On topics of multi-category classification with large margin based methods

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Huixin Wang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Xiaotong Shen, Adviser

January, 2009

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Xiaotong Shen, for his guidance, encouragement and patience through the development of this thesis. His profound understanding of numerous areas has been a great resource for me and has made the challenging Ph.D. research such a pleasant journey. Almost every visit to his office has turned into progress in my work. For me, he is not only an academic advisor, but also a mentor, a friend and inspiration. Dr. Xiaotong Shen, being passionate, energetic and intellectual, has set up a solid professional model that I will be pursuing.

I am thankful to Professor Dennis Cook for his serving as my defense committee chair. My thanks also go to Professor Charlie Geyer and Professor Wei Pan for their time and effort for reviewing my thesis, and for their valuable suggestion regarding my research. I am grateful to the faculty, staff and students in the School of Statistics for making my four years' study at Minnesota such a wonderful experience. I will be missing every single piece of time here.

Finally, I thank my parents and grandmother for their endless love and strongest support. They have always been accompanying me through the moments of frustration and sharing the pleasure with me.

Dedicated to my wife Zhi Huang and my parents

Abstract

In multi-class classification, the cost for misclassification may vary over each class. This is a situation in structured learning, where the focus is how to leverage dependency among different classes to enhance the performance of classification that ignores such dependency structure. Examples include hierarchical classification, sequence alignment, and natural language processing, among others. This paper develops a framework for multi-class margin classification with un-equal (equal) cost. Within the framework, structured learning is formulated, where the dependency is taken into account through the cost of misclassification. This framework is implemented for support vector machines. An application to hierarchical classification is discussed. In addition, some simulations are performed, indicating that the proposed methodology achieves the desired objective.

As a special case of multi-classification, in hierarchical classification, class label is structured in that each label value corresponds to one non-root node in a tree, where the inter-class relationship 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 ignoring such dependency. This is critical when the number of classes becomes large relative to the size of sample. This paper considers single-path hierarchical

classification, where only one path is permitted from the root to one 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 outperforms its competitors, particularly its flat counterpart. Finally, an application to gene function discovery is examined.

Table of Contents

List of tables

List of figures

Chapter 1

Introduction

Currently statistical learning is widely used in different areas in scientific and engineering studies. Large volume of data requires statistical methodologies to process it efficient. Different problems need statistical theory and methodology adept to different structure of data and combine the most information from the observations into statistical learning.

In statistical learning, large margin methods cover a wide range of newly developed methods. Although it is well examed for the binary classification case, there are many areas in multi-classification cases left open. For binary classification, each observation is labeled with a binary variable and the object of classification is to find the "best" labeling theme for any possible unlabeled observations under certain criterion. Statistically, "best" always leads to the optimization problems. For the multi-classification problems, unlike its binary counter part, it is hard to capture the relationship within all the possible classes. Although they are many different studies in multi-class case, different classes were mostly treated equally. In real problems, the relationship between different class labels may vary,

and it is critical to capture different structure of class label into classification. In this thesis, two general cases of multi-classification are studied as, un-equal weighted multi-classification, and hierarchical classification.

1.1 Hierarchical classification

In many applications, a hierarchy is used to organize objects of interest. For instance, biological functions of genes are often organized by a gene function annotation system such as MIPS (Mewes et al., 2002). This kind of system is hierarchical, where major categories are general and sub-categories are detailed. Over a hierarchy, hierarchical classification is performed. In hierarchical classification, the focus has been on how to leverage the inter-category (inter-class) relationship to enhance the performance of classification ignoring such dependency, known as flat classification. This paper develops a large margin approach for single-path hierarchical classification with exclusive class membership.

One traditional approach for hierarchical classification is treating it as multi-class classification, ignoring the inter-class dependency. This approach is not expected to perform well, especially when the number of classes becomes large. Another approach is sequential classification, where a flat classifier is trained locally within a hierarchy. As a result, the classifier is not well trained due to a small training sample locally. Relevant reference can be found in Yang and Liu (1999) for nearest neighbor, Lewis (1998) for naive Bayes, and Joachims (1998) for support vector machines. Recently, the importance of utilizing a hierarchical structure has been recognized in classification, c.f., Hofmann et al. (2003),

Cai and Hofmann (2004), Dekel et al. (2004), and Shahbaba and Neal (2007), among others. Despite progress, problems remain with respect to how to integrate the hierarchical structure into classification for better performance.

It is widely accepted that analysis of microarray data is a efficient way for biological gene function discovery. Classifying gene function through the microarray data has been well studied in many publications, as in Brown and Botstein (1999); Kerr et a (2000); Tusher et al. (2001); Efron et al. (2001); Newton et al. (2003); to name just a few. We use yeast *S. cerevisiae* gene microarray data used in Hughes et al.. (2000) to demonstrate the performance of hierarchical classification in gene oncology discovery.

Some relevant works can be found in, for example, Dumais and Chen (2000); Granitzer (2003); Koller and Sahami (1997); McCallum et al. (1998); Mladenic (1998); Ruiz and Srinivasan (2002); Sun and Lim (2001). In classification, one critical issue is how to utilize the hierarchical structure to further improve upon the standard multi-class classification without using the hierarchical information between different classes.

1.2 Unequal loss multi-classification

In binary classification, support vector machines (SVM; Boser, Guyon, and Vapnik, 1992; Cortes and Vapnik, 1995) and other margin based methods have been effective in delivering high performance. In multi-class classification with equal cost, large margin classifiers have been successful in many applications as well. These large margin classifiers include various generalizations of the same binary SVM (Weston and Watkins, 1998; Bredensteiner and Bennett, 1999; Guermuer, 2002; Cramer and Singer, 2002; and Liu and Shen, 2006),

import vector machines (IVM, Zhu and Hastie, 2005), and multi-class ψ-learning (Liu and Shen, 2006). Despite progress, multi-class large margin classification with un-equal cost, targeting at structured learning, remains less explored.

Structured learning occurs when labeling is structured or dependent. In hierarchical classification, for instance, gene functions are classified and predicted through microarray data. Usually, gene functions are categorized and organized by a tree with each class presenting a node in the tree. For example, the MIPS database (Mewes et al., 2002) is a system of 1411 function categories with six levels in a tree. In such a situation, a tree structure provides additional information that needs to be incorporated to enhance the predictive accuracy of classification. In this paper, this will be treated through multi-class classification.

With regard to large margin classification with un-equal cost, only SVM has been considered by far. For the binary case, Lin, Lee and Wahba (2002) introduced a nonstandard SVM, where a weighted SVM was developed through different costs for the positive and negative classes. For the multi-class case, Lee, Lin and Wahba (2004) discussed a nonstandard situation of multi-class SVM, defined by their own generalized hinge loss instead of other aforementioned generalized hinge losses. Tsochanridis et al. (2004) also adopted a similar approach.

Chapter 2

Hierarchical classification

2.1 Introduction

This chapter develops a large margin method for hierarchical classification with exclusive class membership. Toward this end, we introduce a new concept of margins with respect to hierarchy as a means to fully integrate the corresponding inter-class dependency into classification. As a result, the classification problem's complexity is reduced when the inter-class dependency is strong, translating into better classification performance. In contrast to the aforementioned approaches, our method trains a classifier globally while making a decision sequentially, and casts a sequential decision rule into the framework of large margins. The proposed method is implemented for support vector machines (SVM, Boser and Vapnik, 1992) and ψ-learning (Shen et. al, 2003) through quadratic and difference convex programming.

To examine the operating characteristics, we perform numerical simulations and study

the generalization performance of the proposed method. The numerical and theoretical results suggest that the proposed method achieves the desired objective of delivering higher performance than its flat counterpart and one strong competitor. Furthermore, we find that a stronger inter-class relation tends to lead to better improvement over its flat counterpart.

This chapter is organized as follows. Section 2 discusses the generalization error and decision rules with respect to hierarchy. Section 3 introduces the proposed method and develops computational tools. Section 4 performs simulation studies and presents an application to gene function discovery. Section 5 is devoted to theoretical investigation of the proposed method. Section 6 discusses the method, followed by technical details in the Appendix.

2.2 Single-path hierarchical classification

A rooted tree is a graph with nodes connected by directed paths from the root, where directed edge $i \to j$ indicates the parent-child relationship from i to j. In a tree hierarchy $\mathcal{H}$, the root 0 is a common ancestor of the other nodes, together with $K = |\mathcal{H}| - 1$ non-root nodes consisting of k leaf and $(K - k)$ non-leaf nodes, where a non-leaf node is an ancestor of a leaf one, and $|\mathcal{H}|$ is the size of $\mathcal{H}$. Now define, for each node $t \in \{1, \cdots, K\}$, $par(t)$, $chi(t)$ $sib(t)$, $anc(t)$ and $sub(t)$ to be 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. In our setting, $par(t)$, $chi(t)$ and $sib(t)$ are allowed to be empty with ϕ indicating the empty set.

Assume, without loss of generality, that $par(t) \leq 1$ because multiple parents of any node is not permitted for a tree that uniquely defines a single path from the root to one node.

In single-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 K non-root nodes in a tree. Here Y is structured in that the inter-class relationship among K classes is specified by parent-child relations defined by $\mathcal{H}$, that is, the classification region of class j is a subset of that of class i if there exists a parent-child connection from the corresponding node i to node j in $\mathcal{H}$. Moreover, the class membership is exclusive for all siblings, which is equivalent to single-path from the root to any leaf node, that is, each value of Y corresponding to one and only one of the k leaf and its ancestor nodes.

2.2.1 Sequential decision rule

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; $j = 1, \cdots, K$, mapping from $\mathbb{R}^q$ onto $\mathbb{R}^1$, represents class j. Then it 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 define a top-down decision rule $d^H(\boldsymbol{f}(\boldsymbol{x}))$ with respect to $\mathcal{H}$ through $\boldsymbol{f}$. From the root of the tree, we examine 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 $t \in chi(j)$ if $chi(j) \neq \phi$, and assign $\boldsymbol{x}$ to j if $chi(j) = \phi$.

This top-down rule is sequential, and yields exclusive membership for classes represented by siblings from the same parent. In particular, for each parent t, its children set $chi(t)$ yields a partition of the classification region defined by t. This permits an observation staying at a parent when one of its children is treated as the parent, see Figure 2.6

for such an example.

2.2.2 Losses and generalization error with respect to hierarchy

Three types of losses have been proposed for hierarchical classification. Given a general classifier $d(\boldsymbol{x})$, the 0-1 loss $l_{0-1}(Y, d(\boldsymbol{X}))$ is $I(Y \neq d(\boldsymbol{X}))$. The symmetric difference loss (Tsochantaridis et al., 2004) $l_{\Delta}(Y, d(\boldsymbol{X}))$ is $|anc(Y) \triangle anc(d(\boldsymbol{X}))|$, where $\triangle$ denotes the symmetric difference of two sets. The H-loss (Cesa-Bianchi et al., 2004) is $l_H(Y, d(\boldsymbol{X})) = c_j$, with j the highest node yielding the disagreement between Y and $d(\boldsymbol{X})$ in a tree, which ignores any errors occurring at lower levels. Two common choices of c_j's have been suggested, leading to the subtree based H-loss l_{sub}:

$$c_j = |sub(j)|/K;\ j = 1, \ldots, K, \tag{2.1}$$

and the siblings based H-loss l_{sib}:

$$c_0 = 1,\ c_j = c_{par(j)}/|sib(j)|;\ j = 1, \ldots, K. \tag{2.2}$$

Here l_H penalizes the disagreement at a parent while tolerating subsequent errors at offsprings.

Given a loss l, the generalization error (GE) of classifier $d(\boldsymbol{x})$ is defined as $GE(d) = El(Y, d(\boldsymbol{X}))$, which is used to measure the generalization accuracy of $d(\boldsymbol{x})$.

To evaluate 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 2.1 for K-class classification with non-exclusive membership, where only $k < K$ classes have exclusive membership, determining the class membership of the other $K - k$ classes.

Lemma 2.1 *In K-class hierarchical classification with non-exclusive membership, assume that the k exclusive membership classes uniquely determine the membership of the other $K-k$ non-exclusive classes. Precisely, for any $t \in E$ and $\tilde{t} \notin E$, if $\{Y=\tilde{t}\} \nsubseteq \{Y=t\}$ then $\{Y=\tilde{t}\} \subseteq \{Y \neq t\}$, and vice verse, where E is the set of exclusive membership classes. The Bayes classifier $\bar{d}(\boldsymbol{x}) = \text{argmax}_{j \in E} P(Y=j|\boldsymbol{X}=\boldsymbol{x})$.*

Based on the result of Lemma 2.1, we define Fisher consistency with respect to $\mathcal{H}$ in hierarchical classification.

Definition 1 *In hierarchical classification, denote by E the set of classes corresponding to the leaf nodes in a tree. Loss $l(\cdot,\cdot)$ is said to be Fisher-consistent with respect to $\mathcal{H}$ if a global minimizer of $El(Y, d(\boldsymbol{X}))$ over all possible $d(\boldsymbol{x})$'s is $\bar{d}(\boldsymbol{x})$.*

Lemma 2.2 *Loss l_{0-1} is Fisher-consistent with respect to $\mathcal{H}$; so is l_Δ in presence of a dominating leaf node class in that for any $\boldsymbol{x} \in S$ there exists a leaf node class j such that $P(Y=j|\boldsymbol{X}=x) > 1/2$.*

Unfortunately, l_{sub} and l_{sib} are generally not Fisher-consistent with respect to $\mathcal{H}$ as evident from the following example. Consider hierarchical classification with a tree consisting of three leaf nodes 1-3, where nodes 1 and 2 are children of node 4, and nodes 3 and 4 are children of root node 0. Let $P(Y=j|\boldsymbol{X}=\boldsymbol{x})$ is $0.31, 0.29, 0.4$; $j=1,2,3$ for all $\boldsymbol{x} \in S \subset \mathbb{R}^q$. Then the Bayes classifier $\bar{d}(\boldsymbol{x}) = 3$ for any $\boldsymbol{x}$; because $P(Y=3|\boldsymbol{X}=\boldsymbol{x})$ is the highest at any point $\boldsymbol{x}$. On the other hand, to minimize the risk $El(Y, d(\boldsymbol{X}))$ with $l = l_{sub}$, note that $El(Y, d(\boldsymbol{X})) = E(E(l(Y, d(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}))$. By the definition of l_H with c_j in (2.1), $l_{sub}(1,2) = l_{sub}(2,1) = 0.5$, $l_{sub}(1,3) = l_{sub}(3,1) = l_{sub}(2,3) = l_{sub}(3,2) = 1$.

Then $E(l(Y, d(\boldsymbol{X}))|\boldsymbol{X} = \boldsymbol{x}) = \sum_{j=1}^{3} P(Y = j|\boldsymbol{X} = \boldsymbol{x})l(j, d(\boldsymbol{x}))$, which is $0.545, 0.555, 0.6$ when $d(\boldsymbol{x})$ is $1, 2, 3$. This implies that $d(\boldsymbol{x}) = 1$ for any $\boldsymbol{x}$ is the global minimizer of $El_{sub}(Y, d(\boldsymbol{X}))$. Similarly, $d(\boldsymbol{x}) = 1$ for any $\boldsymbol{x}$ is also the global minimizer of $El_{sib}(Y, d(\boldsymbol{X}))$ because $E(l_{sib}(Y, d(\boldsymbol{X}))|\boldsymbol{X} = \boldsymbol{x})$ is $0.273, 0.278, 0.3$ when $d(\boldsymbol{x}) = 1, 2, 3$. This says that l_{sub} and l_{sib} are not Fisher-consistent with respect to $\mathcal{H}$.

As shown in Lemma 2.2, l_{0-1} and l_{Δ} are Fisher-consistent with respect to $\mathcal{H}$. Through the top-down rule, they take into account the hierarchical structure. In single-path hierarchical classification, l_{0-1} penalizes only the disagreement at leaf nodes, whereas l_{Δ} does at both leaf and non-leaf nodes. Depending on the context of applications, either l_{0-1} or l_{Δ} may be preferred. In our target application—gene function prediction, we are interested in discovering gene functions precisely. This means that partial correctness is not adequate. For instance, identifying a gene in the "Disease, virulence and defense" functional category does not help much to identify a gene in "Detoxification" category, even these two categories share the same parent "Cell rescue, defense and virulence". In this case, l_{0-1} is more preferable than l_{Δ}, although the latter can distinguish "partial correctness" from "total correctness".

On the basis of the foregoing discussion, we shall implement our method based on l_{0-1}, although the method is also applicable to other losses such as l_{Δ}. In our numerical evaluation, we shall use all the four losses l_{0-1}, l_{Δ}, l_{sub} and l_{sib}.

2.3 Proposed method

In single-path 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 such that $\bar{d}(\boldsymbol{x}) = d^H(\boldsymbol{f}(\boldsymbol{x}))$. In our context, we define $\bar{\boldsymbol{f}}$ as follows: for each j, $\bar{f}_j(\boldsymbol{x}) = \max_{t:chi(t)=\phi, t\in sub(j)} P(Y=t|\boldsymbol{X}=\boldsymbol{x})$ if $chi(j) \neq \phi$ and $\bar{f}_j(\boldsymbol{x}) = P(Y=j|\boldsymbol{X}=\boldsymbol{x})$ if $chi(j) = \phi$. To estimate $\bar{\boldsymbol{f}}$, we design a large margin classifier to deliver higher generalization performance than its flat counterpart based on k leaf node classes. This is achieved by introducing a new concept of functional and geometric margins with respect to $\mathcal{H}$.

2.3.1 Margins with respect to $\mathcal{H}$

For motivation, consider standard k-class classification. For classification, any two of the classes are compared through the argmax decision rule, which amounts to a total of $k(k-1)/2$ comparisons, c.f., Liu and Shen (2006). In our case, the required number of comparisons can be smaller because only siblings need to be compared when the top-down rule is employed. For instance, in a situation described by Figure 2.4, with four leaf and two non-leaf nodes, only three sibling comparisons are necessary: node 1 versus node 2, node 3 versus node 4, and node 5 versus node 6. This is in contrast to six comparisons in k-class classification with $k=4$. See Lemma 2.3 for a theoretical result. Now, to compare different siblings, we define $\boldsymbol{u}(\boldsymbol{f}(\boldsymbol{x}), y)$ as $\boldsymbol{u}(\boldsymbol{f}(\boldsymbol{x}), y) = \bigcup_{\{t\in anc(y)\cup\{y\}\}}\{f_t - f_{sib(t)}\} = \bigcup_{\{t\in anc(y)\cup\{y\}\}}\{f_t - f_j : j \in sib(t), t \in anc(y)\cup\{y\}\} \equiv (u_{y,1}, u_{y,2}, \cdots, u_{y,k_y})$ with k_y the length of vector $\boldsymbol{u}(\boldsymbol{f}(\boldsymbol{x}), y)$. This vector compares any class t against $sib(t)$ where t is any

non-root ancestor of class y or y itself. Then we define the generalized functional margin with respect to $\mathcal{H}$ as $u_{min}(\boldsymbol{f}(\boldsymbol{x}), y) = \min\{u_{y,j} : u_{y,j} \in \boldsymbol{u}(\boldsymbol{f}(\boldsymbol{x}), y)\}$. This definition reduces to that in multi-class margin classification when no inter-class relationship is specified.

The hierarchical structure specified by $\mathcal{H}$ can be summarized as the direct parent-child relation and the associated indirect relations, for classification. They are fully integrated into our framework. Whereas the top-down rule specifies $\mathcal{H}$, $u_{min}(\boldsymbol{f}(\boldsymbol{x}), y)$ captures the relations in (2.3) below. As a result, a classification problem's dimension is reduced through $\mathcal{H}$, leading to better generalization. This aspect will be confirmed by our numerical results and detailed calculations for the classification function space in Lemma 2.3.

Many losses can be written as a function of $u = u_{min}(\boldsymbol{f}(\boldsymbol{x}), y)$. For instance, $l_{0-1} = 1 - I\{d(\boldsymbol{x}) = y\} = 1 - I\{f_t - f_{anc(t)} \geq 0, \forall t \in anc(y) \bigcup \{y\}\} = 1 - I\{u_{min} \geq 0\} = I\{u_{min} < 0\}$. And for l_Δ, similar result is shown in Chapter 4. However, they are intractable. For this reason, many surrogate losses have been introduced. In hierarchical classification, given functional margin u, we say that a loss $v(\cdot)$ is a margin loss if it is a function of u. Moreover, it is large margin if $v(u)$ is nonincreasing in u. These definitions allow us to utilize the existing two-class surrogate losses. In the two-class case, the hinge loss $v(u) = (1-u)_+$ and the logistic loss $v(u) = \log(1 + e^{-u})$, c.f., Zhu and Hastie (2005), are two convex large margin losses, whereas ψ-loss $v(u) = \psi(u)$, 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, c.f., Shen et al. (2003), is a nonconvex large margin loss.

For hierarchal classification, we propose our cost function for any margin loss v as

follows:

$$s(\boldsymbol{f}) = J(\boldsymbol{f}) + C\sum_{i=1}^{n} v(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i), y_i)), \tag{2.3}$$

subject to $\sum_{\{t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_t(\boldsymbol{x}) = 0$ for removing the redundancy among f_t's, $j = 1, \cdots, K$; $\forall \boldsymbol{x} \in S$, the domain of X_1. Here $J(\boldsymbol{f})$ is a penalty to be introduced, and $C > 0$ is a tuning parameter regularizing the trade-off between $J(\boldsymbol{f})$ and training. Minimizing (2.3) with respect to $\boldsymbol{f} \in \mathcal{F}$, a candidate function space, yields an estimate $\hat{\boldsymbol{f}}$, thus classifier $d^H(\hat{\boldsymbol{f}}(\boldsymbol{x}))$.

To define $J(\boldsymbol{f})$, we introduce the geometric margin with respect to $\mathcal{H}$ in the L_2-norm. The geometric margin in other norms such as the L_p-norm can be defined similarly. Now consider a generic representation of $\boldsymbol{f}$: $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T\tilde{\boldsymbol{x}} + b_j$; $j = 1, \cdots, K$, where $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ and $\tilde{\boldsymbol{x}} = (\mathcal{K}(\boldsymbol{x}_1, \cdot), \mathcal{K}(\boldsymbol{x}_2, \cdot), \cdots, \mathcal{K}(\boldsymbol{x}_n, \cdot))^T$ for linear and kernel learning. The geometric margin is defined as $\min_{\{(t,j):t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} \gamma_{j,t}$, where $\gamma_{j,t} = \frac{2}{\|f_j - f_t\|^2_{\mathcal{K}}}$ is the usual separation margin defined for two classes j and $t \in sib(j)$ sharing the same parent, representing the vertical L_2-distance between two parallel hyperplanes $f_j - f_t = \pm 1$, c.f., Wang and Shen (2007). Here $\|f_j\|^2_{\mathcal{K}}$ 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 a $n\times n$ kernel matrix. Ideally, $J(\boldsymbol{f})$ is $\max_{\{(t,j):t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} \gamma_{j,t}^{-1} = \max_{\{(t,j):t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} \frac{\|f_j - f_t\|^2}{2}$, the inverse of the geometric margin. However, it is less tractable numerically. In implementation, we work with its upper bound $J(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^{K}\|f_j\|^2_{\mathcal{K}}$ instead.

The sum-to-zero constraint $\sum_{\{t:t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_t(\boldsymbol{x}) = 0$ involves $\forall \boldsymbol{x} \in S$. However, it can reduce to the constraints on $\{\boldsymbol{x}_i\}_{i=1}^n$, which is an analogy of that of Liu and Shen (2006).

Theorem 1 *Assume that* $\{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \cdots, \tilde{\boldsymbol{x}}_n\}$ *spans* $\mathbb{R}^q$*. Then minimizing (2.3) subject to* $\sum_{\{t:t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_j(\boldsymbol{x}) = 0$*;* $j = 1, \cdots, K$*,* $\forall \boldsymbol{x} \in S$*, is equivalent to that subject to* $\sum_{\{t:t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_j(\boldsymbol{x}_i) = 0$*;* $j = 1, \cdots, K$*,* $i = 1, \cdots, n$*.*

Figure 3.1 about here

For hierarchical classification, (2.3) yields different classifiers with different choice of margin loss $v(\cdot)$. Moreover, (2.3) covers multi-class classifiers with equal cost when all the leaf nodes share the same parent —the root; see, for example, Liu and Shen (2006) for multi-class SVM and ψ-learning.

2.3.2 Minimization

This section implements (2.3) in a generic form for linear and kernel learning. In either case, $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T \tilde{\boldsymbol{x}} + b_j$, with $\boldsymbol{w}_j \in \mathbb{R}^q$; $j = 1, \cdots, K$. By Theorem 1, (2.3) subject to $\sum_{\{t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_t(\boldsymbol{x}) = 0$; $j = 1, \cdots, K$ and $\forall \boldsymbol{x} \in S$, reduces to:

$$s(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^{K} \|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{n} v(u_{min}(\boldsymbol{f}(\boldsymbol{x}_i), y_i)), \tag{2.4}$$

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

In (2.4), we obtain the proposed hierarchical classifiers, denoted by HSVM and HPSI when $v(u) = (1-u)_+$ and $v(u) = \psi(u)$. In the first case, (2.4) with $v(u) = (1-u)_+$ is solved by quadratic programming (QP) through its dual form, c.f., Appendix B. In the second case, (2.4) with $v(u) = \psi(u)$ is solved by difference convex (DC) programming. We shall elaborate further next.

Key to DC programming is decomposing $s(\boldsymbol{f})$ in (2.4) 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 $\psi_2(u) = (-u)_+$, see Figure 2.2.

On the basis of 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 the 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 process iterates until it terminates. This method is known as a DC method designed for non-convex minimization in the global optimization literature; c.f., An and Tao (1997).

Figure 2.2 about here

We now design our DC algorithm. Starting from an initial value $\hat{\boldsymbol{f}}^{(0)}$, the solution of HSVM, we iteratively solve primal problems. At the m'th iteration, we solve for

$$\hat{\boldsymbol{f}}^{(m)} = \min_{\boldsymbol{f}} (s_1(\boldsymbol{f}) - \langle \boldsymbol{f}, \nabla s_2(\hat{\boldsymbol{f}}^{(m-1)})\rangle_{\mathcal{K}}), \tag{2.5}$$

subject to $\sum_{\{t:t\in sib(j)\cup\{j\}, sib(j)\neq\phi\}} f_t(\boldsymbol{x}_i) = 0$; $i = 1, \cdots, n$, $j = 1, \cdots, K$, through QP and its dual form, see Appendix B. The above iterative process continues until a termination criterion is met: $\|\hat{\boldsymbol{f}}^{(m)} - \hat{\boldsymbol{f}}^{(m-1)}\|_2 \leq \epsilon$, where is $\epsilon > 0$ 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 finitely, and its speed of convergence is superlinear, by Theorem 3 of Liu, Shen and Wong (2005) for ψ-learning. Although a DC algorithm can not assure the global minimizer, it usually leads to a good local solution when it is not global, c.f., An and Tao (1997), and Liu, Shen and Wong (2005). 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 can be realized if it is not global. More importantly, an ε-global minimizer can be obtained when the algorithm is combined with the branch-and-bound method, as in Liu, Shen and Wong (2005). For a computational consideration, we shall not seek the exact global minimizer.

2.4 Numerical examples

2.4.1 Simulated examples

This section applies HSVM and HPSI to simulated examples. HSVM and HPSI are compared against their flat counterpart—k-class SVM of Liu and Shen (2006), and the hierarchical SVM of Cai and Hofmann (2004), denoted as HSVM$_c$. Note that a comparison with k-class SVM is appropriate rather than K-class SVM.

All numerical analyses are conducted in R2.1.1 for SVM, HSVM, HPSI and HSVM$_c$. 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 set to be the median of the inter-class distances between any two classes, see Jaakkola, Diekhans and Haussler (1999) for the binary case.

A classifier's generalization performance is measured by the test error averaged over 100 simulation replications, approximating the generalization error. The test error is

defined as

$$TE(\boldsymbol{f}) = n_{test}^{-1} \sum_{i=1}^{n_{test}} l(Y_i, d^H(\boldsymbol{f}(\boldsymbol{X}_i))), \tag{2.6}$$

where n_{test} is the size of the test sample, and l is one of the four losses: l_{0-1}, l_Δ, l_{sib} with c_j's defined by (2.1) and l_{sub} with c_j's defined by (2.2). The corresponding test errors are denoted by TE_{0-1}, TE_Δ, TE_{sib} and TE_{sub}.

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

$$\frac{(TE(\text{SVM}) - Bayes) - (TE(\cdot) - Bayes)}{(TE(\text{SVM}) - Bayes)}, \tag{2.7}$$

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

$$\frac{TE(\text{SVM}) - TE(\cdot)}{TE(\text{SVM})}, \tag{2.8}$$

which may underestimate the actual percentage of improvement over SVM.

For each classifier, the test errors are computed on a testing sample of size 5×10^4, and are minimized over tuning parameter C on 61 grid points: $C = 10^{l/10}; l = -30, -29, \cdots, 30$.

The following examples are considered.

Example 1. A complete binary tree of depth 3 is considered, which is displayed in Figure 2.3. 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: $\boldsymbol{X}_i \sim U^2(-1,1)$, $Y_i|\boldsymbol{X}_i = \lceil 8 \times X_{i1} \rceil \in \{1, \cdots, 8\}$. 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, $P(Y_i = j|\boldsymbol{X}_i) = \sum_{\{t \in sub(j):chi(t)=\phi\}} P(Y_i = t|\boldsymbol{X}_i)$. This generates a non-separable case.

Figure 2.3 and Table 1 about here

As suggested by Table 1, HSVM and HPSI outperform SVM and HSVM$_c$ under l_{0-1}, l_Δ, l_{sib} and l_{sub} in all the cases. In the linear case, the amount of improvement of HSVM over SVM ranges from 52.7% to 56.9%, whereas that of HPSI over SVM is from 54.3% to 59.2%. In contrast, the amount improvement of HSVM$_c$ is from 0% to 7.7%, which is small. In the Gaussian kernel case, the amounts for HSVM, HPSI and HSVM$_c$ are 45.0% to 60.4%, 50.0% to 66.7%, and −4.1% to 27.0%, respectively. Overall, HPSI performs the best and outperforms its competitors significantly in all the situations.

With regard to different evaluation losses, we note that the performances of the four classifiers are similar 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 losses.

In the linear and nonlinear cases, the difference for each classifier appears to be small. This is mainly due to the linear Bayes rule in this case.

In summary, HSVM and HPSI indeed yield improvements over their flat counterpart because of the built-in hierarchical structure. Indeed, the hierarchy–a tree of depth 3 is useful in reducing a classification problem's dimension, which can be explained by the

concept of the margins with respect to hierarchy, as discussed in Section 2.3.1.

Example 2. 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 2.4.

Figures 2.4 and 2.5, Tables 2 and 3 about here

Clearly, HSVM$_c$, HSVM and HPSI outperform SVM in all the cases. The amount improvement of HSVM over SVM varies from 22.2% to 46.9% in the linear case and 8.9% to 42.5% in the Gaussian kernel case, whereas that of HPSI ranges from 39.6% to 84.4% and 19.5% to 80.6%, respectively. In contrast, the amount of improvement of HSVM$_c$ is from 7.0% to 23.8% in the linear case and from 3.2% to 15.9% in the Gaussian kernel case. The improvement of HPSI over HSVM becomes more significant than that in Example 1. As expected, HPSI is the best performer and does make a difference in this highly non-separable example. In fact, HPSI nearly achieves the optimal performance of the Bayes rule when the sample size becomes large.

2.4.2 Classification of gene functions

Essential to organizing the huge amount of genomic information is determining biological functions of genes through gene expressions. The microarray data for gene expressions can be obtained in batches. It is generally believed that genes having the same or similar functions tend to be coexpressed, c.f., Broet et al. (2002). Therefore gene expressions can be used to classify a gene's biological function into existing functional categories, with the potential of discovering new functional categories.

Gene function categories are typically organized hierarchically by a gene annotation system, for instance, MIPS (Mewes et al., 2002). Such a system organizes existing genomic information through a hierarchy, with upper levels representing general information about large groups of similar categories, and lower levels indicating more specified functions of smaller groups. Through gene expressions, gene function prediction can be cast into the framework of hierarchical classification, where a vector $\boldsymbol{x}$ represents the log-ratio (base 10) of emission intensities of gene expressions from a two-channel microarray chip equipped with a two-color hybridization scheme, and label Y indicates the location within the hierarchy.

This section applies HSVM and HPSI to predict gene functions through gene expression data in Hughes et al. (2000). There expression profiles of a total of 6316 genes are collected from 300 microarray experiments for yeast *S. cerevisiae*, with the dimension of $\boldsymbol{x}_i$ being 300. Hughes' dataset was collected using an approach of deletions of uncharacterized genes. This approach has been proven successful in identifying, through expression profiling, gene function categories for eight uncharacterized open reading frames encode proteins, which

were confirmed by biological experiments. Specifically, three hundreds expression profiles were generated for the full-genome of yeast *S. cerevisiae* simultaneously for three hundreds experiments, in which expression levels of treatment–a mutant or a compound-treated culture were compared against that of control–a wild-type or a mock-treated culture. They consist of two hundred seventy six deletion mutant genes, 11 tetractcline-regulatable alleles of essential genes, and 13 well-characterized compounds treatments. For these 300 expression profiles, genes from different steps in same biochemistry function have similar expression behavior, and large set of profiles gives sufficient unique expression responses for most function categories.

Deletion mutants were selected in a way that a variety of functional classifications were represented. Experiments were performed under a certain condition to allow direct comparison of the behavior of all genes in response to all mutations and treatments. Expressions of the three hundreds experiments were profiled through a two-channel cDNA chip equipped with a two-color hybridization scheme, where the gene expression ratios of treatment and control were recorded for the 300 experiments.

As suggested in Hughes et. al (2000), the pathway(s) perturbed by an uncharacterized mutation would be ascertained by matching expression patterns most strongly resemble in a comprehensive database of reference profiles such as MIPS. In words, the expression profiles based on such experiments are informative in gene function discovery.

As of May, 2005, there were only 68.5% of the genes annotated by MIPS. Of particular interest is classifying gene functional categories within two major branches of MIPS, which is composed of two functional categories at the highest level: “cell cycle and DNA

processing" and "transcription" and their corresponding offsprings. In our analysis, we use these two major branches with 1103 annotated genes within this tree hierarchy of 23 functional categories, see Figure 2.6 for a display of the hierarchy.

Figure 2.6 about here

First we perform some simulations to gain an insight into HSVM and HPSI for gene function prediction before applying to predict new gene functions. Towards this end, we randomly partition the entire set of data of 1103 genes into training and testing sets, with 300 and 803 genes, respectively. Then HSVM and HPSI are trained, tuned and tested as in Section 4.1. This process is repeated over 100 random partitions. Finally, we predict unknown gene functional categories that had not been annotated in the 2005 version of MIPS. The predicted gene functions are then cross-validated by a newer version of MIPS, dated in March 2008, where about 600 additional gene functions have been added into functional categories, representing the latest biological information.

Table 4 about here

As suggested by Table 4, 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} are seen 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 relationship is loose. 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. In our case, small and large branches have the same parents, leading to large differences in penalties under different losses.

For new gene function prediction, we use the best performed classifiers for HSVM and HPSI over the 100 random partitions. For prediction, ten most confident genes are chosen among those genes that have not been annotated in the 2005 version of MIPS but are annotated in the 2008 version. Here the confidence is measured by the value of the functional margin. Then HSVM and HPSI are applied to yield ten predictions that are cross-validated by the 2008 version of MIPS. For an example, gene "YGR054w" is not annotated in the 2005 version of MIPS, and is predicted to belong to functional categories along a single path "Protein synthesis" $\rightarrow$ "Ribosome biogenesis" $\rightarrow$ "Ribosomal proteins" by HPSI. This predication is confirmed to be exactly correct by the newer version of MIPS. Overall, seven out of the ten genes are predicted correctly for both HSVM and HPSI.

In contrast to the existing results for real large hierarchical problems, e.g., Rousu et al. (2005), Cesa-Bianchi et al. (2004), and Cai and Hofmann (2004), the above results are considered to be a significant improvement over its flat counterpart.

Table 5 about here

2.5 Statistical learning theory

In the literature, neither the generalization accuracy for hierarchical classification nor the role of $\mathcal{H}$ has been investigated. 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 (2.4) 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 how $\mathcal{H}$ improves the performance of flat classification.

2.5.1 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 $e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}}) = GE(d^H(\hat{\boldsymbol{f}})) - GE(d^H(\bar{\boldsymbol{f}})) = E(l(Y, d^H(\hat{\boldsymbol{f}}(\boldsymbol{X}))) - l(Y, d^H(\bar{\boldsymbol{f}}(\boldsymbol{X})))) \geq 0$. Here l is the 0-1 loss, and $GE(d^H(\bar{\boldsymbol{f}}))$ is the optimal performance for any classifier provided that the unknown true distribution $P(\boldsymbol{x}, y)$ would have been available.

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 $V(\boldsymbol{f}, \boldsymbol{Z}) = v(u_{min}(\boldsymbol{f}(\boldsymbol{X}), Y))$, and $v(\cdot)$ is any large margin surrogate loss used in (2.4).

The following theorem quantifies $e(\hat{\boldsymbol{f}}, \bar{\boldsymbol{f}})$ in terms of the tuning parameter C, the sample size n and the complexity of the class of candidate function vectors.

Theorem 2 *Under Assumptions A-C in Appendix A, for any large margin hierarchical classifier $\boldsymbol{d}^H(\hat{\boldsymbol{f}})$ defined by (2.3), there exists a constant $c_6 > 0$ such that*

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

provided that $\lambda^{-1} \geq 2\delta_n^{-2} J_0$, where $\delta_n^2 = \min(\epsilon_n^2 + 2e_V(\boldsymbol{f}^*, \bar{\boldsymbol{f}}), 1)$, $\boldsymbol{f}^* \in \mathcal{F}$ is an approximation of $\mathcal{F}$ 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 α, β, ϵ_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.*

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 is defined by the bracketing L_2 entropy of 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 $EV(\boldsymbol{f}, Z)$.

By comparison, $\mathcal{F}^V(t)$ is induced by a margin loss V in multi-class classification usually is larger than its counterpart in hierarchical classification. This is because V is structured in single-path classification in that functional margin $u_{min}(\boldsymbol{f}(\boldsymbol{X}), Y)$ involves a smaller number of pairwise comparisons among classes, for instance, only pairs between the immediate offsprings for each ancestor of class j are compared for a tree. In contrast, any two classes need to be compared in multi-class classification without such a hierarchy, c.f, (Liu and Shen, 2006). A theoretical description of this phenomenon in terms of the L_2 bracketing entropy is given in Lemma 2.3 for a binary tree.

Lemma 2.3 *Let* $\mathcal{H}$ *be a hierarchy defined by a tree with* K *non-root nodes including* k *leaf nodes. If* $\mathcal{F}_1 = \cdots = \mathcal{F}_K$*, then* $H_B(\epsilon, \mathcal{F}^V(t)) \leq 2c(\mathcal{H}) H_B(\epsilon/(2c(\mathcal{H})), \mathcal{F}_1(t))$ *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* $\mathcal{F}_j(t) = \{f_j : \frac{1}{2}\|f_j\|_{\mathcal{K}} \leq J_0 t\}$*;* $j = 1, \cdots, K$.

2.5.2 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 $K = 2^{p+1} - 1 = 2k - 1$ non-root nodes,

see Figure 2.3 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; $i = k+1, \cdots, K$, $P(Y = j_i|\boldsymbol{X}) = \sum_{t \in sub(j_i), chi(t)=\phi} P(Y = t|\boldsymbol{X})$. Then the Bayes rule $\bar{\boldsymbol{f}} = \{\bar{f}_1, \cdots, \bar{f}_K\}$, where $\bar{f}_{j_i}(\boldsymbol{x}) = \sum_{t=1}^{i}(x_{(1)} - t/k)$; $i = 1, \cdots, k$, and $\bar{f}_{j_i}(\boldsymbol{x}) = \max_{\{t \in sub(j_i), chi(t)=\phi\}} \bar{f}_t$; $i = k+1, \cdots, K$.

Linear learning: Let $\mathcal{F} = \{(f_1, \cdots, f_K) : f_j = \boldsymbol{w}_j^T \boldsymbol{x} + b_j\}$ and $J(\boldsymbol{f}) = \sum_{j=1}^{K} \|\boldsymbol{w}_j\|^2$. We now verify Assumptions A-C. 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 $f_j^* = \max_{\{t \in sub(j): chi(t)=\phi\}} f_t^*$ 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, by Lemma 2.3 with $c(\mathcal{H}) = \sum_{j=0}^{K} |chi(j)|(|chi(j)| - 1)/2 = \sum_{j=0}^{K} I\{chi(j) \neq \phi\} = k - 1$, we have, for HSVM and HPSI, $H_B(\epsilon, \mathcal{F}^V(t)) \leq O(k \log(k/\epsilon))$. Consequently $\sup_{t \geq 2} \phi(\epsilon_n, t) \leq O((k \log(k/\epsilon_n))^{1/2}/\epsilon_n)$. Solving (2.11) in Assumption C leads to $\epsilon_n = (\frac{k \log n}{n})^{1/2}$ for HSVM and HPSI when $C/J_0 \sim \delta_n^{-2}/n \sim \frac{1}{n\epsilon_n^2}$, provided that $\frac{k \log n}{n} \to 0$. Similarly, for multi-class SVM and ψ-learning, $\epsilon_n = (\frac{k(k-1)/2 \log n}{n})^{1/2}$, c.f., 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 $\mathcal{H}$ were a flat tree with only one layer, that is, all the leaf nodes are the direct offsprings of the root node 0. 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 counterpart.

Gaussian kernel learning: Consider the same setting with candidate function class defined by the Gaussian kernel. By the representation theorem of the reproducing kernel Hilbert spaces, c.f., 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 $f_j(\boldsymbol{x}) = b_j + \boldsymbol{w}_j^T \phi(\boldsymbol{x}) = b_j + \sum_{l=0}^{\infty} w_{j,l}\phi_l(\boldsymbol{x}) : \boldsymbol{w}_j \in \mathcal{R}^{\infty}\}$, and $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z})\rangle = \mathcal{K}(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma_n^2})$.

For HSVM, letting $f_j^* = 1 - (1 + \exp(\sum_{t=1}^{j} \tau(x_{(1)} - t/k)))^{-1}$; $j = 1, \cdots, k$, and $f_j^* = \max_{\{t \in sub(j):chi(t)=\phi\}} f_t^*$ 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, it follows from Lemma 7 of Shen and Wang (2007) and Lemma 1, that $H_B(\epsilon, \mathcal{F}^V(t)) \leq O(k(\log((J^*t)^{1/2}k/\epsilon))^3)$, where $J^* = J(\boldsymbol{f}^*)$. Solving (2.11) in Assumption C leads to $\epsilon_n^2 = kn^{-1}(\log((J^*n)^{1/2}))^3$ when $\lambda J^* \sim \epsilon_n^2$. 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 $\delta_n^2 = \max(kn^{-1}(\tau^2\sigma_n^2 + \sigma_n^{-2} + \log n)^3,\ k/\tau) = O_p\Big(kn^{-1/7}\Big)$ 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 $f_j^* = \max_{\{t \in sub(j), chi(t)=\phi\}} f_t^*$ 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). Similarly as in HSVM, solving (2.11) in Assumption C leads to $\epsilon_n^2 = kn^{-1}(\log((J^*n)^{1/2}))^3$

when $\lambda J^* \sim \epsilon_n^2$ for Assumption C. 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 $\delta_n^2 = \max(kn^{-1}(\log(n\tau^2\sigma_n^2) + \sigma_n^{-2})^3,\ k/\tau) = O\Big(kn^{-1}(\log n)^3\Big)$ 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 hierarchy structure reduces the order from $k(k-1)/2$ to k as in the linear case.

2.6 Discussion

This chapter proposed a novel large margin method for single-path hierarchical classification with exclusive membership. In contrast to existing hierarchical classification methods, the proposed method fully utilizes the inter-class relationship 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, as defined by the generalization error with respect to four losses, including the 0-1 loss and several hierarchical losses, has been improved. Our theoretical and numerical analyses suggest that the generalization accuracy of the proposed classifiers has improved over its flat counterpart.

Appendix A

The following assumptions are made for Theorem 2.

Define a truncated $V^T(\boldsymbol{f}, \boldsymbol{Z}) = T \vee V(\boldsymbol{f}, \boldsymbol{Z})$ 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 $\epsilon > 0$,

$$\sup_{\{\boldsymbol{f} \in \mathcal{F}:\ e_{VT}(\boldsymbol{f}, \boldsymbol{f}^*) \leq \epsilon\}} |e(\boldsymbol{f}, \bar{\boldsymbol{f}})| \leq c_1 \epsilon^{\alpha}. \tag{2.9}$$

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

$$\sup_{\{\boldsymbol{f} \in \mathcal{F}:\ e_{VT}(\boldsymbol{f}, \bar{\boldsymbol{f}}) \leq \epsilon\}} Var(V^T(\boldsymbol{f}, Z) - V(\bar{\boldsymbol{f}}, Z)) \leq c_2 \epsilon^{\beta}. \tag{2.10}$$

These assumptions describe local smoothness of $|e(\boldsymbol{f}, \bar{\boldsymbol{f}})|$ and $Var(V^T(\boldsymbol{f}, Z) - V(\bar{\boldsymbol{f}}, Z))$. The exponents α and β depend on the joint distribution of (X, Y).

We now define a complexity measure of a function space $\mathcal{F} = \{f\}$. Given any $\epsilon > 0$, 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 $\|f_j^l - f_j^u\|_2 \leq \epsilon; j = 1, \cdots, m$, where $\|f\|_2 = (Ef^2)^{\frac{1}{2}}$ is the L_2-norm. Then the metric entropy with bracketing $H_B(\epsilon, \mathcal{F})$ is the logarithm of the cardinality of the smallest ϵ-bracketing set for $\mathcal{F}$. Let $\mathcal{F}^V(t) = \{V^T(\boldsymbol{f}, \boldsymbol{z}) - V(\boldsymbol{f}_0, \boldsymbol{z}) : \boldsymbol{f} \in \mathcal{F}, J(\boldsymbol{f}) \leq J_0 t\}$, where $J(\boldsymbol{f}) = \frac{1}{2}\sum_{j=1}^K \|f_j\|^2$ and $J_0 = \max(J(\boldsymbol{f}_0), 1)$.

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

$$\sup_{t \geq 2} \phi(\epsilon_n, s) \leq c_5 n^{1/2}, \tag{2.11}$$

where $\phi(\epsilon_n, s) = \int_{c_3 L}^{c_4^{1/2} L^{\beta/2}} H_B^{1/2}(u, \mathcal{F}^V(s)) du / L$, and $L = L(\epsilon_n, \lambda, s) = \min(\epsilon_n^2 + \lambda J_0(s/2 - 1), 1)$.

Appendix B

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

Proof of Lemma 2.1: Without loss of generality, assume that the membership is exclusive for the first k classes. The 0-1 loss over K non-exclusive membership classes can be expressed as $\max_{t=1}^{K}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t))$, which is the disagreement between the value of Y and that of $d(\boldsymbol{X})$. in $\mathcal{H}$. By assumption, if $\max_{t=1}^{k}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) = 0$, then $\max_{t=k+1}^{K}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) = 0$. On the other hand, $\max_{t=1}^{K}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) \geq \max_{t=1}^{k}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t))$. This implies that if $\max_{t=1}^{k}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) = 1$ then $\max_{t=1}^{K}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) = 1$. Consequently $l_{0-1}(Y, d(\boldsymbol{X})) = \max_{t=1}^{k}(I(Y = t, d(\boldsymbol{X}) \neq t) + I(Y \neq t, d(\boldsymbol{X}) = t)) = \sum_{t=1}^{k} I(d(\boldsymbol{X}) \neq t)I(Y = t)$ by exclusiveness of the membership. Finally

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

This completes the proof.

Proof of Lemma 2.2: The global minimizer $\bar{d}(\boldsymbol{x})$ for $El_{0-1}(Y, d(\boldsymbol{X}))$ follows the fact that it is a global minimizer of $E(l_{0-1}(Y, d(\boldsymbol{X}))|\boldsymbol{X} = \boldsymbol{x})$ for any $\boldsymbol{x}$.

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 we show that $m(\cdot, \cdot)$ satisfies the triangle inequality: for any $1 \leq j_1, j_2, j_3 \leq K$, $m(j_1, j_3) \leq m(j_1, j_2) + m(j_2, j_3)$. To this end, let $A = anc(j_1)$, $B = anc(j_2)$ and $C = anc(j_3)$, and let

$A/B = A \cap \bar{B}$ denote the set difference. Then

$$\begin{aligned} & m(j_1, j_2) + m(j_2, j_3) - m(j_1, j_3) = |A \triangle B| + |B \triangle C| - |A \triangle C| \\ = \; & (|(A \cap C/B)| + |(A/B/C)| + |(B \cap C/A)| + |(B/A/C)|) + \\ & (|(A \cap B/C)| + |(B/A/C)| + |(A \cap C/B)| + |(C/A/B)|) - \\ & (|(A \cap B/C)| + |(A/B/C)| + |(B \cap C/A)| + |(C/A/B)|) \\ = \; & 2(|(A \cap C/B)| + |(B/A/C)|) \geq 0, \end{aligned}$$

which establishes the triangle inequality.

In what follows, we prove that $E(l_\Delta(Y, \bar{d}(\boldsymbol{X}))|\boldsymbol{X} = \boldsymbol{x}) \leq E(l_\Delta(Y, d(\boldsymbol{X}))|\boldsymbol{X} = \boldsymbol{x})$ for any $\boldsymbol{x}$ and classifier $d(\boldsymbol{x})$. Let $\hat{y} = \bar{d}(\boldsymbol{x})$. It follows from the triangle inequality of $m(\cdot, \cdot)$ that $m(y, d(\boldsymbol{x})) - m(y, \hat{y}) \geq -m(d(\boldsymbol{x}), \hat{y})$ for any y with $chi(y) = \phi$. Note that $m(\hat{y}, \hat{y}) = 0$ and $m(\hat{y}, d(\boldsymbol{x})) = m(d(\boldsymbol{x}), \hat{y}) \geq 0$. Then

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

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

Proof of Lemma 2.3: To construct bracket covering for $\tilde{\mathcal{F}}(t)$, 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 $f_j^{i,l} \le f_j \le f_j^{i,u}$ with $\|f_j^{i,u} - f_j^{i,l}\|_2 \le \epsilon$; $j = 1, \cdots, K$. Now construct a set of brackets for $\mathcal{F}^T(t)$. Let $g^u = \max_{\{j' \in sib(j), j \in anc(y) \cup \{y\}\}} v\big(f_j^{i,l} - f_{j'}^{i',u}\big)$ 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, $T \wedge g^l \le V^T(\boldsymbol{f}, \boldsymbol{z}) = T \wedge \max\{v(f_j - f_{j'}) : j' \in sib(j), j \in anc(y) \cup \{y\}\} \le T \wedge g^u$ since $h^T(t) = T \wedge t$ is non-decreasing in t. 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

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

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 $H_B(2c(\mathcal{H})\epsilon, \tilde{\mathcal{F}}^T(t)) \le H_B(2c(\mathcal{H})\epsilon, \mathcal{F}(t))$. 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 $1 + \sum_{j:chi(j) \ne \phi}(|chi(j)| - 1)$ 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, $k = 1 + \sum_{j:chi(j) \ne \phi}(|chi(j)| - 1)$. For $chi(j) \ne \phi$, $|chi(j)| - 1 \ge 0$. Then $\sum_{j:chi(j) \ne \phi}(|chi(j)| - 1)^2 \le \Big(\sum_{j:chi(j) \ne \phi}(|chi(j)| - 1)\Big)^2 = (k-1)^2$. This implies $2c(\mathcal{H}) = \sum_{j:chi(j) \ne \phi}(|chi(j)| - 1)^2 + \sum_{j:chi(j) \ne \phi}(|chi(j)| - 1) \le (k-1)^2 + k - 1 = k(k-1)$. This completes the proof.

Prime and its dual forms of (2.4) for HSVM: The prime and its dual forms can be obtained from those of HPSI below, with $\nabla \hat{\boldsymbol{w}}_j^{(m-1)} = 0$ and $\nabla \hat{b}_j^{(m-1)} = 0$; $j = 1, \cdots, K$

there.

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

Prime and its dual forms of (2.5) for HPSI: The prime problem of (2.5) is

$$\underset{\boldsymbol{f}}{\operatorname{argmin}} \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{j=1}^{K}\langle\nabla\hat{\boldsymbol{w}}_j^{(m-1)}, \boldsymbol{w}_j\rangle - \sum_{j=1}^{K}\langle\nabla\hat{b}_j^{(m-1)}, b_j\rangle, \tag{2.12}$$

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 $\sum_{j\in chi(s), chi(()s)\neq\phi} f_j(\boldsymbol{x}_i) = 0$; $i = 1, \cdots, n$, $s = 1, \cdots, K$.

To solve (2.12), 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 (2.12). Then (2.12) is equivalent to:

$$\begin{aligned}\max_{\alpha_i,\beta_{i,j,t},\delta_{i,s}} L \;=\; & \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{j=1}^{K}\langle\nabla\hat{\boldsymbol{w}}_j^{(m-1)}, \boldsymbol{w}_j\rangle - \sum_{j=1}^{K}\langle\nabla\hat{b}_j^{(m-1)}, b_j\rangle + \\ & \sum_{(j,t)\in Q(y_i):i=1,\cdots,n} \beta_{i,j,t}\left(1 - ((\langle\boldsymbol{w}_j, \boldsymbol{x}_i\rangle + b_j) - (\langle\boldsymbol{w}_t, \boldsymbol{x}_i\rangle + b_t)) - \xi_i\right) \\ & - \sum_{i=1}^{n}\alpha_i\xi_i + \sum_{(i,s):i=1,\cdots,n;chi(s)\neq\phi} \delta_{i,s}\sum_{j\in chi(s)}(\langle\boldsymbol{w}_j, \boldsymbol{x}_i\rangle + b_j).\end{aligned} \tag{2.13}$$

By letting the partial derivatives be zero, we have that

$$\frac{\partial L}{\partial \boldsymbol{w}_j} = 0\,, \frac{\partial L}{\partial \xi_i} = 0\,, \frac{\partial L}{\partial b_j} = 0; i = 1, \cdots, n, j = 1, \cdots, K. \tag{2.14}$$

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

$$\sum_{(j,t)\in Q(y_i)} \beta_{i,j,t} \leq C; i = 1, \cdots, n. \tag{2.15}$$

After substituting (2.14) in (2.13), we obtain a quadratic form in $\{\alpha_i, \beta_{i,j,t}, \delta_{i,s}\}$:

$$\begin{aligned}L \;=\; & \sum_{i=1,\cdots,n;(j,t)\in Q(y_i)} \beta_{i,j,t} - \\ & \frac{1}{2}\sum_{j=1}^{K}\left\|\sum_{\substack{i=1,\cdots,n,\\ y_i\in sub(sib(j))}} \boldsymbol{x}_i\beta_{i,j,y_i} - \sum_{\substack{i=1,\cdots,n,\\ y_i\in sub(j)}} \boldsymbol{x}_i\beta_{i,y_i,j} + \sum_{i=1}^{n}\delta_{i,par(j)}\boldsymbol{x}_i - \nabla\hat{\boldsymbol{w}}_j^{(m-1)}\right\|^2\end{aligned} \tag{2.16}$$

where $\|\cdot\|$ is the usual L_2-Euclidean norm. Maximizing (2.16) subject to $\beta_{i,j,t} \geq 0$; $i = 1, \cdots, n; (j,t) \in Q(y_i)$, (2.14) and (2.15) 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 (2.14). The solution of b_j is derived from Karush-Kuhn-Tucker's condition: $\beta_{i,j,t}(1 - ((\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle + b_j) - (\langle \boldsymbol{w}_t, \boldsymbol{x}_i \rangle + b_t)) - \xi_i) = 0$, $\alpha_i \xi_i = 0$, 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 (2.12), and solve b_j's through linear programming.

Table 2.1: Averaged test errors as well as estimated standard errors (in parenthesis) of SVM, HSVM, HPSI and HSVM$_c$ over 100 simulation replications in Example 1 of Section 2.4.1. 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.

Linear				
Training	Test error			
Method	l_{0-1}	l_Δ	l_{sib}	l_{sub}
SVM	0.326(0.004)	0.179(0.003)	0.148(0.002)	0.122(0.002)
HSVM$_c$	0.322(0.006)	0.169(0.002)	0.148(0.003)	0.120(0.002)
% of impro.	1.7%	7.7%	0%	2.2%
HSVM	0.198(0.003)	0.105(0.002)	0.086(0.001)	0.070(0.001)
% of impro.	52.7%	56.9%	55.4%	55.9%
HPSI	**0.194(0.003)**	**0.102(0.001)**	**0.086(0.002)**	**0.068(0.002)**
% of impro.	54.3%	59.2%	55.4%	58.1%
Bayes Rule	0.083	0.049	0.036	0.029

Gaussian				
Training	Test error			
Methods	l_{0-1}	l_Δ	l_{sib}	l_{sub}
SVM	0.303(0.015)	0.208(0.001)	0.134(0.008)	0.110(0.007)
HSVM$_c$	0.312(0.005)	0.165(0.003)	0.128(0.006)	0.109(0.005)
% of impro.	−4.1%	27.0%	6.1%	1.2%
HSVM	0.204(0.003)	0.112(0.002)	0.087(0.001)	0.069(0.001)
% of impro.	45.0%	60.4%	48.0%	50.6%
HPSI	**0.190(0.002)**	**0.102(0.002)**	**0.085(0.002)**	**0.063(0.002)**
% of impro.	51.4%	66.7%	50.0%	58.0%
Bayes Rule	0.083	0.049	0.036	0.029

Table 2.2: Averaged test errors as well as estimated standard errors (in parenthesis) of SVM, HSVM$_c$, HSVM and HPSI over 100 simulation replications of linear learning in Example 2 of Section 2.4.1, with $n = 50, 150, 500, 1500$. The test errors are computed under the l_{0-1}, l_Δ, l_{sib} and l_{sub}.

Linear						
l	Sample Size	TE and % of impro.				Bayes Rule
		SVM	HSVM$_c$	HSVM	HPSI	
l_{0-1}	$n = 50$	0.344(0.070)	0.334(0.047)	0.312(0.058)	**0.287(0.045)**	0.200
			7.0%	22.2%	39.6%	
	$n = 150$	0.282(0.043)	0.273(0.023)	0.259(0.030)	**0.236(0.016)**	
			11.0%	28.0%	56.1%	
	$n = 500$	0.247(0.014)	0.241(0.014)	0.233(0.013)	**0.213(0.007)**	
			12.8%	29.8%	72.3%	
	$n = 1500$	0.230(0.010)	0.223(0.009)	0.217(0.005)	**0.205(0.003)**	
			23.3%	43.4%	83.3%	
l_{sib}	$n = 50$	0.275(0.056)	0.267(0.037)	0.247(0.046)	**0.227(0.035)**	0.167
			7.4%	25.8%	44.3%	
	$n = 150$	0.228(0.032)	0.220(0.018)	0.209(0.022)	**0.190(0.012)**	
			13.1%	31.0%	62.0%	
	$n = 500$	0.203(0.012)	0.198(0.012)	0.192(0.011)	**0.175(0.005)**	
			13.9%	30.3%	77.1%	
	$n = 1500$	0.188(0.007)	0.183(0.007)	0.178(0.004)	**0.170(0.002)**	
			23.8%	46.9%	84.4%	

Linear						
l	Sample Size	TE and % of impro.				Bayes Rule
		SVM	HSVM_c	HSVM	HPSI	
l_{sub}	$n = 50$	0.251(0.051)	0.242(0.041)	0.225(0.042)	**0.207(0.033)**	0.156
			9.4%	27.2%	46.1%	
	$n = 150$	0.210(0.029)	0.201(0.020)	0.192(0.020)	**0.175(0.010)**	
			16.7%	33.1%	64.3%	
	$n = 500$	0.188(0.011)	0.184(0.011)	0.178(0.010)	**0.162(0.005)**	
			12.5%	30.8%	80.1%	
	$n = 1500$	0.175(0.007)	0.172(0.007)	0.165(0.004)	**0.158(0.002)**	
			15.7%	51.4%	87.4%	
l_{Δ}	$n = 50$	0.183(0.037)	0.178(0.025)	0.165(0.031)	**0.151(0.023)**	0.111
			7.4%	25.8%	44.3%	
	$n = 150$	0.152(0.021)	0.147(0.012)	0.139(0.015)	**0.127(0.008)**	
			13.1%	31.0%	62.0%	
	$n = 500$	0.135(0.008)	0.132(0.008)	0.128(0.007)	**0.117(0.003)**	
			13.9%	30.3%	77.1%	
	$n = 1500$	0.125(0.005)	0.122(0.005)	0.119(0.003)	**0.113(0.002)**	
			23.8%	46.9%	84.4%	

Table 2.3: Averaged test errors as well as estimated standard errors (in parenthesis) of SVM, HSVM$_c$, HSVM and HPSI over 100 simulation replications of kernel learning in Example 2 of Section 2.4.1, with $n = 50, 150, 500, 1500$. The test errors are computed under the l_{0-1}, l_Δ, l_{sib} and l_{sub}.

Gaussian						
l	Sample Size	TE and % of impro.				Bayes Rule
		SVM	HSVM$_c$	HSVM	HPSI	
l_{0-1}	$n = 50$	0.324(0.060)	0.320(0.047)	0.313(0.047)	**0.298(0.045)**	0.200
			3.2%	8.9%	20.9%	
	$n = 150$	0.280(0.036)	0.276(0.030)	0.270(0.027)	**0.261(0.016)**	
			5.0%	12.5%	23.8%	
	$n = 500$	0.257(0.022)	0.252(0.013)	0.240(0.014)	**0.224(0.007)**	
			8.8%	29.8%	57.9%	
	$n = 1500$	0.247(0.013)	0.240(0.011)	0.227(0.010)	**0.215(0.003)**	
			12.9%	42.5%	68.1%	
l_{sib}	$n = 50$	0.259(0.048)	0.254(0.037)	0.250(0.037)	**0.241(0.035)**	0.167
			5.4%	9.8%	19.5%	
	$n = 150$	0.226(0.029)	0.223(0.022)	0.218(0.022)	**0.209(0.012)**	
			5.1%	13.1%	28.8%	
	$n = 500$	0.208(0.016)	0.202(0.011)	0.195(0.010)	**0.181(0.005)**	
			14.6%	31.7%	65.8%	
	$n = 1500$	0.198(0.008)	0.193(0.005)	0.185(0.006)	**0.173(0.003)**	
			16.0%	40.9%	80.6%	

Gaussian						
l	Sample Size	TE and % of impro.				Bayes Rule
		SVM	HSVM$_c$	HSVM	HPSI	
l_{sub}	$n = 50$	0.237(0.044)	0.233(0.041)	0.228(0.034)	**0.220(0.033)**	0.156
			4.9%	11.1%	21.0%	
	$n = 150$	0.208(0.027)	0.205(0.020)	0.201(0.020)	**0.191(0.010)**	
			5.8%	13.5%	32.7%	
	$n = 500$	0.192(0.015)	0.187(0.010)	0.180(0.009)	**0.169(0.005)**	
			13.9%	33.3%	63.9%	
	$n = 1500$	0.187(0.009)	0.182(0.005)	0.176(0.006)	**0.163(0.003)**	
			15.7%	35.4%	77.4%	
l_Δ	$n = 50$	0.173(0.032)	0.169(0.025)	0.166(0.025)	**0.161(0.023)**	0.111
			5.4%	9.8%	19.5%	
	$n = 150$	0.151(0.019)	0.149(0.014)	0.145(0.015)	**0.140(0.008)**	
			5.1%	13.1%	28.8%	
	$n = 500$	0.139(0.011)	0.135(0.007)	0.130(0.007)	**0.120(0.003)**	
			14.6%	31.7%	65.8%	
	$n = 1500$	0.132(0.005)	0.130(0.004)	0.123(0.004)	**0.115(0.002)**	
			16.0%	40.9%	80.6%	

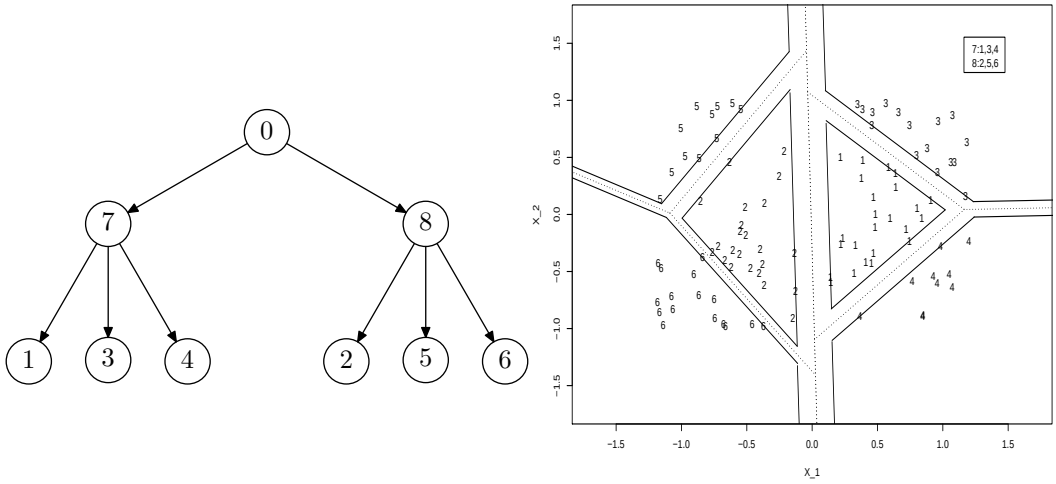
Table 2.4: Averaged test errors as well as estimated standard errors (in parenthesis) of SVM, HSVM and HPSI, in the gene function example in Section 2.4.2, over 100 simulation replications. The testing errors are computed under l_{0-1}, l_{Δ}, l_{H-sib} and l_{H-sub}. The bold face represents the best performance among four competitors for any given loss.

Linear				
Training	Test error			
Methods	l_{0-1}	l_{Δ}	l_{sib}	l_{sub}
SVM	0.972(0.006)	0.651(0.024)	0.926(0.008)	0.593(0.029)
HSVM	0.960(0.009)	0.520(0.023)	0.918(0.022)	0.433(0.041)
% 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)**
% of impro.	1.4%	20.6%	1.0%	27.5%

Gaussian				
Training	Test error			
Methods	l_{0-1}	l_{Δ}	l_{sib}	l_{sub}
SVM	0.976(0.002)	0.669(0.005)	0.921(0.003)	0.617(0.007)
HSVM	0.961(0.008)	0.515(0.020)	0.920(0.019)	0.421(0.030)
% 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)**
% of impro.	1.6%	23.5%	0.1%	32.3%

Table 2.5: Verification of 10 gene predictions using an updated MIPS system and their function categories.

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



(a) Hierarchical structure

(b) Geometric margin

Figure 2.1: Plot of generalized geometric margin with respect to $\mathcal{H}$ defined by a nine-node tree (right) , labelled as $\{0, 1, \cdots, 8\}$. Geometric margins between two subtrees and between different classes within the subtrees are displayed by solid lines, and classification boundaries are displayed by dotted lines. Class 7 consists of classes 1, 3 and 4, and class 8 consists of classes 2, 5 and 6.

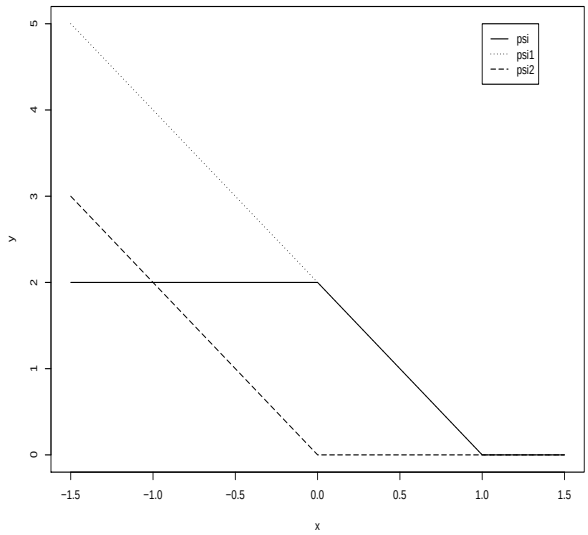


Figure 2.2: Plot of ψ, ψ_1 and ψ_2, for DC decomposition of $\psi = \psi_1 - \psi_2$. Solid, dotted and dashed lines represent ψ, ψ_1 and ψ_2.

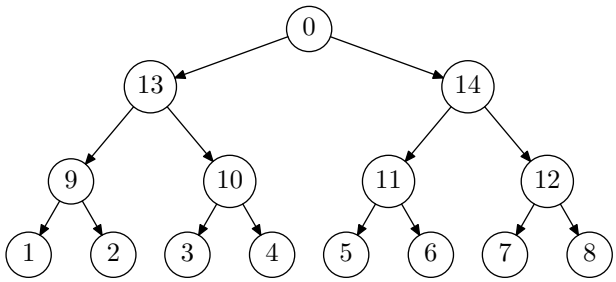


Figure 2.3: Plot of a complete binary tree with depth $p = 3$ and $k = 2^p$ leaf nodes, which is the hierarchy used in Example 1 of Section 2.4.1.

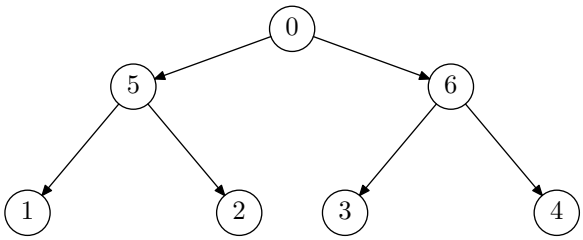
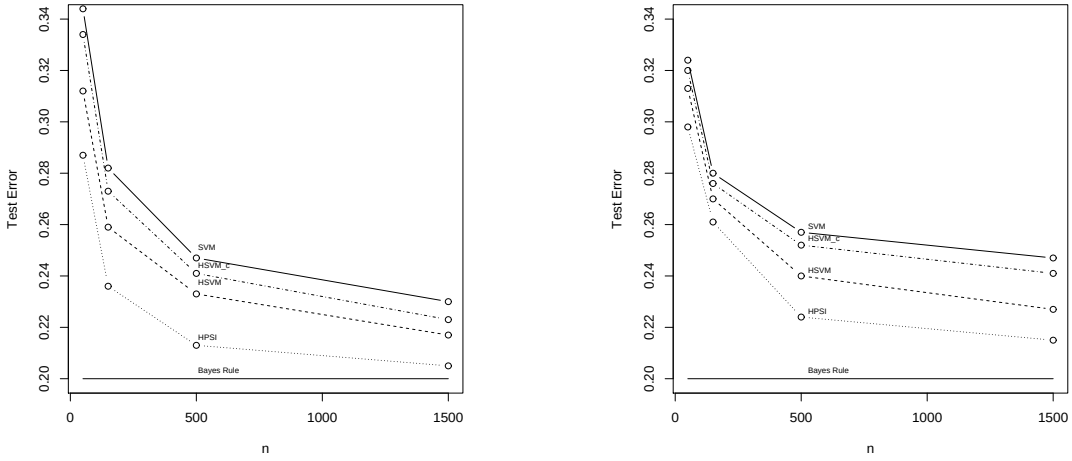


Figure 2.4: Plot of the tree with depth $p = 2$ and $k = 4$ leaf nodes, which is the hierarchy used in Example 2 of Section 2.4.1. Nodes 1 and 2, and 3 and 4 are two pairs of offsprings for Node 5 and 6, 5 and 6 are the offsprings of root node 0.



(a) Linear

(b) Gaussian kernel

Figure 2.5: Plot of the generalization errors of SVM, HSVM$_c$, HSVM and HPSI under l_{0-1} as a function of sample size $n = 50, 150, 500, 1500$ for linear and Gaussian kernel learning in Example 2. Five lines from top to bottom represent SVM, HSVM$_c$, HSVM, HPSI and generalization error of Bayes rule, respectively. Generalization error of Bayes rule is 0.20.

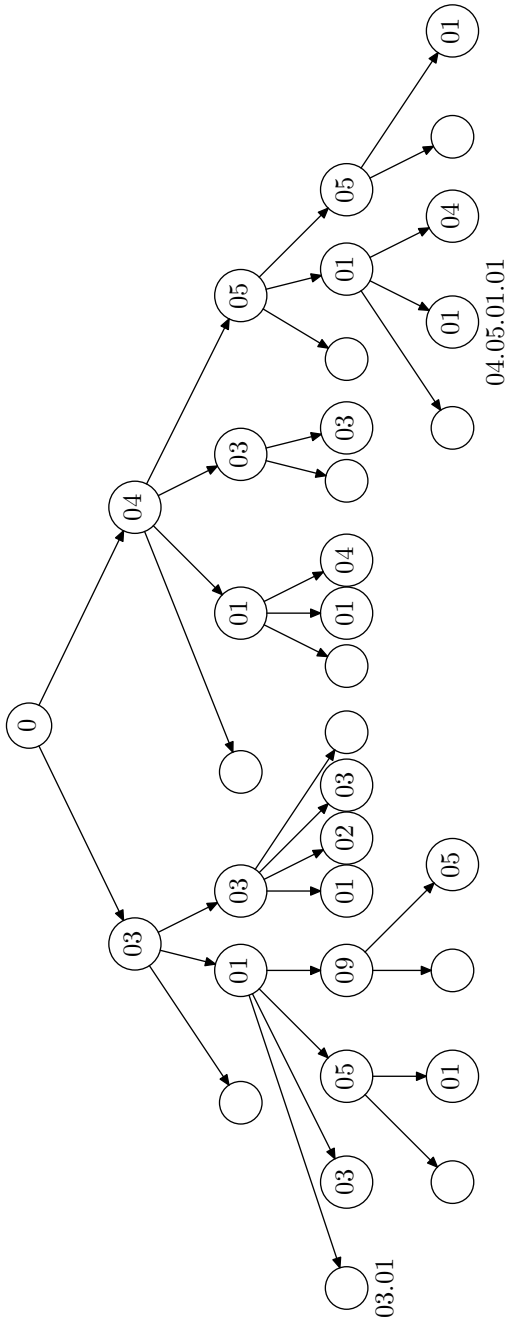


Figure 2.6: Two major branches of MIPS, with two functional categories at the highest level: "Cell cycle and DNA processing" and "Transcription". The number inside each circle stands for one functional class within its parent class. The blank node is numbered by the node and its ancestors. Combining all numbers in each node as well as those in its ancestors yields the exact identification of the corresponding functional category. For instance, the middle node 01 at level 4 stands for functional category 04.05.01.01, which is "General transcription activities" in MIPS.

Chapter 3

Unequal loss multi-classification

3.1 Introduction

In m-class classification, a training sample $\{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i); i = 1, \cdots, n\}$, which is independently identically distributed according to an unknown distribution $P(\boldsymbol{x}, y)$, is given, where $\boldsymbol{x}_i \in S \subset \mathbb{R}^q$ is a vector of predictors and label y_i is coded as $\{1, \cdots, m\}$. Based on $\{\boldsymbol{z}_i : i = 1, \cdots, n\}$, a decision function vector $\boldsymbol{f} = (f_1, \cdots, f_m)$ is estimated, with f_j representing class j, mapping from $S \subset \mathbb{R}^q$ to $\mathbb{R}$. To avoid redundancy in $\boldsymbol{f}$, a zero-sum constraint $\sum_{j=1}^{m} f_j = 0$ is enforced. Before a decision rule is introduced, the cost for misclassification needs to be considered. Denote by $\mathcal{C}(j, k)$ the cost for misclassifying class j to class k for $j \neq k$, and $\mathcal{C}(j, j) = 0$ for correct classification. Let $\mathcal{C} = (\mathcal{C}(j, k))_{m \times m}$ be the matrix for misclassification costs, with the jkth elements $\mathcal{C}(j, k)$. Without loss of generality, each element $\mathcal{C}(j, k)$ of $\mathcal{C}$ can be standardized by dividing each element by $\max_{1 \leq j,k \leq m}\{\mathcal{C}(j, k)\}$, to reflect the relative severance of each misclassification.

This chapter is organized in seven sections. Section 2 introduces the methodology

for multi-class margin classification with un-equal cost. Section 3 concerns minimization involved in this methodology. Section 4 discusses the Bayes rule in multi-class SVM. Section 5 compares our framework with some other methods. Section 6 explores some numerical examples. The results are discussed and summarized in Section 7 and some technical details are provided in Appendix.

3.2 Multi-classification with SVM

Taking the misclassification into account, we now define, for a given new input $\boldsymbol{x} \in S$, the decision rule or the classifier $d(\boldsymbol{x})$ to be: $d(\boldsymbol{x}) = \text{argmin}_{k=1,\cdots,m} \sum_{j=1}^{m} f_j(\boldsymbol{x})\, \mathcal{C}(j,k)$. That is, $d(\boldsymbol{x})$ assigns $\boldsymbol{x}$, $\boldsymbol{x} \in S$, to the class having the minimum value $\{\sum_{j=1}^{m} f_j(\boldsymbol{x})\, \mathcal{C}(j,k)\}$ over $j = 1, \cdots, m$.

In this framework, the generalization error (GE) that reflects different costs is defined as

$$\text{Err}(\boldsymbol{f}) = E\mathcal{C}(Y, d(\boldsymbol{X})) = E(\sum_{j=1}^{m} \mathcal{C}(Y,j)(1 - I\{d(\boldsymbol{X}) \neq j\})). \tag{3.1}$$

The corresponding empirical generalization error becomes

$$\begin{aligned} & n^{-1} \sum_{\substack{i=1,\cdots,n; \\ j=1,\cdots,m}} \mathcal{C}(y_i, j)(1 - I\{d(\boldsymbol{x}_i) \neq j\}) \\ = \;& T + n^{-1} \sum_{\substack{i=1,\cdots,n; \\ j=1,\cdots,m}} (\max_{l,k}\{\mathcal{C}(l,k)\} - \mathcal{C}(y_i, j)) I\{d(\boldsymbol{x}_i) \neq j\}, \end{aligned} \tag{3.2}$$

where T is $n^{-1} \sum_{\substack{i=1,\cdots,n; \\ j=1,\cdots,m}} \mathcal{C}(y_i, j) - (m-1) \max_{j,k}\{\mathcal{C}(j,k)\}$, which is a constant with respect to decision function $d(\cdot)$.

To construct the cost function for large margin classification, we first introduce the notation of a generalized functional margin for class j: $\boldsymbol{u}^{(j)}(\boldsymbol{f}, z) = (u_1^{(j)}, u_2^{(j)}, \cdots, u_k^{(j)}, \cdots,$

$u_m^{(j)})^T$; $k = 1, \cdots, m$, $k \neq j$, where $u_k^{(j)} \equiv u_k^{(j)}(\boldsymbol{f}, z) = \sum_{l=1}^{m} f_l(\boldsymbol{x})(\mathcal{C}(l,k) - \mathcal{C}(l,j))$ comparing class j against class k with respect to classification with un-equal costs. With the generalized margin in place, $I\{j \neq d(\boldsymbol{x}_i)\}$ in (3.1) becomes $I\{\boldsymbol{u}_{\min}^{(j)}(\boldsymbol{x}_i) < 0\}$, which is a function of the generalized margin for class j, where $\boldsymbol{u}_{\min}^{(j)} = \min_{1 \leq k \leq m; k \neq j}\{\boldsymbol{u}_k^{(j)}\}$.

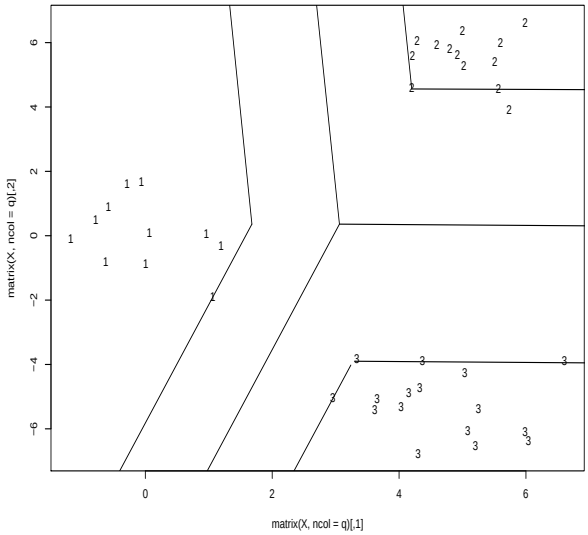


Figure 3.1: Plot of the geometric margin in three classification, with each class labeled as $\{1, 2, 3\}$. The cost matrix used here is the matrix $\mathcal{C}$, defined as $\mathcal{C}(i,i) = 0$; $i = 1, 2, 3$, $\mathcal{C}(1,2) = \mathcal{C}(2,1) = \mathcal{C}(1,3) = \mathcal{C}(3,1) = 1$, and $\mathcal{C}(2,3) = \mathcal{C}(3,2) = 0.9$.

Now we are ready to introduce the cost function in margin classification with un-equal cost through regularization. In (3.3), a surrogate loss $L(\cdot)$ is used to replace intractable $I(d(\boldsymbol{x}_i) \neq j)$ to yield

$$\min_{\boldsymbol{f}} \left\{ J(\boldsymbol{f}) + C \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\max_{j,k}\{\mathcal{C}(j,k)\} - \mathcal{C}(y_i, j)\right) L(\boldsymbol{u}_{\min}^{(j)}(\boldsymbol{f}, z_i)) \right\}, \tag{3.3}$$

subject to the-sum-to-zero constraints: $\sum_{j=1}^{m} f_j(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in S$, where $C(C > 0)$ is

a regularizer regularizing geometric margin and training, and $J(\boldsymbol{f})$ is a penalty that is defined by the geometric margin to be explained next.

To define the geometric margin and $J(\boldsymbol{f})$, we use a generic notation $f_j(x) = \boldsymbol{w}_j^T \tilde{x} + b_j$; $j = 1, \cdots, m$ for representations, where $\tilde{x} = \boldsymbol{x}$ for linear representations and $\tilde{x} = (K(\boldsymbol{x}_1, \cdot), \cdots, K(\boldsymbol{x}_n, \cdot))$ for kernel learning. The geometric margin is defined to be $\min_{1 \le k \le l \le m} \{\gamma_{k,l}\}$, where $\gamma_{k,l} = \frac{2}{\| \sum_{j=1}^m (\mathcal{C}(l,k) - \mathcal{C}(l,j)) \boldsymbol{w}_j \|_2^2}$ is the usual L_2 separation margin, and $\gamma_{k,l} = \frac{2}{\| \sum_j (\mathcal{C}(l,k) - \mathcal{C}(l,j)) \boldsymbol{w}_j \|_p^2}$ is the usual L_p separation margin for $p = 1, \infty$. These margins represent the vertical Euclidean distance between two parallel hyper planes $\sum_{j=1}^m f_j \, \mathcal{C}(j,k) - \sum_{j=1}^m f_j \, \mathcal{C}(j,l) = \pm 1$, as measured by different norms; see Wang and Shen (2006) for a reference.

For large margin classification with un-equal cost, (3.3) yields different classifiers with different choice of margin loss $L(z)$. Margin losses include, but are not limited to, the hinge loss $L(z) = (1-z)_+$ for SVM with its variants $L(z) = (1-z)_+^q$ for $q > 1$; c.f., Lin (2002); the ρ-hinge loss $L(z) = (\rho - z)_+$ for nu-SVM (Schölkopf, Smola, Williamson and Bartlett, 2000) with $\rho > 0$ to be optimized; the ψ-loss $L(z) = \psi(z)$, with $\psi(z) = 1 - \text{Sign}(z)$ if $z \ge 1$ or $z < 0$, and $2(1-z)$ otherwise, c.f., Shen, Tseng, Zhang and Wong (2003), the logistic loss $L(z) = \log(1 + e^{-z})$, c.f., Zhu and Hastie (2005); the sigmoid loss $L(z) = 1 - \tanh(cz)$; c.f., Mason, Baxter, Bartlett and Frean (2000). Moreover, (3.3) covers the standard preceding classifiers with equal cost when $\mathcal{C}(j,k)$ for $j \ne k$ is 1 across j, k; see for example Liu and Shen (2006) for multi-class SVM and ψ-learning. In the binary case, (3.3) reduces to the binary SVM of Lin (2003) when $L(z)$ is the hinge loss.

Maximizing the minimum of the $\gamma_{k,l}$'s is equivalent to minimizing

$$\max_{1\le k\le l\le m} \| \sum_j \mathcal{C}(j,k)\boldsymbol{w}_j - \sum_j \mathcal{C}(j,l)\boldsymbol{w}_j\|^2.$$

As a technical remark, it should be noted that, in (3.4), in order to minimize the above, we minimize $\sum_{1\le k\le m} \| \sum_j \mathcal{C}(j,k)\boldsymbol{w}_j\|^2$ instead. The reason is that these two play a similar role and the latter one is easier to work with.

As suggested in Figure 1, it is possible that some instances (x_i's) may fall inside the margin zone $\{\boldsymbol{x} : |\sum_{j=1}^m f_j(\boldsymbol{x})\,\mathcal{C}(j,k) - \sum_{j=1}^m f_j(\boldsymbol{x})\,\mathcal{C}(j,y_i)| \le 1\};\ k = 1,\cdots,m$, even in the separate case, for any fixed values of tuning parameter C. This occurs when the cost is unequal, which is in contrast to the case of equal cost. In the case of equal cost, minimizing (3.3) yields the loss evaluated at each instance to be zero in the separate case, which in turn leads to the phenomenon that no instances occur inside the margin zone. In the case of unequal cost, however, this is no longer the case. In this process, the unequal cost $C(l,k)$ evidently plays a salient role that each instance can be completely pushed out of the margin zone as in the case of equal cost.

3.3 Minimization

This section implements (3.3) for the L_2-norm MSVM (L2MSVM) in the generic form for both linear and kernel learning scenarios. In the linear case, $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T\tilde{\boldsymbol{x}} + b_j$, with $\boldsymbol{w}_j \in \mathbb{R}^q\ \forall 1 \le j \le m$. For L2MSVM with un-equal cost, (3.3) reduces to

$$\min_{\boldsymbol{w}_j, b_j, j=1,2,\cdots,m} \left\{ \frac{1}{2}\sum_{l=1}^m \| \sum_{k=1}^m \mathcal{C}(k,l)\boldsymbol{w}_k\|^2 + C\sum_{i=1}^n\sum_{j=1}^m \left(\max_{j,k}\{\mathcal{C}(j,k)\} - \mathcal{C}(y_i,j)\right)\left(1 - \boldsymbol{u}_{\min}^{(j)}(\boldsymbol{x}_i)\right)_+ \right\}, \tag{3.4}$$

subject to $\sum_j f_j(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in S$.

By Theorem 2.1 of Liu and Shen(2006), minimization in (3.4) subject to the infinite constraints $\sum_j f_j(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in S$ is equivalent to the constraints for the training sample $\{\boldsymbol{x}_i; i = 1, \cdots, n\}$, i.e., $\sum_{j=1}^n b_j \mathbf{1}_n + \sum_{j=1}^n X\boldsymbol{w}_j = \mathbf{0}_n$, where $\mathbf{1}_n$ and $\mathbf{0}_n$ are the n-dimensional vectors of 1's and 0's respectively, and $X = (\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n)^T$ is the design matrix.

By introducing slack variables $\xi_{i,j}$; $i = 1, \cdots, n; j = 1, \cdots, m$, (3.5) is equivalent to (3.4):

$$\min_{\xi_{i,j}, \boldsymbol{w}_j, b_j} \left\{ \frac{1}{2} \sum_{1 \le k \le m} \| \sum_{1 \le j \le m} \mathcal{C}(j,k) \boldsymbol{w}_j \|^2 + C \sum_{i=1}^n \sum_{j=1}^m \left(\max_{j,k} \{\mathcal{C}(j,k)\} - \mathcal{C}(y_i, j) \right) \xi_{i,j} \right\}, \quad (3.5)$$

subject to

$$\left(\sum_{l=1}^m (\langle \boldsymbol{w}_l, \tilde{\boldsymbol{x}}_i \rangle + b_l) \mathcal{C}(l,k) - \sum_{l=1}^m (\langle \boldsymbol{w}_l, \tilde{\boldsymbol{x}}_i \rangle + b_l) \mathcal{C}(l,j) \right) \ge 1 - \xi_{i,j}, \forall k \ne j, \quad (3.6)$$

$$\xi_{i,j} \ge 0; \forall i = 1, 2, \cdots, n; j = 1, 2, \cdots, m; k = 1, 2, \cdots, m; k \ne j, \quad (3.7)$$

and

$$\tilde{X} \sum_{j=1}^n b_j \mathbf{1}_n + \sum_{j=1}^n X \boldsymbol{w}_j = \mathbf{0}_n. \quad (3.8)$$

After introducing the Lagrange multipliers $\alpha_{i,j,k}; j \ne k$ for (3.6), $\beta_{i,j}$ for (3.7), and $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_n)^T$ for (3.8), we obtain the dual form of (3.4) in Theorem 3, ignoring irrelevant constants.

Theorem 3 *The dual form of (3.5) is,*

$$\min_{\alpha_{i,j}, \delta_i} \left\{ \frac{1}{2} \boldsymbol{\gamma}^T H \boldsymbol{\gamma} + \boldsymbol{b}^T \boldsymbol{\gamma} \right\}, \quad (3.9)$$

subject to, $B\boldsymbol{\gamma} \le \mathbf{0}_{nm(m-1)}, D\boldsymbol{\gamma} \le C\mathcal{C}_Y$, *and* $A\boldsymbol{\gamma} = \mathbf{0}_m$.

In Theorem 3, $\boldsymbol{\gamma}$ is the (nm^2-nm+n)-dimensional vector obtained from the (nm^2+n)-dimensional vector $\boldsymbol{\gamma}^0 = Vec(\alpha_{1,1,1}, \cdots, \alpha_{1,1,m}, \alpha_{1,2,1}, \cdots, \alpha_{1,m,m}, \alpha_{2,1,1}, \cdots, \alpha_{n,m,m}, \boldsymbol{\delta})$ by removing the $r_{i,j}$th element, where $r_{i,j} = (i-1)mm + (j-1)m + j$ corresponding to the redundant $\alpha_{i,j,j}$; $i = 1, \cdots, n$; $j = 1, \cdots, m$. Let $H = U^T \cdot U$, where U is a $(mq) \times (nm^2 - nm + n)$ matrix obtained from a $(mq) \times (nm^2 + n)$ matrix U^0 by removing the $r_{i,j}$th column of U^0; $i = 1, \cdots, n$; $j = 1, \cdots, m$. And

$$U^0 = U^{(1)} \cdot (U^{(2)} \cdot (U^{(3)} - U^{(4)}), U^{(5)}),$$

where $U^{(1)} = I_{\tilde{q}} \otimes \mathcal{C}^{-1}$, $U^{(2)} = X^T \otimes I_m$, $U^{(3)} = I_n \otimes \mathbf{1}_m^T \otimes \mathcal{C}$, $U^{(4)} = I_n \otimes \mathcal{C} \otimes \mathbf{1}_m^T$, and $U_{(5)} = X^T \otimes \mathbf{1}_m$.

Here $X = (\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \cdots, \tilde{\boldsymbol{x}}_n)^T$, $\tilde{q}$ is the dimension of the vector $\tilde{\boldsymbol{x}}$, and I is the identity matrix. Other notations, including vector $\boldsymbol{b}$, matrix A, B, and D are defined as follows:

$$\boldsymbol{b} = Vec(-\mathbf{1}_{nm(m-1)}, \mathbf{0}_n),$$

$$A = [(\mathbf{1}_n^T \otimes I_m) \cdot (U^{(3)} - U^{(4)}), \mathbf{1}_{m \times n}],$$

$$B = -[I_{n \cdot m \cdot (m-1)}, \mathbf{0}_{(n \cdot m \cdot (m-1)) \times n}],$$

$$D = [I_{nm} \otimes \mathbf{1}_{m-1}^T, \mathbf{0}_{(nm) \times n}].$$

$\mathcal{C}_Y$ is the vector of length nm, obtained from $Vec(\mathcal{C}(y_1, \cdot), \mathcal{C}(y_2, \cdot), \cdots, \mathcal{C}(y_n, \cdot))$, where $\mathcal{C}(y_i, \cdot)$ is the y_ith row vector of the matrix $\mathcal{C}$. Here $Vec(\cdot)$ function for a matrix is to span the matrix by columns. And $\mathbf{1}_{(\cdot)\times(\cdot)}$ and $\mathbf{0}_{(\cdot)\times(\cdot)}$ are the matrices with all the elements being 1 and 0, respectively.

Then (3.9) yields the solution of $Vec(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m) = (I_{\tilde{q}} \otimes \mathcal{C}^{-1}) U^0 \boldsymbol{\gamma}^0$, with all the $r_{i,j}$th elements of $\boldsymbol{\gamma}^0$ being 0; $i = 1, \cdots, n$; $j = 1, \cdots, m$. After getting $\boldsymbol{w}_j$'s, we can solve

the KKT conditions or solve the primal problem directly to get the solution for b_j's.

3.4 Bayes decision rule

This section derives the risk minimizer for MSVM, which leads to the Bayes rule in the case of un-equal cost. Toward this end, we derive the theoretically best decision rule based on conditional probability $p_j(\boldsymbol{x}) = \Pr(Y = j|\boldsymbol{X} = \boldsymbol{x})$; $j = 1, \cdots, m$. In the case of equal cost, the Bayes rule is obtained by minimizing the expected misclassification risk, defined as $d_B(\boldsymbol{x}) = \text{argmin}_{1\le j\le m}\{1 - p_j(\boldsymbol{x})\} = \text{argmax}_{1\le j\le m}\{p_j(\boldsymbol{x})\}$. In the case of uneqal cost, the 0-1 loss is replaced by a weighted loss induced by matrix $\mathcal{C}$, i.e., $\mathcal{C}(y, d(\boldsymbol{x}))$ for $d(\cdot)$ at $\boldsymbol{z} = (\boldsymbol{x}, y)$. The corresponding Bayes rule becomes

$$d_B(\boldsymbol{x}) = \underset{1\le j\le m}{\text{argmin}}\{\sum_{k=1}^{m} \mathcal{C}(k, j)p_k(\boldsymbol{x})\}. \tag{3.10}$$

As showed in Theorem 4, the risk minimizer of our proposed MSVM yields the same Bayes rule as in (3.10).

Theorem 4 *The Bayes decision vector $\bar{\boldsymbol{f}}$ satisfies $\boldsymbol{u}_{\min}^{(j)}(\boldsymbol{x}) > 0$, when $\boldsymbol{f} = \bar{\boldsymbol{f}}$, and $j = \text{argmin}_k \sum_l \mathcal{C}(l, k)p_l(\boldsymbol{x})$, and $\boldsymbol{u}_{\min}^{(j)}$ is as a function of $\boldsymbol{f}$, as defined in Section 2.*

3.5 Connection with existing methods

This section establishes a connection with some existing methods, including multi-category classification with equal-cost in Liu and Shen (2006), and structured classification in Tsochanridis, et al. (2004). Within our framework, it recovers to the usual equal-cost (flat) multi-class classification, when the cost matrix is the identity matrix.

3.5.1 Multi-classification with equal cost

We will show that equal cost multi-classification is a special case of our framework. As shown in (3.4), when $\mathcal{C}(j,k)$ is set to be 1 for $j \neq k$, and 0 otherwise, and (3.4) reduces to

$$\min_{\boldsymbol{w}_j, b_j, j=1,2,\cdots,m} \left\{ \frac{m-1}{2} \sum_{l=1}^{m} \| \sum_{k=1}^{m} \mathcal{C}(k,l) \boldsymbol{w}_k \|^2 + C \sum_{i=1}^{n} \left(1 - \boldsymbol{u}_{\min}^{(y_i)}(\boldsymbol{x}_i) \right)_+ \right\},$$

subject to $\sum_{j=1}^{n} b_j \mathbf{1}_n + \sum_{j=1}^{n} X \boldsymbol{w}_j = \mathbf{0}_n$. This reduced to the equal cost multi-category SVM and ψ-learning, with L serving as the generalized hinge loss and ψ-loss respectively (Liu and Shen 2006).

3.5.2 Multi-classification with unequal cost by rescaling methods

In Tsochanridis, et al. (2004), a generalization of SVM for structured output spaces was proposed. Their framework for un-equal cost multi-classification focused on the decision loss for each instance. For each pair of training instance $(\boldsymbol{x}_i, y_i)$, let $\hat{y}_i$ be $d(\boldsymbol{x}_i)$, the original hinge loss for $\hat{y}_i \neq y_i$ is rescaled by the cost for misclassifying y_i as $\hat{y}_i$. Then they solved the following minimization problem,

$$\min_{\boldsymbol{w}} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \mathcal{C}(y_i, y)(1 - \langle \boldsymbol{w}, \delta\Psi_i(y) \rangle)_+ \right\}.$$

This is equivalent to rescaling the corresponding slack variable for the i'th instance in the Lagrange primal problem.

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \right\},$$
$$\text{s.t. } \xi_i \geq 0, \langle \boldsymbol{w}, \delta\Psi_i(y) \rangle \geq 1 - \frac{\xi_i}{\mathcal{C}(y_i, y)}.$$

Also, a second way that taking the un-equal cost function into account was proposed by Taskar et al. (2004). They changed the margin constraints by rescaling the margin with

the cost for misclassifying y_i as $\hat{y}_i$. It is very similar to the approach of rescaling the slack variables.

In Lee, et al. (2004), they also introduced a multi-class SVM for the non-standard situation, with different coding. Under their setup, the minimization problem is equivalent to

$$\min_{\boldsymbol{w}} \left\{ \frac{1}{2}\|\tilde{\boldsymbol{w}}\|^2 + C\sum_{i=1}^{n} \mathcal{C}(y_i, y)(\frac{1}{2} - \langle \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{x}}_i \rangle)_+ \right\},$$

which is very similar to the rescaling slack variable method.

Although these approaches provide good results in many applications, it is still not very clear why the rescaling approaches for the loss of $\hat{y}_i \neq y_i$ will achieve the desired Bayes rule.

3.6 Numerical examples

3.6.1 Multiclass classification with unequal cost

Now consider the three class linear classification in which a training sample $(X_{i1}, X_{i2}, Y_i); i = 1, \cdots, 300$ is generated as follows. First, Y_i is assigned to class label $\{1, 2, 3\}$ with equal probability. Second, $\{U_{i,j}\}; i = 1, \cdots, 300; j = 1, 2$ is sampled from $N(0, 1)$. Third, $X_{i,j} = U_{i,j} + a_j$ is transformed with $(a_1, a_2) = (0, 0)$,$(1, 1)$ and $(1, -1)$ for these three classes. This generates a nonseparable case.

Three methods are compared, including MSVM with equal cost, denoted by EMSVM, rescaling approach of Tsochanridis et al. (2004), denoted by SMSVM, and MSVM with un-equal cost defined in (3.4). The cost matrix for the second and third methods is defined

as

$$\mathcal{C} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 4 \\ 1 & 4 & 0 \end{pmatrix}. \tag{3.11}$$

For EMSVM and MSVM in (3.4), numerical analyses are performed in R2.1.1. For SMSVM, numerical analyses are performed in SVM$^{\text{struct}}$, where SVM$^{\text{struct}}$ is available at http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html.

In this example, the performance of classification is measured by a test error of an independent sample of size 5×10^4, sampled from the true distribution. This test error approximates the generalization error in (3.1) with the cost matrix (3.11). The test error is minimized with respect to the tuning parameter over a set of discrete grid points $\{10^{i/2} : i = -4, -3, \cdots, 5\}$ to eliminate the dependency of classifiers on it. The test error, averaged over 100 simulation replications, is computed for each classifier, together with its standard error. The results are summarized in Table 1.

As indicated in Table 1 and Figure 3.2, MSVM yields smaller test errors as compared to EMSVM, whereas SMSVM performs slightly better than flat classification EMSVM. Our MSVM outperforms EMSVM and SMSVM, which achieves the desired objective.

Appendix

Proof of Theorem 3:

To derive the dual form for (3.5), we first introduce the Lagrange multipliers

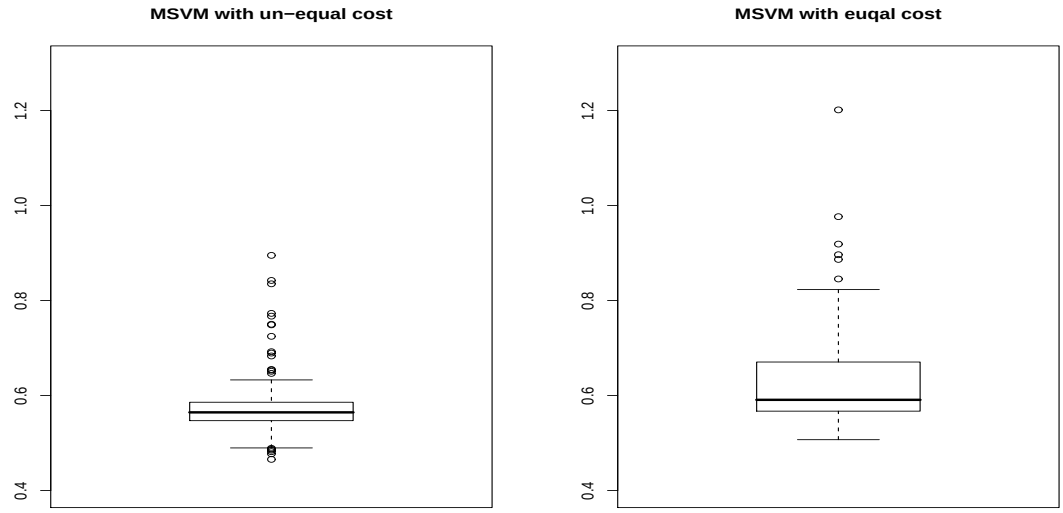


Figure 3.2: Boxplots for the test errors of EMSVM and MSVM for the example in Section 3.6.1, over 100 simulation replications. For each set of training data, the test error is calculated over a testing sample of 5×10^4 instances based on the minimal test error from the grid search over tuning parameter C.

$\boldsymbol{\alpha}_i = (\alpha_{i,1}, \cdots, \alpha_{i,m})^T \in \mathbb{R}^n$, $\boldsymbol{\beta}_i = (\beta_{i,1}, \cdots, \beta_{i,m})^T \in \mathbb{R}^n$; $i = 1, \cdots, n$ and $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_n)^T \in \mathbb{R}^n$, respectively for the constraints of the (3.5): (3.6)-(3.8). Then the primal problem (3.5) is equivalent to the dual problem:

$$\begin{aligned}\max_{\alpha_{i,j,k},\beta_{i,j,k},\boldsymbol{\delta}} L \;=\; & \frac{1}{2}\sum_{1\le l\le m}\| \sum_{1\le k\le m} \mathcal{C}(k,l)\boldsymbol{w}_k\|^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\max_{j,k}\{\mathcal{C}(j,k)\} - \mathcal{C}(y_i,j)\right)\xi_{i,j} \\ & + \sum_{i,j,k}\alpha_{i,j,k}(1-\xi_{i,j} - \left(\sum_{l=1}^{m}(\langle \boldsymbol{w}_l, \tilde{\boldsymbol{x}}_i\rangle + b_l)\,\mathcal{C}(l,k) - \sum_{l=1}^{m}(\langle \boldsymbol{w}_l, \tilde{\boldsymbol{x}}_i\rangle + b_l)\,\mathcal{C}(l,j)\right)) \\ & - \sum_{i,j}\beta_{i,j}\xi_{i,j} + \boldsymbol{\delta}^T\cdot(\tilde{X}\sum_{j}\tilde{\boldsymbol{w}}_j)\end{aligned} \tag{3.12}$$

Table 3.1: Averaged test errors and their standard errors (in parenthesis) of EMSVM, SMSVM and MSVM in the example in Section 3.6.1, over 100 simulation replications. For each set of training data, the test error is calculated over a testing sample of 5×10^4 instances based on the minimal test error from the grid search over tuning parameter C.

	Test error with un-equal cost
EMSVM	0.625 (0.087)
SMSVM	0.623 (0.072)
MSVM	0.581 (0.075)

subject to

$$\alpha_{i,j,k} \geq 0,\ \beta_{i,j} \geq 0 \tag{3.13}$$

after differentiating L with respect to $(\xi_{i,j}, w_{j,h}, b_j)$, we obtain

$$\begin{aligned} \frac{\partial L}{\partial \xi_{i,j}} = d_{i,j}^{(1)} &= C \cdot (\max_{j,k}\{\mathcal{C}(j,k)\} - \mathcal{C}(y_i,j)) - \\ &\quad \sum_k \alpha_{i,j,k} - \beta_{i,j} = 0, \forall i = 1,\cdots,n; j = 1,\cdots,m \end{aligned} \tag{3.14}$$

$$\begin{aligned} \frac{\partial L}{\partial w_{j,h}} = d_{j,h}^{(2)} &= \sum_{1\leq l\leq m} \mathcal{C}(j,l)\,(\sum_{1\leq k\leq m} \mathcal{C}(k,l)\,\boldsymbol{w}_{k,h}) - \\ &\quad \sum_{i,l,k} \alpha_{i,l,k}(\tilde{x}_{i,h}\,(\mathcal{C}(j,k) - \mathcal{C}(j,l))) + \sum_{i=1}^{n} \delta_i \tilde{x}_{i,h} = 0, \\ &\quad \forall j = 1,\cdots,m; h = 1,\cdots,\tilde{q};\ \text{where } \tilde{q} \text{ is the dimension of } \tilde{\boldsymbol{x}} \end{aligned} \tag{3.15}$$

$$\frac{\partial L}{\partial b_j} = d_j^{(3)} = -\sum_{i,l,k} \alpha_{i,l,k}(\mathcal{C}(j,k) - \mathcal{C}(j,l)) + \sum_{i=1}^{n} \delta_i = 0, \forall j = 1,\cdots,m \tag{3.16}$$

Let $\mathcal{C}^{-1} = (\mathcal{C}'_{k,l})_{m\times m}$. After substituting (3.14)-(3.16) into (3.12), we simplify L as

$$L = \left(\sum_{i,j,k} \alpha_{i,j,k}\right) - \frac{1}{2}\sum_{h}\sum_{k'}\left(\sum_{j} \mathcal{C}'_{j,k'}\left(-\sum_{i,l,k}\alpha_{i,l,k}(\tilde{x}_{i,h}\left(\mathcal{C}(j,k)-\mathcal{C}(j,l)\right)) + \sum_{i=1}^{n}\delta_i\tilde{x}_{i,h}\right)\right)^2 \quad (3.17)$$

Consequently, using the notation defined in **Theorem 3**, maximizing L is equivalent to minimizing $-L$, which is

$$\frac{1}{2}\boldsymbol{\gamma}^T U^T U\boldsymbol{\gamma} + \boldsymbol{b}^T\boldsymbol{\gamma}$$

From (3.14) $\beta_{i,j} = C(\max_{j,k}\{\mathcal{C}(j,k)\} - \mathcal{C}(y_i,j)) - \sum_k \alpha_{i,j,k}$. Hence (3.13) is the same as $B\boldsymbol{\gamma} \leq \mathbf{0}_{nm(m-1)}, D\boldsymbol{\gamma} \leq C\mathcal{C}_Y$. In addition (3.16) reduces to $A\boldsymbol{\gamma} = \mathbf{0}_m$, and (3.15) leads to the solution of $w_{j,h}$: $Vec(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_m) = (I_{\tilde{q}}\otimes\mathcal{C}^{-1})U^0\boldsymbol{\gamma}^0$, with the $r_{i,j}$th element of $\boldsymbol{\gamma}^0$ being 0; $i = 1,\cdots,n$; $j = 1,\cdots,m$. Then the desired result follows.

Proof of Theorem 4:

Since $\bar{\boldsymbol{f}}$ is the Bayes decision vector, $\text{argmin}_k \sum_l \mathcal{C}(l,k)\bar{f}_l(\boldsymbol{x}) = \text{argmin}_k \sum_l \mathcal{C}(l,k)p_l(\boldsymbol{x})$, i.e., for $\boldsymbol{f} = \bar{\boldsymbol{f}}$, and $j = \text{argmin}_k \sum_l \mathcal{C}(l,k)p_l(\boldsymbol{x})$, $\boldsymbol{u}_k^{(j)} > 0$ for all the $1 \leq k \leq m, k \neq j$. Then, the desired result follows.

3.7 Summary

In this chapter, we have proposed a novel framework for multi-classification with margin based methods, with possible un-equal misclassification cost. The numerical experiments show that it can achieve better performance in compared to other existing methods, such as EMSVM, and SMSVM (Tsochanridis et al., 2004).

MSVM has been implemented for un-equal cost. Other large margin classifiers such as ψ-learning will be further investigated. Moreover, further investigation of operating characteristics of the proposed methods will be conducted, in addition to theoretical investigation of their statistical properties. To further expedite computation, we intend to derive a regularization path with respect to tuning parameter in (3.4) for model selection. In the literature, some research has been devoted to tuning in the equal cost situation, c.f., Hastie et al. (2004), Wang and Shen (2006).

Chapter 4

Discussion and Future Research

This thesis studies some topics in single-path hierarchical classification within the framework of multi-category classification. First, we introduce a novel framework for large margin hierarchical classification. Unlike in multi-classification, hierarchical classification has a distinctive property in that the class membership is not exclusive. To take into account the dependency among classes, we introduce a concept of margins inducing a classification loss function defined by the hierarchy. The large margin in this way fully integrates the hierarchical structure into large margin classifcation, and improves predictive performance of its flat counterpart. These aspects have been confirmed by both the theoretical and numerical analyses. Furthermore, single-path hierarchical classification is cast into the framework of un-equal cost multi-classification. A theoretical analysis indicates that our classifier converges to the Bayes classifier in this case.

In this thesis, only the 0-1 loss is considered as the target loss. It is possible that the

proposed approach can be extended to other hierarchical losses such as the symmetric difference loss and H-losses. To accommodate these losses, we modify (3.3) in Section 3.2:

$$\min_{\boldsymbol{f}} \left\{ J(\boldsymbol{f}) + C \sum_{i=1}^{n} \sum_{j=1}^{k} (\max_{j,m}\{\mathcal{L}^w(j,m)\} - \mathcal{L}^w(y_i,j))\, v(\boldsymbol{u}_{min}(\boldsymbol{f}(\boldsymbol{x}_i),j)) \right\}, \tag{4.1}$$

where $J(\boldsymbol{f})$ is the functional margin, $u_{min}(\boldsymbol{f}(\boldsymbol{x}_i),j)$ is the geometric margin induced by $\mathcal{H}$, $v(\cdot)$ can be any large margin surrogate loss used for SVM, ψ-learning and other large margin methods respectly, and $\mathcal{L}^w(j,m)$ is the mis-classification cost. For the 0-1 loss $\mathcal{L}^w(j,m) = 0$ if $j = m$, and $\mathcal{L}^w(j,m) = 1$ if $j \neq m$.

Note that, for 0-1 loss, since $\max_{j,m}\{\mathcal{L}^w(j,m)\} = 1$, and $\mathcal{L}^w(y_i,y_i) = 0$, $\sum_{j=1}^{k}(\max_{j,m}\{\mathcal{L}^w(j,m)\} - \mathcal{L}^w(y_i,j))\, v(\boldsymbol{u}_{min}(\boldsymbol{f}(\boldsymbol{x}_i),j))$ reduces to $v(\boldsymbol{u}_{min}(\boldsymbol{f}(\boldsymbol{x}_i),y_i)$. Hence (4.1) reduces to

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

which agrees with (2.3). For the symmetric difference loss, $\mathcal{L}^w(j,m) = l_{\triangle}(j,m) = |anc(j) \triangle anc(m)|$. For H-losses, $\mathcal{L}^w(j,m) = l_H(j,m) = c_s$, where s denotes the highest node yielding the disagreement between nodes j and m in $\mathcal{H}$, with c_s defined in (2.1) and (2.2) for l_{sub} and l_{sib}, respectively. In the setting of (2.3), and denoting $R(\boldsymbol{x},j) = \sum_{m \in E} \mathcal{L}^w(m,j) P(Y = m | \boldsymbol{X} = \boldsymbol{x})$ the Bayes risk for classifying $\boldsymbol{x}$ in leaf node j, the decision rule is modified as :

$$d(\boldsymbol{x}) = \tilde{j}, \iff \operatorname*{argmin}_{j} R(\boldsymbol{x},j) \in sub(\tilde{j}^p), \forall \tilde{j}^p \in anc(\tilde{j}).$$

Therefore, our estimated classifier is the top-down sequential rule generated from the minimizer of (4.1). Further investigation is necessary.

References

[1] An, L. and Tao, P. (1997). Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *J. Global Optimization*, **11**, 253-285.

[2] Broet, P., Lewin, A., Richardson, S., Dalmasso, C., and Magdelenat, H. (2004). A mixture model-based strategy for selecting sets of genes in multiclass respond microarray experiments. *Bioinformatics*, **20**, 2562-2571.

[3] Boser, B., Guyon, I., and Vapnik, V. N. (1992) A Training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Conference on Computational Learning Theory*, Pittsburgh, PA, 144-152.

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

[5] Cesa-Bianchi, N., Conconi, A. and Gentile, C. (2004). Regret bounds for hierarchical classification with linear-threshold functions. *Proceedings of the 17th Annual Conference on Computational Learning Theory*, 93-108.

[6] Dekel, O., Keshet, J. and Singer, Y. (2004) An efficient online algorithm for hierarchical phoneme classification. *Proceedings of the 1st International Workshop on Machine Learning for Multimodal Interaction*, 146-158.

[7] Gu, C. (2000). Multidimension smoothing with splines. *Smoothing and Regression: Approaches, Computation and Application*, edited by M.G. Schimek.

[8] Hofmann, T., Cai, L. and Ciaramita, M. (2003). Learning with taxonomies: classifying documents and words. *NIPS 2003*.

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

[10] Jaakkola, T., Diekhans, M. and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologizes. *In Proc. the Seventh Int. Conf. on Intelligent Systems for Molecular Biology*, 149-158.

[11] Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In Nedellec, C., and Rouveirol, C., editors, Proc. of the 10th European Conf. on Machine Learning (ECML1998), **1398**, 117-142. Springer Velag.

[12] Lewis, D. (1998). Naiver (Bayes) at forty: The independence assumption in information retrieval. In *Proc. of the 10th European Conf. on Machine Learning (ECML1998)*, 4-15.

[13] Liu, S., Shen, X. and Wong, W. (2005). Computational development of ψ-learning. *Proc. SIAM 2005 Int. Data Mining Conf.*, 1-12.

[14] Liu, Y. and Shen, X. (2006) On multicategory ψ-learning and support vector machine. *J. Amer. Statist. Assoc.*, **101**, 500-509.

[15] Mewes HW, Frishman D, G'ldener U, Mannhaupt G, Mayer K, Mokrejs M, Morgenstern B, M'nsterkoetter M, Rudd S, and Weil B (2002) MIPS: a database for genomes and protein sequences. *Nuclerc Acids Res* **30** 31-34.

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

[17] Shahbaba, B. and Neal, R. (2007). Improving classification when a class hierarchy is available using a hierarchy-based prior. *Bayesian Analysis*, **2**, 221-238.

[18] Shen, X., Tseng, G., Zhang, X. and Wong, W. (2003). On ψ-Learning. *J. Amer. Statist. Assoc.*, **98**, 724-734.

[19] Shen,X. and Wang, L. (2007). Generalization error for multi-calss margin classification. *Electronic J. of Statist.*, **1**, 307-330.

[20] Tsochantaridis, I., Hofmann, T., Joachims, T. and Altun, Y. (2004) Support vector machine learning for interdependent and structured output spaces. *Proceedings of the 21st International Conference on Machine Leaning*, Banff, Canada

[21] Vapnik, V. (1998) *Statistical Learning Theory*, Wiley, New York.

[22] Yang, Y. and Liu, X. (1999). A reexamination of text categorization methods. In Proc. of the 22nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retieval, 42-49.

[23] Zhu, J. and Hastie, T. (2005) Kernel logistic regression and the import vector machine. *J. Comput. and Graph. Statist.*, **14**, 185-205.