2013

Gene set based ensemble methods for cancer classification

William Evans Duncan
Louisiana State University and Agricultural and Mechanical College, duncan@csc.lsu.edu

GENE SET BASED ENSEMBLE METHODS FOR CANCER CLASSIFICATION

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The School of Electrical Engineering and Computer Science

by
William E. Duncan
B.S., University of the South, 2000
M.S., The University of Tennessee Knoxville, 2003
August 2013

To Glacia

Acknowledgements

My profound thanks goes to Jian Zhang, my major professor, for his suggestions and guidance during this research. My special gratitude also goes to Sitharama S. Iyengar for his encouragement during the early days of my graduate work.

John R. Stevens of Utah State University provided me some helpful tips and R programming resources for exploratory data analysis during the initial stages of my research. Thanks to Michael A. Langston, my graduate advisor at the University of Tennessee, for providing me a summer research opportunity. Fred Croom of the University of the South was instrumental in providing me grants to cover my living and educational expenses as an undergraduate.

My gratitude to my parents, both of whom are deceased, is beyond measure. It is no exaggeration when I say that this work would have been impossible if they had not met!

Finally, I wish to thank the following: Maxwell (for his unwavering belief in my abilities); Watchen (for thinking she is my mother); and everyone that I met along the way whose friendship and love I have come to cherish.

Baton Rouge, Louisiana — William E. Duncan
April 12, 2013

Table of Contents

List of Tables

List of Figures

Abstract

Diagnosis of cancer very often depends on conclusions drawn after both clinical and microscopic examinations of tissues to study the manifestation of the disease in order to place tumors in known categories. One factor which determines the categorization of cancer is the tissue from which the tumor originates. Information gathered from clinical exams may be partial or not completely predictive of a specific category of cancer. Further complicating the problem of categorizing various tumors is that the histological classification of the cancer tissue and description of its course of development may be atypical.

Gene expression data gleaned from micro-array analysis provides tremendous promise for more accurate cancer diagnosis. One hurdle in the classification of tumors based on gene expression data is that the data space is ultra-dimensional with relatively few points; that is, there are a small number of examples with a large number of genes. A second hurdle is expression bias caused by the correlation of genes.

Analysis of subsets of genes, known as gene set analysis, provides a mechanism by which groups of differentially expressed genes can be identified. We propose an ensemble of classifiers whose base classifiers are $\ell 1$-regularized logistic regression models with restriction of the feature space to biologically relevant genes. Some researchers have already explored the use of ensemble classifiers to classify cancer but the effect of the underlying base classifiers in conjunction with biologically-derived gene sets on cancer classification has not been explored.

Chapter 1
Introduction

Cancer is a disease in which cells in specific tissues in the body undergo uncontrolled division. This condition results in the malignant growth or tumor. Cancerous cells very often invade and destroy surrounding healthy tissues and organs. The National Cancer Institute reports that there are currently over 200 different forms of cancers. The potential to prevent, diagnose and treat any of these forms of cancers require insights into how changes in their genome occur.

Cancer diagnosis has been based largely on the morphologic characteristics of biopsy specimens by clinicians. This is understandable since cancer presents a new physical manifestation (phenotype) in a patient. However, the realization that it is usually preceded by a change in a person's genetic makeup (genotype) offers great potential in giving researchers deeper insights into the biology of cancer.

With the Human Genome Project [10] which succeeded in identifying and mapping nearly 25,000 genes of the human genome from both physical and functional perspectives, the potential to revolutionize medicine, cancer research being no exception, has never been more promising.

Today, microarray technology has provided further impetus for cancer research. Microarray expression analysis is a widely used technique for profiling mRNA expression. The mRNA carries genetic information from DNA to the ribosome, where they specify the amino acid sequence of the protein products of gene expression. DNA

segments containing genes of interest undergo polymerase chain reaction (PCR) [16], a method developed by Kary Mullis in the 1980s. Gene expression microarrays are then constructed by transferring cDNA in salt solutions onto chemically modified glass microscope slides using a contact-printing instrument. The target cDNA then binds with just those probes that have complementary base sequences, in a process known as hybridization. The hybridization process allows relative expression levels to be determined based on the ratio with which each probe hybridizes to an individual array element. A laser scanner is then used to measure fluorescence intensities, which allows for the simultaneous determination of the relative expression levels of all the genes represented in the array.

This technology allows scientists to conduct high-throughput experiments to measure the activity of genes; that is, it allows scientists to examine thousands of genes simultaneously to determine which are on and which are off. A microarray permits many hybridization experiments to be performed in parallel. All the data is collected and a profile is generated for gene expression in the cell. Each cell has some combination of genes turned on, and others turned off depending on the cell's function at a given point in time. Gene expression profiling enables scientists to create a global picture of cellular functions.

Cancer diagnosis is an emerging clinical application of gene expression microarray analysis [53]. There is increasing evidence in the literature to support the notion that the clinical behavior of various forms of cancers is linked to underlying gene expression differences that are detectable at the time of diagnosis.

Machine learning research in which gene expression data is used to put cancers into

categories and discover their genetic markers abounds in the literature. The research includes the development of both supervised and unsupervised learning methods. Some broad categories include simple traditional machine learning models, hybrid methods and complex machine learning methods.

Artificial neural networks are an example of a traditional supervised learning model. Artificial neural networks (ANN) have been used to classify cancer from gene expression profiles. [32]. ANNs were used to develop models for the classification of small, round blue-cell tumors (SRBCTs) into four distinct diagnostic classes.

The Nearest Neighbor classification rule is also a simple traditional machine learning algorithm. A hybrid genetic algorithm K-Nearest Neighbor method (GA/KNN) has been used to classify colon and leukemia cancer data sets [36]. The GA/KNN method was found to be effective in identifying subsets of genes with excellent predictive powers. A nearest neighbor-based algorithm uses a more localized method of classification. It requires no initial processing of the training data, and the classification of new samples is based on their nearest neighbors in the training set. There are a wide variety of KNN classifiers since "nearest neighbor" depends on the proximity metric that is used and the way the decision boundary is computed.

There have also been extensive inroads made into breast cancer classification and detection using gene expression profiles. Rank Nearest Neighbor (RNN) classification rules, another Nearest Neighbor-based approach, have been used in classifying breast cancer [1]. Some cancer classification methods have involved the use of clustering [29], a widely used machine learning algorithm. For example, there was a study that used

clustering to classify breast carcinomas based on variations in gene expression patterns that were derived from cDNA microarrays [52].

Bayesian networks, based on probability theory, are widely used to perform classification tasks. There has been an increased interest in Bayesian Networks for gene expression data analysis [59, 64]. Some use of Bayesian Networks have included scored-searching method that has proven to be effective for learning gene networks from DNA microarray data. Bayesian classifiers based on decision trees [42] have also been pursued by researchers.

There has also been the use of complex learning models such as ensemble classifiers [35]. An ensemble classifier consists of a set of individually trained base classifiers whose predictions are pooled to classify new instances [40]. These include bagging [7] which involves sampling with replacement from the set of training data to create new training sets, random subspace technique [3] which selects random subsets of features to train the base classifiers, and random forest technique [6] which combines both random subspaces and bagging and then uses decision trees as the base classifier. Support Vector Machine (SVM) has also being used in cancer classification. Support Vector Machine [11] is a learning algorithm in which input vectors are non-linearly mapped to a very high-dimension feature space. SVMs have been used as base classifiers in ensembles in several application domains [66, 37, 13]. The k nearest neighbor classifier (KNN) has been used as base classifiers in an ensemble [39]. The OET-KNN [50] algorithm, used in protein fold pattern recognition, is an example of a KNN-based ensemble.

The various kinds of classifiers, whether a simple classifier or a composite of classifiers, have different characteristics. The two key issues which drive the design and choice of classifiers used in the classification of cancer from microarray data are cost and accuracy. The fewer genes that can be used in building a discriminative model that distinguishes categories of cancers, the more cost effective cancer classification/diagnosis would be. It goes without saying that the more accurately the model can predict different categories of cancers, the more useful it is. In addition to other goals, with any classification model based on gene expression data, we want to achieve the highest possible classification accuracy in future instances to be classified with as few genes as possible.

The accuracy of an ensemble classifier depends on many factors. The nature of the problem, the choice of base classifiers and their inherent characteristics and predictive powers have an enormous bearing on how well the ensemble performs. My research focused on four key questions:

1. How effective are logistic regression models [33] with $\ell 1$-regularization [5] in the removal of redundant (ineffective) genes in micro-array gene expression data?
2. Does the flexibility (complexity) of the base classifier lead to an improvement in the classification accuracy of the ensemble?
3. Does the use of a base classifier with regularization enhance the accuracy of the ensemble?
4. Can the performance of an ensemble classifier for cancer that uses microarray data be improved by using subspaces based on biologically-derived gene sets?

In the first phase of this work, we propose a two-stage approach for gene selection. Two common gene selection algorithms are used in preprocessing the data: Kruskal-Wallis nonparametric one-way ANOVA[23] and the ratio of between group variance to

within group variance[14]. After this preliminary gene selection stage, $\ell 1$-regularized logistic regression model is applied to the data set for further gene selection.

The second phase of our work consisted of determining the choice of base classifiers to use in an ensemble classifier. We conducted experiments involving the use of both "flexible" (non-linear) and "inflexible" (linear) classifiers to determine which category of classifiers would yield better performance as a base classifier in an ensemble used to classify cancer-related microarray data.

In the final phase of our work, we propose two kinds of ensembles for the classification of various kinds of cancers. One approach uses subspaces that are based on gene sets derived from *a priori* biological information. In the second approach, the ensembles use bootstrap aggregation with the feature space constituted after gene selection.

Our work led to some key findings. One finding was that $\ell 1$-regularized logistic regression models eliminate redundant/ineffective genes from the feature space; that is, $\ell 1$-regularized logistic regression models implicitly perform gene selection. We also found that if there is *a prior* information regarding biologically-derived gene sets, the performance of an ensemble classifier used to classify various kinds of cancers can be enhanced.

Chapter 2
Background

In this chapter, we describe related cancer-classification works. Both their strengths and limitations are discussed.

2.1 Survey of Related Works

Since the discovery of microarray technology, the cancer classification problem has rapidly gained a foothold in data mining research. Many studies have focused on the classification of various kinds of cancers or cancer subtypes. Many kinds of traditional and hybrid learning approaches have been applied to cancer classification. These approaches have included traditional methods such as artificial neural networks (ANN), nearest neighbor-based (NN) methods, Bayesian networks (BN), decision trees and support vector machines (SVM). They have also included ensemble methods. For example, support vector machines and nearest neighbor based ensemble methods have been used.

2.2 ANN Cancer Classification

Khan *et al* developed a method for classifying childhood small, round blue cell tumors (SRBCTs) into four diagnostic categories based on their gene expression signatures using artificial neural networks (ANNs) [32]. The four SRBCT categories used in their work were neuroblastoma (NB), rhabdomyosarcoma (RMS), nonHodgkin lymphoma (NHL) and the Ewing family of tumors (EWS). There were 63 training examples used in this work. The data consisted of gene-expression profiles from cDNA microarrays.

Principal component analysis [31](PCA) was used to reduce the dimensionality of original gene profiles. The 6567 genes in the gene profiles was narrowed to 88 numbers obtained from PCA eigenvector projections. The 10 most dominant PCA components were then used in subsequent training of the ANN model. To evaluate the model 3-fold cross-validation [22] was used. The experiments were repeated 1250 times, randomly shuffling the data set each time. In total there were 3750 linear ANN models. These models were calibrated to differentiate between the four SRBCT categories. Some additional 25 test samples were then classified by pooling the output of the 3750 ANN models using committee vote.

Although results obtained in this study were very good, like any ANN-based classifier, this approach has some limitations. Any ANN learning process is a 'black box'. Once the general architecture of the network is defined and initial seeding of the random number generator is done, the user exerts very little control over what happens during a training epoch. ANN methods tend to be computation-intensive because they very often require many training epochs. They are very susceptible to over-fitting. They also generally do not provide a mathematical model in the form of an equation other than their own internal weighting scheme.

2.3 GA/KNN Cancer Classification

A hybrid k-nearest neighbor(KNN)-genetic algorithm(GA) method has also been used in the cancer classification problem. Lymphoma and colon data sets were used in a study by Li *et al* [36]. In their experiments, each sample, represented by d genes, was placed into one of k=3 categories depending on its Euclidean distance. Each sample was placed into a class only if its three nearest neighbors belong to the same class.

If its three nearest neighbors did not belong to the same class, the sample remained unclassified.

Next a GA method was used to search for a subset of d genes, a so-called *chromosome*, from all the genes in a given data set that could discriminate between the various tumor types. A chromosome is initially randomly selected from the genes in the data set. A set of 100 chromosomes were then selected to form a so-called *niche*. Then 10 niches were made to evolve in parallel. The fitness function used to classify each chromosome was the ability of the KNN procedure to classify the chromosome - 1 for classifiable chromosomes and 0 for unclassifiable chromosomes. A chromosome is chosen as the *best* based on the correlation coefficient ($\mathbb{R}^2$) obtained after cross-validation. Between one and five genes from the best chromosome are randomly selected and mutated and the chromosome is then used in the niche for the next generation. The remaining 99 chromosomes of the niche are determined by sampling the parent generation based on their weighted fitness values.

A solution is deemed to be found when nearly all the training examples are correctly classified in any one of the niches. When this occurs, the high $\mathbb{R}^2$ chromosome is set aside and the process is restarted. After 10000 high $\mathbb{R}^2$ chromosomes have been chosen, the genes are sorted according to the frequency by which they were chosen. The frequently selected genes are then used to classify the test set.

This approach has a few limitations. The larger the choice of d is, the more computation-intensive the algorithm. Further, the algorithm is very sensitive to the number of genes selected. When too few genes are selected results may be unreliable and when too many genes are selected a lot of noise (irrelevant genes) will be added to the data set.

2.4 Bayesian Cancer Classification

Techniques have been developed in the last three decades to learn Bayesian networks [30] from data. A Bayesian network is a graphical model that encodes probabilistic relationships among variables. It consists of two components. It consists of an acyclic graph (DAG) whose nodes represent random variables and a conditional distribution for each variable given its parent node in the acyclic graph.

Friedman *et al* [21] proposed techniques for using Bayesian networks for the analysis of gene expression data. They recognized that expression level data involves thousands of genes with only a few samples and inferring a network from such a data set would inherently lead to a statistically insignificant network. They further observed that because only a few genes affect the transcription of a gene, Bayesian networks could be readily used in the gene expression domain.

Bayesian networks are computationally expensive. The quality of the networks depends too heavily on the quality of prior knowledge. Bayesian networks do not account for the probability of an unanticipated event which very often comes up in modeling complex systems.

2.5 Cancer Classification and Decision Trees

Decision trees [42] are widely used in classifying samples by filtering them through a tree-like graph or model. A simple attribute of the sample is tested at each node and then a branching decision is made. Models that rely only on one decision tree are very susceptible to over-fitting. Decision trees are easily built and understood because of their hierarchical structure. They are also able to model complex functions.

There are numerous variants to the decision tree algorithm which seek to ameliorate the over-fitting phenomenon. There are several heuristics for *pruning* [17] the decision tree to make it less susceptible to over-fitting and improve its performance. Pruning involves minimizing consecutive branches so that the tree is made to generalize more adequately. Pruning usually involves top-down or bottom-up traversal of the decision tree while removing nodes to improve certain criteria. Popularly used pruning strategy include cost-complexity pruning, reduced error pruning, minimum error pruning, minimum descriptive length pruning, minimum message length pruning and critical value pruning [48].

Increased performance and generalization of trees can be obtained by using an ensemble of trees. The boosting [19] and bagging [7] algorithms are popular approaches to constructing a random forest rather than a classifier based on one tree. Random trees and bootstrap samples have been used to perform gene selection and classification on 10 cancer-related data sets [12]. Random forests perform well in the analysis of micro-array data because they are robust even under conditions in which predictor variables contain noise. A major disadvantage in the use of random forests is that when groups of features are correlated, random forests are bias toward the smaller group [57].

2.6 Cancer Classification with SVM Ensembles

Bagged ensembles of SVM linear classiers were used to differentiate normal and malignant tissues [61]. Experiments were conducted using these ensembles with and without gene selection. The ensembles were tested on both colon cancer and leukemia data sets. The colon cancer data sets consisted of expression levels of 2000 genes across the samples. There were 62 samples, 22 of which were normal tissue samples

while 40 were colon cancer tissues. The leukemia data set contained 7129 genes across the samples. It consisted of 72 samples. There were 47 acute lymphoblastic leukemia tissue samples and 25 acute myeloid leukemia samples.

Based on empirical results from experiments, it was demonstrated that bagged SVMs performed better than a single SVM on both data sets. On the colon cancer data set, bagging improved the accuracy of SVMs.

SVMs are very robust since different kinds of kernel functions can be used. However, how to find an optimal kernel function for an SVM is still the subject of active research. Another limitation of the SVM model is that when it is solved, it cannot be easily expressed in parametric form.

Chapter 3
Cancer Classification

In this chapter, we formally define the gene expression-based cancer classification problem. Some of the challenges and issues associated with deriving a solution to the problem are also discussed.

3.1 Definition

Microarray technology provides the mechanism to measure the activities of cells. Using cDNA micro-array technology, scientists are able to compare gene expression profiles in normal and cancerous genes. The beauty of cDNA micro-array technology is that the entire genome can be measured simultaneously to make a comparison of normal and diseased tissues. The focus of any research in the use of microarray data in cancer classification is to answer one basic question: what kind of cancer is affecting a given cell and which genes influence or are predictive of the cancer.

The classification of cancer using microarray data from samples with known labels is a supervised learning problem; that is, it is a problem that requires inferring a discriminative model from labeled samples that can be used to predict the labels of unlabeled samples. A formal definition of cancer classification from micro-array data is given below.

Definition 3.1.1 (The General Gene Expression Cancer Classification Problem)**.** Assume that there are k classes of cancers. Usually one class is normal.
Input: Consider a set of m samples $\{(x^j, y^j)\}, j = 1, 2 \ldots m$ a set of example gene-expression profiles from cancer patients and cancer-free people. x is an n-dimensional vector of gene expression measurements and y is the label (i.e., type of cancers or no cancer)., where $x^j \in \mathbf{R}^n$ and $y^j \in \{1, 2, \ldots, k\}$, is the class that x^j belongs to and n is the number of genes across each sample. The i^{th} component of x^j, x_i^j, is the

measurement of the expression level of the i^{th} gene in the j^{th} sample.
Output: a function $f : \mathbf{R}^n \rightarrow \{1, 2, \ldots, k\}$ that can classify a gene expression profile whose label is unknown.

The matrix $X = \left[x_i^j\right] m, n$ represents the gene expression levels of all genes across all m samples and n genes and the vector $Y = . \langle y^1, y^2, \cdots, y^m \rangle$ are classes of cancers.

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \cdots & x_n^m \end{bmatrix} \qquad Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix}$$

As in any supervised learning classification problem, we seek to obtain the target function, also known as the classification model, f, that maps each sample x to a label y representing one of the predefined k classes (a cancer type or normal). The goal is to minimize the classification error; that is, minimize the cases where $f(x^p)$ differs from y^p, the correct class of x^p.

3.2 Challenges and Issues

There are several challenges associated with cancer classification using gene expression data from microarray analysis. Some of the challenges are technological; that is, they are due to the current state of the cDNA microarray technology. Other challenges are due to the dimensionality of the data and algorithmic challenges inherent in any supervised learning method.

One technological challenge is due to the microarray chip. Studies have shown that there are chip-related artifacts in microarray experiments [2]. Yu *et al* found that the closer the gene pairs on the micro-array chips, the higher the average correlation coefficient between them [68]. They hypothesized that the observed correlation in microarray experiments was due in part to chip artifacts that stemmed from the fact

that the printing tips are not completely cleaned when printing a spot on the chip and carry over some of the DNA samples from the previous spot. They theorized that the signal of any spot has an artificial component related to the previous one, thus producing an artifact. They conducted empirical studies that confirmed their theory and demonstrated that this artifact carries over to several neighboring spots.

Genes that are close on the microtiter plates also tend to be much more highly correlated [68]. While the chip artifacts affect genes in the same block, genes in different blocks may also be neighbors because they are on the same microtiter plate. When Yu *et al* examined the periodicity of the observed correlation, they found that it corresponded to the size of one printing block in each experiment. They concluded that the corresponding spots in different blocks are actually neighbors on the microtiter plates due to the way printing procedure of the microarray works. They empirically demonstrated that both chip and plate artifacts affect microarray experiments to varying degrees depending on the nature of the experiment. The hope is that with further technological advances in microarray technology, these artifacts will be eliminated or become negligible.

There are also issues that arise because of the nature of the data. One issue which often arises in cancer classification is that there are relatively few examples. When the classification function is simple, an "inflexible" learning algorithm is usually able to learn from the data. However, when the function is complex, as is usually the case with polytomous microarray data, the function can be better learned from a relatively large number of training examples using a "flexible" learning algorithm. Even with a simple classification function, if there are relatively few examples, there is the ever-present potential of over-fitting: the phenomenon in which the model performs well

on the training sample but is unable to generalize to future data that it has not encountered.

Finally, there is the issue of the "curse of dimensionality" [4], a phrase attributed to Richard Bellman. Microarray analysis involves ultra high-dimensional spaces - thousands of genes whose expression levels are examined. In an ultra high-dimension space, convergence is very difficult with a small number of *a priori* examples. This is often true when it comes to cancer classification using gene expression levels. Algorithmic techniques are often employed to ameliorate this phenomenon.

In order to address the challenges/issues that have been discussed, our research focused on addressing several problems. One key area of our research was gene selection; that is, selecting genes with high discriminative power that are uncorrelated. A second problem we focused on is the issue of over-fitting both in the classification of cancer and gene selection. Another problem raised was the use of ensemble classifiers to mitigate the problem of over-fitting. Then there is the problem of determining a good base classifiers for an ensemble. Finally, there is the problem of how to construct ensemble classifiers.

In the next three chapters, we discuss three key techniques that can be used to overcome some of the challenges involved in classifying cancer using microarray data:

1. The use of an $\ell 1$-regularized logistic regression model in removing irrelevant or ineffective genes from the data is discussed.
2. An experiment conducted to demonstrate that an ensemble of inflexible (linear) classifier can be used to derive the classification function for various kinds of cancers from microarray data is also discussed.
3. Finally, the use of subspaces from biologically-derived gene sets in improving the performance of ensembles used in cancer classification is discussed.

3.3 Data and Experiment Settings

The five datasets used throughout this work are described in Table 3.2. The data sets are polytomous cancer-related human gene expression datasets and they are available in the public domain.

The expression data are stored in expression matrices. The columns represent all the gene expression levels recorded during a single experiment, and the rows represent the expression of a gene across all patients or subjects involved in the experiment.

In this work, two biologically-derived gene sets were used. We used the C5 gene sets, genes associated with biological processes (BP), obtained from the Molecular Signatures Database (MSigDB) [38]. We also used C6 gene set that represents signatures of cellular pathways which are often dis-regulated in cancer.

Table 3.1: MSigDB Gene Sets Used In Our Work

Name	Description
c5.bp.v3.1.entrez.gmt [38]	It contains a set of genes that are active in one or more biological processes, during which they perform one or more molecular functions.
c6.all.v3.1.entrez.gmt [38]	It contains genes representing oncogenic signatures; that is, it contains a set of genes that are often dis-regulated in cancer.

The five gene expression data sets for various kinds of cancers are shown in Table 3.2. The data sets were obtained from microarray experiments on humans with various kinds of cancers and in some cases people without any kind of cancers.

Table 3.2: Cancer-related Human Gene Expression Datasets

Name	Description
BrainTumor1 [47]	This data set contains 90 samples. These samples represent five human brain tumor types. There are 5,920 genes.
BrainTumor2 [44]	This data set contains 50 samples. These samples represent four malignant glioma tumor types. There are 10,367 genes.
9Tumors [54]	This data set contains 60 samples. These samples represent nine various human tumor types. There are 5,920 genes.
11Tumors [55]	This data set contains 174 samples. These samples represent eleven various human tumor types. There are 12,533 genes.
ProstateTumor [51]	This data set contains 102 samples. The samples represent prostate tumor and normal tissues. There are 10,509 genes.

In this work, we used a tree classifier from Scikit-learn [45], an integrated machine learning Python implementation. We used Liblinear [18], a library for large linear classification, for SVM, and regularized logistic regression. All experiments were run on a Linux PC with Intel®Core™ i5-450M Processor.

3.4 Cross Validation

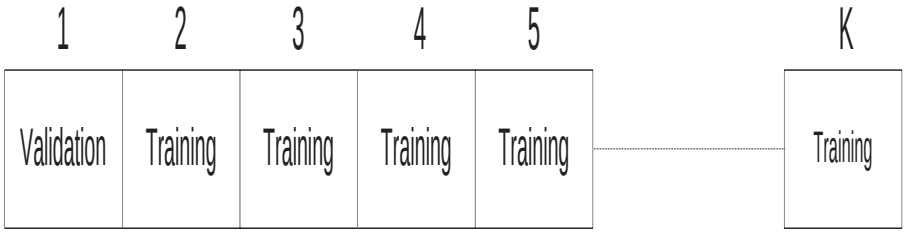

Figure 3.1: A K-fold Partition of a Dataset

We used 10-fold cross-validation to evaluate each of the approaches that were proposed in this work. The general framework for k-fold validation is as follows:

1. Partition the data into approximately k equal parts as shown in Figure 3.1.

2. Let $D = \{(x^i, y^i)\}, i = 1, \ldots, n$ be a data set and $\mathcal{L}_A$ be a learning algorithm that we wish to evaluate. $D = \bigcup_{i=1}^{k} D_i$, where D_i is a partition as shown in Figure 3.1.

3. Fit (train) $\mathcal{L}_A$ using the data set $D - D_i$ to obtain model $M_i = \mathcal{L}_A(D - D_i)$, for $i = 1, \ldots, k$.

4. Let $\hat{y}_{D_i} = M_i(x_{D_i})$, for $i = 1, \ldots, k$.

5. Evaluate $\mathcal{L}_A$ using some evaluation function E:
   $\sum_{i=1}^{k} E(y_{D_i}, \hat{y}_{D_i})$.

In addition to using cross-validation to evaluate the algorithms that are proposed in our work, we also used it for parameter optimization. Given some learning algorithm, $\mathcal{L}_A^{\theta}$, and $\{\theta_1, \theta_2, \ldots, \theta_t\}$, a set of candidate values for the parameter θ, we optimize θ by finding $\hat{\theta}$, the "best" θ_i. This process involves applying k-fold cross validation using $\mathcal{L}_A^{\theta_i}$ and data set D for $i = 1, \ldots, t$. θ is then estimated using $\hat{\theta}$.

Chapter 4

Gene Selection: A Two-Stage Approach

Identifying predictor genes is of practical interest to scientists and researchers. Research in medical genetics stands to benefit from further examination of the top-ranked genes to confirm recent discoveries and theories in cancer research. They may also suggest new hypotheses that are worth pursuing.

DNA micro-array technology provides a mechanism through which a huge amount of gene expression data is simultaneously recorded. Microarray data sets present several challenges, notable among which is the large number of gene expression values in a given gene profile and relatively few examples; the ratio of the number of genes to the number of examples is very large in a microarray analysis. As a consequence of this, the removal of irrelevant/ineffective genes has become a very important task.

Microarray analysis for purposes of cancer classification lends itself to many challenges. One inescapable challenge is noise which could adversely affect classification accuracy. Then there is also the high potential for spurious relationships: wrongly identifying a set of genes as being predictors for a given cancer when the genes may not in fact be predictors. The goal in any gene selection strategy becomes how best to reduce the potential for these problems.

The traditional approach to gene selection has often involved the ranking of genes on the basis of some test statistic. This usually involves the selection of top-ranked

genes on the basis of the significance of ranks assigned to each gene. Genes not meeting a given threshold are removed while those meeting the threshold are regarded as top-ranked genes. This approach ranks genes independently based on their individual discriminative power. For example, the correlation coefficient has been used to rank genes [46, 25]. This gene-by-gene ranking approach is very susceptible to the possibility that highly-ranked gene may be chosen over a good predictor: Consider situation in which we have two highly-ranked genes that are correlated and one lowly-ranked gene that is uncorrelated with the highly ranked ones. If a model is built using only the two highly-ranked genes, that model may not be the best since one of the gene chosen would not add any additional information to the model. On the other hand, if one highly ranked gene is chosen along with the lowly-ranked gene, a more informative model will be built.

Our approach seeks to avoid the pitfall of choosing only highly-ranked correlated genes. We propose a two-stage algorithm. In the first stage, we propose the use of an analysis of variance (ANOVA) algorithm that ranks the genes by how relevant the expression levels in the genes are to the target classes. In the second stage we propose the use of $\ell 1$-regularized logistic regression to remove genes that are correlated.

4.1 Stage One

Gene expression level data from microarray usually contains noise and is often highly correlated. In order to begin any analysis, as a filtering step, it is prudent to denoise and reduce the level of correlation in the data. Several approaches have been suggested in the literature to preprocess data. These include signal-to-noise ranking, Kruskal-Wallis non-parameteric analysis of variance (ANOVA) algorithm[23] and the $F - score$ (ratio of in between group variance to within group variance).

The signal to noise ratio (SNR) identifies the expression patterns with a maximal difference in mean expression between genes across all samples and minimal variation of expression for each gene across all samples. The use of SNR across all genes for ranking does not generally lead to selecting the most relevant genes. There are examples in the literature in which SNR is used in combination with clustering algorithms such as K-means [41] and KNN [58](k nearest neighbor). After clustering, SNR ranking is then applied to each cluster and the top-ranked gene from each cluster is chosen. Applying SNR ranking to gene clusters rather than all genes leads to better selection of relevant genes. SNR computes the difference in mean between two genes across all samples. It is extended for multiple genes by using strategies such as One-Versus-One or One-Versus-Rest[56]. SNR is not as robust as other gene selection algorithm since an assumption is made regarding the distribution from which genes are drawn: it is generally assumed that they are drawn from a Gaussian distribution because of the Central Limit Theorem.

In our work, we used the Kruskal-Wallis (KW) nonparametric one-way ANOVA[23] and the ratio of between group variance to within group (BW) variance[14]. Both of these algorithms are used to rank the genes.

In microarray data analysis, we often want to know how correlated the genes are to the target classes in a given experiment. If they are, then they are likely for serve as excellent predictor genes; that is, these genes are able to discriminate the target classes. One approach to identify genes that are correlated to the target classes in the classification of cancers from gene expression data is using the F-test [43]. The F-test uses variations among means to estimate variations among individual measurements. It assumes that the gene expression measurements are normally distributed. More

generally, the F-test is very sensitive to measurements that are not normal. When the measurements are not normal, the Kruskal-Wallis one way analysis of variance, a non-parametric approximation for the F-test, is used. The Kruskal-Wallis test uses variations among among ranked means to estimate variations among individuals measurements [9]. The preferred test to use with microarray data is the Kruskal-Wallis test rather than the F-test since it is non-parametric. Also, the Kruskal-Wallis test is easier to use and understand. It is very robust since it is less susceptible to noise: actual measurements are not being used but rather the ranks of the measurements. If the measurements are slightly off due to noise it is less likely to alter the mean of the ranks of the measurements than the mean of the actual measurements as would be the case with the F-test. However, when the conditions for the F-test are applicable to microarray data used in cancer classification, the F-test will generally yield better results than the Kruskal-Wallis one way of ANOVA test. The applicable conditions for the F-test are that the gene expression levels are normally distributed and the variance between these measurements with respect to the various genes are homogeneous.

4.1.1 Ranking Genes Using the Kruskal-Wallis Test

The Kruskal-Wallis (KW) one way analysis of variance (ANOVA) non-parametric test examines the significance of each gene through the expression level measurements of gene across all classes. The gene is assigned a rank based on how closely its expression levels across all examples are relevant to the target classes. When using the Kruskal-Wallis test, there are two tables that can be used depending on a set of conditions. These conditions are summarized in Table 4.1.

Table 4.1: Condition for Making a Decision Rule on the Computed K Statistic

Condition	Table to Use	Decision Rule
i=3 or more groups Number of observations in each group exceeds 5	Chi-square tabled values for df=C-l	If observed value of the K statistic is $\geq$ tabled value reject H_o
i=3 or more groups number of observations in each group is less than or equal to 5	Kruskal-Wallis critical values table	If observed value of the K statistic is $\geq$ tabled value accept H_o

We now discuss how the ranking and selection of genes are done during stage one of the algorithm using the Kruskal-Wallis one way ANOVA non-parametric test.

1. Input: Let $G_i = \{g_{i,1}, g_{i,2}, \ldots, g_{i,N}\}$ be a vector of gene expression measurements for the i^{th} gene across N samples. Each $g_{i,j}$ is the expression level of gene G_i in sample j.

2. Output: $R = \{R_1, R_2, \ldots, R_P\}$, where $R_i \in \{1, \ldots, P\}$ and $R_i \neq R_j$ for any $1 \leq i, j \leq P$ and $i \neq j$. R is the ranking of the genes based on the Kruskal-Wallis test statistic and P is the total number of genes.

3. For each gene G_i, rank all gene expression levels across all classes. Assign tied values, if any, the average of the ranks they would have received if they were not tied.

4. Compute the Kruskal-Wallis test statistic K_i for gene G_i:

$$K_i = \frac{12}{N\,(N+1)} \sum_{k=1}^{C} n_k \left(\bar{r}_k - \frac{(N+1)}{2} \right)^2$$

   - where N is the total number of expression level measurements across all samples for the i^{th} gene,
   - n_k is the number of expression level measurements for the i^{th} gene that are from class k, and
   - $\bar{r}_k$ is the mean of the ranks of all expression level measurements for the k^{th} class in the i^{th} gene.

5. We correct for ties by dividing K_i by:

$$1 - \frac{\sum_{i=1}^{T} (t_i^3 - t_i)}{N^3 - N}$$

   where T is the number of tie groups and t_i is the number of ties in the i^{th} tie group.

6. The null hypothesis in using the Kruskal-Wallis test statistic with the microarray cancer data is that expression of gene G_i is independent of the class label. If the test statistic $K_i \geq \chi^2_{\alpha:C-1}$ (χ^2_{C-1} is obtained by looking it up in a χ^2 table), we conclude that the expression is dependent on the class label. If $K < \chi^2_{\alpha:C-1}$, we conclude that the expression level of gene G_i is not dependent on the class label. We use $\alpha = 0.05$ as the level of significance of the test. The median is used as an approximation for difference in the variances of the expression levels of the i^{th} gene across samples.

7. Calculate the *p-value* of the i^{th} gene G_i, $Pr(X \geq K_i)$, where X is a random variable with χ^2 distribution and $C - 1$ degrees of freedom.

8. Rank each gene G_i according to its *p-value*. The lower the *p-value* of a gene, the higher its ranking.

4.1.2 Ranking Genes Using the F-score

The F-score is a ratio of two variables $F = \frac{F_1}{F_2}$, where F_1 is the variance between groups and F_2 is the variance within each group. A high F-score (which leads to a significant p-value depending on α value) means that at least one gene, with respect to its expression, is significantly different from the rest. F-score ranks each gene by grouping its expressions across all classes and then computing the ratio. We based the ranking on the ratio - the higher the F-score, the higher the rank of the gene: a large F-score indicates the gene is more discriminative.

We now discuss how the ranking and selection of genes are done during stage one of the algorithm using the F-score.

1. Input: Let $G_i = \{g_{i,1}, g_{i,2}, \ldots, g_{i,N}\}$ be a vector of gene expression measurements for the i^{th} gene across N samples.

2. Output: $R = \{R_1, R_2, \ldots, R_P\}$, where $R_i \in \{1, \ldots, P\}$ and $R_i \neq R_j$ for any $1 \leq i, j \leq P$ and $i \neq j$. R is the ranking of the genes based on the F-scores and P is the total number of genes.

3. For each gene G_i, group the expression levels across sample by the class.

4. Compute the F-score test statistic F_i for gene G_i:

$$F_{i,1} = \frac{\sum_{k=1}^{C} \left(\bar{Y}_{i,k} - \bar{Y}_i\right)^2}{C-1}$$

$$F_{i,2} = \frac{\sum_{k=1}^{C} \sum_{j=1}^{n_{i,k}} \left(Y_{i,k,j} - \bar{Y}_{i,k}\right)}{N-C}$$

$$F_i = \frac{F_{i,1}}{F_{i,2}}$$

   - where $F_{i,1}$ is the variance of the expression levels between classes of gene G_i,
   - N, the total number of expression level measurements across all samples for the i^{th} gene,
   - $\bar{Y}_{i,k}$ is the mean expression level of class k in gene G_i,
   - $\bar{Y}_i$ is the mean expression level of gene G_i,
   - C is the number of classes,
   - $F_{i,2}$ is the variance of the expression levels within classes of gene G_i,
   - $Y_{i,k,j}$ is the j^{th} expression level in class k of gene G_i,
   - $\bar{Y}_{i,k}$ is the mean expression level of class k in gene G_i, and
   - F_i is the ratio of between class variance to within class variance of gene G_i.

5. Rank each gene g_i according to its *F-score*. The higher the *F-score* of a gene, the higher its ranking.

4.1.3 Filtering Out Genes

After the genes are ranked, the lower ranked genes are then filtered out. In order to filter out some genes, the following steps are applied:

1. Input: Gk and Gf, are the genes sorted by the rankings obtained by using the Kruskal-Wallis test and the F-test, respectively.

2. The number of genes selected from the ranked genes can be either fixed or optimized by cross-validation on the training set. The following parameters are considered during the optimization:

   ⊙ minimum # of genes

   ⊙ maximum # of genes

   ⊙ step size

   For example, if max#genes = 300, min#genes = 100 and step size = 50, then the following number of top-ranked genes are considered for optimization: $\#top\text{-}genes = \{100, 150, 200, 250, 300\}$.

3. Run the linear Support Vector Machine [8] (SVM) extended for multi-class by the One-Versus-Rest strategy [67]. Set the cost of the SVM to some fixed value C. Run SVM while restricting the genes to $\#top\text{-}genes$ genes from the Kruskal-Wallis and F-score rankings.

4. Select the optimal $\#top\text{-}genes$ genes from either G_k or G_f based on the run of SVM which gives the best accuracy.

4.2 Stage Two

The first stage of our proposed two-stage gene selection algorithm consists of the ranking of genes based on their correlations to the target cancer classes. The Kruskal-Wallis test and the F-test do not consider the correlation of genes; that is, they do not remove genes that are redundant and consequently do not add any additional information to the classification model. We found that $\ell 1$-regularized logistic regression model can be used to remove redundant/ineffective genes from data sets. With only a few restrictive parameters, it can significantly reduce the number of genes without any signicant adverse impact on classication accuracy. The coefficients of genes that are not good predictor genes for the cancers under consideration are set to zero in the model. Genes which are redundant are removed from the data.

Logistic regression belongs to the category of statistical models called generalized linear models. Logistic regression allows for the prediction of discrete outcomes from a collection of variables that may be dichotomous, discrete, continuous, or a combination of any of these characteristics. In Simple Logistic Regression (SLR), the dependent variable is dichotomous; that is, the dependent variable can take on only one of two possible values, say +1 and -1, with probability q and $1-q$, respectively. The natural extension of SLR is a logistic regression model in which the dependent variable is polytomous. This is the Multinomial Logistic Regression (MLR).

Logistic regression, as a statistical formulation, can be defined in a probabilistic framework. Suppose C is a number of classes such that $y^{(i)} \in \{1, \ldots, C\}$. Given a vector of gene expression values x belonging to a person, the conditional probability that the person belongs to class t, $t \neq C$, is defined as

$$\Pr(y = t \,|x\,) = \frac{e^{w_t x + b_t}}{z} \qquad \text{(Eq.4.2.1)}$$

and for $t = C$ is

$$\Pr(y = C \,|x\,) = \frac{1}{z} \qquad \text{(Eq.4.2.2)}$$

where $z = 1 + \sum_{t=1}^{C-1} e^{w_t x + b_t}$ normalizes the probabilities of the classes for x so that they sum up to 1; that is, $\sum_{t=1}^{C} \Pr(y = t \,|x\,) \;=\; 1$.

Given a gene expression vector x for a new person, we can calculate the probability $\Pr(y = t \,|x\,)$ for all classes $t = 1 \ldots C$. The data point x is then assigned the class t which gives the largest probability.

The parameters w_t and b_t are obtained by maximizing the log-likelihood (minimizing the negative of the log-likelihood) of the training dat: minimizing

$$\min_{\substack{w_1, w_2, \ldots \\ b_1, b_2, \ldots}} \left\{ - \sum_i \log \Pr(y = y^{(i)} \,\big|x^{(i)}\,) \right\} \qquad \text{(Eq.4.2.3)}$$

where $\Pr(y = t \,|x\,)$ is given in Eq.4.2.1. The logistic regression with $\ell 1$-regularization adds an $\ell 1$-norm penalty term to Eq.4.2.3 and we obtain

$$\min_{\substack{w_1, w_2, \ldots \\ b_1, b_2, \ldots}} \left\{ - \sum_i \log \Pr(y = y^{(i)} \,\big|x^{(i)}\,) \right\} + \sum_t \|w_t\|_1. \qquad \text{(Eq.4.2.4)}$$

The Regularization Theory with respect to a linear classifier may be formally stated as follows: let w be the a vector of parameters for a classifier and *L(X, Y,w)* be a loss function that measures how well the classifier using w can make prediction on the set of training data *(X, Y)*. Regularization penalizes the complexity of a learning model; it penalizes the model depending on the number of parameters that the model has.

Logistic regression yields a linear classifier. For a binary classifier, the decision boundary is given by $e^{w_t x + b_t} = 1$. This equation represents a hyperplane in the data space. Logistic regression implicitly performs feature selection by making some of the parameters w_i's zero in the regression equation: the redundant genes are effectively

removed from the equation. The second stage of our model uses logistic regression to eliminate the redundant genes. The second stage of the algorithm is summarized below:

1. Input: Let G' be the set of t top-ranked genes selected after stage one of the algorithm and X be a set of gene expression data.
2. Restrict the gene expression data, X, to those genes in G'.
3. Apply cross validation to obtain an optimal value for the hyper-parameter λ from the list $\lambda \in \left\{1^k\right\}$, $k = -1, 0, 1, \ldots, 4$.
4. Using the logistic regression model that gives the optimal λ, remove all genes from G' such that the coefficient of the gene in the model is 0 to obtain a new set of genes G".

4.3 Results

An empirical study was conducted to evaluate the performance of our two-stage gene selection algorithm. These studies were intended to show that performing gene selection using our two-stage algorithm will not adversely affect classification accuracy and that $\ell 1$-regularized logistic regression performs implicit gene selection.

First, we applied logistic regression to the data sets without any gene selection as the baseline case, while recording the classification accuracies for each data set. 10-fold cross validation was used to evaluate the accuracy of the regression model on each data set. The hyper-parameter λ used in this study was optimized from $\lambda = \{10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3\}$.

Table 4.2 shows the baseline case: the accuracies obtained from 10-fold cross-validation when applying the logistic regression model to the data sets. The third column shows the total number of genes in the data set before $\ell 1$-regularized logistic regression is applied and the fourth column shows the average number of genes in the data set after the the regression was applied. The fourth column of the table

contains the average number of genes because the number of genes varied over the 10 cross-validation runs.

Table 4.2: Gene Selection Using Only $\ell 1$-regularized Regression

Dataset	(L1LRM)% Accuracy	# of Genes in data set	Average # of Genes After L1LRM
BrainTumor1	88.9	5291	2891
BrainTumor2	40.0	10368	6098
9Tumors	60.0	5727	443
11Tumors	93.6	12534	1186
ProstateTumor	92.1	10510	100

Table 4.3 shows the number of genes in the data set before the second stage of the algorithm and the average number of gene left in the data set after the second stage of the algorithm. The comparison of accuracies and average number of genes across all test cases during the 10-fold cross-validation for $\ell 1$-regularized logistic regression model are shown in Table 4.3.

Table 4.3: Gene Selection Using 2-Stage $\ell 1$-RLRM

Dataset	(L1LRM)% Accuracy	# of Genes Before Stage 2	Average # of Genes After Stage 2
BrainTumor1	91.1	5000	2667
BrainTumor2	44.0	9000	5956
9Tumors	65.0	5000	2160
11Tumors	94.8	11000	3406
ProstateTumor	96.1	9000	102

4.4 Observations

Across all datasets, the two-stage gene selection algorithm succeeded in significantly reducing the number of effective genes in the data sets without hurting classification

accuracies. For example, the two-stage gene selection algorithm results in a model that has only 102 genes that achieves 96.1% accuracy on the ProstateTumor dataset. While the application of only $\ell 1$-regularized logistic regression model to the same data set removes slightly more genes, its accuracy was 92.1%. Similarly, with the 11Tumors dataset set, the two-stage gene selection algorithm results in a model that has 3406 genes and achieves a classification accuracy of 94.8%. This suggests that the two-stage gene selection approach does not have any adverse effect on accuracy. In fact, it improves classification accuracy while at the same time selecting effective genes. For example, in the case of the BrainTumor1 and BrainTumor2 datasets, the two-stage gene selection algorithm results in fewer genes in the model after the algorithm is applied while improving classification accuracies.

While it is true that applying only regression may sometimes remove more genes than the two-stage gene selection algorithm that we have proposed, classification accuracy may be hurt when that is done. From Table 4.2 and Table 4.3, we see that two-stage gene selection algorithm selects more genes than the use of only regression on the original data set on the 11Tumor and the 9Tumors data sets. In particular, applying regression only, selects 443 genes from the 9Tumors data set while the two-stage algorithm selects 2160 genes. Also, applying only regression selects 1186 genes from the 11Tumors data while the two-stage algorithm selects 3406 genes. In these cases, more sparse model does not lead to improved classification accuracy. While these results may appear to be anomalous, when one considers the possibility that the regularized regression may be over-compensating for noise in the data, we can explain them.

In some instances, the use of only regularized regression may lead to over-fitting;

it is creating a more sparse model that performs well on the training sample but does not perform as well on the new instances that it has not seen. This suggests that while a sparse model may be a good thing, one must guard against over-fitting. It would be better to have a model that it not as sparse that generalizes: a model that performs well in classifying instances that it has not seen. This observation reinforces the need for the first stage of the algorithm in eliminating some genes that are low-ranked, so that the regularized regression can create a model that generalizes well even if the model is at times less sparse.

We also observed that in spite of the two-stage gene selection approach, classification accuracy was poor when there were relatively few examples, as was the case with the BrainTumor2 and 9Tumors data sets. This suggests the need for a complementary approach to handle the situation of over-fitting that is due largely to there being only a few examples relative to the number of genes.

Chapter 5

Ensemble Cancer Classification: Choice of Base Classifiers

Even in an ideal world where the best predictor genes are chosen for use in a single-classifier system designed for microarray cancer gene expression data, there is still the challenge of having an ultra high-dimension space with relatively few examples.

A critical task of any learning system is to draw inferences about a general classification function from a training sample. In training a classifier, a learning algorithm searches through a hypothesis space for a hypothesis that "best" fits the training examples. A hypothesis space is a predefined space of candidate hypotheses, often implicitly defined by how the hypothesis is represented. A difficulty that a single-classifier model has to overcome when classifying microarray data is how to find or approximate the true hypothesis. The classification model must search an enormous space of hypotheses using only a few examples. While many of the hypotheses in this space may appear to be good candidate hypotheses for the few examples, there is always the potential that the candidate hypothesis that is selected may fail to *generalize*; it may adequately describe the few gene expression profiles that it has seen but would be unable to correctly classify future profiles.

An approach that might be considered to address this problem of over-fitting is to select a hypothesis that is not the "best" for the training examples. There might be other hypotheses that would more accurately classify the training example than the one chosen. This choice would be made in hopes that it might perform better on

future instances that it has not seen. There is however the inescapable problem of not knowing what future instances will look like. We have no way of knowing that since we can only use the training examples to predict what new instances might look like. After all, It may just be that the "best" candidate hypothesis in the training examples is the best approximation of the true hypothesis. Therefore, either using the "best" hypothesis from the training examples or a good hypothesis that is not necessarily the best as an approximation for the true hypothesis may not lead to good classification accuracy on new instances.

Also, there is still the possibility that given the relatively few training examples, there is no one candidate hypothesis that can adequately describe future data. Instead, the true hypothesis may be more adequately described by combining several hypotheses than using a single hypothesis. By combining several hypotheses, the likelihood of better approximating the true hypothesis will improve. This approach overcomes some of the pitfalls involved in a single-classifier system.

One approach that has been proposed in the literature to alleviate the problems inherent in the use of a single classifier in an ultra high-dimensional space with relatively few examples is an ensemble classifier [35]. An ensemble classifier consists of a set of individually trained *base classifiers* whose predictions are pooled to classify new instances. This is analogous to a real-world situation in which we solicit the views and opinions of several individuals to formulate our own opinion. The individuals whose opinion and views that we solicit play the role of base classifiers while we play the role of an ensemble classifier. We now consider a formal definition of an ensemble:

1. Let $D = \{(x^j, y^j)\}$, $j = 1, 2, \cdots$ be a set of instance-label pairs drawn from some distribution. X is a set of instances and $x^j \in X$. Each $y^j \in Y$ is a label and $Y = \{1, 2, \cdots, k\}$ is a set of labels representing classes.

2. Use a training set construction method to construct $T_1, T_2, T_3, \ldots$ from D.

3. $\varsigma : X \to Y$, a function that maps instances to classes, is called a *classifier* or classification function.

4. $E = \{\varsigma_1, \varsigma_2, \varsigma_3, \ldots\}$ is an ensemble and $\varsigma_{T_i}(x) \to y$ for some $y \in Y$.

5. Let g be a function that combines all ς_i's, then $y_t = g\left(\varsigma_{T_1}(x), \varsigma_{T_2}(x), \ldots\right)$, where $y_t \in Y$. For example, g could be the function $\sum_{i=1}^{n} \omega_i \varsigma_i \to Y$ where $\Omega = \{\omega_1, \cdots, \omega_n\}$ is a set of weights such that the vote of ς_i, the i^{th} classifier, is weighted by ω_i.

There is continuing interest in constructing multiple classifiers and then combining them in some way to obtain a classifier that yields better performance in machine learning applications. It is therefore natural to use an ensemble classifier in cancer classification from microarray data. Popular approaches to constructing an ensemble classifier have included bootstrap aggregation ("bagging") [7] and a generalized additive model construction technique known as boosting [19]. There is also an approach to constructing ensembles that involves manipulating the class labels of the samples [65].

We earlier discussed some of the challenges inherent in the use of a single classifier in classifying cancers from microarray gene expression data. The use of an ensemble classifier addresses several issues relating to learning algorithms in the context of an

ultra high-dimensional feature space such as it is the case with microarray data. An ensemble combines a collection of optimal solutions, each based on a variation of the examples and is therefore able to better approximate the true hypothesis.

Some of the key concerns that are explored when designing an ensemble are

(i). what characteristics make a classifier suitable as a base classifier,

(ii). how should base classifiers then be combined, and

(iii). how does the accuracy of the ensemble compares to the accuracies of the underlying base classifiers.

We considered both linear and non-linear base classifiers. we sought to determine whether a flexible or inflexible base classifier will improve the performance of an ensemble. we conducted several empirical experiments to provide some insights into these questions. We studied the impact of three kinds of base classifiers: decision trees, support vector machine (an $\ell 2$-regularized linear classifier), and an $\ell 1$-regularized linear classifier.

5.1 Decision Tree Classifier

Decision tree learning is widely used in machine learning to approximate discrete-value functions [42]. The goal is to create a model that predicts the value of a target variable based on several input variables. The tree is constructed by recursively growing nodes that partition the training examples into groups with increasingly better impurity measure.

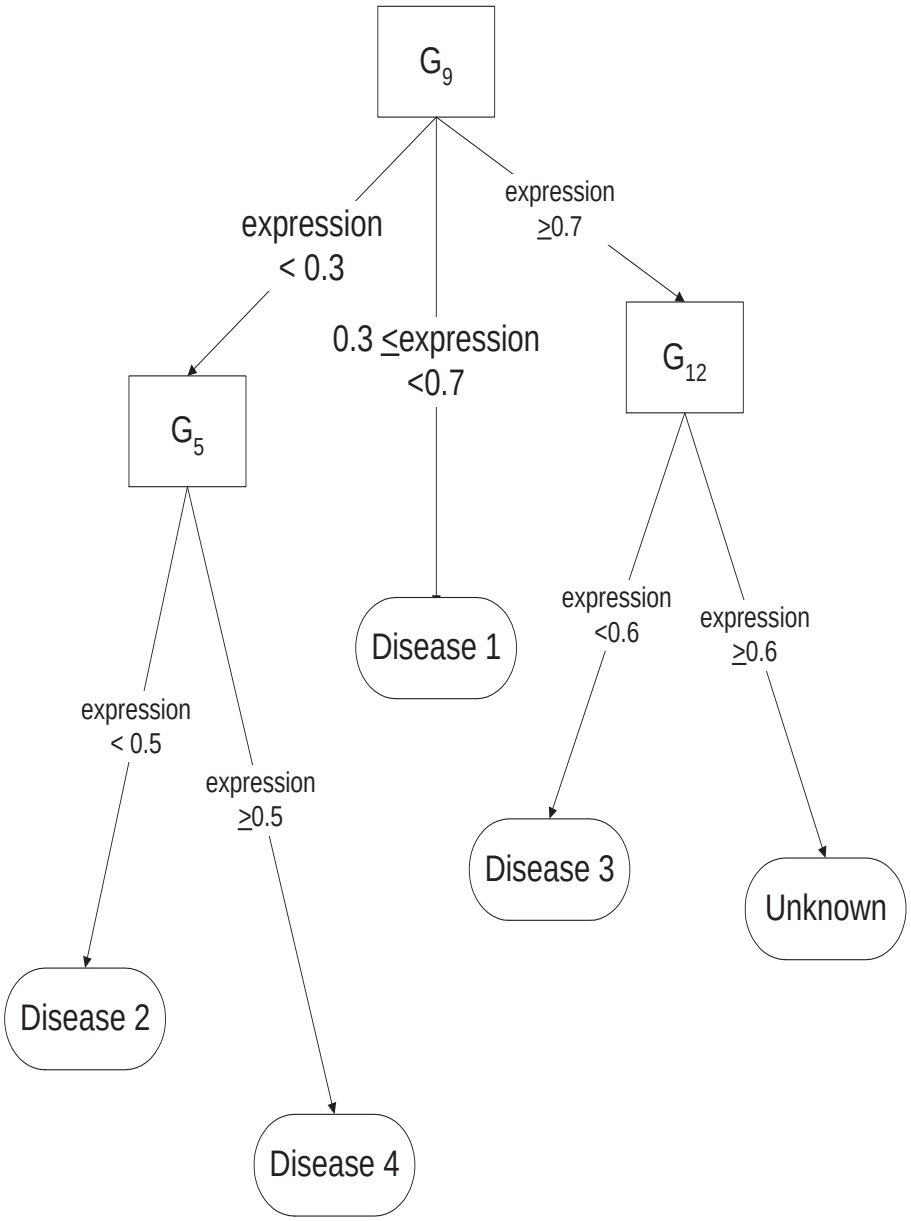

Figure 5.1: A Decision Tree That Classifies Four Diseases Based on Normalized [0,1] Gene Expression Data

Decision trees classifies instances in a hierarchical manner from the root to the terminal nodes of the tree as illustrated in Figure 5.1. Each internal node of the decision tree specifies a test on a given attribute while the branches emanating from a node represent one of the possible values of the attribute.

Four properties associated with decision trees [42] make them suitable for the cancer classification problem:

- The examples can be represented as value-attribute pairs. In the context of cancer classification using gene expression data, each example can be represented as a pair consisting of a gene and its expression. For each instance, there are a series of gene and gene expression pairs in the gene profile in a given microarray experiment. The underlying assumption is that the nature of those pairs determines the classification of the gene profile.
- The target of the classification function is discrete. This applies to the cancer classification problem since the classification function places each gene profile in some class $\{1, \ldots, k\}$, where each label represents some cancer type or no cancer at all.
- Decisions trees are generally considered an excellent way of modeling a disjunction of conjunctions. Each unique path from the root to the terminal node represents a conjunction. The path in the decision tree are ORed together so that they form a disjunction.
- They have been shown to be robust to noise or error in the data. They are less susceptible to errors in the values of the attributes and errors in the classification of the examples. Microarray data may contain one or both of these errors.

One issue which arises when constructing a decision tree is which attribute to test at a given node. In the case of the cancer classification in the gene expression domain, the question would be which gene to test at a given node. The general approach is to use some statistical measure to determine which gene acting alone would correctly classify the most examples. The same approach is followed at every subtree until all the attributes are eventually placed in the tree. In our investigation, we used the

Gini index [24], a measure of statistical dispersion, as the impurity measure while constructing the tree.

Like any supervised learning model, the decision tree model searches a space of hypotheses for a hypothesis the fits the training examples. This space of hypotheses consists of several decision trees. The goal of the search is to find the decision tree that best fits the training examples.

Unlike a linear classifier that partitions the sample space using a hyperplane, it partitions the sample space into non-linear regions. A decision tree is a non-linear classifier. Tree classifiers partition the sample space into non-linear regions: it creates decision boundaries that are non-linear. A Decision Boundary is a partition in n-dimensional space that divides the space into two or more response regions. Multiple planes serve as delimiters between regions. Tree classifiers are more flexible and they are preferable in dealing with data that cannot be effectively separated using a hyperplane. In addition to their ability to handle non-linear data, tree classifiers perform implicit feature selection, they require very little preprocessing of the data by the user, and they are conceptually easy to understand because of the hierarchical nature of a tree.

In spite of this inherent flexibility, tree classifiers may not always generalize. The advantages of tree classifiers are further tempered by the fact that they may grow very rapidly without an effective pruning strategy. This concern may be alleviated in ensembles that use tree classifiers because the trees are often shallow. In our work, we experimented with tree classifiers of varying depths as base classifiers and used the best performing tree classifier in giving a comparative analysis of an ensemble of trees with other ensembles.

5.2 Support Vector Machine

A Support Vector Machine (SVM) [8, 62] is a binary classifier that decides class membership by comparing a linear combination of the features to a threshold. It partitions the data space using a hyper-plane. A linear SVM trains on data that are separable unlike a non-linear SVM which trains on non-superable data. It is a very dynamic learning model because it lends itself to a simple geometrical interpretation. However, its critical limitation stems from the choice of kernel that is used during the learning process.

SVMs are a relatively new model for learning linear decision surfaces. Although neural networks and decision trees are efficient at learning non-linear surfaces, they generally have many more parameters that have to be simultaneously fine-tuned. Both of these models are more susceptible to getting stuck in local optima than other learning models. Over the last two decades, there has been a rash of new developments in computational learning theory. New algorithms with better computational and theoretical properties have been developed over this period. The development of the SVM model is one such innovation. The development of this model stems from new advances in techniques to separate non-linear decision surfaces through the use of kernel functions [49]. The functions can be used to manipulate data in its original space as if it were projected into a higher dimensional space. Kernalization often leads to more efficient learning models. It leads to algorithms based on optimization rather than greedy search. These algorithms, therefore, have very nice theoretical properties.

A support vector machine model finds an optimal hyperplane on the decision surface subject to the condition of linear separability of the data space. When the

decision surface is not linearly separable, kernalization may be used to transform the data into a modified space to ensure linear separability. There are many properties which make SVM a very robust and dynamic learning model:

- The SVM model can be modularize.
- It is not susceptible to the curse of dimensionality.
- The issue of getting stuck in local optima does not arise.
- It can be expressed in parametric form.
- A theoretical upper bound on its performance can be derived.

The support vectors are constituted by using data points that lie closest to the decision surface and are very difficult to classify. They contribute to determining where the decision surface can be located to optimize the performance of the SVM.

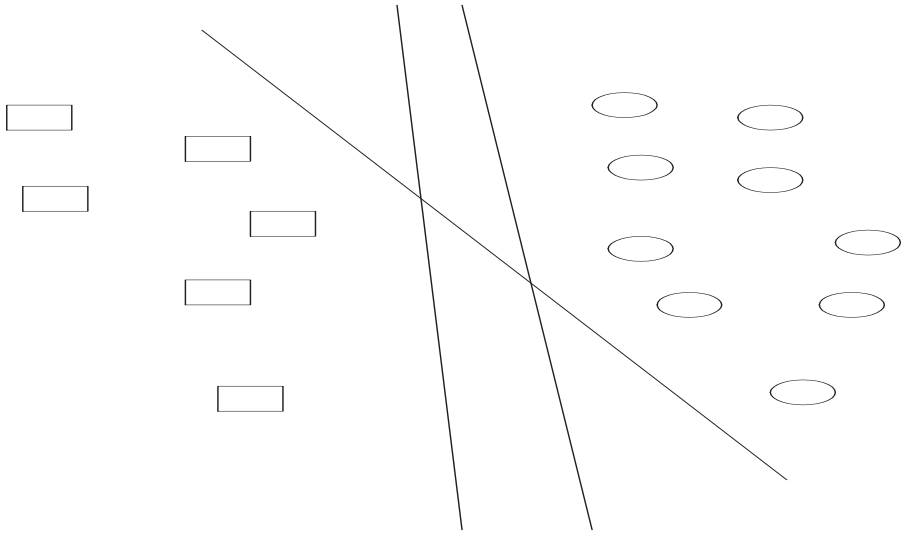

Figure 5.2: Hyperplanes Between Linearly Separable Data

Generally, there may be many hyperplanes that linearly separate the data. The SVM model finds an optimal one. It maximizes the margin around the separating hyperplane. Unlike models such as linear regression and Naïve Bayes which use all the points in the data space, it uses only those that are very difficult to classify. These points form the support vectors. These vectors are those that will alter the direction of the hyperplane when removed. Analytically determining the hyperplane is an optimization problem that may be solved using Lagrangian multipliers [63], a strategy for finding the local maxima and minima of a function subject to equality constraints.

We used SVM as an example of an $\ell 2$-regularized linear classifier, an example of an inflexible classifier. SVM makes classification using a linear function:

$$f(x) = wx + b$$

The function f has two parameters w and b. These parameters are determined from the training data. The function f partitions the sample space into two regions. The parameter w is a normal vector to the hyperplane. The offset of the hyperplane from the origin along the normal vector w is determined by the expression $\frac{b}{\|w\|}$. For any new data point x, $f(x) \geq 0$ or $f(x) < 0$. For a binary classifier, let -1 denote one class and +1 denote the other. $y^i \in \{-1, +1\}$ and $y^i = f(x^i)$. Given a training set of N data points,$\left\{ \left(x^{(i)}, y^{(i)}\right) \right\}_{i=1}^{N}$, where $x^{(i)} \in \mathbb{R}^n$ is the i^{th} input pattern and $y^i \in \mathbb{R}$ is the i^{th} output pattern. The support vector method approach aims at constructing a classifier. Training the SVM consists of solving the optimization problem:

$$\min_{w,b} \lambda \left\{ \sum_i \left[y^{(i)}(wx^{(i)} + b) - 1 \right]_+ \right\} + \|w\|_2^2 \qquad \text{(Eq.5.2.1)}$$

where $[\ ]_+$ is the hinge loss function given by the expression

$$[x]_+ = \max(-x, 0).$$

Although linear SVMs are inherently binary classifiers, they can still be used in scenarios in which a data point may belong to one of many categories as is the case with polytomous microarray cancer data. The multi-classification problem can be trivially formulated as a chain of unlinked binary problems which can then be naturally solved using binary classifiers. The one-vs-all (OVA) and all-vs-all (AVA) formulations are two popular strategies used in extending binary classifiers to solve multi-classification problems. The performance of the linear SVM as base classifiers was studied in our work. Cross-validation was used on the training set to estimate the hyper parameter λ.

5.3 ℓ1-regularized Logistic Regression Model

Logistic regression can be used when our data is dichotomous, polytomous, or ordinal. It works if the data is discrete or continuous. Unlike linear regression, it works with non-normal data since it makes no assumption about normality. In chapter 4, we discussed the mathematical framework for the ℓ1-logistic regression model. (See Eq.4.2.1, Eq.4.2.3, Eq.4.2.4 and Eq.4.2.2 for the probability and optimization equations.)

Logistic regression is a regression analysis used for predicting categorical dependent variable. A categorical dependent variable is one that takes on a finite number of ordinal values whose magnitudes are not important. In some instances, the way in which the magnitudes are ordered may be meaningful. In the context of the use of the logistic regression model in cancer classification, each ordinal value associated

with a cancer type only represents a label and the actual magnitude does not matter.

Logistic regression is very widely used in many application domains. Some properties that make it a useful model are:

1. The independent variable in a logistic regression model does not have to be normally distributed or have the same variance in each group. It is very robust.

2. It makes no assumption regarding linearity and is capable of handling non-linear effects. The classification boundary of a logistic regression classifier is a line; that is, logistic regression is a linear classier. However, the dependent variable is not a linear function of the independent variables ($y = \sum_{i=1}^{n} \omega_i x_i$ in not true for logistic regression.).

3. It does not require that the independent variables be unbounded as is the case in other regression models.

4. The model can be explicitly given as a mathematical equation:

$$\log it(p) = \ln\left(\frac{p}{1-p}\right)$$
$$\log it(y) = \sum_{i=1}^{n} \omega_i x_i$$

Although logistic regression models are robust, they have some limitations. They generally require a lot of data to be stable and provide useful results. A large number of examples are needed to ensure the stability of the logistic regression regression model. The more the dependent variables, the more the number of examples required to train the model. A sample size of at least 400 is recommended for a logistic regression model [28].

The multi-class extension of the logistic regression model with $\ell 1$-regularization is used as base classifiers in our work. The $\ell 1$-regularized regression model is both a linear (inflexible) model and a model that uses regularization. It was used to study the effect of regularization on the choice of the base classifier in an ensemble.

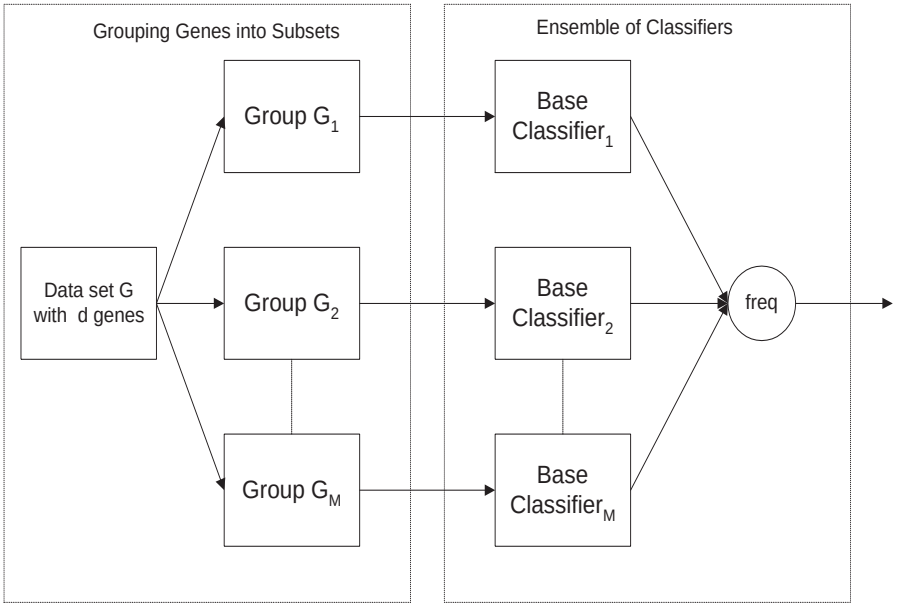

Figure 5.3: A Framework for Ensemble Classification Using Grouping Technique

5.4 Method

Construct base classifiers from randomly generated gene subsets, as shown in Figure 5.3. The randomly-generated gene subsets are constructed as follows: Let $X = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$ be the gene expression data for N training examples. Each $x^{(i)}$ is a *d-dimensional* vector of gene expression level data, where d is the number of genes in each sample. $x_j^{(i)}$ and $x_j^{(k)}$ represent the expression level for $gene_j$ in the i^{th} and the k^{th} samples. Let $Y = \{y^{(1)}, y^{(2)}, \cdots, y^{(N)}\}$ denote corresponding labels for each training sample. An ensemble classifier was constructed as follows:

1. Generate M subsets of genes from the set of d genes involved in the gene expression data that constitute the feature space. Each subset consists of k randomly selected genes.

2. M base classifiers are trained. The i^{th} base classifier is trained using the training samples and genes in the i^{th} subset; that is, each classifier is trained using $\{\hat{x}^{(1)}, \hat{x}^{(2)}, \cdots, \hat{x}^{(N)}\}$ and labels $\{y^{(1)}, y^{(2)}, \cdots, y^{(N)}\}$ where $\hat{x}^{(i)}$ is the vector derived by taking the elements of $x^{(i)}$ that correspond to the genes in the i^{th} gene subset.

3. To classify a new sample, the predictions from each base classifier are pooled together and a final decision is determined by committee vote. So, when there are multiple classes (for example, disease conditions 1, 2, etc), the predicted classification of the new sample is based on the majority vote of the base classifiers.

5.5 Results

The three classifiers - decision tree, support vector machine, and $\ell 1$-regularized logistic regression model - were used as base classifiers in the ensemble classifier. The decision tree is representative of a flexible (non-linear) classifier. The support vector machine and the logistic regression model are representative of linear classifiers that are regularized. The empirical data from these experiments were recorded and analyzed.

We used ensembles consisting of 200 base classifiers. Each gene subset was randomly selected with the number of genes in each subset being equal to the square root of the number of genes in the data set. Preliminary gene selection was done prior to each subset being used in the ensemble. Two kinds of ensembles with linear base classifiers were constructed. One ensemble used $\ell 1$-regularized logistic regression models as base classifiers while the other used support vector machines. SVMs were

used as $\ell 2$-regularized linear classifiers. Finally, we created ensembles of decision trees as an example of the use of non-linear base classifiers in an ensemble.

The comparison of the accuracies of ensembles with the three kinds of base classifiers are reported in Table 5.1. For each data set, 10-fold cross-validation was used to evaluate the effect of the various base classifiers in the ensembles. The second column in the table shows the mean classification accuracy±standard deviation after 10 repetitions of 10-fold cross-validation on the data sets when the $\ell 1$-regularized logistic regression models are used as base classifiers. The third column in the table shows the mean classification accuracy±standard deviation after 10 repetitions of 10-fold cross-validation on the data sets when support vector machines are used as base classifiers. The fourth column in the table shows the mean classification accuracy±standard deviation after 10 repetitions of 10-fold cross-validation when decision trees are used as base classifiers. The comparison is done using the data sets in Table 3.2. A visual depiction of the comparison of the mean accuracies is shown in Figure 5.4

Table 5.1: Comparison of Classification Accuracy of Various Base Classifiers

Datasets	$\ell 1$ LogReg	SVM $\ell 2$	Tree
BrainTumor1	90.4±0.3	91.6±0.6	91.6±1.0
BrainTumor2	84.2±1.4	47.0±3.0	77.2±1.4
9Tumor	79.3±2.5	73.7±1.8	60.7±2.7
11Tumor	94.7±0.5	94.3±0.5	87.2±1.6
Prostate	94.3±0.6	94.2±0.6	92.7±0.8

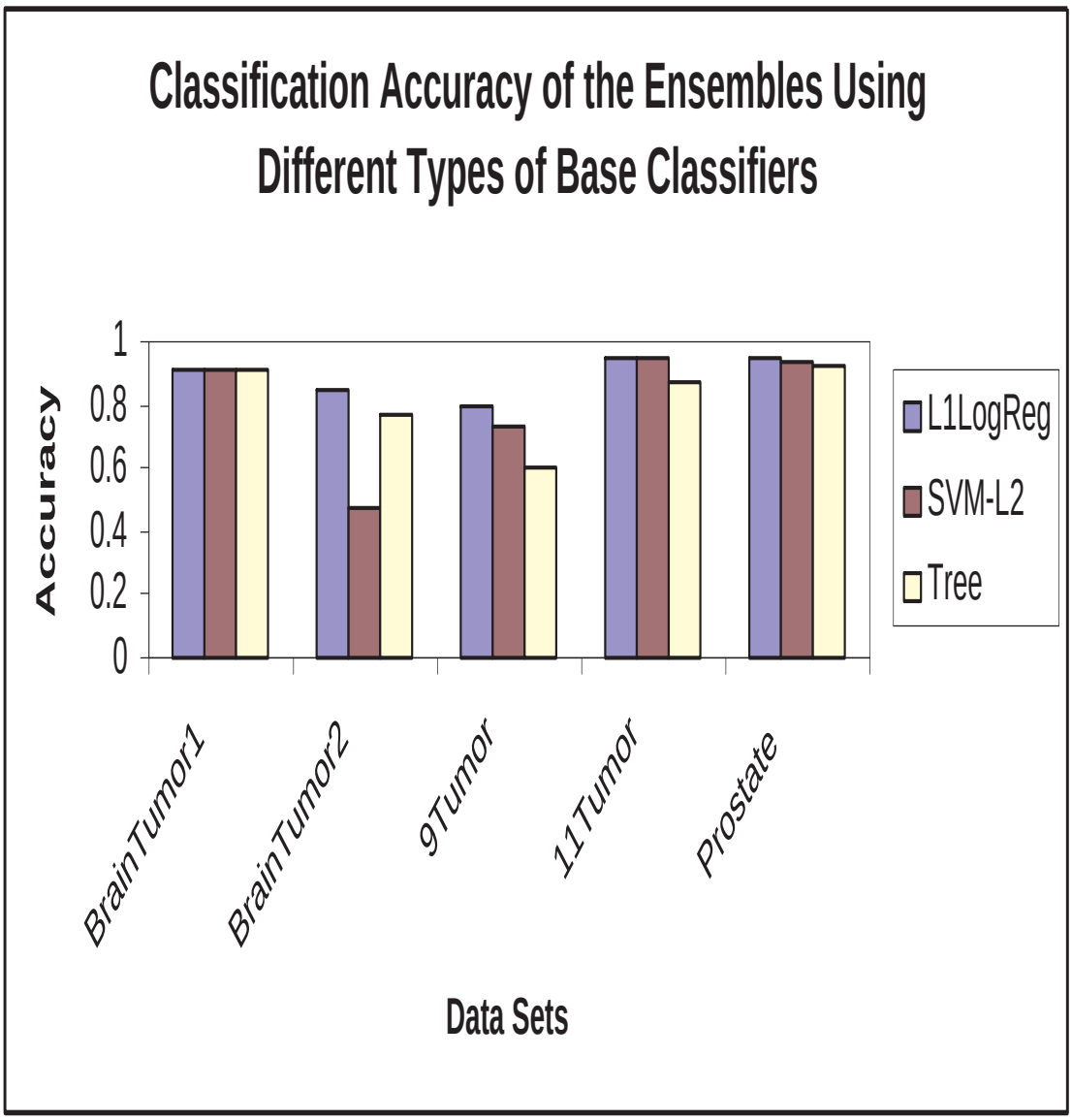

Figure 5.4: Classification Accuracy Using Various Base Classifiers

Tables 5.2-5.9 are the confusion matrices when $\ell 1$-LRM and Tree are applied to the first four data sets in Table 5.1. The entries in the tables represent the total number of classification and misclassification after 10-fold cross-validation is repeated 10 times. Therefore, the sum of the entries in each table is 10 times the number of examples in the data set.

The BrainTumor1 data set consists of five classes: Medulloblastoma(MB) = 0 (60 examples), Malignant glioma (MG) = 1 (10 examples), AT/RT(AR) = 2 (10 examples), Normal Cerebellum(NC) = 3 (4 examples), and PNET(PN) = 4 (6 examples).

Table 5.2: $\ell 1$-LogReg on BrainTumor1

		Predicted Labels				
		MB	MG	AR	NC	PN
True Labels	MB	600	0	0	0	0
	MG	0	90	10	0	0
	AR	7	0	93	0	0
	NC	10	0	0	30	0
	PN	39	12	1	0	8

Table 5.3: Tree on BrainTumor1

		Predicted Labels				
		MB	MG	AR	NC	PN
True Labels	MB	600	0	0	0	0
	MG	1	89	10	0	0
	AR	14	0	86	0	0
	NC	15	0	0	25	0
	PN	52	8	0	0	0

The BrainTumor2 data set consists of four classes: Classic Glioblastomas(CG) = 0 (14 examples), Classic Anaplastic Oligodendrogliomas (CAO) = 1 (7 examples), Non-classic Glioblastomas(NCG) = 2 (14 examples), Non-classic Anaplastic Oligodendrogliomas(NAG) = 3 (15 examples).

Table 5.4: $\ell 1$-LogReg on BrainTumor2

		Predicted Labels			
		CG	CAO	NCG	NAG
True Labels	CG	129	0	11	0
	CAO	30	40	0	0
	NCG	0	0	112	28
	NAG	0	0	13	137

Table 5.5: Tree on BrainTumor2

		Predicted Labels			
		CG	CAO	NCG	NAG
True Labels	CG	128	2	10	0
	CAO	30	40	0	0
	NCG	0	0	94	46
	NAG	0	0	34	116

9Tumors data set consists of nine classes: NSCLC(NS) = 0 (9 examples), Colon(CO) = 1 (7 examples), Breast(BR) = 2 (8 examples), Ovary(OV) = 3 (6 examples), Leukemia(LK) = 4 (6 examples), Renal(RN) = 5 (8 examples), Melanoma(ML) = 6 (8 examples), Prostate(PR) = 7 (2 examples) and CNS(CN) = 8 (6 examples).

Table 5.6: ℓ1-LogReg on 9Tumors

		Predicted Labels								
		NS	CO	BR	OV	LK	RN	ML	PR	CN
True Labels	NS	80	0	0	0	0	7	0	0	3
	CO	0	60	0	10	0	0	0	0	0
	BR	8	13	44	1	0	0	4	0	10
	OV	3	9	10	37	0	1	0	0	0
	LK	0	0	0	0	60	0	0	0	0
	RN	10	0	0	0	0	70	0	0	0
	ML	7	1	0	0	0	2	70	0	0
	PR	10	10	0	0	0	0	0	0	0
	CN	0	0	1	0	0	0	0	0	59

Table 5.7: Tree on 9Tumors

		Predicted Labels								
		NS	CO	BR	OV	LK	RN	ML	PR	CN
True Labels	NS	64	16	1	1	0	1	0	0	7
	CO	26	43	0	1	0	0	0	0	0
	BR	31	11	11	1	1	0	11	0	14
	OV	36	7	6	8	0	3	0	0	0
	LK	0	0	0	0	60	0	0	0	0
	RN	10	0	0	0	0	70	0	0	0
	ML	10	0	0	0	0	0	70	0	0
	PR	19	1	0	0	0	0	0	0	0
	CN	16	0	5	0	0	0	0	0	39

11Tumors data set consists of eleven classes: Ovary(OV) = 0 (27 examples), Bladder/ureter(BU) = 1 (8 examples), Breast(BR) =2 (26 examples), Colorectal(CR) = 3 (23 examples), Gastroesophagus(GS) = 4 (12 examples), Kidney(KN) = 5 (11 examples), Liver(LV) = 6 (7 examples), Prostate(PR) = 7 (26 examples), Pancreas(PC) = 8 (6 examples), Lung Adeno(LA) = 9 (14 examples) and Lung Squamous(LS) = 10 (14 examples).

Table 5.8: ℓ1-LogReg on 11Tumors

		Predicted Labels										
True Labels		OV	BU	BR	CR	GS	KN	LV	PR	PC	LA	LS
	OV	270	0	0	0	0	0	0	0	0	0	0
	BU	0	55	2	0	0	0	0	0	10	0	13
	BR	0	0	260	0	0	0	0	0	0	0	0
	CR	0	0	0	230	0	0	0	0	0	0	0
	GS	0	2	2	15	95	0	0	0	0	6	0
	KN	0	0	0	0	0	101	0	0	0	9	0
	LV	1	0	2	1	0	0	59	0	7	0	0
	PR	0	0	0	0	0	0	0	260	0	0	0
	PC	0	0	0	0	0	0	0	0	53	7	0
	LA	0	0	0	0	0	0	0	0	10	130	0
	LS	0	0	10	0	0	0	0	0	0	0	130

Table 5.9: Tree on 11Tumors

		Predicted Labels										
True Labels		OV	BU	BR	CR	GS	KN	LV	PR	PC	LA	LS
	OV	270	0	0	0	0	0	0	0	0	0	0
	BU	0	36	5	0	0	0	0	0	0	24	15
	BR	0	0	260	0	0	0	0	0	0	0	0
	CR	0	0	0	229	0	0	0	0	0	1	0
	GS	0	0	21	33	54	0	0	0	0	10	2
	KN	0	0	3	0	0	104	0	0	0	3	0
	LV	0	0	27	0	0	0	42	0	0	1	0
	PR	0	0	0	0	0	0	0	260	0	0	0
	PC	0	0	0	0	0	0	0	0	6	54	0
	LA	1	0	0	0	0	0	0	0	0	132	7
	LS	0	0	10	0	0	0	0	0	0	6	124

5.6 Observations

Across all the data sets except BrainTumor2, ensembles using regularized linear base classifiers outperformed those using trees. Also, the ensemble with $\ell 1$-regularized logistic regression model (linear classifier) consistently outperformed the others. In particular, with the 11Tumor data set, both ensembles with regularized base classifiers achieved classification accuracy around 94% while the ensemble that uses tree classifier only achieved around 87% accuracy. Across all the data sets, ensemble using $\ell 1$-regularized logistic regression model is consistently the leading performer. With 9Tumor data set, it reaches around 80% accuracy, with the ensemble using SVM coming second at 74% and the ensemble using classification tree last at only 61%. With the BrainTumor2 data set, it achieved an accuracy of 84% while the ensemble using classification tree achieves 77%. The ensemble using SVM performed poorly and achieved only around 47%. The results suggest that using a flexible classifier as a base classifier may not lead to a better ensemble in many cases. The performance of the ensemble using tree classifier is systematically lower than that of the linear classifiers with regularization. The simple decision boundary employed by the linear classifier does not prevent it from achieving better classification accuracy. Hence the complexity (flexibility) of the base classifier may not be a main concern in designing an ensemble.

Both $\ell 1$- and $\ell 2$-regularization produce classifiers whose coefficients have small magnitude. We further investigated which type of regularization gives a superior performance. From the results presented in Figure 5.4, we see that ensembles based on $\ell 1$-regularized logistic regression outperform those based on SVM. However, we notice

that $\ell 1$-regularized logistic regression and SVM differ not only on the type of regularization but also on the loss function. In order to rule out the possibility that the difference in performance is attributable to the different loss functions, we constructed another ensemble that uses $\ell 2$-regularized logistic regression models as base classifiers. It was the same as the ensemble consisting of $\ell 1$-regularized logistic regression models except that the $\ell 2$-norm was used as the penalty function. The $\ell 2$-regularized logistic regression ensemble performed similar to the SVM-based ensemble. It performed very poorly on the BrainTumor2 dataset. This suggests that it is $\ell 1$-regularization that contributes to a better performance in the ensemble. $\ell 2$-regularization helps to enhance the performance of an ensemble but it does not improve performance as much as $\ell 1$-regularization does. In a few cases, $\ell 2$-regularization may harm classification accuracy. This observation provides insights into the effect of regularization in ensemble classifiers.

From Tables 5.2-5.9, we see that $\ell 1$-regularized logistic regression classifiers are less susceptible to small number of examples per class than a decision trees. For example, on the BrainTumor1 data set, the regularized regression model correctly classified 30 out of 40 Classic Glioblastomas samples while SVM classified 25 out of 40. Although the regression classifier performed poorly in correctly classifying only 8 out of 60 PNET samples, the SVM classifier did not classify any of the samples correctly.

Similarly, on the BrainTumor2 data set, the regression model correctly classified 112 out of 140 non-classic Glioblastomas correctly, while the SVM classifier classified 94 out of 140. On the same data set, it classified 137 out of 150 Non-classic Anaplastic oligodendrogliomas correctly while the SVM classifier classified only 116 out of 150.

On the 11Tumors data set, the $\ell 1$-regularized regression ensemble correctly classified 59 out of 70 and 53 out of 60 liver and pancreas samples, respectively. On the other hand, the ensemble of SVMs correctly classified 42 out of 70 and 6 out of 60 of the same samples.

These results tell us that SVM base classifiers are more sensitive to relatively low class density than the $\ell 1$-regularized logistic regression base classifiers. By class density, we are referring to the ratio of the number of samples in that class to the total number of samples.

Chapter 6

Ensemble Cancer Classification: Construction of Ensemble

In the previous chapter, we showed by means of empirical results that an ensemble of $\ell 1$-regularized logistic regression models can yield excellent performance. Finding a good base classifier, although a significant step, is only a part of the puzzle. There is still the issue of how to construct ensembles that use the base classifier. An ensemble classifier works by running its base classifiers and then pooling their decisions to make its decision. Ensemble learning is a special case of supervised learning. Supervised learning involves inferring a classification function f from a set of labeled data. This set of labeled data is referred to as the *training set*. An ensemble is a multi-classifier system.

The goal in designing any ensemble is to reduce its error relative to the errors of the individual base classifiers. After all, why would an ensemble be an effective learner if it can perform no better than some base classifier. It has been shown that one way to improve the performance of an ensemble is to ensure that the underlying base classifiers are diverse [60]. By diversity, we are referring to some measure of how often each base classifier, $\varsigma_i(x)$, differs from the prediction of the ensemble, $C(x)$. It has been shown that E, the error of an ensemble, is given by the equation

$$E = \bar{E} - \bar{A}, \qquad \text{(Eq.6.0.1)}$$

where $\bar{E}$ is the mean error of the base classifiers and $\bar{A}$ is the mean of the diversity of the classifiers [34]. From Eq.6.0.1, we see that by increasing the diversity of the base classifiers while maintaining their mean error, the accuracy of the ensemble can be improved. We now discuss two approaches to ensemble construction suggested in the literature to improve the diversity of ensemble classifiers: Bootstrap Aggregation, commonly known as bagging [7], and Boosting [20, 19].

The Bootstrap Aggregation for classification tree is summarized as follows [7]:

1. A classifier is constructed from the learning set using k-fold cross-validation.
2. A bootstrap sample ℓ_B is selected from ℓ, and a tree is grown using ℓ_B. This is repeated p times, obtaining tree classiers $\phi_1(x), \ldots, \phi_p(x)$.
3. If $(y_n; x_n) \in \tau$, then the predicted class of x_n is the class with the majority in $\phi_1(x), \ldots, \phi_p(x)$. If there is a tie, the estimated class is the one with the fewest class label.
4. The random partition of the data into ℓ and τ is repeated q times.

We used a variation of this algorithm in our work.

Boosting is an algorithm for constructing a "strong" classier as linear combination of "simple" or "weak" classifiers. Boosting calls a weak base classifier repeatedly in a series of rounds $(r_1, \ldots, r_n)$. The algorithm maintains a set of weights for the training set. The weights are initially set to some fixed value. After each round, the weights of incorrectly classied examples are increased so that the base classifier can focus on the hard examples in the training set. The Boosting algorithm is an adaptive algorithm. It reduces the training error and it is sensitive to noise and over-fitting. We did not use boosting in our work.

We have emphasized the importance of diversity in the effectiveness of an ensemble. In our work, each base classifier is constructed using a set of variations of the

examples. The more diversity in the variations of the examples, the more likely it is to have a better approximation of the true hypothesis. In order to further our goal of diversity, we employed two approaches:

1. Ensembles were constructed from subspaces from biologically-derived gene sets. We theorize that this furthers the goal of diverse variations because the different gene sets are responsible for different biological processes.
2. Ensembles were also constructed using bootstrap aggregation (Bagging). Diversity in the variations is obtained by random sampling of the examples.

6.1 Ensembles Using Biologically-derived Subspaces

Ensemble classifiers used in the classification of cancers are often analyzed on the basis of how sound they are from a statistical formulation standpoint. Very little, if any, attention is paid to the actual biology that undergird them. The use of biologically-derived gene sets is an important area of exploration because of the potential for this research to discover vital functional information about genes and genes interact and work together. While high classification accuracy is a very important goal in cancer classification, scientists involved in genetic research are equally, if not more, interested in any important biological information that the classification model may reveal. The information gleaned from the model can further their understanding of the causation and association of genes with various kinds of cancers.

One issue investigated in this work was whether further efficiencies could be gained by restricting the genes on which a classification model was trained to only those genes that contribute to some biological process. By biological process, we are referring to any clearly defined series of events or molecular functions that has a beginning and

an end. Cancer invariably leads to a set of observable characteristics of an individual resulting from the interaction of an individual's genotype with the environment. We theorize that because biological process modifications occur in cancer tissues, by mining gene expression correlations of genes in those tissues, a more powerful predictive model for cancers affecting those tissues can be built.

In restricting the feature space to biologically-derived gene sets, two kinds of gene sets were used. Details of the gene sets used in our work are give in Table 6.1 and Table 6.2. The second column in the tables represents the total number of genes in all the gene sets for the specified data set. The third column represents the average number of genes per gene set for the data sets. The fourth column represents the number of genes in the smallest gene set for the data sets, while the fifth column represents the size of the largest gene set. The sixth column represents the total number of genes in the gene sets for the specified data set. The gene sets are:

1. BP gene sets: These are gene sets grouped by biological processes from the Gene Ontology (GO) Project database.

Table 6.1: Summary Statistics on BP Gene Sets

data set	# of gene sets	Avg # of genes/set	Min # of genes in set	Max # of genes in set	Total # of unique genes
BrainTumor2	824	70.396	2	1309	4704
9Tumors	824	53.319	1	1007	3503
ProstateTumor	824	70.396	2	1309	4704

2. OS gene sets: These are oncogenic signatures from the Gene Expression Omnibus (GEO) data sets.

Table 6.2: Summary Statistics on OS Gene Sets

data set	# of gene sets	Avg # of genes/set	Min # of genes in set	Max # of genes in set	Total # of unique genes
BrainTumor2	189	115.110	10	292	6771
9Tumors	189	80.487	7	224	4392
ProstateTumor	189	115.110	10	292	6771

Gene sets based on molecular functions and cellular components were also considered but they performed worse than those grouped by biological processes and oncogenic signatures.

Ensembles were constructed using subspaces derived from BP gene sets. The Gene Ontology Project (GO) [26] has a publicly available database which is a rich source for structured, controlled vocabularies and classifications that cover several areas of molecular and cellular biology. The BP gene sets used in our work were obtained from the GO database.

Ensembles were also constructed using subspaces derived from OS gene sets. The gene sets are organized on the basis of their oncogenic signatures. The gene sets represent signatures of cellular pathways that are usually dis-regulated in cancer. They are mostly derived from from microarray gene expression data. These gene sets were obtained from the Gene Expression Omnibus (GEO) [15]. GEO contains gene expression profiles derived from previous experiments involving genes known to be cancer-causing.

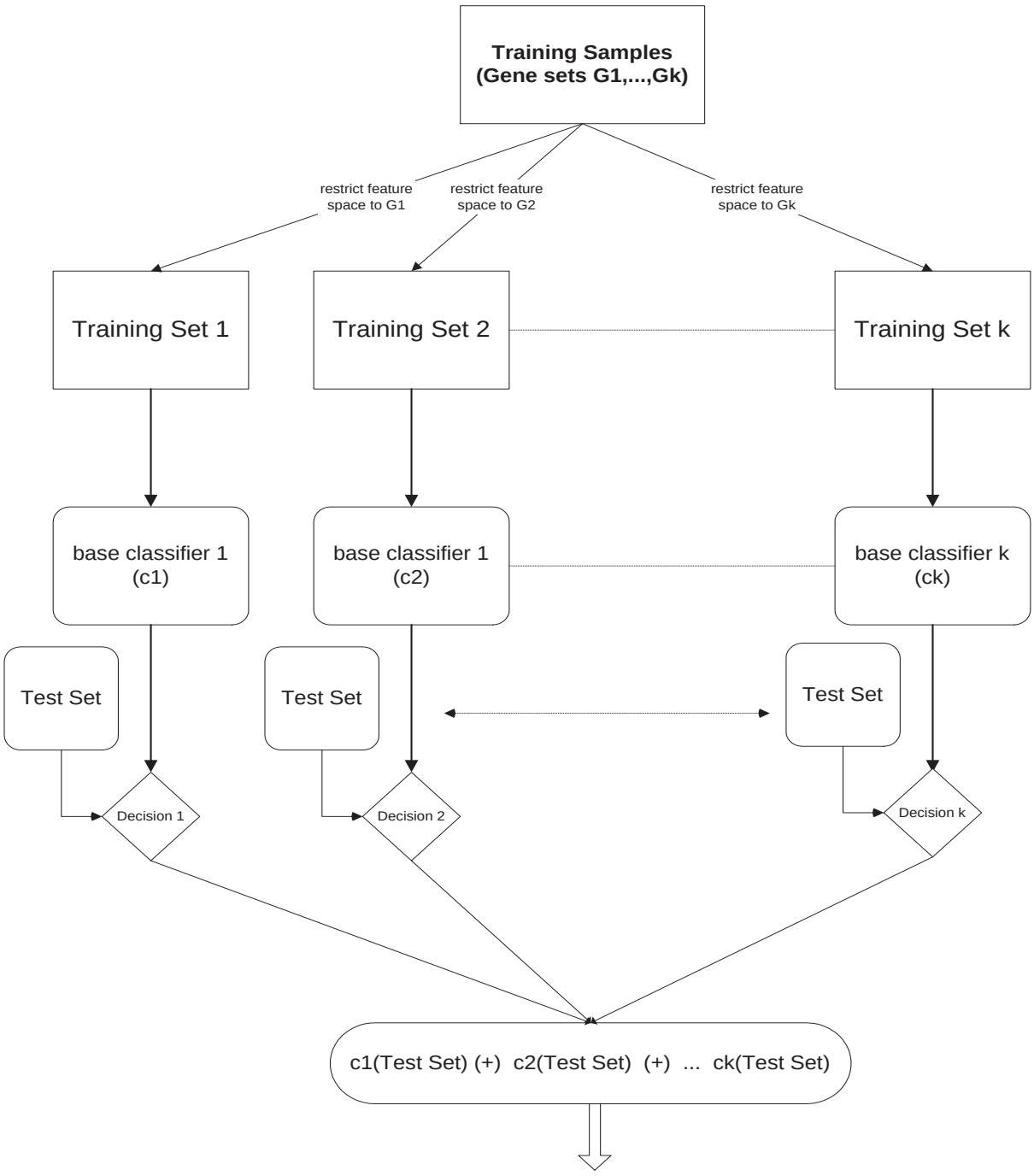

Figure 6.1: Ensemble Using Subspaces Based on Gene Sets

In order to build an ensemble using a subspace derived from biologically-relevant gene sets, the following steps (as depicted in Figure 6.1) are performed:

1. Consider gene sets G_i, $i = 1, \ldots, k$ that are known to be biologically-derived; that is, each G_i is associated with some biologically-relevant attribute. Preprocess the data by removing each gene g not in any of the gene sets from the data: remove any $g \notin \bigcup_{i=1}^{k} \{G_i\}$.

2. Train k base classifiers. The j^{th} base classifier is trained using the training samples and genes in the j^{th} gene set, G_j: each classifier is trained using a subspace constituted by using the genes in G_j, one of the partitions formed after the removal of genes not in any of the gene sets.

3. To classify a new sample, the predictions from each base classifier are pooled together and a final decision is determined by committee vote. When there are multiple classes (for example, disease conditions 1, 2, etc), the predicted classification of the new sample is classification with the highest frequency from the base classifiers.

6.2 Ensembles Using Bootstrap Aggregation

To create an ensemble using bootstrap aggregation, follow these steps as depicted in Figure 6.2:

1. Preprocess the data using the Kruskal-Wallis or ratio of between group variance to within group variance and select top-ranked genes that give the best classification result from either algorithm.

2. Randomly permute the samples.

3. Randomly partition the data set into a test set τ and a learning set ℓ.

4. The training set ℓ is repeatedly sampled to select p%$\times$ N samples (with repetition allowed), where N is the size of the training set to construct bootstrap samples ℓ_B.

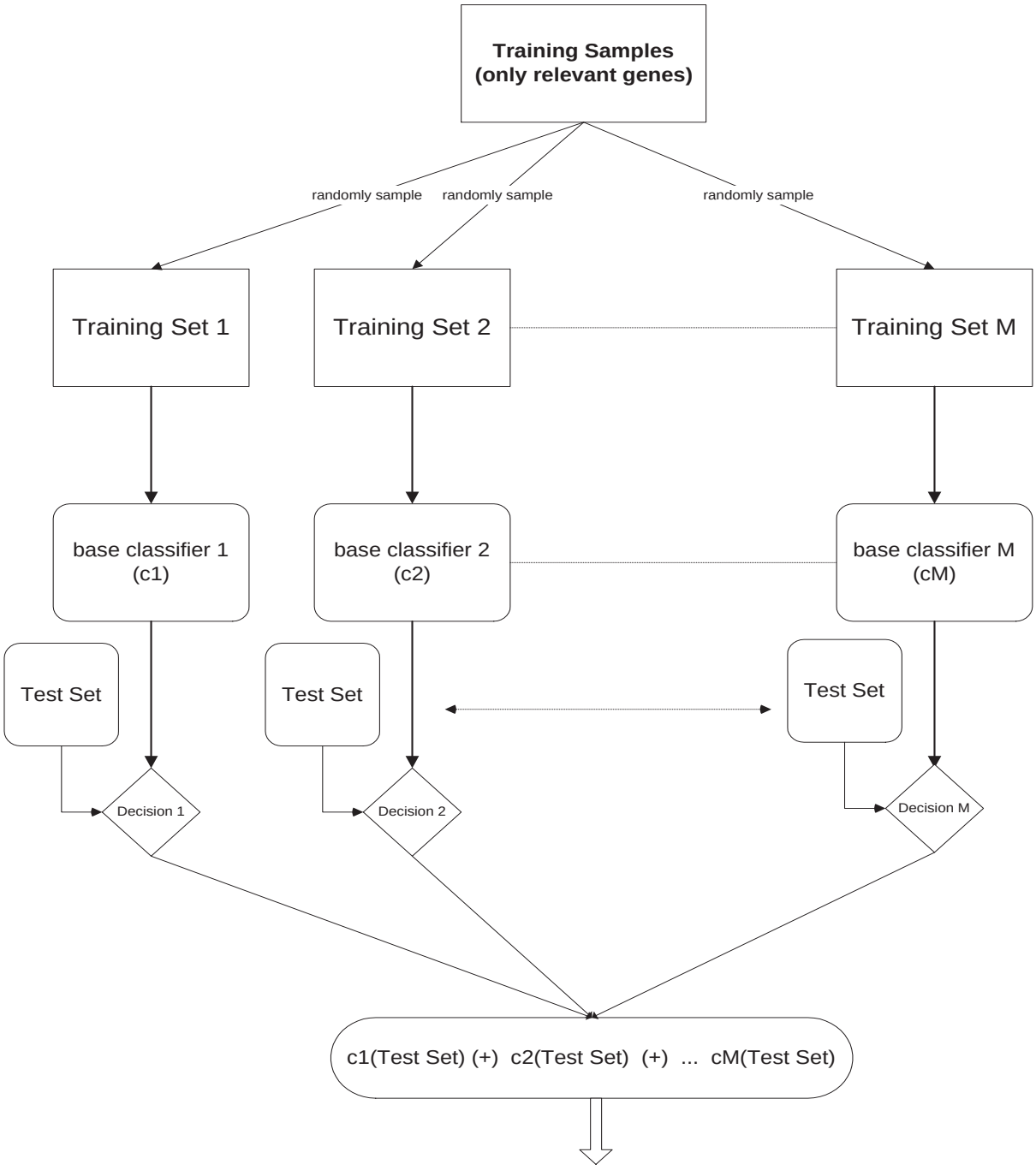

Figure 6.2: Bagged (Boostrapped) Classifiers

5. An $\ell 1$-regularized logistic regression model is constructed from the bootstrap sample ℓ_B while searching for the "best"-performing parameter ($\lambda \in \{1^t\}$, $t = -1, 0, 1, \ldots, 4$) on ℓ_B.

6. Steps 7-8 are repeated until M base classifiers are built.

7. The M base classifiers are then applied to the test set, τ and committee vote is used to determine the classification made by the ensemble. In order to classify a new sample, the predictions from each base classifier are pooled together and a final decision is determined by committee vote.When there are multiple classes (for example, disease conditions 1, 2, etc), the predicted classification of the new sample is classification with the highest frequency from the base classifiers.

6.3 Results

To evaluate each of these approaches, 10-fold cross-validation was used. The results are presented in the charts below. The feature space was initially partitioned using the gene sets in Table 3.1. We then constructed ensembles of $\ell 1$-regularized logistic regression models using subspaces from biologically-derived gene sets. The BrainTumor2, 9Tumors and ProstateTumor data sets were used to construct ensembles in order to observe the effect of using gene sets,

For comparison purposes, we also report results from the ensembles constructed using gene set-based subspaces along with results obtained by using a single $\ell 1$-regularized logistic regression model and the same feature space.

The results obtained from partitioning the feature space using genes associated with biological processes (BP) and oncogenic signatures (OS) are reported in Table 6.3 and Table 6.4, respectively.

The second columns in the tables show the accuracy of the ensemble. The third columns show the *number of genes in the ensemble classifier / the number of genes in the data set*. The fourth columns are the baseline cases; that is, the accuracies

obtained by using a single $\ell 1$-regularized logistic regression model on the same feature space (restricting the genes to those in each gene set). The fifth columns show *number of genes in the model / the number of genes in the data set* when using a single $\ell 1$-regularized logistic regression model.

Table 6.3: Ensembles Constructed Using BP Gene Sets vs Single Classifier

Dataset	L1LRM Ensemble BP Genes Subspaces	# of Effective Genes	Single L1LRM All BP Genes	# of Effective Genes
BrainTumor2	76.0	216/4705	76.0	206/4705
9Tumors	78.3	564/3504	76.7	216/3504
ProstateTumor	100.0	26/4705	100.0	10/4705

Table 6.4: Ensembles Constructed Using OS Gene Sets vs Single Classifier

Dataset	L1LRM Ensemble OS Genes Subspaces	Effective # of Genes	L1LRM All OS Genes	Effective # of Genes
BrainTumor2	78.0	256/6772	76.0	234/6772
9Tumors	80.0	838/7130	76.7	542/7130
ProstateTumor	100.0	81/6772	100.0	9/6772

Bagged $\ell 1$-regularized logistic regression models on the data sets in Table 3.2 were constructed. For comparison purposes, we report the results along with those obtained by using one $\ell 1$-regularized logistic regression model on the feature space (after preprocessing) as the baseline. The results are shown in Table 6.5.

The second column in the table shows the average accuracy$\pm$standard deviation of the bagged $\ell 1$-regularized logistic regression models after 20 repetitions of 10-fold cross-validation. The third column is the baseline case; that is, the average accuracy$\pm$standard deviation obtained after 20 repetitions of 10-fold cross-validation when using a single $\ell 1$-regularized logistic regression model on the same data set.

Table 6.5: Ensemble Constructed Using Bagging vs Single Classifier

Dataset	Bagged L1LR Models	Single L1LR Model
BrainTumor2	42.7±3.5	36.2±4.81
9Tumors	68.3±4.1	63.0±5.09
ProstateTumor	93.9±0.9	92±1.5

6.4 Observations

Across all the data sets, on average, bagged $\ell 1$-regularized regression models outperformed single $\ell 1$-regularized logistic models. In particular, with the 9Tumors data set, the average classification accuracy of the bagged $\ell 1$-regularized models was 68.3±4.1% while the classification accuracy achieved without bagging was 63.0%, an average gain of 10±4.1%. There is also difference in the average accuracy between the ensemble and the single classifier of around 6% and 1% on the BrainTumor2 and ProstateTumor data sets, respectively. This suggests that bagging increases classification accuracies on data sets containing both small and large number of examples. This is attributable to the diversity in the variations of the base classifiers since the same data sets are being used. The only thing that is different is that an ensemble of diverse base classifiers are being used rather than a single classifier.

We also observed that ensembles constructed from subspaces constituted by partitioning the feature space using gene sets performed as well as or better than those based on the same features using only one $\ell 1$-regularized logistic regression model. This is true for both gene sets based on biological processes and oncogenic signatures. The ensembles constructed from subspaces derived from gene sets outperformed a single $\ell 1$-regularized logistic regression model on both BrainTumor2 and 9Tumors data

sets. The ensembles matched the performance of a single $\ell 1$-regularized logistic regression models on the ProstateTumor data set for both biological processes and oncogenic signatures gene sets. From Table 6.3 and Table 6.4, we see that the there is a slight increase in the number of effective genes in the ensemble as compared to the single classifier but this increase also leads to an increase in classification accuracy. So there is a trade-off: there is an increase in accuracy with a marginal increase in the number of genes in the feature space. Again, this increase in accuracy is attributable to the use of diverse base classifiers since the same data sets are being used and the only difference is that an ensemble is used in one case while a single classifier is used in another.

The classification accuracy increases with bagged $\ell 1$-regularized logistic regression models over the use of a single $\ell 1$-regularized logistic regression model as shown in Table 6.5. Also, constructing ensembles from subspaces derived from gene sets lead to better performance than not partitioning the feature space as the results in Table 6.3 and Table 6.4 show. These results suggest that it is worth exploring a hybrid approach in which ensembles are constructed by bagging $\ell 1$-regularized logistic regression models constructed from subspaces derived from gene sets.

Chapter 7
Conclusion

In the introduction, we expressed the hope that this work would lead to greater insights into how the predictive power of ensemble classifiers for cancers can be improved. By employing novel gene selection techniques and choosing underlying base classifiers with requisite properties, the performance of an ensemble can be enhanced. We explored ways in which the base classifiers can be built and trained to improve the performance of an ensemble classifier. We now provide a summary of the insights gained and describe the progress made towards answering the questions that were explored. We also suggest some potentially promising future research directions that could provide additional ways of enhancing the performance of ensemble classifiers.

The aim of this work has been to explore ways in which the predictive power of an ensemble can be enhanced for cancer classification. Throughout the literature, there has been considerable interest in the ways in which multiple classifiers can be combined to improve their predictive power. We set out to answer four key questions:

1. How effective are logistic regression models [33] with $\ell 1$-regularization [5] in the removal of redundant (ineffective) genes in micro-array gene expression data?
2. Does the flexibility (complexity) of the base classifier lead to an improvement in the classification accuracy of the ensemble?
3. Does the use of a regularized base classifier enhance the accuracy of the ensemble?
4. Can the performance of an ensemble classifier for cancer that uses microarray data be improved by using subspaces based on biologically-derived gene sets?

7.1 Contributions

To gain insights into these questions, we focused on how to engineer the ensemble classifier by mitigating some of the inherent challenges in classifying cancers using microarray data. These are the contributions of our work.

- L1-regularized logistic regression model can be used to perform implicit gene selection in microarray cancer data.
- An ensemble used for cancer classification does not need to have a flexible classifier to achieve good classification accuracy.
- Regularized classifiers serve better as base classifier for a ensemble.
- Biologically-derived gene sets have inherent variations thus making them suitable for ensuring diversity when they are used in ensembles.

7.2 Summary

In Chapter 1, the importance of the use of an ensemble in classifying cancers using microarray data was discussed. We discussed how advances in biomedical technology have made gene expression data easily available for use by ensemble classifiers. We also gave the motivation for this work. The motivations include, early diagnosis of cancers, improved prediction of responses to treatment, the development of customized treatment and the potential for better prognosis. We indicated that while phenotypical information is very useful, and even very reliable, in detecting and diagnosing certain kinds of cancers, the availability of genotypical information provides a tremendous amount of promise for detecting cancers. A summary of the key questions

that this work examines and what should be done to address those questions was also highlighted.

In Chapter 2, related works in cancer classification using gene expression data were surveyed. We discussed some approaches that have included traditional methods such as artificial neural networks, nearest-neighbor-based methods, decision trees and support vector machines (SVM). Research involving the use of random forest and an ensemble of decision trees in classification were also discussed. Additionally, we gave two popular strategies of ensemble constructions, bootstrap aggregation and adaptive boosting.

In Chapter 3, the problem of cancer classification using microarray gene expression data was formally defined. We described the nature of the data used in our research: a matrix whose rows represent gene expression profiles and whose columns represent gene expressions of each gene across the samples. We also outlined some of the technological and algorithmic issues that make this a challenging problem. One challenge discussed was that microarray expression data are ultra high-dimensional and potentially contains noise. Another challenge discussed was that some of the genes in the data sets are not good predictors for cancers. The issue of some genes being correlates and the need for the elimination of redundancies was also raised as a challenge. We also enumerated the problems we sought to solve. The solutions to these problems address the challenges and issues discussed in the chapter. The problems identified are gene selection, over-fitting, the use of ensemble classifiers, determining a good base classifier and constructing ensembles. Finally, we discussed the experiment setting, source of data used in this work and the use of k-fold cross-validation to evaluate the approaches that are proposed.

In Chapter 4, we presented our proposed two-stage gene selection algorithm. The first stage of the algorithm involves ranking genes using two ranking schemes. One approach uses the Kruskal-Wallis non-parametric one way analysis of variance test to rank genes. The second approach uses the ratio of between group variance to within group variance (F-score) to rank the genes. The advantages and limitations of each of the ranking schemes were discussed. When the first stage is complete, top-ranked genes are chosen after using the performance of an SVM classifier on the data as an evaluation metric. The ranking scheme yielding better accuracy is used in the selection of the top-ranked genes. In the second stage, the data set is restricted to the top-ranked gene and the $\ell 1$-logistic regression model is applied to the modified data to remove ineffective/redundant genes. Empirical results from the use of the two-stage gene selection approach were reported. These results are compared with those obtained by using no gene selection at all and those involving only the use of the first stage of the algorithm.

In Chapter 5, the importance of the choice of a base classifier on the performance of an ensemble was emphasized. The choice of a base classifier is important because an ensemble classifier pools the decisions of individually trained base classifiers in order to make its own decision. Very often, optimization with respect to a small set of examples does not lead to the true hypothesis. However, combining a collection of optimal solutions, each based on a variation of the examples, can better approximate the true hypothesis. The two problems explored in the chapter are the impact of classifier flexibility and regularization on the performance of an ensemble of classifiers. These based classifiers were used in an empirical study: SVM, decision tree, and $\ell 1$-regularized linear regression. SVM was used to study the impact of linearity and

$\ell 2$-regularization. Decision tree was used as a base classifier to study the effect of a flexible base classifier on the performance of an ensemble. $\ell 1$-regularized logistic regression model was used to study the impact of linearity and $\ell 1$-regularization. Results from empirical studies are presented along with observations and conclusions drawn.

In Chapter 6, two general approaches for the construction of ensembles, bootstrap aggregation and adaptive boosting were discussed. Both their advantages and limitations are described. In this chapter, we also discussed two approaches that were used to construct ensembles for the classification of cancer. The use of bagged $\ell 1$-regularized logistic regression models to classify cancer was discussed. A second approach that uses subspaces from biologically-derived gene sets was also discussed. Empirical results from the use of the bagged ensembles and ensembles constructed from subspaces using gene sets related to biological processes and oncogenic signatures were reported. These results were analyzed and observations were made.

7.3 Future Work

All the approaches to enhancing the performance of ensembles considered in this work have involved either sampling in the feature (gene) space or the sample space during the training of base classifiers. We did not consider any approach involving a hybrid of both of these approaches. A direction that could be explored is a hybrid approach that concurrently samples both spaces during the training of the base classifiers. It goes without saying that this would require increased computing resources during the training phase. Since the building of base classifiers from sampling both the sample space and the feature space concurrently is independent, significant speedup can be achieved through parallelization. This approach would potentially select the relevant

features while at the same time deal with the over-fitting concerns. If this approach proves successful, it would contribute tremendously to the classification of certain rare kinds of cancers and other genetic diseases for which there are very few samples.

In combining the base classifiers to determine the overall decision of the ensembles used in this work, we used committee vote by the base classifiers with a majority voting rule for the ensemble to arrive at a decision. A future research direction would be constructing ensembles that make decisions based on a weighted combination of the classification functions of the base classifiers. These functions could be weighted based on the classification accuracy of the base classifiers on the training samples or some complimentary learning algorithm could be used to learn how the votes should be weighted.

Finally, another direction to explore in the construction of an ensemble classifier for microarray data is some variant of the generalized additive model [27]. This approach would be iterative and the classification function would be dynamic. Given the statistical formulation of an additive model, in theory, there would be the potential to minimize classification error with every iteration.

Further research in these directions is likely going to reveal additional attributes of a base classifier that would contribute to the enhancement of the performance of an ensemble. Additionally, the use of various techniques to weight the base classifiers could improve the performance of ensembles. A generalized additive model is worth exploring since it would lead to an adaptive model.

Bibliography

[1] Subhash C Bagui, Sikha Bagui, Kuhu Pal, and Nikhil R Pal, *Breast cancer detection using rank nearest neighbor classification rules*, Pattern recognition **36** (2003), no. 1, 25–34.

[2] Gábor Balázsi, Krin A Kay, Albert-László Barabási, and Zoltán N Oltvai, *Spurious spatial periodicity of co-expression in microarray data due to printing design*, Nucleic acids research **31** (2003), no. 15, 4425–4433.

[3] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, et al., *A comparison of decision tree ensemble creation techniques*, IEEE transactions on pattern analysis and machine intelligence **29** (2007), no. 1, 173.

[4] Richard Bellman, *Dynamic programming and the smoothing problem*, Management Science **3** (1956), no. 1, 111–113.

[5] Peter J Bickel, Bo Li, Alexandre B Tsybakov, Sara A van de Geer, Bin Yu, Teófilo Valdés, Carlos Rivero, Jianqing Fan, and Aad van der Vaart, *Regularization in statistics*, Test **15** (2006), no. 2, 271–344.

[6] L. Breiman, *Random forests*, Machine learning **45** (2001), no. 1, 5–32.

[7] Leo Breiman, *Bagging predictors*, Machine learning **24** (1996), no. 2, 123–140.

[8] C.J.C. Burges, *A tutorial on support vector machines for pattern recognition*, Data mining and knowledge discovery **2** (1998), no. 2, 121–167.

[9] Yvonne Chan and Roy P Walmsley, *Learning and understanding the kruskal-wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups*, Physical therapy **77** (1997), no. 12, 1755–1761.

[10] F. Collins, D. Galas, et al., *A new five-year plan for the us human genome project*, SCIENCE-NEW YORK THEN WASHINGTON- **262** (1993), 43–43.

[11] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning **20** (1995), no. 3, 273–297.

[12] Ramón Diaz-Uriarte, *Genesrf and varselrf: a web-based tool and r package for gene selection and classification using random forest*, BMC bioinformatics **8** (2007), no. 1, 328.

[13] H. Drucker, D. Wu, and V.N. Vapnik, *Support vector machines for spam categorization*, Neural Networks, IEEE Transactions on **10** (1999), no. 5, 1048–1054.

[14] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed, *Comparison of discrimination methods for the classification of tumors using gene expression data*, Journal of the American statistical association **97** (2002), no. 457, 77–87.

[15] Ron Edgar, Michael Domrachev, and Alex E Lash, *Gene expression omnibus: Ncbi gene expression and hybridization array data repository*, Nucleic acids research **30** (2002), no. 1, 207–210.

[16] Henry A Erlich, *Polymerase chain reaction*, Journal of clinical immunology **9** (1989), no. 6, 437–447.

[17] Floriana Esposito, Donato Malerba, Giovanni Semeraro, and J Kay, *A comparative analysis of methods for pruning decision trees*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **19** (1997), no. 5, 476–491.

[18] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, *Liblinear: A library for large linear classification*, The Journal of Machine Learning Research **9** (2008), 1871–1874.

[19] Yoav Freund, Robert Schapire, and N Abe, *A short introduction to boosting*, Journal-Japanese Society For Artificial Intelligence **14** (1999), no. 771-780, 1612.

[20] Yoav Freund and Robert E Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of computer and system sciences **55** (1997), no. 1, 119–139.

[21] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er, *Using bayesian networks to analyze expression data*, Journal of computational biology **7** (2000), no. 3-4, 601–620.

[22] Seymour Geisser, *Predictive inference: an introduction*, vol. 55, CRC Press, 1993.

[23] Jean Dickinson Gibbons and Subhabrata Chakraborti, *Nonparametric statistical inference*, vol. 168, CRC press, 2003.

[24] C Gini, *Concentration and dependency ratios*, Rivista di Politica Economica **87** (1997), 769–792.

[25] M.A. Hall, *Correlation-based feature selection for machine learning*, Ph.D. thesis, The University of Waikato, 1999.

[26] MA Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, et al., *The gene ontology (go) database and informatics resource*, Nucleic acids research **32** (2004), no. Database issue, D258.

[27] Trevor Hastie and Robert Tibshirani, *Generalized additive models*, Statistical science (1986), 297–310.

[28] David W Hosmer and Stanley Lemeshow, *Applied logistic regression*, vol. 354, Wiley-Interscience, 2004.

[29] Anil K Jain, M Narasimha Murty, and Patrick J Flynn, *Data clustering: a review*, ACM computing surveys (CSUR) **31** (1999), no. 3, 264–323.

[30] Finn V Jensen, *An introduction to bayesian networks*, vol. 74, UCL press London, 1996.

[31] Ian T Jolliffe, *Principal component analysis*, vol. 487, Springer-Verlag New York, 1986.

[32] Javed Khan, Jun S Wei, Markus Ringner, Lao H Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R Antonescu, Carsten Peterson, et al., *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*, Nature medicine **7** (2001), no. 6, 673–679.

[33] DG Kleinbaum, *Logistic regression: a self-learning text*, Statistics (1994).

[34] A. Krogh, J. Vedelsby, et al., *Neural network ensembles, cross validation, and active learning*, Advances in neural information processing systems (1995), 231–238.

[35] Harri Lappalainen and James W Miskin, *Ensemble learning*, Advances in Independent Component Analysis, Springer, 2000, pp. 75–92.

[36] Leping Li, Clarice R Weinberg, Thomas A Darden, and Lee G Pedersen, *Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method*, Bioinformatics **17** (2001), no. 12, 1131–1142.

[37] Y. Li, Y.Z. Cal, R.P. Yin, and X.M. Xu, *Fault diagnosis based on support vector machine ensemble*, Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, vol. 6, IEEE, 2005, pp. 3309–3314.

[38] Arthur Liberzon, Aravind Subramanian, Reid Pinchback, Helga Thorvaldsdóttir, Pablo Tamayo, and Jill P Mesirov, *Molecular signatures database (msigdb) 3.0*, Bioinformatics **27** (2011), no. 12, 1739–1740.

[39] H. Liu, L. Liu, and H. Zhang, *Ensemble gene selection by grouping for microarray data classification*, Journal of biomedical informatics **43** (2010), no. 1, 81–87.

[40] Richard Maclin and David Opitz, *Popular ensemble methods: An empirical study*, arXiv preprint arXiv:1106.0257 (2011).

[41] Debahuti Mishra and Barnali Sahu, *Feature selection for cancer classification: A signal-to-noise ratio approach*, International Journal of Scientific & Engineering Research **2** (2011).

[42] T. Mitchell, *Decision tree learning*, Machine learning **414** (1997).

[43] Barbara J Norton and Michael J Strube, *Guide for the interpretation of one-way analysis of variance*, Physical Therapy **65** (1985), no. 12, 1888–1896.

[44] Catherine L Nutt, DR Mani, Rebecca A Betensky, Pablo Tamayo, J Gregory Cairncross, Christine Ladd, Ute Pohl, Christian Hartmann, Margaret E McLaughlin, Tracy T Batchelor, et al., *Gene expression-based classification of malignant gliomas correlates better with survival than histological classification*, Cancer Research **63** (2003), no. 7, 1602–1607.

[45] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., *Scikit-learn: Machine learning in python*, The Journal of Machine Learning Research **12** (2011), 2825–2830.

[46] Hanchuan Peng, Fuhui Long, and Chris Ding, *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **27** (2005), no. 8, 1226–1238.

[47] Scott L Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M Sturla, Michael Angelo, Margaret E McLaughlin, John YH Kim, Liliana C Goumnerova, Peter M Black, Ching Lau, et al., *Prediction of central nervous system embryonal tumour outcome based on gene expression*, Nature **415** (2002), no. 6870, 436–442.

[48] Lior Rokach and Oded Maimon, *Decision trees*, Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 165–192.

[49] John Shawe-Taylor and Nello Cristianini, *Kernel methods for pattern analysis*, Cambridge university press, 2004.

[50] H.B. Shen and K.C. Chou, *Ensemble classifier for protein fold pattern recognition*, Bioinformatics **22** (2006), no. 14, 1717–1722.

[51] Dinesh Singh, Phillip G Febbo, Kenneth Ross, Donald G Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A Renshaw, Anthony V D'Amico, Jerome P Richie, et al., *Gene expression correlates of clinical prostate cancer behavior*, Cancer cell **1** (2002), no. 2, 203–209.

[52] Therese Sørlie, Charles M Perou, Robert Tibshirani, Turid Aas, Stephanie Geisler, Hilde Johnsen, Trevor Hastie, Michael B Eisen, Matt van de Rijn, Stefanie S Jeffrey, et al., *Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications*, Proceedings of the National Academy of Sciences **98** (2001), no. 19, 10869–10874.

[53] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, *A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis*, Bioinformatics **21** (2005), no. 5, 631–643.

[54] Jane E Staunton, Donna K Slonim, Hilary A Coller, Pablo Tamayo, Michael J Angelo, Johnny Park, Uwe Scherf, Jae K Lee, William O Reinhold, John N Weinstein, et al., *Chemosensitivity prediction by transcriptional profiling*, Proceedings of the National Academy of Sciences **98** (2001), no. 19, 10787–10792.

[55] Andrew I Su, John B Welsh, Lisa M Sapinoso, Suzanne G Kern, Petre Dimitrov, Hilmar Lapp, Peter G Schultz, Steven M Powell, Christopher A Moskaluk, Henry F Frierson Jr, et al., *Molecular classification of human carcinomas by use of gene expression signatures*, Cancer research **61** (2001), no. 20, 7388–7393.

[56] David MJ Tax and Robert PW Duin, *Using two-class classifiers for multiclass classification*, Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2, IEEE, 2002, pp. 124–127.

[57] Laura Toloşi and Thomas Lengauer, *Classification with correlated features: unreliability of feature ranking and solutions*, Bioinformatics **27** (2011), no. 14, 1986–1994.

[58] Thanh N Tran, Ron Wehrens, and Lutgarde Buydens, *Knn-kernel density-based clustering for high-dimensional multivariate data*, Computational Statistics & Data Analysis **51** (2006), no. 2, 513–525.

[59] I. Tsamardinos, L.E. Brown, and C.F. Aliferis, *The max-min hill-climbing bayesian network structure learning algorithm*, Machine learning **65** (2006), no. 1, 31–78.

[60] Kagan Tumer and Joydeep Ghosh, *Error correlation and error reduction in ensemble classifiers*, Connection science **8** (1996), no. 3-4, 385–404.

[61] Giorgio Valentini, Marco Muselli, and Francesca Ruffino, *Cancer recognition with bagged ensembles of support vector machines*, Neurocomputing **56** (2004), 461–466.

[62] V. Vapnik, *The nature of statistical learning theory*, springer, 1999.

[63] IB Vapnyarskii, *Lagrange multipliers*, Encyclopaedia of Mathematics. Springer, Heidelberg (2001).

[64] M. Wang, Z. Chen, and S. Cloutier, *A hybrid bayesian network learning method for constructing gene networks*, Computational Biology and Chemistry **31** (2007), no. 5-6, 361–372.

[65] Qing Wang and Liang Zhang, *Ensemble learning based on multi-task class labels*, Advances in Knowledge Discovery and Data Mining, Springer, 2010, pp. 464–475.

[66] S. Wang, A. Mathew, Y. Chen, L. Xi, L. Ma, and J. Lee, *Empirical analysis of support vector machine ensemble classifiers*, Expert Systems with Applications **36** (2009), no. 3, 6466–6476.

[67] Jason Weston and Chris Watkins, *Support vector machines for multi-class pattern recognition*, Proceedings of the seventh European symposium on artificial neural networks, vol. 4, 1999, pp. 219–224.

[68] Haiyuan Yu, Katherine Nguyen, Tom Royce, Jiang Qian, Kenneth Nelson, Michael Snyder, and Mark Gerstein, *Positional artifacts in microarrays: experimental verification and construction of cop, an automated detection tool*, Nucleic acids research **35** (2007), no. 2, e8–e8.

The Vita

William Evans Duncan received his bachelor's degree in computer science at the University of the South, Sewanee, Tennessee in May 2000. He graduated Phi Beta Kappa summa cum laude. Thereafter, he studied at the University of Tennessee at Knoxville, where he received the master's degree in computer science in May 2003. He is currently a candidate for the degree of Doctor of Philosophy in Computer Science, which will be awarded in August 2013.