2011

Large Margin Random Forests On Mixed Type Data

Sheng Liu
University of Mississippi

Large Margin Random Forests on Mixed Type Data

by

Sheng Liu

A Thesis Submitted in Partial Fulfillment of the Requirements

of the Degree of

Master of Science

at the

University of Mississippi

August 2011

Abstract

Incorporating various sources of biological information is important for biological discovery. For example, genes have a multi-view representation. They can be represented by features such as sequence length and physical-chemical properties. They can also be represented by pairwise similarities, gene expression levels, and phylogenetics position. Hence, the types vary from numerical features to categorical features. An efficient way of learning from observations with a multi-view representation of mixed type of data is thus important.

We propose a large margin random forests classification approach based on random forests proximity. Random forests accommodate mixed data types naturally. Large margin classifiers are obtained from the random forests proximity kernel or its derivative kernels. We test the approach on four biological datasets. The performance is promising compared with other state of the art methods including support vector machines (SVMs) and Random Forests classifiers. It demonstrates high potential in the discovery of functional roles of genes and proteins. We also examine the effects of mixed type of data on the algorithms used.

Acknowledgements

I would like to thank my advisors, Dr. Yixin Chen and Dr. Dawn E. Wilkins, for guidance and patience throughout my graduate study at The University of Mississippi. I would also like to thank my committee member, Dr. H. Conrad Cunningham for their assistance and advice.

I would like to thank my wife Xiaona Chu, and my daughter Yichen Liu for their encouragement and supporting me to pursue a degree in Master of Science.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the advancement in high-throughput technologies applied in biology and biomedicine, the accumulation of biological data provides opportunities and challenges to biological prediction [Bushel et al., 2007]. In many biologically motivated prediction problems, such as gene structure and function prediction, gene network prediction, and protein-protein interaction prediction [Bork et al., 1998; Zhao et al., 2008; Myers and Troyanskaya, 2007; Lee et al., 2008; Pandey et al., 2009], various data collection methods generate different types of data, e.g., DNA sequence, protein sequence, phylogenetics profile, microarray data, gene regulatory network, and protein-protein interaction networks. Some of the features are discrete, for instance, sequences, while others are continuous, for example, gene expression levels. This poses challenges in dealing with mixed type features for classification or other analysis in order to provide insights on the underlying biological problem. An algorithm handling mixed type of data for integrated biological prediction is therefore desirable. We proposed to combine random forests and support vector machines to deal with this Liu et al. [2010].

1.1 Relevant Work

Research on mixed type data is active in a closely related area, clustering. Wang et al. [2007] proposed a heritable clustering method that can be used with multiple types of data. Ng et al. [2007] used different metrics for categorical and for numerical features in a feature vector. They then combined a supervised learning approach (multivariate regression) for numerical variables with an unsupervised learning algorithm (k-mode clustering) for categorical variables to cluster data.

There is an abundance of prior work in supervised learning with mixed type data. Réme et al. [2008] proposed a statistical data reduction approach to convert categorical features into numerical features. de Tayrac et al. [2009] applied a multiple factor analysis approach to a problem of dynamic nature: supplementary groups of categorical or numerical variables are often added on the fly.

Support vector machine (SVM) [Vapnik, 1998] and random forests (RF) [Breiman, 2001] are two of the most popular classification techniques that have also been explored for mixed type data. SVM typically requires normalized numerical features to generate good performance. In many cases, categorical features are directly converted into numerical values. A more stable approach is to encode a categorical value into a binary feature vector [Hsu et al., 2000; Agresti, 2002]. RF is an ensemble learning method using decision trees. One advantage of decision tree based methods is that they can work with both categorical and numerical features. Lee and Kim [2010] proposed to convert mixed type data to purely numerical data based on the theory of learning Bayesian Network Clas-

sifiers. Hamby and Hirst [2008] used random forests directly on mixed type of features that encode glycosylation sites. Jiang et al. [2006] used mixed type of features they selected in searching for disease mutations. In their study, random forests delivered slightly better performance than that of SVM.

1.2 An Overview of the Thesis

As a large margin classifier, SVM in practice produces good performance on numerical data, but cannot handle mixed type data directly. Random forests can naturally handle mixed type data via decision tree learning, yet it is not a large margin classifier. In this thesis, we investigate the effects of data types on the performance of SVM and RF. We propose a method that combines large margin learning with random forests to improve the generalization performance of RF on mixed type data.

The remainder of the thesis is organized as follows. Chapter 2 presents a brief review of SVM and RF. In Chapter 3, we introduce a positive definite kernel based on RF proximity. This connects RF with large margin learning. Chapter 4 describes the extensive experimental studies performed and presents the results. We conclude in Chapter 5.

Chapter 2

VC Theory, Support Vector Machines, and Random Forests

This chapter presents the basic concepts of the VC theory, SVMs, and RFs. For gentle tutorials of VC theory, SVMs, and RFs, we refer interested readers to Burges [1998], Müller et al. [2001], Breiman [1996], and Ho [1998]. More exhaustive treatments can be found in Vapnik [1995, 1998] and Breiman [2001].

2.1 VC Theory

Consider a two-class classification problem of assigning class label $y \in \{+1, -1\}$ to input feature vector $\mathbf{x} \in \mathbf{R}^n$. We are given a set of training samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_\ell, y_\ell)\} \subset \mathbf{R}^n \times \{+1, -1\}$ that are drawn independently from some unknown cumulative probability distribution $P(\mathbf{x}, y)$. The learning task is formulated as finding a machine (a function $f : \mathbf{R}^n \to \{+1, -1\}$) that "best" approximates the mapping generating the training set. In order to make learning feasible, we need to specify a function space, $\mathbf{H}$, from which a machine is chosen.

An ideal measure of generalization performance for a selected machine f

is expected risk (or the probability of misclassification) defined as

$$R_{P(\mathbf{x},y)}(f) = \int_{\mathbf{R}^n \times \{+1,-1\}} \mathrm{I}_{\{f(\mathbf{x}) \neq y\}}(\mathbf{x}, y) dP(\mathbf{x}, y)$$

where $\mathrm{I}_A(z)$ is an indicator function such that $\mathrm{I}_A(z) = 1$ for all $z \in A$, and $\mathrm{I}_A(z) = 0$ for all $z \notin A$. Unfortunately, this is more an elegant way of writing the error probability than practical usefulness because $P(\mathbf{x}, y)$ is usually unknown. However, there is a family of bounds on the expected risk, which demonstrates fundamental principles of building machines with good generalization. Here we present one result from the VC theory due to Vapnik and Chervonenkis [Vapnik and Chervonenkis, 1971]: given a set of l training samples and function space $\mathbf{H}$, with probability $1 - \eta$, for any $f \in \mathbf{H}$, the expected risk is bounded from above by

$$R_{P(\mathbf{x},y)}(f) \leq R_{emp}(f) + \sqrt{\frac{h(1 + \ln \frac{2\ell}{h}) - \ln \frac{\eta}{4}}{\ell}} \tag{2.1}$$

for any distribution $P(\mathbf{x}, y)$ on $\mathbf{R}^n \times \{+1, -1\}$. Here $R_{emp}(f)$ is called the empirical risk (or training error), h is a non-negative integer called the Vapnik Chervonenkis (VC) dimension. The VC dimension is a measure of the capacity of a $\{+1, -1\}$-valued function space. Given a training set of size ℓ, (2.1) demonstrates a strategy to control expected risk by controlling two quantities: the empirical risk and the VC dimension. Next we will discuss an application of this idea: the SVM learning strategy.

2.2 Support Vector Machines

Let $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_\ell, y_\ell)\} \subset \mathbf{R}^n \times \{+1, -1\}$ be a training set. The SVM learning approach attempts to find a canonical hyperplane (A hyperplane

$\{\mathbf{x} \in \mathbf{R}^n : \langle \mathbf{w}, \mathbf{x} \rangle + b = 0,\ \mathbf{w} \in \mathbf{R}^n,\ b \in \mathbf{R}\}$

is called canonical for a given training set if and only if $\mathbf{w}$ and b satisfy

$\min_{i=1,\cdots,\ell} |\langle \mathbf{w}, \mathbf{x_i} \rangle + b| = 1)$

$$\{\mathbf{x} \in \mathbf{R}^n : \langle \mathbf{w}, \mathbf{x} \rangle + b = 0,\ \mathbf{w} \in \mathbf{R}^n,\ b \in \mathbf{R}\}$$

that maximally separates two classes of training samples. Here $\langle \cdot, \cdot \rangle$ is an inner product in $\mathbf{R}^n$. The corresponding decision function (or classifier) $f : \mathbf{R}^n \to \{+1, -1\}$ is then given by

$$f(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \ .$$

Considering that the training set may not be linearly separable, the optimal decision function is found by solving the following quadratic program:

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{w}, \xi) = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \cdots, \ell \end{aligned} \tag{2.2}$$

where $\xi = [\xi_1, \cdots, \xi_\ell]^T$ are slack variables introduced to allow for the possibility

of misclassification of training samples, $C > 0$ is some constant.

How does minimizing (2.2) relate to our ultimate goal of optimizing the generalization? To answer this question, we need to introduce a theorem about the VC dimension of canonical hyperplanes [Vapnik, 1995], which is stated as follows. For a given set of ℓ training samples, let R be the radius of the smallest ball containing all ℓ training samples, and $\Lambda \subset \mathbf{R}^n \times \mathbf{R}$ be the set of coefficients of canonical hyperplanes defined on the training set. The VC dimension h of the function space $\mathbf{H} = \{f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) : (\mathbf{w}, b) \in \Lambda, \|\mathbf{w}\| \leq A, \mathbf{x} \in \mathbf{R}^n\}$ is bounded above by $h \leq \min(R^2 A^2, n) + 1$. Thus minimizing the $\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle$ term in (2.2) amounts to minimizing the VC dimension of $\mathbf{H}$, therefore the second term of the bound (2.1). On the other hand, $\sum_{i=1}^{\ell} \xi_i$ is an upper bound on the number of misclassifications on the training set (A training feature vector $\mathbf{x}_i$ is misclassified if and only if $1 - \xi_i < 0$ or equivalently $\xi_i > 1$. Let t be the number of misclassifications on the training set. We have $t \leq \sum_{i=1}^{\ell} \xi_i$ since $\xi_i \geq 0$ for all i and $\xi_i > 1$ for misclassifications), thus controls the empirical risk term in (2.1). For an adequate positive constant C, minimizing (2.2) can indeed decrease the upper bound on the expected risk.

Applying the Karush-Kuhn-Tucker complementarity conditions, one can show that a $\mathbf{w}$, which minimizes (2.2), can be written as $\mathbf{w} = \sum_{i=1}^{\ell} y_i \alpha_i \mathbf{x}_i$. This is called the dual representation of $\mathbf{w}$. An $\mathbf{x}_j$ with nonzero α_j is called a support vector. Let $\mathcal{S}$ be the index set of support vectors, then the optimal decision

function becomes

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i\in\mathcal{S}} y_i\alpha_i \langle \mathbf{x}, \mathbf{x}_i\rangle + b\right) \tag{2.3}$$

where the coefficients α_i can be found by solving the dual problem of (2.2):

$$\begin{aligned} \text{maximize} \quad & W(\alpha) = \sum_{\mathbf{i=1}}^{\ell} \alpha_{\mathbf{i}} - \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i,j=1}}^{\ell} \alpha_{\mathbf{i}}\alpha_{\mathbf{j}}\mathbf{y_i}\mathbf{y_j} \langle \mathbf{x_i}, \mathbf{x_j}\rangle \\ \text{subject to} \quad & C \geq \alpha_i \geq 0,\ i = 1, \cdots, \ell, \text{ and } \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{aligned} \tag{2.4}$$

The decision boundary given by (2.3) is a hyperplane in $\mathbf{R}^n$. More complex decision surfaces can be generated by employing a nonlinear mapping $\Phi : \mathbf{R}^n \to \mathbf{F}$ to map the data into a new feature space $\mathbf{F}$ (usually has dimension higher than n), and finding the maximal separating hyperplane in $\mathbf{F}$. Note that in (2.4) $\mathbf{x}_i$ never appears isolated but always in the form of inner product $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$. This implies that there is no need to evaluate the nonlinear mapping Φ as long as we know the inner product in $\mathbf{F}$ for any given $\mathbf{x}, \mathbf{z} \in \mathbf{R}^n$. So for computational purposes, instead of defining $\Phi : \mathbf{R}^n \to \mathbf{F}$ explicitly, a function $K : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ is introduced to directly define an inner product in $\mathbf{F}$. Such a function K is also called the Mercer kernel [Cristianini and Shawe-Taylor, 2000; Vapnik, 1995, 1998]. Substituting $K(\mathbf{x}_i, \mathbf{x}_j)$ for $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$ in (2.4) produces a new optimization problem

$$\begin{aligned} \text{maximize} \quad & W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & C \geq \alpha_i \geq 0,\ i = 1, \cdots, \ell, \\ & \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{aligned} \tag{2.5}$$

Solving (2.5) for α gives a decision function of the form

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i \in \mathcal{S}} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \quad , \tag{2.6}$$

whose decision boundary is a hyperplane in $\mathbf{F}$, and translates to nonlinear boundaries in the original space. Several techniques of solving quadratic programming problems arising in SVM algorithms are described in Joachims [1999]; Kaufman [1999]; Platt [1999]. Details of calculating b can be found in Chang and Lin [2001].

2.3 Random Forests

Random forests is an ensemble learning method using decision trees. Decision tree learning creates a tree model that predicts target values. It is performed by recursively splitting data into subsets using one of the variables. Gini impurity [Breiman et al., 1984] and information gain [Quinlan, 1993] are two commonly used data splitting criteria. The decision tree learning algorithm requires little data pre-processing. It can handle mixed type data. The resulting tree classifier is equivalent to a set of decision rules, which is easy to interpret. However, a decision tree is prone to overfitting, especially when data is noisy.

The introduction of an ensemble of decision trees aims at combining decisions from diverse decision tree learners to obtain a better predictive performance than that of individual decision trees. It has more representative power than individual decision trees, but is more prone to overfitting.

Two approaches were proposed to overcome this limitation: bootstrap ag-

gregating and random feature subset selection. Breiman [1996] introduced bootstrap aggregating (Bagging), training each model (tree) in the ensemble (forest) using a randomly selected subset of the training set. Given a training set $L = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_\ell, y_\ell)\} \subset \mathbf{R}^n \times \{+1, -1\}$, the bootstrap aggregating process generates t sets $L_1, \cdots, L_t$ each being a bootstrap from L. A classifier is then built for each bootstrap. All t classifiers vote for the final prediction. Bootstrap aggregating improves the accuracy and helps to reduce variance among models and avoid overfitting. Ho [1998] proposed to combine multiple trees constructed from a random subset of features. It maintains highest accuracy on training data and improves the generalization performance.

RF combines the idea of bagging and random selection of a subset of features. First, the training data are bootstrap sampled. Each boostrap is then used to build a tree. In the tree learning process, a small number (m) of input features are randomly selected out of the entire features in each node split. The prediction of the RF is the majority vote of the trees in the forest.

Breiman [2001] interpreted that the generalization error bound are controlled by strength of individual classifiers and dependence between individual classifiers. Strong individual classifiers at the same time independent classifiers gives better prediction performance.

In RF, the choice of m influences the performance. On one hand, a small m tends to produce independent trees, which is desirable in avoiding overfitting. But it may also destroy the dependency structure of the whole set of input features that is useful for the prediction. On the other hand, a large m tends to preserve

feature dependency. But it may result in trees that are highly dependent, hence overfitting. In practice, the optimal value of m in random forests is tuned with a small number of trees.

In next chapter, we try to make a connection between RF and large margin classifier.

Chapter 3

Large Margin Random Forests

Statnikov et al. [2008] compared the performance of SVM and Random Forests on a collection of microarray datasets. In their experiments, where the data are numerical, SVM outperforms random forests, sometimes even significantly. In this chapter, we attempt to combine the ability of RFs to handle mixed type data with the high performance of SVMs. In particular, a positive definite kernel is first derived from a RF. Large margin learning is then performed using the kernel.

We start with the concept of RF proximity. Given two sample points and a decision tree, the decision tree places the two sample points in either the same leaf node or two different leaf nodes. In the former, we view the proximity between two samples as high, and the latter as low. For a RF, the proximity of two samples is summed through all the trees, and normalized by the total number of trees to get the final proximity measure.

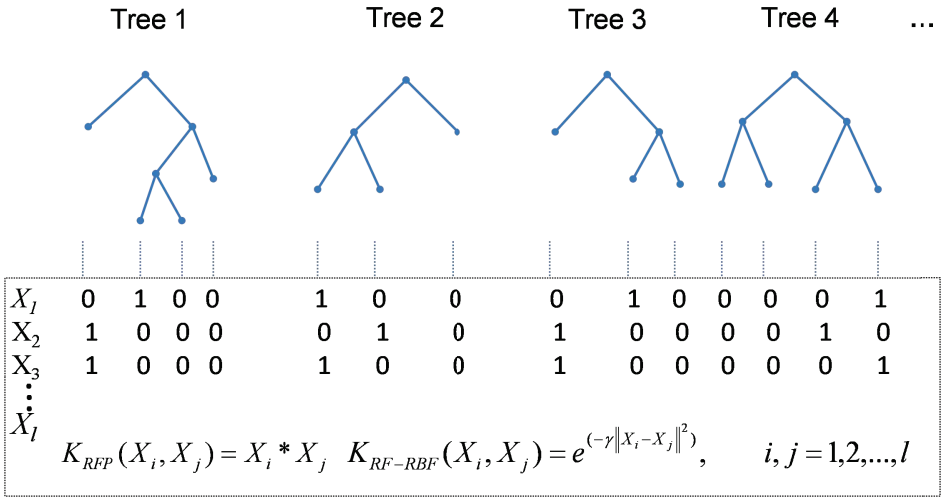
Given a set of observations, their proximities defined by a RF can be organized at a matrix, P, where

$$P_{ij} = prox(\mathbf{x}_i, \mathbf{x}_j) \ .$$

Though not obvious, the proximity matrix P is positive semi-definite. A proof is sketched in Figure 3.1. Because each sample is placed to one and only one leaf node of each decision tree, we define a binary feature vector to capture this structure. For sample $\mathbf{x}_i$, the corresponding binary vector that encodes the leaf nodes assignment is defined as $\mathbf{X}_i = [X_1, \ldots, X_p]^T$ where p is the total number of leaf nodes in a RF,

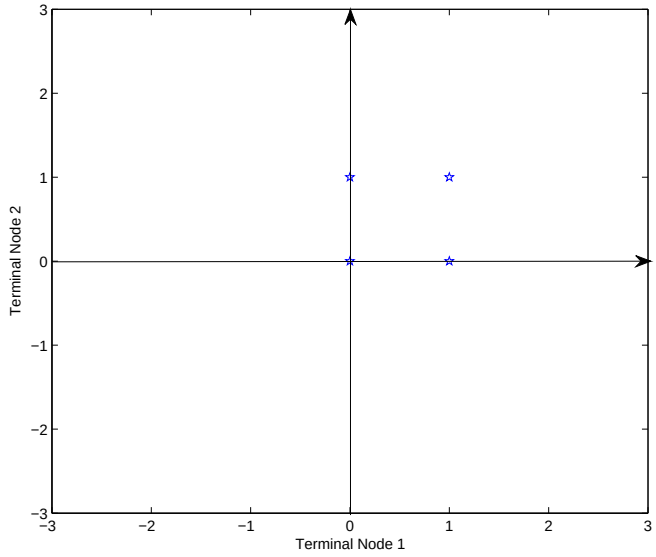
$$X_j = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ falls on the } j\text{-th leaf node,} \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3.1: *Random Forest Proximity Kernels. The terminal nodes of the RF are coded as a binary vector where* 1 *represents the presence of a sample in the leaf node,* 0 *otherwise.*



We call the space of $\mathbf{X}_i$'s a leaf node space. In this space, each sample is mapped to a vertex of a hypercube. Figure 3.2 shows a two terminal nodes example. It is straightforward to derive that the proximity of two observations $\mathbf{x}_i$

Figure 3.2: *Leaf node space representation in 2D. With two leaf nodes, each sample is mapped to a corner of a square.*



and $\mathbf{x}_j$ is defined in the leaf node space as

$$prox(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{T}\mathbf{X}_i^T\mathbf{X}_j \tag{3.1}$$

where T is the number of trees in the RF. Therefore the proximity measure is a kernel. P is positive semi-definite.

In the original feature space, a RF is clearly a nonlinear classifier. In the leaf nodes space, however, a RF is a linear classifier. This is illustrated as follows. Let K be the number of classes. We assign a label to each leaf node. A leaf node has label k, $k \in \{1, \ldots, K\}$, if the leaf node assigns a sample to class k. For each class k, we encode the labels of all leaf node by a vector $\mathbf{w}_k \in \{0, 1\}^p$ where 1's indicate the leaf nodes with label k and 0's otherwise. Therefore, we can define a

linear discriminant function for each class as

$$f_k(\mathbf{X}) = \mathbf{w}^T\mathbf{X} \ .$$

It is not difficult to see that for any observation $\mathbf{x}$, $f_k(\mathbf{X})$ is the number of votes that $\mathbf{x}$ receives for class k. The prediction of a RF is identical to the class that has the maximal discriminant value.

The weight vectors $\mathbf{w}_k$'s obtained by a RF is by no means optimal. This suggests that one may construct an "optimal" classifier in the leaf nodes space. One natural approach is large margin learning using the proximity kernel. RF proximity kernel defines an inner product in the leaf node space. In this thesis, we call a RF proximity kernel RFP. Other kernels can be constructed based on RFP. In this thesis, we consider RFP and radial basis function (RBF) kernel (RF-RBF) for leaf nodes representation of data and formulate RF as a large margin learning problem to optimize the weights $\mathbf{w}_k$'s.

Chapter 4

Results and Discussions

In this chapter, we first describe the four datasets used. We then present the experiments performed and the results obtained.

Table 4.1: *Datasets Used*

Datasets	num. of samples	num. of classes	num. of categorical features	num. of numerical features	Classes are highly unbalanced?[a]
Prospectr	3586	2	2	51	no
class size	(1793, 1793)				
Heart Disease	303	2	8	5	no
class size	(164,139)				
SPECT Heart	267	2	22	0	yes
class size	(212,55)				
Golub	72	3	1	7129	yes
class size	(38,9,25)				

[a]If number of samples in one class is less than 50% of the class with the highest number of samples, we considered it as highly unbalanced.

4.1 Datasets

A summary of the datasets used is shown in Table 4.1. The first dataset is prospectr [Adie et al., 2005] dataset. There are many human hereditary diseases found so far to be caused by mutations in a single gene [O'Connor and Crystal,

2006] or in several genes [Gibson, 2009]. Many of them are important to human well being. Thus it is crucial to identify genes involved these diseases. The higher the correct number of genes identified, the better chance it will help to find ways to cure the diseases. Adie et al. [2005] collected and tested a number of gene-based features that differ between disease related genes and non-disease related genes by using alternate decision tree. These features are integrated from different biological domains and were used further for candidate disease gene prioritization in Adie et al. [2005]. Here we only use it in a classification scenario and evaluate methods with classification performance measure. The goal is to predict disease or health from the various selected features. We use the training (OMIM training set) and test set (HGMD test set) as described in Adie et al. [2005]. There are 3586 samples each with 61 features. Most of them are numerical features. We further removed features with missing data, resulting in 53 features.

Another dataset is Golub dataset [Golub et al., 1999]. There are 72 samples with expression of 7129 probe sets. We add one feature "gender" from clinical data that has least amount of missing information. This forms a vector of length 7130 for each sample. The aim of using this dataset is to predict leukemia (Acute Lymphoblastic Leukemia and Acute Myeloid Leukemia) from this integrated data.

The next two datasets are chosen mainly for evaluation of these methods on different type of data. The third dataset is Heart disease dataset [Detrano et al., 1989]. The purpose is to predict whether a patient presents heart disease or not from features obtained from experiments. It contains a mixture of categorical and numerical features. Eight of thirteen features are categorical. The remaining

features are numerical. To compare the effect of mixed feature type, we use different combinations of categorical and numerical features.

The last dataset we tested is SPECT heart dataset [Kurgan et al., 2001]. This dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. We need to determine whether a patient is normal or abnormal from these images. From these images, 44 continuous features were created for each patient. The features were further processed to obtain final 22 binary features. The number of samples in each class is not balanced.

We test these datasets with RF, SVM, and large margin RF with RFP, RF-RBF kernels. For the heart dataset, different data types are chosen and tested on these algorithms. We randomly split each dataset for twenty repeats. In each split, one set is used for training, the other set is used as the test set. The performance measures are based on an average of all the twenty runs. Details are given below.

4.2 Comparing RF, SVM, and Large Margin RF

The classification results on the prospectr dataset are given in Table 4.2. The parameter C for SVM were selected from 0.8, 1, 2, and 10. The parameter γ (Figure 3.1) for SVM RBF kernel and RF-RBF kernel were chosen from 1, 200, 500, 800 divided by the dimension of data. These two parameters were selected by 5-fold cross validation on the training set. Parameter m for Random Forests was chosen according to the highest accuracy by running a small number of trees with different choice of m's ranging from 1 to the dimension of the data. All classifiers

were implemented in R (2.10-1) [R Development Core Team, 2009] with packages randomForest (4.5-34), $e1071$ (1.5-23) and custom auxiliary functions. Each row of confusion matrix in the Tables represents the true classes.

Table 4.2: *Average confusion matrix and accuracies (± standard deviations) of different prediction methods on prospectr dataset. RFP denotes large margin RF with proximity kernel. RF-RBF stands for large margin RF with RBF kernel defined from proximity.*

	SVM		RF		RFP		RF-RBF	
AUC	0.75		0.76		0.78		0.78	
Accuracy (%)	68.60±0.85		68.85±0.92		69.07±0.73		69.22±0.73	
	Health	Sick	Health	Sick	Health	Sick	Health	Sick
Health	0.71	0.29	0.73	0.27	0.71	0.29	0.71	0.29
Sick	0.34	0.66	0.34	0.66	0.33	0.67	0.32	0.68

The prospectr data were normalized before application of SVM training. We use the built in function *scale* in R package $e1071$ for this purpose. Without normalization, the performance of SVM on this dataset was only 50% in accuracy (data not shown).

On average, RF-RBF kernel gives the highest accuracy, followed by RFP kernel. Comparing to SVM and RF, RF-RBF and RFP give more balanced per class prediction. The results are given in Table 4.2.

The Golub data has multiple classes, we calculate relative classifier information (RCI) [Sindhwani et al., 2001] in addition to accuracy. RFP is significantly better than RF from Table 4.3 with P value 0.02.

Table 4.4 shows the results on heart disease dataset. For the SVM classifier, the categorical features were converted to consecutive integer values. RF and RF-RBF achieved the best performance on this dataset.

Table 4.3: *Average confusion matrix,accuracies (± standard deviation) and relative classifier information (RCI) of RF, RFP on Golub dataset with number coding. RFP denotes large margin RF with proximity kernel.*

	RF			RFP		
RCI	0.62±0.08			0.67±0.15		
Accuracy (%)	89.86±2.85			90.86±5.21		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Class 1	0.99	0.00	0.01	0.97	0.01	0.02
Class 2	0.56	0.29	0.15	0.35	0.46	0.19
Class 3	0.04	0.00	0.96	0.03	0.00	0.97

Table 4.4: *Average confusion matrix and accuracies (± standard deviation) of RF, SVM, RFP, and RF-RBF on heart disease dataset with number coding. RFP denotes large margin RF with proximity kernel. RF-RBF stands for large margin RF with RBF kernel defined from proximity.*

	SVM		RF		RFP		RF-RBF	
AUC	0.89		0.87		0.87		0.89	
Accuracy (%)	80.01±2.03		80.68±2.42		78.94±3.51		80.61±2.86	
	Health	Sick	Health	Sick	Health	Sick	Health	Sick
Health	0.80	0.20	0.80	0.20	0.78	0.22	0.80	0.20
Sick	0.21	0.79	0.18	0.82	0.20	0.80	0.18	0.82

The SPECT heart dataset is unbalanced. We use the Area Under Receiver Operating Characteristic Curve (AUC) as well as the confusion matrix as performance metrics. For AUC, a value closer to 1 indicates better performance. The results are given in Table 4.5. RF has the highest AUC, while its average accuracy is not the highest. The variance of AUC is large across all the methods. The average accuracies of RFP and RF-RBF are also more balanced.

The above results demonstrate the competitive performance of large margin RF against RF and SVM. Large margin RF hence presents another choice when selecting suitable algorithms for integrated biological prediction.

Table 4.5: *Average confusion matrix and AUC of different prediction methods on SPECT heart dataset with number coding. RFP denotes large margin RF with proximity kernel. RF-RBF stands for large margin RF with RBF kernel defined from proximity.*

True classes	SVM		RF		RFP		RF-RBF	
AUC	0.70		0.80		0.74		0.73	
Accuracy (%)	81.50±2.77		80.87±2.45		80.82±2.26		80.41±2.93	
	Health	Sick	Health	Sick	Health	Sick	Health	Sick
Health	0.63	0.37	0.56	0.44	0.58	0.42	0.62	0.38
Sick	0.17	0.83	0.15	0.85	0.16	0.84	0.18	0.82

4.3 Effects of Binary Encoding of Categorical Features

In the previous experiment on heart dataset, the categorical features were converted to consecutive integers. Another popular encoding method in the literature is to convert each value of a categorical variable to a binary string. For example, for a variable with four possible values A, T, C, and G, the binary encoding method converts the four values into 0001, 0010, 0100, and 1000. The results are shown in Table 4.6. As we can see that the performance of SVM decreases significantly compared with Table 4.4. For the other approaches, the performance remains roughly the same. This suggests that SVM is sensitive to the encoding method when handling categorical features. On the contrary, RF and large margin RF are not sensitive to the encoding method.

Table 4.6: *Confusion matrix and accuracies of different prediction methods on heart disease dataset with binary expansion coding. RFP denotes large margin RF with proximity kernel. RF-RBF stands for large margin RF with RBF kernel defined from proximity.*

True classes	SVM		RF		RFP		RF-RBF	
AUC	0.89		0.87		0.86		0.89	
Accuracy (%)	77.83±6.17		80.71±2.19		78.47±2.02		80.79±2.11	
	Health	Sick	Health	Sick	Health	Sick	Health	Sick
Health	0.79	0.21	0.82	0.18	0.81	0.19	0.83	0.17
Sick	0.24	0.76	0.20	0.80	0.24	0.76	0.22	0.78

4.4 Comparing SVM and Random Forests Based Methods on Mixed Type Data

To further compare SVM and RF based methods on mixed type data, we performed a series of experiments on the heart disease dataset, in which we control the ratio of categorical and numerical features in the dataset. Specifically, all the numerical features were included. A fixed number of categorical features were selected randomly from the eight categorical features. The number of categorical features included varies from 1 to 8.

For each case, the experiment was repeated twenty times. The results are shown as box plots in Figures 4.1-4.4. On each box, the central red line is the median, the edges of the box are the 25th and 75th percentiles, the whiskers span the remaining data that are not considered outliers, and outliers are marked with '+'.

The performance of all four methods increases as the number of features selected increases. The increases of median accuracy from one categorical feature

Figure 4.1: *Performance of Support Vector Machines on Heart Disease Data with Different Number of Categorical Features.*

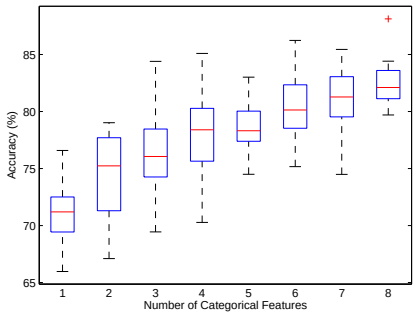


Figure 4.2: *Performance of Random Forests on Heart Disease Data with Different Number of Categorical Features.*

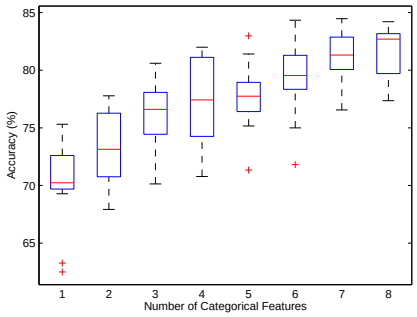


Figure 4.3: *Performance of Random Forests Proximity kernel (RFP) on Heart Disease Data with Different Number of Categorical Features.*

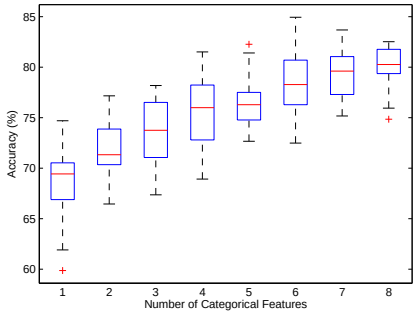
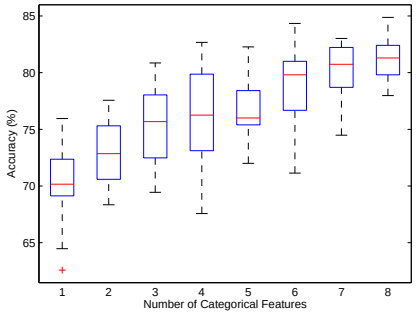


Figure 4.4: *Performance of RF on radial basis function kernel (RF-RBF) on Heart Disease Data with Different Number of Categorical Features.*



to eight categorical features are 0.1091, 0.1246, 0.1083, and 0.1113 for SVM, RF, RFP, and RF-RBF, respectively. This suggests that RF based approaches can better utilize the information provided by the categorical features than SVM.

Chapter 5

Conclusions

We propose a learning method that combines RFs with large margin learning for biological prediction tasks. We observed that on mixed type data, the performance of a RF based approach is not sensitive to the different encoding of the categorical variable, which is in stark contrast to SVMs. We also tested the performance variation of SVM and RF-based methods when the proportion of categorical features changes. Our results show that the performance of both SVM and RF improves as the number of categorical features increases. However, the amount of improvement of RF-based approaches tends to be higher. The proposed large margin RF demonstrates competitive performance in comparison with RFs and SVMs. In terms of the confusion matrix, RFP and RF-RBF generate more balanced per class accuracy.

BIBLIOGRAPHY

E. A. Adie, R. R. Adams, K. L. Evans, D. J. Porteous, and B. S. Pickard. Speeding disease gene discovery by sequence based candidate prioritization. *BMC Bioinformatics*, 6:55, 2005.

A. Agresti. *Categorical Data Analysis (Wiley Series in Probability and Statistics)*. Wiley Interscience, Hoboken, NJ, 2 edition, July 2002. ISBN 0471360937.

P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan. Predicting function: from genes to genomes and back. *Journal of Molecular Biology*, 283(4):707 – 725, 1998. ISSN 0022-2836.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. ISSN 0885-6125.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

P. R. Bushel, R. D. Wolfinger, and G. Gibson. Simultaneous clustering of gene expression data with clinical chemistry and pathological evaluations reveals phenotypic prototypes. *BMC Syst Biol*, 1:15, 2007.

C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 1 edition, March 2000. ISBN 0521780195.

M. de Tayrac, S. Lê, M. Aubry, J. Mosser, and F. Husson. Simultaneous analysis of distinct omics data sets with integration of biological knowledge: Multiple factor analysis approach. *BMC Genomics*, 10:32, 2009.

R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304 – 310, 1989. ISSN 0002-9149.

G. Gibson. Decanalization and the origin of complex disease. *Nat Rev Genet*, 10 (2):134–140, Feb 2009.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, Oct 1999.

S. E. Hamby and J. D. Hirst. Prediction of glycosylation sites using random forests. *BMC Bioinformatics*, 9:500, 2008.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998. ISSN 0162-8828.

C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification, 2000.

R. Jiang, H. Yang, F. Sun, and T. Chen. Searching for interpretable rules for disease mutations: a simulated annealing bump hunting strategy. *BMC Bioinformatics*, 7:417, 2006.

T. Joachims. Making large-scale support vector machine learning practical. pages 169–184, 1999.

L. Kaufman. Solving the quadratic programming problem arising in support vector classification. pages 147–167, 1999.

L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artif Intell Med*, 23(2):149–169, Oct 2001.

N. Lee and J.-M. Kim. Conversion of categorical variables into numerical variables via bayesian network classifiers for binary classifications. *Computational Statistics & Data Analysis*, 54(5):1247 – 1265, 2010. ISSN 0167-9473.

S.-A. Lee, C.-H. Chan, C.-H. Tsai, J.-M. Lai, F.-S. Wang, C.-Y. Kao, and C.-Y. F. Huang. Ortholog-based protein-protein interaction prediction and its

application to inter-species interactions. *BMC Bioinformatics*, 9 Suppl 12:S11, 2008.

S. Liu, Y. Chen, and D. Wilkins. Large margin classifiers and random forests for integrated biological prediction on mixed type data. In *Proc. of the 7th Annual Biotechnology and Bioinformatics Symposium (BIOT)*, pages 11–19, 2010.

K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

C. L. Myers and O. G. Troyanskaya. Context-sensitive data integration and prediction of biological networks. *Bioinformatics*, 23(17):2322–2330, 2007.

M. K. Ng, E. Y. Chan, M. M. C. So, and W.-K. Ching. A semi-supervised regression model for mixed numerical and categorical variables. *Pattern Recogn.*, 40(6):1745–1752, 2007. ISSN 0031-3203.

T. P. O'Connor and R. G. Crystal. Genetic medicines: treatment strategies for hereditary disorders. *Nat Rev Genet*, 7(4):261–276, Apr 2006.

G. Pandey, C. L. Myers, and V. Kumar. Incorporating functional inter-relationships into protein function prediction algorithms. *BMC Bioinformatics*, 10:142, 2009.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

T. Réme, D. Hose, J. D. Vos, A. Vassal, P.-O. Poulain, V. Pantesco, H. Goldschmidt, and B. Klein. A new method for class prediction based on signed-rank algorithms applied to affymetrix microarray experiments. *BMC Bioinformatics*, 9:16, 2008.

V. Sindhwani, P. Bhattacharya, and S. Rakshit. Information theoretic feature crediting in multiclass support vector machines. In *In Proceedings of the first SIAM International Conference on Data Mining*, 2001.

A. Statnikov, L. Wang, and C. F. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9:319, 2008.

V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0387945598.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. ISBN 0471030031.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative

frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Z. Wang, P. Yan, D. Potter, C. Eng, T. H.-M. Huang, and S. Lin. Heritable clustering and pathway discovery in breast cancer integrating epigenetic and phenotypic data. *BMC Bioinformatics*, 8:38, 2007.

X.-M. Zhao, Y. Wang, L. Chen, and K. Aihara. Gene function prediction using labeled and unlabeled data. *BMC Bioinformatics*, 9:57, 2008.

VITA

Sheng Liu was born in Wuhan, Hubei, China on January 21, 1974. In September 1993, he enrolled in School of Life Sciences, Wuhan University in Wuhan, China. In July 1997, he received his Bachelor of Science in Biochemistry. In the fall of 2009, he enrolled in the University of Mssissippi and will receive his Master of Science in Engineering Science with an emphasis in computer science in August of 2011.