorporating the proposed two-stage feature selection approach, the multi-label classifiers with label-dependent features achieve 9.4% performance improvement in Exact-Match on average compared with the original classifiers.

Keywords Multi-label classification · Label correlation · polytree-augmented classifier chain · Label-dependent feature · Label imbalance

1 Introduction

In recent years we have witnessed the increasing demand of multi-label classification (MLC) in a wide range of applications, such as text categorization, semantic image annotation, bioinformatics analysis and audio emotion detection, for which numerous machine learning techniques have been specifically designed and successfully utilized. Unlike traditional multi-class single-label classification, where each instance is associated with only a single label, the task of MLC is to assign a label subset to an unseen instance. The existing MLC methods fall into two broad categories: problem transformation and algorithm adaptation [35]. Problem transformation strategy typically transforms an MLC problem into a set of single-label classification problems, and learns a family of classifiers for modeling the single-label memberships. Algorithm adaptation strategy induces conventional machine learning algorithms in the multi-label settings. A number of MLC methods adopting one of the above two strategies have been developed and succeeded in dealing with various multi-label problems.

The previous efforts on MLC focus mainly on two aspects: label correlation modeling and dimensionality reduction. Many researches [28, 11] have shown that capturing label correlations is crucial for a MLC method to achieve competitive classification performance. On the other hand, a variety of dimension reduction approaches [22, 48, 44] have been proposed for the multi-label problems in order to reduce the resource consumption and improve performance. However, most of these methods build their models on the basis of an identical feature space for all labels. Such a universal hypothesis possibly introduces irrelevant and redundant features, resulting in two problems: decreasing the model's generalization ability and increasing its computational complexity for both learning and prediction. Rather, it is natural to think that each label holds its own specific set of features to distinguish from other labels. For example, in image annotation, an object typically relates to only a few regions in the high-dimensional feature space, and in text categorization, one specific topic is probably relevant to a fraction of words from the massive amounts of vocabulary.

For blinded manuscript

Hence, in this study we presume that modeling label correlations and mining label-dependent features would benefit the generalization ability and prediction accuracy of a MLC method. As in our previous work [32], the basic idea of Polytree-Augmented Classifier Chains (PACC) has already been proposed, which is more flexible on modeling label correlations than conventional MLC methods. In this paper, we improve the PACC by selecting Label-Dependent Features to produce the PACC-LDF method. We employ a hybrid two-stage feature selection algorithm for the polytree structure. Specifically, a information gain-based feature weighting algorithm is employed in the first stage to efficiently remove irrelevant features in each label, and alleviate the label imbalance problem; After construction of a polytree, in the second stage, a correlation-based feature subset selection algorithm is carried out to select label-dependent feature subset by incorporating label correlations modeled by the polytree. In this way, label-dependent features chosen for the polytree structure will be used to learn the classifier chain and to make prediction on a test instance. The proposed two-stage feature selection algorithm is also applicable to the other MLC methods, such as Classifier Chains (CC) based methods [28, 7, 43]. Extensive experiments conducted on both synthetic and real-world datasets demonstrate the performance superiority of the proposed PACC-LDF method compared with several state-of-the-art MLC methods in terms of classification performance and time complexity.

The contributions of this work are cast into three-folds.

- The polytree structure is introduced to model label dependency, according to which, we propose PACC for MLC and present more technical details in this paper than our previous work in [32].
- A two-stage feature selection framework is specifically developed for PACC to select Label-Dependent Features (LDF), which enables to mitigate the label imbalance problem and save label correlations modeled in the built polytree structure.
- Empirical studies show that on average MLC methods can be improved 9.4% in Exact-Match by incorporating LDF. In addition, extensive experimental results demonstrate the efficiency of the proposed PACC-LDF compared with popular MLC methods.

The remainder of this paper is organized as follows. Section 2 gives the mathematical definition of MLC. Section 3 discusses the related works. Section 4 states the challenges confronted with MLC methods. Section 5 illustrates an overview of the system framework, and presents technical details in two parts: Polytree-Augmented Classifier Chains (PACC) and Label-Dependent Feature (LDF) selection. Section 6 presents the statistical properties of benchmark multi-label datasets, and defines four metrics for evaluating multi-label classifiers. The experimental results are reported and discussed in Section 7. Finally, Section 8 concludes this paper and discusses future work.

2 Multi-label classification

In the scenario of MLC, given a finite set of labels $\mathcal{L} = \{\lambda_1, ..., \lambda_L\}$, an instance is typically represented by a pair $(\mathbf{x}, \mathbf{y})$, which contains a feature vector $\mathbf{x} = (x_1, ..., x_M)$ as a realization of the random vector $\mathbf{X} = (X_1, ..., X_M)$ drawn from the input feature space $\mathcal{X} = \mathbb{R}^M$, and the corresponding label vector $\mathbf{y} = (y_1, ..., y_L)$ drawn from the output label space $\mathcal{Y} = \{0, 1\}^L$. In other words, $\mathbf{y} = (y_1, ..., y_L)$ can be viewed as a realization of corresponding random vector $\mathbf{Y} = (Y_1, ..., Y_L)$, $\mathbf{Y} \in \mathcal{Y}$, where $y_j = 1$ if label λ_j is associated with the corresponding instance $\mathbf{x}$, and $y_j = 0$ otherwise.

Suppose that we are given a dataset of N instances $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$, where $\mathbf{y}^{(i)}$ is the label assignment of the ith instance. The task of MLC is to find an optimal classifier $h : \mathcal{X} \to \mathcal{Y}$ which assigns an appropriate label vector $\mathbf{y}$ to each instance $\mathbf{x}$ such that h minimizes a loss function. Given a loss function $loss(\mathbf{Y}, h(\mathbf{X}))$, the optimal h^* is

$$h^* = \arg\min_h \mathbb{E}_{P(\mathbf{x},\mathbf{y})} loss(\mathbf{Y}, h(\mathbf{X})), \tag{1}$$

where $P(\mathbf{x}, \mathbf{y})$ is the joint probability distribution over the feature vector $\mathbf{x}$ and label vector $\mathbf{y}$. The optimal classifier (1) can be rewritten in a pointwise way,

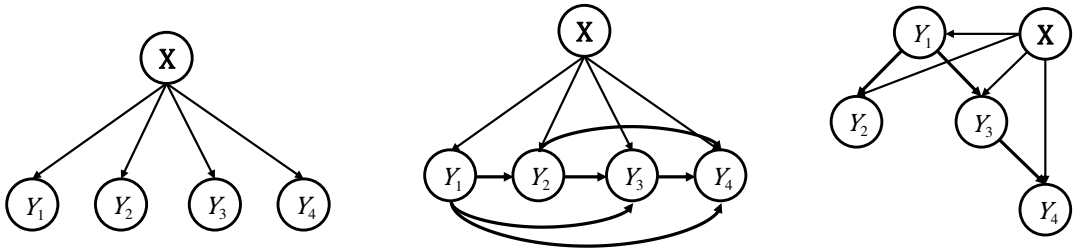
$$\hat{\mathbf{y}} = h^*(\mathbf{x}) = \arg\min_h \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}) loss(\mathbf{y}, h(\mathbf{x})). \tag{2}$$

For the subset 0-1 loss $loss_S(\mathbf{y}, \hat{\mathbf{y}}) = \mathbb{1}_{\mathbf{y} \neq \hat{\mathbf{y}}}$, where $\mathbb{1}_{(\cdot)}$ is the indicator function, (2) becomes

$$\hat{\mathbf{y}} = h^*(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}). \tag{3}$$

Similarly, for the hamming loss $loss_H(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{L} \sum_{j=1}^L \mathbb{1}_{y_j \neq \hat{y}_j}$, we have

$$\hat{y}_j = h_j^*(\mathbf{x}) = \arg\max_{y_j \in \{0,1\}} P(y_j|\mathbf{x}), \quad j = 1, ..., L. \tag{4}$$



(a) BR (b) CC and PCC (c) BCC

Fig. 1: Probabilistic graphical models of CC-based methods for a MLC problem with four label variables.

As proved in [8], (4) coincides with (3) in case of conditional independence of labels. In Section 3, we will show that binary relevance method [3] and several classifier chain based methods [7, 28, 43, 29] are actually indirect approximations of (3).

3 Related works

In recent years, many efforts in MLC have been paid on two aspects: label correlation modeling and dimensionality reduction. It has been shown in a number of researches [28, 11] that modeling label correlations is very crucial to perform accurate classification. On the other hand, various dimension reduction algorithms, including feature selection [22, 13] and feature extraction [48, 44], have been employed in MLC, in order to simplify the learning phase and overcome the curse of dimensionality.

In terms of label correlation modeling, Classifier chains (CC) based methods have been proposed at a tractable time complexity, originating from the simple Binary Relevance (BR) method. In the BR context, a classifier h is comprised of L binary classifiers $h_1, ..., h_L$, where each member classifier h_j predicts $\hat{y}_j \in \{0, 1\}$, forming a vector $\hat{\mathbf{y}} \in \{0, 1\}^L$. In the prediction phase, BR collects the result of the member classifiers, i.e., $\hat{y}_j \leftarrow h_j(\mathbf{x})$, which is identical with (4). In this sense, BR can be seen as a hamming loss risk minimizer [8]. In CC [28], the label correlation is expressed in an ordered chain. In the learning phase, according to a predefined chain order, like $Y_1 \rightarrow Y_2 \rightarrow \cdots \rightarrow Y_L$, it builds L binary classifiers $h_1, h_2, ..., h_L$ such that each classifier predicts the correct value of y_j by referring to the correct values of $\mathbf{pa}(y_j) = \{y_1, y_2, ..., y_{j-1}\}$ in addition to $\mathbf{x}$. In the prediction phase, it predicts in turn the value of y_j using the previously estimated values $\hat{y}_1, ..., \hat{y}_{j-1}$ with $\mathbf{x}$ according to:

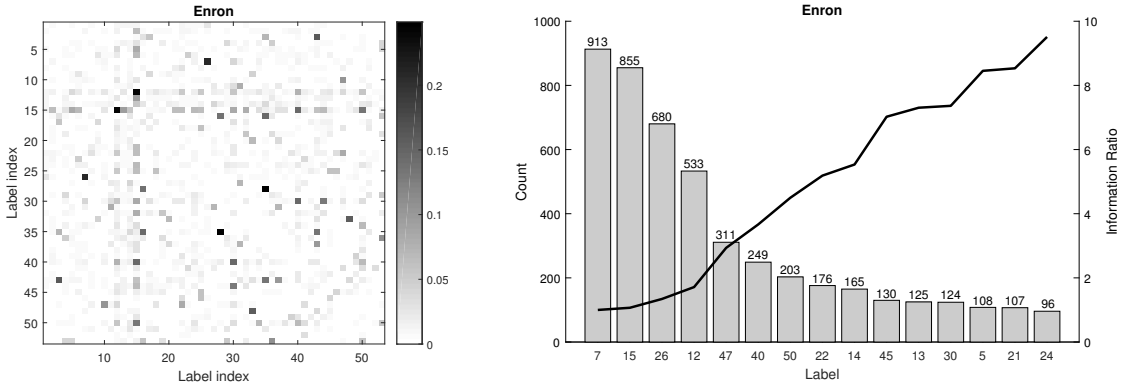
$$\hat{y}_j = \underset{y_j \in \{0,1\}}{\arg\max}\, P(y_j | \hat{\mathbf{pa}}(y_j), \mathbf{x}), \quad j = 1, ..., L. \tag{5}$$

Finally we have $\hat{\mathbf{y}} = (\hat{y_1}, \hat{y_2}, ..., \hat{y_L})$. Note that CC predicts the presence/absence of a label depending on previously predicted label set and its prediction is made in only one path. Probabilistic Classifier Chains (PCC) [7] provides better estimates than CC at the expense of a higher time complexity in the prediction phase. Although PCC shares the learning model (8) with CC, it chooses the best predictor by examining all the 2^L paths in an exhaustive manner according to the following:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\arg\max} \prod_{j=1}^{L} P(y_j | \hat{\mathbf{pa}}(y_j), \mathbf{x}). \tag{6}$$

The exponential cost of PCC in prediction limits its application. To make the prediction tractable for PCC, several methods [21, 9, 30] have been proposed to to find the approximate MAP *maximum a posterior* (MAP) assignment of labels to a test instance. Bayesian Classifier Chains (BCC) [43] introduces a directed tree as the probabilistic structure over labels. The directed tree is established by randomly choosing a label as its root and by assigning directions to the remaining edges. It shares the same model (8) and (5) with CC, but $|\mathbf{pa}(Y_j)| \leq 1$ limits its expression ability on label correlations. Fig. 1 shows an example of the graphical models of BR, CC, PCC and BCC with four labels. In terms of time complexity, all these methods hold linear complexity $O(LMN)$ for training if a linear baseline classifier is utilized. In the prediction phase, BR, CC and BCC have linear complexity $O(LM)$ for testing a single instance, while PCC needs a time complexity of $O(2^L LM)$.

On the other hand, a variety of MLC methods have been proposed to reduce the dimensionality of multi-label problems. The Multi-Label Naive Bayes (MLNB) method [47] incorporates feature selection mechanism into a new-designed naive Bayes classifier. Principal component analysis is employed to



(a) Label correlation (b) Label imbalance

Fig. 2: Label correlation and label imbalance in the Enron dataset ($L = 53$). (a) Visualization of label correlations; (b) The label imbalance problem, where 15 most frequent labels are reported.

remove unnecessary features, and then a wrapper approach with genetic algorithm is performed. However, MLNB is applicable to regular-scale datasets with continuous features due to its feature selection mechanism. Label specIfic FeaTures (LIFT) [44] extracts label-specific features by conducting k-means clustering analysis on the positive and negative instances according to a specific label. It obtained competitive results on a broad range of benchmark multi-label datasets, but spent more prediction time than the MLC methods with linear time complexity. By learning a Hierarchy Of Multi-label classifiER (HOMER) based on a balanced k-means clustering approach, HOMER [36] partitioned the whole label set into a series of smaller and more balanced sets following the layers of the label hierarchy. Label Partition for Sublinear Ranking (LPSR) [39] consists of two-stages: feature space partition and label assignment. It reduces the prediction complexity by learning a hierarchy over base classifiers, but it has a higher training cost than the linear complexity in L. To cope with large-scale problems, a tree-based multi-label method, FastXML, is proposed in [27]. Based on a novel ranking loss function, nDCG, it developed an efficient alternating minimization algorithm to optimize the objective function. In this way, it achieved competitive classification accuracy compared with other scalable MLC methods, and could scale to large-scale datasets even with a million of labels.

For the efficiency, in this paper, we incorporate a two-stage label-dependent feature selection mechanism into the learning phase of a novel polytree-augmented classifier chains method [32] in order to improve its performance on the basis of label-dependent features.

4 Challenges for MLC

4.1 Label correlations

There are two types of label correlations for MLC, *marginal* and *conditional* dependency. According to the structure of a Bayesian network for $P(\mathbf{X}, \mathbf{Y})$, we have the marginal distribution and the conditional distribution given as,

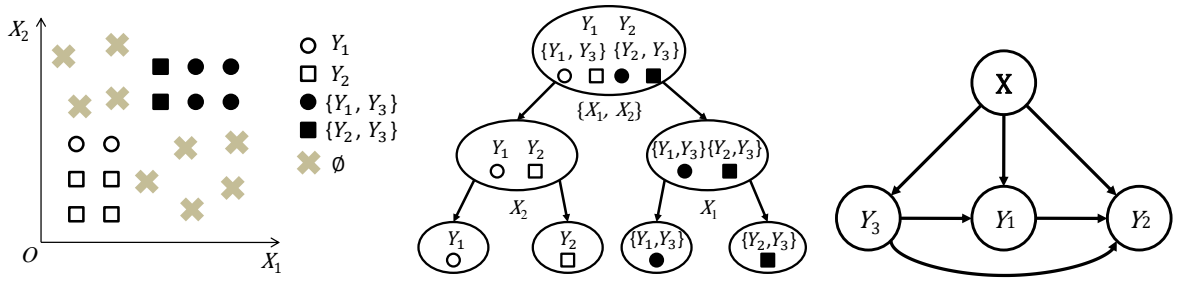
$$P(\mathbf{Y}) = \prod_{j=1}^{L} P(Y_j|\mathbf{pa}(Y_j)), \tag{7}$$

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{L} P(Y_j|\mathbf{pa}(Y_j), \mathbf{X}), \tag{8}$$

where $\mathbf{pa}(Y_j)$ denotes the parent label set of Y_j. Then the definition on label dependence can be induced:

Definition 1 *Label random vector* $\mathbf{Y}$ *is called marginally or conditionally independent if* $\forall Y_j : \mathbf{pa}(Y_j) = \emptyset$ *in (7) or (8).*

We can see that label-pair correlation is prevalent in many datasets from the values of mutual information $I(Y_j; Y_k)$ for any pair of Y_j and Y_k. Fig. 2(a) shows the label correlations in the Enron dataset, which is measured by mutual information.



(a) A sample dataset (b) Classification strategy (c) Graphical model

Fig. 3: Irrelevant and redundant features in a multi-label setting. (a) shows the label distribution over the 2D feature space; (b) shows the classification partition strategy for (a); (c) gives the probability graphical model for (a).

4.2 Label-dependent irrelevant and redundant features

The existence of irrelevant and redundant features for classification increases the computational complexity in both learning and prediction, and moreover often reduces the generalization ability of the classifiers designed from instances due to the curse of dimensionality. Irrelevant and redundant features are two distinct concepts, an irrelevant feature has no discriminative information, while a redundant feature shares the same discriminative information with other features. Removal of these features, therefore, does not lose the discriminative information. Rather, elimination of irrelevant and redundant features simplifies the learning phase and prevents from overfitting.

An MLC example with irrelevant and redundant features is shown in Fig. 3. For label Y_3, features X_1 and X_2 are redundant as Y_3 could be readily classified by either X_1 or X_2. If we have prior information of the presence of Y_3, X_2 is irrelevant since Y_1 can be discriminated from Y_2 based only on X_1. This example shows how irrelevant and redundant features can exist in MLC, and how label-dependent features can be exploited to faciliate the process of classification. Moreover, it shows that the predicted label value can provide some useful information for the unpredicted labels, which could further compact label-dependent feature subsets and promote classification. For example, given the absence of Y_3, only X_2 is the discriminative feature for labels Y_1 and Y_2.

4.3 Label imbalance

Multi-label datasets are typically imbalanced, i.e., the number of instances associated with each label is often unequal. In other words, the ratio of positive instances against the negative ones may be quite low for some labels. The imbalance problem usually harms the performance of the learned classifier from two points of view. On one hand, if we aim to minimize hamming loss or ranking loss, we tend to ignore minority labels. On the other hand, classifiers for minority labels are difficult to design. Such label imbalance problem becomes more serious in multi-label datasets than in single-label datasets, since a label comprising of more classes typically has less samples. To depict the imbalance level of a dataset $\mathcal{D}$, the mean of the Imbalance Ratio (IR) and the Coefficient of Variation of IR (CVIR) [5] are introduced as follows,

$$\overline{\mathrm{IR}} = \frac{1}{L}\sum_{j=1}^{L}\mathrm{IR}(Y_j), \quad \mathrm{CVIR} = \frac{1}{\overline{\mathrm{IR}}}\sqrt{\sum_{j=1}^{L}\frac{(\mathrm{IR}(Y_j)-\overline{\mathrm{IR}})^2}{L-1}}, \tag{9}$$

where $\mathrm{IR}(Y_j) = \max_k \sum_{i=1}^{N} \mathbb{1}_{y_k^{(i)}=1} / \sum_{i=1}^{N} \mathbb{1}_{y_j^{(i)}=1}$. We can see that multi-label datasets are highly imbalanced from the values of these two measures (Table 2 of Section 6.1). Fig. 2(b) shows the label imbalance problem in the Enron dataset ($L = 53$), where only 15 most frequent labels are reported. A simple way to mitigate the imbalance level in correlation measurement is to use the normalized mutual information (defined in (25) of Sec. 7.1), instead of the original mutual information, $I(Y_j; Y_k)$. Another way is to perform undersampling for majority labels and oversampling for minority labels. For more information on the imbalance problems of MLC, see [5].

Table 1: Summary of popular MLC methods from four viewpoints (○: good, △: moderate, ×: bad).

Author	Method	Issues to be considered			
		Label Correlation	Dimension Reduction	Label Imbalance	Time Complexity
J. Read et al. [28]	CC	○	×	×	×
K. Dembczynski et al. [7]	PCC	○	×	×	×
M. Zhang et al. [46]	MLkNN	○	×	×	△
G. Tsoumakas et al. [37]	RAkEL	○	×	△	×
G. Tsoumakas et al. [36]	HOMER	○	○*	△	△
M. Zhang et al. [47]	MLNB	△	○†	×	×
J. Weston et al. [39]	LPSR	○	○‡	△	△
M. Zhang et al. [44]	LIFT	×	○†	△	△
Y. Prabhu et al. [27]	FastXML	○	○‡	△	○
This paper	PACC-LDF	○	○†	△	△

* in the label space
† in the feature space
‡ in both the feature and label spaces

4.4 High complexity for large-scale data

The real applications of MLC often confront with large-scale problems, where either of the number of labels L, attributes M and instances N might be very large. In such a case, the time complexity will become an important aspect for evaluating an MLC algorithm, sometimes more important than classification accuracy for real-world applications.

Up to now, one of the simplest MLC methods is to transform the MLC problem into a series of single-label classification problems, namely Binary Relevance (BR), which has a linear time complexity $O(LMN)$ in terms of the complexity of the baseline classifier. However, even such a linear complexity can be intractable for large-scale MLC problems. To overcome the limitation, as mentioned in Sec. 3, several dimension reduction approaches are applied in MLC methods so as to attain a sublinear complexity. For instance, embedding based methods [38, 41, 2] project the label vectors onto a low-dimensional linear or nonlinear label subspace, leading to a time complexity of $O(\hat{L}MN)$ with $\hat{L} \ll L$. In this paper, we focus on reducing the dimensionality of the feature space, to attain a complexity of $O(L\hat{M}N)$ with $\hat{M} \ll M$.

Table 1 summarizes several MLC methods evaluated from the four viewpoints discussed in this section. It shows that most of popular MLC methods enable to cope with only some of the four issues confronted with MLC. In this paper, we try to deal with all the four aspects with the proposed method.

5 Polytree-augmented classifier chains with label-dependent features

5.1 Polytree-augmented classifier chains

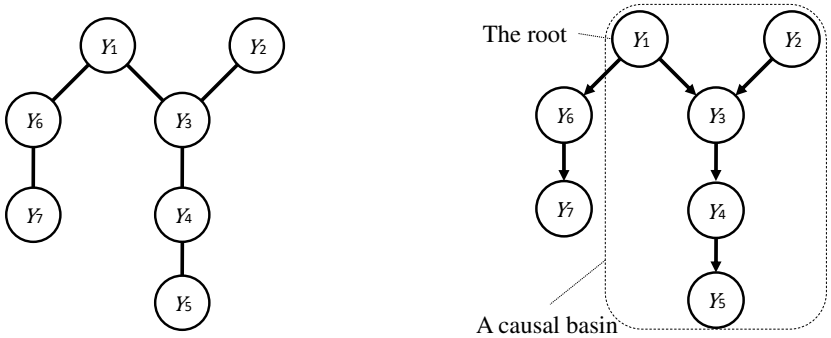
We propose a novel polytree-augmented classifier chains (PACC) as a compromise between the expression ability and the efficiency. A *polytree* (Fig. 4) is a directed acyclic graph whose underlying undirected graph is a tree but a node can have multiple parents [31]. That is, it is more flexible than trees. A *causal basin*, as shown in Fig. 4(b), is a subgraph which starts with a multi-parent node and continues following a causal flow to include all the descendants and their direct parents.

5.1.1 Structure learning

In PACC, the conditional label dependence is obtained by approximating the true distribution $P(\mathbf{Y}|\mathbf{X})$ by another distribution. According to Chou-liu's proof [6] and our previous work [32], we can have its feature-conditioned version.

Theorem 1 *To approximate a conditional distribution $P(\mathbf{Y}|\mathbf{X})$, the optimal Bayesian network B^* in K-L divergence is obtained if the sum of conditional mutual information between each variable of $\mathbf{Y}$ and its parent variables given the observation $\mathbf{X}$ is maximized.*

Proof Here we use Kullback-Leibler (KL) divergence [20], $D_{KL}(P||P_B)$, a quasi distance between two distributions, to evaluate how close an alternative distribution $P_B(\mathbf{Y}|\mathbf{X})$ is to $P(\mathbf{Y}|\mathbf{X})$, where B denotes



(a) A tree skeleton (b) A polytree

Fig. 4: A polytree with a causal basin.

a Bayesian network:

$$\begin{aligned} B^* &= \arg\min_B D_{KL}(P(\mathbf{Y}|\mathbf{X})||P_B(\mathbf{Y}|\mathbf{X})) \\ &= \arg\min_B \mathbb{E}_{P(\mathbf{x},\mathbf{y})} \frac{\log P(\mathbf{Y}|\mathbf{X})}{\log P_B(\mathbf{Y}|\mathbf{X})} \\ &= \arg\max_B \mathbb{E}_{P(\mathbf{x},\mathbf{y})} \log P_B(\mathbf{Y}|\mathbf{X}). \end{aligned} \tag{10}$$

According to the parents-children relationship in B,

$$\begin{aligned} B^* &= \arg\max_B \mathbb{E}_{P(\mathbf{x},\mathbf{y})} \log \prod_{j=1}^{L} P_B(Y_j|\mathbf{pa}(Y_j),\mathbf{X}) \\ &= \arg\max_B \sum_{j=1}^{L} \mathbb{E}_{P(\mathbf{x},y_j,\mathbf{pa}(y_j))} \log P_B(Y_j|\mathbf{pa}(Y_j),\mathbf{X}), \end{aligned}$$

which is maximized if $P_B(\cdot) = P(\cdot)$. Then, since $P(Y_j|\mathbf{X})$ is independent of the parents of Y_j,

$$\begin{aligned} B^* &= \arg\max_{B=\{\mathbf{pa}(Y_j)\}} \sum_{j=1}^{L} \mathbb{E}_{P(\mathbf{x},y_j,\mathbf{pa}(y_j))} \log \frac{P(Y_j,\mathbf{pa}(Y_j)|\mathbf{X})}{P(Y_j|\mathbf{X})P(\mathbf{pa}(Y_j)|\mathbf{X})} \\ &= \arg\max_{B=\{\mathbf{pa}(Y_j)\}} \sum_{j=1}^{L} I_P(Y_j;\mathbf{pa}(Y_j)|\mathbf{X}), \end{aligned} \tag{11}$$

where $I_P(Y_i;\mathbf{pa}(Y_j)|\mathbf{X})$ represents the conditional mutual information between Y_j and its parents $\mathbf{pa}(Y_j)$ given $\mathbf{X}$ in B. As a result, the optimal B^* is obtained by maximizing $\sum_{j=1}^{L} I_P(Y_j;\mathbf{pa}(Y_j)|\mathbf{X})$.

Theorem 1 shows

$$\min_B D_{KL}(P(\mathbf{Y}|\mathbf{X})||P_B(\mathbf{Y}|\mathbf{X})) = \max_B \sum_{j=1}^{L} I_P(Y_j;\mathbf{pa}(Y_j)|\mathbf{X}). \tag{12}$$

That is, we should construct B so as to maximize the mutual information between a child and its parents. However, in practice, we do not know the true $P(\mathbf{Y}|\mathbf{X})$. Therefore we use the empirical distribution $\hat{P}(\mathbf{Y}|\mathbf{X})$ instead. Unfortunately, learning of the optimal B^* is NP-hard in general, we limit our hypothesis B to the ones satisfying $|\mathbf{pa}(Y_j)| \leq 1$ so as to $\mathbf{pa}(Y_j) = Y_k$ for some $k \in \{1, 2, ..., L\}$ or null, indicating the tree skeleton is to be built. In practice, we carry out Chou-liu's algorithm [6] to obtain the maximum-cost spanning tree (Fig. 4(a)), maximizing the weight sum, with edge weights $I_{\hat{P}}(Y_i;\mathbf{pa}(Y_j)|\mathbf{X})$.

5.1.2 Mutual information estimation

It is quite difficult to estimate conditional probability $P(\mathbf{Y}|\mathbf{X})$, when $\mathbf{X}$ is continuous. Recently some methods [8, 43, 45] have been proposed to solve this problem. In BCC [43], as an approximation of conditional probability, marginal probability of labels $\mathbf{Y}$ is obtained by simply counting the frequency

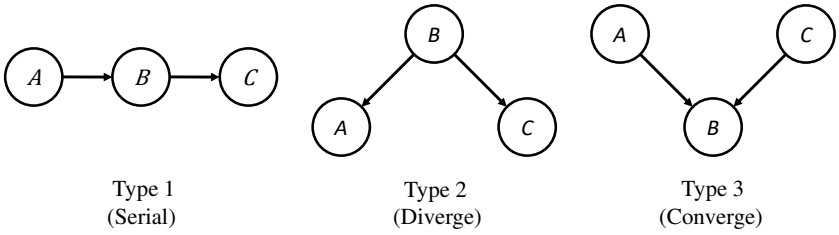


Fig. 5: Three basic types of adjacent triplets A, B, C.

of occurrence. Similar with [8], LEAD [45] directly obtains conditional dependence by estimating the degree of dependency of errors in multivariate regression models.

In [32], we used a more general approach to estimate the conditional probability. The data set $\mathcal{D}$ is splitted into two sets: a training set $\mathcal{D}_t$ and a hold-out set $\mathcal{D}_h$. Probabilistic classifiers, outputing the probability of each label, are learned from $\mathcal{D}_t$ to represent conditional probability of labels, and the probability is calculated based on the output of the learned classifiers over $\mathcal{D}_h$. First, three probabilistic classifiers f_j, f_k and $f_{j|k}$ are learned on $\mathcal{D}_t$ to approximate conditional probabilities $\hat{P}(y_j = 1|\mathbf{x})$, $\hat{P}(y_k = 1|\mathbf{x})$ and $\hat{P}(y_j = 1|y_k, \mathbf{x})$, respectively. Then corresponding probabilities are computed by conducting f_j, f_k and $f_{j|k}$ on $\mathcal{D}_h$. Last, $I_{\hat{P}}(Y_j; Y_k|\mathbf{X})$ is estimated by

$$I_{\hat{P}}(Y_j; Y_k|\mathbf{X}) = \frac{1}{|\mathcal{D}_h|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_h} \mathbb{E}_{\hat{P}(y_j|y_k,\mathbf{x})}\mathbb{E}_{\hat{P}(y_k|\mathbf{x})} \log \frac{\hat{P}(y_j|y_k, \mathbf{x})}{\hat{P}(y_j|\mathbf{x})}. \tag{13}$$

5.1.3 Construction of PACC

After obtaining the skeleton of the polytree, our next task is to assign directions to its edges, that is, an ordering of the nodes to complete the polytree. First we assign some or all directions to the skeleton by finding causal basins. This is implemented by finding multi-parent nodes and the corresponding directionality. The detailed procedure is as follows. Fig. 5 shows three possible graphical models over triplets A, B and C. Here Types 1 and 2 are indistinguishable because they share the same joint distribution, while Type 3 is different from Types 1 and 2. In Type 3, A and C are marginally independent, so that we have

$$I(A; C) = \sum_{a,c} P(a, c) \log \frac{P(a, c)}{P(a)P(c)} = 0. \tag{14}$$

In this case, B is a multi-parent node. More generally, we can do *Zero-Mutual Information* (Zero-MI) testing for a triplet, Y_j with its two neighbors Y_a and Y_b: if $I(Y_a; Y_b) = 0$, then Y_a and Y_b are parents of Y_j, and Y_j becomes a multi-parent node. The other non-parent neighbors will be treated as Y_j's child nodes. By performing the Zero-MI testing for every pair of Y_j's direct neighbors, $\mathbf{pa}(Y_j)$ and a causal flow outside Y_j is determined, by which a causal basin will be found. In PACC, $\mathbf{pa}(Y_j)$ can be more than one node, so that the model is more flexible than that of BCC using a tree.

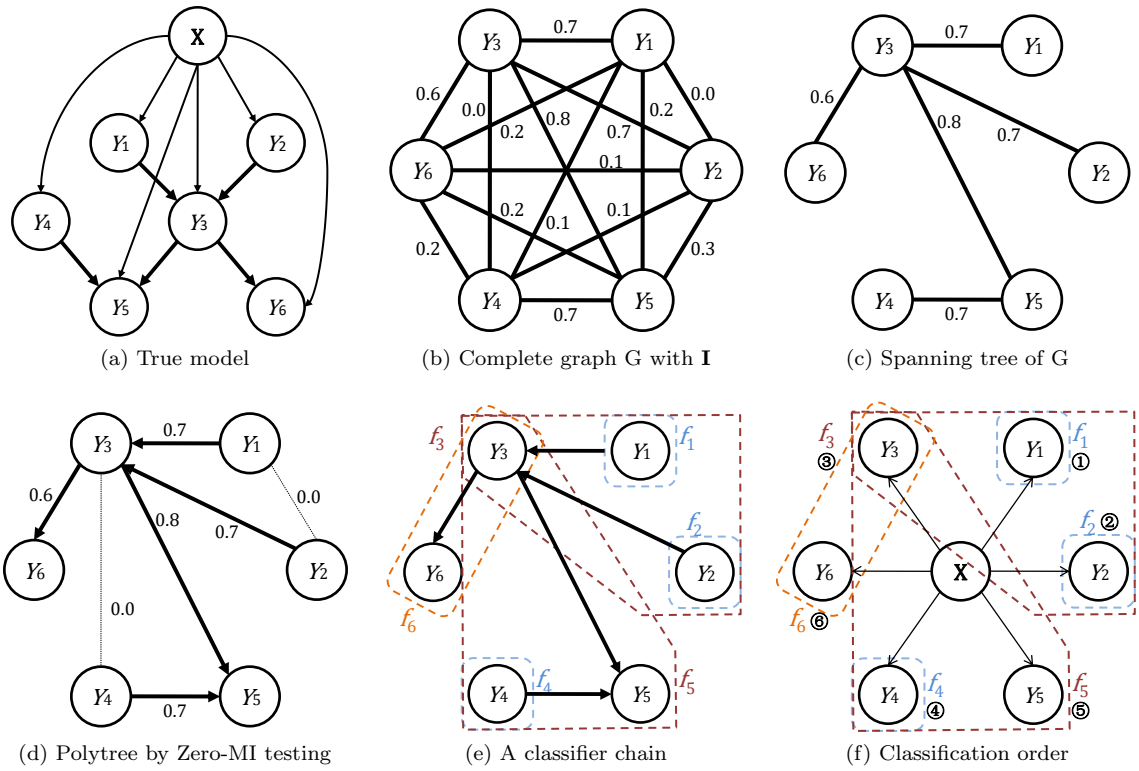
In order to build a classifier chain by the learned directions, we rank the labels to form a chain and then train a classifier for every label following the chain. The ranking strategy is simple: the parents should be ranked higher than their descendants, and the parents sharing the same child should be ranked in the same level. Hence, learning of a label is not performed until the labels with higher ranks, including its parents, have been learned. That is, a kind of lazy decision is made. In PACC, we choose logistic regression with ℓ_2 regularization as the baseline classifier. Therefore, a set of L logistic regressiors $\mathbf{f} = \{f_j\}_{j=1}^{L}$ is learned, each of which is trained by treating the union of $\mathbf{x}$ and $\mathbf{pa}(y_j)$ as new augmented attributes $\tilde{\mathbf{x}}_j = (\mathbf{x}, \mathbf{pa}(y_j))^T$, shown as follows:

$$f_j(\tilde{\mathbf{x}}_j, \boldsymbol{\theta}_j) = P(y_j = 1|\tilde{\mathbf{x}}_j, \boldsymbol{\theta}_j) = \frac{1}{1 + e^{-\boldsymbol{\theta}_j^T \tilde{\mathbf{x}}_j}}, \quad j = 1, ..., L, \tag{15}$$

where $\boldsymbol{\theta}_j$ is the model parameters for Y_j, which could be learned by maximizing the regularized log-likelihood given the training set:

$$\max_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} \log P(y_j^{(i)}|\tilde{\mathbf{x}}_j^{(i)}, \boldsymbol{\theta}_j) - \lambda\|\boldsymbol{\theta}_j\|_2^2, \tag{16}$$

where λ is a trade-off coefficient to avoid overfitting by generating sparse parameters $\boldsymbol{\theta}_j$. Then traditional convex optimization techniques, such as Quasi-Newton method with BFGS iteration [23], can be used to learn the parameters.



(a) True model (b) Complete graph G with **I** (c) Spanning tree of G

(d) Polytree by Zero-MI testing (e) A classifier chain (f) Classification order

Fig. 6: Learning (b-e) and prediction (f) phases of PACC. The true but hidden graphical model (a) is learned from data. (b) Construct a complete graph G with edges weighted by the mutual information **I**. (c) Construct a spanning tree in G. (d) Make directions by Zero-MI testing. (e) Train six probabilistic classifiers f_1-f_6. (f) Prediction is made in the order of circled numbers.

5.1.4 Classification

Exact inference in the prediction phase, as shown in (3), is NP-hard in directed acyclic graphs. However, in polytrees, using the max-sum algorithm [26], we can make exact/exhaustive inference in a reasonable time by bounding the indegree of nodes.

Two phases are performed in order. The first phase, we begin at the root(s) and propagates testing downward to the leaves. The conditional probability table for each node is calculated on the basis of its local graphical structure. In the second phase, message propagation starts upward from the leaves to the root(s). In each node Y_j, we collect all the incoming messages and finding the local maximum with its value $\hat{y}_j$. In this way, we have the *Maximum a Posteriori* (MAP) estimate $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_L)$ such as

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \prod_{j=1}^{L} f_j(\mathbf{x}, \hat{\mathbf{pa}}(y_j)) = \arg\max_{1,\ldots,y_r,y_l,\ldots,y_L} \Big[f_r(\mathbf{x}) \big[\cdots f_l(\hat{\mathbf{pa}}(y_l), \mathbf{x}) \big] \Big], \tag{17}$$

where Y_l represents a leaf and Y_r a root, respectively.

An example of learning and prediction in PACC is shown in Fig. 6. The algorithm of PACC is depicted in Algorithm 1.

5.2 Label-dependent feature selection

A two-stage feature selection approach consisting of classifier-independent *filter* and classifier-dependent *wrapper*, has been recommended to gain a good trade-off between classification performance and computation time in [19]. Motivated by this study, we develop a two-stage feature selection approach for CC-based methods based on the simple filter algorithm, in order to find label-dependent, equivalently class-dependent, features [1] and save label correlations during feature selection. In this way, we expect that the proposed approach enable to improve classification performance and reduce the computational complexity in both learning and prediction phases.

According to whether features are evaluated individually or not, the existing filter algorithms can be categorized into two groups: feature weighing algorithms and subset search algorithms [42]. Feature

Algorithm 1 Algorithm of PACC

Input: $\mathcal{D}$: training set, $\mathcal{T}$: test set, $\mathbf{f} = \{f_j\}_{j=1}^{L}$: multi-label probabilistic classifier
Output: $\hat{\mathbf{y}}$: prediction on a test instance $\hat{\mathbf{x}}$, $\hat{\mathbf{x}} \in T$

Training:
1: Transform $\mathcal{D}$ into $\{\mathcal{D}_j\}_{j=1}^{L}$, where $\mathcal{D}_j = \{\mathbf{x}^{(i)}, y_j^{(i)}\}_{i=1}^{N}$;
2: Calculate the mutual information matrix $\mathbf{I} = \{I_{jk}\}_{L\times L}$ according to (13);
3: Construct a polytree $B = \{\mathbf{pa}(Y_j)\}_{j=1}^{L}$ on $\mathbf{I}$, and form the **chain**;
4: Transform $\{\mathcal{D}_j\}_{j=1}^{L}$ into $\{\mathcal{D}_j^{+}\}_{j=1}^{L}$ based on B, where $\mathcal{D}_j^{+} = \{\mathbf{x}^{(i)} \cup \mathbf{pa}(y_j)^{(i)}, y_j^{(i)}\}_{i=1}^{N}$;
5: **for** $j \in$ **chain do**
6: Learn a probabilistic classifier f_j on $\mathcal{D}_j^{+}$ according to (15) and (16);

Testing:
7: **for** $\hat{\mathbf{x}} \in \mathcal{T}$ **do**
8: Return $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \prod_{j=1}^{L} f_j(\hat{\mathbf{x}}, \hat{\mathbf{pa}}(y_j))$ according to (17);

weighting algorithms evaluate the weights of features individually, and rank them by the relevance to the target class. It is quite efficient to remove irrelevant features, but totally ignores the correlations among features. On the other hand, redundant features that are strongly correlated to others also harm the performance of learning algorithm [17]. Subset search algorithms aim to overcome such limitation, and still maintain a reasonable time complexity compared with the wrapper algorithm. It searches through candidate feature subsets guided by a certain evaluation measure which captures the goodness of each subsets [24]. In this study, we propose a two-stage approach by using both feature weighting and subset search in order to select label-dependent features.

5.2.1 Label-dependent feature weighting

In the first stage, we develop a novel Multi-Label Information Gain (MLIG) algorithm based on feature weighting to efficiently remove irrelevant features for each label. IG has been frequently used as an evaluation criterion for feature weighting in various machine learning tasks [40]. Given a label variable Y_j and a feature variable X_k, IG measures the amount of the entropy of Y_j reduced by knowing X_k,

$$\begin{aligned} I(Y_j; X_K) &= H(Y_j) - H(Y_j|X_k) \\ &= -\sum_{y_j\in\{0,1\}} P(y_j)\log P(y_j) + \sum_{x_k\in\mathcal{V}_k} P(x_k) \sum_{y_j\in\{0,1\}} P(y_j|x_k)\log P(y_j|x_k), \end{aligned} \tag{18}$$

where $\mathcal{V}_k$ denotes the value space of the feature variable X_k. In practice, the numeric features should be discretized beforehand for the computational efficiency. For the multi-label datasets, a straightforward way to apply IG is to rank all the features for each label according to (18), and then select the top-ranked features to feed the post-process.

However, it is a non-trivial thing to choose an appropriate threshold for filtering out irrelevant features. In addition, in the MLC setting, it is unreasonable to set the same threshold for all labels due to the label imbalance problem as stated in Sec. 4.3. For the labels with higher imbalance ratio, the number of positive instance may be insufficient for building an accurate classifier, in which case a smaller number of features should be chosen. To overcome the problem, in MLIG we set the percentage α_j of selected features for the label variable Y_j according to the following:

$$\alpha_j = 2r \cdot \frac{e^{\beta_j} - 1}{e^{\beta_j} + 1} + r, \quad \beta_j = \frac{\overline{\mathrm{IR}}}{\mathrm{IR}(Y_j)\cdot(\mathrm{CVIR}+1)}, \tag{19}$$

where r is a factor controlling the range of α_j so that $\alpha_j \in [r, 3r]$. According to (19), we can see that the value of α_j is close to $3r$ for the majority labels in well balanced datasets, and α_j becomes r for the minority labels in highly imbalanced datasets. As a result, a smaller number of features is selected for each minority label in an imbalanced dataset, and vice versa.

In this way, MLIG first calculates a feature-label information gain matrix according to (18), then ranks the features for each label and selects most relevant label-dependent features up to $m_j = \alpha_j M$, $j = 1, ..., L$. Finally, we transform the original data $\mathcal{D}_j = \{(\mathbf{x}^{(i)}, y_j^{(i)})\}_{i=1}^{N}$ into $\mathcal{Z}_j = \{(\mathbf{z}_j^{(i)}, y_j^{(i)})\}_{i=1}^{N}$, $\mathbf{z}_j \in \mathbb{R}^{m_j}$ by eliminating irrelevant features.

5.2.2 Label-dependent feature subset selection

Although the MLIG approach works for feature selection to some extend, it is unable to eliminate the redundant features. Thus we consider to develop a feature subset selection algorithm in order to find a more compact feature subset by incorporating the label dependency modeled by the polytree structure.

In this stage, we extend the Correlation-based Feature Selection (CFS) [12], one of the subset search algorithms, to remove redundant features. CFS is conducted once the polytree $B = \{\mathbf{pa}(Y_j)\}_{j=1}^{L}$ has been constructed. In the proposed Multi-Label CFS (MLCFS) approach, we apply CFS on the label-specific feature subspace, taking label correlations modeled by B into account. More specifically, given a label variable Y_j with its dataset $\mathcal{Z}_j^+ = \{\tilde{\mathbf{z}}_j^{(i)}, y_j^{(i)}\}_{i=1}^{N}$, where $\tilde{\mathbf{z}}_j = \mathbf{z}_j \cup \mathbf{pa}(y_j)$, the merit of a feature subset S_j of $\tilde{n}_j$ ($\tilde{n}_j = n_j + |\mathbf{pa}(Y_j)|$) features is evaluated by

$$\text{Merit}(S_j) = \frac{\tilde{n}_j \rho_{Y_j \tilde{Z}}}{\sqrt{\tilde{n}_j + \tilde{n}_j(\tilde{n}_j - 1)\rho_{\tilde{Z}\tilde{Z}}}}, \tag{20}$$

where the mean correlations $\rho_{Y_j \tilde{Z}}$ and $\rho_{\tilde{Z}\tilde{Z}}$ are calculated according to

$$\rho_{Y_j \tilde{Z}} = \frac{2}{\tilde{n}_j} \sum_{k=1}^{\tilde{n}_j} \frac{I(\tilde{Y}_j; \tilde{Z}_k)}{H(\tilde{Y}_j) + H(\tilde{Z}_k)}, \quad \rho_{\tilde{Z}\tilde{Z}} = \frac{4}{\tilde{n}_j(\tilde{n}_j - 1)} \sum_{\substack{k,l=1 \\ k \neq l}}^{\tilde{n}_j} \frac{I(\tilde{Z}_k; \tilde{Z}_l)}{H(\tilde{Z}_k) + H(\tilde{Z}_l)}. \tag{21}$$

MLCFS first calculates the feature-feature and feature-label correlation matrices, and then employs a heuristic search algorithm, such as Best First [17], with the start set $\mathbf{pa}(Y_j)$ to search the feature subset of space Y_j by maximizing (20). In this way, the dimensionality of the feature space is reduced from m_j to n_j, typically $n_j \ll m_j$. We transform the data $\mathcal{Z}_j^+ = \{\tilde{\mathbf{z}}_j^{(i)}, y_j^{(i)}\}_{i=1}^{N}$ into $\mathcal{V}_j^+ = \{\tilde{\mathbf{v}}_j^{(i)}, y_j^{(i)}\}_{i=1}^{N}$, where $\tilde{\mathbf{v}}_j = \mathbf{v}_j \cup \mathbf{pa}(y_j)$, $\mathbf{v}_j \in \mathbb{R}^{n_j}$. Finally, $\mathcal{V}_j^+$ is used to learn the probabilistic classifier f_j.

Algorithm 2 gives the algorithm of PACC with Label-Dependent Features, named PACC-LDF. In the training phase, PACC-LDF first performs problem transformation in Step 1, applies MLIG to remove irrelevant features and transforms the training set $\{\mathcal{D}_j\}$ into $\{\mathcal{Z}_j\}$ from Steps 2 to 4. Then a polytree B is built on $\{\mathcal{Z}_j\}$ from Steps 5 to 6, and $\{\mathcal{Z}_j\}$ is transformed into $\{\mathcal{Z}_j^+\}$ based on B in Step 7. After that, MLCFS is performed on $\{\mathcal{Z}_j^+\}$, which is further transformed into $\{\mathcal{V}_j^+\}$ in Steps 8 to 10. Finally, based on the dataset $\{\mathcal{V}_j^+\}$ with label-dependent features, a multi-label probabilistic classifier $\{f_j\}$ is learned at Step 11. In the testing phase, a test dataset $\mathcal{T}$ is first projected into the lower-dimensional feature subspaces, and then feed to the learned classifier for prediction in Steps 12 to 15. Fig. 7 shows the framework of PACC-LDF in terms of training and testing phases.

Algorithm 2 Algorithm of PACC-LDF

Input: $\mathcal{D}$: training set, $\mathcal{T}$: test set, $\mathbf{f} = \{f_j\}_{j=1}^{L}$: multi-label probabilistic classifier
Output: $\hat{\mathbf{y}}$: prediction on a test instance $\hat{\mathbf{x}}$, $\hat{\mathbf{x}} \in \mathcal{T}$

Training:

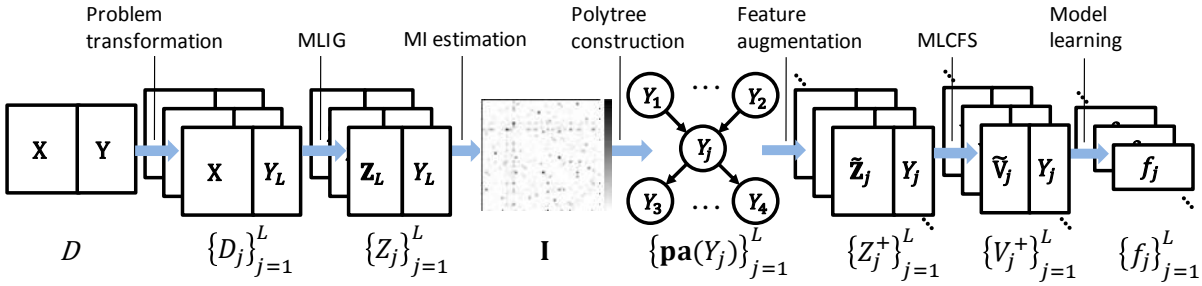
1: Transform $\mathcal{D}$ into $\{\mathcal{D}_j\}_{j=1}^{L}$, where $\mathcal{D}_j = \{\mathbf{x}^{(i)}, y_j^{(i)}\}_{i=1}^{N}$;
2: **for** $j = 1$ to L **do**
3: Perform MLIG on $\mathcal{D}_j$ according to (18), i.e., $g_j' : \mathbf{x}|_{M\times 1} \mapsto \mathbf{z}_j|_{m_j \times 1}$;
4: Transform $\mathcal{D}_j$ into $\mathcal{Z}_j = \{(\mathbf{z}_j^{(i)}, y_j^{(i)})\}_{i=1}^{N}$, where $\mathbf{z}_j = g_j'(\mathbf{x})$;
5: Calculate the mutual information matrix $\mathbf{I} = \{I_{jk}\}_{L\times L}$, where I_{jk} is computed according to (13) based on Z_j and Z_k;
6: Construct a polytree $B = \{\mathbf{pa}(Y_j)\}_{j=1}^{L}$ on $\mathbf{I}$, and form the **chain**;
7: Transform $\{\mathcal{Z}_j\}_{j=1}^{L}$ into $\{\mathcal{Z}_j^+\}_{j=1}^{L}$ based on B, where $\mathcal{Z}_j^+ = \{\mathbf{z}_j^{(i)} \cup \mathbf{pa}(y_j)^{(i)}, y_j^{(i)}\}_{i=1}^{N}$;
8: **for** $j \in$ **chain do**
9: Conduct MLCFS on $\mathcal{Z}_j^+$ by Best First search with the start set $\mathbf{pa}(Y_j)$ according to (20), i.e., $g_j'' : \mathbf{z}_j|_{m_j\times 1} \mapsto \mathbf{v}_j|_{n_j \times 1}$;
10: Transform $\mathcal{Z}_j^+$ into $\mathcal{V}_j^+ = \{(\mathbf{v}_j^{(i)} \cup \mathbf{pa}(y_j), y_j^{(i)})\}_{i=1}^{N}$, where $\mathbf{v}_j = g_j''(\mathbf{z}_j)$;
11: Learn a probabilistic classifier f_j on $\mathcal{V}_j^+$ according to (15) and (16);

Testing:

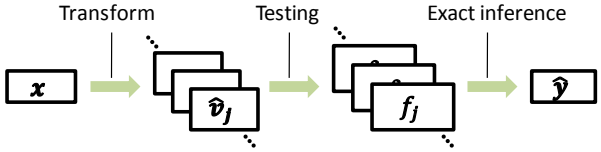
12: **for** $\hat{\mathbf{x}} \in \mathcal{T}$ **do**
13: **for** $j \in$ **chain do**
14: Transform $\hat{\mathbf{x}}_j$ into $\hat{\mathbf{v}}_j$, i.e., $\hat{\mathbf{v}}_j = (g_j'' \circ g_j')(\hat{\mathbf{x}}_j)$;
15: Return $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \prod_{j=1}^{L} f_j(\hat{\mathbf{v}}_j, \hat{\mathbf{pa}}(y_j))$ according to (17);

5.3 Discussion

PACC-LDF can be considered as a general version of PACC, since PACC-LDF selects label-dependent features during the model building of PACC, in order to improve its performance and reduce time complexity. By applying only one stage of the proposed feature selection approach, we can have two



(a) Learning phase of PACC-LDF



(b) Prediction phase of PACC-LDF

Fig. 7: Flow chart of the proposed PACC-LDF method. The learning phase (a) consists of seven steps: problem transformation, MLIG, MI estimation, polytree construction, feature augmentation, MLCFS and model learning. The prediction phase (b) consists of three steps: instance transform, testing and exact inference.

variants of PACC-LDF: PACC with MLIG (PACC-MLIG) (with Steps 2-4 only) and PACC with MLCFS (PACC-MLCFS) (with Steps 8-10 only). Note that the two-stage feature selection can also be applied to the other MLC methods. For instance, it could be directly incorporated with binary relevance (BR) by removing Steps 5 to 7 from Algorithm 2, leading to BR-LDF. For classifier chains (CC) based methods, we can do this by changing the content of $\mathbf{pa}(Y_j)$, producing CC-LDF and BCC-LDF.

6 Multi-label data sets with evaluation metrics

6.1 Multi-label data sets

Given a multi-label dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ with L labels and M features, we introduce the label cardinality $\frac{1}{N}\sum_{i=1}^{N}|\mathbf{y}^{(i)}|$, the label density $\frac{1}{NL}\sum_{i=1}^{N}|\mathbf{y}^{(i)}|$ and the number of distinct label sets $|\{\mathbf{y}|(\mathbf{x},\mathbf{y}) \in D\}|$ in order to depict its statistical properties. In addition, we also report the imbalance level of $\mathcal{D}$ by $\overline{\text{IR}}$ and CVIR defined in (9). As a rule of thumb [5], a dataset $\mathcal{D}$ is considered as an imbalanced dataset if $\overline{\text{IR}}$ is higher than 1.5 and CVIR exceeds 0.2. In this sense, all the datasets except the Scene and Emotions datasets are imbalanced, indicating the necessity of alleviating this problem in MLC methods. Table 2 reports the statistics of twelve benchmark multi-label datasets from a variety of domains used in the experiments. According to the size of N, M and L, we treat the first eight datasets as regular-scale datasets and the last four as large-scale datasets.

6.2 Evaluation metrics

The existing multi-label evaluation metrics can be separated into two groups: instance-based metrics and label-based metrics [35]. To evaluate the performance of a MLC method on a test data set $\mathcal{T} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_t}$, we use two instance-based metrics:

$$\text{Exact-Match} := \frac{1}{N_t}\sum_{i=1}^{N_t}\mathbb{1}_{\hat{\mathbf{y}}^{(i)}=\mathbf{y}^{(i)}}, \quad \text{Accuracy} := \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{\sum_{j=1}^{L} y_j^{(i)}\cdot\hat{y}_j^{(i)}}{\sum_{j=1}^{L} y_j^{(i)} + \sum_{j=1}^{L}\hat{y}_j^{(i)} - \sum_{j=1}^{L} y_j^{(i)}\hat{y}_j^{(i)}}, \tag{22}$$

and two label-based metrics:

$$\text{Macro-F1} := \frac{1}{L}\sum_{j=1}^{L}\frac{2\sum_{i=1}^{N_t}\hat{y}_j^{(i)}\cdot y_j^{(i)}}{\sum_{i=1}^{N_t}\hat{y}_j^{(i)} + \sum_{i=1}^{N_t} y_j^{(i)}}, \quad \text{Micro-F1} := \frac{2\sum_{j=1}^{L}\sum_{i=1}^{N_t}\hat{y}_j^{(i)}\cdot y_j^{(i)}}{\sum_{j=1}^{L}\sum_{i=1}^{N_t}\hat{y}_j^{(i)} + \sum_{j=1}^{L}\sum_{i=1}^{N_t} y_j^{(i)}}. \tag{23}$$

Among the metrics, Exact-Match is the most stringent measure, especially for the MLC problems with a large number of labels, since it does not evaluate the partial match of a label set. In spite of that,

Table 2: Statistics of twelve benchmark multi-label datasets. In below, N, M and L are the data size in instances, features and labels, respectively. Cardinality, Density and Distinct denotes the label cardinality, the label density and the number of distinct label combinations, respectively. $\overline{\text{IR}}$ and CVIR together depict the degree of label imbalance, which are defined in (9).

Dataset†	N	M	L	Type‡	Cardinality	Density	Distinct	$\overline{\text{IR}}$	CVIR	Dom
Emotions	593	72	6	a	1.869	0.311	27	1.478	0.180	music
Scene	2407	294	6	a	1.074	0.179	15	1.254	0.122	image
Yeast	2417	103	14	a	4.237	0.303	198	6.623	1.763	biology
Birds	645	260	19	c	1.014	0.053	133	5.407	0.817	audio
Genbase	662	1186	27	b	1.252	0.046	32	37.315	1.449	biology
Medical	978	1449	45	b	1.245	0.028	94	89.501	1.148	text
Enron	1702	1001	53	b	3.378	0.064	753	73.953	1.960	text
Languagelog	1460	1004	75	b	1.180	0.016	286	39.267	1.311	text
Rcv1s1	6000	944	101	a	2.880	0.029	837	59.333	2.380	text
Corel16k1	13766	500	153	b	2.859	0.019	1791	34.890	0.815	image
Bibtex	7395	1836	159	b	2.402	0.015	2856	12.498	0.405	text
Corel5k	5000	499	374	b	3.522	0.009	3175	189.568	1.527	music

† The source of datasets: http://mulan.sourceforge.net/datasets-mlc.html
‡ Type of features. a: numeric, b: nominal, c: both numeric and nominal

according to the definition, it is a good measure to measure how well label correlations are modeled. Accuracy is useful to measure the performance of a classifier in terms of both positive and negative prediction ability. Unlike Exact-Match, both Macro-F1 and Micro-F1 are able to take the partial match of labels into account. In addition, as stated in [33], Macro-F1 is more sensitive to the performance of rare categories (the labels in minority), while Micro-F1 is affected more by the major categories (the labels in majority). Hence, joint use of Macro-F1 and Micro-F1 should be a good supplement for the instance-based evaluation metrics to evaluate the performances of MLC methods.

7 Experiments

7.1 Implementation issues

In both feature weighting (18) and feature subset selection (20), calculation of mutual information is extensively performed. For the discrete and categorical feature variable X (label variable Y is originally binary), the calculation of mutual information is simple and straightforward. Given a sample of n i.i.d. observations $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, based on the law of large numbers, we have the following approximation:

$$I(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \approx \frac{1}{n} \sum_{i=1}^{n} \log \frac{\hat{P}(x^{(i)}, y^{(i)})}{\hat{P}(x^{(i)})\hat{P}(y^{(i)})}, \tag{24}$$

where $\hat{P}$ denotes the empirical probability distribution. When the feature variable X is continuous, it becomes quite difficult to compute mutual information $I(X;Y)$, since it is typically impossible to obtain $\hat{P}$. One of solutions is to use kernel density estimation [14], but it is computationally expensitve and typically difficult to select good value of its bandwidth. To circumvent this difficulty, in practice, we compute $I(X;Y)$ with continuous feature X by applying data discretization as preprocessing. In this study, the continuous feature X is discretized based on its mean μ_X and standard deviation σ_X. For example, we can apply a similar discretization approach used in [4], which divides a numeric value of feature variable X into one of three categories $\{-1, 0, 1\}$ according to $\mu_X \pm \sigma_X$. The experimental results demonstrate the efficiency of such simple data discretization approach for approximating $I(X;Y)$ to perform feature selection.

In addition, the calculation of conditional mutual information in (13) for building the polytree is computational expensive for large-scale datasets. To reduce the training cost and make the proposed PACC and PACC-LDF tractable for large problems, normalized marginal mutual information estimation, rather than conditional mutual information estimation (13), is used to model label correlations in PACC-related methods for large-scale datasets. The normalized mutual information is defined in the following:

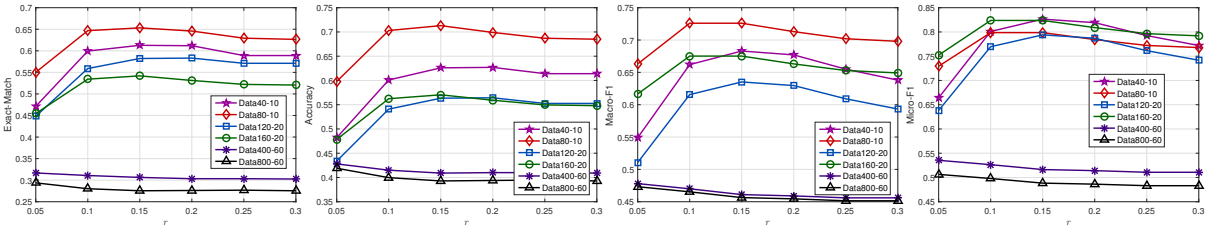
$$NI(X;Y) = \frac{I(X;Y)}{\min\{H(X), H(Y)\}}. \tag{25}$$

Compared with $I(X;Y)$, the advantage of $NI(X;Y)$ is that $NI(X;Y)$ enables to alleviate the negative effect resulting from the label imbalance problem, as we have discussed in Sec. 4.3.

Table 3: Statistics of six synthetic multi-label datasets.

Dataset	N	$M^{\dagger}$	L	Cardinality	Density	Distinct	$\overline{\text{IR}}$	CVIR
Data40-10	500	40	10	1.260	0.126	57	2.235	0.400
Data80-10	500	80	10	1.182	0.118	53	1.466	0.400
Data120-20	1000	120	20	1.404	0.070	227	1.810	0.368
Data160-20	1000	160	20	1.378	0.069	219	1.698	0.354
Data400-60	2000	400	60	2.187	0.036	1302	1.444	0.214
Data800-60	2000	800	60	2.147	0.036	1302	1.478	0.251

† The ratio of relevant, irrelevant and redundant features is $2:1:1$.



(a) Exact-Match (b) Accuracy (c) Macro-F1 (d) Micro-F1

Fig. 8: The performances of PACC-LDF in terms of four evaluation metrics on six synthetic datasets by varying the value of r from 0.05 to 0.3 by step 0.05.

7.2 Experimental setting

The methods used in the experiments were implemented based on Mulan[1] and Meka[2], and performed on six synthetic datasets and twelve benchmark datasets. To evaluate the classification performance, 5-fold and 3-fold cross validation were used for the eight regular-scale and four large-scale datasets, respectively. In the experiments we chose logistic regression with ℓ_2 regularization as the baseline classifier, and set the constant value $\lambda = 0.1$ for the trade-off parameter λ in (16) for all MLC methods. To reduce the training cost, normalized mutual information, instead of conditional mutual information (13), was calculated for large-scale datasets. The experiments were conducted in a computer configured with an Intel Quad-Core i7-4770 CPU at 3.4GHz with 4G RAM.

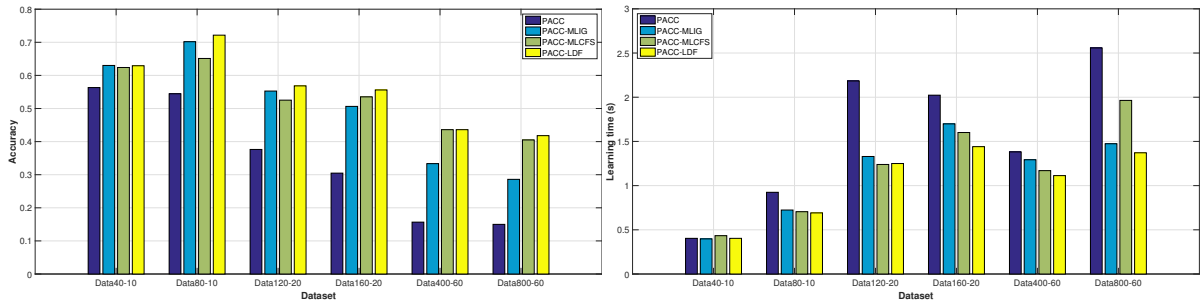
7.3 Experiments on synthetic datasets

In this section, we conduct experiments on six synthetic multi-label datasets to evaluate the performances of PACC with its three variants, PACC-MLIG, PACC-MLCFS and PACC-LDF. In total, six synthetic datasets, including four regular-scale sets and two large-scale sets, were generated according to the method in [34]. In each data set, instances were produced by randomly sampling from R hypercubes (labels) in the M-dimensional feature space, and thus the dataset is represented by Data$M$-R. The M-dimensional features consisted of three parts: relevant features, irrelevant features and redundant features. The irrelevant features were randomly generated, and the redundant features were the copies of existing relevant features. In addition, in order to simulate real-world multi-label data, classification noise was added into these synthetic datasets, which flips the value of each label for a instance in a random manner with a probability of 0.02. The statistics of the synthetic datasets are reported in Table 3.

First, we performed experiments on PACC-LDF by changing the value of factor r in (19), which controls the lower and upper bounds of α_j by $r \leq \alpha_j \leq 3r$ according to (19). Experimental results in four evaluation metrics are shown in Fig. 8, by which we can reach the following two conclusions: (1) In the regular-sized datasets, PACC-LDF works worse for a small value of r, $r < 0.1$, but becomes better and stable as r exceeds 0.15; (2) In the large-sized set, PACC-LDF performs better when the value of r is small, and works slightly worse if r exceeds 0.15. Therefore, in the rest of paper, r is set to the moderate value of 0.15 and 0.05 for regular-scale and large-scale sets, respectively.

In Fig. 9, the performances of PACC, PACC-MLIG, PACC-MLCFS and PACC-LDF in Accuracy and Learning time are reported. Note that we do not show the performances in other metrics here, since similar results and patterns can be observed. The proposed LDF and its variants significantly improve

[1] http://mulan.sourceforge.net/
[2] http://meka.sourceforge.net/



(a) Accuracy (b) Learning time

Fig. 9: The performance of PACC with its three variants in Accuracy and Learning time (in seconds) on six synthetic datasets.

the performance of PACC. Specifically, PACC-LDF works best among the four methods, and achieves at least 10% of performance improvement compared with the original PACC, indicating the effectiveness of the proposed two-stage feature selection approach. In terms of learning time, PACC-LDF consumed the least time on the last five datasets. In testing time, all the methods consumed similar time on regular-scale datasets, but PACC-LDF cost the least time on two large-scale datasets. Therefore, the two-stage feature selection approach, LDF, rather than MLIG and MLCFS, is employed in the following experiments.

7.4 Experiments on real-world datasets

Next we evaluate the performances of popular MLC methods on the twelve real-world benchmark multi-label datasets in Table 2. This part of experiment is composed of three major parts. In the first part, we compared PACC with three CC-based methods, including BR, CC and BCC, to demonstrate the effectiveness of the polytree structure on capturing label correlations. In the second part, PACC-LDF is compared with three state-of-the-art MLC methods in terms of classification accuracy and execution time. In the last part, CC-based methods are compared with their LDF variants in a pairwise way to evaluate the performance of the two-stage feature selection approach. In addition, the comparing results of LDF with traditional feature selection algorithms are presented. The MLC methods used in this section are summarized as follows:

- CC-based methods have been introduced in Section 3. In CC, the chain is established in a randomly determined order. In BCC, the normalized mutual information is used for marginal dependency estimation on each label pair, since the performance could be slightly improved without consuming extra processing time.
- Multi-Label k-Nearest Neighbors (MLkNN) [46] originates from the traditional k-nearest neighbors algorithm. For each test instance, according to the label assignments of its k nearest neighbors in the training set, the prediction is made on the basis of MAP principal. In the experiments, we set $k = 10$, by following the suggestion in the literature [46].
- RAndom k LabELsets (RAkEL) [37] is an ensemble variant of the Label Combination (LC) method. RAkEL transforms an MLC problem into a set of smaller MLC problems, by training m LC models using random k-subsets of the original label set. To make it executable in a limited time cost (24h), RAkEL employed the C4.5 decision tree as its baseline single-label classifier for large-scale datasets. We set $k = 3$ and $m = 2L$ as recommended in [37].
- By building a Hierarchy Of Multi-label classifiERs (HOMER) on the basis of balanced k-means clustering, HOMER [36] reduces the complexity of prediction and addresses the label imbalance problem. According to the experimental results in [36], the number k of clusters for building the hierarchical structure was set to 4. In addition, Binary Relevance with ℓ_2 regularized logistic regression was used as its baseline multi-label classifier.

7.4.1 Comparison of PACC with CC-based methods

The resutls of CC-based methods are summarized in Table 4. In terms of instance-based evaluation metrics, Exact-Match and Accuracy, PACC was the best or competitive with the best methods, except for the Yeast dataset. It is understandable because PACC is a subset 0-1 risk minimizer benefiting from its ability on the polytree structure as well as exact inference. In these metrics, CC is the second best, while BCC is the third in most cases. This is probably because BCC models only label pairwise

Table 4: Experimental results (mean±std) of four CC-based methods on twelve multi-label datasets in terms of four evaluation metrics.

Method	Exact-match											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
BR	.239±.069	.466±.024	.139±.015	**.476±.060**	.949±.030	.580±.046	.089±.017	.224±.032	.047±.003	.006±.001	.110±.005	.003±.002
CC	.260±.044	.546±.017	**.194±.016**	**.476±.056**	.974±.009	**.586±.040**	.099±.011	**.237±.023**	.111±.007	.012±.001	.120±.006	**.009±.001**
BCC	.239±.050	.494±.028	.136±.012	**.476±.064**	**.977±.007**	.575±.042	.095±.017	.234±.018	.059±.004	.007±.001	.108±.009	.003±.001
PACC	**.261±.030**	**.568±.019**	.125±.079	.474±.057	.974±.009	**.586±.033**	**.112±.025**	**.237±.025**	**.113±.008**	**.016±.003**	**.128±.002**	**.009±.004**
Method	**Accuracy**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
BR	.511±.049	.567±.020	**.512±.018**	.576±.055	.973±.017	**.698±.029**	.377±.015	**.315±.020**	.243±.003	**.117±.006**	.287±.001	.097±.003
CC	.516±.038	.608±.020	.505±.013	.579±.051	.986±.006	.672±.028	.380±.009	.275±.012	.289±.003	.093±.001	**.299±.003**	.116±.001
BCC	.502±.035	.582±.028	.494±.014	**.580±.062**	**.987±.006**	.656±.028	.369±.019	.273±.008	.261±.002	.073±.006	.288±.005	.096±.002
PACC	**.523±.033**	**.625±.018**	.400±.131	**.580±.055**	.986±.006	.668±.027	**.384±.017**	.277±.015	**.294±.011**	.101±.005	.295±.002	**.118±.009**
Method	**Macro-F1**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
BR	**.636±.043**	.652±.019	**.424±.014**	.335±.028	.725±.055	**.372±.022**	**.216±.026**	**.090±.016**	.232±.008	**.093±.007**	.297±.008	**.045±.003**
CC	.623±.035	.639±.021	.404±.019	.341±.013	.740±.050	.355±.025	.208±.028	.055±.009	**.240±.013**	.070±.002	**.308±.009**	.043±.004
BCC	.620±.029	.649±.019	.394±.007	.345±.028	**.742±.049**	.352±.024	.209±.024	.056±.009	**.240±.010**	.073±.005	.299±.009	.044±.004
PACC	.632±.034	**.657±.017**	.351±.064	**.349±.019**	.740±.050	.354±.023	.206±.020	.057±.007	**.240±.010**	.076±.001	.302±.002	.043±.004
Method	**Micro-F1**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
BR	**.650±.042**	.645±.019	**.642±.015**	.441±.048	.977±.015	**.761±.020**	**.497±.010**	**.272±.015**	.344±.003	**.194±.008**	.376±.008	.164±.002
CC	.633±.034	.631±.022	.626±.009	.449±.030	.989±.004	.750±.015	**.497±.006**	.225±.024	**.369±.001**	.142±.002	.385±.007	**.178±.002**
BCC	.632±.027	.642±.021	.629±.012	.449±.048	.990±.004	.742±.017	.489±.015	.223±.027	.360±.002	.130±.008	.371±.008	.162±.004
PACC	.642±.032	**.647±.018**	.515±.146	**.454±.037**	.989±.004	.748±.015	.492±.009	.231±.022	**.369±.012**	.154±.001	**.389±.002**	.165±.010

Table 5: Experimental results (mean±std) of PACC-LDF with three state-of-the-art methods on twelve multi-label datasets in terms of four evaluation metrics.

Method	Exact-match											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
MLkNN	.302±.035	**.638±.017**	**.192±.018**	.433±.028	.921±.020	.496±.049	.004±.004	.195±.030	.032±.006	.002±.000	.057±.008	.000±.000
RAkEL	.285±.040	.535±.025	.156±.016	.471±.067	.965±.023	.659±.044	.108±.025	.199±.023	.052±.007	.009±.002	.114±.007	.003±.001
HOMER	.223±.039	.476±.041	.136±.009	.457±.056	.950±.003	.544±.024	.091±.019	.214±.023	.054±.015	.011±.003	.062±.003	.003±.001
PACC-LDF	**.314±.046**	.595±.010	.125±.080	**.513±.047**	**.970±.012**	**.679±.019**	**.149±.018**	**.240±.030**	**.145±.013**	**.018±.004**	**.161±.008**	**.014±.004**
Method	**Accuracy**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
MLkNN	**.568±.033**	**.717±.014**	**.545±.019**	.542±.042	.964±.011	.632±.037	.369±.016	.255±.028	**.283±.009**	**.147±.005**	.188±.009	**.139±.005**
RAkEL	.550±.037	.617±.022	.515±.015	.576±.068	**.983±.011**	**.758±.035**	**.460±.023**	**.278±.019**	.277±.004	.107±.003	**.307±.007**	.083±.003
HOMER	.511±.034	.565±.033	.504±.016	.559±.056	.974±.015	.644±.014	.350±.024	.262±.018	.232±.019	.124±.002	.182±.004	.085±.005
PACC-LDF	.546±.033	.632±.007	.394±.146	**.584±.046**	**.983±.010**	**.758±.018**	.395±.017	.271±.018	.274±.012	.130±.020	.296±.012	.111±.011
Method	**Macro-F1**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
MLkNN	**.656±.025**	**.759±.013**	.426±.012	.300±.055	.619±.077	.241±.014	.126±.010	.052±.016	.183±.004	.051±.005	.178±.005	.026±.003
RAkEL	.654±.032	.673±.015	**.441±.012**	.343±.042	.736±.055	**.372±.043**	.176±.008	.063±.010	.202±.003	.048±.004	**.309±.004**	.024±.001
HOMER	.634±.029	.632±.025	.415±.013	.332±.041	.687±.049	.307±.028	**.178±.017**	.053±.017	.190±.002	.062±.004	.192±.002	.035±.001
PACC-LDF	.651±.039	.651±.011	.303±.054	**.358±.036**	**.740±.051**	.360±.019	.177±.012	**.068±.018**	**.209±.006**	.056±.005	.258±.015	**.040±.006**
Method	**Micro-F1**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
MLkNN	**.683±.027**	**.751±.011**	**.669±.015**	.416±.054	.957±.011	.697±.036	.531±.014	.186±.023	**.405±.006**	**.233±.008**	.296±.003	**.225±.007**
RAkEL	.665±.033	.665±.016	.642±.012	**.457±.063**	.986±.009	.812±.029	**.585±.009**	**.237±.009**	.389±.002	.181±.003	**.407±.007**	.151±.004
HOMER	.644±.028	.630±.027	.635±.017	.427±.044	.970±.019	.680±.012	.454±.019	.170±.033	.318±.012	.178±.002	.249±.004	.132±.006
PACC-LDF	.659±.033	.644±.008	.506±.161	.406±.051	**.988±.005**	**.805±.019**	.492±.018	.209±.037	.350±.011	.147±.023	.401±.016	.167±.015

correlations. In consistent with our theroretical analysis, BR obtains the worst result in Exact-Match due to ignoring label correlations. It is also worth noting that BR works better than CC-related methods only on the Birds set, indicating weak label correlations in that set. In Macro/Micro-F1, BR and BCC obtained competitive results with CC and PACC. This is probably because the label-based evaluation metrics emphasize more on the performance on individual label. Indeed BR is actually the hamming-loss risk minimizer and BCC only models the most important label pairwise dependency.

7.4.2 Results on PACC-LDF with the state-of-the-art

Next, PACC-LDF was compared with three popular MLC methods. The experimental results are shown in Table 5. From Table 5, we can see that PACC-LDF is the best in Exact-Match, and competitive with RAkEL and MLkNN in Accuracy and Macro-F1. In Accuracy and Micro-F1, RAkEL works the best, while MLkNN and PACC-LDF follows. To compare the performance of multiple methods on multiple datasets, we conducted the Friedman test [10] aiming to reject the null-hypothesis as equal performance among the comparing methods. Furthermore, since the null-hypothesis is rejected by the Friedman test in terms of all the metrics (Statistic F_F of Exact-Match, Accuracy, Macro-F1 and Macro-F1 are 8.8995, 3.2960, 2.9237 and 8.5074, respectively, higher than the critical value 2.8805 with significance level $\alpha = 0.05$), Nemenyi test [25] was conducted for pairwise comparison in classification performance. According to [10], the performance of two methods is regarded as significantly different if their average ranks differ by at least the *critical difference* (CD). Figure 10 shows the CD diagrams for four evaluation metrics at 0.05 significance level. In each subfigure, the CD is given above the axis, where the averaged rank is marked. In Figure 10, algorithms which are not significantly different are connected by a thick line. The test said that PACC-LDF is significantly better than both MLkNN and HOMER in Exact-Match

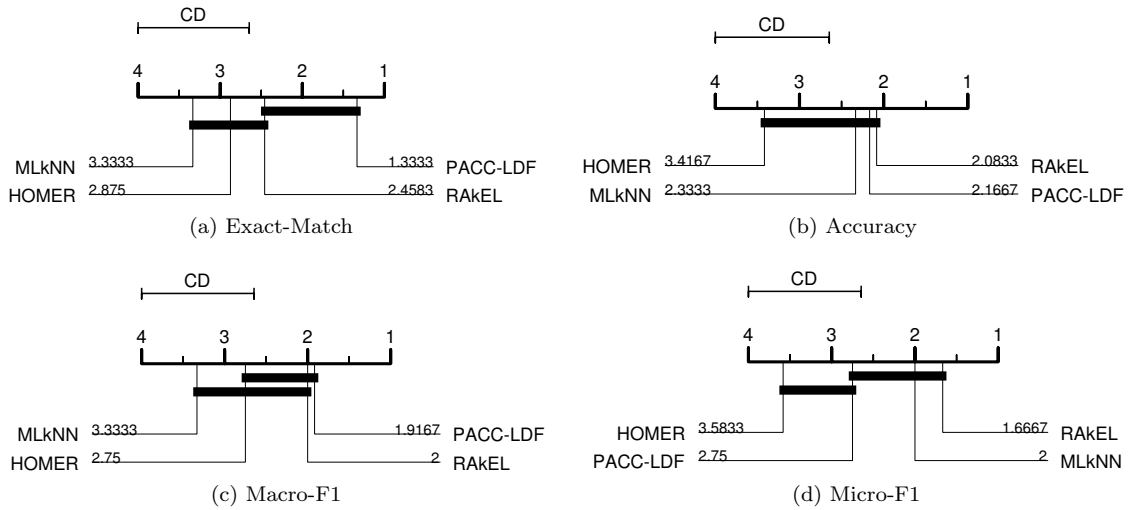


Fig. 10: CD diagrams (0.05 significance level) of four comparing methods in four evaluation metrics. The performance of two methods is regarded as significantly different if their average ranks differ by at least the Critical Difference.

Table 6: Learning and prediction time (in seconds) of eight comparing methods on twelve datasets.

Method	**Learning time**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
ML*k*NN	0.680	12.594	5.506	1.148	3.412	**0.769**	**3.508**	**6.404**	110.673	356.076	**39.467**	**42.202**
RA*k*EL	3.541	62.611	40.115	117.877	9.650	49.955	693.616	559.051	3408.508	5954.040	7744.956	2035.276
HOMER	0.758	3.936	3.259	2.873	2.180	32.025	109.900	68.922	**78.163**	**95.862**	670.111	302.119
BR	0.264	2.623	2.774	2.468	2.726	157.696	118.021	161.566	217.546	523.935	1078.550	779.548
CC	**0.260**	**2.554**	2.580	2.346	**1.819**	167.315	145.068	175.858	188.427	273.912	1231.735	311.401
BCC	0.339	3.736	2.884	2.843	1.915	157.834	129.547	168.368	190.946	254.457	1068.095	211.065
PACC	0.406	3.742	2.935	2.791	1.969	164.628	128.109	175.168	191.379	271.635	1151.284	299.956
PACC-LDF	0.430	4.074	**2.315**	**1.043**	2.403	3.763	12.224	40.733	180.480	810.881	199.482	2020.880
Method	**Prediction time**											
	Emotions	Scene	Yeast	Birds	Genbase	Medical	Enron	Languagelog	Rcv1s1	Corel16k1	Bibtex	Corel5k
ML*k*NN	0.050	2.833	1.131	0.192	0.660	**0.072**	0.734	1.420	54.888	174.226	18.619	18.501
RA*k*EL	0.007	0.050	0.087	0.046	0.109	0.680	1.268	1.944	4.305	**7.201**	33.380	8.660
HOMER	**0.003**	0.022	**0.028**	**0.012**	0.049	0.167	0.643	**0.339**	**1.853**	16.772	21.503	**3.462**
BR	0.006	**0.017**	0.045	0.041	0.170	0.477	1.297	1.605	18.280	89.277	48.485	205.449
CC	0.007	0.024	0.041	0.036	0.172	0.477	1.325	2.143	30.853	92.732	50.062	223.149
BCC	0.007	0.032	0.055	0.041	0.180	0.482	1.265	2.278	27.654	74.457	54.983	174.232
PACC	0.009	0.036	0.052	0.042	0.239	0.490	1.246	2.104	29.328	98.745	29.034	195.372
PACC-LDF	0.007	0.031	0.043	0.020	**0.030**	0.140	**0.373**	0.759	3.024	27.121	**15.845**	73.235

and Macro-F1. However, there was no significant difference between PACC-LDF and other methods in Accuracy and Micro-F1.

Table 6 summarizes the Learning and Prediction time of eight comparing methods. Over all the methods, ML*k*NN needed the least training time due to its lazy strategy, while HOMER cost the least time in the prediction phase as it has sublinear time complexity with respect to the number of labels. RA*k*EL consumed the largest training time in all datasets except for the Medical dataset in spite that it employs the simple decision tree as its baseline classifier. The high complexity of RA*k*EL probably arises from its ensemble strategy and the LC models for modeling label correlations. For the CC-based methods, significant reduction of both learning and prediction time can be observed by employing LDF. Indeed, on average 60% of features were removed in two balanced datasets, Scene and Emotions, while at least 80% of features were eliminated in the other datasets, leading to a remarkable reduction in time complexity. However, PACC-LDF consumed more time in Corel16k1 and Corel5k than CC-based method. It is probably because feature selection dominates the time complexity in these two datasets. In total, PACC-LDF is one of good choices for MLC when the exact matching is expected and less executing time is demanding.

7.4.3 Results of feature selection

From Fig. 11, we can confirm the effectiveness of the proposed Label-Dependent Feature (LDF) selection approach. In terms of Exact-Match, the performances of CC-based methods have been significantly improved in most of datasets, especially in the large-scale datasets. For example, in the Corel5k dataset,

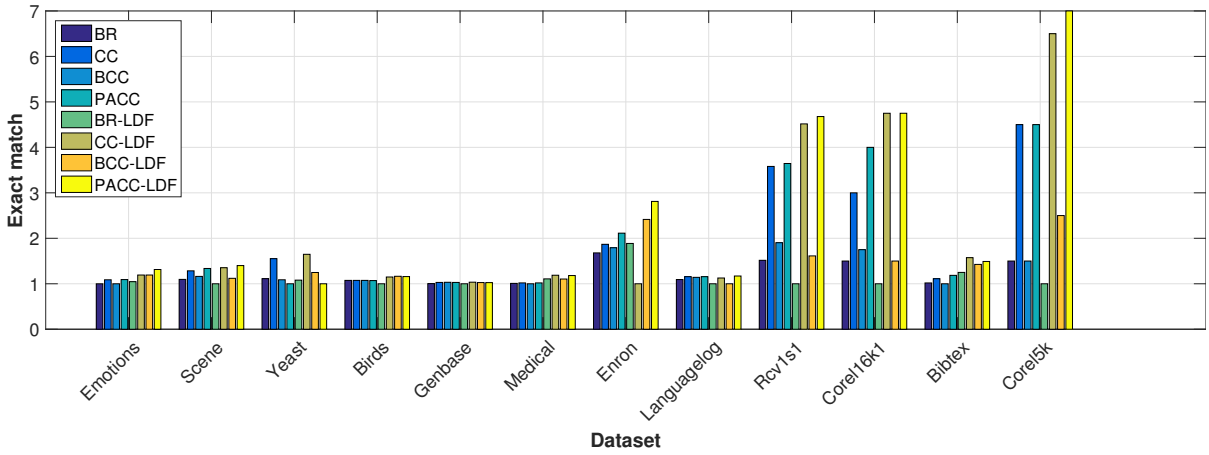


Fig. 11: Comparison of CC-based methods with their LDF variants in Exact-Match. For each dataset, the values in Exact-Match have been normalized by dividing the lowest value in the dataset.

Table 7: Wilcoxon signed-ranks test with significance level $\alpha = 0.05$ for CC-based methods against their LDF variants in terms of four evaluation metrics (p_α-values are shown in brackets). A "win" denotes the existence of a significant difference.

Comparing methods	Exact-Match	Accuracy	Macro-F1	Micro-F1
BR-LDF vs. BR	**win**[7.7e-3]	**tie**[2.3e-1]	**tie**[4.7e-2]	**tie**[7.4e-2]
CC-LDF vs. CC	**win**[7.7e-3]	**win**[7.7e-3]	**tie**[4.7e-2]	**win**[7.7e-3]
BCC-LDF vs. BCC	**win**[7.7e-3]	**tie**[4.3e-2]	**tie**[2.6e-1]	**tie**[1.7e-1]
PACC-LDF vs. PACC	**win**[7.7e-3]	**win**[7.7e-3]	**tie**[9.7e-2]	**tie**[1.5e-2]

PACC-LDF works more than 40% better than PACC, and even 4 times better than BR, demonstrating the performance superiority of selecting label-dependent features for such a large-scale dataset. According to Fig. 11, CC-based methods with LDF achieve 9.4% performance improvement on average in Exact-Match, compared with the original methods. The effectiveness of LDF is also confirmed by Table 7, where the results of Wilcoxon signed-ranks test [10] are shown. The Wilcoxon signed-ranks test was conducted sixteen times, each time on one CC-based method with its LDF counterpart. According the results of Wilcoxon test, all the LDF variants outperform the original methods in Exact-Match, and obtain comparable results in other evaluation metrics.

In addition, to demonstrate the effectiveness of the proposed LDF, also meaning the feature selection algorithm for LDF, we compared LDF with three feature selection approaches, Gain Ratio (GR) [15], ReliefF [16] and Wrapper [18], on the Emotions and Medical datasets. As a classifier, PACC with ℓ_2 regularied logistic regression was chosen. In these feature selection algorithms, backward greedy stepwise search is applied to find the relevant features for each label individually. In order to reduce the time cost of Wrapper, top 50% (emotions) and 10% (medical) relevant features were selected by a filter algorithm [40], before applying the wrapper algorithm [18]. The percentage of features is increased from 0.05 to 0.5 by step 0.05. Fig. 12 shows the experimental results on the two datasets in terms of four evaluation metrics. As shown in Fig. 12, LDF consistently works better than the other algorithms. Wrapper is the second best algorithm, and is competitive with LDF in Macro-F1. ReliefF performs better than Gain Ratio (GR) in most of cases, and can even be comparable with LDF and Wrapper in some cases, but it is sensitive to the number of selected features. In terms of time complexity, ReliefF, GR and LDF have the similar time cost, while Wrapper needs more than hundreds of execution time than the other algorithms.

8 Conclusion and future work

In this paper we have proposed polytree-augmented classifier chains with label-dependent features in order to achieve a better classification accuracy and lower computational cost compared with other popular MLC methods. As verified by the experimental results, the proposed PACC method outperformed other CC-based methods in Exact-Match. In addition, the two-stage label-dependent feature selection approach, LDF, contributed to the improvement of performance and reduction of executing time for PACC and other CC-based methods. In the future work, we consider to conduct dimension reduction in the label space, in order to further decrease the computational complexity of MLC methods and improve their scalability for large-scale datasets.

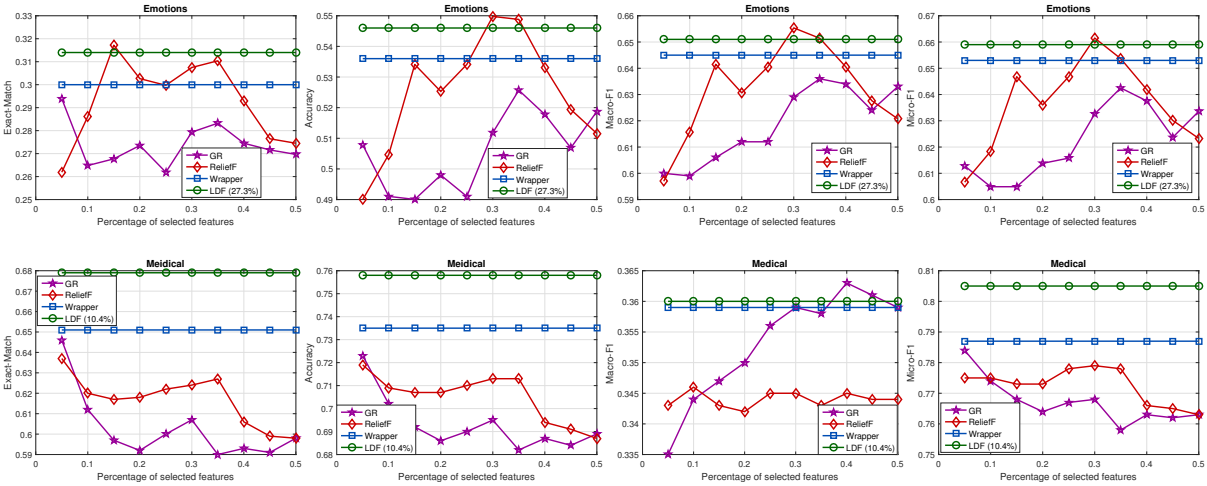


Fig. 12: Comparison of LDF with three conventional feature selection algorithms on the Emotions (the top row) and Medical (the bottom row) datasets. The percentage of selected features is increased from 0.05 to 0.5 by step 0.05. Note that Wrapper and LDF are independent of the percentage of features because Wrapper selects the feature subset which leads to the best performance, and LDF determines the number of label-specific features (on average, 27.3% for Emotions, 10.4% for Medical) by (19).

References

1. Aoki K, Kudo M (2002) Decision tree using class-dependent feature subsets. Structural, Syntactic, and Statistical Pattern Recognition 2396:761–769
2. Bhatia K, Jain H, Kar P, Varma M, Jain P (2015) Sparse local embeddings for extreme multi-label classification. In: Advances in Neural Information Processing Systems 28, pp 730–738
3. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recognition 37(9):1757–1771
4. Brown G, Pocock A, Zhao MJ, Luján M (2012) Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. Journal of Machine Learning Research 13(Jan):27–66
5. Charte F, Rivera A, del Jesus M, Herrera F (2015) Addressing imbalance in multilabel classification: Measures and random resampling algorithms. Neurocomputing 163:3–16
6. Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14(3):462–467
7. Dembczynski K, Cheng W, Hüllermeier E (2010) Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th International Conference on Machine Learning, pp 279–286
8. Dembczynski K, Waegeman W, Cheng W, Hllermeier E (2012) On label dependence and loss minimization in multi-label classification. Machine Learning 88(1-2):5–45
9. Dembczyński K, Waegeman W, Hüllermeier E (2012) An analysis of chaining in multi-label classification. In: Proceedings of the 2012 European Conference on Artificial Intelligence, IOS Press, vol 242, pp 294–299
10. Demšar J (2006) Statistical comparisons of classifiers over multiple sata sets. The Journal of Machine Learning Research 7:1–30
11. Fürnkranz J, Hüllermeier E, Mencia E, Brinker K (2008) Multilabel classification via calibrated label ranking. Machine Learning 73(2):133–153
12. Hall M (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 17th International Conference on Machine Learning, pp 359–366
13. Huang J, Li G, Huang Q, Wu X (2016) Learning label-specific features and class-dependent labels for multi-label classification. IEEE Transactions on Knowledge and Data Engineering 28(12):3309–3323
14. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp 338–345
15. Karegowda AG, Manjunath A, Jayaram M (2010) Comparative study of attribute selection using gain ratio and correlation based feature selection. International Journal of Information Technology and Knowledge Management 2(2):271–277

16. Kira K, Rendell LA (1992) The feature selection problem: Traditional methods and a new algorithm. In: AAAI, vol 2, pp 129–134
17. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97(12):273–324
18. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial intelligence 97(1-2):273–324
19. Kudo M, Sklansky J (1998) Classifier-independent feature selection for two-stage feature selection. Advances in Pattern Recognition 1451:548–554
20. Kullback S, Leibler R (1951) On information and sufficiency. The Annals of Mathematical Statistics 22(1):79–86
21. Kumar A, Vembu S, Menon AK, Elkan C (2012) Learning and inference in probabilistic classifier chains with beam search. In: Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I, Springer-Verlag, Berlin, Heidelberg, ECML PKDD'12, pp 665–680
22. Lee J, Kim DW (2015) Fast multi-label feature selection based on information-theoretic feature ranking. Pattern Recognition 48(9):2761–2771
23. Liu D, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. Mathematical Programming 45(1-3):503–528
24. Liu H, Motoda H (1998) Feature selection for knowledge discovery and data mining. Kluwer Academic Publishers, Norwell, MA, USA
25. Nemenyi P (1963) Distribution-free multiple comparisons. PhD thesis, Princeton University, New Jersey, USA
26. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
27. Prabhu Y, Varma M (2014) Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 263–272
28. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Machine Learning 85(3):333–359
29. Read J, Martino L, Luengo D (2014) Efficient monte carlo methods for multi-dimensional learning with classifier chains. Pattern Recognition 47(3):1535–1546
30. Read J, Martino L, Luengo D (2014) Efficient monte carlo methods for multi-dimensional learning with classifier chains. Pattern Recognition 47(3):1535 – 1546, handwriting Recognition and other {PR} Applications
31. Rebane G, Pearl J (1987) The recovery of causal polytrees from statistical data. In: Proceedings of the 3rd Conference on Uncertainty in Artificial Intelligence, pp 222–228
32. Sun L, Kudo M (2015) Polytree-augmented classifier chains for multi-label classification. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp 3834–3840
33. Tang L, Rajan S, Narayanan V (2009) Large scale multi-label classification via metalabeler. In: Proceedings of the 18th International Conference on World Wide Web, pp 211–220
34. Toms J, Spolar N, Cherman E, Monard M (2014) A framework to generate synthetic multi-label datasets. Electronic Notes in Theoretical Computer Science 302:155–176
35. Tsoumakas G, Katakis I (2007) Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3:1–13
36. Tsoumakas G, Katakis I, Vlahavas I (2008) Effective and efficient multilabel classification in domains with large number of labels. In: Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data
37. Tsoumakas G, Katakis I, Vlahavas L (2011) Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering 23(7):1079–1089
38. Weston J, Bengio S, Usunier N (2011) Wsabie: Scaling up to large vocabulary image annotation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp 2764–2770
39. Weston J, Makadia A, Yee H (2013) Label partitioning for sublinear ranking. In: Proceedings of the 30th International Conference on Machine Learning, pp 181–189
40. Yang Y, Pedersen J (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning, pp 412–420
41. Yu H, Jain P, Kar P, Dhillon IS (2014) Large-scale multi-label learning with missing labels. In: Proceedings of the 31st International Conference on Machine Learning, pp 593–601
42. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning, pp 856–863

43. Zaragoza J, Sucar L, Morales E, Bielza C, naga PL (2011) Bayesian chain classifiers for multidimensional classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp 2192–2197
44. Zhang M, Wu L (2015) Lift: multi-label learning with label-specific features. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(1):107–120
45. Zhang M, Zhang K (2010) Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 999–1008
46. Zhang M, Zhou Z (2007) Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40:2038–2048
47. Zhang M, Pea J, Robles V (2009) Feature selection for multi-label naive bayes classification. Information Sciences 179(19):3218–3229
48. Zhang Y, Zhou ZH (2010) Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data 4(3):14:1–14:21