Multi-label Ensemble Learning

Chuan Shi[1], Xiangnan Kong[2], Philip S. Yu[2], and Bai Wang[1]

[1] School of Computer
Beijing University of Posts and Telecommunications, Beijing, China
[2] Department of Computer Science
University of Illinois at Chicago, IL, USA
{shichuan,wangbai}@bupt.edu.cn, {xkong4,psyu}@uic.edu

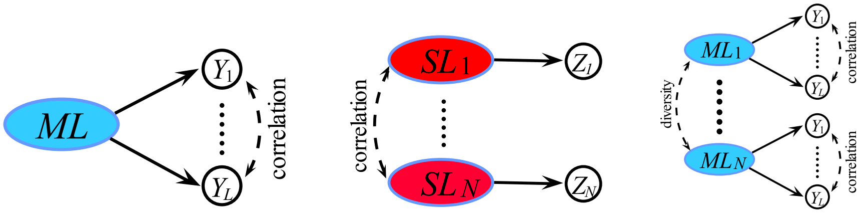
Abstract. Multi-label learning aims at predicting potentially multiple labels for a given instance. Conventional multi-label learning approaches focus on exploiting the label correlations to improve the accuracy of the learner by building an individual multi-label learner or a combined learner based upon a group of single-label learners. However, the generalization ability of such individual learner can be weak. It is well known that ensemble learning can effectively improve the generalization ability of learning systems by constructing multiple base learners and the performance of an ensemble is related to the both accuracy and diversity of base learners. In this paper, we study the problem of multi-label ensemble learning. Specifically, we aim at improving the generalization ability of multi-label learning systems by constructing a *group* of *multi-label base learners* which are both *accurate* and *diverse*. We propose a novel solution, called EnML, to effectively augment the accuracy as well as the diversity of multi-label base learners. In detail, we design two objective functions to evaluate the accuracy and diversity of multi-label base learners, respectively, and EnML simultaneously optimizes these two objectives with an evolutionary multi-objective optimization method. Experiments on real-world multi-label learning tasks validate the effectiveness of our approach against other well-established methods.

Keywords: Multi-label learning, ensemble learning, multi-objective optimization, negative correlation learning.

1 Introduction

Traditional supervised learning works on the single-label scenario, i.e. each instance is associated with one single label within a finite set of labels. However, in many real-world applications, one instance can be associated with multiple labels simultaneously. For example, in text categorization, each document may belong to several topics [20]; in bioinformatics, each gene may be associated with a number of functional classes [6]. This kind of problem is called multi-label learning, which corresponds to the classification problem of classifying each instance with a *set* of labels within the space of possible label sets. Multi-label learning has been drawing increasing attentions from the machine learning and data mining communities in the past decade [5,13,25].

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 223–239, 2011.



(a) Individual multi-label learner

(b) Multiple single-label learners

(c) Multi-label ensemble learning

Fig. 1. Comparison of three strategies of constructing multi-label learning system. *SL* and *ML* represent the single-label and multi-label learner, respectively. *Y* and *Z* represent the single and atomic label, respectively.

The multi-label learning faces a major challenge that the number of possible label combinations grows exponentially. Conventional multi-label learning approaches focus on exploiting the label correlations to improve the accuracy of individual multi-label learner [5,13,15,17,25]. These approaches can be roughly characterized into the following two categories based on the strategy of constructing the learning system: (1) Multi-label learning approaches based upon individual multi-label learner (shown in Figure 1(a)). In this type of approaches, a multi-label learner is constructed to make predictions on all labels. The label correlations are exploited in the structure or learning process of the multi-label learner, such as the neural network structure in ML-RBF [21] and the Bayesian learning in LEAD [25]. (2) Multi-label learning approaches based upon a group of single-label learners (shown in Figure 1(b)), such as EPS [14] and RAKEL [17]. Ensemble learning is used to construct such a group of single-label base learners. Each base learner in the ensemble is constructed to make a prediction on a single label or atomic label (i.e. treating each label subset as a class label). Then those base learners are combined as one multi-label learner to make predictions on all labels. The label correlations are usually exploited among these single-label base learners.

Generally, conventional multi-label learning approaches focus on building one individual multi-label learner. However, the generalization ability of one individual learner can be weak. It is well-known that ensemble learning can improve the generalization ability of a learning system and reduce the overfitting risk by constructing multiple base learners in the single-label setting. In the case of multi-label learning, if we ensemble a group of multi-label base learners to make predictions on all labels, the generalization ability of the multi-label learning system can be significantly improved. This is called the multi-label ensemble learning problem (shown in Figure 1(c)). Since the generalization error of an ensemble is related to the average generalization error of the base learners as well as diversity among the base learners [10], the aim of multi-label ensemble learning is to build a *group* of *multi-label base learners* which are not only *accurate* but also *diverse*. Note that, different from previous ensemble methods for multi-label learning which combine a group of single-label base learners into one multi-label learner, the base learners in the multi-label ensemble learning are the multi-label learners, instead of the single-label learners.

Despite its value and significance, the multi-label ensemble learning has rarely been studied in this context so far. It is challenging to generate a set of accurate and diverse multi-label base learners in the multi-label scenario. The major research challenges are as follows: (1) Conventional ensemble learning approaches usually focus on single-label learning problems. When it is applied to multi-label learning problems, one of the difficulties is the accuracy evaluation of multi-label base learners, which needs to consider the correlations among different labels. (2) In multi-label scenario, it is also difficult to evaluate the diversity of multi-label base learners, since the output of the base learners is a label set (vector), instead of a single label (scale number). (3) It is far more complex when considering the accuracy and diversity of multi-label base learners simultaneously. We need to consider how to balance the trade-off between these two aspects.

In this paper, we first study the problem of multi-label ensemble learning and propose a novel solution, named EnML. Different from conventional multi-label learning approaches, EnML builds a group of accurate and diverse multi-label base learners. First, we propose two novel evaluation objectives to effectively depict the accuracy and diversity of multi-label base learners, respectively. Inspired by the Hilbert-Schmidt Independence Criterion (HSIC) [8], *ML-HSIC* is proposed to evaluate the accuracy of base learners while considering the label correlations in full order. Enlightened by the Negative Correlation Learning (NCL) [11,12], *ML-NCL* is proposed to characterize the diversity of base learners. In order to balance the trade-off between these two objectives for the generalization ability of the ensemble, we then propose a novel evolutionary multi-objective algorithm to search the optimal trade-off among the different objectives. Extensive experiments on the different types of multi-label datasets show that EnML significantly outperforms other popular multi-label learning approaches.

2 Related Work

In order to improve the generalization ability of multi-label learner system, conventional approaches focus on building an accurate multi-label learner by exploiting the label correlations. According to strategies of building learner, conventional multi-label learning approaches can be roughly classified into following two categories. (1) The first type of approaches build an individual multi-label learner to make predictions on all labels. The multi-label learner uses different methods to exploit the label correlations, such as learner structure, optimized criterion, and learning algorithm. For example, the neural network structures in ML-RBF [21] and BP-MLL [23] mix the label relations, the *ranking loss* criterion [6,25] considers the second order correlation of labels, and the Bayesian learning in LEAD [25] learns the label dependency. EnML is different from this type of approaches in building a *group* of multi-label learners. (2) The second type of approaches build a set of single-label base learners. In these approaches, each single-label base learner is built to make a prediction on a single label or atomic label, and then these base learners are combined as a multi-label learner. The label correlations can be exploited among these base learners. Ensemble learning is usually used to build such a set of single-label base learners [14,15,16,17]. For example, RAKEL [17] trains each single-label base learner for the prediction of each element in the powerset of label set, and the single-label base learner in EPS [14] is built for a pruning label subset. Different from these ensemble methods for multi-label learning, the base learners in EnML are the *multi-label learners*.

Ensemble of multiple learners has attracted a lot of research interest in the machine learning community since it is considered as a good approach to improve the generalization ability [2]. Most ensemble learning algorithms train the individual learner independently or sequentially, so the advantages of interaction and cooperation among the individual learners are not effectively exploited. However, Liu and Yao [11,12] have shown that the cooperation with ensemble members is useful for obtaining better ensembles. Negative Correlation Learning (NCL) [3,11,12] introduces a correlation penalty term into the error function of each individual base learner in the ensemble so that the learners are as different as possible on the training error. NCL emphasizes the interaction and cooperation among individual base learners in the ensemble and has performed well on a number of empirical applications. However, the conventional NCL focuses on single-label learning, and has never been applied in multi-label learning so far.

3 The EnML Method

Let $\chi = \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{L} = \{1, 2, \cdots, L\}$ be the finite set of L possible labels. Given a multi-label training set $\mathcal{D} = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq m\}$, where $\mathbf{x}_i \in \chi$ is the i-th instance and $Y_i \subseteq \mathcal{L}$ is the label set associated with $\mathbf{x}_i$. The task of multi-label learning is to learn a multi-label learner $h : \chi \rightarrow 2^{\mathcal{L}}$ from $\mathcal{D}$ which predicts a set of labels for each unseen instance.

As we all know, the ensemble can improve the generalization ability of learning systems by constructing multiple base learners, and the ensemble members should be accurate and diverse [10]. In multi-label ensemble learning, we aim at building such an ensemble, in which each multi-label base learner has good classification performances while these base learners are as diverse as possible. To do so, we propose a multi-objective optimization based solution. Concretely, we proposes two novel criteria, *ML-HSIC* and *ML-NCL*, to evaluate the accuracy and diversity of multi-label base learners, respectively. An evolutionary multi-objective optimization algorithm is then designed to train a set of multi-label base learners which are diverse and all optimal in these two proposed criteria.

3.1 Measure Criteria

ML-HSIC. Many criteria have been proposed to evaluate performances of multi-label learning, such as *hamming loss* [21] and *ranking loss* [23]. These criteria can be used as the accuracy evaluation. However, they fail to directly address the correlations between different labels. An ideal criterion needs to be able to evaluate the accuracy of learners while considering the label correlations.

The accuracy of a multi-label learner h can be considered as the similarity of the true label set $TL = \{Y_1, \cdots, Y_m\}$ and the predicted label set by h on the training data $\mathcal{D}$, $PL = \{h(\mathbf{x}_1), \cdots, h(\mathbf{x}_m)\}$. Furthermore, the similarity can be evaluated with the dependence between PL and TL. That is, the higher dependence between PL and TL, the more similar they are. Many methods can be used to characterize the dependence. In this paper, we derive an evaluation criterion for multi-label learning based upon a dependence evaluation method named Hilbert-Schmidt Independence Criterion (HSIC) [8]. By deriving from the definition of HSIC, we define the accuracy of a learner h as follows:

$$\mathit{ML\text{-}HSIC}(h) = tr(PHQH) \tag{1}$$

where $tr(\cdot)$ is the trace of a matrix and $H = [h_{ij}]_{m\times m}$, $h_{ij} = \delta_{ij} - 1/m$, and δ_{ij} is the indicator function which takes 1 when $i = j$ and 0 otherwise. $P = [p_{ij}]_{m\times m}$ denotes the label kernel matrix based on the true label set TL with the kernel function $p(Y_i, Y_j) = \langle \phi(Y_i), \phi(Y_j)\rangle$. $Q = [q_{ij}]_{m\times m}$ denotes the label kernel matrix based on the predicted label set PL with the kernel function $q(h(\mathbf{x}_i), h(\mathbf{x}_j)) = \langle \psi(h(\mathbf{x}_i)), \psi(h(\mathbf{x}_j))\rangle$. The *ML-HSIC* has the following two advantages: (1) It can effectively evaluate the dependence of TL and PL; (2) The appropriate kernel function can be used to exploit the label correlations. Here, many kernel functions can be applied in P and Q. For example, by using the polynomial kernel of the second degree in the label kernel Q, the second order label correlations can be considered. In this paper, we use RBF kernel in P and Q, since the RBF kernel can potentially exploit the correlations among labels in full order. Therefore, different from conventional accuracy criteria, *ML-HSIC* effectively evaluates the accuracy of multi-label learners with fully considering the correlations among labels.

ML-NCL. Evaluating the diversity of multi-label learners is much more challenging than single-label learning, because, in multi-label learning, the output are a set of labels, instead of a single label. Inspired by the success of Negative Correlation Learning (NCL) in single-label ensemble learning [3,11,12], we propose a criterion to evaluate the diversity of multi-label learners, called *ML-NCL*.

Similar to NCL, the basic idea of *ML-NCL* is to evaluate the negative correlation of each base learner's error with the error for the rest of ensemble. Formally, *ML-NCL* is defined as follows:

$$\begin{aligned}\mathit{ML\text{-}NCL}(h_j) &= -\sum_{i=1}^{m}\{(h_j(\mathbf{x}_i) - h_{ens}(\mathbf{x}_i))^T \sum_{k\neq j}(h_k(\mathbf{x}_i) - h_{ens}(\mathbf{x}_i))\} \\ &= \sum_{i=1}^{m} \|h_j(\mathbf{x}_i) - h_{ens}(\mathbf{x}_i)\|^2\end{aligned} \tag{2}$$

where $h_j(\mathbf{x}_i) \in 2^{\mathcal{L}}$ means the output of leaner h_j on data $\mathbf{x}_i$. h_{ens} is the output of the ensemble of N base learners, which is defined as follows:

$$h_{ens}(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N} h_j(\mathbf{x}_i) \tag{3}$$

The definition shows that *ML-NCL*(h_j) characterizes the significance of difference between the multi-label base learner h_j and the ensemble h_{ens} on training error. Maximizing *ML-NCL* encourages the multi-label base learners to perform differently on training error, so it increases the diversity of base learners. The benefits of *ML-NCL* come from two aspects: (1) It evaluates the diversity of vectors, instead of scale numbers, so *ML-NCL* can be considered as a multi-label version of NCL. (2) It exploits the correlations among multi-label base learners, which has never been done before.

3.2 Multi-objective Optimization Solution

After two criteria are proposed to evaluate the accuracy and diversity of multi-label learners, the next problem is how to optimize them. Different from the single-objective optimization in conventional machine learning, this is a multi-objective optimization problem, i.e. simultaneously maximizing *ML-HSIC* and *ML-NCL*. It can be solved through converting the multi-objective optimization into a single objective optimization by weight sum method. However, this method greatly suffers from the weights setting, because the generalization ability of ensemble largely depends on the trade-off between these two objectives. In this paper, we makes use of an Evolutionary Multi-objective Optimization technology (EMO) [4] to balance the trade-off, since EMO can automatically find optimal trade-off through population evolutionary. Without loss of generality, we focus on solving the multi-objective minimization problem in the following section. The maximization of *ML-HSIC* and *ML-NCL* can be easily converted into a minimization problem.

A good EMO needs to generate a set of solutions that uniformly distributed along the Pareto optimal front [18], which includes two key issues: (1) solutions prone to converge to Pareto optimal front and maintain diversity in the evolutionary process; (2) generating promising solutions in each generation. In order to make EMO fit for multi-label learning, we design many novel mechanisms in the following two sections.

Multi-objective Optimization Mechanism. In this section, we apply the *non-dominated-sort* and *density-assignment* process to make the solutions converge to Pareto optimal front and maintain diversity, respectively.

Non-dominated-sort. The *non-dominated-sort* process sorts solutions according to their raw fitness (i.e. *ML-HSIC* and *ML-NCL*). Instead of the raw fitness, this paper employs the rank-based fitness assignment [7] to reassign the fitness (i.e. a rank value) to the solutions, because the rank-based fitness assignment behaves in a more robust manner. In the rank-based fitness assignment, the solution set is divided into different fronts according to their dominating relations of raw fitness. An example is shown in Figure 2 (*ML-HSIC* and *ML-NCL* are minimized

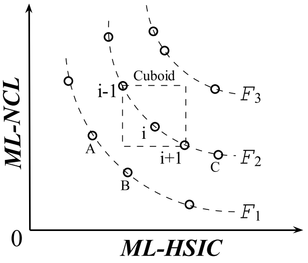


Fig. 2. Illustration of non-dominated-sorting and density-assignment process

here). The solutions in the same front are non-dominated to each other (e.g. solution A and B) and solutions in the higher front are always dominated by some solutions in the lower front (e.g. C in $\mathcal{F}_2$ is dominated by B in $\mathcal{F}_1$). In this way, each solution (i.e. base learner) h_i in a front $\mathcal{F}_a$ has a rank value $h_i^{rank} = a$, and solution h_i is better than solution h_j when $h_i^{rank} < h_j^{rank}$.

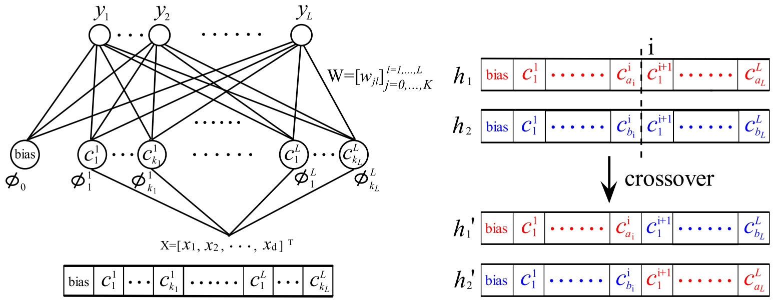
Density-assignment. Along with convergence to the Pareto optimal front, it is also desired that an Evolutionary Algorithm (EA) maintains a good spread of solutions. So the solution in the crowded region is more likely to be deleted. To get a density estimate of solutions surrounding a particular solution in the population, we design the *density-assignment* process that calculates the average distance of two solutions on either side of this solution along each of the objectives. It is simple and effective to estimate the density of solutions. The density estimation of solution h_i, $h_i^{density}$, serves as the perimeter of the cuboid formed by using the nearest neighbors as the vertices. As shown in Figure 2, the density of this i-th solution in its front is the average side length of the cuboid. The small $h_i^{density}$ means solution h_i is in a more crowded region. It implies the solution h_i should be more likely to be deleted.

Select-population. Every solution h_i in the population has two feature values: (1) non-domination rank h_i^{rank}; (2) density estimation $h_i^{density}$, which are determined by the raw fitness *ML-HSIC* and *ML-NCL*. Comprehensively considering both of the features, we define a partial order $\prec$ to compare two solutions. For two solutions h_i and h_j, $h_i \prec h_j$, if and only if

$$h_i^{rank} < h_j^{rank} \vee (h_i^{rank} = h_j^{rank} \wedge h_i^{density} > h_j^{density}) \tag{4}$$

That is, between two solutions with different non-domination ranks, we prefer the solution with the lower rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a less crowded region. After sorting the population with $\prec$, *select-population* process selects top solutions, and guarantees that good solutions will be kept.

Base Learner and Evolutionary Operators. In the framework of EnML, many multi-label base learners can be used, such as HMC tree [19], BP-MLL


y1
y2
yL
W=[wjl] l=1,...,L; j=0,...,K
bias
c11 ... ck11 ...... c1L ... ckLL
ϕ0 ϕ11 ϕk11 ϕ1L ϕkLL
X=[x1, x2, · · ·, xd] T
bias | c11 | ··· | ck11 | ······ | c1L | ··· | ckLL
i
h1 | bias | c11 | ······ | cai i | c1 i+1 | ······ | caL L
h2 | bias | c11 | ······ | cbi i | c1 i+1 | ······ | cbL L
crossover
h1' | bias | c11 | ······ | cai i | c1 i+1 | ······ | cbL L
h2' | bias | c11 | ······ | cbi i | c1 i+1 | ······ | caL L

(a) Genetic representation (b) Crossover operation

Fig. 3. (a) Architecture of RBF and its corresponding genetic representation. (b) The crossover operation. The crossover point i is selected between two prototype vectors.

[23] and ML-RBF [21]. Different types of base learners will lead to different genetic representation and operation. Because the structure can be effectively encoded and the weights can be efficiently calculated in close form, we select the RBF neural network in ML-RBF [21] as the multi-label base learner in EnML, however an additional regularization term is added to reduce overfitting risks.

The architecture of RBF is shown in Figure 3(a). It is described as follows: (1) The input of a RBF corresponds to a d-dimension feature vector. (2) The hidden layer of RBF is composed of L sets of prototype vectors, i.e. $\bigcup_{l=1}^{L} C_l$. Here, C_l consists of k_l prototype vectors $\{c_1^l, c_2^l, \cdots, c_{k_l}^l\}$. For each class $l \in \mathcal{L}$, the k-means clustering is performed on the set of instances U_l with label l. Thereafter, k_l clustered groups are formed for class l and the j-th centroid $(1 \leq j \leq k_l)$ is regarded as a prototype vector c_j^l of basis function $\phi_j^l(\cdot)$. (3) Each output neuron is related to a possible class. In the hidden layer of RBF, the number of clusters k_l is a fraction α of the number of instances in U_l:

$$k_l = \alpha \times |U_l| \tag{5}$$

The scale coefficient α controls the structure and complexity of RBF base learner.

Different from the error function in original RBF, we add a regularization term into the error function. The regularization term greatly reduces the overfitting risk and improves the stability of solutions as observed in the experiments.

$$E = \frac{1}{2}\sum_{i=1}^{m}\sum_{l=1}^{L}(y_l(\mathbf{x}_i) - t_l^i)^2 + \gamma\sum_{j=0}^{K}\sum_{l=1}^{L} w_{jl}^2 \tag{6}$$

where $y_l(\mathbf{x}_i)$ represents the predicted value of instance $\mathbf{x}_i$ on label l, t_l^i is the real value of instance i on label l, $K = \sum_{l=1}^{L} k_l$, and γ is the regularization coefficient. Similar to the derivation of minimizing the error function by scaled-conjugate-gradient descent in [3], the optimal output weights W can be computed in closed form by

$$W = (\varPhi'\varPhi + \gamma I)^{-1}\varPhi' T \tag{7}$$

Here $\Phi = [\phi_{ij}]_{m\times(K+1)}$ with elements $\phi_{ij} = \phi_j(\mathbf{x}_i)$, $W = [w_{jl}]_{(K+1)\times L}$ with elements w_{jl}, and $T = [t_{il}]_{m\times L}$ with elements $t_{il} = t_l^i$. Through extensive experiments, the regularization coefficient γ is fixed at 0.1 in this paper.

Genetic representation. According to the structure of RBF, we propose a novel genetic representation that is the sequence of prototypes $\{bias, c_1^1, c_1^2, \cdots c_{k_L}^L\}$. An example is shown in Figure 3(a). The genetic representation has the following advantages. (1) When the prototypes (c) are determined, the basis functions (ϕ) and the weights (W) can be efficiently computed. It means the performance of RBF mostly depends on the selection of the prototypes. (2) It is easy to design the crossover and mutation operators by tuning these prototypes.

Initialization. When the base learner is RBF, the initialization operation of EnML generates a set of RBF learners with different scale coefficient α (see Equation 5). As suggested in [21], α is randomly selected from [0.01, 0.02] in the experiments, which generates a set of RBF base learners with different structures.

Generate-offspring. Generating new solutions is realized by the *generate-offspring* process. The basic idea is to randomly select parent solutions from the current population based on the roulette wheel selection [1,3] and do crossover and mutation operation to generate new solutions with the ratio of *cro_Rat* and $1-$*cro_Rat* respectively. Following the general rule in EA, *cro_Rat* is fixed at 0.8 in this paper.

The roulette wheel selection [1,3] assigns each solution with the appropriate selection pressure, and guarantees the better solution with a high and appropriate selected probability. This paper adapts the cut and splice crossover [9] which randomly chooses a crossover point for two RBFs and swaps their prototypes beyond this point. Different from traditional cut and splice crossover, the crossover point in EnML is randomly selected between two prototype vectors, rather than in a arbitrary position. It guarantees that each prototype vector in the newly generated RBF is unabridged cluster centroids. Figure 3(b) shows an example of crossover operation. The mutation operator randomly selects some prototype vectors in a RBF, and does the following two structural mutation operations with the same probability. (1) Randomly delete a prototype. (2) Add one prototype whose center is determined by a random combination of all centroids in this prototype vector. The width of the centroid of the new RBF is recalculated as in [21]. The weights are calculated following Equation 7.

3.3 Algorithm Framework

EnML is described in Algorithm 1. In the model training phase, EnML transforms the optimized objectives (i.e. *ML-HSIC* and *ML-NCL*) to a fitness measure by the creation of a number of fronts, sorted according to *non-dominated-sort*. After the fronts have been created, *density-assignment* assigns its members with a density value later to be used for diversity maintenance. In each generation, N new solutions are generated with *generate-offspring*. Of the $2N$ solutions, *select-population* selects the N best solutions for the next generation. In this way, a huge elite can be kept and optimized from generation to generation. In the testing phase, all solutions predict labels of unseen data and combine their

Algorithm 1. EnML

```
Input:
$\mathcal{D}$: training data    $\mathcal{U}$: testing data    $h$: base learner
$N$: # of base learners    $G$: # of generations
output:
$Y(\mathbf{x})$: predicted labels for instance $\mathbf{x} \in \mathcal{U}$
procedure Training
    generate $P_0 = \{h_1, h_2, \cdots, h_N\}$ on $\mathcal{D}$ at random
    $P_1$=$(\mathcal{F}_1, \mathcal{F}_2, \cdots)$=$non\text{-}dominated\text{-}sort(P_0)$
    for $t = 1 : G$ do
        $Q_t$=$generate\text{-}offspring(P_t)$
        $R_t = P_t \bigcup Q_t$
        $F = (\mathcal{F}_1, \mathcal{F}_2, \cdots)$=$non\text{-}dominated\text{-}sort(R_t)$
        $density\text{-}assignment(F)$
        $P_{t+1}$=$select\text{-}population(F)$
        $t = t + 1$
    end for
end procedure
procedure Testing
    For $\mathbf{x} \in \mathcal{U}$, label set $Y(\mathbf{x}) = \{l | \frac{1}{N} \sum_{i=1}^{N} h_i(\mathbf{x}, l) > 0; h_i \in P_t, l \in \mathcal{L}\}$
end procedure
```

Table 1. Summary of experimental datasets

	Dataset						
Characteristic	Image	Yeast	Arts	Health	Science	Recreation	Entertain.
# of instances	2000	2417	5000	5000	5000	5000	5000
# of features	294	103	462	612	743	606	640
# of labels	5	14	26	32	40	22	21
domain	biology	media	text	text	text	text	text

results with a simple vote. Note that EnML can not only optimize *ML-HSIC* and *ML-NCL* but also directly optimize either of these two objectives.

4 Experiments

4.1 Experimental Setup

Data Collections: We tested our algorithm on seven real-world multi-label classification datasets from three different domains as summarized in Table 1. The first dataset is *Yeast* [15,21,23,25] in biology, where the task is to predict the gene functional classes of the Yeast Saccharomyces cerevisiae. The second dataset *Image* [15,21,23,25] involves the task of automatic image annotation for scene images. The other five dataset are from Yahoo [21,24], where the task is to predict topic categories of each text document.

Evaluation Metrics: Here we adopt five state-of-the-art multi-label evaluation metrics which are most popular in the literature. Assume we have a multi-label dataset $\mathcal{U}$ containing n multi-label instances $(\mathbf{x}_i, Y_i)$. Let $h(\mathbf{x}_i)$ denote the predicted label set of a multi-label learner h for $\mathbf{x}_i$, and the real-valued function

$f(\mathbf{x}_i, y_l) \in R$ represents the ranking quality score of learner h on label y_l for input $\mathbf{x}_i$. We have the following evaluation criteria:

• *hamming loss* [5,22]: evaluates the number of labels whose relevance is incorrectly predicted.

$$hammingloss(h,\mathcal{U}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{L}\|h(\mathbf{x}_i) \oplus Y_i\|_1 \tag{8}$$

where $\bigoplus$ stands for the symmetric difference of two sets (XOR operation), and $\|.\|_1$ denotes the l_1-norm. The smaller the value, the better the performance.

• *ranking loss* [6,25]: evaluates the average fraction of label pairs that are misordered for the instance.

$$rankingloss(h,\mathcal{U}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\overline{Y_i}|}|R_i| \tag{9}$$

where $R_i = \{(y_1, y_2)|f(\mathbf{x}_i, y_1) \leq f(\mathbf{x}_i, y_2), (y_1, y_2) \in Y_i \times \overline{Y_i}\}$. Here $\overline{Y_i}$ denotes the complementary set of Y_i in Y. The smaller the value, the better the performance.

• *one-error* [6,25]: evaluates how many times the top-ranked label is not in the set of proper labels of the instance.

$$one\text{-}error(h,\mathcal{U}) = \frac{1}{n}\sum_{i=1}^{n}[\![[argmax_{y\in\mathcal{L}} f(\mathbf{x}_i, y)] \notin Y_i]\!] \tag{10}$$

Here for predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise. The smaller the value, the better the performance.

• *coverage* [6,25]: evaluates how many steps are needed, on average, to move down the ranked label list in order to cover all the proper labels of the instance.

$$coverage(h,\mathcal{U}) = \frac{1}{n}\sum_{i=1}^{n} max_{y\in Y_i} rank^f(\mathbf{x}_i, y) - 1 \tag{11}$$

$rank^f(\cdot,\cdot)$ is derived from the real-valued function $f(\cdot,\cdot)$. If $f(\mathbf{x}_i, y_1) > f(\mathbf{x}_i, y_2)$, then $rank^f(\mathbf{x}_i, y_1) < rank^f(\mathbf{x}_i, y_2)$. The smaller the value, the better the performance.

• *average precision* [6,25]: evaluates the average fraction of proper labels ranked above a particular label $y \in Y_i$.

$$avgprec(h,\mathcal{U}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|P_i|}{rank^f(\mathbf{x}_i, y)} \tag{12}$$

where $P_i = \{y'|rank^f(\mathbf{x}_i, y') \leq rank^f(\mathbf{x}_i, y), y' \in Y_i\}$. The larger the value, the better the performance.

Compared Methods: In order to test performance of our proposed EnML, we do comprehensive comparison with the most representative multi-label learning approaches, including ML-RBF [21], the base learner of our approach, and two ensemble based approaches: ECC [15] and RAKEL [17]. In addition, in order to validate the effectiveness of two objective functions, we include two special cases of EnML that only optimize one single objective (i.e. *ML-HISC* or *ML-NCL*). These approaches are briefly summarized as follows.

- EnML: the proposed approach in this paper. It simultaneously optimizes two objectives: *ML-HSIC* and *ML-NCL*.
- EnML$_{HSIC}$: a special case of EnML, which only optimizes *ML-HSIC*.
- EnML$_{NCL}$: a special case of EnML, which only optimizes *ML-NCL*.
- ML-RBF [21]: the multi-label learning algorithm based on RBF neural network, which is also the base learner we use in EnML.
- ECC [15]: an ensemble method for multi-label learning based on the bagging of classifier chains.
- RAKEL [17]: another ensemble method for multi-label learning, where the single-label base learner is trained for a small random subset of labels.

In order to fit for EnML as a minimization problem, we convert the original objectives into an equivalent minimization problem as follows:

$$\begin{aligned} \textit{ML-HSIC}' &= 1/log(\textit{ML-HSIC}) \\ \textit{ML-NCL}' &= 1 - \textit{ML-NCL}/(m \times L) \end{aligned} \tag{13}$$

Note that the two new objectives both need to be minimized and fall in [0,1], such that it is convenient to perform *non-dominated-sort* and *density-estimate* process in our evolutionary multi-objective optimization algorithm. As in [21], ML-RBF is implemented with fixed default parameters ($\alpha = 0.01$ and $\mu = 1.0$). For ECC, the ensemble size is set to 10 and sampling ratio is set to 67% as suggested in the literature [15]. For RAKEL [17], we always set the parameter k as $\frac{|L|}{2}$ to provide the highest accuracy. The population size and running generation of EnML based approaches are set as 30 and 10 respectively in all experiments.

4.2 Performance Comparison

We perform ten-fold cross-validation on each experimental dataset. On each dataset, we report the mean values performance and standard deviations for each algorithm with the rank based on its results indicated in the parentheses. All experiments are conducted on machines with Intel Xeon Quad-Core CPUs of 2.26 GHz and 24 GB RAM.

The performances of six algorithms are shown in Table 2 to Table 6. It is clearly shown that EnML significantly outperforms the other baseline methods, including the non-ensemble method ML-RBF and two ensemble methods ECC and RAKEL, on all criteria and datasets. The small standard deviations of the rank values of EnML (ranging from 0 to 0.49) also indicate the superior of EnML is consistent on all datasets and evaluated criteria. The results illustrate that the

Table 2. Performance (mean±std.(rank)) of each algorithm in terms of *hamming loss*. Ave. Rank represents the mean and standard deviation of the rank values of each algorithm in all datasets.

	Algorithm					
Dataset	EnML	ML-RBF	ECC	RAKEL	EnML$_{HSIC}$	EnML$_{NCL}$
Image	0.1603±0.0058(2)	0.1653±0.0067(3)	0.1786±0.0108(6)	0.1724±0.0117(5)	0.1586±0.0065(1)	0.1665±0.0051(4)
Yeast	0.1889±0.0052(2)	0.1935±0.0058(4)	0.2056±0.0082(5)	0.2287±0.0105(6)	0.1887±0.0064(1)	0.1894±0.0059(3)
Arts	0.0531±0.0014(2)	0.0542±0.0016(4)	0.0754±0.0045(6)	0.0612±0.0013(5)	0.0528±0.0014(1)	0.0538±0.0015(3)
Health	0.0316±0.0016(2)	0.0331±0.0016(4)	0.0361±0.0021(5)	0.0373±0.0016(6)	0.0314±0.0017(1)	0.0322±0.0017(3)
Science	0.0317±0.0008(2)	0.0324±0.0009(4)	0.0424±0.0054(6)	0.0360±0.0016(5)	0.0313±0.0010(1)	0.0320±0.0008(3)
Recreation	0.0543±0.0023(2)	0.0555±0.0022(4)	0.0688±0.0055(6)	0.0589±0.0028(5)	0.0539±0.0021(1)	0.0553±0.0025(3)
Entertain.	0.0502±0.0016(2)	0.0512±0.0016(3)	0.0654±0.0053(6)	0.0587±0.0030(5)	0.0496±0.0013(1)	0.0514±0.0012(4)
Ave. Rank	2.00±0.00	3.71±0.49	5.71±0.49	5.29±0.49	1.00±0.00	3.29±0.49

Table 3. Performance (mean±std.(rank)) of each algorithm in terms of *ranking loss*

	Algorithm					
Dataset	EnML	ML-RBF	ECC	RAKEL	EnML$_{HSIC}$	EnML$_{NCL}$
Image	0.1478±0.0112(1)	0.1558±0.0121(4)	0.2411±0.0153(6)	0.1765±0.0200(5)	0.1485±0.0112(2)	0.1536±0.0106(3)
Yeast	0.1597±0.0083(1)	0.1621±0.0073(4)	0.2776±0.0223(6)	0.2179±0.0156(5)	0.1603±0.0087(2)	0.1619±0.0073(3)
Arts	0.1119±0.0099(1)	0.1131±0.0098(3)	0.3814±0.0251(6)	0.2589±0.0106(5)	0.1150±0.0104(4)	0.1124±0.0093(2)
Health	0.0482±0.0057(1)	0.0496±0.0051(3)	0.2401±0.0130(6)	0.1822±0.0125(5)	0.0505±0.0056(4)	0.0490±0.0054(2)
Science	0.0957±0.0072(1)	0.1002±0.0071(3)	0.3840±0.0238(6)	0.2854±0.0138(5)	0.1017±0.0079(4)	0.0992±0.0072(2)
Recreation	0.1216±0.0101(1)	0.1253±0.0099(3)	0.3434±0.0203(6)	0.2874±0.0227(5)	0.1257±0.0118(4)	0.1229±0.0095(2)
Entertain.	0.0913±0.0070(1)	0.0946±0.0073(3)	0.2926±0.0193(6)	0.2874±0.0221(5)	0.0949±0.0073(4)	0.0933±0.0062(2)
Ave. Rank	1.00±0.00	3.29±0.49	6.00±0.00	5.00±0.00	3.43±0.98	2.29±0.49

Table 4. Performance (mean±std.(rank)) of each algorithm in terms of *one error*

	Algorithm					
Dataset	EnML	ML-RBF	ECC	RAKEL	EnML$_{HSIC}$	EnML$_{NCL}$
Image	0.2735±0.0236(2)	0.2860±0.0299(4)	0.2935±0.0249(5)	0.3065±0.0335(6)	0.2695±0.0247(1)	0.2815±0.0208(3)
Yeast	0.2156±0.0235(1)	0.2189±0.0175(3)	0.2742±0.0218(5)	0.2751±0.0300(6)	0.2193±0.0286(4)	0.2160±0.0210(2)
Arts	0.4400±0.0134(2)	0.4512±0.0124(4)	0.4734±0.0291(5)	0.5470±0.0137(6)	0.4314±0.0177(1)	0.4450±0.0130(3)
Health	0.2416±0.0204(2)	0.2482±0.0250(4)	0.2430±0.0183(3)	0.2946±0.0184(6)	0.2398±0.0230(1)	0.2494±0.0219(5)
Science	0.4862±0.0185(2)	0.5016±0.0181(5)	0.5008±0.0432(4)	0.5784±0.0199(6)	0.4794±0.0174(1)	0.4916±0.0187(3)
Recreation	0.4492±0.0159(2)	0.4542±0.0220(3)	0.4618±0.0196(5)	0.5304±0.0281(6)	0.4398±0.0187(1)	0.4548±0.0132(4)
Entertain.	0.3824±0.0241(2)	0.3916±0.0251(5)	0.3836±0.0309(3)	0.4746±0.0278(6)	0.3816±0.0222(1)	0.3914±0.0234(4)
Ave. Rank	1.86±0.38	4.00±0.82	4.29±0.95	6.00±0.00	1.47±1.13	3.43±0.98

Table 5. Performance (mean±std.(rank)) of each algorithm in terms of *coverage*

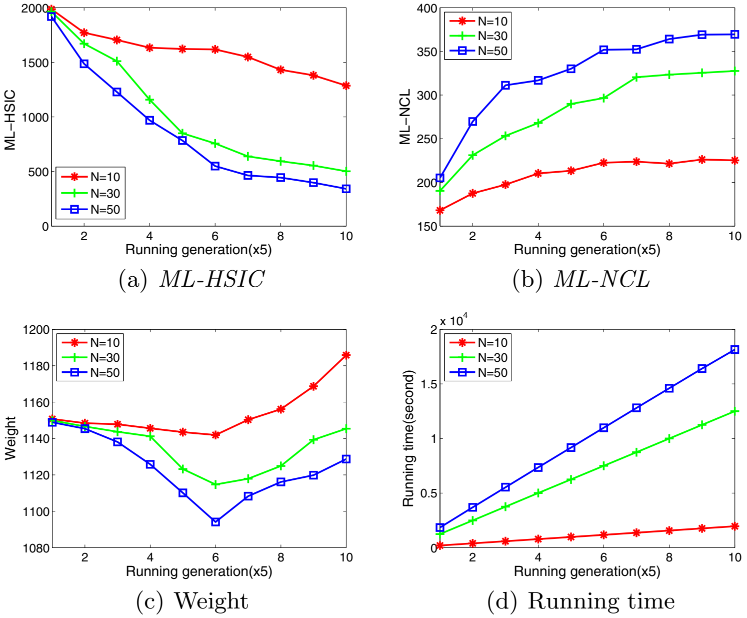
	Algorithm					
Dataset	EnML	ML-RBF	ECC	RAKEL	EnML$_{HSIC}$	EnML$_{NCL}$
Image	0.8740±0.0548(2)	0.8955±0.0562(4)	0.9715±0.0776(5)	0.9795±0.0831(6)	0.8570±0.0487(1)	0.8900±0.0468(3)
Yeast	6.1845±0.1465(1)	6.2465±0.1433(4)	7.1431±0.2688(5)	7.5347±0.2521(6)	6.2138±0.1353(2)	6.2453±0.1384(3)
Arts	4.5738±0.4115(1)	4.6116±0.3783(3)	7.8582±0.4686(5)	8.8862±0.3652(6)	4.7192±0.4110(4)	4.5808±0.3720(2)
Health	3.1998±0.3144(1)	3.2280±0.2797(3)	8.2418±0.3797(5)	8.7686±0.4684(6)	3.2930±0.2942(4)	3.2122±0.3042(2)
Science	5.5188±0.4325(1)	5.6016±0.4341(3)	11.403±0.4453(5)	13.744±0.6340(6)	5.7114±0.4432(4)	5.5476±0.4108(2)
Recreation	3.6858±0.2916(1)	3.7452±0.2983(3)	6.2390±0.4696(5)	7.6552±0.6209(6)	3.7790±0.3304(4)	3.6872±0.2831(2)
Entertain.	2.7686±0.1832(1)	2.8102±0.1739(3)	5.7008±0.2569(5)	7.1750±0.4761(6)	2.8478±0.1885(4)	2.7796±0.1434(2)
Ave. Rank	1.14±0.38	3.29±0.49	5.00±0.00	6.00±0.00	3.29±1.25	2.29±0.49

Table 6. Performance (mean±std.(rank)) of each algorithm in terms of *average precision*

	Algorithm					
Dataset	EnML	ML-RBF	ECC	RAKEL	EnML$_{HSIC}$	EnML$_{NCL}$
Image	0.8288±0.0144(1)	0.8118±0.0145(4)	0.7977±0.0148(5)	0.7952±0.0215(6)	0.8226±0.0141(2)	0.8139±0.0122(3)
Yeast	0.7754±0.0146(1)	0.7720±0.0133(4)	0.7313±0.0236(5)	0.7170±0.0165(6)	0.7747±0.0152(2)	0.7734±0.0127(3)
Arts	0.6433±0.0113(2)	0.6366±0.0116(4)	0.5613±0.0149(5)	0.5122±0.0138(6)	0.6473±0.0123(1)	0.6406±0.0112(3)
Health	0.7988±0.0135(2)	0.7941±0.0151(4)	0.7247±0.0115(5)	0.6986±0.0142(6)	0.7994±0.0142(1)	0.7957±0.0132(3)
Science	0.6128±0.0152(2)	0.6026±0.0155(4)	0.5328±0.0227(5)	0.4712±0.0213(6)	0.6178±0.0172(1)	0.6090±0.0167(3)
Recreation	0.6501±0.0159(2)	0.6435±0.0170(4)	0.5770±0.0145(5)	0.5355±0.0242(6)	0.6520±0.0164(1)	0.6448±0.0134(3)
Entertain.	0.7028±0.0146(2)	0.6971±0.0169(4)	0.6338±0.0151(5)	0.5763±0.0232(6)	0.7029±0.0142(1)	0.6976±0.0143(3)
Ave. Rank	1.71±0.49	4.00±0.00	5.00±0.00	6.00±0.00	1.29±0.49	3.00±0.00

ensemble in our EnML can effectively improve the generalization performance in multi-label learning, compared to non-ensemble methods (e.g. ML-RBF). In addition, the superior of EnML over those ensemble methods for multi-label learning (e.g. ECC and RAKEL) also confirms our assumption: the ensemble of multi-label base learners is more effective to improve the generalization ability of multi-label learning system than the ensemble of single-label base learners. We think one of the important reasons behind the performance improvement of EnML lies in our EnML emphasizes the diversity of multi-label base learners by explicitly optimizing a diversity-related objective, which has never been done in multi-label learning so far.

Then we further study the effect of objective functions in our EnML method on the performances by comparing EnML with EnML$_{HSIC}$ and EnML$_{NCL}$. From Table 2 to Table 6, we can also observe that the three versions of EnML rank top three on most criteria and they always have the best performance on each dataset. By optimizing the diversity-related objective *ML-NCL*, EnML$_{NCL}$ generates a set of diverse base learners, so EnML$_{NCL}$ outperforms the base learner ML-RBF on most criteria. However, without optimizing the accuracy of individual base learner, EnML$_{NCL}$ performs worse than EnML on all criteria. Although EnML$_{HSIC}$ can achieve a little better performances than EnML in *hamming loss*, *one-error*, and *average precision* on some datasets, however, on the other two criteria, *ranking loss* and *coverage*, EnML$_{HSIC}$ is not only worse than EnML and EnML$_{NCL}$, but also worse than the base learner ML-RBF. It can be explained that EnML$_{HSIC}$ optimizes the accuracy-related objective *ML-HSIC*, which makes it perform well on the *ML-HSIC* related criteria, such as *hamming loss*, *one-error*, and *average precision.* However, without emphasizing the diversity of base learners, the optimal base learners obtained by EnML$_{HSIC}$ can be very similar with each other, thus the generalization ability of the ensemble can be weak. By considering the accuracy and diversity objectives simultaneously, EnML can obtain a group of accurate and diverse multi-label base learners and the population evolutionary strategy in EnML automatically finds the optimal trade-off between these two objectives. As a consequence, EnML consistently improves the generalization ability of multi-label ensemble, thus it comprehensively boosts the multi-label classification performances.



(a) *ML-HSIC*

(b) *ML-NCL*

(c) Weight

(d) Running time

Fig. 4. The evolutionary performances of EnML with different parameter settings

4.3 Parameter Settings

In this section, we study the effects of the parameters in our EnML method. There are two genetic operation related parameters in EnML, i.e. the population size N and the running generation G. We perform the experiment on 2000 instances of the Arts dataset in Yahoo dataset collection [21,24] with ten-fold cross-validation under different parameter configurations. Specifically, when the population size N is set as 10, 30, and 50, we report the average of objective values, running time and weights (sum of absolute values in W). The results are shown in Figure 4.

From Figure 4(a) and (b), we can clearly observe that *ML-NCL* goes up but *ML-HSIC* goes down when the running generation increases. The different trend of these two objectives indicate that they have the intrinsic conflict. It is not surprising. The maximization of the *ML-HSIC* guides the predicted labels of base learners to converge to the real labels. So it makes these base learners identical. However, the maximization of the *ML-NCL* encourages base learners to be as diverse as possible on the training error. Therefore, these two objectives are naturally conflicting. The conflict makes EnML seek to find a good balance between the two objectives by population optimization. Note that here *ML-HSIC* and *ML-NCL* just evaluate the accuracy and diversity of base learners, not the performance of the ensemble. The decrease of *ML-HSIC* does not mean the degradation of the ensemble. In fact, the increase of *ML-NCL* shows that base learners become more diverse, which helps to improve the performance of the ensemble. Figure 4(c) shows that the weight goes down and then goes up when the running generation increases. We think the reason is that *ML-NCL* helps to control the model complexity. However, when the running generation becomes too large, these learners become more complex, and thus their weights increase. If we do not add the regularization term in the error function of RBF (see Equation 6), the weights will increase sharply, which means these learners are overfitting. Figure 4(d) illustrates that the running time of EnML increases linearly with the population size N and running generation G.

5 Conclusion

In this paper, we first study the multi-label ensemble learning problem which aims at building a set of accurate and diverse multi-label base learners to improves the generalization ability of multi-label learning system. In order to solve this problem, we propose a novel solution EnML. With an evolutionary multi-objective optimization method, EnML simultaneously optimizes two objective functions that evaluate the accuracy and diversity of multi-label learners, respectively, and constructs a set of accurate and diverse multi-label base learners to make predictions. Extensive experiments show that EnML can effectively improve the generalization ability of multi-label learning system and thus boosts the predictive performance for multi-label classification.

Acknowledgments. It is supported by the National Natural Science Foundation of China (No. 60905025, 61074128, 61035003) and the 863 Program of China No.2009AA04Z136.

References

1. Baker, J.: Adaptive Selection Methods for Genetic Algorithms. In ICGA, pp. 100–111 (1985)
2. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
3. Chen, H., Yao, X.: Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. Transactions on Knowledge and Data Engineering 22(12), 1738–1751 (2010)
4. Deb, K.: Multiobjective Optimization using Evolutionary Algorithms. Wiley, UK (2001)
5. Dembczynski, K., Cheng, W., Hullermeier, E.: Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. In: ICML, pp. 279–286 (2010)
6. Elisseeff, A., Weston, J.: A Kernel Method for Multilabelled Classification. In: NIPS, pp. 681–687 (2002)
7. Goldberg, D.E.: Generic Algorithms in Search Optimization and Machine Learning, USA, Boston (1989)
8. Gretton, A., Bousquet, O., Smola, A., Scholkopf, B.: Measuring Statistical Dependence with Hilbert-Schmidt Norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005)

9. Goldberg, D., Deb, K., Kargupta, H., Harik, G.: Rapid, Accurate Optimization of Difficult Problems using Fast Messy Genetic Algorithms. In: ICGA, pp. 56–64 (1993)
10. Krogh, A., Vedelsby, J.: Neural Network Ensembles, Cross Validation, and Active Learning. In: NIPS, pp. 231–238 (1995)
11. Liu, Y., Yao, X.: Ensemble Learning via Negative Correlation. Neural Networks 12(10), 1399–1404 (1999)
12. Liu, Y., Yao, X.: Simultaneous Training of Negatively Correlated Neural Networks in an Ensemble. Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics 29(6), 716–725 (1999)
13. Petterson, J., Caetano, T.: Reverse Multi-label Learning. In: NIPS (2010)
14. Read, J., Pfahringer, B., Holmes, G.: Multi-label Classification using Ensembles of Pruned Sets. In: ICDM, pp. 995–1000 (2008)
15. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: ECML, pp. 254-269 (2009)
16. Tsoumakas, G., Katakis, I., Vlahavas, I. P.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: ECML/PKDD Workshop (2008)
17. Tsoumakas, G., Vlahavas, I.P.: 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)
18. Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the state-of-the-art. Evolutionary Computation 18(2), 125–147 (2000)
19. Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., Blockeel, H.: Decision Tree for Hierarchical Multi-label Classification. Machine Learning 2(73), 185–214 (2008)
20. Yang, B.S., Sun, J.T., Wang, T.J., Chen Z.: Effective Multi-label Active Learning for Text Classification. In: KDD, pp. 917–925 (2009)
21. Zhang, M.L.: ML-RBF: RBF Neural Networks for Multi-label Learning. Neural Process Letters 29(2), 61–74 (2009)
22. Zhang, X., Yuan, Q., Zhao, S., Fan, W., Zheng, W., Wang, Z.: Multi-label Classification without the Multilabel Cost. In: SDM, pp. 778–789 (2010)
23. Zhang, M.L., Zhou, Z.H.: Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)
24. Zhang, M.L., Zhou, Z.H.: Ml-knn: a Lazy Learning Approach to Multi-label Learning. Pattern Recognition 40(7), 2038–2048 (2007)
25. Zhang, M.L., Zhang, K.: Multi-label Learning by Exploiting Label Dependency. In: KDD, pp. 999–1007 (2010)