ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-294

An Analysis of Chaining in Multi-Label Classification

Krzysztof Dembczyński[1] and **Willem Waegeman**[2] and **Eyke Hüllermeier**[3]

Abstract. The idea of classifier chains has recently been introduced as a promising technique for multi-label classification. However, despite being intuitively appealing and showing strong performance in empirical studies, still very little is known about the main principles underlying this type of method. In this paper, we provide a detailed probabilistic analysis of classifier chains from a risk minimization perspective, thereby helping to gain a better understanding of this approach. As a main result, we clarify that the original chaining method seeks to approximate the joint mode of the conditional distribution of label vectors in a greedy manner. As a result of a theoretical regret analysis, we conclude that this approach can perform quite poorly in terms of subset 0/1 loss. Therefore, we present an enhanced inference procedure for which the worst-case regret can be upper-bounded far more tightly. In addition, we show that a probabilistic variant of chaining, which can be utilized for any loss function, becomes tractable by using Monte Carlo sampling. Finally, we present experimental results confirming the validity of our theoretical findings.

1 INTRODUCTION

Multi-label classification (MLC) differs from conventional binary classification insofar as multiple binary labels have to be predicted simultaneously. This transition from predicting a single label to predicting multiple labels raises a number of computational and statistical challenges, such as the need for modeling statistical dependencies between labels and optimizing a wide range of loss functions in a potentially high-dimensional label space.

Indeed, various types of loss function are encountered in different application domains of MLC. From a probabilistic perspective, it is clear that different properties of the joint conditional distribution over labels are needed for optimizing these loss functions. For example, it is known that the simple binary relevance classifier can perform quite well in terms of label-wise decomposable loss functions like Hamming loss, for which knowledge of the conditional marginal distribution of labels is sufficient for deriving a risk minimizing prediction. On the other hand, there are loss functions like the subset 0/1 loss and the F-measure, for which more complex properties of the joint conditional distribution are needed, and which necessitate the modeling of dependencies between labels [4].

For many advanced MLC algorithms, it is still unclear what type of loss they actually intend to minimize. For example, the recently introduced classifier chains (CC) [9, 10] has not been thoroughly analyzed from a probabilistic perspective, despite being intuitively appealing and showing strong performance in empirical studies. Its probabilistic variant (PCC), introduced in [3], explains the original CC in terms of the application of the product rule of probability. In this view, the output of the classifier chain is an estimate of the joint probability distribution, for which an inference procedure is needed in order to obtain the right prediction for a given loss. The PCC method, however, has been only analyzed with an exhaustive inference that is intractable for problems with more than 12-15 labels. The procedure used in the original CC method, in turn, can be seen as a kind of greedy inference, but little is known about its true behavior.

In this paper, which is an extended and revised version of a previous workshop presentation [5], we aim to provide a thorough probabilistic analysis of chaining methods in an attempt to unravel the true mechanisms that lead to a state-of-the-art predictive performance of this type of methods. Subsequent to a formal definition of multi-label classification in Section 2 and an introduction to chaining in Section 3, we discuss possible inference mechanisms in Section 4, starting with exhaustive search. As a solution to the computational burden of this approach, we propose to use a Monte Carlo sampling, which can be easily implemented for classifier chains. Subsequently, we show that the greedy inference method that was proposed with the original CC algorithm intends to optimize the subset 0/1 loss, but the regret in predictive performance can be high. Therefore, an enhanced approximate inference algorithm is introduced, for which substantially tighter worst-case regret bounds are derived as a function of the running time of the algorithm (assuming that conditional probabilities can be estimated perfectly). Finally, Section 6 presents extensive experimental results, showing that the theoretical analysis of this paper holds in practice.

2 MULTI-LABEL CLASSIFICATION

Let $\mathcal{X}$ denote an instance space and $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ be a finite set of class labels. An instance $\boldsymbol{x} \in \mathcal{X}$ is (non-deterministically) associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is often called the set of relevant labels, while the complement $\mathcal{L} \setminus L$ is considered as irrelevant for $\boldsymbol{x}$. We identify a set L of relevant labels with a binary vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, in which $y_i = 1$ iff $\lambda_i \in L$. By $\mathcal{Y} = \{0, 1\}^m$ we denote the set of possible labelings.

We assume observations to be generated independently and identically according to a probability distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ on $\mathcal{X} \times \mathcal{Y}$, i.e., an observation $\boldsymbol{y} = (y_1, \ldots, y_m)$ is the realization of a corresponding random vector $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$. We denote by $\mathbf{P}(\boldsymbol{y} \mid \boldsymbol{x})$ the conditional distribution of $\mathbf{Y} = \boldsymbol{y}$ given $\mathbf{X} = \boldsymbol{x}$, and by $\mathbf{P}(y_i = b|\boldsymbol{x})$ the corresponding marginal distribution of Y_i, i.e., $\mathbf{P}(y_i = b|\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}: y_i = b} \mathbf{P}(\boldsymbol{y}|\boldsymbol{x})$.

Let us denote a multi-label classifier $\mathbf{h}$ as an $\mathcal{X} \to \mathcal{Y}$ mapping that returns a vector $\mathbf{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x}))$ for a given instance $\boldsymbol{x} \in \mathcal{X}$. Given training data in the form of a finite set of observations $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$, drawn independently from $\mathbf{P}(\mathbf{X}, \mathbf{Y})$,

[1] Poznań University of Technology, Poland, email: Krzysztof.Dembczynski@cs.put.poznan.pl
[2] Gent University, Belgium, email: Willem.Waegeman@ugent.be
[3] Marburg University, Germany, email: eyke@mathematik.uni-marburg.de

the goal in MLC is to learn a classifier $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ that generalizes well beyond these observations in the sense of minimizing the risk with respect to a specific loss function. The risk of a classifier $\mathbf{h}$ is defined as the expected loss over the joint distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$:

$$R_L(\mathbf{h}) = \mathbb{E}_{\mathbf{XY}} L(\mathbf{Y}, \mathbf{h}(\mathbf{X})), \tag{1}$$

where $L(\cdot)$ is a loss function on multi-label predictions. The so-called risk-minimizing model $\mathbf{h}^*$ is determined in a pointwise way by the *risk minimizer*

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

2.1 Joint versus Marginal Mode Prediction

As we are dealing with a multivariate conditional probability distribution over the labels, two of its properties are always of interest: the joint and the marginal mode. Predicting the joint (conditional) mode can be considered as a core operation in many structured output prediction methods such as conditional random fields, leading to a model of the following form:

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

The joint mode of the conditional distribution corresponds to the risk minimizer (2) of the so-called *subset 0/1 loss*, which is formally defined as follows:[4]

$$L_s(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = [\![\boldsymbol{y} \neq \mathbf{h}(\boldsymbol{x})]\!] \tag{4}$$

Prediction of the marginal (conditional) modes, in turn, leads to the model $\mathbf{h}^*(\boldsymbol{x}) = (h_1^*(\boldsymbol{x}), \ldots, h_m^*(\boldsymbol{x}))$ with

$$h_i^*(\boldsymbol{x}) = \arg\max_{b \in \{0,1\}} \mathbf{P}(y_i = b \,|\, \boldsymbol{x}) \ . \tag{5}$$

This is the risk minimizer (2) for the *Hamming loss*, defined as the fraction of labels whose relevance is incorrectly predicted:

$$L_H(\boldsymbol{y}, \mathbf{h}(\boldsymbol{x})) = \frac{1}{m} \sum_{i=1}^{m} [\![y_i \neq h_i(\boldsymbol{x})]\!] \tag{6}$$

Prediction of these two properties of the joint distribution requires different classifiers that exploit the label dependence in a different way, as stated below.

2.2 The Role of Stochastic Label Dependence

From the perspective of modeling label dependencies in MLC, two related notions of label dependence should be carefully distinguished, namely marginal label dependence and conditional label dependence. A vector of labels $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$ is called, respectively, marginally independent and conditionally independent given $\boldsymbol{x}$ if

$$\mathbf{P}(\mathbf{Y}) = \prod_{i=1}^{m} \mathbf{P}(Y_i) \quad \text{resp.} \quad \mathbf{P}(\mathbf{Y}|\boldsymbol{x}) = \prod_{i=1}^{m} \mathbf{P}(Y_i|\boldsymbol{x}). \tag{7}$$

Conditional dependence captures the dependence of the labels given a specific instance $\boldsymbol{x} \in \mathcal{X}$, whereas marginal dependence can be interpreted as a kind of "expected dependence", averaged over all instances. Despite this close connection, one can easily construct examples showing that conditional dependence does not imply marginal dependence nor the other way around [1].

We note that modeling the joint conditional distribution and its joint mode involves exploiting conditional dependence between labels, unlike modeling the marginal modes, where the gain by exploiting the conditional dependence, if any, is rather small. In order to improve the performance in estimating the marginal distributions, the methods should rather exploit marginal dependence, as explained, for example, in [2].

3 PROBABILISTIC CLASSIFIER CHAINS

The Probabilistic Classifier Chains (PCC) method has been introduced in [3] in an attempt to provide a probabilistic interpretation for the previously published Classifier Chains (CC) method [9, 10]. The idea underlying PCC is to repeatedly apply the product rule of probability to the joint distribution of the labels $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$:

$$\mathbf{P}(\boldsymbol{y} \,|\, \boldsymbol{x}) = \prod_{k=1}^{m} \mathbf{P}(y_k | \boldsymbol{x}, y_1, \ldots, y_{k-1}) \tag{8}$$

In other words, PCC represents conditional label dependencies as a fully connected graph. From a theoretical point of view, the order of labels does not play any role, and (8) holds for any permutation of $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$.

Learning a classifier chain can be considered as a simple procedure. According to (8), we decompose the joint distribution into a sequence of marginal distributions that depend on a subset of the labels. These marginal distributions can be learned by m functions $f_k : \mathcal{X} \times \{0,1\}^{k-1} \to [0,1]$ on an augmented input space $\mathcal{X} \times \{0,1\}^{k-1}$, taking $y_1, \ldots, y_{k-1}$ as additional input attributes:

$$f_k : (\boldsymbol{x}, y_1, \ldots, y_{k-1}) \mapsto \mathbf{P}(y_k = 1 \,|\, \boldsymbol{x}, y_1, \ldots, y_{k-1}) \tag{9}$$

We assume that the function $f_k(\cdot)$ can be interpreted as a *probabilistic* classifier whose prediction is the probability that $y_i = 1$, or at least a reasonable approximation thereof. Thus, for any $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, its probability can be estimated by

$$\hat{\mathbf{P}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{m} f_k(\boldsymbol{x}, y_1, \ldots, y_{k-1}) \,. \tag{10}$$

The problem is then to find the risk minimizer for a given loss function over the estimated joint conditional distribution. This process is often referred to as *inference*, and it will be thoroughly analyzed in the next section. To this end, it is convenient to represent the estimated joint conditional distribution as a probability tree. We define the probability tree as a structure (V, E, Π) with V the set of nodes, E the set of edges and $\Pi : E \to [0,1]$ a function that assigns positive weights to edges. Moreover, let us denote a node at depth k as $v_{\boldsymbol{a}} = (a_1, ..., a_k) \in \{0,1\}^k$. Then, the weight of the edge between such a node and its ancestor $\boldsymbol{pa}(v) = (a_1, ..., a_{k-1})$ at depth $k-1$ is given by $\Pi(v_{\boldsymbol{a}}) = \mathbf{P}(Y_k = a_k \mid \boldsymbol{x}, y_1 = a_1, ..., y_{k-1} = a_{k-1})$. As such, depth k of the probability tree represents the decision that is taken in the k-th classifier of the chain. The root of the tree $v_R = \emptyset$ corresponds to depth $k = 0$ with $\Pi(v_R) = 1$.

4 INFERENCE IN PCC

Originally, two approaches have been proposed for inferring a prediction from an estimated chain: an approach based on greedy search,

[4] For a predicate P, the expression $[\![P]\!]$ evaluates to 1 (0) if P is true (false).

being the integral part of the original CC method [9], and an approach based on exhaustive search, as considered in the PCC method [3]. We start with an explanation of the latter, which can be used for any loss. However, this approach turns out to be computationally intractable for problems with many labels. As a simple remedy for this problem, we propose a Monte Carlo sampling. Subsequently, we show that greedy search can be considered as a fast approximate inference algorithm for the conditional joint mode. Since the worst-case regret bound of this method is very high, however, we introduce an enhanced ϵ-approximate algorithm that is tailored for joint mode estimation, resulting in a worst-case regret bound that becomes arbitrarily small as a function of the running time.

4.1 Exhaustive Search and Monte Carlo Sampling

In inference by *exhaustive search*, an optimal prediction is computed explicitly via (2), given an estimate of $\mathbf{P}(\boldsymbol{y} \mid \boldsymbol{x})$ for all $\boldsymbol{y}$ and a loss function $L(\cdot)$. Obviously, this approach is extremely costly, as it comes down to summing over an exponential (2^m) number of label combinations. Moreover, the brute-force search for the optimal solution would also require to check all possible combinations of labels. For some loss functions, like subset 0/1 and Hamming loss, one iteration through the label combinations suffices to compute the optimal solutions, however, this still limits the applicability of the method to datasets with a small to moderate number of labels.

A possible solution for this computational burden consists of conducting a Monte Carlo sampling from the estimated conditional distribution. The sampling procedure is easy to implement by exploiting the probability tree described above. In each node, we flip a biased coin to decide whether the label is relevant or not. Then, we move down in the tree according to this decision. The probability of tails and heads are given, respectively, by the weights $\Pi(\boldsymbol{lc}(v))$ and $\Pi(\boldsymbol{rc}(v))$ of the left and right child of a node v. Thus, one obtains one observation of the conditional distribution as soon as a leaf is reached. Let $S = \{\boldsymbol{y}_i\}_{i=1}^n$ be a sample of observations obtained by repeating the above procedure n times. The prediction is then obtained by minimizing the risk over that sample, i.e.,

$$\mathbf{h}^S(\boldsymbol{x}) = \arg \min_{\boldsymbol{h} \in \{0,1\}^m} \sum_{i=1}^{n} L(\boldsymbol{y}_i, \boldsymbol{h}) \,. \tag{11}$$

From this point of view, PCC can be considered as a general method for multi-label classification, because the above risk minimization problem can be solved efficiently for many multi-label loss functions, including Hamming loss, rank loss and F-measure [4]. However, approximate algorithms might be needed for loss functions for which exact inference becomes intractable, in addition to the approximation that is made by using the sampled conditional distribution instead of the estimated conditional distribution. Nevertheless, in our experiments, we will show that sampling remains competitive in terms of predictive performance.

4.2 Greedy Search

Inference by *greedy search*, for which the pseudo code is given in Algorithm 1, has been introduced as an integral part of the CC method. Briefly summarized, this inference algorithm just follows a single path from the root to one specific leaf. For a new instance $\boldsymbol{x}$ to be classified, the model f_1 predicts y_1, i.e., the relevance of λ_1 for $\boldsymbol{x}$, as usual. Then, f_2 predicts the relevance of λ_2, taking $\boldsymbol{x}$ *plus the predicted value* $y_1 \in \{0, 1\}$ as an input. Proceeding in this way, f_i predicts y_i using $y_1, \ldots, y_{i-1}$ as additional input information. The main advantages of this approach are (a) its low cost and (b) the possibility to use non-probabilistic classifiers, as one only needs to know whether a given label is relevant or not to take a greedy decision in following a path from the root to a leaf. However, we will show for two loss functions that the regret of such an approach can be large.

Algorithm 1 Inference by Greedy Search

```
v ← the root of the probability tree
while v is not a leaf do
    lc(v), rc(v) ← left and right child of v
    if Π(lc(v)) ≥ Π(rc(v)) then
        v ← lc(v)
    else
        v ← rc(v)
    end if
end while
return v = (a_1, ..., a_m) as the mode
```

The regret of a classifier $\boldsymbol{h}$ with respect to a loss function L_z is defined as follows:

$$r_{L_z}(\boldsymbol{h}) = R_{L_z}(\boldsymbol{h}) - R_{L_z}(\boldsymbol{h}_z^*), \tag{12}$$

where R is the risk given by (1), and $\boldsymbol{h}_z^*$ is the Bayes-optimal classifier with respect to the loss function L_z. In the following, we consider the regret with respect to the Hamming loss, given by

$$r_H(\boldsymbol{h}) = \mathbb{E}_{\mathbf{XY}} L_H(\mathbf{Y}, \boldsymbol{h}(\mathbf{X})) - \mathbb{E}_{\mathbf{XY}} L_H(\mathbf{Y}, \boldsymbol{h}_H^*(\mathbf{X})),$$

and the subset 0/1 loss, given by

$$r_s(\boldsymbol{h}) = \mathbb{E}_{\mathbf{XY}} L_s(\mathbf{Y}, \boldsymbol{h}(\mathbf{X})) - \mathbb{E}_{\mathbf{XY}} L_s(\mathbf{Y}, \boldsymbol{h}_s^*(\mathbf{X})).$$

Since both loss functions are decomposable with respect to individual instances, we analyze the expectation over $\mathbf{Y}$ for a given $\boldsymbol{x}$. The following proposition summarizes the highest value of the regret for the greedy approach in terms of the subset 0/1 loss and the Hamming loss (we omit the proof due to space restrictions).

Theorem 1 *Under the assumption that a probabilistic classifier chain obtains a perfect estimate of the conditional probability* $\mathbf{P}(\boldsymbol{y}|\boldsymbol{x})$*, the following tight upper bounds hold for the regret of the prediction* $\boldsymbol{h}_G(\boldsymbol{x})$ *of the greedy approach:*

$$\sup_{\mathbf{P}} \left(\mathbb{E}_{\mathbf{Y}} L_s(\mathbf{Y}, \boldsymbol{h}_G(\boldsymbol{x})) - \mathbb{E}_{\mathbf{Y}} L_s(\mathbf{Y}, \boldsymbol{h}_s^*(\boldsymbol{x}))\right) = 2^{-1} - 2^{-m},$$

$$\sup_{\mathbf{P}} \left(\mathbb{E}_{\mathbf{Y}} L_H(\mathbf{Y}, \boldsymbol{h}_G(\boldsymbol{x})) - \mathbb{E}_{\mathbf{Y}} L_H(\mathbf{Y}, \boldsymbol{h}_H^*(\boldsymbol{x}))\right) = 1 - \frac{1}{m} \sum_{i=1}^{m} 2^{-i},$$

where the supremum is taken over all probability distributions on $\mathcal{Y}$*.*

As we can see, the regret is quite high in both cases, suggesting that inference by greedy search can yield a poor performance for both loss functions. Nevertheless, we argue that this approach is still more appropriate for the subset 0/1 loss. When the number of labels increases, then the regret converges to 0.5 for the subset 0/1 loss, while it even converges to the maximum possible value of 1 for the Hamming loss. Hence, it is tempting to conclude that the greedy search procedure is indeed more suitable for estimating the joint than the marginal mode, all the more since the subset 0/1 loss, in terms of its absolute value, is even higher than the Hamming loss (which, for example, can already be reduced to 1/2 by random guessing). Furthermore, one may wonder whether one can find an optimal order of labels in the chain, for which the regret would decrease downward to zero. Unfortunately, this is provably impossible (proof omitted due to space restrictions). An interesting issue being a subject of our future

Algorithm 2 ϵ-approximate inference

ordered list $Q \leftarrow \{v_R\}$ (contains root node initially)
ordered list $K \leftarrow \{\}$ (non-survived parents)
define $\overline{\Pi}(v_R) = 1$
$\epsilon \leftarrow 2^{-k}$ with $k \leq m$
while $Q \neq \emptyset$ **do**
  $v \leftarrow$ pop first element in Q
  if v is a leaf **then**
    delete all elements in K and **break the while loop**
  end if
  $\boldsymbol{lc}(v), \boldsymbol{rc}(v) \leftarrow$ left and right child of v
  compute $\overline{\Pi}(\boldsymbol{lc}(v)), \overline{\Pi}(\boldsymbol{rc}(v))$ recursively from $\overline{\Pi}(v)$ using (13)
  if $\overline{\Pi}(\boldsymbol{lc}(v)) \geq \epsilon$ **then**
    insert $\boldsymbol{lc}(v)$ in list Q sorted according to $\overline{\Pi}(\boldsymbol{lc}(v))$
  end if
  if $\overline{\Pi}(\boldsymbol{rc}(v)) \geq \epsilon$ **then**
    insert $\boldsymbol{rc}(v)$ in list Q sorted according to $\overline{\Pi}(\boldsymbol{rc}(v))$
  end if
  if $\boldsymbol{lc}(v)$ and $\boldsymbol{rc}(v)$ are not inserted to the list **then**
    insert v in list K sorted according to $\overline{\Pi}(v)$
  end if
end while
$\epsilon \leftarrow 0$
while $K \neq \emptyset$ **do**
  $v' \leftarrow$ pop first element in K and apply Alg. 1 downward on it
  if $\overline{\Pi}(v') \geq \epsilon$ **then**
    $v \leftarrow v'$ and $\epsilon \leftarrow \overline{\Pi}(v')$
  end if
end while
return $v = (a_1, ..., a_m)$ as the mode

work is to check whether the maximal value of the regret becomes smaller if the order of the labels would be changed or even optimized.

Let us remark, however, that the risk minimizers of the Hamming loss and the subset 0/1 loss coincide in many specific situations, like conditional independence of labels, or if the probability of the joint mode is greater than or equal to 0.5. One can easily observe that the worst-case regret of the greedy search algorithm is zero for both losses in these two situations. At the same time, these facts may also explain why algorithms not tailored for specific losses have been reported to obtain good results in many empirical studies.

4.3 An ϵ-approximate algorithm

Since the regret of the greedy search procedure can be high, we propose in this section a specific algorithm for which a much smaller upper bound on the regret can be derived. From a graph-theoretic perspective, the algorithm computes the shortest path between the root of the probability tree and a fictitious dummy node that is connected to the leaves of the probability tree. Given the probability tree structure that was introduced in the previous section, let us define the path distance $\overline{\Pi}(v_{\boldsymbol{a}})$ between the root node $v_R = \emptyset$ and any node $v_{\boldsymbol{a}}$ recursively, as a product of edge weights:

$$\overline{\Pi}(v_{\boldsymbol{a}}) = \Pi(v_{\boldsymbol{a}}) \times \overline{\Pi}(\boldsymbol{pa}(v_{\boldsymbol{a}}))\,, \tag{13}$$

where $\boldsymbol{pa}(v)$ denotes the parent of a given node v.

Using this notation, the pseudo code of our algorithm is summarized in Algorithm 2. In a nutshell, the algorithm starts from the root of the probability tree, which is the single element of an ordered list Q. In every iteration, the top element of the list is popped and the children of the corresponding node are visited. The path distance $\overline{\Pi}(v)$ to the root can be recursively computed for these children, and they are added to the list if the path distance is bigger than the threshold $\epsilon = 2^{-k}$ with $1 \leq k \leq m$. Basically, they are inserted in the list at the appropriate position, so that the order imposed by $\overline{\Pi}(v)$ is respected.

The while loop of the algorithm stops in two situations: (1) when the element popped from the list Q corresponds to a leaf of the probability tree or (2) when the list Q is empty. The label combination corresponding to the leaf is then returned in the former case, while inference by greedy search, as described above, is applied to define a path from all non-survived nodes from the list K (i.e., nodes for which none of their children has been added to Q) to a leaf with corresponding prediction in the latter case. The following theorem states that in both cases the regret of the prediction can be bounded as a function of the number of iterations of the algorithm.

Theorem 2 *Let $k \leq m$. Under the assumption that a probabilistic classifier chain obtains a perfect estimate of the conditional probability $\mathbf{P}(\boldsymbol{y}|\boldsymbol{x})$, Algorithm 2 needs less than $\mathcal{O}(m2^k)$ iterations to find a prediction $h_\epsilon(\boldsymbol{x})$ with low worst-case regret for subset 0/1 loss, i.e.*

$$\sup_{\mathbf{P}} \left(\mathbb{E}_{\mathbf{Y}} L_s(\mathbf{Y}, \boldsymbol{h}_\epsilon(\boldsymbol{x})) - \mathbb{E}_{\mathbf{Y}} L_s(\mathbf{Y}, \boldsymbol{h}^*_s(\boldsymbol{x}))\right) \leq 2^{-k} - 2^{-m}\,.$$

Proof: The algorithm searches for the mode in an ϵ-approximate way, so the bound on the regret necessarily holds for all $k \leq m$. We only need to show that the algorithm needs less than $m2^{-k}$ iterations. To this end, one can observe that the list Q will always contain less than 2^k nodes, because at most 2^k nodes can have a value $\overline{\Pi}(v)$ which is bigger than 2^{-k}. Furthermore, every element in the list Q can be replaced at most m times by one or two new elements. The same reasoning can be applied to the list K. Thus, the algorithm finds an upper bound on the regret as a function of the running time of the algorithm. Consequently, the algorithm will always find the mode of the distribution, if the probability mass of the mode is higher than the upper bound on the regret. This is summarized in the following corollary.

Corollary 1 *Let $k \leq m$ and let $\mathbf{P}$ be a probability distribution for which the joint mode has a probability mass bigger than 2^{-k}, then Algorithm 2 needs less than $m2^k$ iterations to find a prediction $h_\epsilon(\boldsymbol{x})$ that corresponds to this mode.*

Table 1. Statistics of datasets: training and test set sizes, number of features and labels, minimal, average, maximal number of relevant labels.

DATA SET	# TRAIN	# TEST	# ATTR.	# LAB.	MIN	AVE.	MAX
SCENE	1211	1196	294	6	1	1.062	3
YEAST	1500	917	103	14	1	4.228	11
TMC2007-500	21519	7077	500	22	1	2.226	10
MEDICAL	333	645	1449	45	1	1.255	3
ENRON	1123	579	1001	53	1	3.386	11
REUTERS (1)	3000	3000	500	103	1	3.176	11
MEDIAMILL	30993	12914	120	101	0	4.363	18
EMOTIONS	391	202	72	6	1	1.813	3
SYNTH1	471	5045	6000	6	1	2.045	6
SYNTH2	1000	10000	40	10	1	1	1

5 EXPERIMENTAL STUDY

Two types of experiments that we describe here intend to confirm our theoretical claims. To this end, we follow a similar experimental setup as in [6], in which four benchmark and two synthetic datasets with known training and test parts have been used. We extend this setup with four other datasets to emphasize the interesting computational complexity properties of our approach for high-dimensional label spaces. All eight real-world datasets (summarized in Table 1) were downloaded from the MULAN and LibSVM multi-label dataset repositories, and the two synthetic datasets were generated using the description in [6].[5]

[5] The original training and test sets have not been published for the two synthetic datasets. We do not describe these datasets here due to space limitations, and we refer the reader to the original paper. To obtain more stable results, we report the results as an average over 5 replications of these synthetic datasets.

Table 2. Results on benchmark data sets: Training time, test time, Hamming and subset 0/1 loss on test sets with standard error (the best results for each dataset and loss function is highlighted in bold).

	TRAIN TIME [S]	HAMMING LOSS	SUBSET 0/1 LOSS	TEST TIME [S]	TRAIN TIME [S]	HAMMING LOSS	SUBSET 0/1 LOSS	TEST TIME [S]
	SCENE				YEAST			
PCC ϵ=.5	420.641	0.115±.004	0.417±.014	0.375	232.249	0.213±.005	0.787±.014	0.172
PCC ϵ=.25	SAA	0.107±.004	**0.385**±.014	0.375	SAA	0.211±.006	0.764±.014	0.281
PCC ϵ=.0	SAA	0.107±.004	**0.385**±.014	0.375	SAA	0.210±.006	**0.761**±.014	0.344
BR	417.985	**0.102**±.003	0.509±.014	0.328	204.405	**0.199**±.005	0.842±.012	0.141
	MEDIAMILL				REUTERS			
PCC ϵ=.5	37202.797	0.032±.000	**0.885**±.003	41.234	15227.574	0.018±.001	0.615±.009	19.438
PCC ϵ=.25	SAA	0.032±.000	0.886±.003	53.454	SAA	**0.017**±.001	0.601±.009	21.938
PCC ϵ=.0	SAA	0.034±.000	**0.885**±.003	86.547	SAA	**0.017**±.001	**0.598**±.009	23.250
BR	16903.109	**0.030**±.000	0.902±.003	26.062	13476.883	**0.017**±.001	0.689±.008	15.359
	SYNTH1				SYNTH2			
PCC ϵ=.5	7591.826	**0.067**±.002	**0.238**±.006	15.828	26.968	**0.000**±.000	**0.000**±.000	0.735
PCC ϵ=.25	SAA	**0.067**±.002	0.239±.006	15.578	SAA	**0.000**±.000	**0.000**±.000	0.734
PCC ϵ=.0	SAA	**0.067**±.002	0.239±.006	15.735	SAA	**0.000**±.000	**0.000**±.000	0.766
BR	6955.159	**0.067**±.002	0.240±.006	12.687	16.453	0.084±.001	0.832±.004	0.609
	TMC2007-500				ENRON			
PCC ϵ=.5	21703.017	0.056±.001	0.676±.006	9.360	13387.680	0.047±.001	0.869±.014	3.547
PCC ϵ=.25	SAA	0.056±.001	0.670±.006	13.969	SAA	**0.046**±.001	0.848±.015	5.031
PCC ϵ=.0	SAA	0.056±.001	**0.668**±.006	14.359	SAA	0.047±.001	**0.845**±.015	8.907
BR	22942.510	**0.055**±.001	0.685±.006	8.312	11894.534	0.047±.001	0.886±.013	3.046
	EMOTIONS				MEDICAL			
PCC ϵ=.5	14.078	0.224±.013	0.752±.030	0.015	2613.459	0.016±.001	0.546±.020	4.407
PCC ϵ=.25	SAA	**0.219**±.013	**0.718**±.032	0.016	SAA	**0.015**±.001	**0.541**±.020	4.109
PCC ϵ=.0	SAA	0.222±.014	**0.718**±.032	0.015	SAA	**0.015**±.001	**0.541**±.020	4.172
BR	12.328	0.226±.011	0.812±.027	0.016	2337.824	0.016±.001	0.550±.020	3.110

5.1 Greedy and ϵ-approximate Inference

In the first experiment, we show that inference by greedy search is more appropriate for estimating the joint mode, while substantial performance gains can be obtained by applying our ϵ-approximate inference algorithm. Moreover, using this strategy, we reach a computational cost that is more than fair for real-world applications. As a result, we perform a comparison of the three variants of PCC: 1) inference by greedy search for PCC, which resembles the ϵ-approximate inference algorithm to PCC with $\epsilon = 0.5$ (denoted PCC $\epsilon = 0.5$), 2) the ϵ-approximate inference algorithm with $\epsilon = 0.25$ (PCC $\epsilon = 0.25$), 3) the exact inference, meaning $\epsilon = 0$ (PCC $\epsilon = 0.0$). We also compare with a binary relevance (BR) learner that serves as a baseline by training a classifier for each label separately. It should perform well for the Hamming loss, while all the variants of PCC should perform well for the subset 0/1 loss. As a base learner, we use a regularized logistic regression model. We apply an internal three-fold cross-validation[6] on training data for tuning the regularization parameter with possible values $\{1000, 100, 1, 0.1, 0.01, 0.001\}$. This tuning is performed for each base classifier by choosing the model with lowest empirical logistic loss in order to obtain probability estimates that are as accurate as possible.

The results are given in Table 2. We can observe that our ϵ-approximate inference works as expected: with decreasing ϵ, the subset 0/1 loss usually decreases. If this is not the case, then all the inference algorithms perform almost equally. Interestingly, the exact algorithm PCC $\epsilon = 0.0$ performs fast, being in the worse case only 2 times slower than the greedy approach. We can also observe that the greedy approach is appropriate for the subset 0/1 loss. It always obtained better results than BR for this loss, while BR is almost always better for the Hamming loss. In general, BR performs the best in estimating the marginal modes. Two small exceptions are the Synth2 and the Emotions datasets that we will discuss in more detail in Subsection 5.3. Interestingly, for datasets with many labels and for all the algorithms, almost no difference in performance was observed on the Hamming loss, in contrast to the subset 0/1 loss.

[6] for large datasets (with number of training instances $\geq$ 10000) we used 66% split

5.2 Comparison with Structured SVMs

Adopting the experimental setup of [6] enables us to compare our methods with a variety of approximate inference methods for structured SVMs in a straightforward and fair manner. Unfortunately, such a comparison can be made solely for the Hamming loss, which was the only loss function reported in the original paper. By restricting our analysis to the six datasets that were studied in [6], it is shown that PCC can be applied as well for Hamming loss minimization. To this end, we show that the exhaustive search (PCC EX) and Monte Carlo sampling (PCC MC) variants of PCC obtain a competitive performance with the best inference methods for structured SVMs (denoted SSVM Best). We compute the empirical marginal modes for the Monte Carlo sampling variant and the sample size is always set to 1000 elements. Approximate inference is required in structured SVMs as well, if fully connected Markov logic fields are considered as underlying models. Five different inference algorithms were considered in [6], including the exhaustive search procedure. This is the main reason for why we analyze the exhaustive variant of PCC, and why only the most 10 frequent labels are considered for the Mediamill and the Reuters dataset (as in the original study).

The comparison with SSVM can be considered as fair, especially since the underlying models are based on the same representation. In fact, in the original SSVM method, a joint feature mapping was used that models all pairwise dependencies between labels:

$$\Psi(\boldsymbol{x}, \boldsymbol{y}) = (y'_1\boldsymbol{x}, \ldots, y'_m\boldsymbol{x}, y'_1y'_2, y'_1y'_3, \ldots, y'_{m-1}y'_m), \qquad (14)$$

where $y'_i = (2y_i - 1) \in \{-1, 1\}$. Interestingly, one can show that, when using logistic regression as a base learner, PCC leads to a con-

ditional probability model of the following type:

$$\mathbf{P}_{\boldsymbol{w}}(\boldsymbol{y} \mid \boldsymbol{x}) = Z(\boldsymbol{x}, \boldsymbol{w})^{-1} e^{-\boldsymbol{w}^T \Psi(\boldsymbol{x}, \boldsymbol{y})},$$

with $\boldsymbol{w}$ a vector of parameters and $Z(\boldsymbol{x}, \boldsymbol{w})$ a normalization constant.[7] Thus, the main difference is the loss to be minimized, namely structured hinge loss versus log-loss [8]. As a main (computational) benefit, our approach allows one to solve m learning problems independently during the training phase, without imposing any restrictions on modeling label dependencies.

Table 3. Results on the data sets used in [6]; SSVM Best denotes the best result over all inference algorithms used in SSVM for a given dataset.

		TRAIN TIME[S]	HAMMING LOSS	TEST TIME[S]
SCENE	PCC MC	420.6	0.104±.004	1.2
	PCC EX	SAA	0.102±.004	5.1
	SSVM BEST	—	**0.101**±.003	
YEAST	PCC MC	232.2	0.203±.005	4.9
	PCC EX	SAA	**0.201**±.005	463.0
	SSVM BEST	—	0.202±.005	—
MEDIAMILL	PCC MC	5808.1	0.172±.001	27.9
	PCC EX	SAA	**0.170**±.001	403.6
	SSVM BEST	—	0.182±.001	—
SYNTH 1	PCC MC	7591.8	**0.067**±.001	34.9
	PCC EX	SAA	**0.067**±.001	240.7
	SSVM BEST	—	0.069±.001	—
REUTERS (10 LABELS)	PCC MC	2659.8	0.060±.002	11.3
	PCC EX	SAA	0.059±.002	336.7
	SSVM BEST	—	**0.045**±.001	—
SYNTH 2	PCC MC	26.9	**0.000**±.000	1.9
	PCC EX	SAA	**0.000**±.000	114.6
	SSVM BEST	—	0.058±.001	—

The results are given in Table 3, where we report the best result of SSVM over all inference algorithms obtained on a given dataset. As we can see, PCC is competitive to SSVM. Worse results are only obtained on Reuters, but here we used a simple feature selection relying on picking the 500 most frequent features to speed up the logistic regression procedure.

5.3 Discussion: Binary Relevance Revisited

One may wonder whether exploiting conditional dependence, as defined in (7), could also help to improve the Hamming loss. Simple BR obtains a very competitive performance on Hamming loss (also on the reduced Mediamill and Reuters datasets, not reported, because of space limitations). However, we observe two exceptions: the Emotion and the Synth2 datasets. Changing the base learner from a linear to a polynomial basis leads to a performance for BR that is comparable to PCC for the former dataset (this dataset was not used in [6]).

The reasons are more involved for the latter dataset, which defines in fact a simple ordinal classification task without noise. Theoretically, conditional independence holds in such a case, thus joint and marginal modes coincide here. This means that the concept is learnable for algorithms tailored for both losses. However, as the optimal base learner becomes strongly nonlinear for BR, more training examples are needed to reduce the error down to zero. Interestingly, PCC succeeded in learning the concept without error using the original size of the training set, in contrast to structured SVMs. BR performs even worse in this case.

[7] In the same way, a close connection can be established to conditional random fields.

The performance boost of both methods in comparison to BR can be attributed to a hypothesis space extension. Applying a linear base classifier to PCC yields a much richer hypothesis space in comparison to applying the same base learner to BR. A similar argument has been put forward in comparing one-versus-all and one-versus-one multi-class classifiers. It was shown in [11] that the one-vs-all approach performs as good as other reduction schemes if complex base classifiers are used. Moreover, [7] introduced another way of exploiting the max-margin principle for minimizing the Hamming loss in multi-label classification. However, a detailed comparison with this approach is out of the scope of this paper.

6 CONCLUSIONS

Summarizing the above theoretical and empirical results, we conclude that our ϵ-approximate inference algorithm provides accurate and efficient estimates of the joint mode. The greedy inference algorithm, which is an integral part of the original CC algorithm, seems to be mainly tailored for subset 0/1 loss. This was not clear from the original paper. Additionally, we showed that probabilistic classifier chains can be easily extended for marginal mode estimation, leading to a general class of models that exhibit many interesting properties, such as mechanisms for parallelization, possibilities for applying different base learners, strong connections with conditional random fields and a predictive performance that is competitive with structured SVMs.

Due to lack of space, other important issues playing a key role in chaining could not be discussed in detail. Amongst others, we intend to investigate in future work the effect of ensembling multiple classifiers, as considered for CC and PCC in the original papers, and the necessity for considering conditional dependence in marginal mode estimation, which is often put forward as the main shortcoming of binary relevance approaches.

REFERENCES

[1] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.

[2] L. Breiman and J. Friedman, 'Predicting multivariate responses in multiple linear regression', *J R Stat. Soc. Ser. B*, **69**, 3–54, (1997).

[3] K. Dembczyński, W. Cheng, and E. Hüllermeier, 'Bayes optimal multilabel classification via probabilistic classifier chains', in *ICML 2010*. Omnipress, (2010).

[4] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, 'An exact algorithm for F-measure maximization', in *Advances in Neural Information Processing Systems 24 (NIPS-11)*, 223–230, (2011).

[5] K. Dembczynski, W. Waegeman, and E. Hüllermeier, 'Joint mode estimation in multi-label classification by chaining', in *ECML Workshop on Collective Inference and Learning on Structured Data*, Athens, Greece, (2011).

[6] T. Finley and T. Joachims, 'Training structural SVMs when exact inference is intractable', in *ICML 2008*. Omnipress, (2008).

[7] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma, 'Large scale max-margin multi-label classification with priors', in *ICML 2010*. Omnipress, (2010).

[8] P. Pletscher, C.S. Ong, and J.M. Buhmann, 'Entropy and margin maximization for structured output learning', in *ECML/PKDD 2010*. Springer, (2010).

[9] J. Read, B. Pfahringer, G. Holmes, and E. Frank, 'Classifier chains for multi-label classification', in *ECML/PKDD 2009*, pp. 254–269, (2009).

[10] J. Read, B. Pfahringer, G. Holmes, and E. Frank, 'Classifier chains for multi-label classification', *Machine Learning*, **85**(3), 333–359, (2011).

[11] R.M. Rifkin and A. Klautau, 'In defense of one-vs-all classification', *Journal of Machine Learning Research*, **5**, 101–0141, (2004).