Selective Ensemble of Classifier Chains

Nan Li[1,2] and Zhi-Hua Zhou[1]

[1] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
[2] School of Mathematical Sciences, Soochow University, Suzhou 215006, China
{lin,zhouzh}@lamda.nju.edu.cn

Abstract. In multi-label learning, the relationship among labels is well accepted to be important, and various methods have been proposed to exploit label relationships. Amongst them, *ensemble of classifier chains* (ECC) which builds multiple chaining classifiers by random label orders has drawn much attention. However, the ensembles generated by ECC are often unnecessarily large, leading to extra high computational and storage cost. To tackle this issue, in this paper, we propose *selective ensemble of classifier chains* (SECC) which tries to select a subset of classifier chains to composite the ensemble whilst keeping or improving the performance. More precisely, we focus on the performance measure F1-score, and formulate this problem as a convex optimization problem which can be efficiently solved by the stochastic gradient descend method. Experiments show that, compared with ECC, SECC is able to obtain much smaller ensembles while achieving better or at least comparable performance.

Keywords: multi-label, classifier chains, selective ensemble.

1 Introduction

In traditional supervised learning, one instance is associated with *one* concept; but in many real-world applications, an object is naturally associated with *multiple* concepts simultaneously. For examples, a document may belong to multiple topics [14], an image or a music can be annotated with more than one words [1,24]. Obviously, one label per instance is not capable of dealing with such tasks, and *multi-label learning* which associates each instance with multiple labels simultaneously has become an active research topic during the past few years [17,6,8,28,16,2].

A straightforward approach to multi-label learning is the binary relevance method, which decomposes the task into a number of binary classification problems, each for one label [1]. However, such a method is usually not able to achieve good performance, since it neglects the relationships between class labels, which have been widely accepted to be important [4,27]. In the literature, many methods have been proposed to exploit label relationships. Read *et al.* [16] introduced *classifier chain* (CC) to incorporate class relationships, whilst trying to keep the computational efficiency of the binary relevance method. Specifically, like the binary relevance method, each CC is also consist of multiple binary classifiers each for one label, but these binary classifiers are linked in a chain such that each incorporates the class predicted by the previous classifier as additional attributes. Obviously, the quality of CC is dependent on the label

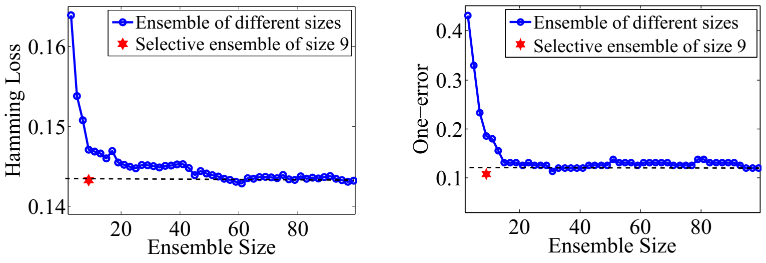

Fig. 1. The performance of ECC with different ensemble sizes on *CAL500*, where hamming loss and one-error are considered. For the selective ensemble, the nine CCs are selected based on the performance on test data from the first twenty CCs via exhaustive search.

orders in the chain. Rather than selecting one good label order, ECC [16] constructs multiple CCs by using different *random* label orders, and it combines them for prediction by a voting scheme. This method has drawn much attention [4,26,11].

Generally speaking, by combining more diverse CCs, the performance of ECC tends to improve and converge. For example, as illustrated in Fig.1, both hamming loss and one-error decrease when the ensemble size increases, and they converge after ensemble size reaches about fifty and thirty, respectively. However, one problem is that the ensembles constructed by ECC tend to be unnecessarily large, which requires large amount of memory storage and also decreases the response time for prediction. As an example, in Fig.1, we can see that similar even better performance can be achieved by using only nine out of the first twenty classifiers, this indicates that the original ECC is quite redundant and it can be largely pruned.

In this paper, we propose the *selective ensemble of classifier chains* (SECC) method, which tries to reduce the ensemble size of ECC whilst keeping or improving the performance. Specifically, by focusing on the frequently-used performance measure F1-score, we try to optimize an upper bound of the empirical risk, and formulate the problem into a convex optimization problem with ℓ_1-norm regularization, also an efficient stochastic optimization method is presented to solve the problem. Experiments on image and music annotation tasks show that SECC is able to obtain much smaller ensembles than ECC, while its performance are better or at least comparable to ECC.

The remainder of the paper is organized as follows. Section 2 briefly reviews some related work. Section 3 presents our proposed SECC method. Section 4 reports the experiment results, which is followed by the conclusion in Section 5.

2 Related Work

In the past few years, many multi-label learning methods have been proposed in the literature, which generally fall into two groups. The first group include algorithm adaptation based methods, which try to modify an algorithm to make multi-label predictions; representatives include BoostTexter [17], ML-kNN [28], HMC tree [25], etc. The other group of methods try to perform *problem transformation*, where a multi-label problem is transformed to one or more other problems, which can be solved by using existing methods. For example, it was transformed into binary classification problems [1,4,16], multi-class classification problems [23,15], and ranking problems [6,7]. In practice, problem transformation based methods are attractive due to its scalability and flexibility, that is, any off-the-shelf methods can be used to suit the requirements.

The *binary relevance* (BR) method [1] is a common problem transformation method, and it transforms a multi-label problem into multiple binary problems, each is to predict the relevance of one label. It is obviously that BR treats each label independently, and does not model relationships existing between class labels. Due to this, its performance is not satisfying, and many methods have been proposed to improve it by exploiting label relationships. Read *et al.* [16] proposed the CC model which is an important improvement over BR. By building chaining classifiers, CC overcomes the disadvantages of BR and achieves higher performance, while retaining the computational complexity. In practice, the performance of CC heavily depends on the label order in the chain; thus ECC is proposed to combine multiple CCs which are based on *random* labels orders. However, a potential problem of ECC is that it may generate unnecessarily large ensembles, reducing its efficiency.

In traditional supervised learning, selective ensemble (*a.k.a.* ensemble pruning or ensemble selection) [30, Chapter 6] is an active research topic, and many technologies has been used to build selective ensembles, such as genetic algorithm [31], semi-definite programming [29], clustering [9], sparse optimization [13]. Obviously, ECC is an ensemble of multi-label classifiers, making the current work essentially different.

3 The SECC Method

In this section, we present our proposed SECC method. Before describing the problem formulation, we begin with some notations and preliminaries. Then, we transform the problem to a convex optimization problem, and give a stochastic optimization solution.

3.1 Preliminaries and Problem Formulation

Let $\mathcal{X}$ be the instance space and $\mathcal{L}$ be a set of l labels. In multi-label learning, each instance $\mathbf{x}_i \in \mathcal{X}$ is associated with multiple labels in $\mathcal{L}$, which is represented as an l-dimensional binary vector $\mathbf{y}_i$ with element 1 indicating $\mathbf{x}_i$ is associated with the k-th label and -1 otherwise. Given a set of training examples $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$, the task of multi-label learning is to learn a multi-label classifier $h : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{Y} \subseteq \{-1, 1\}^l$ is the set of feasible label vectors, such that it can be used predict labels for unseen instances. In practice, it is often to learn a vector-valued function $f : \mathcal{X} \mapsto \mathbb{R}^l$ which determines the label of $\mathbf{x}$ as

$$\hat{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}' \in \mathcal{Y}} \ \mathbf{y}'^{\top} f(\mathbf{x}) \ . \tag{1}$$

It is clear that how the argmax is computed depends on the structure of $\mathcal{Y}$; for example, if $\mathcal{Y} = \{-1, 1\}^l$, $\hat{\mathbf{y}}$ is simply $\mathrm{sign}[f(\mathbf{x})]$.

Instead of learning one multi-label classifier, ECC constructs a group of CCs based on random label orders, and combines them via voting for prediction. Without loss of generality, we denote the group of CCs as $\{h^{(t)} : \mathcal{X} \mapsto \mathcal{Y}\}_{t=1}^{k}$, and ECC combines them to produce the vector-valued function $f : \mathcal{X} \mapsto \mathbb{R}^l$ as

$$f(\mathbf{x}; \mathbf{w}) = \sum\nolimits_{t=1}^{k} w_t h^{(t)}(\mathbf{x}) \ , \tag{2}$$

where $\mathbf{w} = [w_1, \ldots, w_k]^\top$ is the weighting vector; for example, they are simply set to $1/k$ for the simple voting.

As discussed above, one problem with ECC is that the CCs generated based on random label orders are redundant, which results in a large ensemble size, *i.e.*, large value of k. It can be found that in (2) the CC classifier $h^{(k)}$ will be excluded from the ensemble if w_k is zero, and the size of ensemble is exactly $\|\mathbf{w}\|_0$. Based on this, our the task of reducing the size of ECC becomes to finding a weighting vector $\mathbf{w}$, such that the performance of corresponding ensemble is better than or comparable to ECC, while the ensemble size $\|\mathbf{w}\|_0$ is small.

3.2 A Convex Formulation

Here, we find the weighting vector $\mathbf{w}$ based on the principle of empirical risk minimization. That is, given the training example $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$, we consider to solve the vector $\mathbf{w}$ by solving the following optimization problem.

$$\min_{\mathbf{w} \in \mathcal{W}} \ \frac{1}{m} \sum\nolimits_{i=1}^{m} \Delta(\mathbf{y}_i, f(\mathbf{x}_i; \mathbf{w})) \quad \text{s.t. } \|\mathbf{w}\|_0 \leq b \ , \tag{3}$$

where $\mathcal{W}$ is the feasible space of $\mathbf{w}$, $0 < b \leq K$ is the budget of ensemble size, and $\Delta(\mathbf{y}_i, f(\mathbf{x}_i; \mathbf{w}))$ is the empirical risk function measuring the loss of determining $\mathbf{x}_i$'s label vector by $f(\mathbf{x}_i; \mathbf{w})$ while its true label is $\mathbf{y}_i$. More specifically, in current work, we consider the frequently-used performance measure F1-score, and define the loss function Δ as

$$\Delta(\mathbf{y}, f(\mathbf{x}; \mathbf{w})) = 1 - F1(\mathbf{y}, \hat{\mathbf{y}}) \ , \tag{4}$$

where $\hat{\mathbf{y}}$ is the label vector determined by $f(\mathbf{x}; \mathbf{w})$, and $F1(\cdot, \cdot)$ is

$$F1(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2tp}{l + tp - tn} \ ,$$

where $tp = \sum_{i=1} \mathbb{I}(y_i = 1 \wedge \hat{y}_i = 1)$ the number of groundtruth labels in the predicted relevant labels, $tn = \sum_{i=1} \mathbb{I}(y_i = -1 \wedge \hat{y}_i = -1)$ counts how many predicted non-relevant labels are truly non-relevant, and l is the number of possible class labels.

Convex Relaxation. It is easy to see that the problem is not easy to solve, mainly because the risk function Δ is non-decomposable over labels, non-convex and non-smooth. Inspired by some works on directly optimizing performance measures [10,12] which is generally based on structured prediction [19,21], instead of directly minimize the empirical risk as (3), in SECC, we consider to optimize one of its convex upper bounds, which is based on the following proposition.

Proposition 1. *Given a multi-label classifier $f : \mathcal{X} \mapsto \mathbb{R}^L$ in (2) and the empirical risk function Δ in (4), define the loss function $\ell(\mathbf{y}, f(\mathbf{x}))$ as*

$$\ell(\mathbf{y}, f(\mathbf{x})) = \max_{\mathbf{y}' \in \mathcal{Y}} \left[(\mathbf{y}' - \mathbf{y})^\top f(\mathbf{x}) + \Delta(\mathbf{y}, \mathbf{y}') \right] \ , \tag{5}$$

where $\mathcal{Y}$ is the set of feasible label vectors, then the loss function $\ell(\mathbf{y}, f(\mathbf{x}))$ provides a convex upper bound over $\Delta(\mathbf{y}, f(\mathbf{x}))$.

Proof. It is obvious the loss function $\ell(\mathbf{y}, f(\mathbf{x}))$ is convex in f, because it is pointwise maximum of linear functions. Let $\hat{\mathbf{y}} = \mathrm{sign}[f(\mathbf{x})]$, *i.e.*, the prediction determined by $f(\mathbf{x})$, it is easy to find that $\hat{\mathbf{y}}$ is the maximizer of $\mathbf{y}^\top f(\mathbf{x})$. Then, we can get

$$\ell(\mathbf{y}, f(\mathbf{x})) \geq \hat{\mathbf{y}}^\top f(\mathbf{x}) - \mathbf{y}^\top f(\mathbf{x}) + \Delta(\mathbf{y}, \hat{\mathbf{y}}) \geq \Delta(\mathbf{y}, \hat{\mathbf{y}}) \ ,$$

thus $\ell(\mathbf{y}, f(\mathbf{x}))$ is an upper bound over $\Delta(\mathbf{y}, \hat{\mathbf{y}})$, which completes the proof. $\square$

From the optimization problem (3), by replacing the risk function $\Delta(\cdot, \cdot)$ with its upper bound $\ell(\cdot, \cdot)$, and $\|\mathbf{w}\|_0$ with its continuous relaxation $\|\mathbf{w}\|_1$, we can obtain the optimization problem of SECC as

$$\min_{\mathbf{w}} \ \sum\nolimits_{i=1}^{m} \ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w}) + \lambda \|\mathbf{w}\|_1 \ , \tag{6}$$

where λ is the regularization parameter and $\mathrm{H}_i \in \mathbb{R}^{l \times k}$ is the matrix collecting the predictions of $\{h^{(t)}\}_{t=1}^k$ on instance $\mathbf{x}_i$, i.e.,

$$\mathrm{H}_i = [h^{(1)}(\mathbf{x}_i), \cdots, h^{(k)}(\mathbf{x}_i)] \ .$$

Obviously, the problem (6) is an ℓ_1-regularized convex optimization problem, and we solve it via stochastic optimization subsequently.

3.3 Stochastic Optimization

To solve the ℓ_1-regularized convex optimization problem (6), we employ the state-of-the-art stochastic optimization algorithm presented in [18], and the key is how to compute the subgradient of the loss function $\ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$ with respect to $\mathbf{w}$. The following proposition provides the method to compute the subgradient of $\ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$.

Proposition 2. *Given an example $(\mathbf{x}_i, \mathbf{y}_i)$, a set of multi-label classifiers $\{h^{(t)}\}_{t=1}^k$ and a weighing vector $\mathbf{w}_0 \in \mathbb{R}^K$, denote $\mathbf{p}_i = \mathrm{H}_i \mathbf{w}_0$ be the ensemble's prediction on example $(\mathbf{x}_i, \mathbf{y}_i)$, then the vector*

$$\mathbf{g} = (\tilde{\mathbf{y}} - \mathbf{y}_i)^\top \mathrm{H}_i \ ,$$

is a subgradient of $\ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$ at $\mathbf{w}_0$, where $\tilde{\mathbf{y}}$ is the solution of the argmax problem

$$\tilde{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}' \in \mathcal{Y}} \left[\mathbf{y}'^\top \mathbf{p}_i + \Delta(\mathbf{y}_i, \mathbf{y}') \right] \ . \tag{7}$$

Proof. It is obvious that the function $\ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$ is a pointwise maximum of linear functions in $\mathbf{w}$, then it is straightforward to obtain its subgradient if the maximizer of (5) at $\mathbf{w}_0$ can be obtained. Actually, the argmax problem (7) solves the maximizer, which completes the proof. $\square$

Algorithm 1. Stochastic optimization algorithm for SECC

```
Input: $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, CC classifiers $\{h^{(t)}\}_{t=1}^k$, regularization
    parameter $\lambda$, step size $\eta$
Output: weighting vector $\mathbf{w}$
 1: let $\mathbf{w} = 0$, $\boldsymbol{\varrho} = 0$ and $p = 2 \ln k$
 2: repeat
 3:     select $(\mathbf{x}_i, \mathbf{y}_i)$ uniformly at random from $S$
 4:     let $\mathrm{H}_i = [h^{(1)}(\mathbf{x}_i), \cdots, h^{(k)}(\mathbf{x}_i)]$ and $\mathbf{p}_i = \mathrm{H}_i \mathbf{w}$
 5:     solve the argmax, i.e., $\tilde{\mathbf{y}} \leftarrow \mathrm{argmax}_{\mathbf{y}' \in \mathcal{Y}} \left[ \mathbf{y}'^\top \mathbf{p}_i + \Delta(\mathbf{y}_i, \mathbf{y}') \right]$
 6:     compute the sub-gradient, i.e., $\mathbf{g} = (\tilde{\mathbf{y}} - \mathbf{y}_i)^\top \mathrm{H}_i$
 7:     let $\tilde{\boldsymbol{\varrho}} = \boldsymbol{\varrho} - \eta \mathbf{g}$
 8:     $\forall t$, let $\varrho_t = \mathrm{sign}(\tilde{\varrho}_t) \max(0, |\tilde{\varrho}_t| - \eta\lambda)$
 9:     $\forall t$, let $w_t = \mathrm{sign}(\varrho_t)|\varrho_t|^{p-1} / \|\boldsymbol{\varrho}\|_p^{p-2}$
10: until convergence
```

Algorithm. Based on above proposition, we can present the stochastic optimization method for solving the optimization problem (6), whose pseudocode is summarized in Algorithm 1. Specifically, this algorithm is a stochastic subgradient descend method. At each iteration, it first samples an example $(\mathbf{x}_i, \mathbf{y}_i)$ uniformly at random from data S, and then compute the subgradient of $\ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$ (lines 4-6). Since the example $(\mathbf{x}_i, \mathbf{y}_i)$ is chosen at random, the vector g is an unbiased estimate of the gradient of the empirical risk $\sum_{i=1}^m \ell(\mathbf{y}_i, \mathrm{H}_i \mathbf{w})$. Next, the dual vector $\boldsymbol{\varrho}$ is updated with step size η (line 7) so that the empirical risk is decreased; and also it is truncated to decrease the regularizer $\lambda \|\mathbf{w}\|_1$ (line 8). Finally, the updates of $\boldsymbol{\varrho}$ are translated to the variable $\mathbf{w}$ via a link function in line 9. This procedure iterates until convergence. It is worth noting that, at each iteration of Algorithm 1, we even do not need to compute the predictions on all examples, and all the operations are very efficient as long as the argmax problem (7) can be efficiently solved.

Solving the Argmax. To make Algorithm 1 practical, it is obvious that the argmax problem (7) needs to be solved efficiently. Noting that the feasible space of $\tilde{\mathbf{y}}$ is of exponential size (*i.e.*, 2^l), and it is infeasible to solve it by exhaustive search. Fortunately, this problem has been studied in [10], and an efficient solution has been proposed. Roughly speaking, the solution is based on fact that F1-score can be computed from the contingency table (*i.e.*, four numbers: true positive, true negative, false positive, false negative) which take at most $O(l^2)$ different values. Algorithm 2 summarizes the pseudocode, and it is easy to see that its computational complexity is $O(l^2)$, where l is the number of class labels.

Convergence & Complexity. Based on Theorem 3 in [18], we can find that the number of iterations of Algorithm 1 to achieve ϵ-accuracy is bounded by $O(\log k/\epsilon^2)$, where k is the number of CC classifiers. Moreover, in each iteration, all the operations are performed on one single example, and we can see that the computational complexity is dominated by the argmax which as shown above can be solved by Algorithm 2 in $O(l^2)$ time. As a consequence, we can see that the method can converge to a ϵ-accurate

Algorithm 2. Solve the argmax problem (7) [10]

```
Input: true label vector $\mathbf{y}$, current prediction $\mathbf{p}$
Output: label vector $\tilde{\mathbf{y}}$
 1: $\{i_1^+, \ldots, i_{pos}^+\} \leftarrow$ sort $\{i \mid y_i = +1\}$ in descending order of $p_i$'s
 2: $\{i_1^-, \ldots, i_{neg}^-\} \leftarrow$ sort $\{i \mid y_i = -1\}$ in descending order of $p_i$'s
 3: for $tp = 0$ to $pos$ do
 4:     $fn = pos - tp$
 5:     set $y'_{i_1^+}, \ldots, y'_{i_{tp}^+}$ to $+1$ and set $y'_{i_{tp+1}^+}, \ldots, y'_{i_{pos}^+}$ to $-1$
 6:     for $tn = 0$ to $neg$ do
 7:         $fn = neg - tn$
 8:         set $y'_{i_1^-}, \ldots, y'_{i_{fn}^-}$ to $+1$ and set $y'_{i_{fn+1}^+}, \ldots, y'_{i_{neg}^+}$ to $-1$
 9:         let $v \leftarrow \Delta(\mathbf{y}, \mathbf{y}') + \mathbf{y}'^{\top}\mathbf{p}$
10:         if $v$ is the largest so far then
11:             $\tilde{\mathbf{y}} = \mathbf{y}'$
12:         end if
13:     end for
14: end for
```

solution in at most $O(l^2 \cdot \log k/\epsilon^2)$ time. It worth noting that this computational complexity is independent of m, *i.e.* the number of training examples, this constitutes one of the appealing properties of SECC.

4 Experiments

In this section, we perform a set of experiments to evaluate the effectiveness of SECC. Specifically, we first show the effectiveness of optimizing the upper bounds, and then compare SECC with ECC and other state-of-the-art multi-label classifiers.

4.1 Configuration

The experiments are performed on image and music annotation tasks. Specifically,

– **Image annotation** tasks are *Corel5k* [5] which has 5000 images and 374 possible labels, and *Scene* [1] has 2407 images and 6 possible labels;
– **Music annotation** tasks are *CAL500* [24] which has 502 songs and 174 possible labels, and *Emotions* [20] which has 593 songs and 6 possible labels.

The information of these tasks are also summarized in Table 1.

In the experiments, we first train an ECC of size 100, then build SECC upon it. We compare SECC with BSVM [1] which trains one SVM for each label, and state-of-the-art methods including the lazy method ML-kNN [28], label ranking method CLR [7] and the full ECC combining all classifiers (denoted as ECC$_{\text{full}}$). It is also compared with the *random strategy* which selects classifiers randomly (denoted as ECC$_{\text{rand}}$) . Specifically, LIBSVM [3] is used to implement the base classifier in BSVM and ECC; the default implementation of ML-kNN and CLR in MULAN [22] are used.

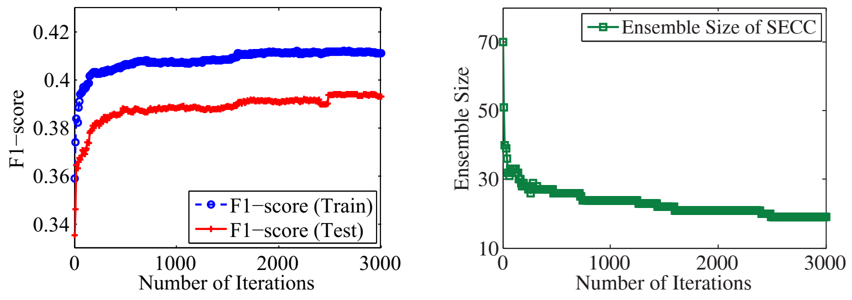

Fig. 2. The F1-score (left) and ensemble size (right) of SECC on the CAL500 task, during the optimization procedure. Both the F1-score on training and test set are ploted.

For each task, these comparative methods are evaluated by 30 times random holdout test, *i.e.*, in each time, 2/3 for training and 1/3 for testing; finally, the mean F1-score over 30 times and the standard derivation are reported; also the sizes of ensembles generated by SECC are reported. In each time of holdout test, SECCs choose the regularization parameter λ by 5 fold cross validation on training set; ECC$_{\text{rand}}$ randomly selects N CC classifiers to form the ensemble, where N is the size of SECC in this time.

4.2 Effectiveness of Optimizing the Upper Bound

A question naturally raised about SECC is whether it is effective to optimize the upper bound of the empirical risk; in other words, whether optimizing the upper bound will improve the performance?

To answer this question, we record the training and test performance on the CAL500 task where the parameter λ is to 2^{-8}, and the results are shown in the left plot of Fig.2. It can be seen that both the F1-score on training and test data improve as the number of iterations increases. This optimizing the upper bound is effective in improving the performance. Also, we record the ensemble sizes of SECC at different iterations, and plot then in the right plot of Fig.2. It can be seen that the ensemble size of SECC decreases with the number of iterations increases. This results shows the effectiveness of SECC in obtaining a smaller ensemble.

Moreover, we can see from Fig.2 that the performance tends to converge after some iterations, *i.e.*, the F1-scores converge after about 1500 iterations. Noting there are 502 examples in CAL500, this means we need to scan the data set for only three times, also the operations in each iteration are very simple, so this method is efficient. For example, on the CAL500 task, it takes only 1.5 seconds for 3000 iterations on a PC with Intel Core 1.7GHz CPU, which is very efficient.

4.3 Performance Comparison Results

The F1-scores achieved by comparative methods are shown in Table 1, where the ensemble size of ensemble methods, including SECC, ECC$_{\text{full}}$ and ECC$_{\text{rnd}}$, are also

Table 1. Experimental results (mean±std.) achieved by comparative methods, which incudes the F1-score and the ensemble sizes for ensemble methods (SECC, ECC$_{\mathrm{full}}$ and ECC$_{\mathrm{rnd}}$). Also, the information of each task is summarized, where *#F*, *#L* and *#N* indicate the number of features, labels and examples, respectively. The mark '•'('∘') indicates that SECC is significantly better (worse) than the corresponding method based on paired t-tests at 95% significance level.

	SECC	**ECC$_{\mathrm{full}}$**	**ECC$_{\mathrm{rand}}$**	**BSVM**	**ML-kNN**	**CLR**
Corel5k: #F=499, #L=374, #N=5000						
F1-score	.149±.006	.134±.006•	.127±.008•	.135±.005•	.016±.003•	.034±.001•
Size	15.3±5.6	100.0	15.3±5.6	–	–	–
Scene: #F=294, #L=6, #N=2407						
F1-score	.672±.002	.668±.012	.657±.014•	.595±.014•	.675±.018	.629±.009•
Size	22.0±10.8	100.0	22.0±10.8	–	–	–
CAL500: #F=68, #L=174, #N=502						
F1-score	.384±.013	.323±.010•	.333±.023•	.314±.029•	.322±.011•	.405±.003∘
Size	19.6±4.7	100.0	19.6±4.7	–	–	–
Emotions: #F=72, #L=6, #N=593						
F1-score	.619±.017	.618±.015	.612±.016	.614±.014	.602±.029•	.622±.016
Size	63.7±21.2	100.0	63.7±21.2	–	–	–

given. For better comparison, we perform paired t-tests at 95% significance level to compare SECC with other methods, and the results are indicated in Table 1 by '•'('∘').

It can be seen from these results that the performance of SECC is quite promising. Compared with the full ensemble ECC$_{\mathrm{full}}$, it achieves 3 significant better F1-scores, while the ensemble size is much smaller. For examples, on *Corel5k*, the F1-score is improved from 0.134 to 0.149, while the ensemble size is reduced to 15.3; on *CAL500* the F1 score is improved from 0.323 to 0.384, whiles the ensemble size of reduced from 100 to less than 20; even on *Scene* and *Emotions*, SECC still achieves comparable performance to ECC, but it still reduces the average ensemble size to 22.0 and 63.7, respectively. Compared with ECC$_{\mathrm{rand}}$ which choose CCs randomly, SECC achieves significantly better performance on *Corel5k*, *CAL500* and *Scene*, and comparable F1-score on *Emotions*. All of these results show the effectiveness of SECC, in both reducing the ensemble sizes and improving the performance.

Comparing with other methods, we can see that the performance of SECC is significantly better than BSVM and ML-kNN on three tasks, also its performance is comparable against the state-of-the-art methods CLR (*i.e.*, 2 wins and 1 loss). It is not hard to find that the superiority of SECC mainly inherits from the good performance of ECC.

In summary, SECC inherits from the good performance of ECC while improve its efficiency by generating smaller ensembles.

5 Conclusion and Future Work

Noticing that ECC tends to generate unnecessarily large ensembles, in this paper, we propose *selective ensemble of classifier chains* (SECC) and it is expected that it can reduce the size of ECC whilst keeping or improving its performance. Specifically, after obtaining multiple CCs, instead of combining all of them by voting like ECC, SECC tries to select some of them to composite the ensemble. In this paper, by focusing on the frequently-used performance measure F1-score, we formulate this selection problem as an ℓ_1-norm regularized convex optimization problem, and present a stochastic subgradient descend method to solve it. Experiments on image and music annotation tasks shows the effectiveness of the proposed method.

In this work, we consider the CC classifier as a common multi-label classifier which has multiple outputs. In practice, CC itself is a group of single-label classifiers, hence it will be interesting to consider SECC at the level of such binary classifiers.

Acknowledgements. We want to thank anonymous reviewers for helpful comments. This research was supported by the National Science Foundation of China (61021062, 61105043) and the Jiangsu Science Foundation (BK2011566).

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
2. Cesa-Bianchi, N., Re, M., Valentini, G.: Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. Machine Learning 88(1), 209–241 (2012)
3. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011)
4. Dembczynski, K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, pp. 279–286 (2010)
5. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
6. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems 14, pp. 681–687. MIT Press, Cambridge (2002)
7. Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning 73(2), 133–153 (2008)
8. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, pp. 195–200 (2005)
9. Giacinto, G., Roli, F., Fumera, G.: Design of effective multiple classifier systems by clustering of classifiers. In: Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, pp. 160–163 (2000)
10. Joachims, T.: A support vector method for multivariate performance measures. In: Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, pp. 377–384 (2005)
11. Kumar, A., Vembu, S., Menon, A.K., Elkan, C.: Learning and inference in probabilistic classifier chains with beam search. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part I. LNCS, vol. 7523, pp. 665–680. Springer, Heidelberg (2012)

12. Li, N., Tsang, I.W., Zhou, Z.-H.: Efficient optimization of performance measures by classifier adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013) (preprint)
13. Li, N., Zhou, Z.-H.: Selective ensemble under regularization framework. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 293–303. Springer, Heidelberg (2009)
14. McCallum, A.: Multi-label text classification with a mixture model trained by EM. Working Notes of AAAI 1999 Workshop on Text Learning (1999)
15. Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets. In: Proceedings of the 8th IEEE International Conference on Data Mining, Pisa, Italy, pp. 995–1000 (2008)
16. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011)
17. Schapire, R., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning 39(2-3), 135–168 (2000)
18. Shalev-Shwartz, S., Tewari, A.: Stochastic methods for l1-regularized loss minimization. Journal of Machine Learning Research 12, 1865–1892 (2011)
19. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Advances in Neural Information Processing Systems 16, pp. 25–32. MIT Press, Cambridge (2003)
20. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proceedings of 2008 International Conference on Music Information Retrieval, Philadelphia, PA, pp. 325–330 (2008)
21. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
22. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: MULAN: A Java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414 (2011)
23. Tsoumakas, G., Vlahavas, I.: Random *k*-labelsets: An ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
24. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. IEEE Transactions on Audio, Speech and Language Processing 16(2), 467–476 (2008)
25. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
26. Zaragoza, J.H., Sucar, L.E., Morales, E.F., Bielza, C., Larrañaga, P.: Bayesian chain classifiers for multidimensional classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Spain, pp. 2192–2197 (2011)
27. Zhang, M.-L., Zhang, K.: Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 999–1007 (2010)
28. Zhang, M.-L., Zhou, Z.-H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)
29. Zhang, Y., Burer, S., Street, W.: Ensemble pruning via semi-definite programming. Journal of Machine Learning Research 7, 1315–1338 (2006)
30. Zhou, Z.-H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton, FL (2012)
31. Zhou, Z.-H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence 137(1-2), 239–263 (2002)