Multi-Label Hypothesis Reuse[*]

Sheng-Jun Huang, Yang Yu, and Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210046, China
{huangsj, yuy, zhouzh}@lamda.nju.edu.cn

ABSTRACT

Multi-label learning arises in many real-world tasks where an object is naturally associated with multiple concepts. It is well-accepted that, in order to achieve a good performance, the relationship among labels should be exploited. Most existing approaches require the label relationship as prior knowledge, or exploit by counting the label co-occurrence. In this paper, we propose the MAHR approach, which is able to automatically discover and exploit label relationship. Our basic idea is that, if two labels are related, the hypothesis generated for one label can be helpful for the other label. MAHR implements the idea as a boosting approach with a *hypothesis reuse* mechanism. In each boosting round, the base learner for a label is generated by not only learning on its own task but also reusing the hypotheses from other labels, and the amount of reuse across labels provides an estimate of the label relationship. Extensive experimental results validate that MAHR is able to achieve superior performance and discover reasonable label relationship. Moreover, we disclose that the label relationship is usually asymmetric.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

General Terms

Algorithm, Experimentation

Keywords

Multi-label learning, label relationship, hypothesis reuse

1. INTRODUCTION

In traditional supervised classification, one instance is associated with one target concept, whereas in many real-world applications, one instance is naturally associated with multiple concepts. For example, in scene classifications (e.g., [1]) a natural scene picture can be annotated as *sky*, *trees*, *mountains*, *lakes* and *water* simultaneously; in text categorizations (e.g., [25]) a piece of news on global warming issue can be categorized as *global warming*, *environment*, *economics* and *politics*.

Multi-label learning tries to address such kind of tasks. The most straightforward solution to multi-label learning is to decompose the task into a series of binary classification problems, each for one label [1]; however, such a solution neglects the relationship among labels, whereas previous studies [8, 16] have revealed that the label relationship is quite helpful and should be considered. Later on, most multi-label learning approaches try to exploit label relationship explicitly. Some approaches [4, 12, 19, 5] rely on external knowledge resources, such as knowledge on label hierarchies from which the label relationship is derived; label relationship obtained in this way is generally helpful, however, label hierarchies are often unavailable in real applications. Some other approaches [10, 24] try to exploit label relationship by counting the co-occurrence of labels in training data; although such approaches can be effective in some cases, there are high risks of overfitting the training data.

In this paper, we propose a novel multi-label learning approach named MAHR (Multi-lAbel Hypothesis Reuse), which is able to discover and exploit label relationship automatically in the learning process. Our basic idea is that, if two labels are related, the hypothesis generated for one label can be helpful for the other. MAHR implements the idea as a boosting approach with a *hypothesis reuse* mechanism. It trains multiple boosted learners simultaneously, each for one label. In each boosting round, in addition to generating a base learner for each label from its own hypothesis space, MAHR tries to *reuse* the hypotheses generated for other labels. The reuse process takes into account all trained hypotheses from all other labels via weighted combination, and helpful hypotheses are identified and reused with large weights through minimizing the loss on the concerned label. The *reuse score*, which is calculated from the weights of cross-label hypothesis reuse throughout the boosting process, provides an estimate of label relationship.

Extensive experiments show that the MAHR approach is superior or highly competitive to state-of-the-art multi-label learning approaches. In addition to the superior predictive

[*] This research was supported by the 973 Program (2010CB327903), NSFC (61073097, 60975043), JiangsuSF (BK2011566) and Baidu fund.

KDD'12, August 12–16, 2012, Beijing, China.

performance, as a prominent advantage, MAHR is able to discover reasonable label relationship. Moreover, it is worth mentioning that our study disclose that the label relationship is usually asymmetric, quite different from symmetric label relationship assumed in many previous studies.

The rest of the paper is organized as follows. Section 2 introduces related work, Section 3 presents MAHR, Section 4 reports on experiments, and Section 5 concludes.

2. RELATED WORK

During the past few years, many multi-label approaches have been developed [17], such as the decomposition-based approaches [1], ranking-based approaches [3], common subspace approaches [14], generative approaches [16, 25], etc.

Some approaches assume that the label relationship can be provided by external knowledge resources as prior knowledge. For example, label hierarchies are required by [4, 19, 5], and a label similarity matrix is required as an input in [12]. In most real-world tasks, however, prior knowledge of label relationship is often unavailable.

Some approaches try to model the label relationship directly. For example, Ghamrawi and McCallum [10] tried to model the impact of an individual feature on the co-occurrence probability of label pairs. Sun et al. [23] used a hypergraph to model the correlation information of labels. Zhang and Zhang [28] used Bayesian network structure to encode the conditional dependencies of both the labels and feature sets. Usually, the label relationship is estimated by considering the co-occurrence or some kinds of equivalence of the labels, and is easy to overfitting the training data.

Tsoumakas et al. [24] proposed the 2BR approach and utilized label relationship to prune stacked single-label classifiers. The label relationship can be measured based on co-occurrence, or their proposed ϕ-coefficient. The ϕ-coefficient for two labels L_i and L_j is defined as $\phi(i,j) = (AD - BC)/\sqrt{(A+B)(C+D)(A+C)(B+D)}$, where A, B, C and D are the frequency counts of $L_i \wedge L_j$, $L_i \wedge \neg L_j$, $\neg L_i \wedge L_j$ and $\neg L_i \wedge \neg L_j$, respectively. We will employ the 2BR approach as a test bed to compare the effectiveness of the *reuse score* generated by our MAHR approach with that of the label co-occurrence and the ϕ-coefficient.

Boosting is a family of learning algorithms with sound theoretical foundation and wide applications. In [20], two boosting approaches for multi-label learning, AdaBoost.MH and AdaBoost.MR, were proposed. They both train additive models to directly optimize multi-label losses, i.e., *hamming loss* for AdaBoost.MH and *ranking loss* for AdaBoost.MR. AdtBoost.MH [6], an extension of AdaBoost.MH, is able to produce a set of comprehensible rules through employing alternating decision trees. MSSBoost [27] maintains a shared pool of base classifiers to reduce the redundancy among labels, where each base classifier is trained in a random subspace and with a random sample; in each round, it selects the best classifier from the pool and takes it as the base classifier for all the labels.

A common idea of these boosting approaches lies in the fact that, in each learning round, one base classifier is generated for all the labels (while only the output threshold may be adjusted for each label). Such a process may suffer from some deficiencies. First, in many cases, especially when there are lots of labels, it is quite difficult to generate a base classifier that can deal well with all the labels simultaneously. Second, it is rare that all labels are strongly related [13], and thus it is not very reasonable to try to generate one base classifier for all labels. In contrast, our MAHR approach generates one base classifier for each label in each round, and does not assume that all labels are strongly related. Moreover, MAHR does not require label relationship as input, and instead, it will produce estimate of the label relationship as output.

3. MAHR

We denote by $\mathcal{D} = \{(\boldsymbol{x}_1, Y_1), (\boldsymbol{x}_2, Y_2), \cdots, (\boldsymbol{x}_m, Y_m)\}$ a multi-label data set of m instances and L possible labels, where $\boldsymbol{x}_i$ is the i-th instance and the corresponding Y_i is a label vector of dimension L, $Y_i(l) = 1$ if the l-th label is a proper label of $\boldsymbol{x}_i$, and -1 otherwise.

3.1 The Algorithm

MAHR, as shown in Algorithm 1, maintains the general outline of boosting. MAHR generates base hypotheses in an iterative manner (the loop from line 3 to line 12). In the round t, it generates a base hypothesis for each label (the loop from line 4 to line 11). For the l-th label, it firstly trains a hypothesis $\hat{h}_{t,l}$ from its own hypothesis space (line 5), and then invokes a *reuse function* R (line 6) to generate another hypothesis $h_{t,l}$ from both $\hat{h}_{t,l}$ and the reuse of the hypotheses in the candidate set $Q_{t,l}$. After that, $h_{t,l}$ is treated as the base hypothesis for label l in round t, and the *edge* of its training error (i.e., 0.5 minus the training error) and the combination weight are calculated (line 7). The training set is then updated for the next round (line 8). Note that the candidate set for reuse in round $t+1$ on label l is defined as $Q_{t+1,l} = \{H_{t,k} | k \neq l\} \cup \{-H_{t,k} | k \neq l\}$ (line 10), where $H_{t,k}$ is the combined hypothesis for label k up to round t (line 9). Here, $-H_{t,k}$ is included in $Q_{t+1,l}$ for simplicity of optimization.

The *reuse function* R is utilized to combine multiple hypotheses, and it can be implemented in different ways. In this paper, we implement R via weighted linear combination. Such a simple implementation leads to decent performance in experiments, and a better implementation of R may even improve the performance. Formally, given a hypothesis pool $Q_{t,l}$ and a newly generated hypothesis $\hat{h}_{t,l}$ at the t-th round on the l-th label, we define a parameterized reuse function $R_{\boldsymbol{\beta}_{t,l}}$ as

$$R_{\boldsymbol{\beta}_{t,l}}(\hat{h}_{t,l}, Q_{t,l}) = \boldsymbol{\beta}_{t,l}(\hat{h}_{t,l}) \cdot \hat{h}_{t,l} + \sum_{H \in Q_{t,l}} \boldsymbol{\beta}_{t,l}(H) \cdot H, \quad (1)$$

where $\boldsymbol{\beta}_{t,l}$ is a vector of reuse weights to be determined and $\boldsymbol{\beta}_{t,l}(H)$ denotes the element of $\beta_{t,l}$ corresponding to H. Different strategies can be employed to solve the parameter $\boldsymbol{\beta}$, aiming to minimize different multi-label losses (e.g., hamming loss and ranking loss).

For hamming loss, which concerns about how each predicted label is consistent with the ground-truth label additively, $\boldsymbol{\beta}_{t,l}$ can be obtained by:

$$\begin{aligned} \min_{\boldsymbol{\beta}_{t,l},\ \xi_i} & \sum\nolimits_{i=1}^{m} D_{t,l}(i) \cdot \xi_i \qquad (2) \\ s.t.\ \ & Y_i(l) R_{\boldsymbol{\beta}_{t,l}}(\boldsymbol{x}_i) \geq 1 - \xi_i \ (\forall i) \\ & \boldsymbol{\beta}_{t,l}(\hat{h}_{t,l}) + \sum\nolimits_{H \in Q} \boldsymbol{\beta}_{t,l}(H) = 1 \\ & \boldsymbol{\beta}_{t,l}(\hat{h}_{t,l}),\ \boldsymbol{\beta}_{t,l}(H),\ \xi_i \geq 0 \ (\forall H, i) \end{aligned}$$

Algorithm 1 The MAHR algorithm

```
1: Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^m$, base learning
   algorithm $\mathcal{L}$, reuse function $R$, number of rounds $T$
2: $D_{1,l}(i) = \frac{1}{m}$, $Q_{1,l} = \emptyset$ $(i = 1, \cdots, m,\ l = 1, \cdots, L)$
3: for $t = 1$ to $T$ do
4:   for $l = 1$ to $L$ do
5:     $\hat{h}_{t,l} \leftarrow \mathcal{L}(\mathcal{D}, D_{t,l})$
6:     $h_{t,l} \leftarrow R(\hat{h}_{t,l}, Q_{t,l})$ such that $h_{t,l}(\cdot) \in [-1, 1]$
7:     $\gamma_{t,l} = \frac{1}{2}\sum_i D_{t,l}(i) h_{t,l}(\boldsymbol{x}_i) Y_i(l)$; $\alpha_{t,l} = \frac{1}{2}\ln(\frac{0.5+\gamma_{t,l}}{0.5-\gamma_{t,l}})$
8:     $D_{t+1,l}(i) = \left(1 + \exp\left(Y_i(l)\sum_{j=1}^t \alpha_{j,l} h_{j,l}(x_i)\right)\right)^{-1}$
9:     $H_{t,l} = \sum_{j=1}^t \alpha_{j,l} h_{j,l}$
10:    $Q_{t+1,l} = \{H_{t,k} | k \neq l\} \cup \{-H_{t,k} | k \neq l\}$
11:  end for
12: end for
13: Output: $H_l(\boldsymbol{x}) = \sum_{t=1}^T \alpha_{t,l} h_{t,l}(\boldsymbol{x})$ $(l = 1, \cdots, L)$
```

For ranking loss, which concerns about whether the relevant labels are ordered before irrelevant ones, we can obtain $\boldsymbol{\beta}_{t,l}$ with the following optimization problem, which enforces the combined hypothesis to order the label pairs correctly:

$$\min_{\boldsymbol{\beta}_{t,l},\ \xi_{i,j}} \sum\nolimits_{i=1}^m \sum\nolimits_{j=1}^L D_{t,l}(i) \cdot \xi_{i,j} \tag{3}$$

$$\begin{aligned} s.t.\ \ & (Y_i(l) - Y_i(j))(R_{\boldsymbol{\beta}_{t,l}}(\boldsymbol{x}_i) - H_{t-1,j}(\boldsymbol{x}_i)) \geq 1 - \xi_{i,j} (\forall i, j) \\ & \boldsymbol{\beta}_{t,l}(\hat{h}_{t,l}) + \sum\nolimits_{H \in Q} \boldsymbol{\beta}_{t,l}(H) = 1 \\ & \boldsymbol{\beta}_{t,l}(\hat{h}_{t,l}),\ \boldsymbol{\beta}_{t,l}(H),\ \xi_{i,j} \geq 0\ (\forall H, i, j) \end{aligned}$$

In this paper, we focus on hamming loss, and implement MAHR according to Eq. 2.

We then look into the combination weight and the distribution update rule shown in line 7 and line 8, respectively. To optimize the hamming loss, we can equivalently optimize the zero-one loss on each label. The zero-one loss, however, is hard to optimize directly, and thus we try to optimize the logistic loss for each label instead:

$$\ell_{\ln}(H_l, \boldsymbol{x}_i, Y_i, l) = \ln(1 + \exp(-Y_i(l) \cdot H_l(\boldsymbol{x}_i))$$

which is a surrogate loss function for zero-one loss. By differentiating the logistic loss function, the combination weight and the update rule can be derived, as similarly done in FilterBoost [2]. One can use other surrogate loss functions such as the exponential loss. Also, as similar as FilterBoost, the distribution of MAHR in line 8 is a probability distribution, and thus normalization is not needed.

Note that the MAHR approach trains L models each for one label, these models may or may not have common base hypotheses, depending on the learning task. In contrast, other multi-label boosting approaches, such as MSSBoost, AdaBoost.MH and AdaBoost.MR, use the same base hypothesis for all the labels. This is a significant difference between MAHR and other approaches. The constraint that all labels share the same base hypothesis might be overly restrictive, and it may even injure the learning performance when some of the labels are not closely related.

For a test instance, we follow the training order and calculate the predictions of base hypotheses round-by-round until the final prediction is obtained. In such a way, the output of any hypothesis needs to be calculated only once, and there is no extra computational cost in the test phase comparing with multiple single-label boosting.

3.2 Estimating Label Relationship

Since MAHR reuses hypotheses across labels, the amount of reuse can be considered as a measure of label relationship. If two labels are independent, the hypothesis for one label may have a large error on the other label, and thus the reuse function for the other label will assign a very small weight to that hypothesis; if two labels are strongly related, the hypothesis for one label may have a small error on the other label, and thus the reuse function will assign a large weight to incorporate the hypothesis. Therefore, we assess the label relationship by the reuse weight, i.e., the $\boldsymbol{\beta}$ in Eq. 1. We define the *reuse score* from label j to i as

$$S(i, j) = \sum\nolimits_{t=2}^T \alpha_{t,i} \left(\boldsymbol{\beta}_{t,i}(H_{t-1,j}) - \boldsymbol{\beta}_{t,i}(-H_{t-1,j})\right), \tag{4}$$

where $\boldsymbol{\beta}_{t,i}(\cdot)$ is weighted by $\alpha_{t,i}$ (the weight of $h_{t,i}$). Here, t starts from 2 because there is no hypothesis for reuse in the first round. Note that $\boldsymbol{\beta}_{t,i}(\cdot)$ is constrained to be positive in the optimization, and $\boldsymbol{\beta}_{t,i}(-H_{\cdot,j})$, the weight of the negation of $H_{\cdot,j}$, actually reflects the degree of label j being negatively reused by label i, and therefore, we do subtraction in Eq. 4.

The reuse score $S(i, j)$ assesses how much label j can help the learning of label i. It is worth noting that, unlike the commonly used relationship, such as co-occurrence and ϕ-coefficient, the reuse score is not constrained to be symmetric, i.e., $S(i, j)$ does not necessarily equal $S(j, i)$.

3.3 Convergence Analysis

Suppose the multi-label learning task has L underlying hypotheses $(f_1, \cdots, f_L)$, each corresponds to one label. Given the training set and a hypothesis space $\mathcal{H}$, an algorithm tries to find a set of hypotheses, each approximating one of the underlying hypothesis. We concern about the training hamming loss $\widehat{err}_{hl} = \frac{1}{mL}\sum_{i=1}^m \sum_{l=1}^L I(sign(H_l(\boldsymbol{x}_i)) \neq Y_i(l))$ and the generalization hamming loss $err_{hl} = \mathbb{E}_{(\boldsymbol{x},Y)}[\frac{1}{L}\sum_{l=1}^L I(sign(H_l(\boldsymbol{x})) \neq Y(l))]$, where $I(\cdot)$ denotes the indicator function.

LEMMA 1. [2] *Let $\gamma = \min_t |\gamma_t|$, where γ_t is the edge of h_t, and let ϵ be the target error. If the number of boosting rounds $T > 2ln(2)/\epsilon/(1 - 2\sqrt{1/4 - \gamma^2})$, then $err_t < \epsilon$ for some $t, 1 \leq t \leq T$. Particularly it is true for $T > \frac{ln(2)}{2\epsilon\gamma^2}$.*

With Lemma 1, we get Theorem 1 which guarantees the convergence rate of MAHR on the training set.

THEOREM 1. *Let $\gamma = \min_{t,l} |\gamma_{t,l}|$, for any $\epsilon > 0$, MAHR achieves the hamming loss $\widehat{err}_{hl} < \epsilon$ on the training set within the number of rounds $T > 0.5 \ln 2 \cdot \epsilon^{-1} \cdot \gamma^{-2}$.*

PROOF. When MAHR achieves the zero-one loss ϵ on every label, it also achieves hamming loss ϵ because the hamming loss is the average of the zero-one loss over all labels. Therefore, we focus on the number of rounds for MAHR to achieve ϵ zero-one loss on a label; it is at most $0.5 \ln 2 \cdot \epsilon^{-1} \cdot \gamma^{-2}$ according to Lemma 1. □

Theorem 1 shows that MAHR efficiently achieves a training hamming loss. Then, based on Lemma 2, we get Theorem 2 for the generalization ability of MAHR.

LEMMA 2. [9] *Let H be a class of binary functions of VC-dimension $d \geq 2$. Then the VC-dimension of $\Theta_T(H)$ is at most $2(d+1)(T+1)\log_2(e(T+1))$ (where e is the base of the natural logarithm). Therefore, if the hypotheses generated by weak learner are chosen from a class of VC-dimension $d \geq 2$, then the final hypotheses after T iterations belong to a class of VC-dimension at most $2(d+1)(T+1)\log_2[e(T+1)]$.*

THEOREM 2. *Let $\gamma = \min_{t,l} |\gamma_{t,l}|$ and d be the VC-dimension of the hypothesis space $\mathcal{H}$. For any $\epsilon > 0$, with probability δ the generalization hamming loss of MAHR is bounded by*

$$err_{hl} \leq \epsilon + 2\sqrt{\frac{1}{m}(\ln\frac{9}{\delta} + D(\ln\frac{2m}{D} + 1))}$$

where $D = 2(d+1)(\frac{0.5\ln 2}{\epsilon \cdot \gamma^2} + 1)\log_2(e(\frac{0.5\ln 2}{\epsilon \cdot \gamma^2} + 1)) + L$.

PROOF. The generalization inequality is directly adapted from the Theorem 6.7 in [26]. The capacity D of the learning system depends on the complexity of the base hypothesis space and the number of hypotheses combined. The VC-dimension d of $\mathcal{H}$ and the total number of hypotheses $T \cdot L$ results in the capacity $2(d+1)(\frac{0.5\ln 2}{\epsilon \cdot \gamma^2}+1)\log_2(e(\frac{0.5\ln 2}{\epsilon \cdot \gamma^2}+1))$ according to Lemma 2. Moreover, the reuse of the other $L-1$ labels through linear combination results in the capacity L. Therefore, quantity of D in Theorem 2 is derived. □

Note that Theorems 1 and 2 are derived for a general case. We then analyze how the hypothesis reuse can be helpful. We characterize the ground-truth label relationship between two labels l_1 and l_2 by their output correlation

$$\kappa(l_1, l_2) = \frac{1}{m}\sum\nolimits_{i=1}^{m} f_{l_1}(\boldsymbol{x}) f_{l_2}(\boldsymbol{x}). \tag{5}$$

Then we have the generalization bound in Theorem 3, which shows that the hypothesis reuse in MAHR can utilize the label relationship to reduce the capacity of the learning system and thus leads to a better generalization ability.

THEOREM 3. *Let $\gamma = \min_{t,l} |\gamma_{t,l}|$, d be the VC-dimension of the hypothesis space $\mathcal{H}$, $\tilde{\kappa} = \min_{l_1,l_2} \kappa(l_1, l_2)$, and $\tilde{w}$, $\tilde{\epsilon}$ be the minimum weight and the maximum training error among all the cross-label hypotheses, respectively. For any $\epsilon > 0$, with probability δ the generalization hamming loss of MAHR is bounded by*

$$err_{hl} \leq \epsilon + 2\sqrt{\frac{1}{m}\left(\ln\frac{9}{\delta} + D\left(\ln\frac{2m}{D} + 1\right)\right)}$$

where $D = 2(d+1)\left(\frac{0.5\ln 2}{\epsilon' \cdot \gamma^2} + 1\right)\log_2\left(e\left(\frac{0.5\ln 2}{\epsilon' \cdot \gamma^2} + 1\right)\right) + L$ and $\epsilon' = 0.5 - \frac{1-2\epsilon}{2\tilde{w}} + \frac{(L-1)\tilde{w}(\tilde{\kappa}-2\tilde{\epsilon})}{2\tilde{w}}$.

PROOF. Suppose we have obtained the hypothesis h_l for the l-th label ($l = 2, \cdots, L$) such that

$$\widehat{err}(h_l \mid f_l) = \frac{1}{m}\sum\nolimits_{i=1}^{m} I[h_l(\boldsymbol{x}_i) \neq f_l(\boldsymbol{x}_i)] \leq \epsilon_l,$$

where $h_l(\boldsymbol{x}_i) \in \{+1, -1\}$. Thus, we have

$$\frac{1}{m}\sum\nolimits_{i=1}^{m} h_l(\boldsymbol{x}_i) f_l(\boldsymbol{x}_i) \geq 1 - 2\epsilon_l. \tag{6}$$

Then, we want to learn the first label up to the error ϵ_1. According to the reuse function, the hypothesis for the first label is $h_1 = sign[w_1\hat{h}_1 + \sum_{l=2}^{L} w_l h_l]$, and the following condition should be satisfied:

$$\frac{1}{m}\sum\nolimits_{i=1}^{m} f_1(\boldsymbol{x}_i) sign[w_1\hat{h}_1(\boldsymbol{x}_i) + \sum\nolimits_{l=2}^{L} w_l h_l(\boldsymbol{x}_i)] \geq 1 - 2\epsilon_1.$$

By the constraint of Eq. 2, it holds that $w_1\hat{h}_1 + \sum_{l=2}^{L} w_l h_l \in [-1, 1]$, and thus we can get a stronger requirement to guarantee the error:

$$\frac{1}{m}\sum\nolimits_{i=1}^{m} f_1(\boldsymbol{x}_i)(w_1\hat{h}_1(\boldsymbol{x}_i) + \sum\nolimits_{l=2}^{L} w_l h_l(\boldsymbol{x}_i)) \geq 1 - 2\epsilon_1$$
$$\Leftarrow \sum_{i=1}^{m} \frac{w_1}{m} f_1(\boldsymbol{x}_i)\hat{h}_1(\boldsymbol{x}_i) \geq 1 - 2\epsilon_1 - \sum_{l=2}^{L}\sum_{i=1}^{m} \frac{w_l}{m} f_1(\boldsymbol{x}_i) h_l(\boldsymbol{x}_i)$$
$$\Leftarrow \frac{1}{m}\sum_{i=1}^{m} f_1(\boldsymbol{x}_i)\hat{h}_1(\boldsymbol{x}_i) \geq \frac{1-2\epsilon_1}{w_1} - \sum_{l=2}^{L} \frac{w_l}{w_1}(\kappa(1, l) - 2\epsilon_l),$$

where the last derivation is obtained by applying Eqs. 5 and 6 together. Here, $A \Leftarrow B$ means that A can be guaranteed if B holds. The last inequality implies that we only need to learn the hypothesis $\hat{h}_1$ to approximate f_1 up to the error $\epsilon' = 0.5 - \frac{1-2\epsilon_1}{2w_1} + \sum_{l=2}^{L} \frac{w_l}{2w_1}(\kappa(1, l) - 2\epsilon_l)$, which can be larger than ϵ_1 when $\kappa(1, l)$ is large and ϵ_l is small. Incorporating the error into Theorem 2, we get Theorem 3. □

Theorem 3 suggests that the hypothesis reuse mechanism works when the labels are not independent and the number of rounds in MAHR is large, such that we have fit some labels well enough for reuse. Correspondingly, the number of rounds for achieving the error ϵ on one label is $0.5\ln 2\left(0.5 - \frac{1-2\epsilon_1}{2w_1} + \sum_{l=2}^{L} \frac{w_l}{2w_1}(\kappa(1, l) - 2\epsilon_l)\right)^{-1}\gamma^{-2}$. Also, from the proof of Theorem 3 we can find that with the related and well-learned hypotheses on other labels, the learning of the base hypothesis becomes easier. Note that if labels are independent, Theorem 3 is as same as Theorem 2.

3.4 Discussion on Improving Efficiency

MAHR requires more computational cost than other boosting approaches such as AdaBoost.MH, because in each round of boosting, the optimization of Eq. 2 needs to be addressed for each label. Fortunately, the optimization is a linear programming problem, and thus it can be solved efficiently. MAHR may train more base classifiers than other boosting approaches, however, this can be compensated in a parallel computing system. Unlike other boosting approaches that are relatively hard to be parallelized, in each round of MAHR, the line 5 to line 10 of Algorithm 1 can be executed for each label in parallel, because there is no interaction among labels in one round. Thus, the total execution time of the default implementation of MAHR can be close to that of AdaBoost.MH.

Moreover, since MAHR employs logistic loss, it can utilize the filtering mechanism of FilterBoost [2] to deal with large-scale data efficiently. We can have a filtering version of the MAHR algorithm: given the data set for training and evaluating the base hypothesis, it uses a *Filter* function to collect a set of training instances according to the number of training instances m, the target training error $\epsilon \in (0, 1)$, the confidence $\delta \in (0, 1)$ upper-bounding the probability of failure, and the error in edge estimation $\tau \in (0, 1)$.

4. EXPERIMENTS

We introduce the experimental settings in Section *4.1*, and then compare MAHR with state-of-the-art multi-label learning approaches in Section *4.2*. In Section *4.3*, we study the label relationship discovered by MAHR. We also study the influence of the number of boosting rounds in Section *4.4*, and the computational cost in Section *4.5*.

Table 1: Comparison of MAHR with state-of-the-art multi-label learning approaches on five evaluation criteria. ●(○) indicates that MAHR is significantly better (worse) than the corresponding method (pairwise *t*-tests at 95% significance level). Note that AdtBoost.MH could not output ranking loss values.

Data	Boosting-style methods						ML-*k*NN	RankSVM	ECC
	MAHR	MAHR-id	AdaBoost.MH	AdaBoost.MR	AdtBoost.MH	MSSBoost			
Hamming loss (the smaller, the better)									
Image	.169±.011	.190±.015●	.176±.007●	.754±.004●	.190±.007●	.210±.009●	.175±.007●	.339±.021●	.180±.010●
Image2	.191±.008	.229±.014●	.200±.008●	.752±.003●	.210±.006●	.230±.007●	.191±.006	.219±.021●	.205±.009●
Scene	.084±.004	.108±.019●	.091±.003●	.821±.001●	.111±.003●	.134±.004●	.090±.003●	.251±.017●	.095±.004●
Reuters	.030±.003	.046±.006●	.031±.003	.835±.002●	.055±.005●	.096±.008●	.049±.003●	.157±.031●	.062±.007●
Yahoo	.039±.012	.043±.014●	.046±.016●	.901±.052●	.044±.013●	.044±.014●	.043±.014●	.042±.014●	.049±.017●
Yeast	.204±.004	.222±.007●	.228±.004●	.697±.002●	.210±.003●	.227±.005●	.196±.003○	.196±.003○	.208±.005●
Enron	.047±.002	.053±.002●	.050±.002●	.932±.009●	.054±.002●	.051±.002●	.051±.002●	.311±.367●	.055±.002●
One error (the smaller, the better)									
Image	.301±.024	.355±.046●	.313±.021●	.307±.020●	.363±.021●	.375±.023●	.325±.024●	.708±.052●	.300±.022
Image2	.368±.019	.493±.063●	.378±.022	.378±.018●	.436±.019●	.427±.018●	.370±.017	.400±.063●	.350±.019○
Scene	.217±.011	.265±.060●	.226±.010●	.215±.012	.293±.012●	.291±.009●	.238±.012●	.457±.065●	.232±.011●
Reuters	.063±.010	.087±.015●	.055±.008○	.059±.008○	.120±.017●	.280±.035●	.126±.012●	.228±.192●	.156±.020●
Yahoo	.398±.122	.506±.188●	.452±.141●	.487±.118●	.483±.148●	.481±.149●	.471±.157●	.412±.130	.391±.133
Yeast	.243±.011	.253±.010●	.291±.012●	.319±.015●	.247±.010	.272±.010●	.235±.012○	.224±.009○	.180±.012○
Enron	.234±.030	.309±.035●	.293±.039●	.362±.035●	.317±.036●	.273±.033●	.299±.031●	.855±.020●	.228±.036
Coverage (the smaller, the better)									
Image	.921±.060	1.06±.122●	.922±.058	.930±.062	1.04±.062●	1.06±.046●	.972±.062●	2.10±.065●	.998±.056●
Image2	1.05±.055	1.49±.180●	1.06±.051	1.09±.048●	1.22±.050●	1.20±.046●	1.08±.048●	1.17±.163●	1.13±.055●
Scene	.471±.032	.555±.132●	.463±.022	.450±.020○	.633±.040●	.582±.028●	.505±.029●	1.15±.195●	.570±.026●
Reuters	.276±.027	.316±.040●	.268±.026	.270±.025	.409±.047●	.891±.105●	.440±.035●	.801±.780●	.664±.083●
Yahoo	4.56±1.60	7.74±2.96●	4.22±1.29○	4.54±1.27	4.27±1.36	4.14±1.32○	4.09±1.23○	4.60±1.47	8.52±1.79●
Yeast	6.70±.092	7.30±.309●	6.66±.087○	7.11±.083●	6.50±.074○	6.71±.097	6.28±.086○	6.62±.134○	7.16±.127●
Enron	14.2±1.07	20.2±1.22●	11.5±.772○	13.5±.865○	14.0±.813	12.5±.898○	13.0±.850○	26.5±1.32●	20.7±1.09●
Ranking loss (the smaller, the better)									
Image	.166±.012	.198±.029●	.164±.011	.171±.012●	N/A	.199±.010●	.177±.013●	.463±.018●	.247±.016●
Image2	.198±.012	.303±.047●	.200±.011	.213±.011●	N/A	.233±.011●	.203±.010●	.225±.041●	.282±.021●
Scene	.077±.006	.093±.026●	.075±.004	.075±.004○	N/A	.099±.005●	.083±.006●	.214±.039●	.139±.008●
Reuters	.019±.003	.025±.005●	.017±.003○	.018±.003	N/A	.119±.018●	.045±.004●	.104±.128●	.105±.014●
Yahoo	.109±.046	.206±.100●	.103±.040	.116±.040	N/A	.103±.042	.102±.045	.112±.047	.332±.084●
Yeast	.184±.005	.209±.014●	.194±.005●	.232±.005●	N/A	.195±.006●	.168±.006○	.172±.006○	.279±.011●
Enron	.098±.010	.145±.011●	.076±.008○	.100±.008	N/A	.085±.009○	.091±.008○	.267±.019●	.246±.018●
Average Precision (the larger, the better)									
Image	.804±.014	.766±.029●	.798±.013●	.796±.013●	.766±.013●	.758±.013●	.788±.014●	.516±.011●	.789±.014●
Image2	.764±.012	.669±.043●	.759±.012	.751±.011●	.718±.012●	.723±.011●	.759±.011●	.739±.041●	.753±.013●
Scene	.869±.008	.841±.038●	.866±.006●	.869±.007	.822±.008●	.828±.006●	.857±.007●	.698±.047●	.846±.007●
Reuters	.962±.005	.948±.009●	.966±.005○	.964±.004○	.925±.011●	.815±.022●	.920±.007●	.846±.145●	.884±.013●
Yahoo	.660±.098	.543±.159●	.634±.109●	.595±.091●	.617±.113●	.618±.112●	.625±.117●	.658±.103	.616±.092●
Yeast	.749±.007	.717±.015●	.727±.008●	.693±.007●	.740±.006●	.729±.007●	.762±.007○	.767±.007○	.731±.007●
Enron	.678±.020	.601±.019●	.681±.021	.596±.021●	.596±.027●	.661±.018●	.636±.015●	.262±.017●	.637±.021●

4.1 Settings

To examine the effectiveness of hypothesis reuse, we compare MAHR with a degenerated version MAHR-id, which is as same as MAHR except that it does not reuse hypothesis (i.e., it does not invoke the reuse function) and learns every label independently. We also compare MAHR with state-of-the-art multi-label learning approaches that are divided into two groups: the boosting group including AdaBoost.MH [20], AdaBoost.MR [20], AdtBoost.MH [6] and MSSBoost [27], and the non-boosting group including ML-*k*NN [29] which considers the first-order correlation, RankSVM [8] which considers the second-order correlation, and ECC [18] which considers higher-order correlation. For the compared approaches, the best parameters reported in their corresponding literatures [20, 6, 27, 29, 8, 18] are used. For MAHR, the base learning algorithm is *decision stump* implemented in Weka [11], the number of rounds, i.e., the parameter T in Algorithm 1 is set as default to $2\times\#feature$ for all the data sets. Note that the influence of the number of learning rounds will be studied in Section 4.4.

There are different criteria for evaluating the performances of multi-label learning. In our experiments, we employ five commonly used criteria, i.e., *hamming loss*, *one error*, *coverage*, *ranking loss* and *average precision*. The definition of these criteria can be found in [21, 7, 30].

Seventeen data sets are used in our experiments. These data sets spanned a broad range of applications: *Image* [29], *Image2* [30] and *Scene* [1] for image classification, *Reuters* [22] for text categorization, *Yeast* [8] for gene function prediction, *Enron* [15] for email analysis, and *Yahoo* [25] for web page categorization. Note that *Yahoo* consists of 11 independent data sets, i.e., *Arts*, *Business*, *Computers*, *Education*, *Entertainment*, *Health*, *Recreation*, *Reference*, *Science*, *Social* and *Society*. For *Yahoo*, the data sets are pre-separated into training and test sets [25]; for the other data sets, we randomly select 1,500 instances for training and use the remaining data for testing. The data partition is repeated randomly for thirty times, and the average results as well as standard deviations over the thirty repetitions are recorded.

4.2 Comparison Results

The comparison results are shown in Table 1. Due to page limit, for *Yahoo* we only report the average results over the eleven data sets. Note that we are mostly interested in *hamming loss* because the current implementation of MAHR is designed for this loss.

Comparing with MAHR-id, MAHR achieves significantly better performance over all the five criteria on all the data sets. This validates the usefulness of the hypothesis reuse mechanism on exploiting label relationship.

Comparing with other approaches, MAHR achieves the best performance on hamming loss in most cases; particularly, it outperforms AdaBoost.MH and AdtBoost.MH that were also designed for optimizing hamming loss. For the other four criteria, MAHR also achieves excellent performances although it was not implemented to optimize these criteria. It is superior to the compared approaches on both one error and average precision in most cases. On coverage, it performs better than ECC, comparable with other approaches, worse than only AdaBoost.MH. On ranking loss, it outperforms ECC and achieves comparable performances with other approaches.

Overall, by exploiting label relationship with the hypothesis reuse mechanism, MAHR drastically improves the performance of its non-reuse counterpart MAHR-id over all criteria. Comparing with other approaches, MAHR is the best approach for optimizing hamming loss, and is highly competitive over the other criteria. These observations suggest that label relationship does help multi-label learning, and the hypothesis reuse mechanism is useful. Note that, currently we just implement R in a simple form of linear combination, and it is expected that better performance can be obtained with better designs of reuse functions. Also, one can incorporate different multi-label learning mechanisms or optimize different objectives in Eq. 2 when other evaluation criteria rather than hamming loss are concerned.

4.3 Label Relationship Discovered

To examine whether the reuse score reflects reasonable label relationship, since we were not given with ground-truth label relationship for real-world data, we first study three synthetic data sets, for which we know the ground-truth label relationship. All the three data sets have five labels, L_1 to L_5, and the 5-th label is assigned to an instance if it has none of L_1 to L_4. In *data-inde*, the first four labels are independent to each other, each label is associated with five different features, and a label is assigned to an instance if the summed value of the corresponding five features is larger than a specified threshold. In *data-equal*, $L_1 = L_2$ and $L_3 = L_4$. In *data-union*, the labels L_2 to L_4 are independent to each other, and L_1 is assigned to an instance if it has at least one of L_2 to L_4, i.e., $L_1 = L_2 \vee L_3 \vee L_4$. We generate 10,000 instances, and randomly divide them into two parts of equal size, using one part for training and the other for testing. The experiments are repeated for 10 times, each with a random partition of the data set, and the average reuse scores are recorded in Tables 2 to 4 for the three data sets, respectively. Note that the values are normalized such that each label has a reuse score 1 for itself.

Table 2: Reuse score on *data-inde* (L_1 to L_4 are independent)

Target label	Reuse label				
	L_1	L_2	L_3	L_4	L_5
L_1	1.00	-0.00	-0.00	0.00	-0.13
L_2	-0.00	1.00	0.00	-0.01	0.02
L_3	0.00	0.00	1.00	-0.01	0.02
L_4	0.01	-0.00	-0.00	1.00	-0.14
L_5	-0.63	-0.33	-0.32	-0.88	1.00

Table 3: Reuse score on *data-equal* ($L_1 = L_2$, $L_3 = L_4$)

Target label	Reuse label				
	L_1	L_2	L_3	L_4	L_5
L_1	1.00	0.25	0.00	-0.00	0.04
L_2	0.28	1.00	-0.00	-0.00	0.05
L_3	0.00	-0.01	1.00	0.20	0.00
L_4	0.00	-0.00	0.19	1.00	0.00
L_5	-0.25	-0.28	-0.24	-0.25	1.00

Table 4: Reuse score on *data-union* ($L_1 = L_2 \vee L_3 \vee L_4$)

Target label	Reuse label				
	L_1	L_2	L_3	L_4	L_5
L_1	1.00	0.45	0.43	0.51	-0.03
L_2	-0.00	1.00	0.00	-0.01	-0.01
L_3	0.00	0.00	1.00	-0.02	-0.01
L_4	0.10	-0.01	-0.01	1.00	-0.09
L_5	-0.03	-0.44	-0.47	-0.58	1.00

It is obvious that all the diagonal entries of the tables are larger than the other entries, implying that each boosted learner mainly relies on its own task. It can be found that in the last row of all three tables, when learning L_5, it takes the other labels as negative references. This is in our expectation because L_5 is assigned to an instance when no other label is assigned. Excluding the diagonal entries, in Table 2, the entries in the first four rows of the first four columns are close to zero, indicating that L_1 to L_4 are not helpful to each other; this is consistent with the ground-truth that these labels are independent. In Table 3 the entries $S(L_1, L_2), S(L_2, L_1), S(L_3, L_4)$ and $S(L_4, L_3)$ show relatively large positive values, and in Table 4 the entries $S(L_1, L_2), S(L_1, L_3)$ and $S(L_1, L_4)$ show large positive values; these results are consistent with the ground-truth label relationship. Particularly, in Table 4 the sum of $S(L_1, L_2)$, $S(L_1, L_3)$ and $S(L_1, L_4)$ exceeds the entry $S(L_1, L_1)$, implying the great impact of labels L_2 to L_4 on L_1.

As shown in Tables 2 to 4, unlike co-occurrence or ϕ-coefficient, the reuse score is asymmetric. In *data-union*, since we set L_1 to appear if any of L_2, L_3 and L_4 appears, L_2 to L_4 are helpful for determining L_1; however, the inverse is not true. Table 4 shows that the reuse scores clearly reflect this asymmetric relationship. In real-world tasks, label relationship is usually asymmetric, for example, the label "lake" implies the label "water", but the inverse may not be true. Therefore, asymmetric relationship is more consistent with realistic situations.

We also examine the reuse scores on two real data sets, i.e., *Image* and *Enron*. *Image* contains five labels: *desert*, *mountains*, *sea*, *sunset* and *trees*. Figure 1 shows some example images, each column for one label. The reuse scores are shown in Table 5. Again, all the entries at the diagonal are large positive values. An interesting observation is that most elements are negative, suggesting that the mutually

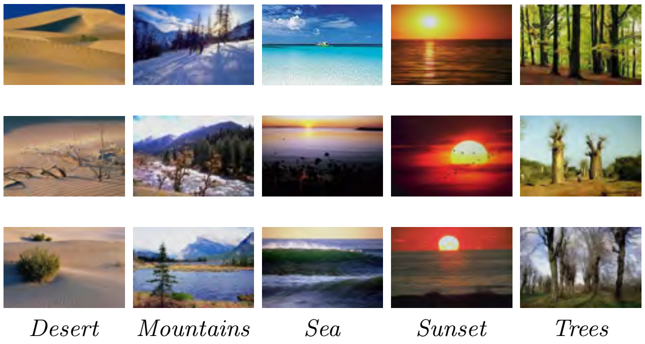


Figure 1: Example images of *Image* data set

Table 5: Reuse score on *Image* data set

Target label	Reuse label				
	desert	*mount.*	*sea*	*sunset*	*trees*
desert	1.00	-0.08	-0.05	-0.11	-0.03
mountains	-0.06	1.00	-0.10	-0.07	-0.06
sea	-0.16	-0.10	1.00	0.01	-0.14
sunset	-0.04	-0.07	-0.00	1.00	-0.06
trees	-0.02	0.06	-0.09	-0.11	1.00

exclusive relationship among labels is important when dealing with multi-label image classification tasks, and is worth paying more attention. Besides the diagonal entries, there are two positive entries. The entry $S(trees, mountains)$ is relatively large, which can be understand by the fact that when there are *mountains* in an image, *trees* are likely to appear, as can be recognized from Figure 1. On the contrary, the entry $S(mountains, trees)$ is negative, because *trees* do not imply *mountains*, which again shows asymmetric property of label relationship. Similarly, the entry $S(sea, sunset)$ is positive, which may be understand by the fact that the *sunset* photos are often taken at *sea*; meanwhile, *sea* does not imply *sunset*, as can be recognized from Figure 1. It is noticed that strong negative entries imply relationship as well. For example, when *desert* occurs in an image, it is unlikely to find *sea* in the image; this explains the strong negative entry $S(sea, desert)$.

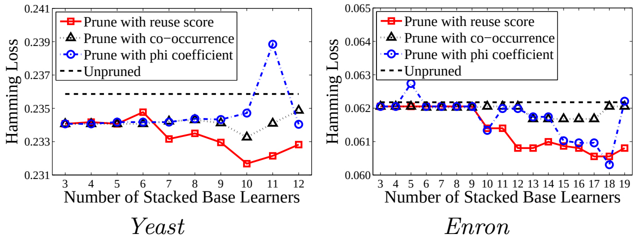
Enron is a collection of email messages written by the employees of Enron. Different from the results on *Image* data, there are more positive reuse scores than negative ones in *Enron*, and the absolute values of the negative scores are relatively small. So, we focus on the positive relationship. Since there are too many labels, we only show the results on a part of example labels. Table 6 presents the three most positively related labels for each example label. It can be seen that the discovered relationship is reasonable. For example, an email about *company business*, *newsletters* and *secrecy* is usually sent to *alliance* members or *partners*; the emotion *pride* is likely to come through an email about *jubilation*, *contributions* and *admiration*; etc. It is also worth noting that MAHR discovers high-order relationship among labels; for example, with different accompanying labels, the label *talking points* can be positively related with *admiration* as well as *sadness, despair*.

While the results on the two real data sets provide an intuitive evaluation of the reuse score, we can further have a quantitative evaluation. The 2BR approach [24] trains multiple classifiers, each for one label, and then prunes some classifiers and uses the remaining ones for a second-level stacking classifier. The pruning is based on the label relationship. Therefore, we can incorporate different label relationships into 2BR, and then evaluating the performances. Here, three kinds of relationships are compared, i.e., the reuse score returned by MAHR, the co-occurrence relationship, and the ϕ-coefficient relationship which is used in [24]. As in [24], we report the results on *Yeast* and *Enron* which have many labels for pruning. Decision tree implemented in Weka is used for the base-level and the stacking classifiers. Figure 2 shows the hamming loss with varied number of stacked classifiers. According to [24], the number of stacked classifiers increases from 3 to 12 for *Yeast* and from 3 to 19 for *Enron*.

First, we observe that pruning based on any of the three relationships improves the performance of stacking all the classifiers (corresponding to the dashed line in Figure 2). Comparing with the baselines, MAHR lead to the best per-

Table 6: Top 3 positively related labels for some example labels in *Enron* data set

Label	Most related labels
alliances, partnerships	1. *company business, strategy* 2. *newsletters* 3. *secrecy, confidentiality*
gratitude	1. *friendship, affection* 2. *alliances, partnerships* 3. *forwarded email(s) including replies*
legal advice	1. *alliances, partnerships* 2. *government action(s)* 3. *secrecy, confidentiality*
admiration	1. *jubilation* 2. *talking points* 3. *pride*
jokes, humor (related to business)	1. *humor* 2. *sarcasm* 3. *talking points*
pride	1. *jubilation* 2. *political influence, contribution* 3. *admiration*
company image changing, influence	1. *company business, strategy* 2. *talking points* 3. *worry, anxiety*
meeting minutes	1. *regulations and regulators* 2. *company business, strategy* 3. *talking points*
secrecy, confidentiality	1. *worry, anxiety* 2. *talking points* 3. *dislike, scorn*
sadness, despair	1. *secrecy, confidentiality* 2. *political influence, contribution* 3. *talking points*



Yeast *Enron*

Figure 2: Comparison of three kinds of label relationships with the 2BR approach

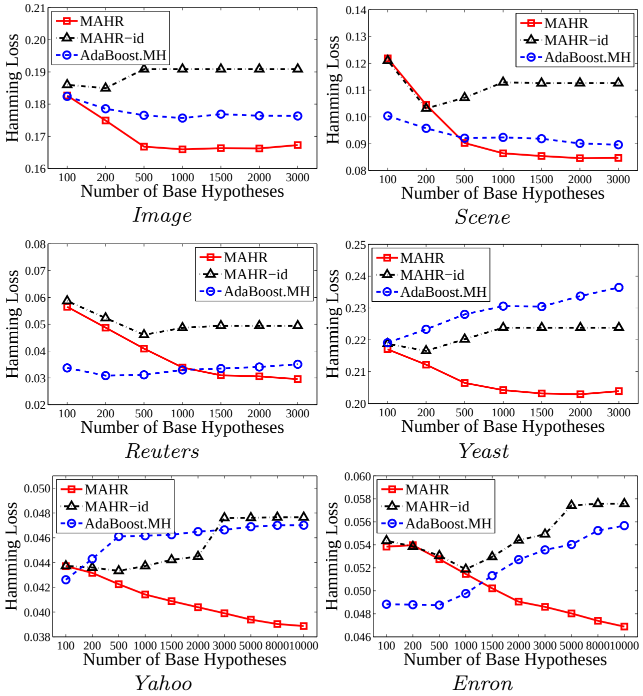


Figure 3: Hamming loss of MAHR, MAHR-id and AdaBoost.MH under varying number of base hypotheses

formance, particularly when the number of stacked classifiers is not too small. This observation further validates the advantage of MAHR on exploiting label relationship.

4.4 Influence of Learning Rounds

An important parameter of boosting algorithms is the number of learning rounds, or in other words, the number of base hypotheses. According to the analysis in [9], to achieve a good performance, the classifier should have low training error and a small number of base hypotheses, whereas a large number of base hypotheses will make the classifier overly complex and may lead to overfitting.

Considering that MAHR and MAHR-id train L (label size) times more hypotheses than the other boosting approaches in each round, to be fair, we compare their performances with the same number of hypotheses. We evaluated three boosting approaches, i.e., MAHR, MAHR-id and AdaBoost.MH, all optimizing hamming loss, up to involving 3,000 base hypotheses for *Image*, *Scene*, *Reuters* and *Yeast*, and 10,000 base hypotheses for *Yahoo* and *Enron*. The hamming loss curves are plotted in Figure 3. (figure for *Image2* is similar with that of *Image*, and thus is not presented due to page limit).

As shown in Figure 3, MAHR significantly improves the performance of MAHR-id with various number of boosting rounds. Contrasting with AdaBoost.MH, MAHR can achieve a better performance after sufficient boosting rounds. Moreover, while MAHR-id and AdaBoost.MH seem suffering from overfitting after a number of rounds, MAHR keeps on improving performance in most cases. We attribute this phenomenon to the fact that MAHR considers high-order relationship among labels, as illustrated in the previous subsection, and thus it can tolerant more complex models.

Table 7: Average CPU time (in seconds) of boosting-style approaches

Methods	*image*	*scene*	*reuters*	*yahoo*	*yeast*	*enron*
MAHR	500	360	255	441	106	488
MAHR-id	50	44	32	23	15	11
AdaBoost.MH	99	127	21	308	92	446
AdaBoost.MR	15	18	10	118	14	188
AdtBoost.MH	1,153	1,578	441	3,036	1,039	2,923
MSSBoost	110	165	226	8,304	952	12,486

4.5 Time Cost

We study the time cost of multi-label boosting approaches. All the experiments are performed with MATLAB 7.6 on a machine with 8 × 2.60 GHz CPUs and 16GB main memory. The average time cost of each round on one label for MAHR-id and MAHR are 0.73 and 0.91 seconds, respectively, confirming that the hypothesis reuse procedure is efficient. The average CPU time (in seconds) of each boosting approach on each data set is shown in Table 7. It can be seen that AdtBoost.MH is relatively time consuming, and MSSBoost costs more time on some data sets because it employs SVM as base classifier. Benefitting from parallel computing, although MAHR performs more boosting rounds, its computational cost is comparable with other boosting approaches.

5. CONCLUSION

In this paper, we propose a novel multi-label learning approach, MAHR, which does not require the input of label relationship but it is able to discover reasonable label relationship automatically, in addition to achieving a high predictive performance. The key of MAHR is a hypothesis reuse mechanism which offers an effective way to discover and exploit label relationship. This mechanism is implemented as a boosting style approach, where multiple boosted learners are trained, each for one label; each learner not only looks into its own single-label task, but also reuses the trained hypotheses from other labels.

Experiments validate that MAHR is superior or highly competitive to state-of-the-art multi-label approaches, and it drastically improves the performance of MAHR-id, which learns single-label tasks independently. Moreover, experiments show that MAHR discovers reasonable and useful label relationship, and suffers little from overfitting even with a large number of base hypotheses. It is worth noting that our study disclose that the label relationship is usually asymmetric, quite different from symmetric label relationship assumed by previous studies.

It is not difficult to develop variant algorithms of MAHR to optimize other multi-label losses, such as the ranking loss, and it is also possible to reuse hypotheses in forms other than weighted combination in the future.

6. REFERENCES

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[2] J. K. Bradley and R. E. Schapire. Filterboost: regression and classification on large datasets. In *Advances in Neural Information Processing Systems 20*, pages 185–192. MIT Press, Cambridge, MA, 2008.

[3] K. Brinker and E. Hüllermeier. Case-based multilabel ranking. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 702–707, Hyderabad, India, 2007.

[4] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 78–87, Washington, DC, 2004.

[5] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: Combining Bayes with SVM. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 177–184, Pittsburgh, PA, 2006.

[6] F. D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label altenating decision tree from texts and data. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 35–49, Leipzig, Germany, 2003.

[7] A. de Carvalho and A. Freitas. A tutorial on multi-label classification techniques. In A. Abraham, A.-E. Hassanien, and V. Snael, editors, *Foundations of Computational Intelligence, Vol.5*, pages 177–195. Springer, 2009.

[8] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, Cambridge, MA, 2002.

[9] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[10] N. Ghamrawi and A. Mccallum. Collective multilabel classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 195–200, Bremen, Germany, 2005.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations*, 11(1):10–18, 2009.

[12] B. Hariharan, L. Zelnik-Manor, S. V. N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning*, pages 423–430, Haifa, Israel, 2010.

[13] S.-J. Huang and Z.-H. Zhou. Multi-label learning by exploiting label correlations locally. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Canada, 2012.

[14] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 381–389, Las Vegas, NV, 2008.

[15] B. Klimt and Y. Yang. Introducing the enron corpus. In *Proceedinds of the 1st Conference on Email and Anti-Spam*, Mountain View, CA, 2004.

[16] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *Working Notes of the AAAI'99 Workshop on Text Learning*, Orlando, FL, 1999.

[17] J. Petterson and T. Caetano. Submodular multi-label learning. In *Advances in Neural Information Processing Systems 24*. MIT Press, Cambridge, MA, 2011.

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

[19] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classifcation models. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 774–751, Bonn, Germany, 2005.

[20] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated prediction. *Machine Learning*, 37(3):297–336, 1999.

[21] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[22] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[23] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 668–676, Las Vegas, NV, 2008.

[24] G. Tsoumakas, A. Dimou, E. Spyromitros, and V. Mezaris. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, pages 101–116, Bled, Slovenia, 2009.

[25] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, pages 721–728. MIT Press, Cambridge, MA, 2003.

[26] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New Yorks, NY, 1982.

[27] R. Yan, J. Tešić, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 834–843, San Jose, CA, 2007.

[28] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 999–1007, Washington, DC, 2010.

[29] M.-L. Zhang and Z.-H. Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[30] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.