Journal of Machine Learning Research 12 (2011) 2649-2649 Submitted 8/09; Revised 4/11; Published 9/11

Large Margin Hierarchical Classification with Mutually Exclusive Class Membership

Huixin Wang HXWANG@STAT.UMN.EDU
Xiaotong Shen XSHEN@STAT.UMN.EDU
School of Statistics
University of Minnesota
Minneapolis, MN 55455

Wei Pan WEIP@BIOSTAT.UMN.EDU
Division of Biostatistics
University of Minnesota
Minneapolis, MN 55455

Nicolo-Cesa Bianchi

Abstract

In hierarchical classification, class labels are structured, that is each label value corresponds to one non-root node in a tree, where the inter-class relationship for classification is specified by directed paths of the tree. In such a situation, the focus has been on how to leverage the inter-class relationship to enhance the performance of flat classification, which ignores such dependency. This is critical when the number of classes becomes large relative to the sample size. This paper considers single-path or partial-path hierarchical classification, where only one path is permitted from the root to a leaf node. A large margin method is introduced based on a new concept of generalized margins with respect to hierarchy. For implementation, we consider support vector machines and ψ-learning. Numerical and theoretical analyses suggest that the proposed method achieves the desired objective and compares favorably against strong competitors in the literature, including its flat counterparts. Finally, an application to gene function prediction is discussed.

Keywords: difference convex programming, gene function annotation, margins, multi-class classification, structured learning

1. Introduction

In many applications, knowledge is organized and explored in a hierarchical fashion. For instance, in one of the central problems in modern biomedical research—gene function prediction, biological functions of genes are often organized by a hierarchical annotation system such as MIPS (the Munich Information Center for Protein Sequences, Mewes et al., 2002)for yeast S. *cerevisiae*. MIPS is structured hierarchically, with upper-level functional categories describing more general information concerning biological functions of genes, while low-level ones refer to more specific and detailed functional categories. A hierarchy of this sort presents the current available knowledge. To predict unknown gene functions, a gene is classified, through some predictors, into one or more gene functional categories in the hierarchy of MIPS, forming novel hypotheses for confirmatory biological experiments (Hughes et al., 2000). Classification like this is called hierarchical classification, which has been widely used in webpage classification and document categorization. Hierarchical classification involves inter-class dependencies specified by a prespecified hierarchy, which is unlike

multiclass classification where class membership is mutually exclusive for all classes. The primary objective of hierarchical classification is leveraging inter-class relationships to enhance multiclass classification ignoring such dependencies, known as flat classification. This is particularly critical in high-dimensional problems with a large number of classes in classification. To achieve the desired objective, this paper develops a large margin approach for single-path or partial-path hierarchical classification with hierarchy defined by a tree.

Hierarchical classification, an important subject which has not yet received much attention, can be thought of as nested classification within the framework of multiclass classification. One major challenge is how to formulate a loosely defined hierarchical structure into classification to achieve higher generalization performance, which, otherwise, is impossible for flat classification, especially in a high-dimensional situation. Three major approaches have been proposed in the literature. The first is the so called "flat approach", which ignores the hierarchical structure. Recent studies suggest that higher classification accuracy results can be realized by incorporating the hierarchical structure (Dekel et al., 2004). Relevant references can be found in Yang and Liu (1999) for nearest neighbor, Lewis (1998) for naive Bayes, Joachims (1998) for support vector machines (SVM, Boser et al., 1992; Vapnik, 1998), among others. The second is the sequential approach, where a multiclass classifier is trained locally at each parent node of the hierarchy. As a result, the classifier may be not well trained due to a small training sample locally and lack of global comparisons. Further investigations are necessary with regard to how to use the given hierarchy in classification to improve the predictive performance, as noted in Dekel et al. (2004) and Cesa-Bianchi et al. (2006). The third is the promising structured approach, which recognizes the importance of a hierarchical structure in classification. Shahbaba and Neal (2007) proposed a Bayesian method through a constrained hierarchical prior and a Markov Chain Monte Carlo implementation. Cai and Hofmann (2004) and Rousu et al. (2006) employed structured linear and kernel representations and loss functions defined by a tree, together with loss-weighted multiclass SVM, whereas Dekel et al. (2004) developed a batch and on-line version of loss-weighted hierarchical SVM, and Cesa-Bianchi et al. (2006) developed sequential training based SVM with certain hierarchical loss functions. The structured approach uses a weighted loss defined by a hierarchy, such as the symmetric difference loss and a sub-tree H-loss, see, for instance, Cesa-Bianchi et al. (2006), as opposed to the conventional 0-1 loss, then maximizes the loss-weighted margins for a multiclass SVM, as described in Lin et al. (2002). Ensembles of nested dichotomies in Dong et al. (2005) and Zimek et al. (2008) have achieved good performance. Despite progress, issues remain with respect to how to fully take into account a hierarchical structure and to what role the hierarchy plays.

To meet the challenge, this article develops a large margin method for hierarchical classification, based on a new concept of structured functional and geometric margins defined for each node of the hierarchy, which differs from the concept of the loss-weighted margins in structured prediction. This concept of margins with respect to hierarchy is designed to account for inter-class dependencies in classification. As a result, the complexity of the classification problem reduces, translating into higher generalization accuracy of classification. Our theory describes when this will occur, depending on the structure of a tree hierarchy. In contrast to existing approaches, the proposed method trains a classifier globally while making sequential nested partitions of classification regions. The proposed method is implemented for support vector machines (SVM, Boser et al., 1992) and ψ-learning (Shen et al., 2003) through quadratic and difference convex (DC) programming.

To examine the proposed method's generalization performance, we perform simulation studies. They indicate that the proposed method achieves higher performance than three strong competitors.

A theoretical investigation confirms that the empirical performance is indeed attributed to a reduced size of the function space for classification, as measured by the metric entropy, through effective use of a hierarchical structure. In fact, stronger inter-class relations tend to lead to better performance over its flat counterpart. In conclusion, both the numerical and theoretical results suggest that a tree hierarchical structure has been incorporated into classification for generalization.

This article is organized as follows. Section 2 formulates the problem of hierarchical classification. Section 3 introduces the proposed method and develops computational tools. Section 4 performs simulation studies and presents an application of the proposed method in gene function prediction. Section 5 is devoted to theoretical investigation of the proposed method and to the study of the role of a hierarchical structure in classification. Section 6 discusses the method, followed by technical details in the Appendix.

2. Single-path and Partial-path Hierarchical Classification

In single-path or partial-path hierarchical classification, input $\boldsymbol{X} = (X_1, \cdots, X_q) \in S \subset \mathbb{R}^q$ is a vector of q covariates, and we code output $Y \in \{1, \cdots, K\}$, corresponding to non-root nodes $\{1, \cdots, K\}$ in a rooted tree $\mathcal{H}$, a graph with nodes connected by directed paths from the root 0, where directed edge $i \to j$ specifies a parent-child relationship from i to j. Here Y is structured in that $i \to j$ in $\mathcal{H}$ induces a subset relation between the corresponding classes i and j in classification, that is, the classification region of class j is a subset of that of class i. As a result, direct and indirect relations among nodes over $\mathcal{H}$ impose an inter-class relationship among K classes in classification.

Before proceeding, we introduce some notations for a tree $\mathcal{H}$ with k leaves and $(K-k)$ non-leaf-nodes, where a non-leaf node is an ancestor of a leaf one. Denote by $|\mathcal{H}|$ the size of $\mathcal{H}$. For each $t \in \{1, \cdots, K\}$, define $par(t)$, $chi(t)$ $sib(t)$, $anc(t)$ and $sub(t)$ to be sets of its parent(s) (immediate ancestor), its children (immediate offsprings), its siblings (nodes sharing the same parent with node t), its ancestors (immediate or remote), and the subtree rooted from t, respectively. Throughout this paper, $par(t)$, $chi(t)$ and $sib(t)$ are allowed to be empty. Assume, without loss of generality, that $|par(t)| = 1$ for non-root node t because multiple parents are not permitted for a tree. Also we define $\mathcal{L}$ to be the set of leaves of $\mathcal{H}$.

To classify $\boldsymbol{x}$, a decision function vector $\boldsymbol{f} = (f_1, \cdots, f_K) \in \mathcal{F} = \prod_{j=1}^{K} \mathcal{F}_j$ is introduced, where $f_j(\boldsymbol{x})$; $j = 1, \cdots, K$, mapping from $\mathbb{R}^q$ onto $\mathbb{R}^1$, represents class j and mimics $P(Y = j | \boldsymbol{X} = \boldsymbol{x})$. Then $\boldsymbol{f}$ is estimated through a training sample $Z_i = (\boldsymbol{X}_i, Y_i)_{i=1}^n$, independent and identically distributed according to an unknown probability $P(\boldsymbol{x}, y)$. To assign $\boldsymbol{x}$, we introduce a top-down decision rule $d^H(\boldsymbol{f}(\boldsymbol{x}))$ with respect to $\mathcal{H}$ through $\boldsymbol{f}$. From the top to the bottom, we go through each node j and assign $\boldsymbol{x}$ to one of its children $l = \text{argmax}_{t \in chi(j)} f_t(\boldsymbol{x})$ having the highest value among f_t's for $t \in chi(j)$ when $j \notin \mathcal{L}$, and assign $\boldsymbol{x}$ to j otherwise.

This top-down rule is sequential, and yields mutually exclusive membership for sibling classes. In particular, for each parent j, $chi(j)$ gives a partition of the classification region of parent class j. This permits an observation staying at a parent when one child of the parent is defined as itself, see, for example, the node labeled 03.01 in Figure 3, which is a case of partial-path hierarchical classification.

Finally, a classifier is constructed through $d^H(\cdot)$ to have small generalization error $El_{0-1}(Y, d^H(\boldsymbol{f}(\boldsymbol{X}))$, with $l_{0-1}(Y, d^H(\boldsymbol{f}(\boldsymbol{X})) = I(Y \neq d^H(\boldsymbol{f}(\boldsymbol{X}))$ the 0-1 hierarchical loss.

3. Proposed Method

In the existing literature on hierarchical classification, the margins are defined by the conventional unstructured margins for multiclass classification, for instance, the loss-weighted hierarchical SVM of Cai and Hofmann (2004), denoted as HSVM$_c$. For unstructured margins in classification, a certain number of pairwise comparisons is required, which is the same as conventional multiclass classification. In what follows, we propose a new framework using a given hierarchy to define margins, leading to a reduced number of pairwise comparisons for hierarchical classification.

3.1 Margins with Respect to $\mathcal{H}$

We first explore a connection between classification and function comparisons, based on the concept of generalized functional margins with respect to a hierarchy is introduced. Over a hierarchy $\mathcal{H}$, the top-down rule $d^H(\boldsymbol{f}(\boldsymbol{x}))$ is employed for classification. To classify, comparing some components of $\boldsymbol{f}$ at certain relevant nodes in $\mathcal{H}$ is necessary, which is in a parallel fashion as in multiclass classification. Consider leaf node 4 in the tree $\mathcal{H}$ described in Figure 2 (c). There $f_4 - f_3$ and $f_6 - f_5$ need to be compared against 0 to classify at node 4 through the top-down rule, that is, $\min(f_4 - f_3, f_6 - f_5)$ is less than 0 or not, which leads to our margin definition for $(\boldsymbol{x}, y = 4)$ $U(\boldsymbol{f}(\boldsymbol{x}), y = 4) = \min(f_4 - f_3, f_6 - f_5)$. More generally, we define set $U(\boldsymbol{f}(\boldsymbol{x}), y)$, for $y \in \{1, \cdots, K\}$ to be $\{f_t - f_j : j \in sib(t), t \in anc(y) \cup \{y\}\} = \{u_{y,1}, u_{y,2}, \cdots, u_{y,k_y}\}$ with k_y elements. This set compares any class t against sibling classes defined by $sib(t)$ for y and any of its ancestors t, permitting hierarchical classification at any location of $\mathcal{H}$ and generating a single-path or partial-path from the root to the node corresponding to class y.

For classification evaluation, we define the generalized functional margin with respect to $\mathcal{H}$ for $(\boldsymbol{x}, y)$ as $u_{min}(\boldsymbol{f}(\boldsymbol{x}), y) = \min\{u_{y,j} : u_{y,j} \in U(\boldsymbol{f}(\boldsymbol{x}), y)\}$. In light of the result of Lemma 1, this quantity is directly related to the generalization error, which summarizes the overall error in hierarchical classification as the 0-1 loss in binary classification. That is, a classification error occurs if and only if $u_{min}(\boldsymbol{f}(\boldsymbol{x}), y) < 0$. Moreover, *this definition reduces to that of multiclass margin classification of Liu and Shen (2006) when no hierarchical structure is imposed.* In contrast to the definition of multiclass classification, the number of comparisons required for classification over a tree $\mathcal{H}$ is usually smaller, owing to the fact that only siblings need to be compared through the top-down rule, as opposed to comparisons of all pairs of classes in multiclass classification.

Lemma 1 establishes a key connection between the generalization error and our definition of $u_{min}(f(X), Y)$.

Lemma 1 *With $I(\cdot)$ denoting the indicator function,*

$$GE(d) = El_{0-1}(Y, d(\boldsymbol{X})) \equiv EI(Y \neq d(\boldsymbol{X})) = EI(u_{min}(f(X), Y) < 0),$$

where l_{0-1} is the 0-1 loss in hierarchical classification, and $I(\cdot)$ is the indicator function.

This lemma says that a classification error occurs for decision function $\boldsymbol{f}$ and an observation $(\boldsymbol{x}, y)$, if and only if the functional margin $u_{min}(\boldsymbol{f}(\boldsymbol{x}), y)$ is negative.

3.2 Cost Function and Geometric Margin

To achieve our objective of constructing classifier $d^H(\hat{\boldsymbol{f}}(\boldsymbol{x}))$ having small generalization error, we construct a cost function to yield an estimate $\hat{\boldsymbol{f}}$ for $d^H(\hat{\boldsymbol{f}}(\boldsymbol{x}))$. Ideally, one may minimize the

empirical generalization error $n^{-1}\sum_{i=1}^{n} I(u_{min}(f(X_i),Y_i)<0)$ based on $(\boldsymbol{X}_i,Y_i)_{i=1}^{n}$. However, it is computationally infeasible because of discontinuity of $I(\cdot)$. For this reason, we replace $I(\cdot)$ by a surrogate loss $v(\cdot)$ to use the existing two-class surrogate losses in hierarchical classification. In addition to computational benefits, certain loss functions $v(\cdot)$ may also lead to desirable large margin properties (Zhu and Hastie, 2005). Given functional margin $u = u_{min}(\boldsymbol{f}(\boldsymbol{x}),y)$, we say that a loss $v(\cdot)$ is a margin loss if it can be written as a function of u. Moreover, it is a large margin if $v(u)$ is nonincreasing in u. Most importantly, $v(u_{min}(\boldsymbol{f}(\boldsymbol{x}),y))$ yields Fisher-consistency in hierarchical classification, which constitutes a basis of studying the generalization error in Section 5. Note that in the two-class case a number of margin losses have been proposed. Convex margin losses are the hinge loss $v(u) = (1-u)_+$ for SVM and the logistic loss $v(u) = \log(1+e^{-u})$ for logistic regression (Zhu and Hastie, 2005). Nonconvex large margin losses include, for example ψ-loss $v(u) = \psi(u)$ for ψ-learning, with $\psi(u) = 1 - \text{sign}(u)$ and $\text{sign}(u) = I(u > 0)$, if $u \geq 1$ or $u < 0$, and $1-u$ otherwise (Shen et al., 2003).

Placing a margin loss $v(\cdot)$ in the framework of penalization, we propose our cost function for hierarchical classification:

$$s(\boldsymbol{f}) = C\sum_{i=1}^{n} v(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i),y_i)) + J(\boldsymbol{f}), \tag{1}$$

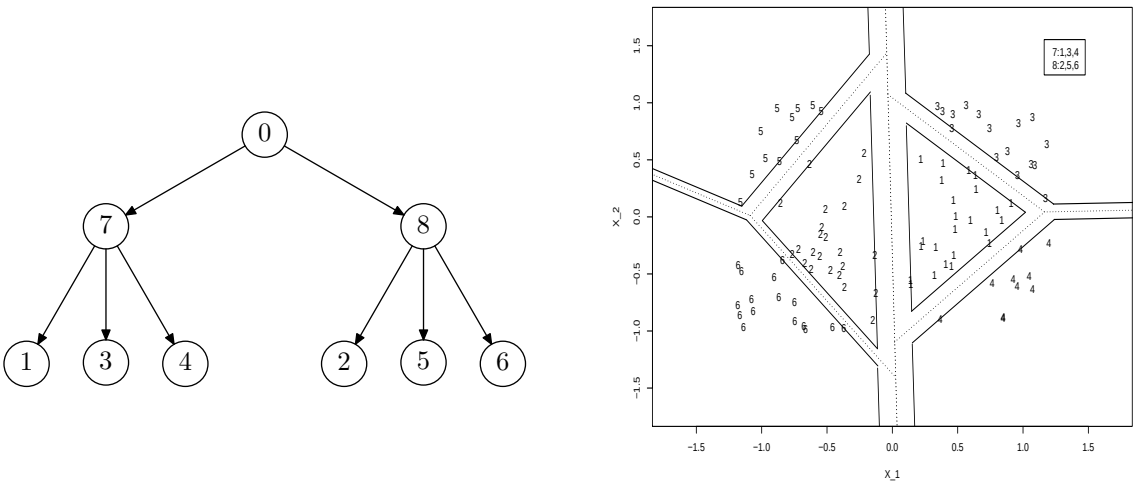
subject to sum to zero constraints $\sum_{\{t\in sib(j)\cup\{j\}\}} f_t(\boldsymbol{x}) = 0;\ \forall j = 1\cdots,K, sib(j)\neq\varnothing, \boldsymbol{x}\in S$, the domain of X_1, for removing redundancy among the components of $\boldsymbol{f}$. For example, for the tree $\mathcal{H}$ in Figure 2 (c), three constraints are imposed: $f_1+f_2=0$, $f_3+f_4=0$ and $f_5+f_6=0$, for three pairs of siblings. In (1), penalty $J(\boldsymbol{f})$ is the inverse geometric margin to be introduced, and $C>0$ is a tuning parameter regularizing the trade-off between minimizing $J(\boldsymbol{f})$ and minimizing training error. Minimizing (1) with respect to $\boldsymbol{f}\in\mathcal{F}$, a candidate function space, yields an estimate $\hat{\boldsymbol{f}}$, thus classifier $d^H(\hat{\boldsymbol{f}}(\boldsymbol{x}))$. Note that *(1) reduces to that of multiclass margin classification of Liu and Shen (2006) when no hierarchical structure is specified.*

To introduce the geometric margin with respect to $\mathcal{H}$ in the L_2-norm, (with other norms applied similarly), consider a generic vector of functions $\boldsymbol{f}$: $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T\tilde{\boldsymbol{x}} + b_j;\ j = 1,\cdots,K$, with $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ and $\tilde{\boldsymbol{x}} = (\mathcal{K}(\boldsymbol{x}_1,\cdot),\cdots,\mathcal{K}(\boldsymbol{x}_n,\cdot))^T$ for linear and kernel learning. The geometric margin is defined as $\min_{\{(t,j):t\in sib(j)\}}\gamma_{j,t}$, where $\gamma_{j,t} = \frac{2}{\|\boldsymbol{w}_j-\boldsymbol{w}_t\|^2_{\mathcal{K}}}$ is the usual separation margin defined for classes j versus $t\in sib(j)$, representing the vertical distance between two parallel hyperplanes $f_j - f_t = \pm 1$ (Shen and Wang, 2007). Here $\|\boldsymbol{w}_j\|^2_{\mathcal{K}}$ is $\|\boldsymbol{w}_j\|^2$ in the linear case and is $\boldsymbol{w}_j^T\mathcal{K}\boldsymbol{w}_j$ in the kernel case with $\mathcal{K}$ being an $n\times n$ kernel matrix. Note that the other form of the margin in the L_p-norm (with $1\leq p\leq\infty$) can be defined similarly. Ideally, $J(\boldsymbol{f})$ is $\max_{\{(t,j):t\in sib(j)\}}\gamma_{j,t}^{-1} = \max_{\{(t,j):t\in sib(j)\}}\frac{\|\boldsymbol{w}_j-\boldsymbol{w}_t\|^2}{2}$, the inverse of the geometric margin. However, it is less tractable numerically. Practically, we work with its upper bound $J(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2_{\mathcal{K}}$ instead.

For hierarchical classification, (1) yields different classifiers with different choices of margin loss $v(\cdot)$. Specifically, (1) covers multiclass SVM and ψ-learning of Liu and Shen (2006), with equal cost when all the leaf nodes share the same parent—the root, which are called SVM and ψ-learning in what follows.

3.3 Classification and Hierarchy $\mathcal{H}$

The hierarchical structure specified by $\mathcal{H}$ is summarized as the direct parent-child relation and the associated indirect relations, for classification. They are integrated into our framework. Whereas



(a) Hierarchical structure

(b) Geometric margin

Figure 1: Plot of generalized geometric margin with respect to $\mathcal{H}$ in (b), defined by a tree in (a). Classification processes sequentially with a partition of classes 7 and 8 at the top level, and a further partition of class 7 into classes 1,3 and 4, and that of class 8 into classes 3,5 and 6, where classification boundaries are displayed by dotted lines. Geometric margin is defined as the minimal vertical distances between seven pairs of solid parallel lines, representing separations between classes 7 and 8, 2 and 5, 2 and 6, 5 and 6, 1 and 3, 1 and 4, and 3 and 4.

the top-down rule is specified by $\mathcal{H}$, $u_{min}(\boldsymbol{f}(\boldsymbol{x}),y)$ captures the relations through (1). As a result, a problem's complexity is reduced when classification is restricted to $\mathcal{H}$, leading to higher generalization accuracy. This aspect will be confirmed by the numerical results in Section 4, and by a comparison of the generalization errors between hierarchical SVM (HSVM) and hierarchical ψ-learning (HPSI) against their flat counterparts—SVM and ψ-learning in Section 5.

3.4 Minimization

We implement (1) in a generic form: $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T \tilde{\boldsymbol{x}} + b_j$; $j = 1, \cdots, K$. Note that the sum-to-zero constraints may be infinite, which occurs when the domain of x has infinitely many values. To overcome this difficulty, we derive Theorem 1, which says that reinforcement of the sum-to-zero constraints for (1) suffices at the observed data instead of all possible x-values.

Theorem 1 *Assume that* $\{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \cdots, \tilde{\boldsymbol{x}}_n\}$ *spans* $\mathbb{R}^q$. *Then, for* $j = 1, \cdots, K$, *minimizing (1) subject to* $\sum_{\{t:t\in sib(j)\cup\{j\}\}} f_j(\boldsymbol{x}) = 0$; $\forall j = 1 \cdots, K, sib(j) \neq \varnothing, \boldsymbol{x} \in S$, *is equivalent to minimizing (1) subject to* $\sum_{\{t:t\in sib(j)\cup\{j\}\}} f_j(\boldsymbol{x}_i) = 0$; $\forall j = 1 \cdots, K, sib(j) \neq \varnothing, i = 1, \cdots, n$.

Based on Theorem 1, minimizing (1) is equivalent to

$$\text{minimizing} s(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{n} v(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i), y_i)), \tag{2}$$

subject to $\sum_{\{t:t\in sib(j)\cup\{j\}\}} f_t(\boldsymbol{x}_i) = 0$; $i = 1,\cdots,n$, $j = 1,\cdots,K$, $sib(j) \neq \varnothing$.

Subsequently, we work with (2), where the proposed classifiers are denoted by HSVM and HPSI when $v(u) = (1-u)_+$ and $v(u) = \psi(u)$, respectively. In the first case, HSVM is solved by quadratic programming (QP), see Appendix B. In the second case, (2) for HPSI is solved by DC programming, to be described next.

For HPSI, we decompose $s(\boldsymbol{f})$ in (2) with $v(u) = \psi(u)$ into a difference of two convex functions: $s(\boldsymbol{f}) = s_1(\boldsymbol{f}) - s_2(\boldsymbol{f})$, where $s_1(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{n}\psi_1(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i), y_i))$ and $s_2(\boldsymbol{f}) = C\sum_{i=1}^{n}\psi_2(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i), y_i))$, derived from a DC decomposition of $\psi = \psi_1 - \psi_2$, with $\psi_1(u) = (1-u)_+$ and $\psi_2(u) = (-u)_+$. Through our DC decomposition, a sequence of upper approximations of $s(\boldsymbol{f})$ $s_1(\boldsymbol{f}) - \langle \boldsymbol{f} - \hat{\boldsymbol{f}}^{(m-1)}, \nabla s_2(\hat{\boldsymbol{f}}^{(m-1)})\rangle_{\mathcal{K}}$ is constructed iteratively, where $\langle\cdot,\cdot\rangle_{\mathcal{K}}$ is the inner product with respect to kernel $\mathcal{K}$ and $\nabla s_2(\hat{\boldsymbol{f}}^{(m-1)})$ is a gradient vector of $s_2(\boldsymbol{f})$ at the solution $\hat{\boldsymbol{f}}^{(m-1)}$ at iteration $m-1$, defined as a sum of partial derivatives of s_2 over each observation, with $\nabla\psi_2(u) = 0$ when $u > 0$ and $\nabla\psi_2(u) = -1$ otherwise. Note that $s_1(\boldsymbol{f}) - \langle \boldsymbol{f} - \hat{\boldsymbol{f}}^{(m)}, \nabla s_2(\hat{\boldsymbol{f}}^{(m)})\rangle_{\mathcal{K}}$ is a convex upper bound of $s(\boldsymbol{f})$ by convexity of s_2. Then the upper approximation $s_1(\boldsymbol{f}) - \langle \boldsymbol{f} - \hat{f}^{(m-1)}, \nabla s_2(\hat{\boldsymbol{f}}^{(m-1)})\rangle_{\mathcal{K}}$ is minimized to yield $\hat{\boldsymbol{f}}^{(m)}$. This is called a DC method for non-convex minimization in the global optimization literature (An and Tao, 1997).

To design our DC algorithm, starting from an initial value $\hat{\boldsymbol{f}}^{(0)}$, the solution of HSVM, we solve primal problems iteratively. At the mth iteration, we compute

$$\hat{\boldsymbol{f}}^{(m)} = \underset{\boldsymbol{f}}{\text{argmin}}(s_1(\boldsymbol{f}) - \langle \boldsymbol{f}, \nabla s_2(\hat{\boldsymbol{f}}^{(m-1)})\rangle_{\mathcal{K}}), \tag{3}$$

subject to $\sum_{\{t:t\in sib(j)\cup\{j\}\}} f_t(\boldsymbol{x}_i) = 0$; $i = 1,\cdots,n$, $j = 1,\cdots,K$, $sib(j) \neq \varnothing$, through QP and its dual form in Appendix B. The above iterative process continues until a termination criterion is met: $|s(\hat{\boldsymbol{f}}^{(m)}) - s(\hat{\boldsymbol{f}}^{(m-1)})| \leq \varepsilon$, where is $\varepsilon > 0$ is a prespecified tolerance precision. The final estimate $\hat{\boldsymbol{f}}$ is the best solution among $\hat{\boldsymbol{f}}^{(m)}$ over m.

The above algorithm terminates, and its speed of convergence is superlinear, by Theorem 3 of Liu et al. (2005) for ψ-learning. A DC algorithm usually leads to a good local solution even when it is not global (An and Tao, 1997). In our DC decomposition, s_2 can be thought of correcting the bias due to convexity imposed by s_1 that is the cost function of HSVM, which assures that a good local solution or a global solution can be realized. More importantly, an ε-global minimizer can be obtained when the algorithm is combined with the branch-and-bound method, as in Liu et al. (2005). Due to computational consideration, we shall not seek the exact global minimizer.

3.5 Evaluation Losses and Test Errors with Respect to Hierarchy

In hierarchical classification, three types of losses have been proposed for measuring a classifier's performance with respect to $\mathcal{H}$, as a generalization of the 0-1 loss in two-class classification. In addition to $l_{0-1}(Y, d(\boldsymbol{X}))$, there are the symmetric difference loss $l_\Delta(Y, d(\boldsymbol{X}))$ (Tsochantaridis et al., 2004) and the H-loss $l_H(Y, d(\boldsymbol{X}))$ (Rousu et al., 2006; Cesa-Bianchi et al., 2004). As in Tsochantaridis et al. (2004); Rousu et al. (2006); Cesa-Bianchi et al. (2004, 2006), we use the 0-1 loss, symmetric difference loss and H-losses as performance measurements for our examples. Given a classifier $d(\boldsymbol{x})$, $l_\Delta(Y, d(\boldsymbol{X}))$ is $|anc(Y) \triangle anc(d(\boldsymbol{X}))|$, where $\triangle$ denotes the symmetric difference

of two sets. Here $l_H(Y, d(\boldsymbol{X})) = c_j$, with j the highest node yielding the disagreement between Y and $d(\boldsymbol{X})$ in a tree, ignoring any errors occurring at lower levels. In other words, it penalizes the disagreement at a parent while tolerating subsequent errors at offsprings. Two common choices of c_j's have been suggested, leading to the subtree based H-loss l_{sub} and the siblings based H-loss l_{sib}:

$$c_j = |sub(j)|/K;\ j = 1,\ldots,K, \tag{4}$$

$$c_0 = 1,\ c_j = c_{par(j)}/|sib(j) \cup \{j\}|;\ j = 1,\ldots,K. \tag{5}$$

A classifier's generalization performance is measured by the test error, defined as

$$TE(\boldsymbol{f}) = n_{test}^{-1} \sum_{i=1}^{n_{test}} l(Y_i, d^H(\boldsymbol{f}(\boldsymbol{X}_i))), \tag{6}$$

where n_{test} is the size of a test sample, and l is one of the four evaluation losses: l_{0-1}, l_Δ, l_{sib} with c_j's defined by (4) and l_{sub} with c_j's defined by (5). The corresponding test errors are denoted as TE_{0-1}, TE_Δ, TE_{sib} and TE_{sub}.

4. Numerical Examples

NEED TEXT HERE

4.1 Simulated Examples

This section applies HSVM and HPSI to three simulated examples, where they are compared against their flat counterparts—*k*-class SVM and *k*-class ψ-learning of Liu and Shen (2006), and two strong competitors—HSVM$_c$ and the sequential hierarchical SVM (SHSVM). For SHSVM, we train SVMs separately for each parent node, and use the top-down scheme to label the estimated classes. See Davies et al. (2007) for more details.

All numerical analyses are conducted in R version 2.1.1 for SVM, ψ-learning, HSVM, HPSI, HSVM$_c$ and SHSVM. In linear learning, $\mathcal{K}(x,y) = \langle x,y\rangle$. In Gaussian kernel learning, $\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|x_1 - x_2\|^2/\sigma^2)$ is used, where σ is the median of the inter-class distances between any two classes, see Jaakkola et al. (1999) for the binary case.

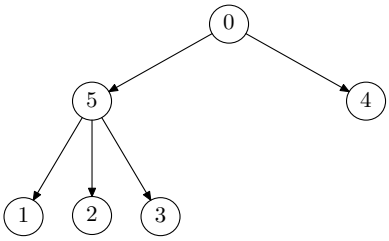
For comparison, we define the amount of improvement based on the test error. In simulated examples, the amount of improvement of a classifier is the percentage of improvement over SVM, in terms of the Bayesian regret:

$$\frac{(TE(\text{SVM}) - Bayes) - (TE(\cdot) - Bayes)}{(TE(\text{SVM}) - Bayes)},$$

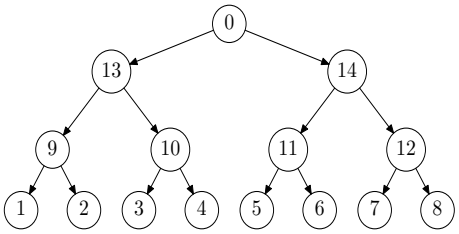
where $TE(\cdot)$ denotes the test error of a classifier, and *Bayes* denotes the Bayes error, which is the ideal optimal performance and serves as a benchmark for comparison. In a real data example where the Bayes rule is unavailable, the amount of improvement is

$$\frac{TE(\text{SVM}) - TE(\cdot)}{TE(\text{SVM})},$$

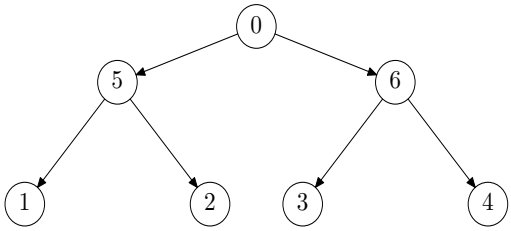
which may underestimate the actual percentage of improvement over SVM.



(a) Example 1



(b) Example 2



(c) Example 3

Figure 2: Hierarchies used in Examples 1, 2 and 3 of Section 4.1, described by four leave-nodes asymmetric tree in (a), and complete binary trees with depth $p = 3$ and $k = 2^p = 8$ leaf nodes in (b) and with depth $p = 2$ and $k = 4$ leaf nodes in (c), respectively.

In addition to test errors, F1-scores are computed for each classifier, which are between 0 and 1 and measure a classification (test)'s accuracy. A F1-score is defined as $2\frac{\rho \cdot r}{\rho + r}$, where the precision ρ is the number of correct results over the number of all results classified to a class by the trained classifier, and the recall r is the number of correct results divided by the number of instances with true label of a class. Specifically, for a given classifier, a F1-score is defined as a weighted average of F1-scores over all classes, weighted by the sample distribution.

For each classifier, we use one independent tuning sample of size n and one independent testing sample of 5×10^4, for tuning and testing. For tuning, the optimal C is obtained by minimizing the tuning error defined in (6) on 61 grid points: $C = 10^{l/10}; l = -30, -29, \cdots, 30$. Given the estimated optimal C, the test error in (6) is computed over the test sample.

Example 1. A random sample $(Y_i, \boldsymbol{X}_i = \{X_{i1}, X_{i2}\})_{i=1}^n$ is generated as follows. First, $\boldsymbol{X}_i \sim U^2(0,1)$ is sampled from the two-dimensional uniform distribution. Second, $Y_i \in \{1,2,3,4\}$ is sampled through conditional distributions: $P(Y_i = 1|X) = 0.17$, $P(Y_i = 2|X) = 0.17$, $P(Y_i = 3|X) = 0.17$, $P(Y_i = 4|X) = 0.49$. This generates a simple asymmetric distribution over a tree hierarchy with a four leaf-nodes as displayed in Figure 2(a).

Clearly, HSVM and HPSI outperform their competitors - HSVM$_c$, SHSVM and SVM under each the four evaluation losses in both linear and Gaussian kernel situations. Specifically, the improvement amount of HSVM over SVM varies from 1.5% to 3.1% in the linear case and 1.6% to 1.9% in the Gaussian kernel case, whereas that of HPSI ranges from 94.5% to 94.7% and 100.0%, respectively. By comparison, the amount of improvement of HSVM$_c$ is from 0.7% to 1.0% in

Linear					
Training Method	Test error				
	l_{0-1}	l_Δ	l_{sib}	l_{sub}	F1-score
SVM	0.545 (0.048)	0.527 (0.023)	0.521 (0.014)	0.521 (0.014)	0.328 (0.012)
ψ-learning	0.515 (0.032)	0.512 (0.015)	0.511 (0.009)	0.511 (0.009)	0.321 (0.011)
% of impro.	86.9%	86.9%	86.9%	86.8%	
HSVM$_c$	0.545 (0.044)	0.527 (0.022)	0.521 (0.012)	0.520 (0.012)	0.328 (0.015)
% of impro.	1.0%	1.0%	0.7%	0.9%	
SHSVM	0.659 (0.130)	0.580 (0.061)	0.554 (0.038)	0.554 (0.038)	0.248 (0.013)
% of impro.	-321.8%	-315.8%	-311.9%	-312.7%	
HSVM	0.545 (0.043)	0.526 (0.021)	0.520 (0.013)	0.520 (0.013)	0.327 (0.005)
% of impro.	1.5%	2.6%	3.0%	3.1%	
HPSI	**0.512(0.019)**	**0.511(0.009)**	**0.511(0.006)**	**0.511(0.006)**	0.322 (0.102)
% of impro.	94.7%	94.6%	94.5%	94.5%	
Bayes Rule	0.51	0.51	0.51	0.51	0.322
Gaussian					
Training Method	Test error				
	l_{0-1}	l_Δ	l_{sib}	l_{sub}	F1-score
SVM	0.547 (0.055)	0.528 (0.026)	0.521 (0.017)	0.521 (0.017)	0.326 (0.012)
ψ-learning	**0.510(0.000)**	**0.510(0.000)**	**0.510(0.000)**	**0.510(0.000)**	0.322 (0.000)
% of impro.	100%	100%	100%	100%	
HSVM$_c$	0.547 (0.054)	0.527 (0.022)	0.521 (0.015)	0.521 (0.015)	0.325 (0.011)
% of impro.	1.0%	1.0%	1.1%	1.1%	
SHSVM	0.626 (0.115)	0.565 (0.054)	0.544 (0.034)	0.544 (0.034)	0.280 (0.078)
% of impro.	-214.6%	-212.3%	-209.8%	-209.8%	
HSVM	0.546 (0.050)	0.527 (0.024)	0.521 (0.015)	0.521 (0.015)	0.324 (0.010)
% of impro.	1.6%	1.6%	1.9%	1.9%	
HPSI	**0.510(0.000)**	**0.510(0.000)**	**0.510(0.000)**	**0.510(0.000)**	0.322 (0.000)
% of impro.	100%	100%	100%	100%	
Bayes Rule	0.51	0.51	0.51	0.51	0.322

Table 1: Averaged test errors as well as estimated standard deviations (in parenthesis) of SVM, ψ-learning, SHSVM, HSVM, HPSI and HSVM$_c$ over 100 simulation replications in Example 1 of Section 4.1. The testing errors are computed under the l_{0-1}, l_Δ, l_{sib} and l_{sub}. The bold face represents the best performance among four competitors for any given loss. For reference, F1-scores, as defined in Section 4.1, for these classifiers are given as well.

the linear case and from 1.0% to 1.1% in the Gaussian kernel case, and that of SHSVM is from -321.8% to -311.9% and -214.6% to -209.8%, which means it is actually much worse than SVM. From hypothesis testing view, the differences of the means for HPSI and SVM are more than three times of the standard error of the differenced means, indicating that these means are statistically different at level of $\alpha = 5\%$. Moreover, HPSI get F1-scores very close to that of the Bayes rule.

In summary, HSVM, especially HPSI indeed yield significant improvements over its competitors.

Example 2. A complete binary tree of depth 3 is considered, which is displayed in Figure 2 (b). There are eight leaf and six non-leaf nodes, coded as $\{1,\cdots,8\}$ and $\{9,\cdots,14\}$, respectively. A random sample of 100 instances $(Y_i, \boldsymbol{X}_i = (X_{i1}, X_{i2}))_{i=1}^{100}$ is generated as follows: $\boldsymbol{X}_i \sim U^2(-1,1)$, where $U^2(-1,1)$ is the uniform distribution on unit square, $Y_i|\boldsymbol{X}_i = \lceil 8 \times X_{i1} \rceil \in \{1,\cdots,8\}$. Then 5% of the samples are randomly chosen with the label values redefined as $Y_i = Y_i + 1$ if $Y_i \neq 8$ and $Y_i = Y_i$ if $Y_i = 8$. Another 5% of the samples are randomly chosen with the label values redefined as $Y_i = Y_i - 1$ if $Y_i \neq 1$ and $Y_i = Y_i$ if $Y_i = 1$. For non-leaf node j, $P(Y_i = j|\boldsymbol{X}_i) = \sum_{\{t \in sub(j) \cap \mathcal{L}\}} P(Y_i = t|\boldsymbol{X}_i)$. This generates a non-separable case.

Linear					
Training Method	Test error				
	l_{0-1}	l_Δ	l_{sib}	l_{sub}	F1-score
SVM	0.326(0.004)	0.179(0.003)	0.148(0.002)	0.122(0.002)	0.671(0.004)
ψ-learning	0.21(0.004)	0.107(0.003)	0.091(0.002)	0.072(0.002)	0.787(0.004)
% of impro.	47.7%	55.4%	50.9%	53.8%	
HSVM$_c$	0.323(0.006)	0.169(0.002)	0.148(0.003)	0.120(0.002)	0.677(0.006)
% of impro.	1.2%	7.7%	0%	2.2%	
SHSVM	0.201(0.003)	0.106(0.002)	0.086(0.001)	0.070(0.001)	0.798(0.003)
% of impro.	51.4%	56.1%	55.4%	55.9%	
HSVM	0.199(0.003)	0.105(0.002)	0.086(0.001)	0.070(0.001)	0.800(0.003)
% of impro.	52.3%	56.9%	55.4%	55.9%	
HPSI	**0.195(0.003)**	**0.102(0.001)**	**0.086(0.002)**	**0.068(0.002)**	0.804(0.003)
% of impro.	53.9%	59.2%	55.4%	58.1%	
Bayes Rule	0.083	0.049	0.036	0.029	0.916
Gaussian					
Training Methods	Test error				
	l_{0-1}	l_Δ	l_{sib}	l_{sub}	F1-score
SVM	0.305(0.015)	0.209(0.001)	0.135(0.008)	0.110(0.007)	0.696(0.015)
ψ-learning	0.206(0.005)	0.113(0.003)	0.087(0.004)	0.069(0.003)	0.798(0.005)
% of impro.	44.6%	60.0%	48.5%	50.6%	
HSVM$_c$	0.313(0.005)	0.166(0.003)	0.128(0.006)	0.109(0.005)	0.685(0.005)
% of impro.	−3.6%	26.9%	7.1%	1.2%	
SHSVM	0.202(0.003)	0.110(0.002)	0.086(0.001)	0.068(0.001)	0.792(0.003)
% of impro.	46.4%	61.9%	49.5%	51.9%	
HSVM	0.205(0.003)	0.112(0.002)	0.087(0.001)	0.069(0.001)	0.795(0.003)
% of impro.	45.0%	60.6%	48.5%	50.6%	
HPSI	**0.190(0.002)**	**0.102(0.002)**	**0.085(0.002)**	**0.063(0.002)**	0.815(0.002)
% of impro.	51.8%	66.9%	50.5%	58.0%	
Bayes Rule	0.083	0.049	0.036	0.029	0.916

Table 2: Averaged test errors as well as estimated standard deviations (in parenthesis) of SVM, ψ-learning, SHSVM, HSVM, HPSI and HSVM$_c$ over 100 simulation replications in Example 2 of Section 4.1. The testing errors are computed under the l_{0-1}, l_Δ, l_{sib} and l_{sub}. The bold face represents the best performance among four competitors for any given loss. For reference, F1-scores, as defined in Section 4.1, for these classifiers are given as well.

As suggested in Table 2, HSVM and HPSI outperform the three competitors under l_{0-1}, l_Δ, l_{sib} and l_{sub} in the linear case, whereas HSVM performs slightly worse than SHSVM in the Gaussian

case. In both cases, the amount of improvement of HSVM and HPSI over their flat counterpart varies. Clearly, HPSI is the winner and outperforms its competitors in all the situations.

With regard to the test errors in Table 2, we also observe the following aspects. First, the five classifiers perform similarly under l_Δ, l_{sib} and l_{sub}. This is because all the eight leaf node classes are at level 3 of the hierarchy, resulting a similar structure under these evaluation losses. Second, the classifiers perform similarly for linear learning and Gaussian kernel learning. This is mainly due to the fact that the ideal optimal decision rule—Bayes rule is linear in this case. Moreover, HPSI and HSVM always have better F1-scores, which are the two most close to that of the Bayes rule.

In summary, HSVM and HPSI indeed yield improvements over their flat counterparts because of the built-in hierarchical structure, and HPSI outperforms its competitors. Here, the hierarchy—a tree of depth 3 is useful in reducing a classification problem's complexity which can be explained by the concept of the margins with respect to hierarchy, as discussed in Section 3.1.

Example 3. A random sample $(Y_i, \boldsymbol{X}_i = \{X_{i1}, X_{i2}\})_{i=1}^n$ is generated as follows. First, $\boldsymbol{X}_i \sim U^2(-1,1)$ is sampled. Second, $Y_i = 1$ if $X_{i1} < 0$ and $X_{i2} < 0$; $Y_i = 2$ if $X_{i1} < 0$ and $X_{i2} \geq 0$; $Y_i = 3$ if $X_{i1} \geq 0$ and $X_{i2} < 0$; $Y_i = 4$ if $X_{i1} \geq 0$ and $X_{i2} \geq 0$. Third, 20% of the sample are chosen at random and their labels are randomly assigned to the other three classes. For non-leaf nodes 5 and 6, $P(Y_i = 5|\boldsymbol{X}_i) = P(Y_i = 1|\boldsymbol{X}_i) + P(Y_i = 2|\boldsymbol{X}_i)$, and $P(Y_i = 6|\boldsymbol{X}_i) = P(Y_i = 3|\boldsymbol{X}_i) + P(Y_i = 4|\boldsymbol{X}_i)$. This generates a complete binary tree of depth 2, where nodes 1 and 2 are siblings of node 5, and nodes 3 and 4 are siblings of node 6, see Figure 2 (c). Experiments are performed with different training sample sizes of 50, 150, 500 and 1500.

Again, HSVM and HPSI outperform their competitors- HSVM$_c$, SHSVM and SVM under the four evaluation losses in all the situations. The amount improvement of HSVM over SVM varies from 22.4% to 52.6% in the linear case and 8.9% to 42.5% in the Gaussian kernel case, whereas that of HPSI ranges from 39.5% to 89.5% and 20.6% to 80.6%, respectively. By comparison, the amount of improvement of HSVM$_c$ is from 6.4% to 23.8% in the linear case and from 2.4% to 18.8% in the Gaussian kernel case, and that of SHSVM is from 21.1% to 47.4% and 9.5% to 45.2%. With regard to F1-scores, HPSI and HSVM remain to be the best, and are much more close to that of the Bayes rule.

In summary, the improvement of HPSI over HSVM becomes more significant when the training size increases. As expected, HPSI is the winner and nearly achieves the optimal performance of the Bayes rule when the sample size gets large.

4.2 Classification of Gene Functions

Biological functions of many known genes remain largely unknown. For yeast *S. cerevisiae*, only 68.5% of the genes were annotated in MIPS, as of May, 2005, for which many of them have only general functions annotated in some top-level categories. Discovery of biological functions therefore becomes very important in biomedical research. As effective means, gene function prediction is performed through known gene functions and gene expression profiles of both annotated and unannotated genes. Biologically, it is generally believed that genes having the same or similar functions tend to be coexpressed (Hughes et al., 2000). By learning the patterns of expression profiles, a gene with unknown functions can be classified into existing functional categories, as well as newly created functional categories. In the process of prediction, classification is essential, as to be discussed next.

Linear							
l &	Sample	TE and % of impro.					
Bayes Rule	Size	SVM	ψ-learning	HSVM$_c$	SHSVM	HSVM	HPSI
l_{0-1}	n=50	0.347(0.070)	0.315(0.058)	0.337(0.047)	0.316(0.047)	0.314(0.058)	**0.289(0.045)**
			21.8%	6.8%	21.1%	22.4%	39.5%
	n=150	0.284(0.043)	0.261(0.030)	0.275(0.023)	0.263(0.023)	0.260(0.030)	**0.237(0.016)**
0.200			27.4%	10.7%	25.0%	28.6%	56.0%
	n=500	0.247(0.014)	0.234(0.013)	0.241(0.014)	0.235(0.014)	0.233(0.013)	**0.213(0.007)**
			27.7%	12.8%	25.5%	29.8%	72.3%
	n=1500	0.230(0.010)	0.217(0.005)	0.223(0.009)	0.218(0.009)	0.217(0.005)	**0.205(0.003)**
			43.3%	23.3%	40.0%	43.3%	83.3%
l_{sib}	n=50	0.276(0.056)	0.249(0.046)	0.269(0.037)	0.250(0.037)	0.248(0.046)	**0.229(0.035)**
			24.8%	6.4%	23.9%	25.7%	43.1%
	n=150	0.230(0.032)	0.211(0.022)	0.222(0.018)	0.213(0.018)	0.210(0.022)	**0.191(0.012)**
0.167			30.2%	12.7%	27.0%	31.7%	61.9%
	n=500	0.203(0.012)	0.193(0.011)	0.198(0.012)	0.194(0.012)	0.192(0.011)	**0.175(0.005)**
			27.8%	13.9%	25.0%	30.6%	77.8%
	n=1500	0.188(0.007)	0.178(0.004)	0.183(0.007)	0.179(0.007)	0.178(0.004)	**0.170(0.002)**
			47.6%	23.8%	42.9%	47.6%	85.7%
l_{sub}	n=50	0.252(0.051)	0.227(0.042)	0.244(0.041)	0.228(0.041)	0.226(0.042)	**0.210(0.033)**
			26.0%	8.3%	25.0%	27.1%	43.8%
	n=150	0.212(0.029)	0.194(0.020)	0.203(0.020)	0.196(0.020)	0.193(0.020)	**0.176(0.010)**
0.156			32.1%	16.1%	28.6%	33.9%	64.3%
	n=500	0.188(0.011)	0.179(0.010)	0.184(0.011)	0.180(0.011)	0.178(0.010)	**0.162(0.005)**
			28.1%	12.5%	25.0%	31.3%	81.3%
	n=1500	0.175(0.007)	0.165(0.004)	0.172(0.007)	0.166(0.007)	0.165(0.004)	**0.158(0.002)**
			52.6%	15.8%	47.4%	52.6%	89.5%
l_{Δ}	n=50	0.184(0.037)	0.166(0.031)	0.179(0.025)	0.167(0.025)	0.165(0.031)	**0.153(0.023)**
			24.7%	6.4%	23.7%	25.6%	42.9%
	n=150	0.153(0.021)	0.141(0.015)	0.148(0.012)	0.142(0.012)	0.140(0.015)	**0.127(0.008)**
0.111			28.6%	12.6%	26.8%	31.5%	61.4%
	n=500	0.135(0.008)	0.128(0.007)	0.132(0.008)	0.129(0.008)	0.128(0.007)	**0.117(0.003)**
			29.2%	13.7%	24.7%	30.1%	76.7%
	n=1500	0.125(0.005)	0.119(0.003)	0.122(0.005)	0.119(0.005)	0.119(0.003)	**0.113(0.002)**
			42.9%	23.3%	41.9%	46.5%	83.7%
F1-score	n=50	0.557(0.106)	0.588(0.107)	0.571(0.075)	0.588(0.076)	0.589(0.106)	**0.597(0.110)**
			12.8%	5.8%	12.8%	13.2%	16.5%
	n=150	0.672(0.063)	0.701(0.055)	0.683(0.026)	0.710(0.026)	0.691(0.054)	**0.719(0.034)**
0.800			22.7%	8.6%	29.7%	14.8%	36.7%
	n=500	0.721(0.016)	0.741(0.017)	0.729(0.015)	0.749(0.014)	0.737(0.017)	**0.754(0.012)**
			25.3%	10.1%	35.4%	20.3%	41.8%
	n=1500	0.746(0.017)	0.763(0.015)	0.755(0.010)	0.763(0.010)	0.764(0.015)	**0.779(0.004)**
			31.5%	16.7%	31.5%	33.3%	61.1%

Table 3: Averaged test errors as well as estimated standard deviations (in parenthesis) of SVM, ψ-learning, HSVM$_c$, SHSVM, HSVM and HPSI over 100 simulation replications of linear learning in Example 3 of Section 4.1, with $n = 50, 150, 500, 1500$. The test errors are computed under the l_{0-1}, l_{Δ}, l_{sib} and l_{sub}. For reference, F1-scores, as defined in Section 4.1, for these classifiers are given as well.

Hughes et al. (2000) demonstrated the effectiveness of gene function prediction through genome-wide expression profiles, and identified and experimentally confirmed eight uncharacterized open reading frames as protein-coding genes. Specifically, three hundred expressions were profiled for the genome of yeast *S. cerevisiae*, in which transcript levels of a mutant or a compound-treated culture were compared against that of a wild-type or a mock-treated culture. Three hundred experiments, consisting of 276 deletion mutants, 11 tetracycline-regulatable alleles of essential genes, and 13 well-characterized compounds. Deletion mutants were selected such that a variety of functional

Gaussian							
l &	Sample	TE and % of impro.					
Bayes Rule	Size	SVM	ψ-learning	HSVM$_c$	SHSVM	HSVM	HPSI
l_{0-1}	n=50	0.326(0.060)	0.313(0.047)	0.323(0.047)	0.314(0.047)	0.313(0.047)	**0.300(0.045)**
			10.3%	2.4%	9.5%	8.9%	20.6%
	n=150	0.280(0.036)	0.27 (0.027)	0.276(0.030)	0.270(0.030)	0.270(0.027)	**0.261(0.016)**
0.200			12.5%	5.0%	12.5%	12.5%	23.8%
	n=500	0.257(0.022)	0.24 (0.014)	0.252(0.013)	0.239(0.013)	0.240(0.014)	**0.224(0.007)**
			29.8%	8.8%	31.6%	29.8%	57.9%
	n=1500	0.247(0.013)	0.227(0.010)	0.240(0.011)	0.226(0.011)	0.227(0.010)	**0.215(0.003)**
			42.6%	14.9%	44.7%	42.5%	68.1%
l_{sib}	n=50	0.263(0.048)	0.250(0.037)	0.257(0.037)	0.251(0.037)	0.250(0.037)	**0.243(0.035)**
			13.5%	6.3%	12.5%	9.8%	20.8%
	n=150	0.229(0.029)	0.218(0.022)	0.225(0.022)	0.219(0.022)	0.218(0.022)	**0.211(0.012)**
0.167			17.7%	6.5%	16.1%	13.1%	29.0%
	n=500	0.208(0.016)	0.195(0.010)	0.202(0.011)	0.194(0.011)	0.195(0.010)	**0.181(0.005)**
			31.7%	14.6%	34.1%	31.7%	65.9%
	n=1500	0.198(0.008)	0.185(0.006)	0.193(0.005)	0.184(0.005)	0.185(0.006)	**0.173(0.003)**
			41.9%	16.1%	45.2%	40.9%	80.6%
l_{sub}	n=50	0.241(0.044)	0.227(0.034)	0.237(0.041)	0.230(0.041)	0.228(0.034)	**0.222(0.033)**
			16.5%	4.7%	12.9%	11.1%	22.4%
	n=150	0.211(0.027)	0.2 (0.020)	0.207(0.020)	0.202(0.020)	0.201(0.020)	**0.192(0.010)**
0.156			20.0%	7.3%	16.4%	13.5%	34.5%
	n=500	0.192(0.015)	0.179(0.009)	0.187(0.010)	0.179(0.010)	0.180(0.009)	**0.169(0.005)**
			36.1%	13.9%	36.1%	33.3%	63.9%
	n=1500	0.188(0.009)	0.175(0.006)	0.182(0.005)	0.175(0.005)	0.176(0.006)	**0.163(0.003)**
			40.6%	18.8%	40.6%	35.4%	78.1%
l_{Δ}	n=50	0.175(0.032)	0.167(0.025)	0.171(0.025)	0.167(0.025)	0.166(0.025)	**0.162(0.023)**
			12.5%	6.2%	12.4%	9.8%	20.7%
	n=150	0.153(0.019)	0.145(0.015)	0.150(0.014)	0.146(0.014)	0.145(0.015)	**0.141(0.008)**
0.111			19.0%	6.4%	16.0%	13.1%	28.8%
	n=500	0.139(0.011)	0.13 (0.007)	0.135(0.007)	0.129(0.007)	0.130(0.007)	**0.121(0.003)**
			32.1%	14.5%	33.7%	31.7%	65.1%
	n=1500	0.132(0.005)	0.123(0.004)	0.129(0.004)	0.123(0.004)	0.123(0.004)	**0.115(0.002)**
			42.9%	15.9%	44.4%	41.3%	79.4%
F1-score	n=50	0.559(0.105)	0.589(0.107)	0.573(0.076)	0.590(0.076)	0.591(0.105)	**0.595(0.109)**
			12.4%	5.8%	12.9%	13.3%	14.9%
	n=150	0.674(0.062)	0.703(0.053)	0.686(0.024)	0.713(0.025)	0.695(0.051)	**0.717(0.033)**
0.800			23.0%	9.5%	31.0%	16.7%	34.1%
	n=500	0.723(0.016)	0.744(0.017)	0.732(0.014)	0.752(0.014)	0.740(0.016)	**0.753(0.012)**
			27.3%	11.7%	37.7%	22.1%	39.0%
	n=1500	0.747(0.017)	0.765(0.015)	0.757(0.011)	0.766(0.010)	0.767(0.015)	**0.776(0.004)**
			34.0%	18.9%	35.8%	37.7%	54.7%

Table 4: Averaged test errors as well as estimated standard deviations (in parenthesis) of SVM, ψ-learning, HSVM$_c$, SHSVM, HSVM and HPSI over 100 simulation replications of kernel learning in Example 3 of Section 4.1, with $n = 50, 150, 500, 1500$. The test errors are computed under the l_{0-1}, l_{Δ}, l_{sib} and l_{sub}. For reference, F1-scores, as defined in Section 4.1, for these classifiers are given as well.

classifications were represented. Experiments were performed under a common condition to allow direct comparison of the behavior of all genes in response to all mutations and treatments. Expressions of the three hundred experiments were profiled through a two-channel cDNA chip technology (or hybridization assay). As suggested in Hughes et al. (2000), the expression profiles were indeed informative to gene function prediction.

In gene function prediction, one major difficulty is the presence of a large number of function categories with relatively small-sample size, which is known as the situation of large number of cate-

gories in classification. To battle the curse of dimensionality, a structured approach needs to be used with built-in biological knowledge presented in a form of annotation system such as MIPS, where a flat approach does not perform better than a classifier that uses a correct hierarchical structure. Comparisons can be found in Shahbaba and Neal (2007), and Cesa-Bianchi and Valentini (2009). The problem of gene function prediction is an ideal test case for hierarchical classification, where accuracy of prediction is key. In the literature, recalibration and combination of different large margin methods, including sequential HSVM and loss scaled SVM, were used in gene function prediction, see, for example, Obozinski et al. (2008), Guan et al. (2008), and Valentini and Re (2009). Astikainen et al. (2008) used a different representation with a loss-scaled SVM. Cesa-Bianchi et al. (2006), and Cesa-Bianchi and Valentini (2009) employed a Bayesian ensemble method.

Through gene expression data in Hughes et al. (2000), we apply HSVM and HPSI to predict gene functions. Of particular consideration is prediction of functional categories of unknown genes within two major branches of MIPS, composed of two functional categories at the highest level: "cell cycle and DNA processing" and "transcription" and their corresponding offsprings. Within these two major branches, we have $n = 1103$ annotated genes together with $p = 300$ expressions for each gene and a tree hierarchy of $K = 23$ functional categories, see Figure 3 for a display of the tree hierarchy. In this case, the predictor x represents the expression levels of a gene, consisting of the log-ratios (base 10) of the mRNA abundance in the test samples relative to the reference samples, and label Y indicates the location within the MIPS hierarchy. For this MIPS data, some genes belong to two or more functional categories. To place the problem of gene function prediction into our framework of hierarchical classification, we may create a new separate category for all genes that are annotated with a common set of categories. For an example, in Figure 2 (b), if we observed cases of common members for Categories 2 and 3, we will create a category, say Category 15, which is the sibling of Categories 9 and 10. Those common members will be assigned to Category 15.

Notice that, although we have nearly balanced situation in this example, in general we may see unbalanced situations.

We now perform some simulations to gain insight into HSVM and HPSI for gene function prediction before applying to predict new gene functions with respect to MIPS. For this purpose, we use the 2005 version of MIPS and proceed 100 randomly partition the entire set of data of 1103 genes into training, tuning and testing sets, with 303, 300, and 500 genes, respectively. Then HSVM and HPSI are trained with training samples, tuned and tested as in Section 4.1, over 100 different random partitions to avoid homologous gene clusters. Their performance is measured by the test errors is averaged over 100 random partitions.

As suggested by Table 5, besides similar results in F1-score, HSVM and HPSI outperform SVM and HSVM$_c$ under l_{0-1}, l_Δ, l_{sib} and l_Δ, in both the linear and Gaussian kernel cases. With respect to these four losses, the amount of improvement of HSVM over SVM ranges from 0.1% to 31.8%, whereas that of HPSI over SVM is from 0.1% to 32.3%. Among these four losses, l_Δ and l_{sub} yield the largest amount of improvement. This suggests that HPSI and HSVM classify more precisely at the top levels than at the bottom levels of the hierarchy, where the inter-class dependencies are weak. Note that l_Δ and l_{sub} penalize misclassification more at relevant nodes at lower levels in deep branches, whereas l_{sib} only does so at upper levels. Interestingly, small and large branches have the same parents, leading to large differences in penalties under different losses. It is also noted that the F1-scores are not significantly above 0 for all the large margin methods we are comparing here.

We are now ready to apply our method to real challenge of predicting unknown gene functional categories that had not been annotated in the 2005 version of MIPS. The predicted gene functions

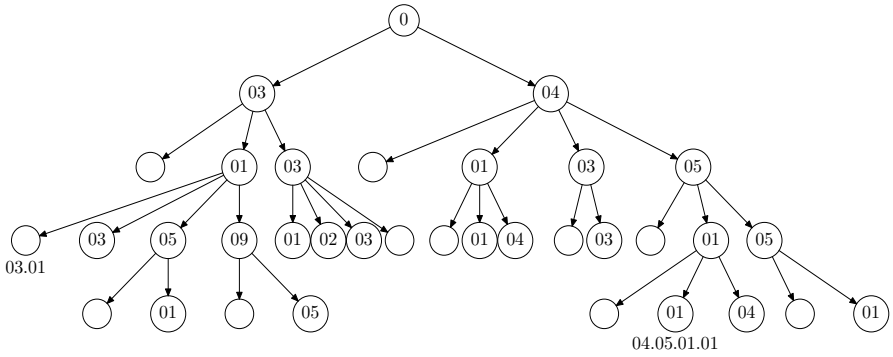


Figure 3: Two major branches of MIPS, with two functional categories at the highest level: "Cell cycle and DNA processing" and "Transcription". For each node, the corresponding functional category is defined by a combination of numbers of itself and all of its ancestors. For instance, the middle node 01 at level 4 indicates functional category 04.05.01.01 in MIPS, which corresponds to "General transcription activities". Notes that a blank node represents its parent itself, for instance, the left blank node at level 3 indicates functional category 03.01.

will be cross-validated by a newer version of MIPS, dated in March 2008, where about 600 additional genes have been added into functional categories, representing the latest biological advance. We proceed in three steps. First, we use the tuned HSVM and HPSI trained through the training samples in the 2005 version of MIPS, which are the best performer over the 100 random partitions. Second, the trained HPSI and HSVM are applied to ten most confident genes for prediction, which are chosen among unannotated genes in the 2005 version but annotated in the 2008 version. Here the confidence is measured by the value of the functional margin. Third, ten predictions from HSVM and HPSI are cross-validated by the 2008 version of MIPS.

As indicated in Table 6, seven out of the ten genes are predicted correctly for both HSVM and HPSI. For an example, gene "YGR054w" is not annotated in the 2005 version of MIPS, and is predicted to belong to functional categories along a path "Protein synthesis" $\rightarrow$ "Ribosome biogenesis" $\rightarrow$ "Ribosomal proteins" by HPSI. This prediction is confirmed to be exactly correct by the newer version of MIPS.

5. Statistical Learning Theory

In the literature, the generalization accuracy for hierarchical classification and the role of $\mathcal{H}$ have not been widely studied. This section develops an asymptotic theory to quantify the generalization accuracy of the proposed hierarchical large margin classifier $d^H(\hat{\boldsymbol{f}})$ defined by (2) for a general loss v. In particular, the rate of convergence of $d^H(\hat{\boldsymbol{f}})$ is derived. Moreover, we apply the theory to one illustrative example to study when and how $\mathcal{H}$ improves the performance over flat classification.

Linear					
Training Methods	Test error				
	l_{0-1}	l_{Δ}	l_{sib}	l_{sub}	F1-score
SVM	0.972(0.006)	0.651(0.024)	0.926(0.008)	0.593(0.029)	0.007(0.002)
ψ-learning	0.961(0.009)	0.633(0.023)	0.92(0.022)	0.581(0.041)	0.007(0.002)
% of impro.	1.13%	2.8%	0.7%	2.0%	
SHSVM	0.962(0.007)	0.552(0.031)	0.927(0.008)	0.442(0.036)	0.015(0.002)
% of impro.	1.0%	15.2%	0.1%	25.5%	
HSVM	0.960(0.009)	0.520(0.023)	0.918(0.022)	0.433(0.041)	0.008(0.002)
% of impro.	1.2%	20.0%	0.8%	27.0%	
HPSI	**0.958(0.008)**	**0.517(0.020)**	**0.917(0.020)**	**0.430(0.038)**	0.009(0.002)
% of impro.	1.4%	20.6%	1.0%	27.5%	
Gaussian					
Training Methods	Test error				
	l_{0-1}	l_{Δ}	l_{sib}	l_{sub}	F1-score
SVM	0.976(0.002)	0.669(0.005)	0.921(0.003)	0.617(0.007)	0.007(0.002)
ψ-learning	0.961(0.008)	0.660(0.020)	0.92(0.019)	0.601(0.030)	0.007(0.002)
% of impro.	1.5%	1.3%	0.1%	2.6%	
SHSVM	0.963(0.006)	0.558(0.033)	0.920(0.009)	0.430(0.029)	0.016(0.002)
% of impro.	1.3%	16.6%	0.1%	30.3%	
HSVM	0.961(0.008)	0.515(0.020)	0.920(0.019)	0.421(0.030)	0.008(0.002)
% of impro.	1.5%	23.0%	0.1%	31.8%	
HPSI	**0.960(0.008)**	**0.512(0.021)**	**0.920(0.020)**	**0.418(0.030)**	0.009(0.002)
% of impro.	1.6%	23.5%	0.1%	32.3%	

Table 5: Averaged test errors as well as estimated standard deviations (in parenthesis) of SVM, ψ-learning, SHSVM, HSVM and HPSI, in the gene function example in Section 4.2, over 100 simulation replications. The testing errors are computed under l_{0-1}, l_{Δ}, l_{H-sib} and l_{H-sub}. The bold face represents the best performance among four competitors for any given loss. For reference, F1-scores, as defined in Section 4.1, for these classifiers are given as well.

5.1 Theory

In classification, the performance of our classifier $d^H(\hat{\boldsymbol{f}})$ is measured by the difference between the actual performance of $\hat{\boldsymbol{f}}$ and the ideal optimal performance of $\bar{\boldsymbol{f}}$, defined as $e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}}) = GE(d^H(\hat{\boldsymbol{f}})) - GE(d^H(\bar{\boldsymbol{f}})) = E(l_{0-1}(Y,d^H(\hat{\boldsymbol{f}}(\boldsymbol{X}))) - l_{0-1}(Y,d^H(\bar{\boldsymbol{f}}(\boldsymbol{X})))) \geq 0$. Here $GE(d^H(\bar{\boldsymbol{f}}))$ is the optimal performance for any classifier provided that the unknown true distribution $P(\boldsymbol{x},y)$ would have been available. In hierarchical classification with k leaf and $(K-k)$ non-leaf node classes, the Bayes decision function vector $\bar{\boldsymbol{f}}$ is a decision function vector yielding the Bayes classifier under d^H, that is, $d^H(\bar{\boldsymbol{f}}(\boldsymbol{x})) = \bar{d}(\boldsymbol{x})$. In our context, we define $\bar{\boldsymbol{f}}$ as follows: for each j, $\bar{f}_j(\boldsymbol{x}) = \max_{t:t\in sub(j)\cap\mathcal{L}} P(Y = t|\boldsymbol{X}=\boldsymbol{x})$ if $j \notin \mathcal{L}$ and $\bar{f}_j(\boldsymbol{x}) = P(Y=j|\boldsymbol{X}=\boldsymbol{x})$ if $j \in \mathcal{L}$.

Let $e_V(\boldsymbol{f},\bar{\boldsymbol{f}}) = E(V(\boldsymbol{f},\boldsymbol{Z}) - V(\bar{\boldsymbol{f}},\boldsymbol{Z})) \geq 0$ and $\lambda = (nC)^{-1}$, where $V(\boldsymbol{f},\boldsymbol{Z})$ is defined as $v(u_{min}(\boldsymbol{f}(\boldsymbol{X}),Y))$, $\boldsymbol{Z} = (\boldsymbol{X},Y)$, and $v(\cdot)$ is any large margin surrogate loss used in (2).

The following theorem quantifies Bayesian regret $e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}})$ in terms of the tuning parameter C through $\lambda = \frac{1}{nC}$, the sample size n, the smoothness parameters (α,β) of a surrogate loss V-based classification model, and the complexity of the class of candidate function vectors $\mathcal{F}$. Note that the assumptions below are parallel to those of Theorem 3 in Liu and Shen (2006) for a statistical

Gene	Function category	Prediction verified			
		HSVM	HPSI	HSVM$_c$	SHSVM
YGR054w	translation initiation	Yes	Yes	Yes	Yes
YCR072c	ribosome biogenesis	Yes	Yes	Yes	Yes
YFL044c	transcriptional control	Yes	Yes	Yes	Yes
YNL156c	binding / dissociation	No	No	No	No
YPL201c	C-compound and carbohydrate utilization	Yes	Yes	Yes	Yes
YML069W	mRNA synthesis	Yes	Yes	Yes	Yes
YOR039W	mitotic cell cycle and cell cycle control	Yes	Yes	No	Yes
YNL023C	mRNA synthesis	No	Yes	No	No
YPL007C	mRNA synthesis	No	No	No	No
YDR279W	DNA synthesis and replication	Yes	No	No	No

Table 6: Verification of 10 gene predictions using an updated MIPS system and their functional categories.

learning theory for multiclass SVM and ψ-learning. In particular, Assumptions A-C described in Appendix are used to quantity the error rate of the classifier, in addition to a complexity measure the metric entropy with bracketing H_B for function space $\mathcal{F}$ defined before Assumption C.

Theorem 2 *Under Assumptions A-C in Appendix A, for any large margin hierarchical classifier $\boldsymbol{d}^H(\hat{\boldsymbol{f}})$ defined by (1), there exists a constant $c_6 > 0$ such that for any $x \geq 1$,*

$$P\left(e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) \geq c_1 x \delta_n^{2\alpha}\right) \leq 3.5 \exp(-c_6 x^{2-\min(\beta,1)} n(\lambda J_0)^{2-\min(\beta,1)}),$$

provided that $\lambda^{-1} \geq 2\delta_n^{-2} J_0$, where $\delta_n^2 = \min(\varepsilon_n^2 + 2e_V(\boldsymbol{f}^*, \bar{\boldsymbol{f}}), 1)$, $\boldsymbol{f}^* \in \mathcal{F}$ is an approximation in $\mathcal{F}$ to $\bar{f}$, $J_0 = \max(J(\boldsymbol{f}^*), 1)$ with $J(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^K \|f_j\|_{\mathcal{K}}^2$, and $\alpha, \beta, \varepsilon_n$ are defined in Assumptions A-C in Appendix A.

Corollary 1 *Under the assumptions in Theorem 2, $|e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}})| = O_p\left(\delta_n^{2\alpha}\right)$ and $E|e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}})| = O\left(\delta_n^{2\alpha}\right)$, provided that $n(\lambda J_0)^{2-\min(\beta,1)}$ is bounded away from 0 $n \to \infty$.*

The convergence rate for $e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}})$ is determined by δ_n^2, $\alpha > 0$ and $\beta > 0$, where δ_n captures the trade-off between the approximation error $e_V(\boldsymbol{f}^*, \bar{\boldsymbol{f}})$ due to use the surrogate loss V and estimation error ε_n^2, where ε_n is defined by the bracketing L_2 entropy of candidate function space $\mathcal{F}^V(t) = \{V^T(\boldsymbol{f}, \boldsymbol{z}) - V(\bar{\boldsymbol{f}}, \boldsymbol{z}) : \boldsymbol{f} \in \mathcal{F}, J(\boldsymbol{f}) \leq J_0 t\}$, and the last two quantify the first and second moments of $EV(\boldsymbol{f}, Z)$, where $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{Z} = (\boldsymbol{x}, Y)$.

By comparison, with V induced by a margin loss v, $\mathcal{F}^V(t)$ in multiclass classification is usually larger than its counterpart in hierarchical classification. This is because V is structured in that functional margin $u_{min}(\boldsymbol{f}(\boldsymbol{X}), Y)$ involves a smaller number of pairwise comparisons in hierarchical classification. In fact, only siblings for Y or one of Y's ancestors are compared. In contrast, any two classes need to be compared in multiclass classification without such a hierarchy (Liu and Shen, 2006). A theoretical description regarding the reduced size of the effective parameter space $\mathcal{F}^V(t)$ is given in the following lemma.

With regard to tightness of the bounds derived in Theorem 1, note that it reduces to multiclass margin classification, where the linear example in Shen and Wang (2007) indicates that the n^{-1} rate obtained from the upper bound theory agrees with the optimal rate of convergence.

Lemma 2 *Let $\mathcal{H}$ be a tree hierarchy with K non-root nodes including k leaf nodes. If $\mathcal{F}_1 = \cdots = \mathcal{F}_K$, then $H_B(\varepsilon, \mathcal{F}^V(t)) \leq 2c(\mathcal{H})H_B(\varepsilon/(2c(\mathcal{H})), \mathcal{F}_1(t))$ with v being the hinge and ψ losses, where $c(\mathcal{H}) = \sum_{j=0}^{K} \frac{|chi(j)|(|chi(j)|-1)}{2} \leq \frac{k(k-1)}{2}$ is the total number of comparisons required for hierarchical classification, and $\mathcal{F}_j(t) = \{f_j : \frac{1}{2}\|f_j\|_{\mathcal{K}} \leq J_0 t\}$; $j = 1, \cdots, K$.*

5.2 Bayes Classifier and Fisher-consistency

To compare different losses for the purpose of hierarchical classification, we introduce a new concept called "Fisher-consistency" with respect to $\mathcal{H}$. Before proceeding, we define the Bayes rule in Lemma 3 for K-class classification with non-exclusive membership, where only $k < K$ classes have mutually exclusive membership, determining the class membership of the other $K-k$ non-exclusive classes.

Lemma 3 *In K-class classification with non-exclusive membership, assume that the k mutually exclusive membership classes uniquely determine the membership of the other $K-k$ non-exclusive classes. That is, for any $t \in E$ and $\tilde{t} \notin E$, either $\{Y = \tilde{t}\} \supseteq \{Y = t\}$, or $\{Y = \tilde{t}\} \subseteq \{Y \neq t\}$, where E is the set of k mutually exclusive membership classes. Then the Bayes classifier $d(\boldsymbol{x}) = \operatorname{argmax}_{j \in E} P(Y = j | \boldsymbol{X} = \boldsymbol{x})$.*

Based on Lemma 3, we define Fisher-consistency with respect to $\mathcal{H}$ in hierarchical classification, which can be regarded as a generalization of Fisher-consistency in multi classification cases.

Definition 1 *In hierarchical classification, denote by $\mathcal{L}$ the set of classes corresponding to the leaf nodes in a tree. With $\mathcal{L}$ being a set of mutually exclusive membership classes, a loss $l(\cdot,\cdot)$ is said to be Fisher-consistent with respect to $\mathcal{H}$ if a global minimizer $El(Y, \boldsymbol{f}(\boldsymbol{X}))$ over all possible $\boldsymbol{f}(\boldsymbol{x})$ is $\bar{\boldsymbol{f}}$.*

Lemma 4 *Loss l_{0-1} is Fisher-consistent with respect to $\mathcal{H}$; so is l_Δ in the presence of a dominating leaf node class, that is, a class such that for any $\boldsymbol{x} \in S$ there exists a leaf node class j such that $P(Y = j | \boldsymbol{X} = x) > 1/2$.*

As shown in Lemma 4, l_{0-1} and l_Δ are Fisher-consistent with respect to $\mathcal{H}$.

Lemma 5 *Surrogate loss $v(u_{min}(\boldsymbol{f}(\boldsymbol{x}), y))$ is Fisher-consistent with respect to $\mathcal{H}$ when $v(\cdot)$ is either the hinge loss or the ψ loss.*

5.3 Theoretical Examples

Consider hierarchical classification with $\mathcal{H}$ defined by a complete binary tree with depth p. For this tree, there are $k = 2^p$ leaf nodes and $K = 2^{p+1} - 2 = 2k - 2$ non-root nodes, see Figure 2 (b) for an example of $p = 3$. Without loss of generality, denote by $\{j_1, \cdots, j_k\}$ the k leaf nodes. In what follows, we focus on the 0-1 loss with $l = l_{0-1}$.

A random sample is generated: $\boldsymbol{X} = (X_{(1)}, X_{(2)})$ sampled from the uniform distribution over $S = [0,1]^2$. For any leaf node j_i; $i = 1, \cdots, k$, when $X_{(1)} \in [(i-1)/k, i/k)$, $P(Y = j_i | \boldsymbol{X}) = (k-1)/k$,

and $P(Y=j|\boldsymbol{X})=1/[k(k-1)]$ for $j\neq j_i$. For any non-leaf node j_i; $i=k+1,\cdots,K$, $P(Y=j_i|\boldsymbol{X})=\sum_{t\in sub(j_i)\cap\mathcal{L}}P(Y=t|\boldsymbol{X})$. Then the Bayes rule $\bar{d}$ is defined from the Bayes decision function $\bar{\boldsymbol{f}}=\{\bar{f}_1,\cdots,\bar{f}_K\}$ through the top-down rule, where $\bar{\boldsymbol{f}}$ is defined as follows: For leaf nodes, $\bar{f}_{j_i}(\boldsymbol{x})=\sum_{t=1}^{i}(x_{(1)}-t/k)$; $i=1,\cdots,k$, so that when $x_{(1)}\in[(i_0-1)/k,i_0/k)$, $\bar{f}_{j_{i_0}}(\boldsymbol{x})=\max_{i=1,\cdots,k}\bar{f}_{j_i}(\boldsymbol{x})$. For non-leaf nodes, let it be the maximum over the leaf nodes in the subtree, that is, $\bar{f}_{j_i}(\boldsymbol{x})=\max_{\{t\in sub(j_i)\cap\mathcal{L}\}}\bar{f}_t$; $i=k+1,\cdots,K$.

Linear learning: Let $\mathcal{F}=\{(f_1,\cdots,f_K):f_j=\boldsymbol{w}_j^T\boldsymbol{x}+b_j\}$ and $J(\boldsymbol{f})=\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2$, where $\|\cdot\|$ is the EuclideanL_2-norm. We now verify Assumptions A-C for Corollary 1. It follows from Lemma 3 of Shen and Wang (2007) with $\boldsymbol{f}^*=\arg\inf_{\boldsymbol{f}\in\mathcal{F}}El_{0-1}(\boldsymbol{f},\boldsymbol{Z})$ for HSVM and $f_j^*=\sum_{t=1}^{j}n(x_{(1)}-t/k)$ for HPSI; $j=1,\cdots,k$, and $f_j^*=\max_{\{t\in sub(j)\cap\mathcal{L}\}}f_t^*$ otherwise. Assumptions A and B there are met with $\alpha=\frac{1}{2}$ and $\beta=1$ for HSVM, and with $\alpha=\beta=1$ for HPSI. For Assumption C, note that $H_B(\varepsilon,\mathcal{F}_1(t))\leq O(\log(1/\varepsilon))$, by Lemma 2 with $c(\mathcal{H})=\sum_{j=0}^{K}|chi(j)|(|chi(j)|-1)/2=\sum_{j=0}^{K}I\{j\notin\mathcal{L}\}=k-1$, we have, for HSVM and HPSI, $H_B(\varepsilon,\mathcal{F}^V(t))\leq O(k\log(k/\varepsilon))$ (Kolmogorov and Tihomirov, 1959). Consequently, $L\leq O(\varepsilon_n^2)$ in Assumption C, where

$$\phi(\varepsilon_n,s)=\int_{c_3L}^{c_4^{1/2}L^{\beta/2}}H_B^{1/2}(u,\mathcal{F}^V(s))du/L$$

and

$$\sup_{t\geq 2}\phi(\varepsilon_n,t)\leq O((k\log(k/\varepsilon_n))^{1/2}/\varepsilon_n).$$

Solving (7) in Assumption C leads to $\varepsilon_n=(\frac{k\log n}{n})^{1/2}$ for HSVM and HPSI when $C/J_0\sim\delta_n^{-2}/n\sim\frac{1}{n\varepsilon_n^2}$, provided that $\frac{k\log n}{n}\to 0$, with δ_n as defined in Theorem 2. Similarly, for multiclass SVM and ψ-learning, $\varepsilon_n=(\frac{k(k-1)/2\log n}{n})^{1/2}$ (Shen and Wang, 2007).

By Corollary 1, $|e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}})|=O_p\Big((k\log n/n)^{1/2}\Big)$ and $E\,|e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}})|=O\Big((k\log n/n)^{1/2}\Big)$ for HSVM, and $|e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}})|=O_p\Big(k\log n/n\Big)$ and $E\,|e(\hat{\boldsymbol{f}},\bar{\boldsymbol{f}})|=O\Big(k\log n/n\Big)$ for HPSI, when $\frac{k\log n}{n}\to 0$ as $n\to\infty$. By comparison, the rates of convergence for SVM and ψ-learning are $O\Big((\frac{k(k-1)}{2}\log n/n)^{1/2}\Big)$ and $O\Big(\frac{k(k-1)}{2}\log n/n\Big)$. In this case, the hierarchy enables to reduce the order from $\frac{k(k-1)}{2}$ down to k.

Note that $\mathcal{H}$ is a flat tree with only one layer, that is, all the leaf nodes are the direct offsprings of the root node 0, which means that $|chi(0)|=k$. Then $c(\mathcal{H})=\frac{|chi(0)|(|chi(0)|-1)}{2}=\frac{k(k-1)}{2}$. This would lead to the same rates of convergence for HSVM and HPSI as their counterparts.

Gaussian kernel learning: Consider the same setting with candidate function class defined by the Gaussian kernel. By the Aronszajn representation theorem of the reproducing kernel Hilbert spaces (Gu, 2000), it is convenient to embed a finite-dimensional Gaussian kernel representation into an infinite-dimensional space $\mathcal{F}=\{\boldsymbol{x}\in\mathcal{R}^2:\boldsymbol{f}(\boldsymbol{x})=(f_1(\boldsymbol{x}),\ldots,f_K(\boldsymbol{x}))$ with $f_j(\boldsymbol{x})=b_j+\boldsymbol{w}_j^T\phi(\boldsymbol{x})=b_j+\sum_{l=0}^{\infty}w_{j,l}\phi_l(\boldsymbol{x}):\boldsymbol{w}_j\in l_2\}$, and $\langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\rangle=\mathcal{K}(\boldsymbol{x},\boldsymbol{z})=\exp(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma_n^2})$, where σ_n is a scaling tuning parameter for the Gaussian kernel, which may depend on n. In what follows, we verify Assumptions A-C for HSVM and HPSI separately, and calculate δ_n in Corollary 1.

For HSVM, letting $f_j^*=1-(1+\exp(\sum_{t=1}^{j}\tau(x_{(1)}-t/k)))^{-1}$; for $j=1,\cdots,k$, and letting $f_j^*=\max_{\{t\in sub(j)\cap\mathcal{L}\}}f_t^*$ otherwise, $e(\boldsymbol{f}^*,\bar{\boldsymbol{f}})=O(k/\tau)$ and $J(\boldsymbol{f}^*)=O(ke^{\tau^2\sigma_n^2})$. Assumptions A and B are met with $\alpha=\beta=1$ by Lemmas 6 and 7 of Shen and Wang (2007). For Assumption C, following from Section 5.3 of Shen and Wang (2007), we have $H_B(\varepsilon,\mathcal{F}_1(t))\leq O((\log((J_0t)^{1/2}/\varepsilon))^3)$.

By Lemma 2, with $c(\mathcal{H}) = k-1$ as calculated in the linear cases, we have that $H_B(\varepsilon, \mathcal{F}^V(t)) \leq O(k(\log((J_0 t)^{1/2} k/\varepsilon))^3)$, where $J_0 = \max(J(\boldsymbol{f}^*), 1)$. Note that $L \leq O(\varepsilon_n^2)$. Then $\sup_{t\geq 2}\phi(\varepsilon_n, t) \leq O((k(\log((J_0 t)^{1/2} k/\varepsilon_n))^3)^{1/2}/\varepsilon_n)$. Solving (7) in Assumption C leads to $\varepsilon_n^2 = kn^{-1}(\log((J_0 n)^{1/2}))^3$ when $\lambda J_0 \sim \varepsilon_n^2$. By Corollary 1, $e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = O_p(\delta_n^2)$ and $Ee(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = O(\delta_n^2)$, with $\delta_n^2 = \max(kn^{-1}(\tau^2\sigma_n^2 + \sigma_n^{-2} + \log n)^3,\ k/\tau) = O_p\left(kn^{-1/7}\right)$ with $\tau \sim n^{1/7}$ when σ_n^2 is fixed, and $O_p(kn^{-1/4})$ when $\tau \sim \sigma_n^{-2} \sim n^{1/4}$.

For HPSI, let $f_j^* = \sum_{\tilde{j}=1}^{j} \tau(x_{(1)} - \tilde{j}/k)$; $j = 1, \cdots, k$, and $f_j^* = \max_{\{t \in sub(j)\cap \mathcal{L}\}} f_t^*$ otherwise. Then it can be verified that $e_L(\boldsymbol{f}^*, \bar{\boldsymbol{f}}) = O(k/\tau)$ and $J(\boldsymbol{f}^*) = O(k\tau^2\sigma_n^2)$. Assumptions A and B are met with $\alpha = \beta = 1$ by Theorem 3.1 of Liu and Shen (2006). Also $H_B(\varepsilon, \mathcal{F}_1(t)) \leq O((\log((J_0 t)^{1/2}/\varepsilon))^3)$, thus $\sup_{t\geq 2}\phi(\varepsilon_n, t) \leq O((k(\log((J_0 t)^{1/2} k/\varepsilon_n))^3)^{1/2}/\varepsilon_n)$. Similarly as in HSVM, solving (7) in Assumption C leads to $\varepsilon_n^2 = kn^{-1}(\log((J_0 n)^{1/2}))^3$ when $\lambda J_0 \sim \varepsilon_n^2$. By Corollary 1, $e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = O_p(\delta_n^2)$ and $Ee(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = O(\delta_n^2)$, with $\delta_n^2 = \max(kn^{-1}(\log(n\tau^2\sigma_n^2) + \sigma_n^{-2})^3,\ k/\tau) = O\left(kn^{-1}(\log n)^3\right)$ with $\tau \sim n(\log n)^{-3}$ and fixed σ_n^2, or $\sigma_n^2 \sim 1/\log n$.

An application of Theorem 1 in Shen and Wang (2007) yields the convergence rates of SVM and ψ-learning to be $O\left(\frac{k(k-1)}{2} n^{-1/7}\right)$ and $O\left(\frac{k(k-1)}{2} n^{-1}(\log n)^3\right)$, respectively. Again, the hierarchical structure reduces the order from $k(k-1)/2$ to k as in the linear case.

6. Discussion

This paper proposed a novel large margin method for single-path or partial-path hierarchical classification with mutually exclusive membership at the same level of a hierarchy. In contrast to existing hierarchical classification methods, the proposed method uses inter-class dependencies in a hierarchy. This is achieved through a new concept of generalized functional margins with respect to the hierarchy. By integrating the hierarchical structure into classification, the classification accuracy, or the generalization error defined by hierarchical losses, has been improved over its flat counterpart, as suggested by our theoretical and numerical analyses. Most importantly, the proposed method compares favorably against strong competitors in the large margin classification literature, especially from different settings of our synthetic simulations.

At present, the hierarchical structure is assumed to be correct. However, in applications, some classes may be mislabeled or unlabeled. In such a situation, a further investigation is necessary to generalize the proposed method, and also to allow for novel class detection.

Acknowledgments

This work is supported in part by NSF grants DMS-0604394 and DMS-0906616, NIH grants 1R01GM081535-01 and 2R01GM081535-01, NIH grants HL65462 and R01HL105397, and the Supercomputing Institute at University of Minnesota. The authors thank the action editor and referees for their helpful comments and suggestions.

Appendix A.

The following assumptions are made for Theorem 2.

For a given loss V, we define a truncated $V^T(\boldsymbol{f},\boldsymbol{Z}) = T \wedge V(\boldsymbol{f},\boldsymbol{Z})$ for any $\boldsymbol{f} \in \mathcal{F}$ and some truncation constant T, and $e_{V^T}(\boldsymbol{f},\bar{\boldsymbol{f}}) = E(V^T(\boldsymbol{f},\boldsymbol{Z}) - V(\bar{\boldsymbol{f}},\boldsymbol{Z}))$.

Assumption A: There exist constants $0 < \alpha \leq \infty$ and $c_1 > 0$ such that for any small $\varepsilon > 0$,

$$\sup_{\{\boldsymbol{f}\in\mathcal{F}\,:\, e_{V^T}(\boldsymbol{f},\boldsymbol{f}^*)\leq\varepsilon\}} |e(\boldsymbol{f},\bar{\boldsymbol{f}})| \leq c_1\varepsilon^{\alpha}.$$

Assumption B: There exist constants $\beta \geq 0$ and $c_2 > 0$ such that for any small $\varepsilon > 0$,

$$\sup_{\{\boldsymbol{f}\in\mathcal{F}\,:\, e_{V^T}(\boldsymbol{f},\bar{\boldsymbol{f}})\leq\varepsilon\}} Var(V^T(\boldsymbol{f},Z) - V(\bar{\boldsymbol{f}},Z)) \leq c_2\varepsilon^{\beta}.$$

These assumptions describe local smoothness of $|e(\boldsymbol{f},\bar{\boldsymbol{f}})|$ and $Var(V^T(\boldsymbol{f},Z) - V(\bar{\boldsymbol{f}},Z))$. In particular, Assumption A describes a first moment relationship between the Bayes regret $|e(\boldsymbol{f},\bar{\boldsymbol{f}})|$ and $e_{V^T}(\boldsymbol{f},\boldsymbol{f}^*)$. Assumption B is a second moment condition over the neighborhood of $\bar{\boldsymbol{f}}$. The exponents α and β depend on the joint distribution of (X,Y).

We now define a complexity measure of a function space $\mathcal{F}$. Given any $\varepsilon > 0$, denote $\{(f_j^l, f_j^u)\}_{j=1}^m$ as an ε-bracketing function set of $\mathcal{F}$ if for any $f \in \mathcal{F}$, there exists an j such that $f_j^l \leq f \leq f_j^u$ and $\|f_j^l - f_j^u\|_2 \leq \varepsilon; j = 1,\cdots,m$, where $\|f\|_2 = (Ef^2)^{\frac{1}{2}}$ is the L_2-norm. Then the metric entropy with bracketing $H_B(\varepsilon,\mathcal{F})$ is the logarithm of the cardinality of the smallest ε-bracketing set for $\mathcal{F}$. Let $\mathcal{F}^V(t) = \{V^T(\boldsymbol{f},\boldsymbol{z}) - V(\boldsymbol{f}^*,\boldsymbol{z}) : \boldsymbol{f} \in \mathcal{F}, J(\boldsymbol{f}) \leq J_0 t\}$, where $J(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^K \|f_j\|^2$ and $J_0 = \max(J(\boldsymbol{f}^*),1)$.

Assumption C: For some constants $c_i > 0; i = 3,\ldots,5$ and $\varepsilon_n > 0$,

$$\sup_{t\geq 2}\phi(\varepsilon_n,s) \leq c_5 n^{1/2}, \quad \phi(\varepsilon_n,s) = \int_{c_3 L}^{c_4^{1/2}L^{\beta/2}} H_B^{1/2}(u,\mathcal{F}^V(s))du/L, \tag{7}$$

where $L = L(\varepsilon_n,\lambda,s) = \min(\varepsilon_n^2 + \lambda J_0(s/2-1),1)$.

Appendix B.

Proof of Theorem 1: The proof is the same as that of Liu and Shen (2006), and is omitted.

Proof of Lemma 1: When 0-1 loss is used, $l_{0-1}(Y,d(\boldsymbol{X})) = I\{Y \neq d(\boldsymbol{X})\}$. From the sequential decision rule described in Section 2, we know that $y = d(\boldsymbol{x})$ is equivalent to for every $t \in anc(y) \cup \{y\}$, $f_t(\boldsymbol{x}) \geq f_j(\boldsymbol{x}) : j \in sib(t)$. Furthermore, it is also equivalent to $\min\{u_{y,j} : u_{y,j} \in U(\boldsymbol{f}(\boldsymbol{x}),y) = \{u_{y,1},u_{y,2},\cdots,u_{y,k_y}\}\} \geq 0$. Therefore, $GE(d) = El_{0-1}(Y,d(\boldsymbol{X})) = EI(u_{min}(f(X),Y) < 0)$ follows.

Proof of Lemma 2: To construct bracket covering for $\mathcal{F}^V(t)$, note that $J(\boldsymbol{f}) \leq J_0 t$ implies $\frac{1}{2}\|f_j\|^2 \leq J_0 t;\ j = 1,\cdots,K$. Furthermore, consider a pairwise difference $f_j - f_{j'}$ with $f_j \in \mathcal{F}_j(t)$ and $f_{j'} \in \mathcal{F}_{j'}(t)$. Let $\{(f_j^{i,l}, f_j^{i,u})_i\}$ be a set of an ε-bracket functions for $\mathcal{F}_j(t)$ in that for any $f_j \in \mathcal{F}_j(t)$, there exists an i such that $f_j^{i,l} \leq f_j \leq f_j^{i,u}$ with $\|f_j^{i,u} - f_j^{i,l}\|_2 \leq \varepsilon;\ j = 1,\cdots,K$. Now construct a set of brackets for $\mathcal{F}^V(t)$. Define $g^u = \max_{\{j'\in sib(j),j\in anc(y)\cup\{y\}\}} v\big(f_j^{i,l} - f_{j'}^{i',u}\big)$ and $g^l = \max_{\{j'\in sib(j),j\in anc(y)\cup\{y\}\}} v\big(f_j^{i,u} - f_{j'}^{i',l}\big)$, where $v(t)$ is $(1-t)_+$ for HSVM and $\psi(t)$ for HPSI. By construction,

$$T \wedge g^l \leq V^T(\boldsymbol{f},\boldsymbol{z}) = T \wedge \max\{v(f_j - f_{j'}) : j' \in sib(j), j \in anc(y)\cup\{y\}\} \leq T \wedge g^u$$

since $h^T(t) = T \wedge t$ is non-decreasing in t, where $z = (x, y)$. By Lipschitz continuity of $h^T(t)$ in t, $0 \le (T \wedge g^u - T \wedge g^l) \le g^u - g^l$, implying

$$\|T \wedge g^u - T \wedge g^l\|_2 \le \|g^u - g^l\|_2 \le \sum_{\{j' \in sib(j), j \in anc(y) \cup \{y\}\}} \|(f_j^{i,u} - f_{j'}^{i',l}) - (f_j^{i,l} - f_{j'}^{i',u})\|_2 \le 2c(\mathcal{H})\varepsilon,$$

with $c(\mathcal{H}) = \sum_{j=0}^{K} \frac{|chi(j)|(|chi(j)|-1)}{2}$ be the total number of sibling pairs (j, j') in $\mathcal{H}$. It follows that $H_B(2c(\mathcal{H})\varepsilon, \mathcal{F}^V(t)) \le H_B(2c(\mathcal{H})\varepsilon, \mathcal{F}_1(t))$. The desired result then follows.

To prove that $c(\mathcal{H}) \le k(k-1)/2$, we count the total number of different paths from the root to a leaf node. On one hand, given each non-leaf node j, there is only one path from the root to the node j but when there are additional $|chi(j)| - 1$ paths from the root to its children. An application of this recursively yields that there are $1 + \sum_{j \notin \mathcal{L}} (|chi(j)| - 1)$ paths from the root of the k leaf nodes. On the other hand, by definition, there are k different paths corresponding to k leaf nodes. Consequently, $k = 1 + \sum_{j \notin \mathcal{L}} (|chi(j)| - 1)$. For $j \notin \mathcal{L}$, $|chi(j)| - 1 \ge 0$. Then

$$\sum_{j \notin \mathcal{L}} (|chi(j)| - 1)^2 \le \Big(\sum_{j \notin \mathcal{L}} (|chi(j)| - 1) \Big)^2 = (k-1)^2.$$

This implies

$$2c(\mathcal{H}) = \sum_{j \notin \mathcal{L}} (|chi(j)| - 1)^2 + \sum_{j \notin \mathcal{L}} (|chi(j)| - 1) \le (k-1)^2 + k - 1 = k(k-1).$$

This completes the proof.

Proof of Lemma 3: Without loss of generality, assume that the membership is mutually exclusive for the first k classes. The 0-1 loss over K non-exclusive membership classes can be expressed as $\max_{t=1}^{K}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t))$, which is the disagreement between the value of Y and that of $d(X)$ in $\mathcal{H}$. By assumption, if

$$\max_{t=1}^{k}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) = 0,$$

then $\max_{t=k+1}^{K}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) = 0$. On the other hand,

$$\max_{t=1}^{K}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) \ge \max_{t=1}^{k}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)),$$

which implies that $\max_{t=1}^{K}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) = 1$ when $\max_{t=1}^{k}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) = 1$. Consequently

$$l_{0-1}(Y, d(X)) = \max_{t=1}^{k}(I(Y = t, d(X) \ne t) + I(Y \ne t, d(X) = t)) = \sum_{t=1}^{k} I(d(X) \ne t) I(Y = t)$$

by exclusiveness of the membership. Finally

$$\begin{aligned}
\bar{d}(x) &= \operatorname*{argmin}_{j=1}^{k} E l_{0-1}(Y, d(X) = j) | X = x) \\
&= \operatorname*{argmin}_{j=1}^{k} \sum_{t=1}^{k} P(Y = t | X = x) I(t \ne j) = \operatorname*{argmin}_{j=1}^{k} \sum_{t \ne j, t=1}^{k} P(Y = t | X = x) \\
&= \operatorname*{argmin}_{j=1}^{k} \Big(1 - P(Y = j | X = x)\Big) = \operatorname*{argmax}_{j=1}^{k} P(Y = j | X = x).
\end{aligned}$$

This completes the proof.

Proof of Lemma 4: The decision function $\bar{d}(\boldsymbol{x})$, which minimizes $E(l_{0-1}(Y,d(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x})$ for any $\boldsymbol{x}$, thus minimizing its expectation $El_{0-1}(Y,d(\boldsymbol{X}))$.

For $l_\Delta(Y,d(X))=|anc(Y)\triangle anc(d(\boldsymbol{X}))|$, let $m(j_1,j_2)$ to be $|anc(j_1)\triangle anc(j_2)|$. First note that we have a a length K(size of the tree) vector of bits for each class after introducing the binary 0-1 coding for each node including the ancestor nodes. Therefore $m(\cdot,\cdot)$ satisfies the triangle inequality since it is equivalent to the Hamming distance.

In what follows, we prove that $E(l_\Delta(Y,\bar{d}(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x})\le E(l_\Delta(Y,d(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x})$ for any $\boldsymbol{x}$ and classifier $d(\boldsymbol{x})$. Let $\hat{y}=\bar{d}(\boldsymbol{x})$. By the triangle inequality, $m(y,d(\boldsymbol{x}))-m(y,\hat{y})\ge -m(d(\boldsymbol{x}),\hat{y})$ for any y. Note that $m(\hat{y},\hat{y})=0$ and $m(\hat{y},d(\boldsymbol{x}))=m(d(\boldsymbol{x}),\hat{y})\ge 0$. Then

$$\begin{aligned}
& E\Big(l_\Delta\big(Y,d(\boldsymbol{X})\big)-l_\Delta\big(Y,\bar{d}(\boldsymbol{X})\big)\big|\boldsymbol{X}=\boldsymbol{x}\Big)=E\Big(m\big(Y,d(\boldsymbol{x})\big)-m(Y,\hat{y})\big|\boldsymbol{X}=\boldsymbol{x}\Big)\\
= \;& E\Big(\Big(m\big(Y,d(\boldsymbol{x})\big)-m(Y,\hat{y})\Big)\big(I(Y=\hat{y})+I(Y\neq\hat{y})\big)\Big|\boldsymbol{X}=\boldsymbol{x}\Big)\\
= \;& E\Big(\Big(m\big(\hat{y},d(\boldsymbol{x})\big)-m(\hat{y},\hat{y})\Big)I(Y=\hat{y})+\Big(m\big(Y,d(\boldsymbol{x})\big)-m(Y,\hat{y})\Big)I(Y\neq\hat{y})\Big|\boldsymbol{X}=\boldsymbol{x}\Big)\\
\ge \;& E\Big(m\big(\hat{y},d(\boldsymbol{x})\big)I(Y=\hat{y})|\boldsymbol{X}=\boldsymbol{x}\Big)-E\Big(m\big(d(\boldsymbol{x}),\hat{y}\big)I(Y\neq\hat{y})\big|\boldsymbol{X}=\boldsymbol{x}\Big)\\
= \;& m\big(\hat{y},d(\boldsymbol{x})\big)\big(P(Y=\hat{y}|\boldsymbol{X}=\boldsymbol{x})-P(Y\neq\hat{y}|\boldsymbol{X}=\boldsymbol{x})\big)\ge 0.
\end{aligned}$$

The last inequality follows from the fact that $\hat{y}=\text{argmax}_{j\in\mathcal{L}}P(Y=j|\boldsymbol{X}=x)$ and $P(Y=\hat{y}|\boldsymbol{X}=\boldsymbol{x})\ge 1/2\ge P(Y\neq\hat{y}|\boldsymbol{X}=\boldsymbol{x})$ by the assumption of dominating class. The desired result then follows.

Proof of Lemma 5: We prove the case of $v(z)=(1-z)_+$ for HSVM. Denote by $\hat{\boldsymbol{f}}(\boldsymbol{x})$ a minimizer of $E(v(u_{min}(\boldsymbol{f}(\boldsymbol{X}),Y))|\boldsymbol{X}=\boldsymbol{x})$ for any $\boldsymbol{x}$. At a given $\boldsymbol{x}$, without loss of generality, assume $p_j(\boldsymbol{x})>0$; $\forall 1\le j\le k$. Let $\hat{j}=d^H(\hat{\boldsymbol{f}}(\boldsymbol{x}))$ and $\hat{u}=u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),\hat{j})$. By definition, $\hat{f}_{j'}(\boldsymbol{x})-\hat{f}_{j''}(\boldsymbol{x})\ge\hat{u}\ge 0$, $\forall j'\in anc(\hat{j})$ and $j''\in sib(j')$. First consider the case of $\hat{u}>0$. For all other leaf node $j\neq\hat{j}$, there exists $j_a\in anc(j)$ and $\hat{j}_a\in anc(\hat{j})$ such that $j_a\in sib(\hat{j}_a)$. Then $u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),j)\le\hat{f}_{j_a}(\boldsymbol{x})-\hat{f}_{\hat{j}_a}(\boldsymbol{x})\le -u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),\hat{j})=-\hat{u}$, by the fact that $u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),\hat{j})\le -(\hat{f}_{j_a}(\boldsymbol{x})-\hat{f}_{\hat{j}_a}(\boldsymbol{x}))$. Now we prove the equality of $u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),j)\le -\hat{u}$ holds through construction of $\boldsymbol{f}'$: $f'_{j'}(\boldsymbol{x})-f'_{j''}(\boldsymbol{x})=\hat{u}$, and $f'_j(\boldsymbol{x})=0,\forall j\notin sib\circ anc(\hat{j})$. By construction, $u_{min}(\boldsymbol{f}'(\boldsymbol{x}),j)=-\hat{u}$, for $1\le j\le k$, $j\neq\hat{j}$, and $u_{min}(\boldsymbol{f}'(\boldsymbol{x}),\hat{j})=\hat{u}$. Note that

$$\begin{aligned}
& E(v(u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{X}),Y))|\boldsymbol{X}=\boldsymbol{x})-E(v(u_{min}(\boldsymbol{f}'(\boldsymbol{X}),Y))|\boldsymbol{X}=\boldsymbol{x})\\
= \;& \sum_{1\le j\le k;j\neq\hat{j}}p_j(\boldsymbol{x})(v(u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),j))-v(-\hat{u}))\ge 0.
\end{aligned}$$

By the fact that $\hat{\boldsymbol{f}}(\boldsymbol{x})$ is the minimizer, for $1\le j\le k$, $j\neq\hat{j}$, $v(u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),j))-v(-\hat{u})=0$, then $u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}),j)=-\hat{u}$. Moreover, for the Bayes rule $\bar{d}(\boldsymbol{x})$, if $\hat{j}\neq\bar{d}(\boldsymbol{x})$, we construct $\boldsymbol{f}^*$ such that $u_{min}(\boldsymbol{f}^*(\boldsymbol{x}),\bar{d}(\boldsymbol{x}))=\hat{u}$, and $u_{min}(\boldsymbol{f}^*(\boldsymbol{x}),j)=-\hat{u}$, for any leaf node $j\neq\bar{d}(\boldsymbol{x})$, similar as above. This implies that

$$\begin{aligned}
& E(v(u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{X}),Y))|\boldsymbol{X}=\boldsymbol{x})-E(v(u_{min}(\boldsymbol{f}^*(\boldsymbol{X}),Y))|\boldsymbol{X}=\boldsymbol{x})\\
= \;& (p_{\hat{j}}(\boldsymbol{x})v(\hat{u})+p_{\bar{d}(\boldsymbol{x})}(\boldsymbol{x})v(-\hat{u}))-(p_{\hat{j}}(\boldsymbol{x})v(-\hat{u})+p_{\bar{d}(\boldsymbol{x})}(\boldsymbol{x})v(\hat{u}))\\
= \;& (p_{\hat{j}}(\boldsymbol{x})-p_{\bar{d}(\boldsymbol{x})}(\boldsymbol{x}))(v(\hat{u})-v(-\hat{u}))>0,
\end{aligned}$$

because $p_{\hat{j}}(\boldsymbol{x}) < p_{\bar{d}(\boldsymbol{x})}(\boldsymbol{x})$ and $\hat{u} > 0$. This contradicts the fact that $\hat{\boldsymbol{f}}(\boldsymbol{x})$ is the minimizer. Consequently, $\hat{j} = \bar{d}(\boldsymbol{x})$. For the case of $\hat{u} = 0$, it can be shown that $u_{min}(\hat{\boldsymbol{f}}(\boldsymbol{x}), j) = 0, \forall j = 1, \cdots, k$, and $\hat{f}_j(\boldsymbol{x}) = 0, \forall j = 1, \cdots, K$, which reduces to the trivial case $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{0}$.

For HPSI, the proof is the same as that of Theorem 2 in Liu and Shen (2006), and is omitted.

Proof of Theorem 2: The proof is similar to that in Shen and Wang (2007) and is omitted.

Proof of Corollary 1: The $O_p(\cdot)$ result follows from the exponential bound in Theorem 2. To see the risk result, note that

$$\delta_n^{-2\alpha} E e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = \int_0^\infty P(e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) > (\delta_n^{2\alpha} t)^{1/2\alpha}) dt.$$

The result then follows.

The primal and the dual of (2) for HSVM: The primal and the dual for HSVM can be obtained from those of HPSI below, with $\nabla \hat{\boldsymbol{w}}_j^{(m-1)} = 0$ and $\nabla \hat{b}_j^{(m-1)} = 0$; $j = 1, \cdots, K$.

The primal and the dual of (3) for HPSI: The primal of (3) is

$$\underset{\boldsymbol{f}}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^K \|\boldsymbol{w}_j\|^2 + C \sum_{i=1}^n \xi_i - \sum_{j=1}^K \langle \nabla \hat{\boldsymbol{w}}_j^{(m-1)}, \boldsymbol{w}_j \rangle - \sum_{j=1}^K \langle \nabla \hat{b}_j^{(m-1)}, b_j \rangle, \tag{8}$$

subject to $\xi_i > 0$, $(f_j(x_i) - f_t(x_i)) + \xi_i \geq 1$, $(j,t) \in Q(y_i) = \{(j,t) : t \in sib(j), j \in \{y_i\} \cup anc(y_i)\}$, and $\sum_{\{j \in chi(s), s \notin \mathcal{L}\}} f_j(\boldsymbol{x}_i) = 0$; $i = 1, \cdots, n$, $s = 1, \cdots, K$.

To solve (8), we employ the Lagrange multipliers: $\alpha_i \geq 0$, $\beta_{i,j,t} \geq 0$ and $\delta_{i,s} \geq 0$ for each constraint of (8). Then (8) is equivalent to:

$$\begin{aligned} \max_{\alpha_i, \beta_{i,j,t}, \delta_{i,s}} L \quad = \quad & \frac{1}{2} \sum_{j=1}^K \|\boldsymbol{w}_j\|^2 + C \sum_{i=1}^n \xi_i - \sum_{j=1}^K \langle \nabla \hat{\boldsymbol{w}}_j^{(m-1)}, \boldsymbol{w}_j \rangle - \sum_{j=1}^K \langle \nabla \hat{b}_j^{(m-1)}, b_j \rangle + \\ & \sum_{(j,t) \in Q(y_i): i = 1, \cdots, n} \beta_{i,j,t} \Big(1 - \big((\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle + b_j) - (\langle \boldsymbol{w}_t, \boldsymbol{x}_i \rangle + b_t) \big) - \xi_i \Big) \\ & - \sum_{i=1}^n \alpha_i \xi_i + \sum_{(i,s): i = 1, \cdots, n; s \notin \mathcal{L}} \delta_{i,s} \sum_{j \in chi(s)} (\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle + b_j). \end{aligned} \tag{9}$$

By letting the partial derivatives be zero, we have that

$$\frac{\partial L}{\partial \boldsymbol{w}_j} = 0, \frac{\partial L}{\partial \xi_i} = 0, \frac{\partial L}{\partial b_j} = 0; i = 1, \cdots, n, j = 1, \cdots, K. \tag{10}$$

implying that $\alpha_i \geq 0; i = 1, \cdots, n$, and

$$\sum_{(j,t) \in Q(y_i)} \beta_{i,j,t} \leq C; i = 1, \cdots, n. \tag{11}$$

After substituting (10) in (9), we obtain a quadratic form of L in terms of $\{\alpha_i, \beta_{i,j,t}, \delta_{i,s}\}$ Maximizing L subject to $\beta_{i,j,t} \geq 0$; $i = 1, \cdots, n$; $(j,t) \in Q(y_i)$, (10) and (11) yields the solution of $\{\alpha_i, \beta_{i,j,t}, \delta_{i,s}\}$. The solution of $\boldsymbol{w}_j$ and ξ_i's can be derived from (10). The solution of b_j is derived from Karush-Kuhn-Tucker's condition: $\beta_{i,j,t} \Big(1 - \big((\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle + b_j) - (\langle \boldsymbol{w}_t, \boldsymbol{x}_i \rangle + b_t) \big) - \xi_i \Big) = 0$, $\alpha_i \xi_i = 0$, and $\delta_{i,s} \sum_{j \in chi(s)} (\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle + b_j) = 0$, for all suitable i, j, t and s. In case of these conditions are not applicable to b_j's, we substitute the solution of $\boldsymbol{w}_j$'s in (8), and solve b_j's through linear programming.

References

L. An and P. Tao. Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. *J. Global Optimization*, 11:253–285, 1997.

K. Astikainen, L. Holm, S. Szedmak E. Pitknen, and J. Rousu. Towards structured output prediction of enzyme function. *BMC Proceedings*, 2(S4):S2, 2008.

B. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *Proc. Fifth Ann. Conf on Computat. Learning Theory Pittsburgh, PA*, pages 144–152, 1992.

L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. *CIKM-04, Washington, DC*, 2004.

N. Cesa-Bianchi and G. Valentini. Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *MLSB 09: The 3rd International Workshop on Machine Learning in Systems Biology 2009*, 2009.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. Regret bounds for hierarchical classification with linear-threshold functions. *Proc. the 17th Ann. Conf. on Computat. Learning Theory*, pages 93–108, 2004.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: Combining bayes with svm. *Proc. of the 23rd Int. Conf. on Machine Learning, ACM Press (2006)*, pages 177–184, 2006.

M. N. Davies, A. Secker, A. A. Freitas, M. Mendao, J. Timmis, and D. R. Flower. On the hierarchical classification of g protein-coupled receptors. *Bioinformatics*, 23(23):3113–3118, 2007.

O. Dekel, J. Keshet, and Y. Singer. An efficient online algorithm for hierarchical phoneme classification. *Proc. the 1st Int. Workshop on Machine Learning for Multimodal Interaction*, pages 146–158, 2004.

L. Dong, E. Frank, and S. Kramer. Ensembles of balanced nested dichotomies for multi-class problems. *Lecture Notes in Computer Science*, 3721/2005:84–95, 2005.

C. Gu. Multidimension smoothing with splines. *Smoothing and Regression: Approaches, Computation and Application*, 2000.

Y. Guan, C. Myers, D. Hess, Z. Barutcuoglu, A. Caudy, and O. Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2), 2008.

T. Hughes, M. Marton, A. Jones, C. Roberts, R. Stoughton, C. Armour, H. Bennett, E. Coffey, H. Dai, Y. He, M. Kidd, A. King, M. Meyer, D. Slade, P. Lum, S. Stepaniants, D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

T. Jaakkola, M. Diekhans, and D. Haussler. Using the fisher kernel method to detect remote protein homologizes. *In Proc. the Seventh Int. Conf. on Intelligent Systems for Molecular Biology*, pages 149–158, 1999.

T. Joachims. Text categorization with support vector machines: learning with many relevant features. *Proc. of the 10th European Conf. on Machine Learning (ECML1998)*, 1398:117–142, 1998.

A. N. Kolmogorov and V. M. Tihomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Mat. Nauk.*, 14:3–86, 1959. In Russian. English translation, *Ameri. Math. Soc. transl.* **2 , 17**, 277-364. (1961).

D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. *Proc. of the 10th European Conf. on Machine Learning (ECML1998)*, pages 4–15, 1998.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

S. Liu, X. Shen, and W. Wong. Computational development of ψ-learning. *Proc. SIAM 2005 Int. Data Mining Conf.*, pages 1–12, 2005.

Y. Liu and X. Shen. Multicategory ψ-learning. *J. Amer. Statist. Assoc.*, 101:500–509, 2006.

H. W. Mewes, D. Frishman, U. G'ldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. M'nsterkoetter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nuclerc Acids Res*, 30:31–34, 2002.

G. Obozinski, G. Lanckriet, C. Grant, M. Jordan, and W. Noble. Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6), 2008.

J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *J. Mach. Leaning Res.*, 7:1601–1626, 2006.

B. Shahbaba and R. Neal. Improving classification when a class hierarchy is available using a hierarchy-based prior. *Bayesian Analysis*, 2:221–238, 2007.

X. Shen and L. Wang. Generalization error for multi-class margin classification. *Electronic J. of Statist.*, 1:307–330, 2007.

X. Shen, G. Tseng, X. Zhang, and W. Wong. On ψ-learning. *J. Amer. Statist. Assoc.*, 98:724–734, 2003.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *Proc. the 21st Int. Conf. on Machine Leaning*, 2004.

G. Valentini and M. Re. Weighted true path rule: a multilabel hierarchical algorithm for gene function prediction. *The 1st International Workshop on learning from Multi-Label Data, ECML/PKDD 2009*, 2009.

V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

Y. Yang and X. Liu. A reexamination of text categorization methods. *Proc. the 22nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 42–49, 1999.

J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *J. Comput. and Graph. Statist.*, 14:185–205, 2005.

A. Zimek, F. Buchwald, E. Frank, and S. Kramer. A study of hierarchical and flat classification of proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 2008*, 2008.