Polytree-Augmented Classifier Chains for Multi-Label Classification

Lu Sun and **Mineichi Kudo**
Graduate School of Information Science and Technology
Hokkaido University, Sapporo, Japan
{sunlu, mine}@main.ist.hokudai.ac.jp

Abstract

Multi-label classification is a challenging and appealing supervised learning problem where a subset of labels, rather than a single label seen in traditional classification problems, is assigned to a single test instance. Classifier chains based methods are a promising strategy to tackle multi-label classification problems as they model label correlations at acceptable complexity. However, these methods are difficult to approximate the underlying dependency in the label space, and suffer from the problems of poorly ordered chain and error propagation. In this paper, we propose a novel polytree-augmented classifier chains method to remedy these problems. A polytree is used to model reasonable conditional dependence between labels over attributes, under which the directional relationship between labels within causal basins could be appropriately determined. In addition, based on the max-sum algorithm, exact inference would be performed on polytrees at reasonable cost, preventing from error propagation. The experiments performed on both artificial and benchmark multi-label data sets demonstrated that the proposed method is competitive with the state-of-the-art multi-label classification methods.

1 Introduction

Unlike traditional single label classification problems where an instance is associated with a single-label, multi-label classification (MLC) attempts to allocate multiple labels to any input unseen instance by a multi-label classifier learned from a training set. Obviously, such a generalization greatly raises the difficulty of obtaining a desirable prediction accuracy at a tractable complexity. Nowadays, MLC has drawn a lot of attentions in a wide range of real world applications, such as text categorization, semantic image classification, music emotions detection and bioinformatics analysis. Fig. 1[1] shows a multi-label example, where Fig. 1(a) belongs to labels "fish" and "sea" and Fig. 1(b) is assigned with labels "windsock" and "sky." Note that both objects are *fish*, but the contexts (backgrounds), in other words, the label dependencies, distinguish them clearly.

(a) "fish","sea" (b) "windsock","sky"

Figure 1: A multi-label example to show label dependency for inference. (a) A fish in the sea; (b) carp shaped windsocks (koinobori) in the sky.

The existing MLC methods can be cast into two strategies: problem transformation and algorithm adaptation [Tsoumakas and Katakis, 2007]. A convenient and straightforward way for MLC is to conduct problem transformation in which a MLC problem is transformed into one or more single label classification problems. Problem transformation mainly comprises three kinds of methods: binary relevance (BR) [Zhang and Zhou, 2007], pairwise (PW) [Fürnkranz *et al.*, 2008] and label combination (LC) [Tsoumakas *et al.*, 2011]. BR simply trains a binary classifier for each label, ignoring the label correlation, which is apprantly crucial to make accurate prediction for MLC. For example, it is quite difficult to distinguish the labels "fish" and "windsock" in Fig. 1 from the visual features, unless we consider the label correlations with "sea" and "sky." PW and LC are designed to capture label dependence directly. PW learns a classifier for each pair of labels, and LC transforms MLC into the possible largest single-label classification problem by treating each possible label combination as a meta-label. At the expense of their high ability on modeling label correlations, the complexity increases quadratically and exponentially with the number of labels for PW and LC, respectively, thus they typically become impracticable even for a small number of labels.

In the second strategy, algorithm adaptation, multi-label problems are solved by modifying conventional machine learning algorithms, such as support vector machines [Elisseeff and Weston, 2001], k-nearest neighbors [Zhang and

[1] http://www.yunphoto.net

Zhou, 2007], adaboost [Schapire and Singer, 2000], neural networks [Zhang and Zhou, 2006], decision trees [Comite *et al.*, 2003] and probabilistic graphical models [Ghamrawi and McCallum, 2005; Qi *et al.*, 2007; Guo and Gu, 2011; Alessandro *et al.*, 2013]. They achieved competitive performances to those of problem transformation based methods. However, they have several limitations, such as difficulty of choosing parameters, high complexity on the prediction phase, and sensitivity on the statistics of data.

Recently, a BR based MLC method, namely classifier chains (CC) [Read *et al.*, 2009], is proposed to overcome the intrinsic drawback of BR and to achieve higher predictive accuracy. It succeeded in modeling label correlations at low computational complexity, and produced furthermore two variants: probabilistic classifier chains (PCC) [Dembczynski *et al.*, 2010] and Bayesian classifier chains (BCC) [Zaragoza *et al.*, 2011]. PCC makes an attempt to avoid the problem of error propagation that was possessed by CC, however, its ability is greatly limited by the number of labels due to its high prediction cost resulting from the exhaustive search manner. Benefiting from mining marginal dependence between labels as a directed tree, BCC can establish classifier chains in particular chain orderings, but its performance is limited due to the modeling with only second-order correlations of labels. Moreover, these CC based methods cannot model label correlations in a reasonable way, since CC and PCC randomly establish fully connected Bayesian networks, which have a possible risk to lead to trivial label dependence. In addition, the expression ability of BCC is strongly restricted by the adopted tree structure.

To overcome these limitations of CC based methods, this paper proposes a novel polytree-augmented classifier chains (PACC). In contrast to CC and PCC, PACC seeks a reasonable ordering by a polytree that represents the underlying dependence among labels. Unlike BCC, PACC is derived from conditional dependence between labels and is able to model both low and high-order label correlations by virtue of polytrees. Polytree structure is ubiquitous in real world MLC problems, for example, we can readily observe such a structure on the popular Wikipedia[2] and Gene Ontology[3] data sets. In addition, the polytree structure makes it possible to perform exact inference in reasonable time, avoiding the problem of error propagation. Moreover, by introduction of causal basins, the directionality between labels is mostly determined, which further decreases the number of possible orderings.

The rest of this paper is organized as follows. Section 2 gives the mathematical definition of MLC. Section 3 presents related works of CC based methods. Section 4 illustrates an overview of PACC, and presents its implementation. Section 5 reports experimental results performed on both artificial and benchmark multi-label data sets. Finally, Section 6 concludes this paper and gives discussions for the future work.

2 Multi-label classification

In the scenario of MLC, given a finite set of labels $\mathcal{L} = \{\lambda_1, ..., \lambda_d\}$, 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 $\mathbf{y} = (y_1, ..., y_d)$ drawn from the output label space $\mathcal{Y} = \{0, 1\}^d$, where $y_j = 1$ if and only if label λ_j is associated with instance $\mathbf{x}$, and 0 otherwise. In other words, $\mathbf{y} = (y_1, ..., y_d)$ can also be viewed as a realization of corresponding random vector $\mathbf{Y} = (Y_1, ..., Y_d)$, $\mathbf{Y} \in \mathcal{Y}$.

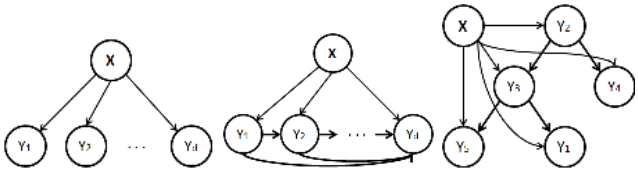
Assume that we are given a data set of N instances $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$, where $\mathbf{y}^{(i)}$ is the label assignment of the ith instance. The task of MLC is to find an optimal classifier $h : \mathcal{X} \to \mathcal{Y}$ which assigns a label vector $\mathbf{y}$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

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

In the BR context, a classifier h is comprised of d binary classifiers $h_1, ..., h_d$, where each h_j predicts $\hat{y}_j \in \{0, 1\}$, forming a vector $\hat{\mathbf{y}} \in \{0, 1\}^d$. Fig. 2(a) shows the probabilistic graphical model of BR. Here, the model

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{d} P(Y_j|\mathbf{X}) \quad (2)$$

means that the class labels are mututally independent.



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

Figure 2: Probabilistic graphical models of BR and CC based methods.

3 Related works

3.1 Classifier chains (CC)

Classifier chains (CC) [Read *et al.*, 2009] models label correlations in a randomly ordered chain based on (3).

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{d} P(Y_j|\mathbf{pa}(Y_j), \mathbf{X}). \quad (3)$$

Here $\mathbf{pa}(Y_j)$ represents the parent labels for Y_j. Obviously, $|\mathbf{pa}(Y_j)| = p$, where p denotes the number of labels prior to Y_j following the chain order.

In the training phase, according to a predefined chain order, it bulids d binary classifiers $h_1, ..., h_d$ such that each classifier predicts correctly the value of y_j by referring to $\mathbf{pa}(y_j)$ in addition to $\mathbf{x}$. In the testing phase, it predicts the value of y_j in a greedy manner:

$$\hat{y}_j = \arg\max_{y_j} P(y_j|\hat{\mathbf{pa}}(y_j), \mathbf{x}),\ j = 1, ..., d. \quad (4)$$

[2]https://www.kaggle.com/c/lshtc

[3]http://geneontology.org/

The final prediction is the collection of the results, $\hat{\mathbf{y}}$. The computational complexity of CC is $O(d \times T(m, N))$, where $T(m, N)$ is the complexity of constructing a learner for m attributes and N instances. Its complexity is identical with BR's, if a linear baseline learner is used. Fig. 2(b) shows the probabilistic graphical model of CC following the order of $Y_1 \rightarrow Y_2 \rightarrow \cdots \rightarrow Y_d$.

CC suffers from two risks: if the chain is wrongly ordered in the training phase, then the prediction accuracy can be degraded, and if the previous prediction of labels was wrong in the testing phase, then such a mistake can be propagated to the succeeding prediction.

3.2 Probabilistic classifier chains (PCC)

In the light of risk minimization and Bayes optimal prediction, probabilistic classifier chains (PCC) [Dembczynski *et al.*, 2010] is proposed. PCC approximates the joint distribution of labels, providing better estimates than CC at the cost of higher computational complexity.

The conditional probability of the label vector $\mathbf{Y}$ given the feature vector $\mathbf{X}$ is the same as CC. Accordingly, PCC shares the model (3) and Fig. 2(b) with CC.

Unlike CC which predicts the output in a greedy manner by (4), PCC examines all the 2^d paths in an exhaustive manner:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}). \tag{5}$$

PCC is a better method in accuracy, but the exponential cost limits its application even for a moderate number of labels, typically no more than 15.

3.3 Bayesian classifier chains (BCC)

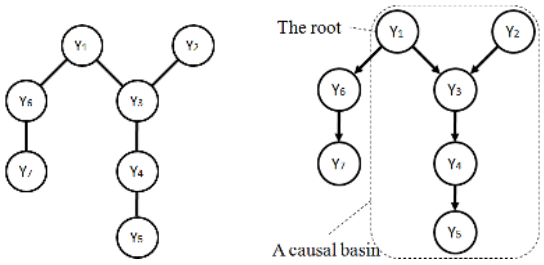
BCC [Zaragoza *et al.*, 2011] introduces a Bayesian network to find a reasonable connection between labels before building classifier chains. Specifically, BCC constructs a Bayesian network by deriving a maximum-cost spanning tree from marginal label dependence.

The learning phase of BCC consists of two stages: learning of a tree-structured Bayesian network and building of d binary classifiers following a chain. The chain is determined by the directed tree, which is established by randomly choosing a label as its root and by assigning directions to the remaining edges.

BCC also shares the same model (3) with CC and PCC. Note that because of the tree structure, $|\mathbf{pa}(Y_j)| \leq 1$ in BCC unlike $|\mathbf{pa}(Y_j)| = p$ in CC and PCC, limiting its ability on modeling label dependence. Fig. 2(c) shows an example of the probabilistic graphical model of BCC with five labels.

4 Polytree-augmented classifier chains (PACC)

We propose a novel polytree-augmented classifier chains (PACC) based on the polytree and also sharing the model (3) with CC based methods. A *polytree* (Fig. 3(b)) is a singly connected causal network in which variables may arise from multiple causes [Rebane and Pearl, 1987], i.e., a node can have multiple parents unlike BCC. A *causal basin* 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. Rebane and Pearl [Rebane and Pearl, 1987] demonstrated that a polytree is a directed acyclic graph whose underlying skeleton is an undirected tree (Fig. 3(a)).



(a) A tree skeleton (b) A polytree

Figure 3: Example of polytree with its latent skeleton.

4.1 Learning of the skeleton

In PACC, modeling of conditional label dependence is made by approximating the true distribution $P(\mathbf{Y}|\mathbf{X})$ by another distribution. Therefore, Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951], a measure of distance between two distributions, is used to evaluate how close an alternative distribution $P_B(\mathbf{Y}|\mathbf{X})$ is to $P(\mathbf{Y}|\mathbf{X})$, where B is a Bayesian network:

$$\begin{aligned} &\min_B D_{KL}(P(\mathbf{Y}|\mathbf{X})||P_B(\mathbf{Y}|\mathbf{X})) \\ &= \max_B \mathbb{E}_{P(\mathbf{x},\mathbf{y})}(\log P_B(\mathbf{Y}|\mathbf{X}) - \log P(\mathbf{Y}|\mathbf{X})). \end{aligned} \tag{6}$$

Using the empirical distribution $\hat{P}_D(\mathbf{Y}|\mathbf{X})$ instead of $P(\mathbf{Y}|\mathbf{X})$, we evaluate above as:

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

which is maximized if and only if $P_B(\cdot) = \hat{P}_D(\cdot)$,

$$= \max_B \sum_{j=1}^{d} I_{\hat{P}_D}(Y_j; \mathbf{pa}(Y_j)|\mathbf{X}), \tag{7}$$

where $I_{\hat{P}_D}(Y_i; \mathbf{pa}(Y_j)|\mathbf{X})$ is the conditional mutual information between Y_j and its parents $\mathbf{pa}(Y_j)$ over $\mathbf{X}$ in B:

$$\mathbb{E}_{\hat{P}_D(\mathbf{x},y_j,\mathbf{pa}(y_j))} \log \frac{\hat{P_D}(Y_j, \mathbf{pa}(Y_j)|\mathbf{X})}{\hat{P}_D(Y_j|\mathbf{X})\hat{P}_D(\mathbf{pa}(Y_j)|\mathbf{X})}. \tag{8}$$

As a result, the optimal B^* is obtained by maximizing $\sum_{j=1}^{d} I_{\hat{P}_D}(Y_j; \mathbf{pa}(Y_j)|\mathbf{X})$. Since learning of B^* is NP-hard in general, we restrict our hypothesis B to satisfy $|\mathbf{pa}(Y_j)| \leq 1$, indicating the tree skeleton is to be built. In practice, we carry out Chou-liu's algorithm [Chow and Liu, 1968] to obtain the maximum-cost spanning tree (Fig. 3(a)) with edge weights $I_{\hat{P}_D}(Y_i; \mathbf{pa}(Y_j)|\mathbf{X})$.

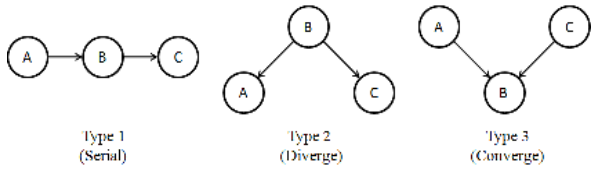


Figure 4: Three basic types of adjacent triplets A, B, C.

However, it is quite difficult to get conditional probability $\hat{P}_D(\mathbf{Y}|\mathbf{X})$, especially when $\mathbf{X}$ is continuous. Recently some methods [Dembczynski *et al.*, 2012; Zaragoza *et al.*, 2011; Zhang and Zhang, 2010] have been proposed to solve this problem. In BCC [Zaragoza *et al.*, 2011], as an approximation of conditional probability, marginal probability of labels $\mathbf{Y}$ is obtained by simply counting the frequency of occurrence. Similar with [Dembczynski *et al.*, 2012], LEAD [Zhang and Zhang, 2010] directly obtains conditional dependence by estimating the dependence of errors in multivariate regression models.

In this paper, we use a generalized approach to estimate conditional probability. The data set D is splitted into two sets: a training set D_t and a hold-out set D_h. Probabilistic classifiers are learned from D_t to represent conditional probability of labels, and the probability is calculated based on the output of the learned classifiers over D_h. Assuming $\mathbf{pa}(Y_j) = Y_k$, we can rewrite (8) as follows:

$$\mathbb{E}_{\hat{P}_D(\mathbf{x})}\mathbb{E}_{\hat{P}_D(y_j|y_k,\mathbf{x})}\mathbb{E}_{\hat{P}_D(y_k|\mathbf{x})}\log\frac{\hat{P}_D(Y_j|Y_k,\mathbf{X})}{\hat{P}_D(Y_j|\mathbf{X})},$$

assuming an equal weight of every instance in D_h,

$$=\frac{1}{|D_h|}\sum_{\mathbf{x}}\mathbb{E}_{\hat{P}_D(y_j|y_k,\mathbf{x})}\mathbb{E}_{\hat{P}_D(y_k|\mathbf{x})}\log\frac{\hat{P}_D(Y_j|Y_k,\mathbf{X})}{\hat{P}_D(Y_j|\mathbf{X})}. \quad (9)$$

Hence we can train probabilistic classifiers h_j, h_k and $h_{j|k}$ on D_t and compute $\hat{P}_D(y_j|\mathbf{x})$, $\hat{P}_D(y_k|\mathbf{x})$ and $\hat{P}_D(y_j|y_k,\mathbf{x})$ by utilizing the classifiers to make prediction on D_h, respectively. Lastly, $I_{\hat{P}_D}(Y_j;Y_k|\mathbf{X})$ is estimated according to (9).

4.2 Constructing of the PACC

First we assign directions to the skeleton by finding causal basins. This is implemented by finding multi-parent labels and the corresponding directionality. The detailed procedure is as follows.

Fig. 4 shows three possible graphical models among 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

$$I(A,C) = \mathbb{E}_{P(a,c)}\log\frac{P(A,C)}{P(A)P(C)} = 0. \quad (10)$$

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. By performing the zero-MI testing for every pair of Y_j's direct neighbors, $\mathbf{pa}(Y_j)$ would be determined. We can see that although PACC shares the model (3) with other CC based methods, it holds $|\mathbf{pa}(Y_j)| \leq p$, demonstrating the flexibility on modeling label dependencies.

In order to build a classifier chain by the learned polytree, 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. Hence, learning of a label is not performed until the labels with higher ranks, including its parents, have been learned. In this paper, we choose logistic regression with L_2 regularization as the baseline classifier. Therefore, d logistic regression classifiers are learned, each of which is trained by treating $\mathbf{pa}(y_j)$ and $\mathbf{x}$ as new attributes, shown as follows:

$$P(y_j = 1|\mathbf{pa}(y_j),\mathbf{x},\boldsymbol{\theta}_j) = \frac{1}{1+e^{-\boldsymbol{\theta}_j^T(\mathbf{x},\mathbf{pa}(y_j))}} \quad (11)$$

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

$$\max_{\boldsymbol{\theta}_j}\sum_{i=1}^{N}\log P(y_j^{(i)}|\mathbf{pa}(y_j)^{(i)},\mathbf{x}^{(i)},\boldsymbol{\theta}_j) - \frac{\lambda}{2}\|\boldsymbol{\theta}_j\|_2^2, \quad (12)$$

where λ is a trade-off coefficient to avoid overfitting by generating sparse parameters $\boldsymbol{\theta}$. Then traditional convex optimization techniques can be used to learn the parameters.

4.3 Classification

Exact inference, as (5), is NP-hard in directed acyclic graphs, for instance, the computation cost of PCC on prediction increases exponentially in number of labels, with complexity $O(2^d)$. Thanks to the max-sum algorithm [Pearl, 1988], despite the fact that the complexity of exact inference on the polytree is $O(2^d)$ in the worst case ($d-1$ roots with 1 common leaf), exact inference can be performed in reasonable time on polytrees by bounding the indegree of nodes.

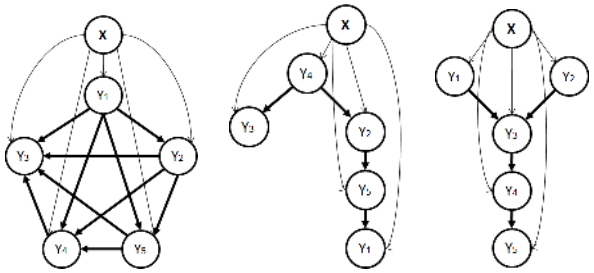
Two phases are performed to make the exact inference. The first phase begins at the roots and propagates downwards to the leaves: the conditional probability table for each node is calculated based on its local graphical structure. In the second phase, message propagation starts upwards from the leaves to the roots, with each node Y_j collecting all the incoming messages and finding the local maximum with its value $\hat{y}_j$. In this way, the *Maximum a Posteriori* (MAP) assignment $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_d)$ is obtained according to

$$\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \max_{y_r}\Big[P(y_r|\mathbf{x})\big[\cdots\max_{y_l}P(y_l|\mathbf{pa}(y_l),\mathbf{x})\big]\Big], \quad (13)$$

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

5 Experiments

We compared the proposed PACC with seven state-of-the-art MLC methods, including BR, CC, PCC, BCC, multi-label k-nearest neighbours (MLkNN) [Zhang and Zhou, 2006], calibrated label ranking (CLR) [Fürnkranz *et al.*, 2008] and conditional dependency network (CDN) [Guo and Gu, 2011].



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

Figure 5: Graphical models built by CC based methods.

The methods were implemented based on Weka[4], Mulan[5] and Meka[6], and performed on one artificial data set and twelve benchmark data sets[5,6] including seven regular and five large data sets. For calculation of the accuracy, 10-fold and 3-fold cross validation were used for the regular and large data sets, respectively. In the experiments we chose logistic regression with L_2 regularization as the baseline classifier, set the number of neighbors to $k = 10$ in MLkNN, and used 100 iterations as the burn-in time for CDN. To reduce the training cost, marginal probability, instead of conditional probability, was calculated in PACC for the large data sets.

For evaluation, two popular multi-label metrics were used.

1. *Global accuracy* (accuracy per instance):

$$Acc = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\hat{\mathbf{y}}^{(i)}=\mathbf{y}^{(i)}}, \tag{14}$$

where $\mathbb{1}_{(\cdot)}$ represents the indicator function.

2. *Macro-averaged F-measure* (F measure averaged over labels)

$$F_{macro} = \frac{1}{d}\sum_{j=1}^{d} \frac{2\sum_{i=1}^{N} \mathbb{1}_{\hat{y}_j^{(i)}=y_j^{(i)}}}{\sum_{i=1}^{N}\hat{y}_j^{(i)} + \sum_{i=1}^{N} y_j^{(i)}}. \tag{15}$$

5.1 Artificial data

Here we focused on the performances of BR and CC based methods to evaluate their ability on modeling label correlations. One artificial data set with two features $\mathbf{X} = (X_1, X_2)$ and five labels $\mathbf{Y} = (Y_1, Y_2, Y_3, Y_4, Y_5)$ was introduced. The feature variables were uniformly sampled in a square, i.e., $\mathbf{x} \in [-0.5, 0.5]^2$, and the label dependency was given by $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}}(y_1)P_{\mathbf{x}}(y_2)P_{\mathbf{x}}(y_3|y_1, y_2)P_{\mathbf{x}}(y_4|y_3)P_{\mathbf{x}}(y_5|y_4)$. We defined $P_{\mathbf{x}}(y_j) = 1/(1+\exp(-f_j(\mathbf{x})))$, where $f_j(\mathbf{x})$ was linear functions given as: $f_1(\mathbf{x}) = x_1 + x_2$, $f_2(\mathbf{x}) = x_1 - x_2$, $f_3(\mathbf{x}) = -x_1 - x_2 - 6y_1 + 6y_2$, $f_4(\mathbf{x}) = -x_1 + x_2 - 4y_3 + 2$ and $f_5(\mathbf{x}) = -2x_1 + x_2 + 2y_4 - 1$. According to this distribution, $10,000$ instances were generated.

Table 1 reports the experimental results of the artificial data. The best accuracy is shown in bold face. Fig. 5 shows the learned graphical models, from which we can see that only PACC successfully found the latent dependence (Fig. 5(c)). As shown in Fig. 5(a), CC and PCC seem to have modeled many trivial correlations. PACC outperformed the other methods in terms of global accuracy. In the light of exact inference, PCC performed better than CC although they share the same network. BCC modeled a wrong dependency network (Fig. 5(b)) over labels based on marginal dependence. BR was the worst since it totally ignores label correlations. Nevertheless, BR outperformed the others in macro-averaged F measure.

Table 1: Experimental results on the artificial data set.

	BR	CC	PCC	BCC	PACC
Acc	.210 (5)	.256 (3)	.262 (2)	.251 (4)	**.265 (1)**
F_{macro}	**.566 (1)**	.554 (4.5)	.554 (4.5)	.558 (2)	.556 (3)
Rank	3 (2.5)	3.8 (5)	3.3 (4)	3 (2.5)	**2 (1)**

5.2 Benchmark data

Next, all the eight methods were compared on a collection of twelve multi-label data sets whose statistical properties are given in Table 2. The experimental results are reported in Tables 3 and 4. Note that PCC and CLR could not finish for large data sets due to their exponential and quadratic complexity in number of labels, respectively.

Table 2: The statistics of data sets. I: Instances, A: Attributes, L: Labels, C: Cardinality, D: Domains.

Data	# I	# A	# L	C	D
Scene	2407	294	6	1.07	image
Emotions	593	72	6	1.87	music
Flags	194	19	7	3.39	image
Yeast	2417	103	14	4.24	biology
Birds	645	260	19	1.01	audio
Genbase	662	1186	27	1.25	biology
Medical	978	1449	45	1.25	text
Enron	1702	1001	53	3.38	text
Language log	1460	1004	75	1.18	text
Mediamill	43907	120	101	4.38	video
Bibtex	7395	1836	159	2.40	text
Corel5k	5000	499	374	3.52	music

In the global accuracy, PACC was the best or comparable with the best methods on data sets except the Yeast, Birds and Language log sets. A possible explanation is that PACC is a global accuracy risk minimizer benefiting from its ability on modeling both low-order and high-order label correlations based on the polytree structure. PCC is comparable with PACC on all the data sets when it finished. This means that PCC also makes exact inference as PACC. BCC works worse than CC, especially on large data sets. This is probably because it models only second-order correlations. In consistent with our theoretical analysis, BR obtains the worst result compared with CC based methods, since it simply ignores label correlations. It is also worth noting that BR works better than CC based methods on Birds, indicating weak label cor-

[4]http://www.cs.waikato.ac.nz/ml/weka/

[5]http://mulan.sourceforge.net/

[6]http://meka.sourceforge.net/

Table 3: Global accuracy of experimental results on 12 benchmark data sets.

Data	BR	CC	PCC	BCC	MLkNN	CLR	CDN	PACC
Scene	.428 (7)	.503 (4.5)	.503 (4.5)	.444 (6)	**.633 (1)**	.510 (2.5)	.238 (8)	.510 (2.5)
Emotions	.241 (5.5)	.254 (3)	.263 (2)	.241 (5.5)	.244 (4)	.213 (8)	.237 (7)	**.264 (1)**
Flags	.150 (5)	.176 (4)	**.182 (1.5)**	.181 (3)	.135 (6)	.088 (8)	.103 (7)	**.182 (1.5)**
Yeast	.144 (6)	.190 (2)	.182 (3)	.145 (5)	**.203 (1)**	.045 (7)	.033 (8)	.165 (4)
Birds	.431 (2)	.429 (3)	N/A	.380 (6)	**.445 (1)**	.129 (7)	.386 (5)	.424 (4)
Genbase	.917 (5)	.965 (2)	N/A	.964 (3)	.906 (6)	.610 (7)	.941 (4)	**.967 (1)**
Medical	.417 (5.5)	.419 (3.5)	N/A	.390 (7)	.417 (5.5)	.436 (2)	**.479 (1)**	.419 (3.5)
Enron	.045 (3.5)	**.050 (1.5)**	N/A	.045 (3.5)	.009 (7)	.021 (6)	.023 (5)	**.050 (1.5)**
Language log	.164 (4)	.158 (6)	N/A	.157 (7)	.190 (2)	**.197 (1)**	.173 (3)	.162 (5)
Mediamill	.069 (5)	.130 (2)	N/A	.099 (4)	.105 (3)	.042 (6)	.027 (7)	**.137 (1)**
Bibtex	.016 (4)	.014 (5)	N/A	.010 (6)	**.064 (1)**	N/A	.018 (3)	.020 (2)
Corel5k	.001 (4.5)	**.003 (1.5)**	N/A	.001 (4.5)	.000 (6)	N/A	.002 (3)	**.003 (1.5)**
Ave. Rank	4.8 (5)	3.2 (3)	2.8 (2)	5.1 (6.5)	3.6 (4)	5.5 (8)	5.1 (6.5)	**2.4 (1)**

Table 4: Macro-averaged F measure of experimental results on 12 benchmark data sets.

Data	BR	CC	PCC	BCC	MLkNN	CLR	CDN	PACC
Scene	.617 (3)	.605 (6.5)	.605 (6.5)	.612 (4)	**.744 (1)**	.676 (2)	.285 (8)	.610 (5)
Emotions	.628 (6)	.633 (3)	.638 (2)	.612 (7)	.629 (5)	.632 (4)	.591 (8)	**.645 (1)**
Flags	.608 (3)	.559 (7)	.600 (4)	**.617 (1)**	.563 (6)	.581 (5)	.536 (8)	.614 (2)
Yeast	.423 (2)	.395 (5.5)	.418 (4)	.392 (7)	.420 (3)	**.432 (1)**	.305 (8)	.395 (5.5)
Birds	.221 (7)	.242 (5)	N/A	.259 (3)	**.298 (1)**	.263 (2)	.232 (6)	.256 (4)
Genbase	.595 (5)	.638 (2)	N/A	.634 (3)	.549 (7)	.582 (6)	.629 (4)	**.652 (1)**
Medical	.279 (6)	.298 (5)	N/A	.302 (2)	.235 (7)	.300 (3.5)	**.303 (1)**	.300 (3.5)
Enron	.129 (5)	.146 (4)	N/A	.159 (3)	.047 (7)	**.202 (1)**	.121 (6)	.166 (2)
Language log	.038(7)	**.059 (1.5)**	N/A	.058 (3)	.056 (4)	.055 (5)	.045 (6)	**.059 (1.5)**
Mediamill	**.267 (1)**	.182 (6)	N/A	.199 (5)	.240 (2)	.237 (3)	.064 (7)	.210 (4)
Bibtex	.106 (6)	.122 (4)	N/A	.132 (2)	**.179 (1)**	N/A	.120 (5)	.123 (3)
Corel5k	.033 (4)	.036 (3)	N/A	**.038 (1)**	.025 (5)	N/A	.024 (6)	.037 (2)
Ave. Rank	4.6 (7)	4.4 (6)	4.1 (4.5)	3.4 (3)	4.1 (4.5)	3.3 (2)	6.1 (8)	**2.9 (1)**

relations in that set. Among eight methods, MLkNN ranked first over four data sets, but it fails largely in the Genbase, Enron and Corel5k sets. It probably means that MLkNN is sensitive to the number of neighbors and noise. CLR was the worst, meaning that modeling pairwise relationship between labels is not enough in practice. It seems that, to extract the best performance of CDN with fully complicated dependency of labels, we need more samples.

In macro-averaged F measure, PACC performed the best. CLR ranked at the second, showing a good performance compared with the global accuracy. BCC ranked at the third, and was comparable with other CC based methods. The performance of MLkNN varied significantly over the data sets.

In terms of computational time[7], PACC needed several minutes to hours for learning depending on the data sets size and seconds to minutes for prediction. We can say that PACC finds a better trade-off between computational complexity and prediction accuracy in contrast to other CC based methods by virtue of the polytree structure and max-sum algorithm. BR, CC, BCC paid similar prediction cost with PACC. PCC needs the same time with CC for learning, but needs $O(2^d)$ complexity for prediction, for which reason it was applicable only on smallest data sets. As a lazy method, MLkNN could finish classification within minutes even in the largest data sets, demonstrating its high efficiency. CLR consumed hours or could not complete learning, due to its quadratic complexity in the number of labels. Since CDN employs Gibbs sampling to approximate inference, it cost much more time on prediction compared with other methods, typically several minutes on large data sets.

[7]In a Intel Quad-Core CPU at 3.4 GHz with 8 GB RAM.

6 Conclusions and future work

In this paper, we have proposed a polytree-augmented classifier chains (PACC) method in order to achieve a better prediction accuracy compared with the state-of-the-art MLC methods by modeling the underlying label dependency as a polytree. The experimental results on one synthetic and twelve real data sets demonstrated its efficiency. As future work, it is interesting to see whether PACC can be further improved by incorporating feature selection into the construction step. Furthermore, introduction of ensemble methods and other baseline learners for PACC is also worth researching.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Number 15H02719 and China Scholarship Council with Hokkaido University.

References

[Alessandro *et al.*, 2013] Antonucci Alessandro, Giorgio Corani, Denis Mauá, and Sandra Gabaglio. An ensemble of bayesian networks for multilabel classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1220–1225. AAAI Press, 2013.

[Chow and Liu, 1968] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[Comite *et al.*, 2003] Francesco De Comite, Remi Gilleron, and Marc Tommasi. Learning multi-label alternating decision trees from texts and data. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM'03, pages 35–49, Berlin, Heidelberg, 2003. Springer-Verlag.

[Dembczynski *et al.*, 2010] Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In Johannes Frnkranz and Thorsten Joachims, editors, *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 279–286. Omnipress, 2010.

[Dembczynski *et al.*, 2012] Krzysztof Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

[Elisseeff and Weston, 2001] Andr Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *In Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2001.

[Fürnkranz *et al.*, 2008] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, November 2008.

[Ghamrawi and McCallum, 2005] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 195–200, New York, NY, USA, 2005. ACM.

[Guo and Gu, 2011] Yuhong Guo and Suicheng Gu. Multilabel classification using conditional dependency networks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1300–1305. AAAI Press, 2011.

[Kullback and Leibler, 1951] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03 1951.

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

[Qi *et al.*, 2007] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 17–26, New York, NY, USA, 2007. ACM.

[Read *et al.*, 2009] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, 5782:254–269, 2009.

[Rebane and Pearl, 1987] G. Rebane and J. Pearl. The recovery of causal polytrees from statistical data. In *Third Conference on Uncertainty in Artificial Intelligence*, pages 222–228, 1987.

[Schapire and Singer, 2000] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *MACHINE LEARNING*, 39:135–168, 2000.

[Tsoumakas and Katakis, 2007] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

[Tsoumakas *et al.*, 2011] G. Tsoumakas, I. Katakis, and L. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, July 2011.

[Zaragoza *et al.*, 2011] Julio H. Zaragoza, L. Enrique Sucar, Eduardo F. Morales, Concha Bielza, and Pedro Larrañaga. Bayesian chain classifiers for multidimensional classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 2192–2197. AAAI Press, 2011.

[Zhang and Zhang, 2010] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 999–1008, New York, NY, USA, 2010. ACM.

[Zhang and Zhou, 2006] Min-ling Zhang and Zhi-hua Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 18:1338–1351, 2006.

[Zhang and Zhou, 2007] Min-ling Zhang and Zhi-hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *PATTERN RECOGNITION*, 40:2007, 2007.