Journal of Machine Learning Research 18 (2017) 1-38 Submitted 4/16; Revised 7/17; Published 9/17

An Easy-to-hard Learning Paradigm for Multiple Classes and Multiple Labels

Weiwei Liu LIUWEIWEI863@GMAIL.COM
School of Computer Science and Engineering
The University of New South Wales, Sydney
NSW 2052, Australia
and
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia

Ivor W. Tsang IVOR.TSANG@UTS.EDU.AU
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia

Klaus-Robert Müller KLAUS-ROBERT.MUELLER@TU-BERLIN.DE
Machine Learning Group, Computer Science
Berlin Institute of Technology (TU Berlin)
Marchstr 23, 10587 Berlin, Germany
and
Max Planck Institute for Informatics
Stuhlsatzenhausweg, 66123, Saarbrcken
and
Department of Brain and Cognitive Engineering
Korea University, Seoul 02841, Korea

Editor: Karsten Borgwardt

Abstract

Many applications, such as human action recognition and object detection, can be formulated as a multiclass classification problem. One-vs-rest (OVR) is one of the most widely used approaches for multiclass classification due to its simplicity and excellent performance. However, many confusing classes in such applications will degrade its results. For example, *hand clap* and *boxing* are two confusing actions. *Hand clap* is easily misclassified as *boxing*, and vice versa. Therefore, precisely classifying confusing classes remains a challenging task. To obtain better performance for multiclass classifications that have confusing classes, we first develop a classifier chain model for multiclass classification (CCMC) to transfer class information between classifiers. Then, based on an analysis of our proposed model, we propose an easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes. Similar to CCMC, the classifier chain (CC) model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. However, CC does not consider the order of difficulty of the labels and achieves degenerated performance when there are many confusing labels. Therefore, it is non-trivial to learn the appropriate label order for CC. Motivated by our analysis for CCMC, we also propose the easy-to-hard learning paradigm for multi-label classification to automatically identify easy and hard labels, and then use the predictions from simpler labels to help solve harder labels. We also demonstrate that our proposed strategy can be successfully applied to a wide range of applications, such as ordinal classification and relationship prediction. Extensive empirical studies validate our analysis and the effectiveness of our proposed easy-to-hard learning strategies.

Keywords: Multiclass Classification, Multi-label Classification, Classifier Chain, Easy-to-hard Learning Paradigm

1. Introduction

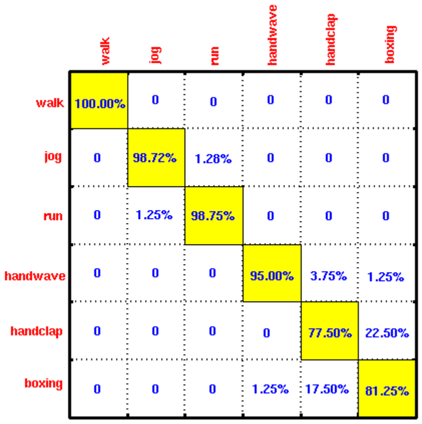
Many applications can be formulated as a multiclass classification problem. For example, human action recognition aims to classify videos into different categories of human action. In the KTH data set (Schüldt et al., 2004), for example, there are six types of human action: *walk*, *jog*, *run*, *hand wave*, *hand clap* and *boxing*. These actions can be classified into two main categories: leg movements (*walk*, *jog* and *run*) and hand movements (*hand wave*, *hand clap* and *boxing*). It is easy to differentiate between leg and hand movements, such as *walk* and *hand wave*, but actions within leg or hand movements, such as hand clap and boxing, are easily confused. *Hand clap* is easily misclassified as *boxing*, and vice versa. Confusing classes are ubiquitous in real world applications, especially for data sets with many classes. For example, there are many confusing classes in the ALOI data set from the LIBSVM website[1], which contains 1,000 classes. Therefore, precisely classifying multiclass data sets with confusing classes is a challenging task.

From the confusion matrix of one-vs-rest (OVR) on the KTH data set shown in Figure 1, we observe that *walk* is the easiest action to classify and *hand clap* is the hardest action to classify, as the *walk* action can be correctly identified by OVR whereas the percentage of *hand clap* images that are misclassified as *boxing* is 22.5%. To achieve accurate prediction performance, according to Figure 1, we should classify *walk* first, followed by *run*, *jog* and *hand wave*. *Boxing* and *hand clap* are the last two actions to classify. The motivation behind this paper is to solve classification tasks from easy to hard, and to use the predictions from simpler tasks to help solve the harder tasks.

To achieve our goal, a classifier chain model for multiclass classification (CCMC) is proposed to transfer class information between classifiers. Furthermore, we generalize CCMC over a random class order and provide a theoretical analysis of the generalization error for the proposed generalized model. Our results show that the upper bound of the generalization error depends on the sum of the reciprocal of the square of the margin over the classes. Therefore, we conclude that class order does affect the performance of CCMC, and a globally optimal class order exists only when the minimization of the upper bound is achieved over this CCMC. Lastly, based on our results, we propose the easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes.

Multi-label classification, where each instance can belong to multiple labels simultaneously, has garnered significant attention from researchers as a result of its various applica-

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets



	walk	jog	run	handwave	handclap	boxing
walk	100.00%	0	0	0	0	0
jog	0	98.72%	1.28%	0	0	0
run	0	1.25%	98.75%	0	0	0
handwave	0	0	0	95.00%	3.75%	1.25%
handclap	0	0	0	0	77.50%	22.50%
boxing	0	0	0	1.25%	17.50%	81.25%

Figure 1: Confusion Matrix of OVR on the KTH Data Set. In the confusion matrix, the entry in the i-th row and j-th column is the percentage of images from action i that are misclassified as action j. Average classification accuracy rates for individual actions are listed along the diagonal, which is colored yellow.

tions, which range from document classification and gene function prediction, to automatic image annotation. For example, a document can be associated with a range of topics, such as *Sports*, *Finance* and *Education* (Schapire and Singer, 2000); a gene belongs to the functions of *protein synthesis*, *metabolism* and *transcription* (Barutcuoglu et al., 2006); an image may have both *beach* and *tree* tags (Boutell et al., 2004).

Similar to CCMC, a classifier chain (CC) model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. It also tries to use information from previous labels to help train the classifier for the next label. However, CC's performance degenerates when there are many confusing labels, because the main drawback of CC is that it does not consider the order of difficulty of the labels. Therefore, it is non-trivial to learn the appropriate label order for CC.

Motivated by our analysis for CCMC, we first generalize CC over a random label order and provide the generalization error bound for the proposed generalized model. Then we propose the easy-to-hard learning paradigm for multi-label classification to automatically identify easy and hard labels. Lastly, we use the predictions from simpler labels to help solve harder labels. To learn the objective of our proposed easy-to-hard learning paradigms, it is very expensive to search over $q!$ different class or label orders[2], where q denotes the number of classes or labels, which is computationally infeasible for a large q. We thus propose a set of easy-to-hard learning algorithms to simplify the search process of the optimal learning sequence.

Experiments on a wide spectrum of data sets show that our proposed methods excel in all data sets for multi-label and multiclass classification problems. The results validate our analysis and the effectiveness of our proposed easy-to-hard learning algorithms. Lastly, we demonstrate that our proposed easy-to-hard learning strategies can be successfully applied

[2] ! represents the factorial notation.

to a wide range of applications, such as ordinal classification (Chu and Ghahramani, 2005) and relationship prediction (Massa and Avesani, 2006).

We organize this paper as follows. Section 2 summarizes existing related works and problems. The easy-to-hard learning paradigm for multi-label and multiclass classification are proposed in Sections 3 and 4. Learning algorithms and time complexity analysis are described in Section 5. We present two applications in Section 6. Section 7 shows the comprehensive experimental results. The last section provides concluding remarks.

Notations: Assume $\mathbf{x}_t \in \mathbb{R}^d$ is a real vector representing an input or instance (feature) for $t \in \{1, \cdots, n\}$. n denotes the number of training instances. $\mathcal{Y}_t \subseteq \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ is the corresponding output (class or label). $\mathbf{y}_t \in \{0, 1\}^q$ is used to represent the set $\mathcal{Y}_t$, where $\mathbf{y}_t(j) = 1$ if and only if $\lambda_j \in \mathcal{Y}_t$. Note that, there is only one element in $\mathcal{Y}_t$ for the multiclass problem.

2. Related Work and Problems

2.1 Multiclass Classification

2.1.1 OVR and OVO

One-vs-rest (OVR) and one-vs-one (OVO) are two famous strategies for decomposing multiclass classification problems into multiple binary classification problems. Hsu and Lin (2002), Rifkin and Klautau (2004), Demirkesen and Cherifi (2008) and Lapin et al. (2015, 2016) have already shown that OVR and OVO are successful schemes that are as accurate as more complicated approaches, such as error-correcting output coding (ECOC) (Dietterich and Bakiri, 1995), the tree-based method (Beygelzimer et al., 2009a,b; Bengio et al., 2010; Yang and Tsang, 2011; Liu and Tsang, 2016) and multi-class SVM (Weston and Watkins, 1999; Lapin et al., 2015, 2016).

OVR works as follows: a binary classifier is trained for each class λ_j, with all of the instances in the j-th class having positive labels, and all other instances having negative labels. q binary classifiers are then trained on $\{\mathbf{x}_t, \mathbf{y}_t(1)\}_{t=1}^n, \cdots, \{\mathbf{x}_t, \mathbf{y}_t(q)\}_{t=1}^n$. The final output of OVR for each testing instance is the class that corresponds to the classifier with the highest output value. OVR ignores correlations between classes and each classifier is trained independently. OVO trains all possible $q(q-1)/2$ binary classifiers from a training set of q classes, where each classifier is trained on only two out of q classes.

The main differences between OVR and OVO relate to computational issues and applications: 1) Computational issues: OVO requires $\mathcal{O}(q^2)$ classifiers, while OVR trains $\mathcal{O}(q)$ classifiers. If q is very large, then the cost of OVO will be prohibitive. 2) Applications: many real world data sets are partially labelled, as a result of the heavy burden of labelling data. We call this kind of data “background data” (See also (Niu et al., 2016)). It arises in many applications, for example, multiclass image segmentation (Guillaumin et al., 2010; Pham et al., 2015), object detection (Torralba et al., 2004; Huo et al., 2016) and multiclass video segmentation (Budvytis et al., 2010). OVR can be used both in supervised and semi-supervised settings, while OVO can not be used in semi-supervised settings. Thus, OVR can be used to tackle this kind of background data, but OVO cannot. Milgram et al. (2006) have already shown that OVR appears to be significantly more accurate than OVO for handwriting recognition.

2.1.2 Rifkin and Klautau's conjecture

OVR is a lower cost approach with many more applications than OVO. However, Rifkin and Klautau (2004), who conduct a thorough study on OVR, point out that the condition of OVR working as well as any other clever schemes is that the classes are independent - we do not necessarily expect instances from class “A” to be closer to those in class “B” than those in class “C”. They also speculate that an algorithm that exploits the relationship between classes could offer superior performance, and that this would remains an open problem. To the best of our knowledge, this problem has still not been well-addressed, which is why the present paper studies this problem to provide an answer.

2.2 Multi-label Classification

One popular strategy for multi-label classification is to reduce the original problem into many binary classification problems. Many works have followed this strategy. For example, binary relevance (BR) (Tsoumakas et al., 2010) is a simple approach for multi-label learning which independently trains a binary classifier for each label. Recently, Chen and Lin (2012); Liu and Tsang (2015a,b); Zhang and Zhou (2014); Gong et al. (2017); Liu and Tsang (2017) have shown that multi-label learning methods that explicitly capture label dependency will usually achieve better prediction performance. Therefore, modeling label dependency is one of the major challenges in multi-label classification problems.

To capture label dependency, Hsu et al. (2009) first use the compressed sensing technique to handle multi-label classification problems. They project the original label space into a low dimensional label space. A regression model is then trained on each transformed label. Lastly, multi-labels are recovered from the regression output, which usually involves solving a quadratic programming problem (Hsu et al., 2009). Many works have been developed in this way (Zhang and Schneider, 2011, 2012; Tai and Lin, 2012). Such methods mainly aim to use different projection methods to transform the original label space into another effective label space. However, an expensive encoding and decoding procedure prevents these methods from being practical.

Another important approach attempts to exploit the different orders (first-order, second-order and high-order) of label correlations (Zhang and Zhang, 2010; Zhang and Zhou, 2014). Following this way, some works try to provide a probabilistic interpretation for label correlations. For example, Guo and Gu (2011) model the label correlations using a conditional dependency network; PCC (Dembczynski et al., 2010) exploits a high-order Markov Chain model to capture the correlations between the labels and provide an accurate probabilistic interpretation of classifier chain (CC) (Read et al., 2009). Some other works (Kang et al., 2006; Read et al., 2009; Huang and Zhou, 2012) focus on modeling the label correlations in a deterministic way. Among them, the CC model is one of the most popular methods due to its simplicity and promising experimental results (Read et al., 2009).

CC works as follows: one classifier is trained for each label. For the $(i+1)$th label, each instance is augmented with the 1st, 2nd, $\cdots$, ith label as the input to train the $(i+1)$th classifier. Given a new instance to be classified, CC firstly predicts the value of the first label, then takes this instance together with the predicted value as the input to predict the value of the next label. CC proceeds in this way until the last label is predicted. However, here the question is: *Does the label order affect the performance of CC?* Apparently yes, because different classifier chains involve different classifiers trained on different training sets. Thus, to reduce the influence of the label order, Read et al. (2009) propose the ensemble of classifier chains (ECC) to average the multi-label predictions of CC over a set of random ordering chains. Since the performance of CC is sensitive to the choice of label, there is another important question: *Is there any globally optimal classifier chain which can achieve the optimal prediction performance for CC?* If yes, *how can the globally optimal classifier chain be found?* This paper studies this problem and provides an answer.

2.3 Curriculum Learning

Curriculum learning (Bengio et al., 2009) can be seen as a sequence of training criteria. Each training criterion in the sequence is associated with a different set of weights in the training examples, or more generally, in a re-weighting of the training distribution. Initially, the weights favor easier examples that can be learned most easily. The next training criterion involves a slight change in the weighting of examples that increases the probability of sampling slightly more difficult examples. Overall, curriculum learning aims to find easier examples. However, up to now, curriculum learning has not defined what easy examples mean, or equivalently, how to sort the examples into a sequence that illustrates the simpler concepts first.

Inspired by curriculum learning, this paper clearly defines easy and hard tasks and provides the strategy to learn easy and hard tasks. Our empirical studies verify that our method is able to automatically identify easy and hard tasks, and use the predictions of classifiers from easier tasks to train the classifier for harder tasks.

3. The Easy-to-hard Learning Paradigm for Multiclass Classification

3.1 Classifier Chain for Multiclass Classification

In multi-label classification, each instance can belong to multiple labels simultaneously, while multiclass classification classifies instances into one of more than two classes. Therefore, multiclass classification (Hsu and Lin, 2002) is a quite different learning task compared to multi-label classification (Zhang and Zhou, 2014). Many applications, such as human action recognition and object detection, can be formulated as a multiclass classification problem. However, many confusing classes in such applications will degrade the existing solver's performance. For example, *hand clap* and *boxing* are two confusing actions. *Hand clap* is easily misclassified as boxing, and vice versa. Therefore, it is non-trivial to precisely classify confusing classes for multiclass classification.

To solve these issues, motivated by CC for multi-label classification, we propose a classifier chain model for multiclass classification (CCMC) which aims to transfer class information between classifiers. CCMC trains q binary classifiers h_j ($j \in \{1, \cdots, q\}$), with all of the instances in the j-th class having positive labels, and all other instances having negative labels. Similar to CC, classifiers of CCMC are linked along a chain. CCMC works as follows: binary classifier h_1 is first trained for class λ_1, then the augmented vector $\{\mathbf{x}_t, h_1(\mathbf{x}_t)\}_{t=1}^n$ is used as the input to train classifier h_2 for class λ_2. Similarly, $\mathbf{x}_t$ augments all previous prediction values of $h_1, \cdots, h_j$ as the input to train classifier h_{j+1} for class λ_{j+1}. CCMC proceeds in this way until the last classifier h_q for class λ_q has been trained.

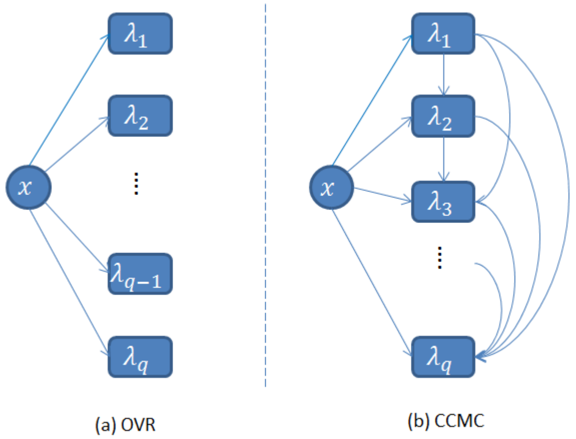


Figure 2: Schematic illustration of OVR and CCMC. A circle represents the input $\mathbf{x}$. A rectangle represents class $\lambda_i(t \in \{1, \cdots, q\})$. The starting point of the arrow denotes the input to train the classifier for the class to which the arrow points.

Given a new testing instance $\mathbf{x}$, classifier h_1 in the chain is responsible for predicting the value for λ_1 using input $\mathbf{x}$. h_2 predicts the value for λ_2 taking $\mathbf{x}$ plus the predicted value of $h_1(\mathbf{x})$ as an input. Similarly, h_{j+1} predicts the value for λ_{j+1} using $\mathbf{x}$ plus all previous prediction results from $h_1, \cdots, h_j$ as the input. CCMC proceeds in this way until the value of the last class λ_q has been predicted. Similar to OVR, the final output of CCMC for $\mathbf{x}$ is the class that corresponds to the classifier with the highest output value. CCMC exploits the class dependence by passing class information between classifiers. Figure 2 illustrates the working scheme between OVR and CCMC.

3.2 Generalized Classifier Chain for Multiclass Classification

We generalize the CCMC model over a random class order, called *generalized classifier chain for multiclass classification* (GCCMC). Assume that classes $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ are randomly reordered as $\{\zeta_1, \zeta_2, \cdots, \zeta_q\}$, where $\zeta_j = \lambda_k$ means class λ_k moves to position j from k. In the GCCMC model, classifiers are also linked along a chain where each classifier h_j deals with the binary classification problem for class ζ_j (λ_k). GCCMC follows the same training and testing procedures as CCMC, the only difference being the class order.

3.3 Analysis

In this subsection, we analyze the generalization error bound of the multiclass classification problem using GCCMC based on some techniques, such as fat shattering dimension (Kearns and Schapire, 1990) and the upper bound theorem of covering numbers (Shawe-Taylor et al., 1998).

Let $\mathbf{X}$ represent the input space. Both $\mathbf{s}$ and $\bar{\mathbf{s}}$ are m multiclass examples drawn independently according to an unknown distribution D. We denote logarithms to base 2 by log. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ denotes its cardinality. $\|\cdot\|$ means the l_2 norm.

We begin with the definition of the fat shattering dimension.

Definition 1 (Kearns and Schapire (1990)) *Let $\mathcal{H}$ be a set of real valued functions. We say that a set of points P is γ-shattered by $\mathcal{H}$ relative to $r = (r_p)_{p \in P}$ if there are real numbers r_p indexed by $p \in P$ such that for all binary vectors b indexed by P, there is a function $f_b \in \mathcal{H}$ satisfying*

$$f_b(p) = \begin{cases} \geq r_p + \gamma & \textit{if } b_p = 1 \\ \leq r_p - \gamma & \textit{otherwise} \end{cases}$$

The fat shattering dimension $fat(\gamma)$ of the set $\mathcal{H}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite, or infinity otherwise.

Assume that $\mathcal{H}$ is the real valued function class and $h \in \mathcal{H}$. $l(y, h(x))$ denotes the loss function. The expected error of h is defined as $er_D[h] = E_{(x,y)\sim D}[l(y, h(x))]$, where (x, y) drawn from the unknown distribution D. Here we select 0-1 loss function. So, $er_D[h] = P_{(x,y)\sim D}(h(x) \neq y)$. The empirical risk $er_{\mathbf{s}}[h]$ is defined as $er_{\mathbf{s}}[h] = \frac{1}{n}\sum_{t=1}^{n} \mathbf{I}\big(y_t \neq h(x_t)\big)$.[3]

Suppose that $\mathcal{N}(\epsilon, \mathcal{H}, \mathbf{s})$ is the ϵ-covering number of $\mathcal{H}$ with respect to the l_∞ pseudo-metric measuring the maximum discrepancy on the example $\mathbf{s}$. The notion of the covering number can be referred to Appendix A. We introduce the following general corollary regarding the bound of the covering number:

Corollary 2 (Shawe-Taylor et al. (1998)) *Let $\mathcal{H}$ be a class of functions $X \to [a, b]$ and D a distribution over X. Choose $0 < \epsilon < 1$ and let $d = fat(\epsilon/4) \leq em$. Then*

$$E(\mathcal{N}(\epsilon, \mathcal{H}, \boldsymbol{s})) \leq 2\Big(\frac{4m(b-a)^2}{\epsilon^2}\Big)^{d\log(2em(b-a)/(d\epsilon))} \tag{1}$$

where the expectation E is over examples $\boldsymbol{s} \in X^m$ drawn according to D^m.

We study the generalization error bound of the specified GCCMC with the specified number of classes and margins. Let G be the set of classifiers of GCCMC, $G = \{h_1, h_2, \cdots, h_q\}$. $er_{\mathbf{s}}[G]$ denotes the fraction of the number of errors that GCCMC makes on $\mathbf{s}$. Define $\hat{\mathbf{x}} \in \mathbf{X} \times \{0, 1\}$, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1 - \mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j)$.

We introduce the following proposition:

Proposition 3 *If an instance $\boldsymbol{x} \in \boldsymbol{X}$ is misclassified by a GCCMC model, then $\exists h_j \in G, \hat{h}_j(\hat{\mathbf{x}}) \geq 0$.*

Proof For multiclass problem, assume that an instance $\mathbf{x}$ belongs to class ζ_i: $\mathbf{y}(i) = 1$, $\mathbf{y}(g) = 0 (\forall g \in \{1, 2, \cdots, q\}, g \neq i)$, and that it is misclassified as ζ_j. Suppose that classifier $h_j \in G$ reports the highest confidence score for this instance: $h_j(\mathbf{x})$.

[3] The expression $\mathbf{I}\big(y_t \neq h(x_t)\big)$ evaluates to 1 if $y_t \neq h(x_t)$ is true and to 0 otherwise.

Case 1: $h_j(\mathbf{x}) \geq 0$. In this case, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1 - \mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j) = h_j(\mathbf{x}) \geq 0$.

Case 2: $h_j(\mathbf{x}_t) < 0$. In this case, all classifiers will output negative real numbers as the result of h_j reporting the highest confidence score. Thus, $\hat{h}_i(\hat{\mathbf{x}}) = h_i(\mathbf{x})(1 - \mathbf{y}(i)) - h_i(\mathbf{x})\mathbf{y}(i) = -h_i(\mathbf{x}) \geq 0$. ■

Lemma 4 *Given a specified GCCMC model with q classes and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each class satisfying $k_i = fat(\gamma^i/8)$, where fat is continuous from the right. If GCCMC has correctly classified m multiclass examples $\boldsymbol{s}$ generated independently according to the unknown (but fixed) distribution D and $\bar{\mathbf{s}}$ is a set of another m multiclass examples, then we can bound the following probability to be less than δ: $P^{2m}\{\boldsymbol{s}\bar{\boldsymbol{s}} : \exists$ a GCCMC model, it correctly classifies $\boldsymbol{s}$, fraction of $\bar{\mathbf{s}}$ misclassified $> \epsilon(m, q, \delta)\} < \delta$, where $\epsilon(m, q, \delta) = \frac{1}{m}(Q \log(32m) + \log \frac{2^q}{\delta})$ and $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$.*

Proof (of Lemma 4). Suppose that G is a GCCMC model with q classes and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$, the probability event in Lemma 4 can be described as

$$A = \{\mathbf{s}\bar{\mathbf{s}} : \exists G, k_i = fat(\gamma^i/8), er_{\mathbf{s}}[G] = 0, er_{\bar{\mathbf{s}}}[G] > \epsilon\}.$$

Let $\hat{\mathbf{s}}$ and $\hat{\bar{\mathbf{s}}}$ denote two different set of m multiclass examples, which are drawn i.i.d. from the distribution $D \times \{0, 1\}$. Applying the definition of $\hat{\mathbf{x}}$, $\hat{h}$ and Proposition 3, the event can also be written as $A = \{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}} : \exists G, \hat{\gamma}^i = \gamma^i/2, k_i = fat(\hat{\gamma}^i/4), er_{\mathbf{s}}[G] = 0, r_i = max_t \hat{h}_i(\hat{\mathbf{x}}_t), 2\hat{\gamma}^i = -r_i, |\{\hat{\mathbf{y}} \in \hat{\bar{\mathbf{s}}} : \exists h_i \in G, \hat{h}_i(\hat{\mathbf{y}}) \geq 2\hat{\gamma}^i + r_i\}| > m\epsilon\}$. Here, $-max_t \hat{h}_i(\hat{\mathbf{x}}_t)$ means the minimal value of $|h_i(\mathbf{x})|$ which represents the margin for class ζ_i, so $2\hat{\gamma}^i = -r_i$. Let $\gamma_{k_i} = min\{\gamma' : fat(\gamma'/4) \leq k_i\}$, so $\gamma_{k_i} \leq \hat{\gamma}^i$, we define the following function:

$$\pi(\hat{h}) = \begin{cases} 0 & \text{if } \hat{h} \geq 0 \\ -2\gamma_{k_i} & \text{if } \hat{h} \leq -2\gamma_{k_i} \\ \hat{h} & \text{otherwise} \end{cases}$$

so $\pi(\hat{h}) \in [-2\gamma_{k_i}, 0]$. Let $\pi(\hat{G}) = \{\pi(\hat{h}) : h \in G\}$.

Let $B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$ represent the minimal γ_{k_i}-cover set of $\pi(\hat{G})$ in the pseudo-metric $d_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$. We have that for any $h_i \in G$, there exists $\tilde{f} \in B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$, $|\pi(\hat{h}_i(\hat{\mathbf{z}})) - \pi(\tilde{f}(\hat{\mathbf{z}}))| < \gamma_{k_i}$, for all $\hat{\mathbf{z}} \in \hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}$. For all $\hat{\mathbf{x}} \in \hat{\mathbf{s}}$, by the definition of r_i, $\hat{h}_i(\hat{\mathbf{x}}) \leq r_i = -2\hat{\gamma}^i$, and $\gamma_{k_i} \leq \hat{\gamma}^i$, $\hat{h}_i(\hat{\mathbf{x}}) \leq -2\gamma_{k_i}$, $\pi(\hat{h}_i(\hat{\mathbf{x}})) = -2\gamma_{k_i}$, so $\pi(\tilde{f}(\hat{\mathbf{x}})) < -2\gamma_{k_i} + \gamma_{k_i} = -\gamma_{k_i}$. However, there are at least $m\epsilon$ points $\hat{\mathbf{y}} \in \hat{\bar{\mathbf{s}}}$ such that $\hat{h}_i(\hat{\mathbf{y}}) \geq 0$, so $\pi(\tilde{f}(\hat{\mathbf{y}})) > -\gamma_{k_i} > max_t \pi(\tilde{f}(\hat{\mathbf{x}}_t))$. Since π only reduces the separation between output values, we conclude that the inequality $\tilde{f}(\hat{\mathbf{y}}) > max_t \tilde{f}(\hat{\mathbf{x}}_t)$ holds. Moreover, the $m\epsilon$ points in $\hat{\bar{\mathbf{s}}}$ with the largest $\tilde{f}$ values must remain for the inequality to hold. By the permutation argument, at most $2^{-m\epsilon}$ of the sequences obtained by swapping corresponding points satisfy the conditions for fixed $\tilde{f}$.

As for any $h_i \in G$, there exists $\tilde{f} \in B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$, so there are $|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|$ possibilities of $\tilde{f}$ that satisfy the inequality for k_i. Note that $|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|$ is a positive integer which is usually bigger than 1, and by the union bound, we obtain the following inequality:

$$P(A) \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) + \cdots + E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon}$$

Since every set of points γ-shattered by $\pi(\hat{G})$ can be γ-shattered by $\hat{G}$, so $fat_{\pi(\hat{G})}(\gamma) \leq fat_{\hat{G}}(\gamma)$, where $\hat{G} = \{\hat{h} : h \in G\}$. Hence, by Corollary 2 (setting $[a, b]$ to $[-2\gamma_{k_i}, 0]$, ϵ to γ_{k_i} and m to $2m$),

$$E(|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) = E(\mathcal{N}(\gamma_{k_i}, \pi(\hat{G}), \hat{\mathbf{s}}\hat{\bar{\mathbf{s}}})) \leq 2(32m)^{d\log(\frac{8em}{d})}$$

where $d = fat_{\pi(\hat{G})}(\gamma_{k_i}/4) \leq fat_{\hat{G}}(\gamma_{k_i}/4) \leq k_i$. Thus $E(|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \leq 2(32m)^{k_i \log(\frac{8em}{k_i})}$, and we obtain

$$P(A) \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} \leq \prod_{i=1}^{q} 2(32m)^{k_i \log(\frac{8em}{k_i})} = 2^q(32m)^Q$$

where $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$. And so $(E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} < \delta$ provided

$$\epsilon(m, q, \delta) \geq \frac{1}{m}\Big(Q\log(32m) + \log\frac{2^q}{\delta}\Big)$$

as required. ■

Lemma 4 applies to a particular GCCMC model with a specified number of classes and a specified margin for each class. In practice, we will observe the margins after running the GCCMC model. Thus, we must bound the probabilities uniformly over all of the possible margins to obtain a practical bound. The generalization error bound of the multiclass classification problem using GCCMC is shown as follows:

Theorem 5 *Suppose that random m multiclass examples can be correctly classified using a GCCMC model, and suppose this GCCMC model contains q classifiers with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each class. Then we can bound the generalization error with probability greater than $1 - \delta$ to be less than*

$$\frac{130R^2}{m}\Big(Q'\log(8em)\log(32m) + \log\frac{2(2m)^q}{\delta}\Big)$$

where $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma^i)^2}$ and R is the radius of a ball containing the support of the distribution.

Before proving Theorem 5, we state one key symmetrization lemma and Theorem 7.

Lemma 6 (Symmetrization) *Let $\mathcal{H}$ be the real valued function class. $\boldsymbol{s}$ and $\bar{\boldsymbol{s}}$ are m examples both drawn independently according to the unknown distribution D. If $m\epsilon^2 \geq 2$, then*

$$P_{\boldsymbol{s}}(\sup_{h\in\mathcal{H}} |er_D[h] - er_{\boldsymbol{s}}[h]| \geq \epsilon) \;\leq\; 2P_{\boldsymbol{s}\bar{\boldsymbol{s}}}(\sup_{h\in\mathcal{H}} |er_{\bar{\boldsymbol{s}}}[h] - er_{\boldsymbol{s}}[h]| \geq \epsilon/2) \tag{2}$$

The proof of this lemma can be found in Appendix B.

Theorem 7 (Bartlett and Shawe-Taylor (1998)) *Let $\mathcal{H}$ be restricted to points in a ball of $\boldsymbol{M}$ dimensions of radius R about the origin, then*

$$fat_{\mathcal{H}}(\gamma) \leq \min\Big\{\frac{R^2}{\gamma^2}, \boldsymbol{M} + 1\Big\} \tag{3}$$

Proof (of Theorem 5). We must bound the probabilities over different margins. We first use Lemma 6 to bound the probability of error in terms of the probability of the discrepancy between the performance on two halves of a double example. Then we combine this result with Lemma 4. We must consider all possible patterns of k_i's for ζ_i. The largest value of k_i is m. Thus, for fixed q, we can bound the number of possibilities by m^q. Hence, there are m^q of applications of Lemma 4.

Let $c_i = \{\gamma^1, \gamma^2, \cdots, \gamma^q\}$ denote the i-th combination of margins varied in $\{1, \cdots, m\}^q$. $\mathcal{G}$ denotes a set of GCCMC models. The generalization error of G can be represented as $er_D[G]$ and $er_{\mathbf{s}}[G]$ is 0, where $G \in \mathcal{G}$. The uniform convergence bound of the generalization error is

$$P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \geq \epsilon)$$

Applying Lemma 6,

$$P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G]-er_{\mathbf{s}}[G]| \geq \epsilon) \leq 2P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2)$$

Let $J_{c_i} = \{\mathbf{s}\bar{\mathbf{s}} : \exists$ a GCCMC model G with q classes and with margins $c_i : k_i = fat(\gamma^i/8), er_{\mathbf{s}}[G] = 0, er_{\bar{\mathbf{s}}}[G] \geq \epsilon/2\}$. Clearly,

$$P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2) \leq P^{m^q}\Big(\bigcup_{i=1}^{m^q} J_{c_i}\Big)$$

As k_i still satisfies $k_i = fat(\gamma^i/8)$, Lemma 4 can still be applied to each case of $P^{m^q}(J_{c_i})$. Let $\delta_k = \delta/m^q$. Applying Lemma 4 (replacing δ by $\delta_k/2$), we get:

$$P^{m^q}(J_{c_i}) < \delta_k/2$$

where $\epsilon(m, k, \delta_k/2) \geq 2/m(Q\log(32m) + \log\frac{2\times 2^q}{\delta_k})$ and $Q = \sum_{i=1}^q k_i \log(\frac{4em}{k_i})$. It suffices to show by the union bound that $P^{m^q}(\bigcup_{i=1}^{m^q} J_{c_i}) \leq \sum_{i=1}^{m^q} P^{m^q}(J_{c_i}) < \delta_k/2 \times m^q = \delta/2$. Applying Lemma 6,

$$\begin{aligned} P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \geq \epsilon) &\leq 2P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2) \\ &\leq 2P^{m^q}\Big(\bigcup_{i=1}^{m^q} J_{c_i}\Big) < \delta \end{aligned}$$

Thus, $P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \leq \epsilon) \geq 1 - \delta$. Let R be the radius of a ball containing the support of the distribution. Applying Theorem 7, we get $k_i = fat(\gamma^i/8) \leq 65R^2/(\gamma^i)^2$. Note that we have replaced the constant $8^2 = 64$ by 65 in order to ensure the continuity from the right required for the application of Lemma 4. We have upperbounded $\log(8em/k_i)$ by $\log(8em)$. Thus,

$$\begin{aligned} er_D[G] &\leq 2/m\Big(Q\log(32m) + \log\frac{2(2m)^q}{\delta}\Big) \\ &\leq \frac{130R^2}{m}\Big(Q'\log(8em)\log(32m) + \log\frac{2(2m)^q}{\delta}\Big) \end{aligned}$$

where $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma^i)^2}$. ■

Given the training data size and the number of classes, Theorem 5 reveals an important factor in reducing the generalization error bound for the GCCMC model: the minimization of the sum of the reciprocal of the square of the margin over the classes. Thus, we obtain the following Corollary:

Corollary 8 (Globally Optimal Classifier Chain for Multiclass Classification) *Suppose that random m multiclass examples with q classes can be correctly classified using a GCCMC model, this GCCMC model is the globally optimal classifier chain if and only if the minimization of Q' in Theorem 5 is achieved over this classifier chain.*

Based on Corollary 8, we propose the following easy-to-hard learning paradigm for multiclass classification problems.

Definition 9 (Easy-to-hard Learning Paradigm for Multiclass Classification) *The easy-to-hard learning paradigm for multiclass classification problem is to minimize Q' in Theorem 5. By minimizing Q', we can automatically identify easy and hard classes.*

Remark. Classes with a larger margin are easier to identify than those with a smaller margin. Thus, the intuitive idea of the easy-to-hard learning paradigm is to identify the class with a larger margin first, followed by ones with a smaller margin.

Discussion of Rifkin and Klautau's Conjecture. Rifkin and Klautau (2004) speculate that an algorithm which exploits the relationship between classes can offer superior performance, and this remains an open problem. Theoretically, Corollary 10 provides an affirmative answer to Rifkin and Klautau (2004)'s conjecture based on Theorem 5:

Corollary 10 *By exploiting the relationship between classes, our proposed GCCMC model is able to achieve a lower generalization error bound. Furthermore, our proposed easy-to-hard learning paradigm can optimize the performance of GCCMC.*

Motivated by the above analysis, we derive the following easy-to-hard learning paradigm for multi-label classification.

4. The Easy-to-hard Learning Paradigm for Multi-label Classification

4.1 Classifier Chain for Multi-label Classification

Similar to CCMC, the *classifier chain* (CC) model (Read et al., 2009) is proposed to train q binary classifiers h_j ($j \in \{1, \cdots, q\}$) for multi-label problems. Classifiers are linked along a chain where each classifier h_j deals with the binary classification problem for label λ_j. The augmented vector $\{\mathbf{x}_t, \mathbf{y}_t(1), \cdots, \mathbf{y}_t(j)\}_{t=1}^{n}$ is used as the input for training classifier h_{j+1}. Given a new testing instance $\mathbf{x}$, classifier h_1 in the chain is responsible for predicting the value of $\mathbf{y}(1)$ using input $\mathbf{x}$. Then, h_2 predicts the value of $\mathbf{y}(2)$ taking $\mathbf{x}$ plus the predicted value of $\mathbf{y}(1)$ as an input. Following in this way, h_{j+1} predicts $\mathbf{y}(j+1)$ using the predicted value of $\mathbf{y}(1), \cdots, \mathbf{y}(j)$ as additional input information. CC passes label information between classifiers, allowing CC to exploit the label dependence and thus overcome the label independence problem of BR.

Different classifier chains involve different classifiers learned on different training sets and thus the order of the chain itself clearly affects the prediction performance. To solve the issue of selecting a chain order for CC, Read et al. (2009) propose the extension of CC, called *ensembled classifier chain* (ECC), to average the multi-label predictions of CC over a set of random chain ordering. ECC first randomly reorders the labels $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ many times. Then, CC is applied to the reordered labels for each time and the performance of CC is averaged over those times to obtain the final prediction performance.

4.2 Generalized Classifier Chain for Multi-label Classification

We generalize the CC model over a random label order, called *generalized classifier chain* (GCC) model. Assume the labels $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ are randomly reordered as $\{\zeta_1, \zeta_2, \cdots, \zeta_q\}$, where $\zeta_j = \lambda_k$ means label λ_k moves to position j from k. In the GCC model, classifiers are also linked along a chain where each classifier h_j deals with the binary classification problem for label ζ_j (λ_k). GCC follows the same training and testing procedures as CC, while the only difference is the label order. In the GCC model, for input $\mathbf{x}_t$, $\mathbf{y}_t(j) = 1$ if and only if $\zeta_j \in \mathcal{Y}_t$.

4.3 Analysis

Motivated by our analysis in Section 3, we analyze the generalization error bound of the multi-label classification problem using GCC.

Let $\mathbf{X}$ represent the input space. Both $\mathbf{s}$ and $\bar{\mathbf{s}}$ are m multi-label examples drawn independently according to an unknown distribution D.

We first study the generalization error bound of the specified GCC with the specified number of labels and margins. Let G be the set of classifiers of GCC, $G = \{h_1, h_2, \cdots, h_q\}$. $er_{\mathbf{s}}[G]$ denotes the fraction of the number of errors that GCC makes on $\mathbf{s}$. Define $\hat{\mathbf{x}} \in \mathbf{X} \times \{0, 1\}$, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1 - \mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j)$.

We introduce the following proposition:

Proposition 11 *If an instance $\boldsymbol{x} \in \boldsymbol{X}$ is misclassified by a GCC model, then $\exists h_j \in G, \hat{h}_j(\hat{\mathbf{x}}) \geq 0$.*

Proof For multi-label problem, it is easy to verify that if an instance $\mathbf{x} \in \mathbf{X}$ is correctly classified by h_j, then $\hat{h}_j(\hat{\mathbf{x}}) < 0$, otherwise, $\hat{h}_j(\hat{\mathbf{x}}) \geq 0$. ■

Lemma 12 *Given a specified GCC model with q labels and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each label satisfying $k_i = fat(\gamma^i/8)$, where fat is continuous from the right. If GCC has correctly classified m multi-labeled examples $\boldsymbol{s}$ generated independently according to the unknown (but fixed) distribution D and $\bar{\mathbf{s}}$ is a set of another m multi-labeled examples, then we can bound the following probability to be less than δ: $P^{2m}\{\boldsymbol{s}\bar{\mathbf{s}} : \exists$ a GCC model, it correctly classifies $\boldsymbol{s}$, fraction of $\bar{\mathbf{s}}$ misclassified $> \epsilon(m, q, \delta)\} < \delta$, where $\epsilon(m, q, \delta) = \frac{1}{m}(Q \log(32m) + \log \frac{2^q}{\delta})$ and $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$.*

The proof can be adapted from the proof for Lemma 4.

Based on Lemma 12, we can bound the probabilities uniformly over all of the possible margins to obtain a practical bound. The generalization error bound of the multi-label classification problem using GCC is shown as follows:

Theorem 13 *Suppose that random m multi-labeled examples can be correctly classified using a GCC model, and suppose this GCC model contains q classifiers with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each label. Then we can bound the generalization error with probability greater than $1-\delta$ to be less than*

$$\frac{130R^2}{m}\Big(Q' \log(8em)\log(32m) + \log\frac{2(2m)^q}{\delta}\Big)$$

where $Q' = \sum_{i=1}^{q}\frac{1}{(\gamma^i)^2}$ and R is the radius of a ball containing the support of the distribution.

The proof can be adapted from the proof for Theorem 5.

Theorem 13 reveals an important factor in reducing the generalization error bound for the GCC model: the minimization of the sum of the reciprocal of the square of the margin over the labels, given the training data size and the number of labels. Thus, we obtain the following Corollary:

Corollary 14 (Globally Optimal Classifier Chain for Multi-label Classification) *Suppose that random m multi-labeled examples with q labels can be correctly classified using a GCC model, this GCC model is the globally optimal classifier chain if and only if the minimization of Q' in Theorem 13 is achieved over this classifier chain.*

Based on Corollary 14, we propose the following easy-to-hard learning paradigm for multi-label classification.

Definition 15 (Easy-to-hard Learning Paradigm for Multi-label Classification) *The easy-to-hard learning paradigm for multi-label classification problem is to minimize Q' in Theorem 13. By minimizing Q', we can automatically identify easy and hard labels.*

Discussion of Label's Relationship. Recently, many works, such as Read et al. (2009) and Guo and Schuurmans (2011), have conducted extensive experiments to show that multi-label learning methods which explicitly capture the label's relationship will usually achieve better prediction performance. However, to the best of our knowledge, very few works study the reasons behind these promising empirical results. Based on Theorem 13, Corollary 16 provides theoretical support for this problem:

Corollary 16 *By exploiting the relationship between labels, our proposed GCC model is able to achieve a lower generalization error bound. Furthermore, our proposed easy-to-hard learning paradigm can optimize the performance of GCC.*

Given the number of classes or labels q, there are $q!$ different class or label orders. It is very expensive to find the globally optimal CCMC or CC, which can minimize Q', by searching over all of the class or label orders. Next, we discuss some simple easy-to-hard learning algorithms.

5. Easy-to-hard Learning Algorithm

In this section, we propose some simple easy-to-hard learning algorithms. To clearly state the algorithms, we redefine the margins with class or label order information. Given class or label set $\mathcal{M} = \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$. Let $o_i (1 \leq o_i \leq q)$ denote the order of λ_i in the GCCMC or GCC model, $\gamma_i^{o_i}$ represents the margin for λ_i, with previous $o_i - 1$ classes or labels as the augmented input. If $o_i = 1$, then γ_i^1 represents the margin for λ_i, without augmented input. Then Q' is redefined as $Q' = \sum_{i=1}^q \frac{1}{(\gamma_i^{o_i})^2}$.

5.1 Dynamic Programming Algorithm

To simplify the search algorithm mentioned before, we propose the CCMC-DP and CC-DP algorithm to find the globally optimal CCMC and CC, respectively. Note that $Q' = \sum_{i=1}^q \frac{1}{(\gamma_i^{o_i})^2} = \frac{1}{(\gamma_q^{o_q})^2} + \cdots + \big[\frac{1}{(\gamma_{k+1}^{o_{k+1}})^2} + \sum_{j=1}^k \frac{1}{(\gamma_j^{o_j})^2}\big]$, we explore the idea of dynamic programming (DP) to iteratively optimize Q' over a subset of $\mathcal{M}$ with the length of $1, 2, \cdots, q$. Lastly, we obtain the optimal Q' over $\mathcal{M}$. Assume $i \in \{1, \cdots, q\}$. Let $V(i, \eta)$ be the optimal Q' over a subset of $\mathcal{M}$ with the length of $\eta (1 \leq \eta \leq q)$, where the class or label order ends by λ_i. M_i^η represents the corresponding class or label set for $V(i, \eta)$. When $\eta = q$, $V(i, q)$ is the optimal Q' over $\mathcal{M}$, where the class or label order ends by λ_i. The DP equation is written as:

$$V(i, \eta + 1) = \min_{j \neq i, \lambda_i \notin M_j^\eta} \left\{ \frac{1}{(\gamma_i^{\eta+1})^2} + V(j, \eta) \right\} \tag{4}$$

where $\gamma_i^{\eta+1}$ is the margin for λ_i, with M_j^η as the augmented input. The initial condition of DP is: $V(i, 1) = \frac{1}{(\gamma_i^1)^2}$ and $M_i^1 = \{\lambda_i\}$. The optimal Q' over $\mathcal{M}$ can be obtained by solving $\min_{i \in \{1, \cdots, q\}} V(i, q)$. Assume that the training time of linear SVM takes $\mathcal{O}(nd)$. The CCMC-DP or CC-DP algorithm is shown as the following bottom-up procedure: from the bottom, we first compute $V(i, 1) = \frac{1}{(\gamma_i^1)^2}$, which takes $\mathcal{O}(nd)$. Then we compute $V(i, 2) = \min_{j \neq i, \lambda_i \notin M_j^1} \{\frac{1}{(\gamma_i^2)^2} + V(j, 1)\}$, which requires at most $\mathcal{O}(qnd)$, and set $M_i^2 = M_j^1 \cup \{\lambda_i\}$. Similarly, it takes at most $\mathcal{O}(q^2 nd)$ time complexity to calculate $V(i, q)$. Lastly, we iteratively solve this DP equation, and use $\min_{i \in \{1, \cdots, q\}} V(i, q)$ to obtain the optimal solution, which requires at most $\mathcal{O}(q^3 nd)$ time complexity.

Theorem 17 (Correctness of DP) *Q' can be minimized by CCMC-DP or CC-DP, which means this Algorithm can find the globally optimal CCMC or CC.*

The proof can be found in Appendix C.

5.2 Greedy Algorithm

We propose the CCMC-Greedy and CC-Greedy algorithm to find a locally optimal CCMC and CC, respectively. To save time, we construct only one classifier chain with the locally optimal class or label order. If the maximum margin can be achieved over this class or label, without augmented input, we select this class or label from $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ as the first class or label. The first class or label is denoted by ζ_1. We then select the class or label from the remaining classes or labels as the second class or label, if the maximum margin can be achieved over this class or label with ζ_1 as the augmented input. We continue in this way until the last class or label has been selected. CCMC-Greedy and CC-Greedy take $\mathcal{O}(q^2nd)$ time, respectively. We show the details of the CC-Greedy algorithm in Appendix D.

5.3 Fast Greedy Algorithm

In multiclass problems, CCMC-DP and CCMC-Greedy are intractable for data sets with a large number of classes. To further speed up the CCMC-Greedy algorithm, we propose fast greedy algorithm (CCMC-FG), which scales linearly with q, to greedily optimize the order of the top ω classes. Similar to CCMC-Greedy, if the maximum margin can be achieved over this class without augmented input, we select this class from $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ as the first class. The first class is denoted by ζ_1. Then we select the class from the remaining classes as the second class, if the maximum margin can be achieved over this class with prediction values of the classifier trained for class ζ_1 as the augmented input. We continue in this way until the ω class has been selected. Lastly, we use the remaining $q - \omega$ classes to form the classifier chain. CCMC-FG takes $\mathcal{O}(q\omega nd)$ time.

Remark. CCMC-Greedy converges to the locally optimal CCMC, while CCMC-FG finds the top ω locally optimal class order.

5.4 Tree-Based Algorithm

For multi-label problem, CC-DP and CC-Greedy are very time-consuming for data sets with many labels. We propose Tree-DP and Tree-Greedy algorithms to further speed up CC-DP and CC-Greedy, respectively, which scale linearly with q. We create a tree recursively in a top-down manner.

Assume that $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^n$ is the input data for the root node. Suppose that the label set $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ in the root node is randomly split into two subsets with about the same size for left and right child nodes: $leftset = \{\lambda_1, \cdots, \lambda_{q/2}\}$ and $rightset = \{\lambda_{q/2+1}, \cdots, \lambda_q\}$. A training example can be considered annotated with $leftset$ and $rightset$ if it is annotated with at least one of the labels in $leftset$ and $rightset$, respectively. In this way, $leftset$ and $rightset$ can be seen as two labels. Then, in the root node, we train the CC with $leftset$ and $rightset$ as two labels using CC-DP or CC-Greedy. After that, left and right child nodes only keep the examples that are annotated with $leftset$ and $rightset$, respectively. This approach recurses into each child node that contains more than a single label.

Starting from the root node, we use the trained CC classifier on this node for prediction and we follow the recursive process. Finally, this process may lead to the prediction of some labels corresponding to some leaves. We provide the following corollary pertaining to the Tree-DP.

Corollary 18 *After building a tree using the Tree-DP algorithm, we can find the globally optimal CC in each decision node of the tree.*

Proof Given the structure of the tree, according to Theorem 17, we can find the globally optimal label order in each decision node using the CC-DP algorithm. ■

For each internal node, we only deal with two labels, thus the training time of Tree-DP and Tree-Greedy only take $\mathcal{O}(8nd)$ and $\mathcal{O}(4nd)$, respectively. The number of internal nodes in such a tree is equal to $q-1$. In total, Tree-DP and Tree-Greedy take $\mathcal{O}(8(q-1)nd)$ and $\mathcal{O}(4(q-1)nd)$ training time, respectively. Assume that the testing instance goes along ι paths in our tree during the testing procedure and the depth of the tree is $\log(q)$. In each decision node, we take $\mathcal{O}(d)$ time for testing. Totally, the testing time for Tree-DP and Tree-Greedy is $\mathcal{O}(\iota \log(q) d)$.

6. Applications

This section shows that our framework can be used for various applications, such as ordinal classification and relationship prediction.

6.1 Ordinal Classification

Many practical applications involve situations exhibiting an order among the different categories. For example, a user rates movies by giving them grades based on quality. These grades represent the ranking information. For example, grade classes are ordered as $D < C < B < A$. This is a learning task for predicting ordinal classes, referred to as ordinal classification (Seah et al., 2012).

Several algorithms and methods have been developed to deal with ordinal classification, such as SVM techniques (Shashua and Levin, 2002), binary decomposition (Destercke and Yang, 2014), Gaussian processes (Chu and Ghahramani, 2005) and monotone functions (Tehrani et al., 2012). However, all these methods do not capture and use correlated information between ordinal classes. To achieve this goal, we transform ordinal classification into multiclass classification and then apply CCMC for ordinal classification.

Consider an ordinal classification problem with q ordered categories. We denote these categories as $\mathcal{Y}_t \subseteq \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ to keep the known ordering information. $\mathbf{y}_t \in \{0,1\}^q$ is used to represent the set $\mathcal{Y}_t$, where $\mathbf{y}_t(j) = 1$ if and only if $\lambda_j \in \mathcal{Y}_t$, and there is only one element in $\mathcal{Y}_t$ for the ordinal classification problem. So, we transform ordinal classification into a multiclass classification problems. Then, we apply CCMC for ordinal classification as follows: the binary classifier h_1 is first trained for the ordinal class λ_1, then the augmented vector $\{\mathbf{x}_t, h_1(\mathbf{x}_t)\}_{t=1}^n$ is used as the input to train classifier h_2 for ordinal class λ_2. Similarly, $\mathbf{x}_t$ augments all previous prediction values of $h_1, \cdots, h_j$ as the input to train classifier h_{j+1} for ordinal class λ_{j+1}. CCMC proceeds in this way until the last classifier h_q for ordinal class λ_q has been trained.

6.2 Relationship Prediction

Relationship prediction problems in the online review website Epinions (Massa and Avesani, 2006) attempt to predict whether people trust or distrust others based on their reviews. Such social networks can be modeled as a signed network where trust/distrust are modeled as positive/negative edges between entities (Leskovec et al., 2010). The problem then becomes predicting unknown relationship between any two users given the network.

Many approaches, such as Hsieh et al. (2012) and Chiang et al. (2014), perform matrix completion on an adjacency matrix and then use the sign of the completed matrix for relationship prediction. Recently, Chiang et al. (2015) achieve state-of-the-art performance by incorporating the feature information of users. Based on feature information, we provide new insight into the design of relationship prediction algorithms. Specifically, we first transform relationship prediction into a multi-label classification problem by considering trust/distrust as positive/negative labels. Then, we apply our proposed easy-to-hard learning strategy to solve relationship prediction tasks.

Relationship prediction is represented as a graph with the adjacency matrix $R \in \{0, 1\}^{n \times n}$, which denotes relationships between users as follows:

$$R_{ij} = \begin{cases} 1, & \text{if user } i \text{ and user } j \text{ have positive relationship;} \\ 0, & \text{if user } i \text{ and user } j \text{ have negative relationship} \end{cases}$$

We assume that user i and user i have a positive relationship. The attribute information of user i can be extracted as the input or instance (feature) $\mathbf{x}_i$, and the i-th row vector of R can be used as label $\mathbf{y}_i$. So, we transform relationship prediction into a multi-label classification problem. Then, CC model can be used to deal with the transformed problems, and we can apply our proposed easy-to-hard learning strategy to solve relationship prediction tasks.

7. Experiment

In this section, we perform experimental studies on a number of real world data sets to evaluate the performance of our proposed algorithms for multiclass and multi-label classification problems. To perform a fair comparison, we use the same linear classification/regression package LIBLINEAR (Fan et al., 2008) with L2-regularized square hinge loss (primal) to train the classifiers for all methods and use the default parameter settings in LIBLINEAR. All experiments are conducted on a workstation with a 3.4GHZ Intel CPU and 32GB main memory running on a Linux platform.

7.1 Experiment on Multiclass Classification

In this subsection, we first demonstrate our motivation on the recognition of human action (Schüldt et al., 2004). We then consider a variety of benchmark multiclass data sets without background from the LIBSVM website[4] to evaluate the performance of the proposed algorithms for multiclass classification. Lastly, we conduct experiments on two multiclass data sets with background collected from Silberman et al. (2012) and He et al. (2004). The training/testing partition is either predefined or the data is randomly split into 80% training and 20% testing. The statistics of each data set are reported in Table 1. We compare our algorithms with some baseline methods: OVO, OVR, ECOC (Dietterich and Bakiri, 1995) and Top-k SVM (Lapin et al., 2015, 2016). The library for ECOC is from Pedregosa et al. (2011) and the size of code for ECOC is selected using 5-fold cross validation over the range $\{2, 10, 30, 50\}$. Top-k SVM (Lapin et al., 2015, 2016) is one of the state-of-the-art generalized multiclass SVM for top-k error optimization. The code is provided by their authors. Following the similar parameter settings in Lapin et al. (2015, 2016) for Top-k SVM, k is selected using 5-fold cross validation over the range $\{1, 3, 5, 10\}$.

[4] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets

Table 1: Multiclass data sets used in the experiments.

Data set	# TRAINING	# TESTING	# Classes	Background
KTH	1,910	477	6	×
IRIS	120	30	3	×
SEGMENT	1,848	4,62	7	×
SEISMIC	78,823	19,705	3	×
RCV1	15,564	518,571	51	×
ALOI	86,400	21,600	1,000	×
ILSVRC2012	58,700	31,300	1,000	×
NYU	60,232	11,768	5	√
SOWERBY	528,356	110,620	6	√

Table 2: Testing error rate (in %) on the KTH data set.

OVO	OVR	ECOC	Top-k SVM	CCMC	CCMC-Greedy	CCMC-DP
8.81	8.18	7.73	8.13	7.34	6.29	5.66

7.1.1 Human action recognition

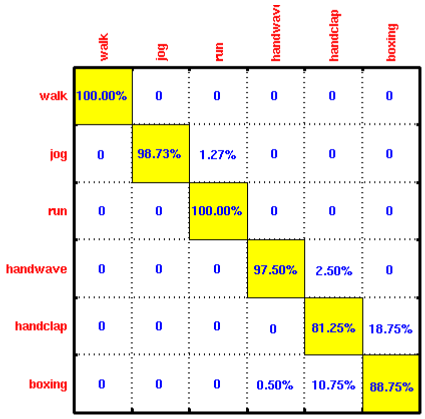
We first validate our methods by recognizing complex human actions on the KTH data set Schüldt et al. (2004). KTH contains six types of human action: *walk*, *jog*, *run*, *hand wave*, *hand clap* and *boxing*. The confusion matrix of OVR on the KTH data set is shown in Figure 1. From Figure 1, we can see that it is easy to identify *walk* from other actions, whereas it is difficult to distinguish between *boxing* and *hand clap*. This observation is consistent with commonsense. Our model aims to classify classes from simple to hard; the classifier of the hard actions will benefit significantly from the prediction of the simple actions.

The testing error rates of the different methods are shown in Table 2. The order of actions identified by CCMC-Greedy is: *run*, *hand wave*, *walk*, *jog*, *boxing* and *hand clap*. CCMC-DP finds the action order: *walk*, *run*, *jog*, *hand wave*, *boxing* and *hand clap*. From these results, we observe that:

- For the KTH data set, our methods achieve a better prediction performance than OVO, OVR, ECOC and Top-k SVM, which verifies that our model effectively uses the predictions of previous classifiers to improve the performance of OVR.

- CCMC-Greedy and CCMC-DP improve CCMC, which also verifies our motivation: we should classify the classes from easy to hard. The order of actions (from easy to hard) found by CCMC-Greedy and CCMC-DP are consistent with the confusion matrix, which demonstrates that our model can automatically identify easy and hard classes, and use the predictions of classifiers from easier classes to train the classifier for harder classes.

Figure 3 shows the confusion matrix of CCMC-DP on the KTH data set. By comparing Figure 1 with Figure 3, we observe that our method significantly improves OVR in terms of the classification accuracy rate for each action. For example, we achieve 100% accuracy for *run* and improve the accuracy for *hand clap* from 77.50% to 81.25%.



	walk	jog	run	handwav	handclap	boxing
walk	100.00%	0	0	0	0	0
jog	0	98.73%	1.27%	0	0	0
run	0	0	100.00%	0	0	0
handwave	0	0	0	97.50%	2.50%	0
handclap	0	0	0	0	81.25%	18.75%
boxing	0	0	0	0.50%	10.75%	88.75%

Figure 3: Confusion Matrix of CCMC-DP on the KTH Data Set.

7.1.2 Results without background data

We compare OVO, OVR and ECOC with our algorithms on the benchmark data sets without background. The classification results for our methods and baseline approaches on the IRIS, SEGMENT, SEISMIC and RCV1 data sets are reported in Table 3. Based on these results, we make the following observations.

- Our results show that OVR and OVO perform as accurate as ECOC and Top-k SVM, which is consistent with the empirical results in Rifkin and Klautau (2004), Demirkesen and Cherifi (2008) and Lapin et al. (2015, 2016).

- CCMC consistently improves the prediction performance of OVR and other baselines on all data sets. The results verify Rifkin and Klautau (2004)'s conjecture: an algorithm which exploits the relationship between classes can offer superior performance.

- When OVO outperforms OVR on certain data sets such as SEISMIC and SEGMENT, our methods are able to achieve superior prediction performance to OVO on these data sets.

- CCMC-Greedy is better than CCMC, and CCMC-DP outperforms CCMC-Greedy and CCMC. The results validate our theoretical analysis: i) Class order affects the performance of CCMC. ii) CCMC-DP is able to find the globally optimal CCMC which achieves the best prediction performance compared to CCMC-Greedy and CCMC. iii) The CCMC-Greedy algorithm achieves comparable prediction performance with CCMC-DP.

We also evaluate the performance of our fast greedy algorithm on the ALOI data set, which contains 1,000 classes. Here, ω is set to 10, 30, 50 and 100. The classification results and training time are reported in Table 4, from which we can see that CCMC-FG outperforms CCMC and improves the prediction performance of OVR, ECOC and Top-k SVM by about 9%, 6% and 5%, respectively. The prediction performance of CCMC-FG improves with the increased value of ω.

Table 3: Testing error rate (in %) on data sets without background.

						CCMC-	
Data set	OVO	OVR	ECOC	Top-k SVM	CCMC	Greedy	DP
IRIS	10.00	10.00	8.67	10.33	3.33	3.33	3.33
SEGMENT	5.63	7.14	7.08	7.06	6.28	5.63	5.19
SEISMIC	27.92	29.87	27.62	28.19	26.85	26.50	26.48
RCV1	11.09	11.90	11.83	11.98	11.78	11.71	11.68

Table 4: Testing error rate (in %) and training time (in second) on the ALOI data set.

Method	Testing Error	Training Time
OVO	6.88	14,748s
OVR	13.69	1,559s
ECOC	10.31	85,316s
Top-k SVM	9.4	65s
CCMC	5.46	12,715s
CCMC-FG($\omega = 10$)	5.27	16,327s
CCMC-FG($\omega = 30$)	5.08	31,886s
CCMC-FG($\omega = 50$)	4.94	67,582s
CCMC-FG($\omega = 100$)	4.78	163,891s

7.1.3 Results with background data

The OVO, ECOC and Top-k SVM approaches cannot be directly applied to background data. Table 5 shows the classification results for our methods and OVR on the NYU and SOWERBY data sets. From the results of Table 5, we can see that:

- CCMC consistently outperforms OVR.
- CCMC-Greedy achieves better prediction performance than CCMC and is comparable to CCMC-DP.
- CCMC-DP improves the prediction performance of OVR on the data sets with background by 3%.

7.1.4 Training time

This section studies the training time of the proposed methods and baselines on all data sets. The results are shown in Tables 4 and 6. From these results, we can see that:

- Top-k SVM is much faster than other methods on the ALOI data set with 1000 classes.
- Compared to OVR, CCMC maintains the training time over an acceptable threshold, while CCMC consistently improves OVR.

Table 5: Testing error rate (in %) on data sets with background.

			CCMC-	
Data set	OVR	CCMC	Greedy	DP
NYU	21.07	20.61	19.39	18.82
SOWERBY	18.03	16.27	15.54	15.24

Table 6: Training time (in second).

							CCMC-
Data set	OVO	OVR	ECOC	Top-k SVM	CCMC	Greedy	DP
KTH	0.09s	0.20s	0.16s	0.13s	0.24s	0.60s	1.37s
IRIS	0.009s	0.008s	0.008s	0.02s	0.012s	0.019s	0.019s
SEGMENT	0.04s	0.05s	0.30s	0.06s	0.32s	1.10s	5.85s
SEISMIC	2.94s	4.60s	167.28s	2.23s	22.08s	32.88s	98.28s
RCV1	56.06s	18.16s	31.15s	346.79s	30.27s	982.89s	16,450.55s
NYU	N/A	4.74s	N/A	N/A	4.58s	14.46s	83.75s
SOWERBY	N/A	23.63s	N/A	N/A	39.73s	105.42s	558.02s

- CCMC-Greedy is much faster than CCMC-DP.
- With the increasing value of ω, the training time of CCMC-FG rises, but the prediction performance of CCMC-FG becomes better. ω can be set according to the time and accuracy requirements of applications.

The testing time of the proposed methods is similar to OVR. Although the training time of the proposed approaches is slower than OVR, the time required to test is of more importance than the time required to train for many applications.

7.1.5 Comparisons with deep learning methods

ADIOS (Cissé et al., 2016) is a state-of-the-art deep learning architecture for solving multiple class and label tasks. Unlike traditional deep learning methods that use a flat output layer, ADIOS aims to capture the complex dependency between labels/classes to improve deep learning methods. Their approach is to split the label/class set into two subsets, G_1 and G_2, such that given G_1, the labels/classes in G_2 are independent. Our strategy to leverage label/class dependency is very different from that of ADIOS, as shown in Figure 2, we use a classifier chain model for multi-class classification and find the optimal class ordering. After that, we use the predictions of classifiers from easier classes to train the classifiers for harder classes. As such, the assumptions and constraints used in ADOIS are not applicable in our model.

This subsection conducts the experiments on the ILSVRC2012 data set[5]. It contains 1,000 object categories (Liu et al., 2017). Due to the limit of computational resources, we randomly sample 58,700 training instances and 31,300 testing instances from the ILSVRC2012 data set. We use the source code provided by the authors of ADIOS with default parameters. According to Cissé et al. (2016), we use one hidden layer with 1024 rectified linear units (ReLUs) (Glorot et al., 2011) between inputs and G_1, and another 512-dimensional ReLUs between the hidden layer before G_1 and G_2 as well as direct connections between G_1 and G_2. We also compare with VGG (Simonyan and Zisserman, 2014) and residual nets (ResNet) (He et al., 2016). Both VGG and ResNet use a flat output layer, in which do not model the dependency between the classes(Cissé et al., 2016), so the rich structure information among classes is missing in VGG and ResNet. We use the source code provided by the respective authors with default parameters. Following He et al. (2016), we use the 34-layer residual nets due to the limit of computational resources, and also extract

[5] http://www.image-net.org/challenges/LSVRC/2012/

Table 7: Testing error rate (in %) of VGG, ResNet-34, ADIOS and CCMC-FG on the ILSVRC2012 data set.

Method	VGG	ResNet-34	ADIOS	CCMC-FG+VGG features	CCMC-FG+ResNet features
Testing Error	23.95	21.51	21.28	23.98 ($\omega = 10$) 22.67 ($\omega = 30$) 20.73 ($\omega = 50$)	21.85 ($\omega = 10$) 20.11 ($\omega = 30$) 19.84 ($\omega = 50$)

2048-dimensional features by the ResNet-34. According to Simonyan and Zisserman (2014), we extract 4096-dimensional features from the 16-layer of the VGG. Here, ω is set to 10, 30 and 50 for our method.

The classification results are reported in Table 7. From this table, we can observe that

- ADIOS outperforms VGG and ResNet-34, which verifies ADIOS's claim: existing deep learning approaches do not take into account the often unknown but nevertheless rich relationships between classes, this knowledge about the rich class structure (and other deep structure in data) is sometime referred to as dark knowledge (e.g. by Hinton et al. (2015) and Ba and Caruana (2014)).
- Without the restriction of the assumptions and constraints used in ADOIS, our method achieves better performance than ADIOS with the increasing value of ω.
- CCMC-FG with ResNet features obtains better performance than CCMC-FG with VGG features, which demonstrates that our proposed method can be further improved based on better features.
- Based on the deep learning features, CCMC-FG consistently improves VGG and ResNet-34 with the increasing value of ω. The results validate our analysis and the better ordering of classes obtains the better performance. Note that the above mentioned results were obtained using 58,700 training data points. We conclude that with limited data, the usage of structure information is helpful. We conjecture that this advantage may ultimately vanish as more and more data becomes available for training.

7.2 Experiment on Multi-label Classification

We conduct experiments on ten real-world multi-label data sets with various domains from three websites.[6][7][8] The EURLEX_SM and EURLEX_ED data sets are preprocessed according to the experimental settings in Dembczynski et al. (2010) and Zhang and Schneider (2012). The statistics of data sets are presented in Table 8. We compare our algorithms with some baseline methods: BR, CC, ECC, CCA (Zhang and Schneider, 2011) and MMOC (Zhang and Schneider, 2012). ECC is averaged over several CC predictions with random

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

[7] http://meka.sourceforge.net/#datasets

[8] http://cse.seu.edu.cn/people/zhangml/Resources.htm#data

Table 8: Multi-label data sets used in the experiments.

Data	# inst.	# attr.	# labels	Domain
YEAST	2,417	103	14	BIOLOGY
IMAGE	2,000	294	5	IMAGE
SLASHDOT	3,782	1,079	22	TEXT
ENRON	1,702	1,001	53	TEXT
LLOG	799	1,004	10	LINGUISTICS
ART	6,849	23,146	10	ART
EURLEX_SM_10	11,454	5,000	10	TEXT
EURLEX_ED_10	6,540	5,000	10	TEXT
EURLEX_SM	19,348	5,000	201	TEXT
EURLEX_ED	19,348	5,000	3,993	TEXT

order and the ensemble size in ECC is set to 10 according to Dembczynski et al. (2010); Read et al. (2009). In our experiment, the running time of PCC and EPCC (Dembczynski et al., 2010) on most data sets, like SLASHDOT and ART, takes more than one week. From the results in Dembczynski et al. (2010), ECC is comparable with EPCC and outperforms PCC, so we do not consider PCC and EPCC here. CCA and MMOC are two state-of-the-art encoding-decoding (Hsu et al., 2009) methods. We cannot get the results of CCA and MMOC on ART , EURLEX_SM_10 and EURLEX_ED_10 data sets in one week.

We consider the following evaluation measurements Mao et al. (2013) to measure the prediction performance of all methods fairly:

- Example-F1: computes the F-1 score for all the labels of each testing example and then takes the average of the F-1 score.
- Macro-F1: calculates the F-1 score for each label and then takes the average of the F-1 score.
- Micro-F1: computes true positives, true negatives, false positives and false negatives over all labels, and then calculates an overall F-1 score.

The larger the value of those measurements, the better the performance. We perform 5-fold cross-validation on each data set and report the mean and standard error of each evaluation measurement.

7.2.1 Small-scale results

Three measurement results for CC-Greedy, CC-DP and baseline approaches in respect to the different small-scale data sets are reported in Tables 9, 10 and 11. We conduct the pairwise t-test at a 5% significance level to show that our methods perform significantly better than the compared methods. From the results, we can see that:

- BR generally underperforms in terms of Macro-F1 and Micro-F1 and it is much inferior to other methods in terms of Example-F1. Our experiment provides empirical evidence that the label correlations exist in many real word data sets and because BR ignores the information about the correlations between the labels, BR achieves poor performance on most data sets.

Table 9: Results of Example-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. “-” denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.6076±0.02[6]	0.5850±0.03[7]	0.6096±0.02[5]	0.6109±0.02[4]	0.6132±0.02[3]	**0.6144**±0.02[1]	0.6135±0.02[2]
IMAGE	0.5247±0.03[7]	**0.5991**±0.02[1]	0.5947±0.02[4]	0.5947±0.01[4]	0.5960±0.01[3]	0.5939±0.02[6]	0.5976±0.02[2]
SLASHDOT	0.4898±0.02[6]	0.5246±0.03[4]	0.5123±0.03[5]	0.5260±0.02[3]	0.4895±0.02[7]	0.5266±0.02[2]	**0.5268**±0.02[1]
ENRON	0.4792±0.02[7]	0.4799±0.01[6]	0.4848±0.01[4]	0.4812±0.02[5]	**0.4940**±0.02[1]	0.4894±0.02[2]	0.4880±0.02[3]
LLOG	0.3138±0.02[6]	0.3219±0.03[4]	0.3223±0.03[3]	0.2978±0.03[7]	0.3153±0.03[5]	0.3269±0.02[2]	**0.3298**±0.03[1]
ART	0.4840±0.02[5]	0.5013±0.02[4]	0.5070±0.02[3]	-	-	0.5131±0.02[2]	**0.5135**±0.02[1]
EURLEX_SM_10	0.8594±0.00[5]	**0.8609**±0.00[1]	0.8606±0.00[3]	-	-	0.8600±0.00[4]	**0.8609**±0.00[1]
EURLEX_ED_10	0.7170±0.01[5]	0.7176±0.01[4]	0.7183±0.01[2]	-	-	0.7183±0.01[2]	**0.7190**±0.01[1]
Average Rank	5.88	3.88	3.63	4.60	3.80	2.63	1.50

Table 10: Results of Macro-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. “-” denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.3543±0.01[4]	**0.3993**±0.03[1]	0.3763±0.02[2]	0.3496±0.02[5]	0.3431±0.02[7]	0.3441±0.02[6]	0.3596±0.02[3]
IMAGE	0.5852±0.01[7]	**0.6013**±0.02[1]	0.5988±0.01[4]	0.6010±0.01[2]	0.5975±0.01[6]	0.5987±0.02[5]	0.6010±0.01[2]
SLASHDOT	0.3416±0.01[4]	0.3485±0.02[2]	0.3331±0.01[7]	**0.3512**±0.02[1]	0.3334±0.01[6]	0.3431±0.01[3]	0.3408±0.01[5]
ENRON	0.2089±0.02[2]	0.2066±0.02[5]	0.2088±0.02[3]	0.1594±0.03[6]	0.1539±0.02[7]	**0.2090**±0.02[1]	0.2082±0.02[4]
LLOG	0.3452±0.03[2]	0.3428±0.03[4]	0.3425±0.04[5]	0.3189±0.04[7]	0.3303±0.04[6]	0.3448±0.03[3]	**0.3471**±0.04[1]
ART	0.4836±0.01[4]	0.4816±0.01[5]	0.4851±0.02[3]	-	-	0.4876±0.01[2]	**0.4884**±0.02[1]
EURLEX_SM_10	0.8546±0.00[5]	0.8558±0.00[2]	0.8554±0.00[3]	-	-	0.8550±0.00[4]	**0.8559**±0.00[1]
EURLEX_ED_10	0.7201±0.01[5]	0.7202±0.01[4]	0.7205±0.01[3]	-	-	0.7208±0.01[2]	**0.7217**±0.01[1]
Average Rank	4.13	3.00	3.75	4.20	6.40	3.25	2.25

- CC improves on the performance of BR, however, it underperforms compared to ECC. This result verifies the answer to our first question stated in Section 2.2: the label order does affect the performance of CC; ECC, which averages over several CC predictions with random order, improves the performance of CC.

Table 11: Results of Micro-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. “-” denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.6320±0.02[4]	0.6185±0.03[7]	0.6306±0.02[5]	**0.6362**±0.03[1]	0.6361±0.02[2]	0.6303±0.02[6]	0.6328±0.02[3]
IMAGE	0.5840±0.02[7]	0.5994±0.02[2]	0.5955±0.01[5]	**0.6003**±0.01[1]	0.5958±0.01[4]	0.5946±0.02[6]	0.5980±0.01[3]
SLASHDOT	0.5233±0.02[6]	0.5278±0.03[3]	0.5175±0.03[7]	**0.5844**±0.02[1]	0.5720±0.02[2]	0.5266±0.02[5]	0.5272±0.02[4]
ENRON	0.5052±0.01[6]	0.5013±0.01[7]	0.5056±0.01[5]	0.5335±0.02[2]	**0.5401**±0.01[1]	0.5104±0.01[3]	0.5096±0.01[4]
LLOG	**0.3768**±0.03[1]	0.3712±0.03[6]	0.3730±0.04[5]	0.3623±0.03[7]	0.3760±0.03[3]	0.3744±0.03[4]	0.3762±0.03[2]
ART	0.5122±0.02[5]	0.5130±0.02[4]	0.5156±0.02[3]	-	-	**0.5184**±0.01[1]	**0.5184**±0.02[1]
EURLEX_SM_10	0.8718±0.00[5]	0.8727±0.00[2]	0.8725±0.00[3]	-	-	0.8722±0.00[4]	**0.8733**±0.00[1]
EURLEX_ED_10	0.7419±0.01[5]	0.7421±0.01[4]	0.7424±0.01[3]	-	-	0.7425±0.01[2]	**0.7432**±0.01[1]
Average Rank	4.88	4.38	4.50	2.40	2.40	3.88	2.38

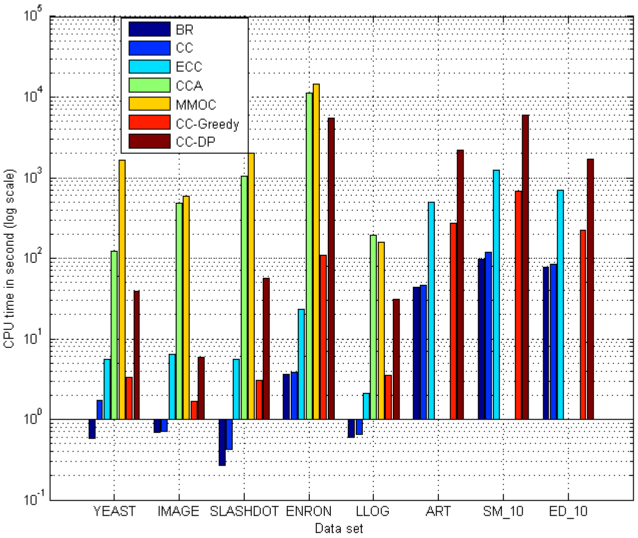


Figure 4: Training time (in second) of all methods on the small-scale data sets. EURLEX_SM_10 and EURLEX_ED_10 are abbreviated to SM_10 and ED_10.

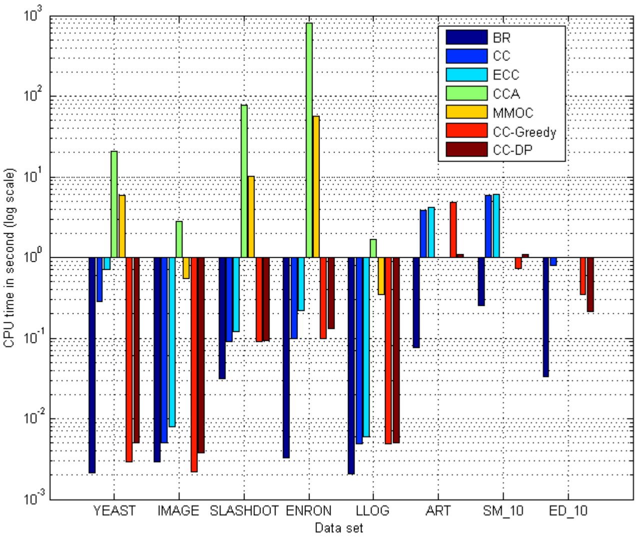


Figure 5: Testing time (in second) of all methods on the small-scale data sets. EURLEX_SM_10 and EURLEX_ED_10 are abbreviated to SM_10 and ED_10.

- Our algorithms outperform CCA and MMOC. This study verifies that optimal CC achieves competitive results compared with state-of-the-art encoding-decoding approaches.

- Our proposed CC-DP and CC-Greedy algorithms are successful on most data sets. This empirical result verifies the effectiveness of our easy-to-hard learning strategies, and we provide an answer to the last two questions stated in Section 2.2: a globally optimal CC exists and CC-DP is able to find the globally optimal CC that achieves the best prediction performance. The CC-Greedy algorithm achieves comparable prediction performance with CC-DP.

Figures 4 and 5 show the training and testing time of CC-Greedy, CC-DP and the baseline methods on the small-scale data sets, respectively. According to these two figures, we can see that:

- Our proposed algorithms are much faster than CCA and MMOC in terms of both training and testing time.
- CC-Greedy and CC-DP achieve comparable testing time with BR, CC and ECC. Though the training time of our algorithms are slower than BR, CC and ECC, our extensive empirical studies show that our algorithms achieve superior prediction performance than those baselines.
- The CC-Greedy algorithm is much faster than CC-DP in terms of training time, and it achieves comparable prediction performance with CC-DP.

7.2.2 Large-scale results

This subsection studies the performance of Tree-Greedy, Tree-DP and other baselines on the EURLEX_SM and EURLEX_ED data sets with many labels. We cannot get the results of CCA and MMOC on EURLEX_SM and EURLEX_ED data sets in one week. And we also cannot get the results of ECC on EURLEX_ED data set in one week. The prediction performance of Tree-Greedy, Tree-DP and the baselines are reported in Tables 12, 13 and 14. We conduct the pairwise t-test at a 5% significance level to show that our methods perform significantly better than the compared methods. From the results, we can see that: our proposed Tree-Greedy and Tree-DP algorithms consistently outperform BR, CC and ECC on the data sets with many labels.

The training and testing time of Tree-Greedy, Tree-DP and the baselines on the EURLEX_SM and EURLEX_ED data sets are shown in Figure 6. According to this figure, we can observe that compared to BR, CC and ECC, our algorithms maintain the testing time over an acceptable threshold, while our methods are much faster than the baselines in terms of training time.

7.3 Experiment on Ordinal Classification

This subsection conducts experiments on four ordinal data sets with various domains from website[9]. The statistics of these data sets are presented in Table 15. We compare our algorithm with some baseline methods: SVM (Shashua and Levin, 2002), MAP (Chu and Ghahramani, 2005), EP (Chu and Ghahramani, 2005) and BD (Destercke and Yang, 2014).

[9] http://www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html

Table 12: Results of Example-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.6970±0.02[5]	0.7233±0.01[4]	0.7263±0.01[3]	0.7292±0.01[2]	**0.7301**±0.01[1]
EURLEX_ED	0.4345±0.03[4]	0.4528±0.02[3]	-	0.4550±0.01[2]	**0.4563**±0.01[1]
Average Rank	4.5	3.5	3	2	1

Table 13: Results of Macro-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.4777±0.02[5]	0.4785±0.02[4]	0.4800±0.02[3]	0.4817±0.01[2]	**0.4834**±0.01[1]
EURLEX_ED	0.1660±0.01[4]	0.1812±0.00[3]	-	0.1844±0.00[2]	**0.1848**±0.00[1]
Average Rank	4.5	3.5	3	2	1

The results are shown in Table 16. From this table, we can see that our proposed CCMC-DP outperforms the other baselines on all data sets, which verifies that our method is able to capture and use the correlated information between ordinal classes, and boost the performance of ordinal classification problems.

7.4 Experiment on Relationship Prediction

This subsection conducts experiments on the Epinions data set (Massa and Avesani, 2006). According to Chiang et al. (2015), we collect 10,000 users and 41 features in this data set. We compare our proposed Tree-DP algorithm with some popular methods: IMC (Jain and Dhillon, 2013), MF-ALS (Hsieh et al., 2012), HOC-3 (Chiang et al., 2014), HOC-5 (Chiang et al., 2014) and DirtyIMC (Chiang et al., 2015). We perform 5-fold cross-validation on this data set and report the mean and standard error of Example-F1. The results are shown in Table 17. From this table, we can see that 1) DirtyIMC outperforms the other baselines, which verifies the effectiveness of using feature information and is consistent with the empirical results in Chiang et al. (2015). 2) Our proposed Tree-DP algorithm is able to achieve the best accuracy among all baselines, which demonstrates the superior performance of the easy-to-hard learning strategy.

8. Conclusion

To precisely classify multiclass data sets with confusing classes, this paper aims to solve classification tasks from easy to hard, and then use the predictions from simpler tasks to help solve the harder tasks. To achieve our goal, we first build the classifier chain model for

Table 14: Results of Micro-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.7321±0.01[5]	0.7422±0.02[3]	0.7363±0.01[4]	0.7440±0.01[2]	**0.7454**±0.01[1]
EURLEX_ED	0.4200±0.01[4]	0.4477±0.01[3]	-	0.4527±0.00[2]	**0.4549**±0.00[1]
Average Rank	4.5	3	4	2	1

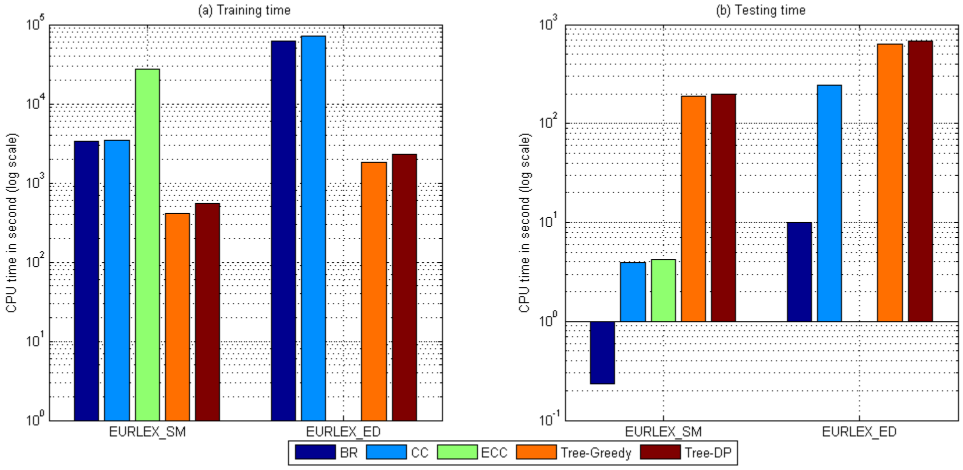


Figure 6: Training and testing time (in second) of BR, CC, ECC, Tree-Greedy and Tree-DP on EURLEX_SM and EURLEX_ED data sets.

multiclass classification (CCMC) to transfer class information between classifiers. Then, we generalize the CCMC model over a random class order and provide a theoretical analysis of the generalization error for the proposed generalized model. Our results show that the upper bound of the generalization error depends on the sum of the reciprocal of the square of the margin over the classes. Based on our results, we propose the easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes.

Similar to CCMC, a CC model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. However, confusing labels decrease the generalization performance of CC, especially when there are many confusing labels, because it ignores the label's order of difficulty. Thus, it is imperative to learn the appropriate label order for CC. Motivated by our analysis of CCMC, we first generalize the CC over a random label order and provide the generalization error bound for the proposed generalized model, and then we also propose the easy-to-hard learning paradigm for multi-label classification

Table 15: Ordinal data sets used in the experiments.

Data set	# TRAINING	# TESTING	# Rank
Stocks Domain	600	350	5
Machine CPU	150	59	5
Abalone	1,000	3,177	6
Boston Housing	300	206	5

Table 16: Testing error rate (in %) on ordinal data sets.

Data set	SVM	MAP	EP	BD	CCMC-DP
Stocks Domain	10.84	11.99	12.00	10.05	9.87
Machine CPU	19.15	18.47	18.56	18.71	17.34
Abalone	22.93	23.22	23.37	23.43	21.62
Boston Housing	26.72	26.04	25.85	25.86	23.50

to automatically identify easy and hard labels. Lastly, we use the predictions from simpler labels to help solve harder labels.

It is very expensive to search over $q!$ different class or label orders for learning the objective of our proposed easy-to-hard learning paradigms, which is computationally infeasible for a large q. We thus propose the CCMC-DP and CC-DP algorithms to find the globally optimal solution, respectively, which requires $\mathcal{O}(q^3nd)$ complexity. To speed up the CCMC-DP and CC-DP algorithm, we propose the CCMC-Greedy and CC-Greedy algorithms to find a locally optimal CCMC and CC, respectively, which takes $\mathcal{O}(q^2nd)$ time. Fast greedy and tree-based algorithms are further developed to handle large data sets with many classes and labels, respectively, which scale linearly with q.

Comprehensive experiments on extensive multiclass data sets, without and with background, demonstrate that our proposed methods consistently improve the prediction performance of OVR and outperforms ECOC and Top-k multiclass SVM. Our human action recognition experiment results also validate our analysis and the success of our proposed easy-to-hard learning strategies: we can automatically identify easy and hard classes, and use the predictions of classifiers from easier classes to train the classifiers for harder classes. Furthermore, this paper also provide an affirmative answer to Rifkin and Klautau's conjecture.

Empirical results on ten real-world multi-label data sets from different domains verify the effectiveness of our easy-to-hard learning strategies, and we provide an answer to the last two questions stated in Section 2.2: a globally optimal CC exists, and CC-DP is able to find the globally optimal CC which achieves the best prediction performance. Moreover, we provide theoretical support for the argument: multi-label learning methods which explicitly capture the label's relationship will usually achieve better prediction performance.

Table 17: Results of Example-F1 on the Epinions data sets (mean ± standard deviation).

Data set	IMC	MF-ALS	HOC-3	HOC-5	DirtyIMC	Tree-DP
Epinions	0.910 ± 0.002	0.936 ± 0.001	0.926 ± 0.001	0.927 ± 0.002	0.939 ± 0.001	0.946 ± 0.001

Lastly, we demonstrate our proposed easy-to-hard learning strategies can be successfully applied to a wide range of applications, such as ordinal classification and relationship prediction. From a more philosophical point of view, our work has shown that the proper usage of structure information in multiclass and multi-label problems yields better modeling, in other words structuring output class information may be an attractive path to incorporate more dark knowledge into learning models.

Acknowledgments

We would like to thank the action editor and reviewers for their valuable comments and constructive suggestions on our paper. This project was partially supported by the ARC Future Fellowship FT130100746, ARC grant LP150100671, DP170101628, DP150102728, DP150103071, NSFC 61232006 and NSFC 61672235. Klaus-Robert Müller acknowledges partial funding from DFG & BMBF and was also supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (No. 2017-0-00451).

Appendix A. Covering Numbers

Definition 19 (Covering Numbers) *Let (X, d) be a (pseudo-) metric space, A be a subset of X and $\epsilon > 0$. A set $B \subseteq X$ is an ϵ-cover for A, if for every $a \in A$, there exists $b \in B$ such that $d(a, b) < \epsilon$. The ϵ-covering number of A, $\mathcal{N}_d(\epsilon, A)$, is the minimal cardinality of an ϵ-cover for A (if there is no such finite cover then it is defined to be ∞).*

In the main paper, let $\mathcal{N}(\epsilon, \mathcal{H}, \mathbf{s})$ be the ϵ-covering number of $\mathcal{H}$ with respect to the l_∞ pseudo-metric measuring the maximum discrepancy on the sample $\mathbf{s}$, that is, with respect to the distance $d(f, g) = \max_{1 \leq t \leq m} |f(x_t) - g(x_t)|$, for $f, g \in \mathcal{H}$.

Appendix B. Proof of Lemma 6

Proof (of Lemma 6). For each $\mathbf{s}$, let $\bar{h}_\mathbf{s}$ be a function for which $|er_D[\bar{h}_\mathbf{s}] - er_\mathbf{s}[\bar{h}_\mathbf{s}]| \geq \epsilon$ if such a function exists, and any fixed function in $\mathcal{H}$ otherwise. Then

$$\begin{aligned} P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{h \in \mathcal{H}} |er_{\bar{\mathbf{s}}}[h] - er_\mathbf{s}[h]| \geq \epsilon/2) \geq & P_{\mathbf{s}\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}] - er_\mathbf{s}[\bar{h}_\mathbf{s}]| \geq \epsilon/2) \\ \geq & P_{\mathbf{s}\bar{\mathbf{s}}}(\{|er_D[\bar{h}_\mathbf{s}] - er_\mathbf{s}[\bar{h}_\mathbf{s}]| \geq \epsilon\} \bigcap \{|er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}] - er_D[\bar{h}_\mathbf{s}]| \leq \epsilon/2\}) \\ = & E_\mathbf{s}[\mathbf{I}\big(|er_D[\bar{h}_\mathbf{s}] - er_\mathbf{s}[\bar{h}_\mathbf{s}]| \geq \epsilon\big) P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}] - er_D[\bar{h}_\mathbf{s}]| \leq \epsilon/2)] \end{aligned} \tag{5}$$

Now the conditional probability inside can be bounded using Chebyshev's inequality:

$$P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}] - er_D[\bar{h}_\mathbf{s}]| \leq \epsilon/2) \geq 1 - \frac{\mathbf{Var}_{\bar{\mathbf{s}}}[er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}]]}{\epsilon^2/4} \tag{6}$$

Since $\bar{\mathbf{s}} \sim D^m$ and $er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}]$ is $1/m$ times a Binomial random variable with parameters $(m, er_D[\bar{h}_\mathbf{s}])$, we have $\mathbf{Var}_{\bar{\mathbf{s}}}[er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}]] = \frac{er_D[\bar{h}_\mathbf{s}](1 - er_D[\bar{h}_\mathbf{s}])}{m} \leq \frac{1}{4m}$. This gives

$$P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_\mathbf{s}] - er_D[\bar{h}_\mathbf{s}]| \leq \epsilon/2) \geq 1 - \frac{1}{m\epsilon^2} \geq \frac{1}{2} \tag{7}$$

whenever $m\epsilon^2 \geq 2$. Thus we get

$$\begin{aligned} P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{h \in \mathcal{H}} |er_{\bar{\mathbf{s}}}[h] - er_\mathbf{s}[h]| \geq \epsilon/2) \geq & \frac{1}{2} P_\mathbf{s}(|er_D[\bar{h}_\mathbf{s}] - er_\mathbf{s}[\bar{h}_\mathbf{s}]| \geq \epsilon) \\ = & \frac{1}{2} P_\mathbf{s}(\sup_{h \in \mathcal{H}} |er_D[h] - er_\mathbf{s}[h]| \geq \epsilon) \end{aligned} \tag{8}$$

where the last step of Eq. (8) is by definition of $\bar{h}_\mathbf{s}$. ■

Appendix C. Proof of Theorem 15

Proof (of Theorem 15). We proof the theorem using the mathematical induction. For $i \in \{1, \cdots, q\}$,

Case 1: $V(i,1) = \frac{1}{(\gamma_i^1)^2}$, where γ_i^1 is the margin for λ_i, without augmented input and $M_i^1 = \{\lambda_i\}$.

Case 2: $V(i,2) = \min_{j \neq i, \lambda_i \notin M_j^1} \{(\frac{1}{(\gamma_i^2)^2} + V(j,1)\}$, where γ_i^2 is the margin for λ_i, with M_j^1 as the augmented input. As in case 1, we already calculated $V(i,1)$, so we can easily find the solution of $V(i,2)$. Assume $V(j,1)$ is the optimal value for computing $V(i,2)$, then we can get $M_i^2 = M_j^1 \cup \{\lambda_i\}$.

Case 3: Assume $V(i,k-1), k \leq q$ is the optimal Q' over a subset of $\mathcal{M}$ with the length of $k-1$, where the class or label order ends by λ_i and M_i^{k-1} denote the corresponding class or label set for $V(i,k-1)$.

Case 4: $V(i,k) = \min_{j \neq i, \lambda_i \notin M_j^{k-1}} \{\frac{1}{(\gamma_i^k)^2} + V(j,k-1)\}$, where γ_i^k is the margin for λ_i, with M_j^{k-1} as the augmented input. Based on the assumption in case 3, we can obtain $V(i,k), i \in \{1, \cdots, q\}$. Thus, we can find the optimal Q' over $\mathcal{M}$ by using CCMC-DP or CC-DP algorithm. ■

Appendix D. Greedy Algorithm

This section presents the details of the CC-Greedy algorithm. Let $\{\mathbf{x}_t\}_{t=1}^n$ be the feature and $\{\mathbf{y}_t(\zeta_j)\}_{t=1}^n$ be the label, the output parameter of SVM is defined as $[\mathbf{w}_j, b] = SVM(\{\mathbf{x}_t, \mathbf{y}_t(\zeta_1), \cdots, \mathbf{y}_t(\zeta_{j-1})\}_{t=1}^n, \{\mathbf{y}_t(\zeta_j)\}_{t=1}^n)$.

Algorithm 1 CC-Greedy

```
Input: training data $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^n$ with size $n$ and label set $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$.
Set $\mathcal{M} = \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$.
for $\lambda_j \in \mathcal{M}$ do
    Calculate $[\mathbf{w}_j, b] = SVM(\{\mathbf{x}_t\}_{t=1}^n, \{\mathbf{y}_t(\lambda_j)\}_{t=1}^n)$.
    Calculate $\gamma_j^1 = \frac{1}{||\mathbf{w}_j||^2}$.
end for
Calculate $\nu = \arg_{\lambda_j \in \mathcal{M}} \min \frac{1}{(\gamma_j^1)^2}$.
Set $\mathcal{M} = \mathcal{M} - \{\lambda_\nu\}$
Set $C[1] = \lambda_\nu$.
for $s = 2$ to $q$ do
    for $\lambda_k \in \mathcal{M}$ do
        Calculate $[\mathbf{w}_k, b] = SVM(\{\mathbf{x}_t, \mathbf{y}_t(C[1]), \cdots, \mathbf{y}_t(C[s-1])\}_{t=1}^n, \{\mathbf{y}_t(\lambda_k)\}_{t=1}^n)$.
        Calculate $\gamma_k^s = \frac{1}{||\mathbf{w}_k||^2}$.
    end for
    Calculate $\nu = \arg_{\lambda_k \in \mathcal{M}} \min \frac{1}{(\gamma_k^s)^2}$ .
    Set $\mathcal{M} = \mathcal{M} - \{\lambda_\nu\}$.
    Set $C[s] = \lambda_\nu$.
end for
Output this locally optimal CC.
```

References

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, pages 2654–2662, 2014.

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods - Support Vector Learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1998.

Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems 23*, pages 163–171, 2010.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

Alina Beygelzimer, John Langford, Yury Lifshits, Gregory B. Sorkin, and Alexander L. Strehl. Conditional probability tree estimation analysis and algorithms. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 51–58, 2009a.

Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error-correcting tournaments. In *Proceedings of the 20th Conference on Algorithmic Learning Theory*, pages 247–262, 2009b.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and C.M.Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *British Machine Vision Conference*, pages 1–12. British Machine Vision Association, 2010.

Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems 25*, pages 1538–1546, 2012.

Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S. Dhillon, and Ambuj Tewari. Prediction and clustering in signed networks: a local to global perspective. *Journal of Machine Learning Research*, 15(1):1177–1213, 2014.

Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S. Dhillon. Matrix completion with noisy side information. In *NIPS*, pages 3447–3455, 2015.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

Moustapha Cissé, Maruan Al-Shedivat, and Samy Bengio. ADIOS: architectures deep in output space. In *ICML*, pages 2770–2779, 2016.

Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 279–286, Haifa, Israel, 2010. Omnipress.

Can Demirkesen and Hocine Cherifi. An evaluation of divide-and-combine strategies for image categorization by multi-class support vector machines. In *23rd International Symposium on Computer and Information Sciences*, pages 1–6, 2008.

Sébastien Destercke and Gen Yang. Cautious ordinal classification by binary decomposition. In *ECML/PKDD*, pages 323–337, 2014.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

Chen Gong, Dacheng Tao, Wei Liu, Liu Liu, and Jie Yang. Label propagation via teaching-to-learn and learning-to-teach. *IEEE Trans. Neural Netw. Learning Syst.*, 28(6):1452–1465, 2017.

Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 902–909. IEEE Computer Society, 2010.

Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In Toby Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1300–1305, Barcelona, Catalonia, Spain, 2011. AAAI Press.

Yuhong Guo and Dale Schuurmans. Adaptive large margin training for multilabel classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, pages 695–702, 2004.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S. Dhillon. Low rank modeling of signed networks. In *KDD*, pages 507–515, 2012.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 13(2):415–425, 2002.

Daniel Hsu, Sham Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems*, pages 772–780, 2009.

Sheng-Jun Huang and Zhi-Hua Zhou. Multi-label learning by exploiting label correlations locally. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada, 2012. AAAI Press.

Lina Huo, Licheng Jiao, Shuang Wang, and Shuyuan Yang. Object-level saliency detection with color attributes. *Pattern Recognition*, 49:162–173, 2016.

Prateek Jain and Inderjit S. Dhillon. Provable inductive matrix completion. *CoRR*, abs/1306.0626, 2013.

Feng Kang, Rong Jin, and Rahul Sukthankar. Correlated label propagation with application to multi-label learning. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, NY, USA, 2006. IEEE Computer Society.

Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 31st Symposium on the Foundations of Computer Science*, pages 382–391, Los Alamitos, CA, 1990. IEEE Computer Society Press.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k multiclass SVM. In *Advances in Neural Information Processing Systems 28*, pages 325–333, 2015.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.

Weiwei Liu and Ivor W. Tsang. Large margin metric learning for multi-label prediction. In *AAAI*, pages 2800–2806, 2015a.

Weiwei Liu and Ivor W. Tsang. On the optimality of classifier chain for multi-label classification. In *NIPS*, pages 712–720, 2015b.

Weiwei Liu and Ivor W. Tsang. Sparse perceptron decision tree for millions of dimensions. In *AAAI*, pages 1881–1887, 2016.

Weiwei Liu and Ivor W. Tsang. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research*, 18:1–36, 2017.

Weiwei Liu, Xiaobo Shen, and Ivor W. Tsang. Sparse embedded k-means clustering. In *NIPS*, 2017.

Qi Mao, Ivor Wai-Hung Tsang, and Shenghua Gao. Objective-guided image annotation. *IEEE Transactions on Image Processing*, 22(4):1585–1597, 2013.

Paolo Massa and Paolo Avesani. Trust-aware bootstrapping of recommender systems. In *Proceedings of ECAI 2006 Workshop on Recommender Systems*, pages 29–33, 2006.

Jonathan Milgram, Mohamed Cheriet, and Robert Sabourin. "one against one" or "one against all": Which one is better for handwriting recognition with SVMs? In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

Gang Niu, Marthinus Christoffel du Plessis, Tomoya Sakai, Yao Ma, and Masashi Sugiyama. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *NIPS*, pages 1199–1207, 2016.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Trung T. Pham, Ian Reid, Yasir Latif, and Stephen Gould. Hierarchical higher-order regression forest fields: An application to 3D indoor scene labelling. In *ICCV*, 2015.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In Wray L. Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, pages 254–269, Berlin, Heidelberg, 2009. Springer-Verlag.

Ryan M. Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *17th International Conference on Pattern Recognition*, pages 32–36. IEEE Computer Society, 2004.

Chun-Wei Seah, Ivor W. Tsang, and Yew-Soon Ong. Transductive ordinal regression. *TNNLS*, 23(7):1074–1086, 2012.

Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *NIPS*, pages 937–944, 2002.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *12th European Conference on Computer Vision*, pages 746–760. Springer, 2012.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

Ali Fallah Tehrani, Weiwei Cheng, and Eyke Hüllermeier. Preference learning using the choquet integral: The case of multipartite ranking. *IEEE Trans. Fuzzy Systems*, 20(6): 1102–1113, 2012.

Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*, pages 1401–1408, 2004.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US, 2010.

Jason Weston and Chris Watkins. Support vector machines for multi-class pattern recognition. In *7th European Symposium on Artificial Neural Networks*, pages 219–224, 1999.

Jian-Bo Yang and Ivor W. Tsang. Hierarchical maximum margin learning for multi-class classification. In *UAI*, pages 753–760, 2011.

Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang, editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 999–1008, Washington, DC, USA, 2010. ACM.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

Yi Zhang and Jeff Schneider. Maximum margin output coding. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, pages 1575–1582, New York, NY, USA, 2012. Omnipress.

Yi Zhang and Jeff G. Schneider. Multi-label output codes using canonical correlation analysis. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 873–882, Fort Lauderdale, USA, 2011. JMLR.org.