Bayes-Optimal Hierarchical Multilabel Classification

Wei Bi and Jame T. Kwok

Abstract—Hierarchical multilabel classification allows a sample to belong to multiple class labels residing on a hierarchy, which can be a tree or directed acyclic graph (DAG). However, popular hierarchical loss functions, such as the H-loss, can only be defined on tree hierarchies (but not on DAGs), and may also under- or over-penalize misclassifications near the bottom of the hierarchy. Besides, it has been relatively unexplored on how to make use of the loss functions in hierarchical multilabel classification. To overcome these deficiencies, we first propose hierarchical extensions of the Hamming loss and ranking loss which take the mistake at every node of the label hierarchy into consideration. Then, we first train a general learning model, which is independent of the loss function. Next, using Bayesian decision theory, we develop Bayes-optimal predictions that minimize the corresponding risks with the trained model. Computationally, instead of requiring an exhaustive summation and search for the optimal multilabel, the resultant optimization problem can be efficiently solved by a greedy algorithm. Experimental results on a number of real-world data sets show that the proposed Bayes-optimal classifier outperforms state-of-the-art methods.

Index Terms—Hierarchical classification, multilabel classification, loss function, Bayesian decision theory

1 Introduction

In multilabel classification, a sample can be associated with multiple class labels. This is different from binary or multiclass classification in which only one label can be assigned to each sample. In recent years, there has been a lot of interest on multilabel classification in diverse application domains. For example, a document can belong to more than one topic categories in text categorization [1], [2]; a gene may be assigned with more than one functions in bioinformatics [3]; and an image can be associated with multiple semantic classes in image annotation [4]. Recent surveys on the mining of multilabel data can be found in [5], [6], [7].

Often, the labels are organized into hierarchies, which can be a tree (e.g., in text classification and bioinformatics [3], [8], [9]), or a directed acyclic graph (DAG) (as in the gene ontology (GO) [10]). The hierarchy is usually constructed by domain experts, but can also be learned automatically from the data using procedures such as hierarchical clustering [11] and Bayesian network structure learning [12]. It is well-known that the label hierarchy information is beneficial for prediction [3], [8], [9], [13]. For example, if we predict an article as belonging to entertainment, it is more likely that this also belongs to music (a subcategory of entertainment in the hierarchy) than to politics. Algorithms that make use of the label hierarchy are called hierarchical classification algorithms; while those ignoring the hierarchy are called flat classification algorithms.

Hierarchical classification can be further categorized as hierarchical multiclass classification, in which a feasible multilabel consists of only a single branch as positive labels; and hierarchical multilabel classification, the feasible multilabel of which allows multiple branches and partial paths as positive labels. A number of techniques have been developed for hierarchical multiclass classification [14], [15], [16]. Hierarchical multilabel classification, on the other hand, may have a large number of feasible multilabels even for a label hierarchy with a small size, thus the problem is more complicated [8], [9]. Moreover, most of the works on hierarchical multilabel classification can only deal with the simpler tree-structured hierarchies [8], [9], [17], [18]. Recently, more powerful algorithms that can be used on both tree- and DAG-structured label hierarchies have also been proposed [3], [13], [19].

In any classification problem, the loss function is of central importance [20]. For binary or multiclass classification, the zero-one loss is the most popular. It simply checks if the prediction is identical to the ground-truth. However, in multilabel classification, the predicted multilabel on a sample may partially overlap with its set of ground-truth labels. To assign different penalties to different degrees of partial correctness, loss functions such as the Hamming loss, top-k precision and ranking loss have been proposed for flat classification [21], [22], [23], [24]. In hierarchical multilabel classification, the H-loss [8], which takes the hierarchy into account, has been commonly used. Yet, it can only be defined on label trees (but not DAGs), and may under-penalize misclassifications near the bottom of the hierarchy. On the other hand, other hierarchical loss functions, such as the matching loss [25], may penalize these misclassifications too heavily.

Once a loss function is chosen, it should be properly used in the classification algorithm [20], [26]. A classification algorithm usually operates in two phases—training and prediction. The loss function can be incorporated into either of the phases. For example, the structured SVM [26] uses the loss function to re-scale the slack variable or margin in training. Generally, it needs to solve different optimization algorithms if different loss functions are used. Meanwhile,

• *The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: {weibi, jamesk}@cse.ust.hk.*

Manuscript received 3 June 2014; revised 9 Mar. 2015; accepted 19 Mar. 2015. Date of publication 3 June 2015; date of current version 2 Oct. 2015. Recommended for acceptance by I. Davidson. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2015.2441707

the Bayes-optimal classifier exploits the loss functions in the prediction phase, which derives the optimal decision rule with minimum risk over all feasible outputs. A good property of the Bayes-optimal classifier is that the training algorithm is independent of the loss function, as long as it can obtain the prediction distribution of the outputs. Thus, it is more convenient to be used when multiple loss functions are available. Also, the flexible design of the training algorithm for the Bayes-optimal classifier makes it achieve better training efficiency compared with the structured SVM, which is often not scalable with a large output space [27]. Thus, we focus on the Bayes-optimal classifier in this work. In flat multilabel classification, Bayes-optimal decision rules corresponding to a number of loss functions have been derived [23], [28], [29]. However, this is still relatively unexplored for hierarchical multilabel classification, as the risks over all its feasible multilabels are more difficult to evaluate and minimize. The only exception is the B-SVM [30], which is Bayes-optimal for the H-loss.

In this paper, we revisit the Hamming loss and ranking loss in flat classification, and extend them for hierarchical multilabel classification by incorporating misclassification weights based on the label hierarchy. We show that these hierarchical extensions can avoid the limitations of existing hierarchical multilabel loss functions. Then, based on Bayesian decision theory, we compute the Bayes-optimal prediction rules with respect to different loss functions by minimizing their (conditional) risks. On both tree- and DAG-structured label hierarchies, the training algorithm only needs to estimate the conditional probability for each node to compute such risks. Moreover, all the risks can be minimized by simple greedy algorithms, without the need for summing and searching over a potentially exponential number of feasible multilabels.

The rest of this paper is organized as follows. Section 2 first reviews the related work , particularly on the loss functions used in flat/hierarchical multiclass/multilabel classification. Section 3 revisits the Hamming loss and ranking loss commonly used in flat classification, and extends them for use in hierarchical multilabel classification. Section 4 shows how the risks can be efficiently computed and minimized. Experimental results are presented in Section 5. In the last section, we gives some concluding remarks. All the proofs are in the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2441707.

Preliminary results of this paper have been reported in a shorter version [31]. Besides providing a more thorough literature review, this paper considers the matching loss, hierarchical Hamming loss and hierarchical ranking loss; while the preliminary version [31] only discussed the hierarchical Hamming loss. In particular, we will show that for all these loss functions, their Bayes-optimal classifiers can be cast under the same problem formulation by setting with different weighting parameters. Also, more experimental evidence is provided to demonstrate the merits of the proposed algorithm.

Notations. In the following, $\mathcal{H}$ denotes the label hierarchy. Its nodes are indexed as 0 (for the root), $1, 2, \ldots, N-1$, where N is the number of nodes in $\mathcal{H}$. For a node i, we use $\mathrm{pa}(i)$ to denote its (unique) parent when $\mathcal{H}$ is a tree, $\mathrm{Pa}(i)$ for the set of its parent(s) when $\mathcal{H}$ is a DAG, and $\mathrm{anc}(i)$ for its set of ancestors. $(\mathbf{x}, \mathbf{y})$ denotes the sample, where $\mathbf{x}$ is the input and $\mathbf{y}$ is the multilabel $[y_0, \ldots, y_{N-1}]^T \in \{0,1\}^N$ denoting the memberships of $\mathbf{x}$ to each of the nodes (labels) in $\mathcal{H}$. Moreover, for a given set S and a label vector $\mathbf{y}$, $\mathbf{y}_S$ denotes the subvector of $\mathbf{y}$ with indices from S.

2 Related Work

2.1 Hierarchy Constraints in Hierarchical Classification

In hierarchical classification, if the label hierarchy is a tree, a node i (excluding the root) can be labeled positive only if its parent is also labeled positive, i.e.,

$$y_i = 1 \Rightarrow y_{\mathrm{pa}(i)} = 1. \tag{1}$$

For DAG-structured label hierarchies, there are two interpretations of its hierarchy constraint [3], [13]. One is the AND-interpretation, which means that a node can be labeled positive only if all its parents are positive. The other is the OR-interpretation, which means a node can be labeled positive if at least one of its parents is positive. In this paper, we adopt the AND-interpretation which is more common. Thus, for each node i (again excluding the root),

$$y_i = 1 \Rightarrow \mathbf{y}_{\mathrm{Pa}(i)} = \mathbf{1}. \tag{2}$$

2.2 Loss Functions in Flat and Hierarchical Classification

2.2.1 Flat Classification

In this section, we first review some loss functions that have been popularly used in flat multilabel classification.

1) Zero-one loss: Let $\hat{\mathbf{y}}$ be the predicted label vector for sample $\mathbf{x}$. The zero-one loss is defined as

$$\ell_{0/1}(\hat{\mathbf{y}}, \mathbf{y}) = I(\hat{\mathbf{y}} \neq \mathbf{y}), \tag{3}$$

where $I(\cdot)$ is the indicator function that returns 1 when the argument holds, and 0 otherwise. It has been popularly used for both multiclass classification [15] and multilabel classification [23], [32]. However, the zero-one loss cannot distinguish near-misses (where $\hat{\mathbf{y}}$ is very similar to $\mathbf{y}$) from completely incorrect predictions.

2) Hamming loss (or symmetric loss) [1], [21], [33]:

$$\ell_{\mathrm{hamming}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i \in \mathcal{H}} I(\hat{y}_i \neq y_i). \tag{4}$$

It is commonly used for multilabel classification, and more informative than the zero-one loss.

3) Top-k precision: In some applications, one is only interested in the k most-confident predicted (positive) labels for each sample [22]. This can be measured by the top-k precision, which is defined as $\frac{1}{k} \times (\#\text{true positives in the top-}k\text{ labels of }\hat{\mathbf{y}})$. The corresponding loss is then

$$\ell_{\text{top-}k}(\hat{\mathbf{y}}, \mathbf{y}) = 1 - \text{top-}k\text{-precision}(\hat{\mathbf{y}}, \mathbf{y}). \tag{5}$$

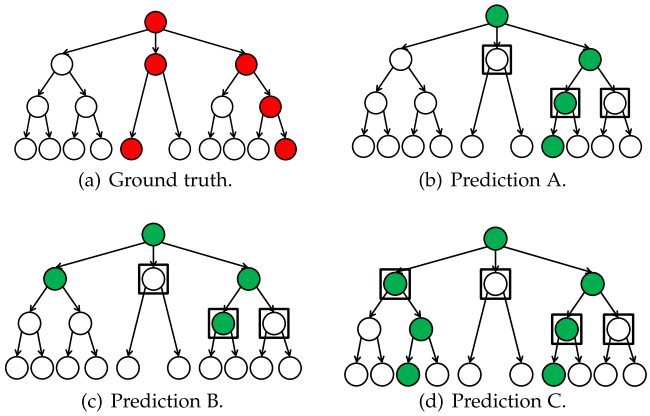


Fig. 1. An example illustrating the deficiencies of various loss functions. Here, colored nodes are the positive nodes in the given multilabel. Misclassified nodes that contribute to the H-loss are marked in squares. Please refer to the text for details.

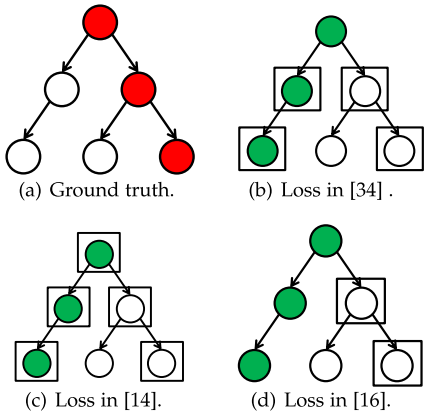


Fig. 2. Example illustrating some popular loss functions in hierarchical multiclass classification. Here, colored nodes are the positive labels for the given multilabel. Misclassified nodes that contribute to each loss are marked in squares.

4) Ranking loss: Multilabel classification is sometimes cast as a bipartite ranking problem [23], [24]. A nonzero loss is incurred when a positive label is ranked after a negative one, leading to

$$\ell_{\text{rank}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{(i,j):y_i > y_j} \left(I(\hat{y}_i < \hat{y}_j) + \frac{1}{2} I(\hat{y}_i = \hat{y}_j) \right). \quad (6)$$

Many of these have also been used in hierarchical multilabel classification [9]. Yet, a major deficiency is that their definitions are independent of the hierarchy. For example, both predictions A and B in Fig. 1 have the same Hamming loss. However, in hierarchical classification, misclassifications at the upper hierarchy levels (which correspond to more generic concepts) are often considered more expensive than those at the lower levels (which correspond to more specific concepts) [8], [9]. Hence, B should be inferior.

2.2.2 Hierarchical Multiclass Classification

In hierarchical multiclass classification, only a single path can be predicted positive. The following commonly used loss functions are all closely related.

1) Cai and Hofmann [34] defined the loss

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i \in \mathcal{H}} c_i I(\hat{y}_i \neq y_i), \quad (7)$$

where $c_i \geq 0$ (in [34], all the nodes have the same c_i). Intuitively, nodes that are on the path from the most specific nodes (i.e., node at the lowest level) of $\hat{\mathbf{y}}$ and $\mathbf{y}$ to the first common ancestor of $\hat{\mathbf{y}}$ and $\mathbf{y}$ are penalized. An illustration is shown in Fig. 2b.

2) Dekel et al. [14] defined the loss as the number of nodes on the path between the most specific nodes of $\hat{\mathbf{y}}$ and $\mathbf{y}$. This is similar to (7), but counts also the first common ancestor (Fig. 2c).

3) Mcauley et al. [16] defined the loss as the path length from the most specific node in $\mathbf{y}$ to $\hat{\mathbf{y}}$ (Fig. 2d).

2.2.3 Hierarchical Multilabel Classification

Popular loss functions for hierarchical multilabel classification include:

1) H-loss [8], [9], [30], [35], which counts the first classification mistakes as

$$\begin{aligned} \ell_H(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i:y_i=1, \hat{y}_i=0} c_i I(\hat{\mathbf{y}}_{\text{anc}(i)} = \mathbf{y}_{\text{anc}(i)}) \\ + \beta \sum_{i:y_i=0, \hat{y}_i=1} c_i I(\hat{\mathbf{y}}_{\text{anc}(i)} = \mathbf{y}_{\text{anc}(i)}). \end{aligned} \quad (8)$$

Here, α and β are used to weight false negatives (FN) (the first term on the right of (8)) and false positives (FP) (the second term) differently. In Fig. 1, predictions A and B have the same Hamming loss, but A is better w.r.t. the H-loss (all the c_i's are assumed to be 1).

As mentioned in Section 2.2.1, misclassifications at the upper hierarchy levels are often considered more expensive than those at the lower levels. Thus, c_i in (8) can be used to encode node i's importance by penalizing upper-level misclassified nodes more heavily. When the label hierarchy is a tree, c_i is defined in [9] as

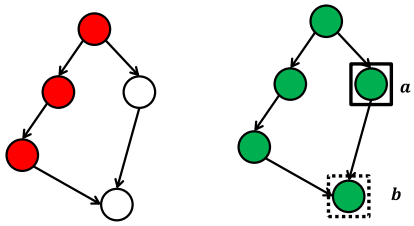
$$c_i = \begin{cases} 1 & i = 0 \text{ (the root)} \\ \frac{c_{\text{pa}(i)}}{n_{\text{sibl}}(i)} & i > 0, \end{cases} \quad (9)$$

where $n_{\text{sibl}}(i)$ is the number of siblings of i (including i). Intuitively, the penalty associated with the parent is equally shared by all its children.

However, since the H-loss only counts the first classification mistakes, it may encourage more mistakes near the bottom of the hierarchy which are not counted. As an example, predictions B and C in Fig. 1 have the same H-loss. Another limitation is that the H-loss can only be used on tree-structured hierarchies. In a DAG, since the root may have multiple paths to a node, definition of the "first" classification mistake can be ambiguous (Fig. 3).

2) For two multilabels $\hat{\mathbf{y}}$ and $\mathbf{y}$, Nowak et al. [25] defined the matching loss as[1]

1. For simplicity, we set the annotation agreement factors and α-factor in [25] to 1.



(a) Ground truth. (b) Prediction.

Fig. 3. Ambiguity in defining the H-loss. Node a is always counted as a classification mistake, but node b is a classification mistake only if the left path is taken.

$$\ell_{\text{match}}(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i:y_i=1} \phi(i, \hat{\mathbf{y}}) + \beta \sum_{i:\hat{y}_i=1} \phi(i, \mathbf{y}), \tag{10}$$

where $\phi(i, \mathbf{y}) = \min_{j:y_j=1} \text{cost}(j \to i)$, and $\text{cost}(j \to i)$ is the cost to traverse from node j to node i in the hierarchy. This can be defined as the path length from j to i, or the total weight along this path if the edges are weighted (Fig. 4a). In particular, in a label tree, since every node (except the root) has unit in-degree, one can use (9) as the weight of the edge incident on a node. We can then rewrite $\phi(i, \mathbf{y})$ as

$$\phi(i, \mathbf{y}) = \sum_{(u,v) \in \text{ path from } \mathbf{y} \text{ to } i} c_v. \tag{11}$$

The same holds for $\phi(i, \hat{\mathbf{y}})$.

While the H-loss penalizes only the first misclassified node on the path from $\mathbf{y}$ (or $\hat{\mathbf{y}}$)) to i, the matching loss penalizes all misclassified nodes on this path, with those further away (e.g., near the bottom of the hierarchy) more heavily penalized. On the other hand, similar to the H-loss, the matching loss can be ambiguous on DAGs, where multiple paths from $\mathbf{y}$ (or $\hat{\mathbf{y}}$) to a node may exist (Fig. 4b).

3) Verspoor et al. [36] introduced hierarchical versions of precision, recall and F-score. However, these measures are more expensive to compute and thus not considered in this paper.

2.3 Hierarchical Multilabel Classification: Condensing Sort and Selection Algorithm (CSSA)

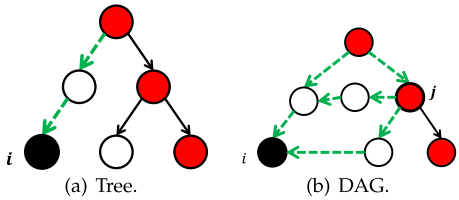
Recently, Bi and Kwok [13] proposed a novel hierarchical multilabel classification algorithm which can be used on both tree- and DAG-structured hierarchies. A key step is to find the multilabel $\hat{\mathbf{y}}$ that is (i) most similar to a given crudely estimated multilabel $\tilde{\mathbf{y}}$; (ii) agrees with the label hierarchy; and (iii) has a pre-determined number of nodes (say, L) predicted positive. For the label tree, they formulated this as the following optimization problem:

$$\max_{\{\psi_i\}_{i \in \mathcal{H}}} \quad \sum_{i \in \mathcal{H}} \psi_i \tilde{y}_i \tag{12}$$

$$\text{s.t.} \quad \psi_i \le \psi_{\text{pa}(i)} \quad \forall i \in \mathcal{H} \backslash \{0\}, \tag{13}$$

$$\psi_0 = 1, \ \psi_i = \{0, 1\}, \tag{14}$$

$$\sum_{i=0}^{N-1} \psi_i = L. \tag{15}$$



(a) Tree. (b) DAG.

Fig. 4. Example illustrating $\phi(i, \mathbf{y})$ in (10). Here, nodes in the ground-truth multilabel are colored red, and node i is a particular misclassified node. (a): The shortest path corresponding to $\phi(i, \mathbf{y})$ is shown in green. (b): Multiple paths exist between $\mathbf{y}$ and node i.

Here, ψ_i is a binary indicator such that $\psi_i = 1$ denotes that node i is predicted positive in $\hat{\mathbf{y}}$; and 0 otherwise. Constraint (13) encodes the hierarchy constraint in (1); while constraint (15) requires that L nodes are predicted positive. It can be shown that problem (12) can be solved efficiently in $O(N \log N)$ time, via a greedy algorithm called *condensing sort and selection algorithm* [13].

When the label hierarchy is a DAG, one only has to replace constraint (13), which is used to encode the hierarchy constraint for label trees, to

$$\psi_i \le \psi_j \quad \forall i \in \mathcal{H} \backslash \{0\}, \forall j \in \text{Pa}(i), \tag{16}$$

which corresponds to the hierarchy constraint in (2). The resultant optimization problem can also be solved efficiently by a DAG extension of CSSA (called CSSAG). Interested readers are referred to [13] for details.

As mentioned above, both CSSA and CSSAG require the user to pre-determine the number of positive labels (L) in $\hat{\mathbf{y}}$. Moreover, loss function is not explicitly considered in their formulation.

3 Revisiting Hamming Loss and Ranking Loss

As the decision on each label node represents the classifier's cognition on that label, mistake at every node should be taken into consideration. In this section, we revisit the Hamming loss in (4), and the ranking loss in (6) (Section 2.2.1), which are originally used for flat classification. It will be seen that their hierarchical extensions can (i) avoid the problem with H-loss (resp. matching loss) that misclassfications at the lower levels may not be penalized (or penalized too heavily); and (ii) be used on both tree and DAG label hierarchies.

3.1 Hierarchical Hamming Loss

Denote the misclassification cost associated with node i by $c_i \ge 0$. The hierarchical Hamming loss is defined as

$$\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i:y_i=1 \wedge \hat{y}_i=0} c_i + \beta \sum_{i:y_i=0 \wedge \hat{y}_i=1} c_i. \tag{17}$$

The first term on the right corresponds to false negatives, while the second term is for false positives.

In [34], their loss function in (7) for hierarchical multiclass classification can also been seen as an extension of Hamming loss, thus is similar to our extension. However, all the nodes have the same c_i in (7). Here, as discussed in Section 2.2.3, we want to incorporate hierarchy information by penalizing upper-level misclassifications more heavily. Thus, when the label hierarchy is a tree, we follow the H-loss and define c_i's as in (9). When the label hierarchy is a DAG, (9) can be

analogously extended as

$$c_i = \begin{cases} 1 & i = 0 \\ \sum_{j \in \mathrm{Pa}(i)} \frac{c_j}{n_{\mathrm{child}}(j)} & i > 0 \end{cases}, \quad (18)$$

where $n_{\mathrm{child}}(j)$ is the number of child nodes of j.

3.1.1 Special Cases

The definition in (17) is quite flexible. Even for the special case where all c_i's are 1, it already encompasses many of the loss functions in Section 2.2. For example, when used for flat classification:

- On setting $\alpha = \beta = 1$, $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ reduces to the Hamming loss in (4).
- Suppose that a given number (k) of labels are to be predicted positive. On setting $\alpha = 0, \beta = 1/L$,

$$\begin{aligned} \ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y}) &= \frac{1}{k} \sum_i I(y_i = 0 \wedge \hat{y}_i = 1) \\ &= 1 - \frac{1}{k} \sum_i I(\hat{y}_i = y_i = 1). \end{aligned}$$

  Note that $\sum_i I(\hat{y}_i = y_i = 1)$ is the number of true positives in $\hat{\mathbf{y}}$. Hence, $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ becomes (5), and minimizing $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ becomes maximizing the top-k precision.

When used for hierarchical multiclass classification, both $\hat{\mathbf{y}}$ and $\mathbf{y}$ consist of one single path, and $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ can be reduced to the following loss functions in Section 2.2.2:

- When $\alpha = \beta = 1$, $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ reduces to the loss in (7). As discussed in Section 2.2.2, this differs from the loss function in [14] by 1.
- On setting $\alpha = 1$ and $\beta = 0$, $\ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ reduces to the loss proposed in [16].

3.2 Hierarchical Ranking Loss

The ranking loss in (6) can also be easily extended for hierarchical classification as

$$\ell_{\text{H-rank}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{(i,j): y_i > y_j} c_{ij} \left(I(\hat{y}_i < \hat{y}_j) + \frac{1}{2} I(\hat{y}_i = \hat{y}_j) \right), \quad (19)$$

where c_{ij} is the misclassification cost on a misplaced label pair (i, j). To ensure a high penalty when an upper-level positive label is ranked after a lower-level negative label, we set

$$c_{ij} = c_i c_j, \quad (20)$$

where c_i is as defined in (9) for tree hierarchies or (18) for DAG hierarchies. Obviously, other settings can also be used.

3.3 Remarks

The hierarchical Hamming loss counts every misclassification node, and the hierarchical ranking loss counts every misplaced label pair. Thus, they avoid the problem with H-loss that some lower-level misclassifications may not be penalized. Moreover, by setting the c_i's according to the label hierarchy as in (9), upper-level label misclassifications/misplacements can be penalized more, thus alleviating the problem with matching loss that lower-level mistakes may be over-penalized. Besides, as discussed, these two losses can be readily used on both tree and DAG label hierarchies.

4 Minimizing the Risk

Given a sample $\mathbf{x}$, the conditional risk (or simply the risk) $\mathcal{R}(\hat{\mathbf{y}})$ of predicting multilabel $\hat{\mathbf{y}}$ is the expectation of $\ell(\hat{\mathbf{y}}, \mathbf{y})$ over all possible $\mathbf{y}$'s as ground truth, i.e.,

$$\mathcal{R}(\hat{\mathbf{y}}) = \sum_{\mathbf{y}} \ell(\hat{\mathbf{y}}, \mathbf{y}) P(\mathbf{y}|\mathbf{x}). \quad (21)$$

From Bayesian decision theory [20], the optimal $\hat{\mathbf{y}}^*$ is the one that minimizes the risk:

$$\hat{\mathbf{y}}^* = \arg\min_{\hat{\mathbf{y}} \in \Omega} \mathcal{R}(\hat{\mathbf{y}}), \quad (22)$$

where Ω is the set of feasible multilabel predictions satisfying the hierarchy constraint (1) or (2).

Though risk minimization has been widely discussed in flat multilabel classification [23], [28], it has only received limited attention in the more complicated hierarchical multilabel classification. To the best of our knowledge, only the H-loss has been considered [30]. In this section, we discuss how $\hat{\mathbf{y}}^*$ can be obtained for a number of loss functions shown in Sections 2.2 and 3. We will focus our discussion on the matching loss, hierarchical Hamming loss and hierarchical ranking loss. Since hierarchical Hamming loss admits many loss functions as special cases (Section 3.1.1), the results for such special cases can be directly obtained thus we do not discuss them to avoid redundancy. Specifically, the following issues will need to be addressed:

1) How to estimate $P(\mathbf{y}|\mathbf{x})$ in (21) from the data?
2) How to efficiently compute $\mathcal{R}(\hat{\mathbf{y}})$ for a particular $\hat{\mathbf{y}}$, without exhaustively summing all the possibly up to 2^N combinations of $\mathbf{y}$'s in (21)?
3) How to efficiently minimize $\mathcal{R}(\hat{\mathbf{y}})$ in (22), without exhaustively enumerating all the possibly up to 2^N combinations of $\hat{\mathbf{y}}$?

Specifically, the first issue corresponds to the training phase, which is independent of the loss function; the latter two correspond to the prediction phase, which will be loss-specific. These will be covered in Sections 4.1, 4.2, and 4.3, respectively.

4.1 Estimation of $P(\mathbf{y}|\mathbf{x})$

As in [8], we assume that the labels of a group of sibling nodes in the label hierarchy are conditionally independent given their parent label(s). This simplification is standard in Bayesian networks and also commonly used in hierarchical multilabel classification [37], [38]. Thus, for a tree label hierarchy, we have

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i \in \mathcal{H} \setminus \{0\}} P(y_i | y_{\mathrm{pa}(i)}, \mathbf{x}). \quad (23)$$

Moreover, $P(y_i = 1 | y_{\mathrm{pa}(i)} = 0, \mathbf{x}) = 0$ as such a label combination violates the hierarchy constraint (1). Similarly, for a DAG label hierarchy, we have

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i \in \mathcal{H} \setminus \{0\}} P(y_i|\mathbf{y}_{\mathrm{Pa}(i)}, \mathbf{x}), \tag{24}$$

and $P(y_i = 1|\mathbf{y}_{\mathrm{Pa}(i)}, \mathbf{x}) = 0$ if $y_j = 0$ for any $j \in \mathrm{Pa}(i)$.

With this simplification, we only need to train estimators for $p(y_i = 1|y_{\mathrm{pa}(i)} = 1, \mathbf{x})$ (resp. $p(y_i = 1|\mathbf{y}_{\mathrm{Pa}(i)} = \mathbf{1}, \mathbf{x})$) for each tree (resp. DAG) label node i, using methods such as logistic regression or support vector machines. If the number of labels is very large, we can first do some label dimension reduction or label selection, then efficiently train the model in a much smaller reduced space, and recover the probabilistic estimations for all the labels from the reduced space or selected label subset [13], [39]. Also, if auxiliary properties about the labels exist, such as label imbalance or sparsity, label reduction methods can help to achieve better prediction performance [13], [40]. The algorithm in this work is nevertheless independent of the way these probability estimators are learned.

4.2 Efficient Computation of $\mathcal{R}(\hat{\mathbf{y}})$

4.2.1 Matching Loss in (10)

Let p_i be the probability that node i is labeled positive given $\mathbf{x}$. By the hierarchy constraint in (1), all ancestors of i must also be positive. Hence,

$$p_i = P(y_i = 1|\mathbf{x}) = P(y_i = 1, \mathbf{y}_{\mathrm{anc}(i)} = 1|\mathbf{x}). \tag{25}$$

Recall that the matching loss is only defined on tree label hierarchies. Using (23),

$$\begin{aligned} p_i &= P(y_i = 1|y_{\mathrm{pa}(i)} = 1, \mathbf{x}) \prod_{j \in \mathrm{anc}(i) \setminus \{0\}} P(y_j = 1|y_{\mathrm{pa}(j)} = 1, \mathbf{x}) \\ &= P(y_i = 1|y_{\mathrm{pa}(i)} = 1, \mathbf{x}) p_{\mathrm{pa}(i)}. \end{aligned}$$

Note that the root is always labeled positive, and so $p_0 = 1$. By traversing the tree with breadth-first-search (BFS) or depth-first-search (DFS), all p_i's can be obtained in $O(N)$ time.

With the p_i's obtained, the following Proposition shows that the corresponding risk $\mathcal{R}_{\mathrm{match}}(\hat{\mathbf{y}}) = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \ell_{\mathrm{match}}(\hat{\mathbf{y}}, \mathbf{y})$ can be easily computed (proof is in Appendix A, available online).

Proposition 1. *With a label tree $\mathcal{T}$,*

$$\mathcal{R}_{\mathrm{match}}(\hat{\mathbf{y}}) = \sum_{i:\hat{y}_i = 0} \phi(i, \hat{\mathbf{y}}) p_i + \sum_{i:\hat{y}_i = 1} q_i, \tag{26}$$

where

$$q_i = \sum_{j=0}^{d(i)-1} \sum_{l=j+1}^{d(i)} c_{\mathrm{anc}_l(i)} P(\mathbf{y}_{\mathrm{anc}_{0:j}(i)} = \mathbf{1}, y_{\mathrm{anc}_{j+1}(i)} = 0|\mathbf{x}), \tag{27}$$

$d(i)$ is the depth of i (the root has depth 0), $\mathrm{anc}_j(i)$ is i's ancestor at depth j, $anc_{0:j}(i) = \{anc_0(i), anc_1(i), \ldots, anc_j(i)\}$ is the set of i's ancestors at depths 0 to j. The two terms on the RHS of (26) are due to false negatives and false positives, respectively.

Note that the q_i's in (27) can be efficiently computed. First, on using (23), we have

$$\begin{aligned} &P(\mathbf{y}_{\mathrm{anc}_{0:j}(i)} = \mathbf{1}, y_{\mathrm{anc}_{j+1}(i)} = 0|\mathbf{x}) \\ &\quad = P(y_{\mathrm{anc}_{j+1}(i)} = 0|y_{\mathrm{anc}_j(i)} = 1, \mathbf{x}) \\ &\qquad \cdot \prod_{l=1}^{j} P(y_{\mathrm{anc}_l(i)} = 1|y_{\mathrm{anc}_{l-1}(i)} = 1, \mathbf{x}) \\ &\quad = p_{\mathrm{anc}_j(i)} (1 - P(y_{\mathrm{anc}_{j+1}(i)} = 1|y_{\mathrm{anc}_j(i)} = 1, \mathbf{x})). \end{aligned} \tag{28}$$

Given p_i's obtained above and $P(y_i = 1|y_{\mathrm{pa}(i)} = 1, \mathbf{x})$'s estimated in Section 4.1, one can compute $\{P(\mathbf{y}_{\mathrm{anc}_{0:j}(i)} = \mathbf{1}, y_{\mathrm{anc}_{j+1}(i)} = 0|\mathbf{x})\}_{i=1}^N$ in (28) in $O(N)$ time. Then we compute q_i using (27) by traversing the path from root to node i, which takes $O(d(i))$ time. Assuming that the nodes have been indexed in topological order, we have $d(i) \leq i$. Thus, computing $\{q_i\}_{i=1}^N$ takes $O(N + \sum_{i=1}^N d(i)) \leq O(N + \sum_{i=1}^N i) = O(N^2)$ time. Hence, $\mathcal{R}_{\mathrm{match}}(\hat{\mathbf{y}})$ can be computed in a total of $O(N + N^2) = O(N^2)$ time.

4.2.2 Hierarchical Hamming Loss in (17)

For both tree and DAG label hierarchies, the risk $\mathcal{R}_{\text{H-hamming}}(\hat{\mathbf{y}}) = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \ell_{\text{H-hamming}}(\hat{\mathbf{y}}, \mathbf{y})$ can be easily computed by the following Proposition (proof is in Appendix B, available online). As in Proposition 1, the two terms on the RHS are due to false negatives and false positives, respectively.

Proposition 2.

$$\mathcal{R}_{\text{H-hamming}}(\hat{\mathbf{y}}) = \alpha \sum_{i:\hat{y}_i = 0} c_i p_i + \beta \sum_{i:\hat{y}_i = 1} c_i (1 - p_i). \tag{29}$$

For a label tree, we have shown in Section 4.2.1 that the p_i's can be computed in $O(N)$ time. The following shows that they can also be computed efficiently for a label DAG. Specifically, on using (24),

$$p_i = P(y_i = 1|\mathbf{y}_{\mathrm{Pa}(i)} = \mathbf{1}, \mathbf{x}) \prod_{j \in \mathrm{anc}(i) \setminus \{0\}} P(y_j = 1|\mathbf{y}_{\mathrm{Pa}(j)} = \mathbf{1}, \mathbf{x}). \tag{30}$$

Observe that $\mathrm{anc}(i) = \mathrm{Pa}(i) \cup \{\mathrm{anc}(j)\}_{j \in \mathrm{Pa}(i)}$. Hence, all the $\mathrm{anc}(i)$'s can be obtained recursively by traversing the DAG using topological sort, which takes $O(N + E)$ time (where E is the number of edges in $\mathcal{H}$). Note that this only needs to be computed once, as part of preprocessing. With $\mathrm{anc}(i)$ and $P(y_i = 1|\mathbf{y}_{\mathrm{Pa}(i)} = \mathbf{1}, \mathbf{x})$'s estimated in Section 4.1, each p_i in (30) can be computed in $O(|\mathrm{anc}(i)|)$ time. Each $|\mathrm{anc}(i)|$ in turn is upper-bounded by the number of nodes placed before it in the topological order. Hence, the time to compute all p_i's is $\sum_{i=1}^N |\mathrm{anc}(i)| \leq \sum_{i=1}^N i = O(N^2)$. Thus, $\mathcal{R}_{\text{H-hamming}}(\hat{\mathbf{y}})$ can be computed in a total of $O(N + N^2) = O(N^2)$ time.

Remark 1. Recall from Section 3.1.1 that the hierarchical Hamming loss encompasses many loss functions, the corresponding $\mathcal{R}(\hat{\mathbf{y}})$'s can be easily obtained by proper settings of α, β and c_i's. For example, the risk for the Hamming loss is $\sum_{i:\hat{y}_i = 0} p_i + \sum_{i:\hat{y}_i = 1} (1 - p_i)$.

4.2.3 Hierarchical Ranking Loss in (19)

For both tree and DAG label hierarchies, the risk $\mathcal{R}_{\text{H-rank}}(\hat{\mathbf{y}}) = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \ell_{\text{H-rank}}(\hat{\mathbf{y}}, \mathbf{y})$ can be easily computed

from the p_i's in (25) using the following Proposition (proof is in Appendix C, available online).

Proposition 3.

$$\mathcal{R}_{\text{H-rank}}(\hat{\mathbf{y}}) = \sum_{0\le i<j\le N-1} c_{ij}(p_i I(\hat{y}_i < \hat{y}_j) + p_j I(\hat{y}_i > \hat{y}_j) + \frac{p_i+p_j}{2} I(\hat{y}_i = \hat{y}_j)) - C, \tag{31}$$

where C is independent of $\hat{\mathbf{y}}$.

4.3 Efficient Minimization of $\mathcal{R}(\hat{\mathbf{y}})$

For the loss functions considered in Section 4.2, we will now show how to efficiently find the $\hat{\mathbf{y}}^*$ that minimizes the corresponding risk.

4.3.1 Common Optimization Problem

First, we decompose problem (22) into N subproblems, each for a fixed value of $L \in \{1,\ldots,N\}$:

$$\hat{\mathbf{y}}^* = \arg\min_{L=1,\ldots,N} \mathcal{R}(\hat{\mathbf{y}}^*_{(L)}), \tag{32}$$

where

$$\hat{\mathbf{y}}^*_{(L)} = \arg\min_{\hat{\mathbf{y}}\in\Omega} \mathcal{R}(\hat{\mathbf{y}}) \ : \ |\text{supp}(\hat{\mathbf{y}})| = L \tag{33}$$

is the optimal multilabel with L nodes labeled positive. The following Proposition shows how $\mathbf{y}_{(L)}$ can be obtained for the various loss functions (proof is in Appendix D, available online).

Proposition 4. *Let $\{n_1, n_2, \ldots, n_L\}$ be the L nodes labeled positive in $\hat{\mathbf{y}}^*_{(L)}$, $\hat{\mathbf{y}}_{(l)}$ be the multilabel with only $\{n_1, n_2, \ldots, n_l\}$ labeled positive, and for node i,*

$$\delta(i) = \begin{cases} \alpha c_i\left(p_i + \sum_{j\in desc(i)} p_j\right) - \beta q_i & \textit{matching loss} \\ c_i(\alpha p_i - \beta(1-p_i)) & \textit{hier. Hamming loss} \\ \frac{1}{2}\sum_{j\in\mathcal{H}} c_{ij}(p_i - p_j) & \textit{hier. ranking loss}, \end{cases}$$

where $\text{desc}(i)$ is its set of descendants. Then the n_i's can be obtained as:

$$\max_{n_1,n_2,\ldots,n_L} \ \sum_{l=1}^{L} \delta(n_l) \quad \text{s.t.} \quad \{\hat{\mathbf{y}}_{(l)}\}_{l=1}^{L} \textit{ are valid multilabels in } \mathcal{H}. \tag{34}$$

The $\hat{\mathbf{y}}^*_{(L)}$ for zero-one loss can also be obtained in a similar manner. Interested readers are referred to Proposition 1 in [41] for details.

4.3.2 Solving the Optimization Problem (34)

We associate a binary indicator $\theta(i) \in \{0,1\}$ with each node i, where $\theta(i) = 1$ denotes that node i is selected by $\hat{\mathbf{y}}^*_{(L)}$, and 0 otherwise. The objective in (34) can then be written as $\sum_{i=0}^{N-1} \theta(i)\delta(i)$. Moreover, the hierarchy constraints on $\hat{\mathbf{y}}_{(l)}$'s can be enforced by adding constraints (13) for label trees, or (16) for label DAGs. Thus, we obtain the following reformulations of (34).

Proposition 5. *For a label tree, problem (34) can be reformulated as the following problem:*

$$\begin{aligned} \max_{\theta} \ & \sum_i \theta(i)\delta(i) \\ \text{s.t.} \ & \theta(i) \le \theta(pa(i)) \quad \forall i \in \mathcal{H}\backslash\{0\}, \\ & \theta(0) = 1, \ \theta(i) = \{0,1\}, \sum_{i=0}^{N-1} \theta(i) = L. \end{aligned} \tag{35}$$

Proposition 6. *For a label DAG, problem (34) can be reformulated as the following problem:*

$$\begin{aligned} \max_{\theta} \ & \sum_i \theta(i)\delta(i) \\ \text{s.t.} \ & \theta(i) \le \theta(j) \quad \forall i \in \mathcal{H}\backslash\{0\}, \forall j \in Pa(i), \\ & \theta(0) = 1, \ \theta(i) = \{0,1\}, \sum_{i=0}^{N-1} \theta(i) = L. \end{aligned} \tag{36}$$

Interestingly, (35) and (36) are of the same form as the optimization problem in (12), except that $\tilde{y}_i$ in (12) is now replaced by $\delta(i)$. Thus, we can reuse the efficient CSSA (resp. CSSAG) algorithm in [13] for the tree (resp. DAG) label hierarchy.

Recall from (32) that we need to first compute the risks for $L = 1,\ldots,N$ using the above procedure, and then pick the L with the smallest risk. A straightforward procedure is to run CSSA/CSSAG $N-1$ times (the case for $L=1$ trivially yields the multilabel with only the root labeled positive). However, since CSSA/CSSAG is a greedy algorithm, the optimal solution of size L_1 contains all the optimal solutions of sizes $L_2 < L_1$. Thus, we can simply set $L = N$, and keep track of the optimal solution obtained for each intermediate value of L. The total time to obtain the optimal solution (with the L value yielding the smallest risk) is then still $O(N \log N)$.

The complete algorithm, which will be called HIerarchical Risk-Optimizing Multilabel classification (HIROM), is shown in Algorithm 1. Recall that CSSA/CSSAG in [13] requires as input the number of labels (L) to be predicted. On the other hand, HIROM can automatically determine L.

Algorithm 1. Hierarchical Risk-Optimizing Multilabel Classification

1: Train estimators for $p(y_i = 1|y_{\text{pa}(i)} = 1, \mathbf{x})$ (resp. $p(y_i = 1|y_{\text{Pa}(i)} = 1, \mathbf{x})$) for all $i \in \mathcal{H}$.
2: **for** each testing sample **do**
3: For label tree: Traverse the tree $\mathcal{H}$ using BFS or DFS to compute p_i for all $i \in \mathcal{H}$.
For label DAG: Perform BFS and obtain $\text{anc}(i)$'s for all $i \in \mathcal{H}$, then compute p_i's with $\text{anc}(i)$'s.
4: For matching loss only: Compute q_i for all $i \in \mathcal{H}$.
5: Compute $\delta(i)$'s for all $i \in \mathcal{H}$ using the corresponding formula in Proposition 4.
6: Use CSSA (resp. CSSAG) for label tree (resp. DAG) in [13] with the computed $\delta(i)$'s, and obtain $\{\hat{\mathbf{y}}^*_{(1)},\ldots,\hat{\mathbf{y}}^*_{(N)}\}$.
7: $\hat{\mathbf{y}}^* \leftarrow \arg\min_{\hat{\mathbf{y}}^*_{(L)}} \mathcal{R}(\hat{\mathbf{y}}^*_{(L)})$.
8: **end for**

TABLE 1
Data Sets with Tree Label Hierarchies

data set	#sample	#feature	#label	cardinality
rcv1-subset1	6,000	47,236	104	4.23
rcv1-subset2	6,000	47,236	104	4.23
rcv1-subset3	6,000	47,236	104	4.22
rcv1-subset4	6,000	47,236	104	4.21
rcv1-subset5	6,000	47,236	104	4.21
enron	1,648	1,001	57	6.30
eurolex-dc	19,348	5,000	865	6.00
seq	3,919	489	500	9.55
pheno	1,591	170	456	10.08
struc	4,291	19,629	500	9.67
hom	3,854	47,035	500	9.57
cellcycle	4,190	77	500	9.76
church	4,192	26	500	9.75
derisi	3,725	63	500	9.79
eisen	2,424	79	462	10.20
gasch1	3,764	173	500	9.74
gasch2	3,779	52	500	9.74
spo	3,703	83	500	9.78
expr	3,779	551	500	9.74

("cardinality" is the average number of labels per sample).

TABLE 2
Data Sets with DAG Label Hierarchies

data set	#sample	#feature	#label	cardinality
seq	3,828	489	143	5.87
pheno	1,556	170	58	5.03
struc	3,752	19,629	142	5.78
hom	3,766	47,035	140	5.88
cellcycle	3,680	77	141	5.75
church	3,678	26	141	5.74
derisi	3,648	63	141	5.77
eisen	2,414	79	101	5.69
gasch1	3,687	173	141	5.74
gasch2	3,701	52	141	5.74
spo	3,626	83	139	5.75
expr	3,702	551	141	5.74

5 Experiments

5.1 Setup

In this section, we perform experiments on a number of real-world data sets commonly used in multilabel classification. These include 19 data sets with tree label hierarchies (Table 1).

- Five subsets of the RCV1v2 data [42].[2] These contain documents in the REUTERS topics categories with the topic hierarchy.
- Enron[43]:[3] This is a text data set for email analysis, and its label hierarchy describes the relationships of the email users.
- eurolex-dc [44]:[4] It contains a collection of documents on the European Union law. The labels include several EuroVoc descriptors, directory codes and subject matters. Here, we use the second one, as it has four level label hierarchy.
- Twelve genomic data sets [3]:[5] These contain different aspects of genes in the yeast genome, with annotations from MIPS's functional catalogue (Funcat).

For the DAG-structured data sets[6] (Table 2), they are the same set of genomic data sets, but the samples are annotated from the GO [3]. There are three subgraphs in its label DAG. The first subgraph is used in the experiment. For preprocessing, we remove labels with fewer than 10 positive training samples as in [35].

Each of the constituent probabilities in $P(\mathbf{y}|\mathbf{x})$ are obtained from a SVM trained on the training set. Specifically, for each node i, we first train a binary (linear) SVM, using those samples that the parent(s) of i is labeled positive as training examples [30]. The SVM output is then converted to a probability estimate using the procedure in [45]. The SVM's C parameter is chosen from $\{2^{-10}, 2^{-9}, \ldots, 1, \ldots, 2^9, 2^{10}\}$ using three-fold cross-validation on the training set.

The proposed HIROM algorithm will be compared with the following state-of-the-art methods.

1) CSSA (resp. CSSAG) [13]: As discussed in Section 2.3, this is most similar to the proposed algorithm, except that the loss function is not used in its formulation. Recall that CSSA requires as input the number of labels to be predicted (L). Here, we first run HIROM and use the number of labels obtained as input to CSSA.
2) H-SVM [8]: This trains a binary classifier at each node. On prediction, if $p(y_i = 1|y_{\mathrm{pa}(i)} = 1, \mathbf{x})$ or $p(y_i = 1|y_{\mathrm{Pa}(i)} = \mathbf{1}, \mathbf{x})$ is $\geq \tau$, node i is predicted positive and the process continues to its children. τ is set by a 3-fold cross validation on the given training set.
3) B-SVM [35]: We use the cost-sensitive extension in [30]. It is optimal w.r.t. the H-loss, but can only be used on tree label hierarchies.
4) CLUS-HMC [3]: This extends the decision tree for multilabel classification on both tree and DAG label hierarchies.

Performance evaluation is based on five-fold cross-validation.

5.2 Matching Loss (10)

The matching loss can only be used with data sets having tree label hierarchies. We set c_i using (9), and $\alpha = \lambda\beta$, where $\lambda = \frac{n_-}{n_+}$ balances the misclassification costs between the n_+ positive labels (i.e., $y_i = 1$ on all training samples), and n_- negative labels (i.e., $y_i = 0$ on all training samples). The α, β values are normalized as $\alpha + \beta = 2$. Note that for the various algorithms under comparison, only HIROM and B-SVM depend on α.

Results are shown in Table 3. As can be seen, HIROM achieves the smallest loss as expected, and outperforms CSSA by a significant margin.

5.3 Hierarchical Hamming Loss (17)

The hierarchical Hamming loss can be used on both tree and DAG label hierarchies. We set c_i using (9) for the label

2. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html
3. http://www.cs.cmu.edu/~enron/
4. http://www.ke.tu-darmstadt.de/resources/eurlex
5. http://dtai.cs.kuleuven.be/clus/hmcdatasets/
6. http://dtai.cs.kuleuven.be/clus/hmcdatasets/

TABLE 3
Matching Loss Values on Data Sets with Tree Label Hierarchies

data set	HIROM	CSSA	H-SVM	B-SVM	CLUS-HMC
rcv1-subset1	**0.074** $\pm$ 0.003	0.131 $\pm$ 0.003	0.081 $\pm$ 0.004	0.164 $\pm$ 0.003	0.204 $\pm$ 0.015
rcv1-subset2	**0.072** $\pm$ 0.003	0.131 $\pm$ 0.004	0.082 $\pm$ 0.006	0.159 $\pm$ 0.004	0.204 $\pm$ 0.019
rcv1-subset3	**0.074** $\pm$ 0.005	0.132 $\pm$ 0.005	0.082 $\pm$ 0.006	0.167 $\pm$ 0.007	0.217 $\pm$ 0.013
rcv1-subset4	**0.074** $\pm$ 0.002	0.126 $\pm$ 0.006	0.081 $\pm$ 0.006	0.162 $\pm$ 0.004	0.199 $\pm$ 0.026
rcv1-subset5	**0.074** $\pm$ 0.005	0.124 $\pm$ 0.004	0.083 $\pm$ 0.005	0.164 $\pm$ 0.007	0.204 $\pm$ 0.006
enron	**0.148** $\pm$ 0.011	0.231 $\pm$ 0.023	0.204 $\pm$ 0.043	0.305 $\pm$ 0.014	0.224 $\pm$ 0.037
eurolex-dc	**0.033** $\pm$ 0.001	0.190 $\pm$ 0.003	0.225 $\pm$ 0.002	0.248 $\pm$ 0.008	-
seq	**0.145** $\pm$ 0.004	0.657 $\pm$ 0.234	0.228 $\pm$ 0.007	0.749 $\pm$ 0.026	0.212 $\pm$ 0.019
pheno	**0.177** $\pm$ 0.011	0.904 $\pm$ 0.073	0.269 $\pm$ 0.010	0.974 $\pm$ 0.036	0.242 $\pm$ 0.041
struc	**0.156** $\pm$ 0.004	0.786 $\pm$ 0.043	0.249 $\pm$ 0.009	0.872 $\pm$ 0.045	0.220 $\pm$ 0.006
hom	**0.145** $\pm$ 0.007	0.591 $\pm$ 0.024	0.207 $\pm$ 0.007	0.700 $\pm$ 0.019	0.247 $\pm$ 0.007
cellcycle	**0.150** $\pm$ 0.003	0.741 $\pm$ 0.008	0.231 $\pm$ 0.005	0.828 $\pm$ 0.011	0.231 $\pm$ 0.030
church	**0.157** $\pm$ 0.006	0.821 $\pm$ 0.028	0.262 $\pm$ 0.013	0.888 $\pm$ 0.069	0.212 $\pm$ 0.023
derisi	**0.156** $\pm$ 0.030	0.792 $\pm$ 0.027	0.261 $\pm$ 0.005	0.894 $\pm$ 0.040	0.209 $\pm$ 0.012
eisen	**0.156** $\pm$ 0.008	0.723 $\pm$ 0.045	0.243 $\pm$ 0.013	0.800 $\pm$ 0.021	0.248 $\pm$ 0.022
gasch1	**0.146** $\pm$ 0.004	0.693 $\pm$ 0.017	0.242 $\pm$ 0.005	0.765 $\pm$ 0.012	0.236 $\pm$ 0.012
gasch2	**0.155** $\pm$ 0.016	0.757 $\pm$ 0.096	0.757 $\pm$ 0.096	0.814 $\pm$ 0.030	0.320 $\pm$ 0.035
spo	**0.156** $\pm$ 0.003	0.817 $\pm$ 0.029	0.256 $\pm$ 0.006	0.856 $\pm$ 0.025	0.218 $\pm$ 0.012
expr	**0.146** $\pm$ 0.003	0.660 $\pm$ 0.015	0.234 $\pm$ 0.006	0.736 $\pm$ 0.018	0.223 $\pm$ 0.012

The results with best mean values are in bold Note that CLUS-HMC runs out of memory on the Euro-lex(dc) *data set.*

tree, and (18) for the label DAG. The α and β values are set as in Section 5.2.

Table 4 shows the hierarchical Hamming loss values on the tree-structured data sets. As can be seen, HIROM again achieves the smallest loss as expected. The B-SVM is also quite competitive. Table 5 shows the results on the DAG-structured data sets. Again, HIROM consistently outperforms CSSA and H-SVM. Note that the bottom-up strategy used in B-SVM cannot be extended to handle DAG label structures and so cannot be compared here. Because of the lack of space, we do not report the sensitivity of the performance w.r.t. λ. Interested readers are referred to [31] for details.

5.4 Hierarchical Ranking Loss (19)

We set c_{ij} as in (20). Results of the tree-structured and DAG-structured data sets are shown in Tables 6 and 7, respectively. For text data sets (rcv1, H-SVM performs the best and HIROM almost achieves the best performance (except on rcv1-subset5). However, for the genomic data sets, H-SVM can not perform well consistently as HIROM. A possible reason is that the cardinalities of the text data sets is generally small, and most positive labels are on the top levels of the hierarchy, thus predicting the positive labels from top to bottom as in H-SVM can obtain a ranking of all the labels well. However, for genomic data sets, the cardinality increases and some positive labels lie on

TABLE 4
Hierarchical Hamming Loss Values on Data Sets with Tree Label Hierarchies

data set	HIROM	CSSA	H-SVM	B-SVM	CLUS-HMC
rcv1-subset1	**0.039** $\pm$ 0.001	0.044 $\pm$ 0.001	0.046 $\pm$ 0.001	0.042 $\pm$ 0.001	0.086 $\pm$ 0.007
rcv1-subset2	**0.038** $\pm$ 0.001	0.054 $\pm$ 0.002	0.045 $\pm$ 0.001	0.041 $\pm$ 0.001	0.093 $\pm$ 0.005
rcv1-subset3	**0.040** $\pm$ 0.002	0.052 $\pm$ 0.003	0.047 $\pm$ 0.002	0.042 $\pm$ 0.002	0.097 $\pm$ 0.005
rcv1-subset4	**0.039** $\pm$ 0.000	0.053 $\pm$ 0.003	0.046 $\pm$ 0.003	0.042 $\pm$ 0.001	0.089 $\pm$ 0.010
rcv1-subset5	**0.040** $\pm$ 0.002	0.050 $\pm$ 0.002	0.047 $\pm$ 0.002	0.043 $\pm$ 0.002	0.091 $\pm$ 0.003
enron	**0.129** $\pm$ 0.012	0.130 $\pm$ 0.017	0.137 $\pm$ 0.010	0.137 $\pm$ 0.011	0.143 $\pm$ 0.016
eurolex-dc	**0.017** $\pm$ 0.000	0.050 $\pm$ 0.001	0.029 $\pm$ 0.002	0.022 $\pm$ 0.000	-
seq	**0.083** $\pm$ 0.002	0.228 $\pm$ 0.005	0.091 $\pm$ 0.002	0.090 $\pm$ 0.003	0.089 $\pm$ 0.006
pheno	**0.101** $\pm$ 0.004	0.299 $\pm$ 0.016	0.107 $\pm$ 0.004	0.110 $\pm$ 0.007	0.105 $\pm$ 0.011
struc	**0.091** $\pm$ 0.001	0.263 $\pm$ 0.012	0.095 $\pm$ 0.002	0.098 $\pm$ 0.004	0.096 $\pm$ 0.002
hom	**0.081** $\pm$ 0.003	0.208 $\pm$ 0.009	0.089 $\pm$ 0.001	0.087 $\pm$ 0.002	0.098 $\pm$ 0.002
cellcycle	**0.087** $\pm$ 0.002	0.251 $\pm$ 0.003	0.093 $\pm$ 0.001	0.095 $\pm$ 0.003	0.097 $\pm$ 0.003
church	**0.092** $\pm$ 0003	0.271 $\pm$ 0.004	0.095 $\pm$ 0.004	0.099 $\pm$ 0.006	0.094 $\pm$ 0.005
derisi	**0.092** $\pm$ 0.002	0.264 $\pm$ 0.003	0.096 $\pm$ 0.002	0.099 $\pm$ 0.002	0.093 $\pm$ 0.003
eisen	**0.091** $\pm$ 0.004	0.248 $\pm$ 0.011	0.104 $\pm$ 0.004	0.103 $\pm$ 0.004	0.100 $\pm$ 0.002
gasch1	**0.085** $\pm$ 0.002	0.239 $\pm$ 0.004	0.093 $\pm$ 0.002	0.093 $\pm$ 0.002	0.097 $\pm$ 0.003
gasch2	**0.089** $\pm$ 0.009	0.254 $\pm$ 0.022	0.094 $\pm$ 0.009	0.097 $\pm$ 0.010	0.119 $\pm$ 0.013
spo	**0.091** $\pm$ 0.002	0.271 $\pm$ 0.004	0.095 $\pm$ 0.002	0.099 $\pm$ 0.002	0.094 $\pm$ 0.004
expr	**0.083** $\pm$ 0.001	0.231 $\pm$ 0.003	0.091 $\pm$ 0.001	0.093 $\pm$ 0.002	0.092 $\pm$ 0.003

The results with best mean values are in bold.

TABLE 5
Hierarchical Hamming Loss Values on Data Sets with DAG Label Hierarchies

data set	HIROM	CSSAG	H-SVM	CLUS-HMC
seq	**0.105** ± 0.002	0.291 ± 0.007	2.250 ± 0.005	2.284 ± 0.005
pheno	**0.222** ± 0.013	0.281 ± 0.014	2.151 ± 0.010	2.224 ± 0.008
struc	**0.117** ± 0.007	0.301 ± 0.012	2.247 ± 0.006	2.280 ± 0.005
hom	**0.096** ± 0.004	0.244 ± 0.013	2.247 ± 0.008	2.281 ± 0.007
cellcycle	**0.117** ± 0.004	0.327 ± 0.019	2.244 ± 0.003	2.278 ± 0.003
church	**0.128** ± 0.004	0.360 ± 0.015	2.244 ± 0.006	2.278 ± 0.005
derisi	**0.125** ± 0.004	0.344 ± 0.022	2.244 ± 0.004	2.278 ± 0.004
eisen	**0.137** ± 0.008	0.276 ± 0.025	2.224 ± 0.021	2.272 ± 0.019
gasch1	**0.116** ± 0.006	0.226 ± 0.005	2.244 ± 0.003	2.278 ± 0.003
gasch2	**0.120** ± 0.007	0.230 ± 0.016	2.244 ± 0.010	2.279 ± 0.011
spo	**0.126** ± 0.004	0.234 ± 0.005	2.242 ± 0.003	2.277 ± 0.003
expr	**0.113** ± 0.005	0.312 ± 0.026	2.244 ± 0.006	2.278 ± 0.005

The results with best mean values are in bold.

TABLE 7
Hierarchical Ranking Loss Values on Data Sets with DAG Label Hierarchies

data set	HIROM	CSSAG	H-SVM	CLUS-HMC
seq	**0.158** ± 0.004	0.171 ± 0.004	1.450 ± 0.002	1.470 ± 0.002
pheno	**0.185** ± 0.007	0.240 ± 0.012	1.151 ± 0.010	1.188 ± 0.002
struc	**0.168** ± 0.006	0.179 ± 0.003	1.449 ± 0.001	1.468 ± 0.001
hom	**0.136** ± 0.008	0.150 ± 0.005	1.440 ± 0.003	1.460 ± 0.003
cellcycle	**0.184** ± 0.004	0.194 ± 0.005	1.446 ± 0.001	1.466 ± 0.001
church	**0.194** ± 0.004	0.202 ± 0.006	1.447 ± 0.001	1.466 ± 0.002
derisi	**0.191** ± 0.003	0.200 ± 0.007	1.456 ± 0.001	1.476 ± 0.001
eisen	**0.189** ± 0.007	0.216 ± 0.014	1.389 ± 0.005	1.416 ± 0.005
gasch1	**0.179** ± 0.006	0.191 ± 0.006	1.446 ± 0.002	1.466 ± 0.001
gasch2	**0.182** ± 0.002	0.194 ± 0.005	1.446 ± 0.003	1.466 ± 0.003
spo	**0.187** ± 0.003	0.199 ± 0.012	1.440 ± 0.001	1.460 ± 0.001
expr	**0.175** ± 0.006	0.188 ± 0.010	1.447 ± 0.002	1.466 ± 0.001

The results with best mean values are in bold.

some deep levels of some long paths, then top-down prediction scheme may fail to given a good ranking. Overall, HIROM can achieve stable good performance on all cases.

5.5 Comparison among H-Loss, Matching Loss, and Hierarchical Hamming Loss

In this section, we further examine the different behaviors of Bayes-optimal classifiers with respect to H-loss, matching loss, and hierarchical Hamming loss, which penalize FP and FN of a partially mistaken prediction differently. Here we ignore the comparison with hierarchical ranking loss, since it is not directly defined on FP and FN. Experiments are performed on the tree-structured rcv1-subset1, enron and seq data sets. Fig. 5 shows the numbers of FP, and FN at different levels of the label hierarchy as obtained by the B-SVM (which minimizes the H-loss), and the two HIROM variants (which minimize the matching loss and hierarchical Hamming loss, respectively).

As can be seen, at lower levels of the hierarchy, B-SVM has the largest FP and smallest FN; HIROM (matching loss) has the smallest FP and largest FN, while the FP and FN values of HIROM (hierarchical hamming loss) are in between these two. This is consistent with the discussions in Sections 2.2.3 and 3 that the H-loss (resp. matching loss) is most (resp. least) willing to predict positive labels near the bottom of the hierarchy; while the hierarchical Hamming loss is intermediate between the two. Thus, for applications that are interested in retrieving more TP (i.e., less FN) labels, such as information retrieval tasks, the H-loss-minimizing B-SVM can be used; whereas for applications that prefer fewer classification FP mistakes, such as medical imaging annotation, the matching-loss-minimizing HIROM is recommended. On the other hand, for applications preferring a compromise of the two, the hierarchical-Hamming-loss-minimizing HIROM may be more desirable.

5.6 Prediction Time Comparison

In this section, we compare the prediction efficiency of the various classifiers with respect to hierarchical Hamming loss. For all results, all standard deviations are smaller than 0.001, thus we only show the mean values. The results are shown in Tables 8 and 9. As can be seen, though HIROM

TABLE 6
Hierarchical Ranking Loss Values on Data Sets with Tree Label Hierarchies

data set	HIROM	CSSA	H-SVM	B-SVM	CLUS-HMC
rcv1-subset1	0.076 ± 0.002	0.202 ± 0.009	**0.075** ± 0.002	0.230 ± 0.006	0.217 ± 0.006
rcv1-subset2	**0.075** ± 0.003	0.199 ± 0.007	**0.075** ± 0.002	0.226 ± 0.009	0.216 ± 0.006
rcv1-subset3	0.075 ± 0.002	0.083 ± 0.003	**0.074** ± 0.003	0.232 ± 0.006	0.213 ± 0.004
rcv1-subset4	0.078 ± 0.006	0.199 ± 0.013	**0.073** ± 0.004	0.229 ± 0.004	0.213 ± 0.008
rcv1-subset5	0.081 ± 0.003	0.195 ± 0.006	**0.075** ± 0.003	0.233 ± 0.004	0.233 ± 0.004
enron	**0.097** ± 0.014	**0.097** ± 0.014	0.099 ± 0.014	0.389 ± 0.028	0.328 ± 0.008
eurolex-dc	**0.058** ± 0.011	0.162 ± 0.008	0.059 ± 0.001	0.657 ± 0.021	-
seq	**0.227** ± 0.007	0.478 ± 0.012	0.239 ± 0.008	0.996 ± 0.014	0.384 ± 0.018
pheno	**0.266** ± 0.006	0.567 ± 0.031	0.275 ± 0.009	1.113 ± 0.032	0.497 ± 0.011
struc	**0.251** ± 0.007	0.557 ± 0.019	0.258 ± 0.008	1.073 ± 0.023	0.337 ± 0.015
hom	**0.213** ± 0.005	0.454 ± 0.008	0.222 ± 0.006	0.941 ± 0.017	0.346 ± 0.010
cellcycle	**0.246** ± 0.005	0.340 ± 0.005	0.255 ± 0.005	1.056 ± 0.022	0.414 ± 0.011
church	**0.261** ± 0.018	0.336 ± 0.011	0.268 ± 0.016	1.123 ± 0.021	0.412 ± 0.020
derisi	**0.260** ± 0.005	0.334 ± 0.004	0.267 ± 0.005	1.116 ± 0.017	0.422 ± 0.009
eisen	**0.240** ± 0.011	0.317 ± 0.015	0.254 ± 0.013	0.982 ± 0.019	0.396 ± 0.022
gasch1	**0.239** ± 0.003	0.322 ± 0.004	0.251 ± 0.005	1.018 ± 0.021	0.391 ± 0.009
gasch2	**0.250** ± 0.022	0.337 ± 0.027	0.258 ± 0.024	1.034 ± 0.057	0.414 ± 0.007
spo	**0.255** ± 0.006	0.574 ± 0.003	0.263 ± 0.006	1.102 ± 0.018	0.413 ± 0.006
expr	**0.232** ± 0.006	0.315 ± 0.002	0.243 ± 0.007	0.955 ± 0.026	0.387 ± 0.014

The results with best mean values are in bold.

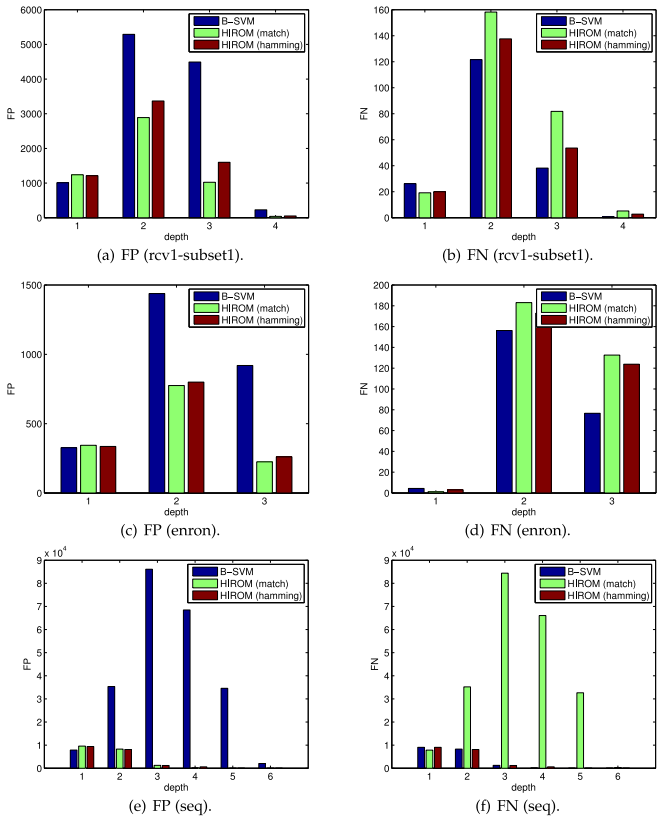


Fig. 5. FP and FN obtained by the B-SVM (minimizing the H-loss), HIROM (minimizing the matching loss) and HIROM (minimizing the hierarchical Hamming loss) at different levels of the hierarchy.

takes more time to performs testing than other methods, it finishes the testing of a sample in less than 0.02 second per sample on average for tree data sets and 0.04 second per sample on average for DAG data sets, which is still very efficient. Thus, considering both classification performance and testing efficiency, HIROM is most encouraged to use.

TABLE 8
Prediction Time per Instance (in Seconds) on Data Sets with Tree Label Hierarchies

data set	HIROM	CSSA	H-SVM	B-SVM	CLUS-HMC
rcv1-subset1	0.006	0.002	0.000	0.000	0.000
rcv1-subset2	0.004	0.002	0.000	0.001	0.000
rcv1-subset3	0.006	0.002	0.000	0.000	0.000
rcv1-subset4	0.006	0.002	0.000	0.000	0.000
rcv1-subset5	0.006	0.002	0.000	0.000	0.000
enron	0.005	0.000	0.000	0.000	0.002
eurolex-dc	0.012	0.006	0.000	0.000	-
seq	0.017	0.006	0.000	0.000	0.005
pheno	0.017	0.006	0.000	0.000	0.005
struc	0.016	0.006	0.000	0.000	0.004
hom	0.017	0.006	0.000	0.000	0.005
cellcycle	0.016	0.006	0.000	0.000	0.004
church	0.016	0.006	0.000	0.002	0.005
derisi	0.019	0.006	0.000	0.002	0.004
eisen	0.016	0.006	0.000	0.000	0.005
gasch1	0.018	0.006	0.000	0.000	0.005
gasch2	0.018	0.006	0.000	0.000	0.004
spo	0.017	0.006	0.000	0.000	0.004
expr	0.020	0.007	0.000	0.000	0.004

TABLE 9
Prediction Time per Instance (in Seconds) on Data Sets with DAG Label Hierarchies

data set	HIROM	CSSAG	H-SVM	CLUS-HMC
seq	0.032	0.017	0.000	0.004
pheno	0.016	0.038	0.001	0.004
struc	0.033	0.011	0.001	0.004
hom	0.029	0.015	0.000	0.002
cellcycle	0.040	0.014	0.001	0.004
church	0.040	0.014	0.000	0.002
derisi	0.038	0.012	0.000	0.004
eisen	0.035	0.011	0.000	0.002
gasch1	0.035	0.012	0.001	0.004
gasch2	0.032	0.014	0.001	0.003
spo	0.003	0.012	0.001	0.001
expr	0.035	0.011	0.000	0.003

6 Conclusion

In this paper, we compared various loss functions and extended the Hamming loss and ranking loss for hierarchical multilabel classification. Following the Bayesian decision theory, we developed Bayes-optimal classifiers that minimize the risks corresponding to these loss functions. Both the computation and minimization of the risk can be efficiently obtained without exhaustive enumeration of an exponential number of possible multilabels. Experimental results on a large number of real-world data sets with both tree-and DAG-structured label hierarchies demonstrate the superiority of the proposed Bayes-optimal classifier. In the future, we will study some theoretical guarantees for the proposed Bayes-optimal classifiers about how the errors in the probability estimates will transfer to errors and how much the hierarchy information used in these classifiers can help increase the performance over the non-risk-minimizing classifiers and flat classifiers.

Acknowledgments

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 614012).

References

[1] R. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, no. 2, pp. 135–168, 2000.

[2] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[3] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Mach. Learn.*, vol. 73, no. 2, pp. 185–214, 2008.

[4] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random K-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[5] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. New York, NY, USA: Springer, 2010, pp. 667–685.

[6] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[7] L. Sun, S. Ji, and J. Ye, *Multi-Label Dimensionality Reduction*. Boca Raton, FL, USA: CRC Press, 2013.
[8] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *J. Mach. Learn. Res.*, vol. 7, pp. 31–54, 2006.
[9] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Kernel-based learning of hierarchical multilabel classification models," *J. Mach. Learn. Res.*, vol. 7, pp. 1601–1626, 2006.
[10] A. Clare, "Machine learning and data mining for yeast functional genomics," Ph.D. dissertation, Univ. Wales, Cardiff, Wales, U.K., 2003.
[11] K. Punera, S. Rajan, and J. Ghosh, "Automatically learning document taxonomies for hierarchical classification," in *Proc. Int. Conf. World Wide Web*, Chiba, Japan, 2005, pp. 1010–1011.
[12] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, Washington, DC, USA, 2010, pp. 999–1008.
[13] W. Bi and J. Kwok, "Multi-label classification on Tree- and DAG-structured hierarchies," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 17–24.
[14] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proc. Int. Conf. Mach. Learn.*, Banff, Alberta, Canada 2004, pp. 27–34.
[15] D. Zhou, L. Xiao, and M. Wu, "Hierarchical classification via orthogonal transfer," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 801–808.
[16] J. McAuley, A. Ramisa, and T. Caetano, "Optimization of robust loss functions for weakly-labeled image taxonomies," *Int. J. Comput. Vis.*, vol. 104, pp. 343–361, 2012.
[17] C. Silla and A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discovery*, vol. 22, nos. 1/2, pp. 1–42, Jan. 2010.
[18] R. Cerri, R. C. Barros, and A. C. De Carvalho, "Hierarchical multi-label classification using local neural networks," *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 39–56, 2014.
[19] Z. Barutcuoglu and O. Troyanskaya, "Hierarchical Multi-label prediction of gene function," *Bioinformatics*, vol. 22, pp. 830–836, 2006.
[20] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.
[21] B. Hariharan, L. Zelnik-Manor, S. Vishwanathan, and M. Varma, "Large scale max-margin multi-label classification with priors," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 423–430.
[22] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 309–316.
[23] K. Dembczynski, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 279–286.
[24] S. Bucak, R. Jin, and A. Jain, "Multi-label learning with incomplete class assignments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 2801–2808.
[25] S. Nowak, H. Lukashevich, P. Dunker, and S. Rüger, "Performance measures for multilabel evaluation: A case study in the area of image classification," in *Proc. Int. Conf. Multimedia Inf. Retrieval*, 2010, pp. 35–44.
[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, no. 2, pp. 1453, 2005.
[27] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.
[28] J. Petterson and T. Caetano, "Reverse multi-label learning," in *Proc. Adv. Neural Inf. Process. Syst. 23*, 2010, pp. 1912–1920.
[29] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "An exact algorithm for F-measure maximization," in *Proc. Adv. Neural Inf. Process. Syst. 24*, 2011, pp. 223–230.
[30] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Hierarchical classification: Combining Bayes with SVM," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 177–184.
[31] W. Bi and J. Kwok, "Hierarchical multilabel classification with minimum Bayes risk," in *Proc. Int. Conf. Data Mining*, Brussels, Belgium, 2012, pp. 101–110.
[32] H. Lo, S. Lin, and H. Wang, "Generalized K-labelsets ensemble for Multi-label and Cost-sensitive classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1679–1691, Jul. 2014.
[33] X. Kong, M. Ng, and Z.-H. Zhou, "Transductive multilabel learning via label set propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 704–719, Mar. 2013.
[34] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2004, pp. 78–87.
[35] N. Cesa-Bianchi and G. Valentini, "Hierarchical cost-sensitive algorithms for genome-wide gene function prediction," in *Proc. Int. Workshop Mach. Learn. Syst. Biol.*, Ljubljana, Slovenia, 2009, pp. 14–29.
[36] K. Verspoor, J. Cohn, S. Mnizewski, and C. Joslyn, "A categorization approach to automated ontological function annotation," *Protein Sci.*, vol. 15, pp. 1544–1549, 2006.
[37] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni, "Incremental algorithms for hierarchical classification," in *Proc. Adv. Neural Inf. Process. Syst. 17*, 2005, pp. 233–240.
[38] J. Zaragoza, L. Sucar, and E. Morales, "Bayesian chain classifiers for multidimensional classification," in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Catalonia, Spain, 2011, pp. 2192–2197.
[39] W. Bi and J. Kwok, "Efficient Multi-label classification with many labels," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 405–413.
[40] T. Zhou, D. Tao, and X. Wu, "Compressed labeling on distilled labelsets for Multi-label learning," *Mach. Learn.*, vol. 88, no. 1-2, pp. 69–126, 2012.
[41] W. Bi and J. Kwok, "Mandatory leaf node prediction in hierarchical multilabel classification," in *Proc. Adv. Neural Inf. Process. Syst. 25*, 2012, pp. 153–161.
[42] D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.
[43] B. Klimt and Y. Yang, "The Enron corpus: A new dataset for email classification research," in *Proc. Eur. Conf. Mach. Learn.*, Pisa, Italy, 2004, pp. 217–226.
[44] E. Mencía and J. Fürnkranz, "Efficient pairwise multilabel classification for Large-scale problems in the legal domain," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Antwerp, Belgium, 2008, pp. 50–65.
[45] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.

Wei Bi received the bachelor's degree in computer science from the Sun Yat-Sen University in 2010. Currently, she is working toward the PhD degree in computer science at the Hong Kong University of Science and Technology. Her research interests focus on machine learning, data mining, application problems on computer vision and other problems in artificial intelligence. She received the Google PhD fellowship in machine learning in 2013, and Google Anita Borg scholarship in 2014.

James T. Kwok received the PhD degree in computer science from the Hong Kong University of Science and Technology in 1996. He was with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, as an assistant professor. He is currently a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include kernel methods, machine learning, example recognition, and artificial neural networks. He received the IEEE Outstanding 2004 Paper Award, and the Second Class Award in Natural Sciences by the Ministry of Education, People's Republic of China, in 2008. He has been a program cochair for a number of international conferences, and served as an associate editor for the *IEEE Transactions on Neural Networks and Learning Systems* from 2006-2012. Currently, he is an associate editor for the *Neurocomputing* journal.