# Stony Brook University

# Ensemble Methods for Classification with Applications to Genomics

A Dissertation Presented

by

**Melissa Jane Fazzari**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**

Stony Brook University

**August 2007**

**Stony Brook University**

The Graduate School

**Melissa Jane Fazzari**

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

**Hongshik Ahn - Dissertation Advisor**
Associate Professor
Applied Mathematics and Statistics Department

**Stephen Finch- Chairperson of Defense**
Professor
Applied Mathematics and Statistics Department

**Wei Zhu**
Associate Professor
Applied Mathematics and Statistics Department

**John M. Greally**
Associate Professor
Departments of Medicine and Molecular Genetics
Albert Einstein College of Medicine

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

**Ensemble Methods for Classification with Applications to Genomics**

by

**Melissa Jane Fazzari**

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**
Stony Brook University

**2007**

The wealth of data generated in genomics research gives the promise of individualized medicine, treatment, and risk assessment. Proper classification of individuals or samples into different disease states or risk groups has become an increasingly important step in this process. There are a multitude of classifiers available, from standard statistical models to more current machine learning classification. This dissertation examines commonly used classifiers in the genomics setting, decomposing error into bias and variance components, and evaluates the effects of high dimension. By understanding how errors are made, we may begin to understand and improve upon the model building process. With this goal in mind, classifier ensembles are explored. Ensembles are created using a diverse set of stand-alone classifiers and several aggregation methods are evaluated. In addition, two novel ensemble methods are explored. The first creates partitions in the feature space and builds classifiers within each subspace. The second allows the ensemble combining weights to vary, depending on the location of each test point for which prediction is sought. Using estimates of bias and variance, the most stable classifiers receive the most weight. Finally, variable importance is briefly examined, comparing simple univariate ranks with multivariable summaries. Several real-world, high dimensional data sets are explored and serve as representative examples from the genomics domain.

*To Pete and Julia.  With much love and gratitude for the life we share.*

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

I would like to thank my advisor, Dr. Honshik Ahn and committee members Dr. Wei Zhu and Dr. Stephen Finch. I very much appreciate your time, support, and efforts during all four of my years at Stony Brook.

Dr. Ram Srivastav – for a wonderful linear algebra course.

Dr. John Greally – you are a valued friend and colleague. Thank you for your generosity.

I would also like to thank my entire family for their support and patience. Although it feels a bit grandiose writing this down, there are few times in life when you can publicly thank the people who have given you so much. With that…….

My mom – who taught me how to work hard and, more importantly, how to be a good mother.

My husband – I am lucky enough to be married to someone I deeply respect and admire, so he had to endure more one-sided, boring conversations about this dissertation than anyone should.

My daughter – my wonderful, sweet, loving, smart, funny girl.

And my father – who is always part of everything I do.

# Chapter 1

# Introduction

---

"The message to biologists is clear: If you want to work with microarrays, you need to find yourself one of these precious experts…."

---

The above is a quote from a featured article in Nature [63], one of the most influential and high-impact scientific publications in the world. The "precious experts" being described in this review of microarray technology are, perhaps surprisingly, statisticians.

Why are statisticians increasingly such important players in this arena? In short, the promise of genomics with respect to personalized medicine, drug discovery, and individual health management is possible only if the wealth of data coming from these technologies is properly collected, processed, and analyzed. And, at the end of the line, the translation from lab to clinic is one of the highest priorities, for both scientific and financial reasons [56, 47]. Although it may be of scientific interest to examine the heterogeneity of genomic markers in a population, the impact of this observation must also be quantified and understood with respect to clinically relevant outcomes. It is only this step that will allow an interesting observation to become something that will change the course of medicine – from diagnosis to treatment to management of disease. Successful translation relies heavily on computational, bioinformatic, and statistical

collaboration, and is often focused on what is the topic of this dissertation – building successful and generalizable classifiers.

Genomics represents a broad domain; however whether the platform is copy number, methylation, gene expression or DNA sequence features, statistical and computational analyses within this area usually follow the same general process and have the same deficiencies. First, given the current cost of technology in many applications, the sample size is typically low. This is expected to change for the better in the near future since more genomics-based exploratory studies are being included as primary or secondary goals in grants and clinical trials. Small sample size presents a challenge even for the most simple of analytic goals, because statistical separation of signal from noise is harder to detect when there are few independent observations. In addition, the use of complex models tends to be almost superfluous given the inability to detect interacting and complex relationships. Second, important features are seldom known *a priori*; they are simultaneously mined and analyzed with thousands of other potential features. Most of the collected data is likely noise, at least for the purpose for which it was mined. It is usually left to the statistician to filter out the low signals or noise and then decide how to prioritize remaining features and build the model. Third, the dimension of the problem tends to be huge. Not too many years ago, the term "large p" meant a variable list in the tens or hundreds. Today, "large p" means tens or hundreds of thousands of potentially informative and overlapping predictor variables. Overlapping features are common - gene expression levels tend to be organized within larger clusters, representing gene pathways. Sequence features such as repetitive sequences also co-localize within blocks of DNA. Given these obscure relationships, lack of *a priori* information, and small sample sizes, the model building process becomes very complex.

In this thesis, the successful building of classifiers with genomic data is examined. An in-depth study of individual classifier performance is done, and the bias-variance decomposition of each classifier is explored. Although the relative performances of many types of standard and novel classifiers have been evaluated numerous times [6], there has been little to date that compares classifiers in terms of the bias and variance breakdown. In addition, changes in bias and variance across increasing dimensions are

explored. Two classifiers with equal accuracy may have very different breakdowns with respect to error.

Ensemble methodology for combining individual classifiers is then examined, and two novel combination methods are presented. The first combination method is made up of classifiers that are based on mutually exclusive partitions of the original input feature set. The second uses diverse base classifiers, each built on the same feature set. Finally, a brief exploration of variable importance is presented. For the primary data set, a study of imprinted genes, an in-depth analysis of feature importance is presented. A univariate screening method is compared to multivariable methods in order to highlight the strengths and drawbacks of both approaches. In addition, the confidence of each prediction is established, and hard-to-fit observations are identified.

## 1.1 Dissertation Overview

### 1.1.1 Individual classifiers

There are several types of classifiers considered in this dissertation. For each, an in-depth examination is provided. This thesis illustrates, supportive of the *no free lunch theorem* [72], that there is no universally best classifier for the genomics domain, or any domain. Going further, there is likely no "best" classifier even within one dataset. This is supported by comparisons of accuracy over several data sets as well as observing the behavior of a set of classifiers within a particular data set. To gain further insight into classifier performance, classification error is broken down into bias and variance components. Given the complexities of many classifiers, examination of bias and variance is informative in understanding the effect of tuning parameters, dimension, and complexity. It has been observed, and shown by others [17, 19, 22, 37] that bias and variance for classification error do not operate as observed in the squared error setting, where there is an additive effect due to both components. In classification, there is an interaction between bias and variance such that the typical bias-variance tradeoff fails to hold. In addition, typical "high bias" estimators, can be powerful classifiers. Since the

actual prediction is not important in classification all that is required is that the prediction be on the correct side of the decision boundary. High variance estimators can be low variance classifiers because high variance in the estimation setting does not necessarily mean that the classification coming from noisy predictions is also noisy.

In addition, we examine the potential for each classifier to over-fit the training data and find that certain classifiers are much more dependent on the training set than others, thus subject to larger amounts of over-fitting.

### 1.1.2 Ensembles

In the book *The Wisdom of Crowds* [60], it is observed that the general consensus (of lay persons and experts alike) is often much more accurate than the testimony of one "expert". Similarly, in a trivia-based television show, the vote of the audience tends to have a much higher success rate in correctly answering the question posed than that of the friend that is called to be the "expert" helper. Why is this?

Predictions that are made from a consensus of different experiences, biases, and attitudes tend to be, on average, highly accurate. We can theoretically examine this to show that the majority vote of a set of diverse opinions has a higher accuracy than each individual opinion.

Using this theoretical justification, ensemble methodology is growing in popularity across many domains. In genomics, one of the most common ensemble methods used to date is Breiman's Random Forest [10]. A reason for the popularity of Random Forest is that it is easy to use, well understood, and in a convenient procedure in a popular statistical software package [44]. It also happens to be a highly consistent and accurate classifier across many applications. But ensemble methodology is just in the early stages of development, and other methods of classifier combination must be explored. Ensembles work because of both variance and bias reduction capabilities, depending on the construction. Regardless of which method is used, the success of the ensemble approach depends on the diversity of the individual members.

We may achieve diversity in multiple ways, and in this work we build ensembles using a set of distinct classifiers, relying on the differences in flexibility, loss function,

and regularization imposed by each. We then examine the simple and weighted averages of these diverse predictions, and compare the performance to the individual results.

A natural extension to the weighted average is the assignment of weights that are based on the individual test point. Global performance of a classifier in terms of accuracy and similar measures fails to take into account regions in the input space that are not well-served by this classifier. In this dissertation, a novel combining strategy is proposed, which takes local performance into account. Instead of applying the same set of classifiers to each test point, this method allows the weights to change, depending on locality. Some classifier weights may be set to zero, accounting for highly biased regions in the input space. It is hypothesized that certain classifiers may perform differently in various regions of the input space; therefore any method that takes this locality into account will offer an advantage over global forms of classification. We show this gain through simulation, with the assumption that the highly biased region is known. In reality, understanding of the input space is difficult, and the estimated weights may be unstable, and estimating them may over-fit the training data. Location of the test point is assessed by Euclidean distance, which may also be unreliable in high dimensions. In addition, a new global classifier is briefly described and examined [1, 45]. This classifier partitions the feature space randomly and fits models in each sub-region defined by the partition. A simple majority or average vote across all of the base decisions is performed to generate a final prediction.

Taking an average of the prediction scores generated from an ensemble of good stand-alone classifiers often results in stable performance, as well as providing better measures of confidence for each observation. If classifier selection bias is taken into account, simple averaging is likely more accurate and has lower variability than selection of the "best" classifier. As we will illustrate, there is no best classifier; therefore selection of one classifier on the basis of a training set may not yield the best performances when applied to another data set.

### 1.1.3 Variable importance and screening

Variable importance can mean many things, depending upon the application and goals. In genomics data, variable importance measures can simple signify which genes are representative of a subgroup of genes that are most informative. They represent one representation of a multitude of many good feature sets that can be used in a classifier.

Univariate ranking methods summarize one kind of importance. It addresses those variables that are informative when examined singularly in the model. For the most part, this importance measure appears to reflect importance on a broader scale. However, there is likely a set of features that are not important by themselves, but informative only in the presence of other features. Univariate assessments of variable importance will likely discard or down-weight these predictors.

Methods to examine variable importance in different ways are emerging. Backwards elimination using a support vector machine is able to capture which features have the highest weights and are thus retained in multiple iterations of backward selection. Random Forests, due to the use of bagging and flexible trees, is able to capture the importance of features in a multivariable fashion by assessing the impact of deletion or permutation of features on the overall accuracy [9, 10, 16]. In addition, joint effects are able to be detected given the flexible nature of the classifier. The variable importance statistics output by this procedure are extremely informative, and often yield new insights into the analysis.

In this work, we examine the variable importance measures for our primary data set: the prediction of imprinted genes. A detailed analysis gives an overview of the relationships between different sequence features, as well as which features are important in predicting imprinting class status. The stability of each observation in this data set with respect to predictions across classifiers is used to examine prediction confidence. It is expected that the larger the stability at $X=x$, the further away the observation is from the theoretical decision boundary. The grouping of observations into "hard" and "easy" cases provides more information about genes on an individual level.

## 1.2 Organization of this dissertation

This dissertation presents a detailed evaluation of many commonly applied classifiers as well as an overview of ensemble methodology, bias-variance decomposition, and variable screening and importance. The main contributions of this dissertation and results are found in Chapters 6-9 and represent methodological development in ensembles, further understanding of high dimension in classification, and a novel application of ensembles in the prediction of imprinted genes. The work is organized as follows:

- Chapter 2 gives an introduction to classification in the genomics setting and presents the several real-world applications in this domain. The primary analysis is a study of imprinted genes.
- Chapter 3 provides an extensive overview of different classifiers that are commonly used, as well as their similarities and differences.
- Chapter 4 presents the background of error decomposition in the classification setting, describing the bias-variance breakdown that will be used extensively throughout this dissertation.
- Chapter 5 presents an overview of ensemble methodology.
- Chapter 6 presents a novel approach to combine classifiers using combining weights that are derived based on local performance.
- Chapter 7 presents the performance of individual classifiers with respect to varying dimension. This is a novel examination of the bias-variance decomposition for classifiers in the genomics domain.
- Chapter 8 presents the performance of several ensemble methods, including the proposed combination strategy using location detailed in Chapter 6.
- Chapter 9 presents a brief overview of variable importance for gene imprinting. This represents the first comprehensive analysis of univariate and multivariable variable importance measures in the prediction of human imprinted genes.
- Chapter 10 presents conclusions and future work in this area.

# Chapter 2

# Classification in Genomics and Data Sets Used

One of the most important analyses conducted in the area of genomics is the classification of samples or genes on the basis of processed array signals or sequence data mined from publicly available genome browsers. Gene expression "signatures" or "profiles" are typically statistical or machine learning-derived classifiers containing a few up to thousands of gene expression intensities as predictors, along with binary outcomes such as disease recurrence or tumor grade. The goal of these expression models is to adequately represent and predict the variation in outcome using genetic signatures. Gene expression is an indication of what proteins the cell is trying to express, although this relationship is an imperfect one. Gene expression is only one type of measurement that can be measured using arrays. Commonly used arrays include the following: comparative genomic hybridization (CGH), protein expression, single nucleotide polymorphisms (SNPs), and methylation. In CGH, researchers look directly at the genomic DNA, rather than the expression profile of RNA, allowing direct measurement of the copy number of a given gene. The pre-processing of these types of arrays is complex, and not a subject of this dissertation. Rather, we will focus on the processed data, and the building of classifiers to predict clinically relevant outcomes. This is the likely end-product of genomics-based research – the translation to clinical medicine.

As with any clinical model, we wish to not only predict, but also to understand. By understanding which genes are involved with outcome, we may focus therapies and limit expensive testing to a handful of important genes. However, until we are able to predict well, our understanding is, at best, incomplete. Therefore, accurate prediction remains the primary goal, with variable importance as a secondary, but vital, part of the process. There is a multiplicity of good classifiers, and their relationships to one another are difficult to quantify. Given this, important features should be evaluated and isolated based on multiple versions of events in order to be fully robust to deficiencies in the training set, or selected model, or selected classifier.

Building genomic-based classifiers was originally performed by the biologists themselves. Early publications in this field centered on the "two-fold" effect of genes, and less attention was paid to the variance of each gene's measurement or building multivariable models. Gene "profiles" or "genetic signatures" became popular as statistical and machine-learning techniques were used with increased frequency and as sample sizes became larger. Common classifiers used have ranged from simple DLDA-type models [27], to more complex Random Forest techniques [13, 33, 36,61], Support Vectors Machines [12, 15, 30, 35], and combined models [51,11]

## 2.1 Imprinting in humans

The process of sexual reproduction dictates that mammals inherit two copies of every gene, one from the mother and one from the father. At most loci, both alleles are actively transcribed and functionally equivalent. Imprinted genes represent an exception to this rule, as the transcriptional activity of each allele is determined by the gender of the parental germ line to which it was most recently exposed [73]. Genomic imprinting is an epigenetic marking on a gene based on the parent-of-origin resulting in one allele being "turned off" [21]. This parental legacy is initiated by epigenetic modifications such as DNA methylation, which is established in the parental germ line and maintained throughout somatic development in the offspring. Individual germ-line marks can control the allele-specific silencing or activation of multiple neighboring genes, which leads in

many instances to clusters of imprinted transcripts. The reasons behind this phenomenon are not fully understood, however it is believed that the "imprinting tag" is due to the methylation of parent-specific domains that are established during gametogenesis.

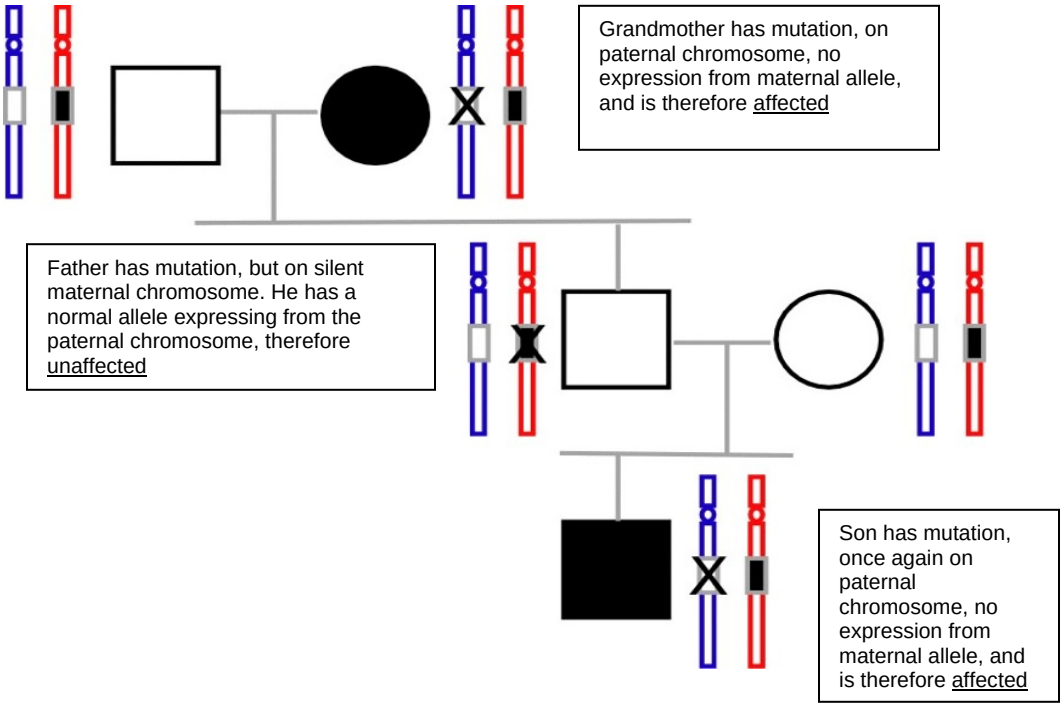



Figure 1: Parent-of-Origin – A paternally expressed gene. Offspring are affected only if they inherit the mutation from the paternal side, otherwise the gene is silent.

There are numerous genetic diseases caused by imprinting defects, such as Prader-Willi and Angelman syndromes, that result in devastating and often fatal defects found at birth. Imprinting has also been increasingly studied with respect to its effect on cancer. Numerous tumors have shown to have preferential loss of a particular parental chromosome, which indicates that imprinted genes may be involved. Loss of the only functional gene at an imprinted region may cause a tumor suppressor gene to be non-expressed. Loss of the imprinting mark may cause a gene that promotes cell growth to be over-expressed. There have been tens of imprinted genes discovered and validated, however the total number of genes expected to be imprinted in the genome may be in the hundreds [21,28]. Testing and validation are costly, and it is impossible to cover the

entire genome. Instead, researchers are looking for ways to prioritize genes with respect to testing.

It has been observed previously that DNA sequence characteristics can be used to predict regions along the genome where imprinted genes reside [28, 46]. Using these genomics-based tools to predict imprinted genes, researchers will be able to prioritize genes for further testing and validation.

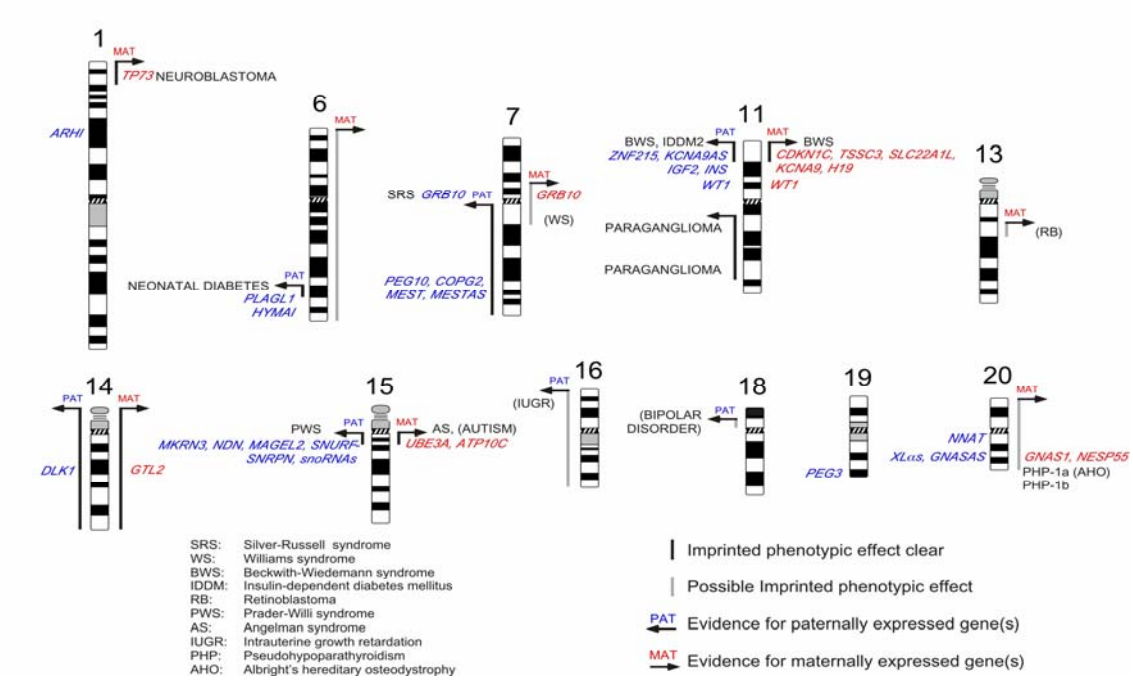



Figure 2: Human Imprinting Map. The 11 chromosomes presented are those where imprinted genes have been found to date. It is expected that there are many more undiscovered imprinted genes in the genome.

In this dissertation, a set of human imprinted genes are compared with a set of non-imprinted genes on the basis of many DNA sequence features flanking each gene's transcription start site. Previous research on human imprinting examined sequence features singularly [28], and was more descriptive in nature. Other groups have examined the prediction of murine imprinted genes using support vector machines [21].

To date, there is no systematic and large-scale study of human imprinted genes using sequence features alone that generates predictions for the rest of the genome. Although similarities are expected to exist with respect to mouse models, it is the goal of this project to build a classifier based on DNA sequence features designed specifically for the human genome. The window of each flanking region is varied from 10kb to 500kb and the sequence information is collected. The goal of the analysis is to build a highly accurate classifier based on easily attained DNA sequence characteristics, since this classifier will be used in regions of interest to pinpoint genes that are good candidates for further testing. The sequence information collected included data on repetitive sequences, CpG islands, and exon number. Repetitive sequences are those patterns in DNA that repeat – such as LINES (long interspersed repetitive sequences) and SINEs (short interspersed). These sequences are known specifically as retrotransposons, since they can reproduce themselves and insert themselves into DNA. ALUs (about 300 base pairs in length) are part of the SINE class of retroelements, and are the largest family within the SINE class of elements. ALU's main purpose, it is believed, is to replicate and copy themselves into new areas of the genome. This insertion is responsible for a 10% growth in the human genome since divergence from the chimpanzee. Since ALUs do not actually contain the proper machinery by which to insert themselves, they use the enzymes produced by LINE elements. This makes ALU and L1s (in the LINE class) highly related, since it is dependent upon the continued activity of LINE elements for survival [29]. Of great interest is the fact that ALUs are rich in CpG dinucleotides, which is the principal substrate for DNA methylation. When ALU inserts itself, it is inserting CpGs. L2 and MIRs have a similar piggy-backing relationship, where if L2 elements become inactive, MIR elements become inactive. CpG islands are more formally defined to be regions in the genome with a higher than expected concentration of CG dinucleotides. It is expected that the methylation occurs in these regions, therefore characterization of the gene with respect to the number of CpG islands is informative.

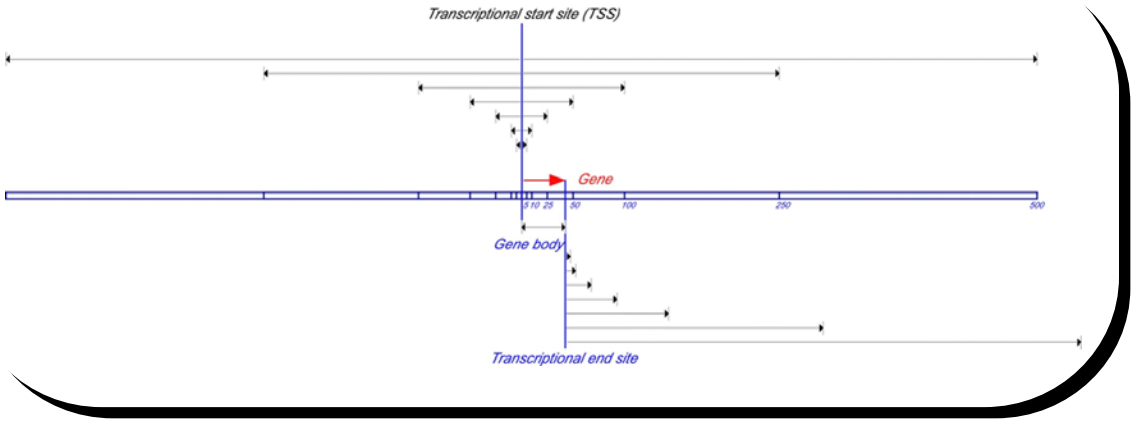



Figure 3: Flanking region of each gene

## 2.2 Publicly available genomic data sets used

The data sets detailed in this section are all publicly available data sets that have been previously used as benchmarking and illustrative problems in the field of genomics. A brief description is provided, along with a summary of sample size and feature set size. Further details may be found in Appendix A and in each of the referenced journal articles that are the source for this data. These data sets represent common classification applications in genomics, and are varied with respect to noise, sample size, class balance, and features.

### Colon cancer (Alon, 1999)

The data set used is composed of 40 colon tumor samples and 22 normal colon tissue samples, analyzed with an Affymetrix oligonucleotide array [5]. A set of 2,000 high intensity gene measurements is publicly available (see Appendix A). This data set is a commonly used benchmarking data set, representative of many high dimensional

genomics-based data sets. The goal of the analysis is to accurately classify samples as cancerous or normal using gene expression levels.

## Estrogen receptor (Blair, 2000)

A number of environmental chemicals known as endocrine-disrupting chemicals (EDC's) are suspected of disrupting function by mimicking or antagonizing natural hormones in animals and humans. The FDA's National Center for Toxicological Research (NCTR) estrogen activity database contains 232 samples (structurally diverse chemicals) and 312 predictors [7]. Out of the 232 chemicals, 131 exhibit estrogen receptor binding activity, while the remaining 101 are inactive, meaning that no activity was detectable in the assay. The data set is publicly available (see Appendix A) and consists of 232 independent observations (representing structurally diverse chemicals), activity status, and 312 potential predictors of activity.

## Prostate cancer (Singh, 2002)

Prostate tumors are heterogeneous tumors, resulting in high variability in outcome measures, even after adjustment for important clinical characteristics. While age, serum PSA, Gleason score and performance index are all independent clinical correlates of outcomes, whether or not a set of gene expressions could predict outcome is of interest. If gene expression can predict outcomes such as disease progression and PSA recurrence at the time of diagnosis or treatment, the course of therapy could be tailored to patients with higher or lower risk, as well as provide information with respect to prognosis.

The prostate data set [57] in this dissertation consists of 52 prostate cancer samples, and 50 normal samples. Gene expression measurements of over 6,000 genes are available along with class status (see Appendix A). The goal of this study is to build models that can accurately predict class status. While this is not the final goal – the ultimate goal is to identify genes that are differentially expressed between cancerous and normal tissue – this data set provides a good benchmarking set to build classifiers.

## Lymphoma data set (Alizadeh, 2000)

Diffuse large B-cell lymphomas (DLBCL's) are the most common subtype of non-Hodgkin's lymphoma. They represent a clinically heterogeneous group, with respect to both treatment response and clinical outcomes. It was of interest to examine whether these differences could be further explained by the molecular characterization of the tumor. Alizadeh [4] identified two variants of DLBCL samples with different patterns of clinical behavior in a total of 46 samples. Using the 4,026 gene expression measurements available, the goal is to build a classifier that classifies DLBCL subtype using gene expression measurements.

# Chapter 3

# Commonly Used Classifiers in Genomics

Some of the standard classification techniques will be discussed in this chapter. There is a mix of statistical methods that have been used for decades, as well as machine-learning techniques that are relatively new. The main advantage to the newer techniques is that they offer some form of regularization internally, which gives them good generalization performance. Whether these new methods yield a consistent and substantial improvement over much simpler standard techniques is a current subject of debate [20, 72, 31], and depends on the goals of analysis.

## 3.1 The support vector machine (SVM)

A Support Vector Machine [70] is known as a maximum margin classifier. The SVM maps the input vectors into a feature space, often of higher dimension, depending on the kernel used. In this feature space, a maximal separating hyperplane is constructed. This hyperplane is called maximal separating, because it maximizes the distance from it to the closest positive and negative correctly classified points in the feature space. By maximizing this margin, we are ensuring good generalization performance relative to other hyperplanes that separate the data. The closest points in terms of class separation are the so-called support vectors, and these are the primary data points on which the SVM

is based. Points that exist far from the boundary have no impact on the resulting classifier.

If we look at an SVM with the feature space as the original input space (i.e. using the simple dot product as our kernel), we can easily apply this understanding to higher dimensional feature spaces. Suppose we have a data set with two predictors, X and Z and the data are linearly separable. The small data set in Table 1 is a good illustration.

Table 1: An illustrative data set for input into SVM

| **Y** | **X1** | **X2** |
|---|---|---|
| +1 | 1 | 0 |
| +1 | 2 | -1 |
| -1 | 0 | 1 |
| -1 | -1 | 2 |

The data are obviously separable in the input space; therefore we may illustrate SVM using the simple dot product for the kernel. We would like to use these predictors in a model to classify new cases vs. controls (Y=+1 or -1 respectively).

The hyperplane is such that $w \cdot x - b = 0$. $w$ represents the norm to the hyperplane, and $b$ is the offset that forces the hyperplane to pass through the origin. The closest points to the hyperplane defined by w are those where $w \cdot x - b = k$. Without loss of generality, we let k=1. Then, the two parallel hyperplanes to the optimal hyperplane are defined as:

1) $w \cdot x - b \geq +1$ for all y = +1
2) $w \cdot x - b \leq -1$ for all y = -1

All other points are then defined relative to the points on the margin.

If the training data are linearly separable, we are able to have no data points within the two margins, and the distance between the two hyperplanes, which is equal to 2/||w||, is thus maximized in the SVM algorithm.

The resulting optimization may be expressed as [69]:

$$\min_{w} \frac{1}{2}\|w\|^2 \qquad \text{subject to: } y_i(w \cdot x_i) \geq 1$$

This means that we are maximizing the margin, under the constraint that all of the cases (y = +1) are on one side of the hyperplane, and all of the controls (y = -1) are on the other. By maximizing the margin, we are ensuring good generalizability since we have greater margin for error on new testing observations, compared to non margin-maximizing hyperplanes.

In our small example, the best separator is the line going through the origin and passing through the point [1,1]. The two support vectors are the points closest to the hyperplane, and are the only ones that are utilized in the SVM to determine the decision boundary. If we removed the two extreme points, the classification would not change. This adds robustness to the support vector machine, as extreme points do not contribute to the decision function as they would in (for example) discriminant analysis, which utilizes this information in the estimation of covariance. Figure 1 illustrates the data points from this simple artificial example, as well as the separating line found using linear SVM.

Test points that fall directly onto the hyperplane are equal to zero by design, meaning no informative decision on class. To the right of this hyperplane are the predicted cases, the further away from the decision plane the more confident the prediction.

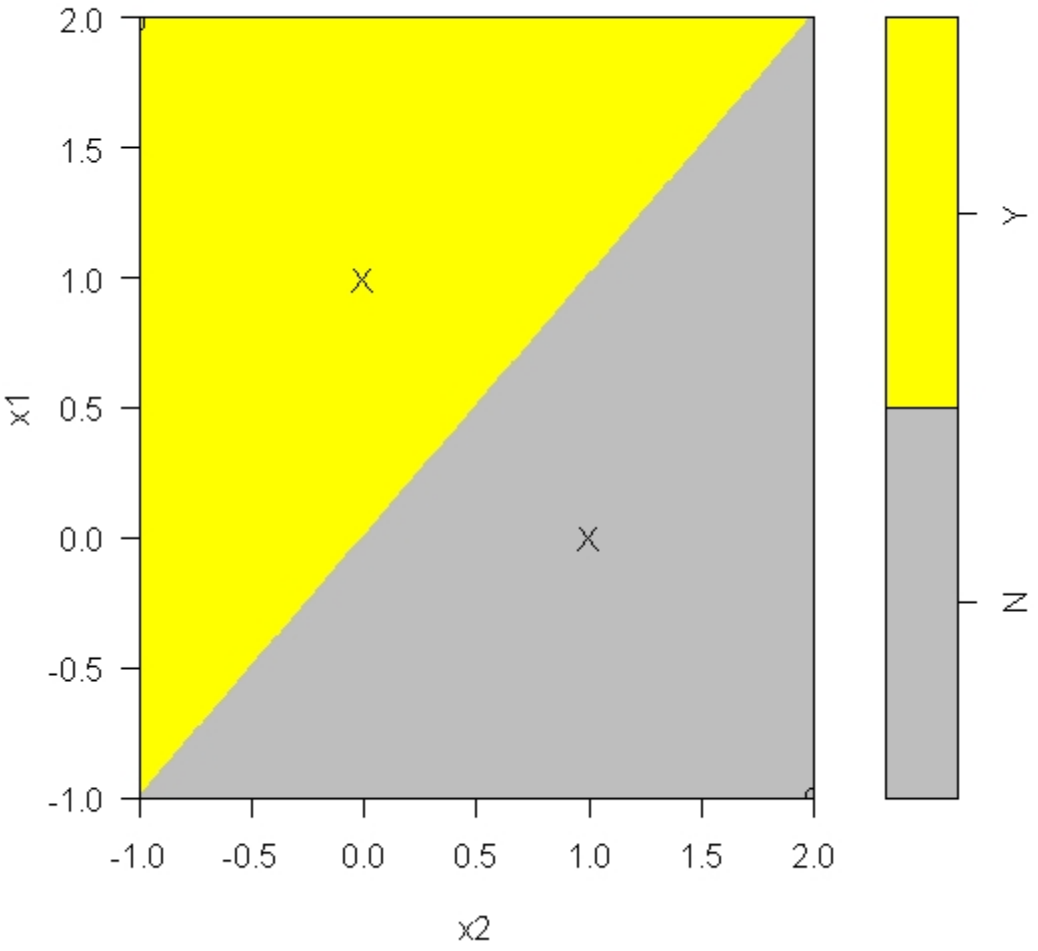
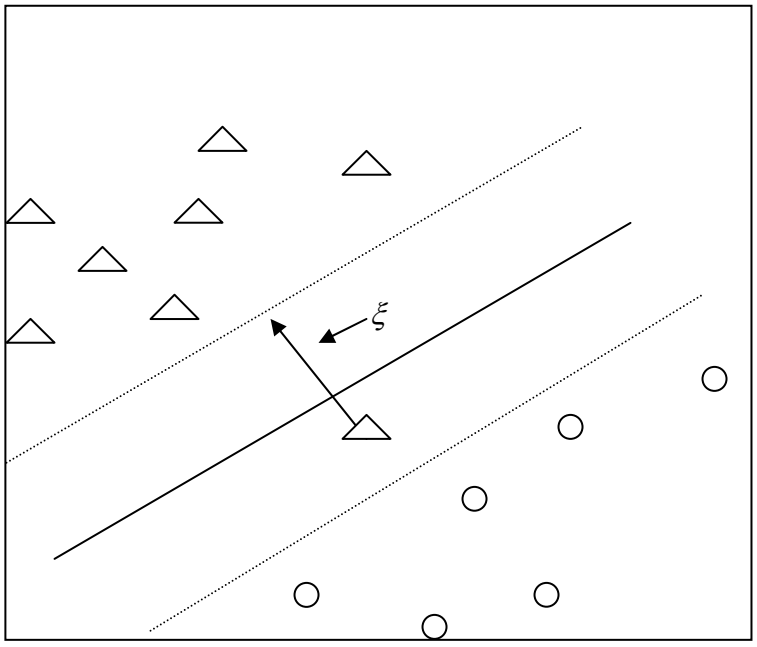



Figure 4: Classification in input space using SVM. The gray area depicts the region where the negative (-1) class is the predicted class. The yellow area is predicted to be the positive (+1) class.

## Soft Margin SVM

Soft-margin SVM allows for errors in the classification of some training set observations by the addition of slack variables ($\xi$). The slack variable for observation i from the training set measures the distance of this observation from the correct class margin.

Figure 5: Example of a slack variable for soft-margin SVM




Therefore, if observation *i* is correctly classified, but on the margin, $\xi_i = 0$. If the observation falls past the margin, but still on the correct side of the decision (boundary observation), then $\xi_i$ is a value between 0 and 1.0. If the observation falls on the wrong side of the decision hyperplane then $\xi_i$ is a value greater than 1.0. Figure 5 illustrates such a situation. The observation is misclassified and the value of the slack variable is therefore greater than 1.0. Therefore, the total sum of these slack variables becomes part

of the minimization, and its impact is controlled by the cost parameter C. The objective function becomes:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n} \xi_i$$

Subject to:

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i$$

The *C* parameter controls the total amount of error in the training set, and represents a trade-off between margin size and misclassification of the training samples. If C is very small, then there is a low cost to misclassification of training samples. We sacrifice some training samples in order to achieve a more robust decision. This generally results in a wider margin, but more bias. If *C* is large, then there is a higher cost of misclassifying training samples, resulting in a classifier that is more tailored to the training set, with a smaller margin, and likely higher variance (over-fit). If *C* is infinite (meaning that the penalty for misclassified training samples is infinitely large), then we have a hard-margined SVM, if it exists.

## Non-linear SVM

When the data are not linearly separable in the original space, it is possible to build the linear classifier in an expanded space, called the feature space, and work with the same optimization problem in this new space. As the number of features is usually large and exhibit complex relationships both to each other and to outcome, it may be that the data are linearly separable in an expanded space, such as one that considers interactions and non-linear functions of the predictors. Mapping the original data points into a higher dimension is done through kernels. Kernels are functions that give us the inner product between two data points in feature space. Therefore, if we know the kernel, we may determine the inner product, or similarity between any two points in any space. Selection of the kernel is non-intuitive for many genomics problems; however standard kernels used are linear, polynomial and Radial Basis [62]. The choice of kernel dictates the overall flexibility, or capacity of the resulting classifier.

Table 2: Commonly used kernels for Support Vector Machines

| Kernel $K(x_i,x_j)$ | Form | Parameters needed |
|---|---|---|
| Linear | $\left(x_i \cdot x_j\right)$ | Cost parameter only |
| Radial Basis (RBF) | $\exp\left(\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)$ | Cost, sigma |
| Polynomial | $\left(x_i \cdot x_j + 1\right)^d$ | Cost, polynomial degree |

We may use the same optimization as before, but w can now be written as a linear function of the *n* training points (since the training points exist in a subspace of the expanded feature set).

$w = \sum_{i=1}^{n} c_i \Phi(x_i)$ where $\Phi(x_i)$ is the mapping from input to feature space Most observations will have $c_i$=0 because the weight is only positive if the point is closest to the maximum margin hyper-plane - a support vector.

If we re-write the objective function in its general form:

$$\min_f \sum_{i=1}^{n} (1 - y_i f(x))_+ + \frac{1}{C}\|f\|_k^2$$

The first term minimizes the hinge loss across all training samples, ensuring a small training error, while the second term is a regularization term that controls the tradeoff between smoothness and error. *C* is defined as before – if *C* is large, then we likely have

a small margin and higher dependence on the training set boundary observations. If C is small, we discard the maximum margin separator in favor of a more regularized solution.

## Parameter Tuning

In addition to kernel choice, the parameters must be carefully tuned in order to obtain a meaningful classifier. If, for example, we use the RBF kernel with a very small sigma, we essentially create the "Christmas tree" effect – with small balls around the support vectors- and highly biased decisions elsewhere (see Figure 6), whose value depends on the balance of cases to controls. This type of model is useless, as the generalization to new data sets will be low. The result of such a model will be high bias in most regions of the input space. An example of this is a set of points called "Admiral's delight". The base of the arrow are cases (Y=+1), while the two lines making up the arrow are controls (Y=-1). Figure 6 illustrates the effect of having a very small sigma parameter. The hyper-balls have width that is controlled by sigma; therefore setting the width too small will yield small balls around all of the instances. This will result in 100% training accuracy, but dismal generalization.

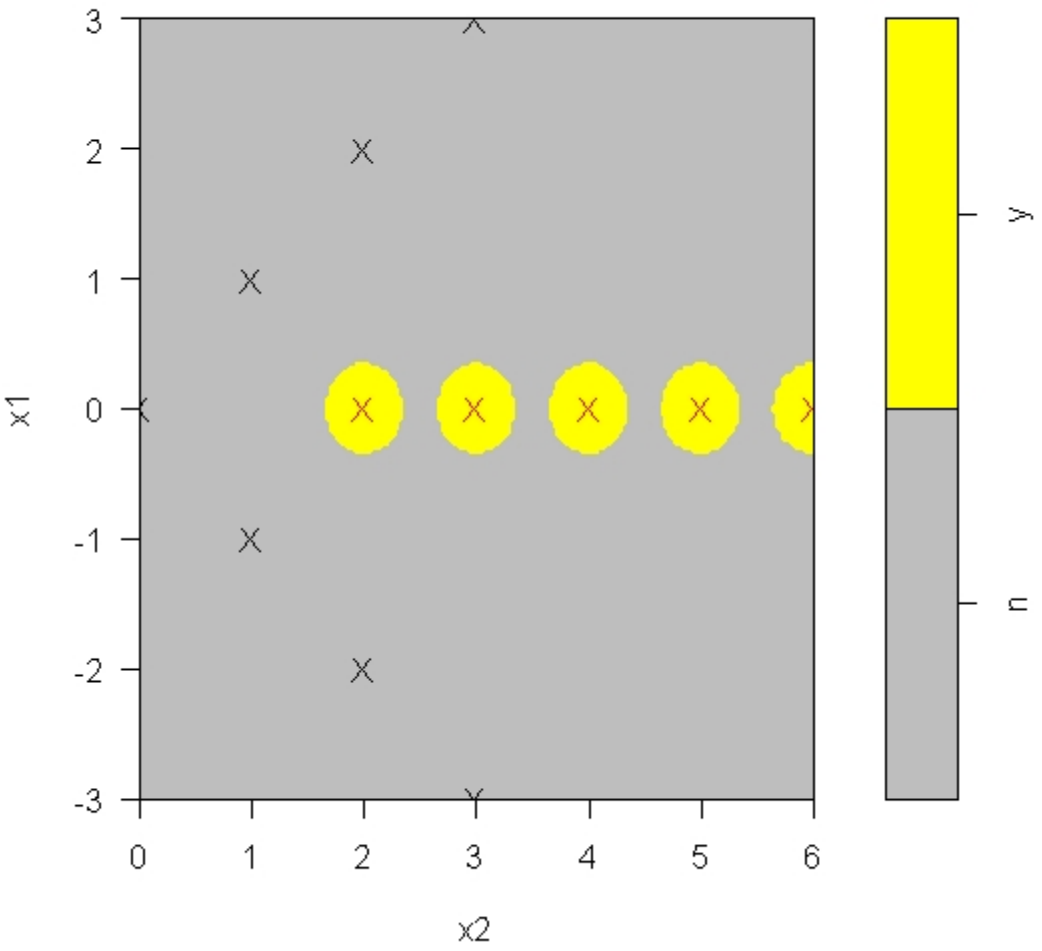



Figure 6: The "Christmas Tree" Effect in Radial Basis SVM. The width of the Gaussian kernel is too small, thus creating small hyper-balls around training samples that are predicted to be the positive class.

The final SVM discriminant function is a weighted sum of the similarities between the test point, and the support vectors. The Vapnik-Chervonenkis (VC) dimension [63] (related to the number of parameters, but not always) of the problem can be infinite using the RBF kernel, since perfect classification (shattering the points) can always be achieved on the training set regardless of the number of data points. *K* Nearest neighbor (KNN) classification with *K*=1 also has infinite VC dimension, however KNN is an instance-based rather than margin-based learner, therefore it tends to exhibit high variance (over-

fitting). On the other hand, SVM's do not overfit, even with this high flexibility, because it depends on only the support vectors and maximizes the margin of these vectors. We use SVM at the default parameters (gamma=1/p and *C*=1), since these are reasonable parameter levels [62] given the dimension, however further evaluation (though not an exhaustive search) of parameters within a range between 0.001 and 10 for gamma (which is defined to be $1/\sigma^2$) and *C*=0.10 to 10 for cost yielded negligible or deleterious effects on classification performance.

## 3.2 Diagonal linear discriminant analysis (DLDA)

Diagonal linear discriminant analysis is a simple classifier that uses the weighted contributions from a set of predictors to classify observations. The form of the classifier is:

$$\arg\min_k \sum_{g=1}^{G} \left\{ \frac{(x_g - \mu_{kg})^2}{{\sigma_g}^2} \right\}$$

where *k* is the class (0,1) and G represents the set of G predictors. Basically, for each new testing observation, we examine the normalized Euclidean distance (assuming a common diagonal covariance matrix) between the new observation and each class mean vector. The class that gives the smallest distance is the resulting classification for the test observation. DLDA can be regarded as a weighted vote of univariate classifiers, which can be highly successful if enough diverse features are combined, resulting in a classifier with both low bias and low variance. However, DLDA may also have a lot of bias due to the simplicity of its decision function, and this bias may increase as more features are added.

As is clear from this, the individual contributions are not modified by the multivariable relationships between different predictors. Therefore, two highly correlated predictors are treated as independent. In addition, we estimate a common covariance matrix for each class, which may smooth between-class differences. Although these assumptions are quite unrealistic in the genomics setting since 1) gene expression and other measures tend to follow a pattern of co-regulation and 2) it is likely that the

expression variability is greater in diseased samples than in controls or vice-versa, DLDA has been shown to be a useful and consistently good classifier [20].

## 3.3 Classification and regression trees (CART)

Classification and Regression Trees are a commonly used method, though often now seen as base members in ensembles [10]. It is a flexible, non-parametric procedure that creates a decision tree based on a greedy splitting process. The root node, containing all of the observations, is split based on the Gini index (see section 3.6), which measures the purity in each daughter call. This split is achieved through an exhaustive search of the best variable and split. Once the two daughter nodes are formed, the process starts over. Multiple splits on the same variable as permitted, allowing for a flexible representation of the data. It is clear from the process that CART is a high variance classifier, and multiple runs across different training sets often yields highly variable decisions. Often, the bias can be made very low since on average these flexible trees are able to represent the decision boundary well. CART trees are the base members used in the Random Forest ensemble [10].

## 3.4 Random Forest (RF)

Random Forest is a method developed recently by Breiman [8, 10]. It is an ensemble-based approach, using fully grown CART trees as base members of an ensemble. As mentioned in the previous section, CART trees are highly flexible representations; they are completely non-parametric and offer an attractive solution to the high dimensional data issue. Since the tree is built one split at a time, the search for the "best" tree is as simple as searching for the best splitting predictor and cut-point at each node. At the final leaves of the tree, the class decision is based on the proportion of cases vs. controls in each particular node. In RF, the trees are fully grown, therefore they are grown to a depth that is usually considered to be too training-set specific, therefore over-fit. As stand alone classifiers, decision trees without pruning tend to display extremely

high variance and therefore lower generalization accuracy. For example, in an illustrative data set, the average accuracy of a base tree in the RF ensemble was 66%, with 60% of the error accounted for by net variance. This leaves a lot of room for improvement, and by taking many of these trees, the variance is subsequently reduced to 10% of the error. The bias may be slightly reduced; however this reduction is inconsistent across applications and usually of minor benefit. Each base classifier contributes one vote to the majority-vote-based ensemble decision. Test cases are subsequently predicted by dropping the observation down all of the trees in the ensemble. Since fully-grown CART trees are likely to be over-fit, they tend to have highly variable performance across many different training sets. The RF ensemble exploits this high variance and creates an ensemble with much reduced variance (see Chapter 4), thus reducing overall error. Since the amount of variance reduction is highly dependent on the level of diversity in the base CART trees, the RF procedure attempts to create trees that are as diverse as possible. Random Forest creates diversity in two ways: it perturbs the training set used by taking bootstrap samples for each tree generated (bagging) and it selects a random subspace of the predictors at each node. Bagging is a common approach to building ensembles, which allows different models of the same class to be built. By perturbing the data for each bagged sample, the model is being estimated a slightly different training set each time, therefore increasing the diversity (reducing the correlation) between classifiers. Based on a small empirical study of bagging, the average correlation of the base trees appears to be reduced, as expected, however the average accuracy of each base tree also decreases, thus resulting in zero net gain in ensemble accuracy (Table 3). It is likely that due to bootstrap selection, there are fewer independent pieces of information for training, and the resulting base trees have higher error. The main advantage of bagging in random forest is the useful information about accuracy and variable importance, as well as an internal testing set for each tree. Roughly 1/3 of the data are not selected for any one bootstrap sample. Therefore, 1/3 of the data is considered *out of bag (OOB)* samples. The OOB samples may then be used as the test data to assess error rates. The OOB error estimates reported in Random Forest are quite similar to those reported using cross validation (data not shown).

The other diversity increasing process is how Random Forest changes the potential predictors available for selection. At each node, a random subset of the predictors is retained, and the best splitter is selected out of this smaller group. The random selection of predictors at each node allows the resulting trees to be as different as possible, with the possibility of different variables at or close to the important root splits. Given this, each CART tree is a different representation of the data. Due to this variable selection process, the resulting tree is not optimal with respect to any fixed subspace of the data.

Table 3: Impact of Bagging on the Imprinting Data Set

| Type | Ensemble accuracy | Base Tree accuracy | Min | Max | Avg. corr | Min | Max |
|---|---|---|---|---|---|---|---|
| **Bagged** | 0.875 | 0.654 | 0.338 | 0.931 | 0.086 | -0.70 | 0.96 |
| **Non-Bagged** | 0.880 | 0.696 | 0.408 | 0.944 | 0.125 | -0.61 | 0.98 |

## 3.5 K nearest neighbors (K-NN)

K nearest neighbors is a well-known non-parametric instance-based learner. For each testing sample, the K nearest neighbors (usually in Euclidean distance) are found and classification is based on the majority vote. Different variants of K-NN exist – such as weighted majority voting or different distance metrics. Usually, better performance is found by standardizing the data prior to analysis to avoid high variance predictors having too much influence [53].

The main issue in K-NN is dimensionality. In large dimensions, data become sparse and the concept of distance becomes meaningless. In very large dimensions, there is

usually little difference between the two closest points and the two points that are furthest from each other . Therefore, variable pre-screening or reduction to the first several principal components is typically performed prior to analysis to reduce dimension.

## 3.6 Logistic regression

Logistic regression is a standard statistical model used to relate a binary outcome variable (y=0,1 for example) to a set of predictors. The form of the model is:

$$\log\left(\frac{p}{1-p}\right) = x'\beta$$

where $p$ is the probability that $y=1$. Therefore, we transform probability to odds and then fit a linear function. Logistic regression is a commonly used model, especially when the primary goal is description. This is because the parameter estimates have direct probabilistic interpretation.

In high dimensional classification, using logistic regression poses a challenge. Maximum likelihood estimation can be problematic when the data are sparse. Infinite parameter estimates occur when the data are perfectly separated – which may be due to sparseness. Although this is not ideal, the posterior predictions generated based on this model are still useable; however they may be extreme (0 or 1). This may cause the predictions on the test set to be completely opposite of the true class. Of larger concern, the number of variables able to be modeled must be less than the number of observations in the training set, and sometimes far less. Given the uncertainty in which predictors out of thousands are most informative in a multivariable sense, standard logistic regression may be unstable.

The apparent instability of logistic regression has prevented it from being used in many high dimensional classification settings. Different forms of regularization exist such as ridge regression which will constrain the size of the parameter estimates, however they are not considered in this dissertation.

Logistic regression has some similarities to SVM. Both methods weight observations that are far from the decision boundary less than those observations that are close to the

boundary [32]. This is in contrast to discriminant analysis (LDA, DLDA, and variants) which use all of the observations equally in estimating covariance and means. This property can be advantageous with outlying observations, but loses efficiency compared to LDA when the data are multivariate normal. In order to better understand SVM, our primary linear classifier for this study; and its similarities to logistic regression, we briefly provide more detail in the next section.

## 3.7 The diversity of different classifiers

For linear classifiers such as logistic regression, DLDA, and SVM with a linear kernel, the feature space is divided by a hyperplane. Random Forest divides the feature space into hyper-rectangles. Nearest neighbor approaches divide the space into polyhedral cells – a Voronoi tessellation. Non-linear classifiers, such as SVM with an RBF kernel, fit non-linear complex surfaces in input space using an expanded feature representation to linearly separate the classes.

The diversity of classifiers comes from many sources. First, diversity is in the different geometries of the way the feature space is divided. As described above, the decision surface being fit is very diverse. Second, due to the typically small sample size available for each problem, each classifier needs to discern between noise and pattern relying on relatively few training points. Too few data points in the space can make the resulting classifier too simple, and therefore biased. Also, too few data points with a flexible classifier can make the space appear to be governed by a very complex decision boundary – and therefore the result is over-fitting. Third, each classifier is being optimized based on different criterion. Logistic regression attempts to maximize the log likelihood. SVM minimizes the hinge loss. Random Forest minimizes the Gini index at each node, which is an impurity measure based on the products of the proportions of each class in each node. This is a measure of the variance of the prediction within each node.

Table 4: Loss functions used by different classifiers

| **Classifier** | **Loss** (y in [-1,1] ) |
|---|---|
| Logistic regression | Logit loss: $\log\left(1+e^{-y_i f(x_i)}\right)$ |
| SVM | Hinge loss: $\max\left(0,1-y_i f(x_i)\right)$ |
| Random Forest | Gini index: $1-\left(p_0 p_0 + p_0 p_1 + p_1 p_1\right)$ |

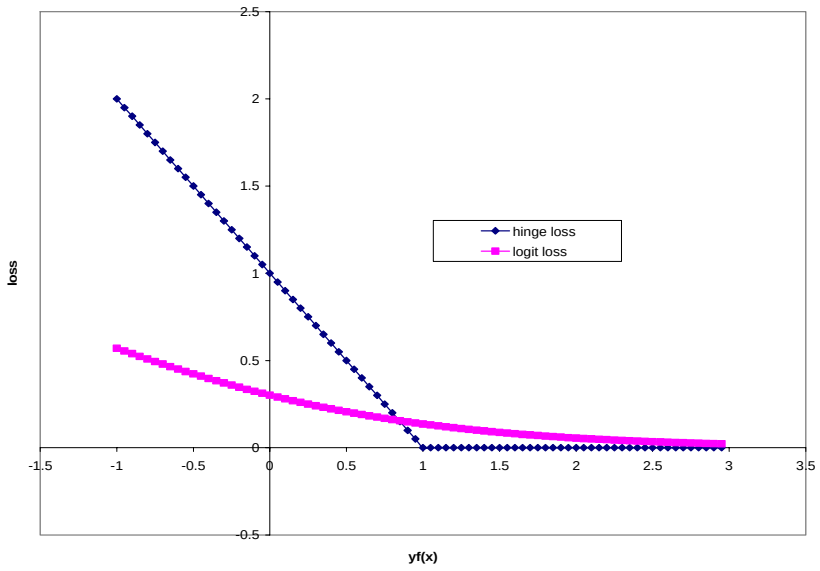



Figure 7: Hinge (blue) versus logistic loss (pink)

In Figure 6, we observe that the hinge and logit loss operate in a similar way. Observations that fall to the right of zero are ones that are correctly classified. Both loss functions assign a positive penalty which diminishes linearly as the prediction moves away from the boundary. For misclassified observations, to the right of zero, both loss functions assign higher penalties the further away from the decision boundary.

## An array of diverse classifiers

In trying to maximize diversity, we consider the following classifiers:

1. SVM using a simple dot product (linear kernel) in input space
2. SVM using the RBF (Gaussian) kernel in expanded feature space
3. DLDA using a simple weighted sum of univariate distances
4. CART with a simple splitting rule based on node size (min n for split=10)
5. k-NN with k=3. The number of nearest neighbors was set to be 3 prior to analysis. It represents a compromise between higher variance (smaller k) and bias (larger k), favoring a slightly more over-fit classifier.

Of course each classifier is attempting to classify the same set of observations; therefore there will always be a good level of overlap between them for easy-to-fit observations, as well as observations that are incorrectly classified due to noise (Bayes error).

The output of each classifier may be a binary decision, or some sort of prediction score. A prediction in the interval [0,1] is easily obtained in CART and kNN by considering the proportion of each class that falls into the same node or neighborhood as the test point. DLDA and SVM output a class label, but it is possible to obtain predictions by using logistic regression, treating the function value as the input variable.

By taking the classifier with the highest accuracy, we are failing to take the selection bias into account. By including the selection of the best classifier (which we denote as *best*) into the cross validation, we are obtaining a more realistic assessment of the true accuracy of the *best* classifier. Thus for each cross validation, we select the best classifier out of the five using bootstrapped accuracy estimates and use this classifier on the test set. This is important since the best classifier on the training set (or perturbed versions of the training set) may not be the best classifier on the test set. This captures the variability in selecting an optimal classifier and corrects for this bias in the final estimates of accuracy. The level of selection bias depends on the variability of the best classifier across iterations. Therefore, we present the individual accuracies along with the accuracy of the *best* classifier.

## 3.8 Variable screening

In many of the classification methods presented above, the dimension of the problem should be considered. A good portion of the potential predictors in any large mined study are simply noise and can be removed from the analysis without further consideration or deleterious effect. For simplicity, noise is usually defined based on the univariate test performance. Those variables with small absolute values of the test statistic are likely to be uninformative overall. In this work, we used the BW statistic to rank genes. The BW statistic is defined as the between to within sums of squares and is a common ranking criterion [20]. However, any non-parametric analogue or variant may be used. Variables that are highly associated with class status are retained. Although p-values may be used as a tool to assess whether this association is statistically significant, the simultaneous testing of thousands of hypotheses creates a multiple testing issue, and the chance for many false positives using $p < 0.05$. The correction of p-values for multiple testing is an active area of research and is not explored in this work. Given the exploratory nature of the analyses, and the fact a simple ranking of predictors based on the magnitude of the test statistic is invariant to p-value adjustment, we do not perform any formal p-value adjustment to determine the set of selected features.

In this thesis, initial variable screening is performed for many of the analyses. If we allow variable screening to be a part of the cross validation, then the error rates are more realistic than if we fix the set of predictors on the basis of their performance on the entire data set. Both approaches are valid if the inference of the results is conditional upon the fixed parameters. In this work, variable pre-screening is performed by augmenting the feature set with 10% noise, randomly generated from a N(0,1) distribution. A natural threshold in the ranked list would then be where the first appearance of an artificial variable is detected. If a feature is ranked below an artificial variable generated independently of class status, then it likely contains negligible information. Although this method is quite *ad hoc*, it appears to correlate well to the "hump" of the information curve. This approach has been used previously in other applications [48, 55] to provide a

stopping rule for model selection using an exhaustive search of all models, but it has not been utilized in genomic variable screening to our knowledge. Nonetheless, it provides an intuitive and reasonable stopping point for inclusion into the informative set.

The reason we do not fix the number of predictors selected is because the relevant number will vary for each data set. In data with a high number of correlated predictors, the ranked list has clusters of variables that take the top positions. If p is fixed, then we may miss information due to only a few clusters being selected. By allowing p to vary, we allow the informative set to be data-specific.

The drawback to using a univariate approach is obviously that we may miss important joint effects or interactions. However, given the sample sizes involved, as well as the lack of *a priori* knowledge about such relationships, we assume that the likelihood of such important interactions between features not already included in the relevant set of predictors is small. This assumption will be explored further in Chapter 9.

## 3.9 Statistical packages used

All analyses and data manipulation were performed using R statistical software. Specifically, we used the following procedures and tuning parameters:

| | **Procedure** | **Tuning parameters** |
|---|---|---|
| CART | rpart | minsize=10 |
| Random Forest | randomForest | mtry=sqrt(p) (default),<br>ntrees=500 (prediction) or 1000 (VI) |
| SVM-linear | e1071 | kernel=”linear”, C=1 (default) |
| SVM-RBF | e1071 | kernel=”radial”, C=1, gamma=1/p (default) |
| DLDA | stat.diag.da | none |
| kNN | kknn | k=3 |

# Chapter 4

# The Decomposition of Error into Bias and Variance

In the linear regression setting, the evaluation of a predictive model may be based on the mean squared error of prediction at $X=x$.

$$\text{MSEP} = \quad E_Y(y-\hat{y})^2 = ((f(x)-\hat{y})^2) + E_Y(y-f(x))^2$$

MSEP is an estimate of the average loss (over $Y$) by using $\hat{y}$ as the prediction for y at input $X=x$ for a fixed dataset. The systematic component of y is f(x). The first term represents an error term that is under our control through our selected model and will vary as our dataset varies. The second term represents an irreducible error (the variation of y around the true function f at X=x) that is independent of the training set used. We take the expectation over all training sets D:

$$E_D\left[MSEP\right] = \quad E_{D,Y}(y-\hat{y})^2 = E_D((f(x)-\hat{y})^2) + E_Y(y-f(x))^2$$

and further decompose the first, reducible, term:

$$E_D((f(x)-\hat{y})^2) = (f(x)-E_D\left[\hat{y}\right])^2 + E_D\left[(\hat{y}-E\left[\hat{y}\right])^2\right]$$
$$= bias^2(\hat{y}) + \text{var}(\hat{y})$$

The bias term is the difference between the systematic part of the true function (f(x)) and the expected prediction based on the model. Therefore, if a model has very low bias, then across different datasets we expect our model to be close to the "truth" on average.

The variance term represents the variance of prediction at $X=x$ about its expectation (across D). This variance is independent of the underlying function f(x), since it is entirely dependent on the classifier and its stability across all datasets D. There is a well-understood bias-variance tradeoff [49]. The effects of bias and variance are additive, and therefore for a given error rate, we may decrease the variance, but at a cost of increasing the bias. For an extreme example, if we always predict y to be 7, regardless of the dataset D, then we have no variance in our predictions across D, however we will probably have lots of bias since we will not be capturing the true function over the input space. If we over-fit the model, we expect the variance to be high, but the bias to be lower. Why is the variance high? Across D, our predictions in fitting this complex model will be quite variable. Different data sets will produce quite different parameter estimates for the model. If we have built the model using data-driven methods such as variable selection and estimation on the training set; then this variability will be even more of an issue. The more intense the model selection procedure, the more likely it is to introduce variability when repeating this procedure across D. In addition, if we allow the best model to be selected from a large model space, then we will likely have high variance even if the final model is in fact small. The size and components of the final model will be varied from training set to training set, which will also contribute to the variance. When sample sizes are small many models are not well-supported by the data, and there is a multiplicity of equally "good" models which results in large variability across data sets. In classification, the same concerns about over- and under-fitting are present for the reasons described. However, the *impact* of variance and bias may be different in the classifier setting where we try to classify each observation into one of $k$ classes. When we build a predictive model we are likely not as concerned with our estimates of the posterior probability P(Y=c|X=x), c=0,1,2,…,k. Typically, we either want just the predicted class or a ranking of observations (from which we may take the top m observations as the "highest-risk" subset). Both situations do not require precise estimates for the posterior probability. For example in the two-class problem, low estimation error $P(Y=1|X)-\hat{P}(Y=1|X)$ is likely less of a goal than a low classification error $\Pr(y \neq \hat{y})$. In this setting, our classifier may have bias, however the predicted class is correct. The bias of the model is only important when the expected

prediction at X=x takes us out of the correct class. Friedman [22] calls this positive boundary bias. When a particular region of the input space at X=x has positive boundary bias, we are predicting the incorrect class on average across D.

For classification-based models, evaluation of error based on squared error is less appealing and new ways to evaluate these models were proposed [19, 22, 38, 39] based on 0/1 loss. Using a 0/1 loss function does not result in the familiar additive relationship between bias and variance as for squared error loss.

## 4.1 Decomposition in classification

### 4.1.1 Friedman's 0/1 loss

For classification problems, Friedman [22] defines the probability of misclassification: $\Pr\left(\hat{y}(x) \neq y\right)$ where $\hat{y}$ (x) is the predicted class and equals

$$\begin{aligned} &1\ ,\ \hat{f}(x) \geq 0.50\,, \\ &0,\ \ \text{else.} \end{aligned}$$

Given a particular training set, D, the misclassification error rate depends on whether the predicted class agrees with the Bayes (optimal) class $y_B(x)$. If it does, then the error rate is the irreducible error associated with the Bayes rule. If it does not agree, then there is an increased error rate above that of the Bayes risk which is equal to:

$$\Pr\left(\hat{y}(x) \neq y \mid \hat{y}(x) \neq y_B(x)\right) = \max\left[f(x), 1-f(x)\right] = \left|2f(x)-1\right|$$

Therefore, the error rate may be decomposed as:

$$\Pr(\hat{y}(x) \neq y) = \left|2f(x)-1\right| \Pr(\hat{y}(x) \neq y_B(x)) + \Pr(y_B(x) \neq y)$$

When we average across all datasets D we have (conditioned on point **x** in the input space):

$$\Pr(\hat{y} \neq y) = \left|2f-1\right| \Pr(\hat{y} \neq y_B) + \Pr(y_B \neq y)$$

Friedman then shows that $\Pr(\hat{y} \neq y_B)$ is the tail area (direction depending on whether f(x) is greater than or less than 0.50) of the predicted distribution function. To

understand the distribution of $\hat{f}(x)$ (where the population is based on *training sets D*), Friedman then approximates it using a normal distribution with parameters $E(\hat{f}(x)), Var(\hat{f}(x))$. Since the computation of $\hat{f}(x)$ is usually a complex averaging process, the normal distribution is reasonable. Using this, Friedman then shows that

$$\Pr(\hat{y}(x) \neq y_B(x)) = \Phi\left[ sign(f - 0.50)\frac{E\hat{f} - 0.50}{\sqrt{\text{var}\,\hat{f}}} \right]$$

To be clear, if *f(x)* is 0.20 (Bayes class is class 0) and we have prediction $\hat{f}(x)$, the probability that the classes will not agree is equal to the probability that $\hat{f}(x)$ is greater than 0.50 – where $\hat{f}(x)$ is assumed to be normally distributed with mean and variance given above. Since

$$\Pr(\hat{y}(x) \neq y_B(x)) = \Pr(\hat{f}(x) > 0.50) = \Pr(Z > \frac{0.50 - E\hat{f}}{\sqrt{\text{var}\,\hat{f}}})$$

$$= \Phi\left[ \frac{0.50 - E\hat{f}}{\sqrt{\text{var}\,\hat{f}}} \right]$$

$$= \Phi\left[ (-)\frac{E\hat{f} - 0.50}{\sqrt{\text{var}\,\hat{f}}} \right]$$

Where Φ(Z)= Pr(Z>z) is the upper tail probability of the standard normal distribution. So what does this error depend on?

With $\text{var}(\hat{f}(x)) > 0$, we have a classifier that varies depending on the training set used (variance is equal to zero if the learner gives a constant prediction at X=x regardless of training set D used). Given this variability across training sets, the error depends on how far away our prediction is from the boundary of 0.50. If we have a bias from 0.50 and a small variance, then the error is maximized. If we have a bias, but a large variance, then the error is smaller. This is because we have a chance with a high variance classifier to swing over to the correct side of the classification from time to time. If we have no classification bias, that is $\hat{f}(x)$ is on the same side of 0.50 as f then $\hat{y}(x) = y_B(x)$ and the error is reduced to the irreducible error of the Bayes risk. Therefore, if we have a learner

with estimation bias $\left|f(x)-\hat{f}(x)\right|$, the impact in the classification setting is dependent on whether the resulting classification is the same as the Bayes classifier. A learner that is biased, but on average correct with respect to class, is just as good in the classification setting. Therefore, should we adopt more typically biased learners with low variance? As long as the bias is the "negative" kind described in Friedman caused by over-smoothed estimates of the posterior probabilities (like naïve Bayes or kNN), then it seems we should.

### 4.1.2 Kohavi and Wolpert's decomposition

Kohavi and Wolpert [39] provide one of the most widely used decompositions into bias and variance for 0/1 loss. In addition, they also provided a method by which these two quantities may be estimated from the data. They propose that $Y_F$ and $Y_H$ are independent, where $Y_F$ denotes the Pr(y=1) at X=x and H denotes the hypothesized class. This is true, because the probability of class 1 at input **X=x** depends only on f(x). The expected 0/1 loss is then defined to be:

$$\mathrm{E(C)} = \sum_x P(x)(\sigma_x^2 + bias_x^2 + \mathrm{var}_x) \text{ where}$$

$$bias_x^2 = \frac{1}{2}\sum_{y\in Y}\{\Pr(Y_F = y \mid X = x) - \Pr(Y_H = y \mid X = x)\}^2$$

$$\mathrm{var}_x = \tfrac{1}{2}\left(1 - \sum_{y\in Y} P(Y_H = y \mid X = x)^2\right)$$

$$\sigma_x^2 = \frac{1}{2}(1 - \sum_{y\in Y} P(Y_F = y \mid X = x)^2)$$

In this definition, the squared bias of the classifier is the squared difference between the average y at X=x and the predicted y. Therefore, we are comparing the *distribution functions* at X=x, as opposed to the Bayes class and the predicted class. If the two class designations agree, the bias could still be non-zero because the probability of Y=1 at X=x may be different. This is problematic, since we would like the bias term to be equal to zero if the Bayes class and predicted class are the same. However, consistent with what

we want, if the classifier is constant (independent of the dataset used) then the variance of the prediction will be equal to zero since at X=x the probability that the prediction is equal to y is 1 or 0 leaving just the irreducible error.

### 4.1.3 Domingos' unified bias-variance decomposition

One of the problems in Friedman's decomposition is that while he explains the impact of bias and variance in the classification setting, he largely leaves both quantities undefined. Kohavi and Wolpert's decomposition, while popular, also suffers from some important drawbacks, such as letting the Bayes optimal classifier to have a non-zero bias. While both of these decompositions both illustrate and enhance the understanding of bias and variance in classification, they have drawbacks and limitations, as mentioned above. Due to these issues, we have used the decomposition as per Domingos [19].

Domingos proposes a single definition of bias and variance for any loss function. *(Keeping with the terminology of the Domingos paper: t=the true class, y=predicted class, $y_m$ =E(y) and $y_B$=optimal (Bayes) class)*

$$E_{D,t}\left[L(t,y)\right] = c_1 E_t[L(t,y_B)] + L(y_B,y_m) + c_2 E_D\left[L(y_m,y)\right]$$

$$= c_1 \text{ noise} + \text{bias} + c_2 \text{ variance(y)}$$

Under two-class classification and a **symmetric loss function**

$$c_1 = \begin{aligned} & P_D(y=y_B) - P_D(y\neq y_B) = 1 - 2P_D(y\neq y_B) = 2P_D(y=y_B) - 1 \\ & = (1-2B(X))(1-2V(X)) \end{aligned}$$

$c_2$ = (1-2B(X)) which is equal to 1 if unbiased instance, and -1 if biased.

Therefore we have:

$$E_{D,t}(\hat{y}\neq y) = (2P_D(y=y_B)-1)E_t\left[1(t\neq y_B)\right] + E_D\left[1(E(\hat{y})-y)\right] \qquad (D1)$$

for an **unbiased classifier**, and

$$E_{D,t}(\hat{y}\neq y) = 1 + (2P_D(y=y_B)-1)E_t\left[1(t\neq y_B)\right] - E_D\left[1(E(\hat{y})-y)\right] \quad (D2)$$

for a **biased classifier**.

This is equivalent to:

$$E_{D,t}(\hat{y}\neq y) = (1-2B(X))(1-2V(X))N(X) + B(X) + (1-2B(X))V(X)$$

When the predicted class, on average, is consistent with the Bayes class (D1), the average error of the unbiased classifier is made up of two terms – the noise term N(x) and the variance term V(x). The *net variance* is the difference of the unbiased variance (the variance attributed to the unbiased inputs) and the biased variance. The noise term is multiplied by a quantity that is positive, and equal to 1 if the individual predictions are always the optimal ones. If this is the case, then the decomposition reduces to just the N(x) term. Otherwise, the noise term is modified by the probability that for an individual prediction, it is correct while the Bayes prediction is wrong. This is what is called the *wrongly-right* impact as described by Friedman [22].

When the predicted class, on average, is inconsistent with the Bayes classification then the average error has a **subtractive** variance component. This agrees with Friedman's assessment of the impact of variance on error – that increasing variance will reduce error in situations where there is bias. In addition, the noise term will be multiplied by a negative term. If the noise term is large, such that the Bayes classifier has a large error component, then we will have more situations where the prediction agrees with t, even though it disagrees with the Bayes class.

**Types of variance in classification – good and bad**

- $\mathbf{V_u}$ - Variation of predictions around the correct (optimal) prediction.
- $\mathbf{V_b}$ - Variation of predictions around incorrect class. We want this to be high.
- **Net variance** $\mathbf{V_n} = V_u - V_b$

Both Domingos and Friedman show that the relationship between bias and variance is multiplicative and that the impact of bias is specific to the input region and whether this input region has positive or negative boundary bias. The minimization of positive boundary bias should be a focus of classification, especially base classifiers for ensembles, since when the entire input space has negative boundary bias; we may minimize the variance and end up with a strong classifier.

The error rate of a classifier across all datasets D can be decomposed as:

$$\begin{aligned}
&\Pr(\hat{y} \neq y) = \Pr(\hat{y} \neq y_B)\Pr(y_B = y) + \Pr(\hat{y} = y_B)\Pr(y_B \neq y) \\
&= \Pr(\hat{y} \neq y_B)\left[\Pr(y_B = y) - \Pr(y_B \neq y)\right] \\
&= \Pr(\hat{y} \neq y_B)[2\Pr(y_B = y) - 1] \\
&= \Pr(\hat{y} \neq y_B)[2f(x) - 1]
\end{aligned}$$

where *f(x)* is the Bayes (optimal) classifier. Therefore, the interest is in how $\Pr(\hat{y} \neq y_B)$ depends on $\hat{f}$. This quantity is the probability that the Bayes class is not equal to the predicted class from the model. Friedman calls the mis-estimation of f(x) "boundary bias", since we can view f(x) as the optimal boundary separating class 1 from class 0. The boundary bias may be written as:

$$b\left(f, E\hat{f}\right) = sign(0.5 - f)(E\hat{f} - 0.5)$$

This is showing that the effect of mis-estimating the probability function f(x) on the misclassification error is dependent only on whether the expected predicted class falls on the correct side of the Bayes boundary or not. If we consistently get the class wrong, then we have positive boundary bias, which will impact the misclassification rate. If we have bias, but are consistently on the correct side of the decision boundary, then there is no impact to the error rate.

The impact of this is as follows: For low variance classifiers, the classification error rate depends entirely on whether there is positive boundary bias over a large portion of the input space. For high variance classifiers, the error rate will decrease with decreasing variance (through aggregation or model averaging) only if there is no positive boundary bias. Otherwise, the presence of positive boundary bias will cause the error rate to *increase* as variance decreases.

### 4.1.4 James' bias-variance effects

James details explicit rules that any decomposition should satisfy under any loss function [37]. Using this framework, he produces definitions of bias and variance suitable for non-squared error loss functions, including 0/1 loss.

The rules for any decomposition are as follows:

*i) When using squared error loss, the definition of bias and variance must reduce to the standard definitions*
*ii) The variance term must measure the variance of the predictions, and must therefore be independent of the true response function.*
*iii) The bias term must measure the differences between the systematic parts of the classifier and the response and should be equal to zero if there are no systematic differences.*

Using these rules, James defined an additive decomposition comprised of noise, variance *effect* and systematic *effect* components. Variance effect (VE) is a measure of how prediction error changes when we use $\hat{y}$ instead of E($\hat{y}$) to predict y. The systematic effect (SE) is a measure of how prediction error changes when we use E($\hat{y}$) instead of E(y) to predict the response.

$$VE = P_{T,C}(T \neq C) - P_T(T \neq SC)$$
$$SE = P_{T,}(T \neq SC) - P_T(T \neq ST)$$

The basic definitions used in this paper are identical to the ones given in Domingos. The net variance of Domingos ($V_n$), is identical to the variance effect (VE) of James when the noise effect N(X) is zero.

## 4.2 Estimation of bias and variance from real data

Estimation of bias and variance from real data is hindered because we do not know the true functional relationship. However, we may assume that the noise component is equal to zero (thus putting the noise term as part of the bias term). Kohavi and Wolpert's approach is to divide the data into two parts – one training set (D) and a test set (E). Since we want to get many different training sets (N training sets in total), we sample *m* observations without replacement from D. To get training samples of size m, D was chosen to be of size 2m. Each classifier built on the N training samples is evaluated on the same test set E. James [37] uses a bootstrap approach by producing 50 bootstrap samples to fit the classifier to each. For the noise term, he uses a neighborhood approach instead of assuming that the noise is zero. The general approach of Webb [71] for generating test samples is used for this study. We perform k fold cross validation multiple times, thus ensuring that each unique observation has an equal number of tests for computation of accuracy. Bias and variance terms are computed for each observation on all loops where the observation was in the hold-out test set. Therefore, we evaluate the stability of prediction at X=x for each observation 50 times. In all analyses, the value of k is set to 4 in order to have a large test set for each run. This four-fold iteration was performed 50 times, as per Webb. Experiments in allowing the number of iterations to be higher (k=100) did not change the resulting estimates of bias and variance substantially enough to warrant the extra computing time.

**Algorithm for estimation of bias and variance from real data sets when the true function is unknown**

| | |
|---|---|
| Step 1 | Perform k-fold cross validation (here, k=4). |
| Step 2 | Repeat J times (J=50) |
| Step 3 | This generates J test sets and J learners based on training samples |
| Step 4 | At each X=x, we estimate the **mode** of the predictions. The variability of the classifier is then the average loss incurred by the prediction to the main (mode) prediction. |
| Step 5 | The **bias term** is the loss incurred by the mode relative to the optimal prediction (here just the true class designation). |
| Step 6 | **Variance** is separated in $V_u$ and $V_b$ terms, depending on whether it comes from a biased or unbiased observation. |
| Step 7 | Bias and variance components are then **averaged over all examples** (X=x) to get an overall summary for the classifier. |
| Step 8 | The **margin** M(x) of the classifier at X=x is $2Pr(Y_H=1\|X=x) - 1$. We want to maximize this across all X=x. Maximizing the margin is a combination of minimizing bias and increasing variance for biased observations and decreasing variance for unbiased observations. |

# Chapter 5

# Ensemble Methodology

Commonly used ensembles are believed to be primarily *variance reduction* tools which can optimize classifier performance by driving down the variance on low-bias classifiers. Ensemble methods such as Random Forest have been shown to have good performance over a wide range of classification problems. The ensemble accuracy tends to be considerably higher than the accuracy of a single tree and is often competitive with or superior to other methods such as SVM, LDA, and logistic regression. The gain comes primarily from the variance reduction of combining fully grown decision trees. Bias reduction may be present if the forest of trees is majority-correct on average. However, if there is a substantial area of the input space with positive boundary bias, the combining of models will deteriorate classification performance and the ensemble performance as a whole may decline or remain the same.

## 5.1 Bias versus variance reduction in ensembles

If an observation is unbiased with a particular classifier, then on average it is classified correctly. However, due to over-fitting, there is some variability with the prediction. If we consider a set of classifiers, each predicting independently at X=x, then it is understood that the average prediction smoothes out some of the noise, and variance is

lower. If an observation is biased, then we do not want this variance reduction, we want the majority of the classifiers to be correct on average.

If the observation at X=x is located at a point close or on the wrong side of the theoretical decision boundary, or is outlying, the predictions for this case will likely be biased for some of the classifiers. If we combine a set of classifiers such that the biases are diverse, then we may reduce the overall bias of the ensemble by producing a majority unbiased classification.

## 5.2 Prentice's extended Beta-Binomial model

The general results of the probability of consensus was first recognized by Condorcet in 1795 [14] and later expanded upon by Lam and Suen [43]. The impact of aggregation on ensemble accuracy is also illustrated via simulation using the beta-binomial model and the extended beta-binomial model. Prentice showed that the beta-binomial model may be extended to cases where $\rho < 0$ for certain values [52]. He calls this the extended beta-binomial distribution, however the interpretation as a convolution of a beta and binomial random variable does not hold under negative correlation structures. We may use these distributions based on average correlation and average accuracy to obtain the expected ensemble accuracy gains when combining K classifiers. Based on the formulas, and illustrated by Table 5, the ensemble accuracy is quickly driven to 1.0 as the number of base classifiers increases and the correlation is negative. There is a limit to the magnitude of negative correlation as K increases. This is due to the constraint that the covariance matrix must be non-negative definite. The model used is a restrictive parametric model, with constant correlation between classifiers and constant accuracies assumed. In real applications, this consistency would be unlikely to hold. Therefore, the table serves as a useful guide for expected gains of an ensemble, but actual ensemble gain is likely to be lower than that projected by this model.

| | | Pr(correct) for each base classifier | | | | |
|---|---|---|---|---|---|---|
| **Num Trees** | **Rho** | **0.55** | **0.60** | **0.70** | **0.80** | **0.90** |
| 3 | -0.05 | 0.579 | 0.656 | 0.798 | 0.911 | 0.983 |
| | 0.00 | 0.575 | 0.648 | 0.784 | 0.896 | 0.972 |
| | 0.10 | 0.568 | 0.635 | 0.762 | 0.871 | 0.953 |
| | 0.30 | 0.559 | 0.618 | 0.732 | 0.836 | 0.927 |
| 7 | -0.025 | 0.617 | 0.726 | 0.895 | 0.980 | NA |
| | 0.00 | 0.608 | 0.710 | 0.874 | 0.967 | 0.997 |
| | 0.10 | 0.586 | 0.669 | 0.814 | 0.919 | 0.979 |
| | 0.30 | 0.565 | 0.630 | 0.751 | 0.857 | 0.941 |
| 25 | -0.01 | 0.719 | 0.880 | 0.993 | NA | NA |
| | 0.00 | 0.692 | 0.846 | 0.986 | 1.000 | 1.000 |
| | 0.10 | 0.608 | 0.708 | 0.868 | 0.958 | 0.993 |
| | 0.30 | 0.570 | 0.639 | 0.766 | 0.872 | 0.951 |
| 101 | -0.01 | NA | NA | NA | NA | NA |
| | 0.00 | 0.844 | 0.980 | 1.000 | 1.000 | 1.0 |
| | 0.10 | 0.619 | 0.728 | 0.891 | 0.971 | 0.996 |
| | 0.30 | 0.572 | 0.642 | 0.771 | 0.877 | 0.954 |

Table 5: Theoretical Accuracy Gains: Ensembles of Correlated Classifiers.
The pdf of the beta-binomial model is valid when $\rho \geq \max\{ -p (n-p-1)^{-1} , -(1-p)(n-(1-p)-1)^{-1}\}$. NA - Denotes ρ resulting in a non admissible pmf for the extended beta-binomial distribution

## 5.3 Bias-variance and ensemble gains

Friedman [22] showed that in two-class problems, we may decompose classification error into:

$$P(\hat{y} \neq y) = |2f - 1|\,\mathrm{P}(\hat{y} \neq y_B) + P(y_B \neq y)$$

where *f* is our target function, the last term is irreducible Bayes risk, and the first term is the additive risk of our classifier. We may further analyze the first term by considering the distribution of our model's outputs. If we assume that $\hat{f}(x)$ is approximately normal in distribution, then it is easily shown that, for classification, all we require is $\mathrm{E}\,\hat{f}(x)$ to be on the same side of the decision rule as the Bayes classifier, regardless of its distance with respect to the actual prediction.

In using the ensemble-based prediction $\hat{f}_A(\mathrm{x})$ in place of E[f(x)] and making the reasonable assumption that $\hat{f}_A$ (x) has an approximately normal distribution, Friedman shows that the reduced variance of $\hat{f}_A$ (x) increases the predictive accuracy. The average of many independent classifiers will have a reduction in variance of the order of *1/K*, where K is the number of base classifiers in the ensemble. However, this assumes that the boundary bias of each input x is negative. Points with negative boundary bias for a particular classifier are those points described above that are consistently on the same side of the Bayes decision function. If the bias is positive, then decreasing the variance will have the opposite effect and will actually increase the error. Therefore, it is important to combine classifiers that are diverse in their errors and reasonable with respect to individual predictive ability. Ensembles can make a set of poor classifiers even worse, since a reduction in variance for a set of points with positive boundary bias will reduce the chance of correct classification. Therefore, the base classifiers of an ensemble should be high variance classifiers, as opposed to high bias ones, since the gains in accuracy with aggregation are primarily obtained though variance reduction, as opposed to bias reduction.

Table 6: Variance Reduction with Ensembles. K denotes the number of base classifiers and $\bar{C}$ denotes the average pair-wise correlation among classifiers.

| Base classifiers | Var[ $\hat{f}_A$ (x)] |
|---|---|
| Independent | $1/K\, Var(\hat{f}(x))$ |
| Correlated | $1/K\, Var(\hat{f}(x)) + (K-1)/K\, \bar{C} \times Var(\hat{f}(x))$ |

Based on these results, current research has been focused on creating ensembles with the maximum diversity between base classifiers as possible. Diversity between classifiers will decrease the variance, providing a more stable estimate of E[f(x)]. Measuring diversity, however, has been less than clear-cut. There are many ways to measure diversity of an ensemble and no consistent definition for diversity that has a clear relationship to gains in accuracy [40, 41]. This may be due to the confounding issue of boundary bias, as discussed above. The reduction in variance with independent classifiers does not consistently result in comparable gains in accuracy.

## 5.4 Ensemble diversity

It is clear from the variance decomposition given above that by combining predictions of several classifiers, we will be able to obtain a more stable estimate of E[f(x)]. The stability of the estimate depends on the relationships between the base classifiers, with less or negatively correlated classifiers (i.e.: diverse classifiers) resulting in lower variance. However, this formation is, at best, only tending towards lower ensemble errors. Diversity may be measured in several ways, including pair-wise correlations and non-pair wise entropy base measures, but there is no single way that has a direct relationship with ensemble accuracy. The reason for this is that there is a trade-off

between bias and variance. High bias classifiers tend to be less correlated with respect to making errors, however the interaction between boundary bias and variance may result in ensemble performance that is worse than the average performance of a single classifier. High variance classifiers, though likely to have low bias, will often be quite non-diverse and therefore the ensemble gains will often be modest at best [3].

## 5.5 Breakdown in ensemble gains

In our beta-binomial models, we illustrate the improvement in accuracy over when combining *n* multiple classifiers. Here, we assume average pair-wise correlation ($\rho$) and classifier accuracy (p). From the model-based estimates, adding even moderately correlated classifiers will always reduce ensemble error. Negatively correlated classifiers further enhance the predictive ability of the ensemble.

In Kuncheva et al., the accuracy in multiple classifier systems was explored [42]. Diversity was measured by Yule's Q statistic

$$Q = \frac{P(\hat{y}_1 = \hat{y}_2 = y) - P(\hat{y}_1 = y)(\hat{y}_2 = y)}{P(\hat{y}_1 = \hat{y}_2 = y) + P(\hat{y}_1 = y)(\hat{y}_2 = y)}$$

(Note: $|Q| > |\rho|$), and the limits of majority vote accuracy were derived. In this paper, an example of the breakdown in the ensemble gains is simply illustrated using a three classifier ensemble with ten observations. Each individual classifier has p=0.60, and average Q equal to 0.33, (positive correlation between classifiers). When the pattern below was observed, the ensemble accuracy was 0.40 – a decrease over any individual model.

| | |
|---|---|
| 3 classifiers correct | 4/10 |
| 1st classifier correct | 2/10 |
| 2nd classifier correct | 2/10 |
| 3rd classifier correct | 2/10 |

The average correlation was 0.33 for each pair. From our table, we expect the ensemble accuracy to be increased – from 0.60 to 0.616. Not a large gain, since only three classifiers, but a gain nonetheless.

Examination of the pattern above shows why this ensemble system failed. There were 4 observations out of ten that were classified correctly with any individual classifier. The rest of the observations could be considered *hard to classify* observations. Each individual classifier was able to correctly classify 2/6 hard to classify observations; however each had a different set of observations where the model worked correctly. If each set corresponded to some locality within the input space, and this location was identifiable, then we may achieve 100% accuracy using an ensemble of properly weighted classifiers.

The assumption for the beta-binomial and extended beta-binomial is that the Pr(correct classification) is equal to p (on average). This means that each observation has the same chance of being correctly classified by the model and the areas of error are random. When an observation is hard to classify, the areas of error are not random and the theoretical results do not agree with the actual accuracy observed.

## 5.6 Simulation of the effects of positive boundary bias

If, for a portion of the input space, we have positive boundary bias, the reduction in variance will increase the error rate. Therefore, the asymptotic results illustrated using beta-binomial models all assume that the expected predictive ability of each classifier is > 0.50 over the *entire input space*. If we have hard-to-classify observations, then we will have the gains in reducing variance offset by those regions where there is positive boundary bias.

To examine this, I simulated observations with positive boundary bias using the extended beta-binomial distribution and differing proportions of bias. Figure 8 illustrates the results of this simulation. The top curve is the accuracy gains that we expect under no bias. All observations have the same probability of having a correct prediction for each base learner – there are no hard-to-fit observations. It is clear that the accuracy never decreases, and approaches 1 as the number of base classifiers tends to infinity. This is the situation that illustrates the theoretical justification of why ensembles work. However for the simulated situations where there are a proportion of observations

with positive boundary bias, the effect of combining trees can actually be negative. If a large proportion of the input space is biased, then reducing the variance will actually increase the error rate. The bottom curve illustrates the situation where the average accuracy is greater than 50%, however when combining these classifiers we end up with an ensemble with much lower accuracy. Many of the observations are biased; therefore decreasing the variance has a deleterious effect on error rates. For moderate levels of bias, the gain is in combining the first 3-9 trees and then the accuracy is observed to decrease as more trees are added. This is consistent to what has been observed in real data, where the gain is in the first few members of the ensemble.

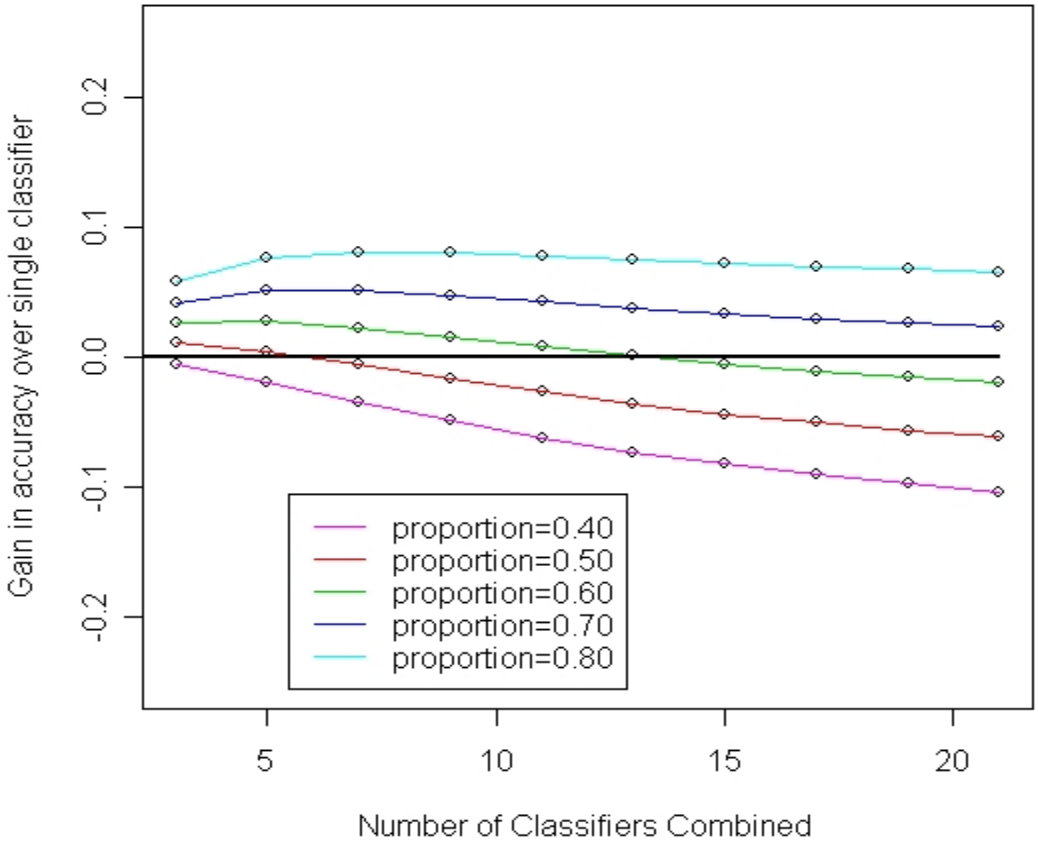



Figure 8: Loss of accuracy due to averaging biased classifiers. The proportion varied is the proportion of unbiased observations such that Pr(correct)=0.85.

# Chapter 6

# Building Ensembles

Ensembles have been constructed in multiple ways. Usually, the base classifiers are constructed to be as diverse as possible, while maintaining a reasonable level of accuracy. Diversity is introduced in many ways, for example:

- Random selection of features
- Perturbation of the data set
- Re-weighting of the data set to focus on misclassified observation

Randomly selecting features out of the set is a staple of the Random Forest algorithm [10], as detailed in Chapter 3. Ho [33] also uses a random feature selection. By considering only portions (mutually exclusive or with overlap) of the input space, different classifiers may be constructed. It is expected that classifier accuracy will vary, since some selected subsets may contain no informative features, while others will contain highly informative features. However, if there are many base classifiers in the ensemble, the effect of a few poorly fitting classifiers on the ensemble decision will be negligible. Diversity will also vary since many data sets contain highly correlated features.

Perturbation of the data set is another method by which different classifiers may be constructed. Perturbation often involved bootstrapping the original data set B times, and creating B classifiers using the perturbed data. If the data set is representative of the population, then this process mimics the random sampling from the parent distribution. The classifiers built are often diverse due to the selection and estimation biases inherent in variable screening, model selection, and parameter estimation.

Although bootstrapping can be considered a re-weighting of observations, the third type of diversity is based on a special re-weighting. This type of ensemble is used in boosting algorithms. Boosting creates diverse models iteratively. Iteration 1 fits a weak model (often a decision stump) to the data, all observations equally weighted. The next iteration then re-focuses on those observations that are misclassified by the initial model. This process continues for many iterations (usually 50 or more) and then the final decision is based on the weighted vote of all of the classifiers. Boosting has been shown to be a highly successful ensemble technique, often reducing both bias and variance, however it suffers when a hard-to-fit observation is repeatedly focused on, regardless of whether it is an outlier or not.

In this chapter, two new methods for creating ensembles are explored. The first, **C**lassification by **E**nsembles of **R**andom **P**artitions (CERP) creates diversity in a similar way to Random Forest, however the features are partitioned at the initial stage of analysis and this remains fixed. Therefore, the ensemble is made up of k classifiers, one from each of the k disjoint feature sets. Multiple ensembles could be constructed, each with a different partitioning scheme, and the ensemble of ensembles combined into a final classification. The advantage of this approach is that any base classifier may be used, including decision trees (CT-CERP) [1] and logistic regression models (LR-CERP) [45].

The second method of creating an ensemble of classifiers is quite different than the first. Diversity is achieved by using different classifiers, instead of different subsets of features. Different classifiers will create a set of diverse decision boundaries. The combination of these classifiers is explored, with simple averaging and weighting methods compared. In addition, local weights are derived which allow the ensemble weights to be varied depending on the location of the test point in the feature space. This

is another novel method which uses bootstrap estimates of bias and variance at each training point X to weight the different classifiers.

As mentioned previously, there are several methods available for combining classifiers using order statistics [64-66], majority vote [1, 10, 45], and weighted averaging [2, 59, 74]. We examine a subset of these approaches in the next section.

## 6.1 Global methods of combining

Pepe shows that a linear combination of markers (or in this case classifiers) is optimal under Neyman-Pearson lemma [50].

Linear combination of K classifiers: $\sum_{i=1}^{K} w_i C_i$

where Ci is the decision or prediction of the ith classifier in the ensemble.

If the predictions are multivariate normal, then the optimal weights are the LDA weights [59]. We then may use these K variables as inputs into a linear discriminant anslysis to derive weights for each classifier. Similar to the LDA approach, we may estimate weights using logistic regression, under the logit assumptions, to maximize the likelihood. Given the potential for over-fitting, a bootstrap or cross validated estimate of each observation's predictive score is used for all classifiers.

**The Neyman-Pearson Lemma:**

Let θ` and θ`` be distinct fixed values of θ so that Ω={ θ`, θ``} and let k be a positive number.

$$L\frac{(\theta`; X_1, X_2, ...X_n)}{(\theta``; X_1, X_2, ...X_n)} \le k \text{ for each point } (X_1,\ldots X_n) \in C \text{, otherwise } \in C^*$$

The C is the best critical region of size α for testing:

Ho: θ= θ`
Ha: θ= θ``

If we use this lemma, then the best test for each input point is based on the likelihood ratio. If we fix the FPR (size α), then for this fixed FPR (fixed specificity), the test with

the most power (i.e. highest TPR or sensitivity) is based on the likelihood ratio. Therefore, when we have K prediction scores from K classifiers, then the uniformly most powerful test is based on the linear combination of $Y_{1,}\ldots,Y_{k.}$ The risk score is some monotone increasing function of $L_{\beta}(Y)$, which approximates some transformation of the likelihood ratio.

## Simple Average

Simple averaging has been shown to be a robust method to combine classifiers and under independence and equal performance assumptions across base classifiers, is optimal [24, 25, 59, 64] with respect to overall error. Using each classifier, the ensemble prediction is:

$$\sum_{i=1}^{K} w_i C_i \text{ where } w_i = \frac{1}{K}$$

## Weighted averaging

When we have a set of K classifiers with varying levels of accuracy, a weighted average may prove to be beneficial. It makes intuitive sense that a classifier with a high level of generalization accuracy (assessed via cross validation or bootstrap) has a higher weight assigned to it, given that we are more confident in its prediction. If we directly use the estimated accuracies of the classifiers, we can assign weights to each classifier without fitting a model, using the optimal weights constructed in [24,25]. The weights are constrained to be non-negative and have a unit sum. The construction of weights should be based on an out-of-bag or test set assessment of accuracy to avoid over-fitting and are estimated as:

$$w_i = \frac{1/(1-p_i)}{\sum_{k=1}^{K} 1/(1-p_k)}$$ where $p_i$ is the estimated accuracy for classifier i; i=1,2,…,K

Under independence assumptions, the weights above are optimal [24] with respect to minimizing error. If classifiers are correlated, but have equal pair-wise correlations, this optimality also holds. This weighting scheme penalizes uneven performance across classifiers more heavily than weights derived from each individual classifier's proportion of the total accuracy, and yields very similar weights to the normalized Adaboost weights. Differences may be observed if the range of individual performances is very high. Determination of the optimal weights is difficult, given the varying correlations between classifiers and the range of accuracies for a particular problem. In addition, computation of optimal weights may have little impact on the overall accuracy compared to a simple average, unless the range of accuracies across classifiers is large [24, 25].

## Other methods for combining classifiers:

### Low Bias

Low-bias combining relies on the fact that averaging is generally a variance reduction tool in many ensemble systems, therefore we may deal with the bias component separately and then seek to reduce variance. Valentini [67, 68] successfully applies this approach to SVM classifiers by first exploring the lowest bias combination of tuning parameters, and then bagging the data to produce hundreds to SVMs built under the selected tuning parameters.

### Low correlation

Low-correlation combining selects the three classifiers with the lowest pair-wise correlation. First, the two classifiers with the lowest pair-wise correlation are selected, and then third member is selected if it has the lowest average correlation to the two already selected. The reasoning behind this is that the variance reduction is impacted strongly by diversity, thus we attempt to create the most diverse ensemble [41]. The

drawback to this approach is that low correlation is often achieved by poorer performing classifiers - an example of the tradeoff between diversity and performance.

## 6.2 Classification by ensembles of random partitions (CERP)

A global ensemble method was developed [1] based on randomly partitioning the feature set to create k disjoint sets of predictors of roughly equal numbers of predictors. The strength of this method is that any base classifier may be used in the ensemble. CERP, our initial ensemble system, is based on optimal classification trees. LR-T CERP constructs logistic regression trees. Further details, as well as the performance of this method across many data sets may be found in [1, 45]. The number of partitions was set to be N/j, where j (j=1,2,….) is found based on a cross-validated search. The optimal threshold is based on a grid search.

Through partitioning of the feature set, we obtain sets of predictors that may be quite diverse. In addition, a big advantage of CERP is that we do not need to handle the entire data set as a whole; we may process large data sets by analyzing smaller dimensions in parallel. Although for 1,000 predictors this is not much of an advantage, for p as large as 100,000 predictors it promises great computational efficiency. In addition, CERP provides a way to integrate the results across several platforms, while keeping each model platform-specific.

The results of CERP have shown it to be comparable or superior across a wide-range of data sets, compared to classifiers such as Random Forest, DLDA, SVM, LDA and Logitboost. In addition, CERP achieves better balance with respect to sensitivity and specificity than Random Forest, which is a strength in many classification studies when there is a large unbalance. Many classifiers are naturally biased towards the majority class.

## 6.3 Local combining of classifiers

Local weighting allows the set of classifier weights or expert classifiers to vary, depending on what input point we are examining. Classifiers are built on the entire data set, thus retaining overall performance measures; however the aggregation of these classifiers is weighted to allow higher weights for those classifiers that are both locally unbiased and low variance. Classifier weights are determined using a bootstrap evaluation of bias and variance of each classifier at the training input point, according to Domingos' decomposition of classification error (see section 4.1.3). If an observation is easily and consistently classified, then the prediction based on D training sets is not likely to vary significantly. If an observation is harder to classify, then the classifier may exhibit higher variance or higher bias in that region. Examination of several good base classifiers can highlight whether this behavior is consistent over all classifiers, or the result of one or two poor classifiers for that region of the input space. Location of the test observation is determined by a nearest neighbor approach. The main advantage of this approach is that it applies variance reduction through averaging classifiers only in areas of the input space where variance reduction is beneficial. In areas of the space where high variance is important (i.e. input points where all classifiers are biased), there is no aggregation and the highest variance biased classifier is applied. The main difficulty of this approach is the determination of the neighborhood for each test observation.

### Neighborhood of the test point

The location of the test observation is determined using the nearest neighbors. Although there is no a priori number of neighbors able to be justified, we examined k=1 and k=3 in both the original feature space and also in the first 5-10 dimensions defined by the principal directions using SVD on the standardized feature matrix. Given the high dimensionality of the datasets under consideration, we consider the main directions of the data with respect to variability to better determine the nearest neighbor. Other methods of nearest neighbor could easily be considered.

**Schema for Local weighting:**

| | |
|---|---|
| i=1,2,…n | denote the n observations in the dataset |
| b=1,2,…B | denote the B bootstrap samples generated |
| j=1,2,…p | denote the p classifiers used in the ensemble |
| y | denote the true class (assuming noise is zero) |
| $\hat{y}_{i,b}$ | denote the predicted class for obs i, in bootstrap sample b |

1. For the training set, create B perturbed samples via bootstrap
2. Examine the **stability** of each classifier using the OOB samples as test samples.
3. Stability $s_{i,j}$: the proportion of times classified correctly by classifier j

$$s_{i,j} = \frac{\sum_{b=1}^{B} I_{(y_i = \hat{y}_{i,j,b})}}{B}$$

4. Estimate Bias for classifier j at X=x corresponding to obs i :

$$B_{i,j}(X=x) = \begin{cases} 1, s_{i,j} < 0.50 \\ 0, s_{i,j} \geq 0.50 \end{cases}$$

5. Estimate Variance for classifier j at X=x corresponding to obs i:

$$V_{i,j}(X=x) = 1 - s_{i,j}$$

6. Derive weighting scheme for each observation based on the stabilities of the $r \leq p$ (**selected unbiased** or full set of ) classifiers.

$$w_{i,j} = \frac{1 \big/ \max(0.05, V_{i,j})}{\sum_{j=1}^{r} 1 \big/ \max(0.05, V_{i,j})}$$

# Chapter 7

# Individual Classifiers – Performance and Decomposition

The performance of each classifier is examined in four of the genomic data sets both by accuracy estimates, as well as estimated bias and variance. Random Forest, an ensemble approach, is included in this examination, as well as the simple average of all classifiers. An outline of the procedure used is included in the Appendix (B). Included in this assessment is the ensemble-based prediction of all classifiers, computed via simple averaging of individual predictions. The impact of dimensionality is assessed by taking only a subset of the total set of predictors available. The subset of predictors is selected on the basis of the BW ranking, therefore it is expected that the amount of information added will diminish with each new variable added, given that there are seemingly no joint effects or interactions that add large amounts of information.

## 7.1 Imprinting data

The imprinting data set has many highly correlated variables by design. Since each window was constructed to be overlapping segments of the flanking region of each gene transcription start site (TSS), there is high correlation among elements of varying window sizes, as well as correlation between repetitive elements that are highly similar biologically.

Figure 9 shows each individual classifier's and ensemble-based performance as the number of retained predictors is varied from 5 to 150 predictors. When the set of predictors is small, most of the information is overlapping due to the large clusters of highly correlated features that exist in the data set. Many of the top ranks are taken up by ALU elements, thus representing such a cluster. Therefore, it is not surprising to see that the information gain continues for all classifiers up to about 100 predictors. Random Forest and SVM-RBF clearly dominate across this range of p, with overlapping and sometimes crossing accuracy curves. Linear-SVM as 3-NN are similar in performance, with a consistently lower accuracy than RF and SVM-RBF. DLDA shows the most movement across this range, with accuracy increasing to about 80% from a starting point of less than 70%. CART appears to be consistently the worse performer in this set, with accuracy that is quite low. At about p=100, the amount of information being extracted clearly levels off for many classifiers, indicating that the addition of more predictors has a marginal impact on performance.

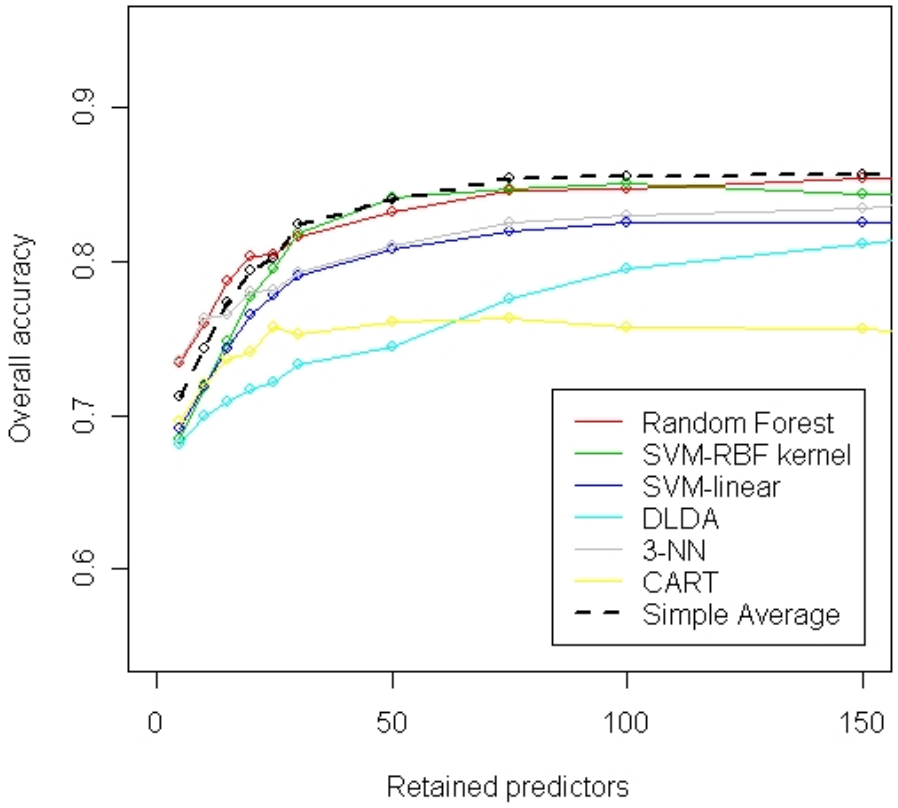



Figure 9: Individual Classifier Accuracy for the Imprinting Data for p=5 to 150 predictors.

As the number of predictors is expanded into the hundreds, differences between classifiers begin to emerge. Most classifiers maintain the total accuracy achieved in the earlier stages (when the information leveled off). For CART and Random Forest –noise included in the feature list is easily handled due to the greedy way it adds variables, therefore it was expected that moderate dimension (and likely increasing amounts of noise) would have little impact on performance. SVM-RBF and 3-NN both decline in accuracy with increasing dimension. Linear SVM stabilizes at p=500 predictors and retains accuracy. Interestingly, DLDA shows the same momentum as before, with

accuracy steadily increasing as variables are added to the weighted sum. When p reaches 1,000 predictors, DLDA's performance is almost the same as that achieved by Random Forest.

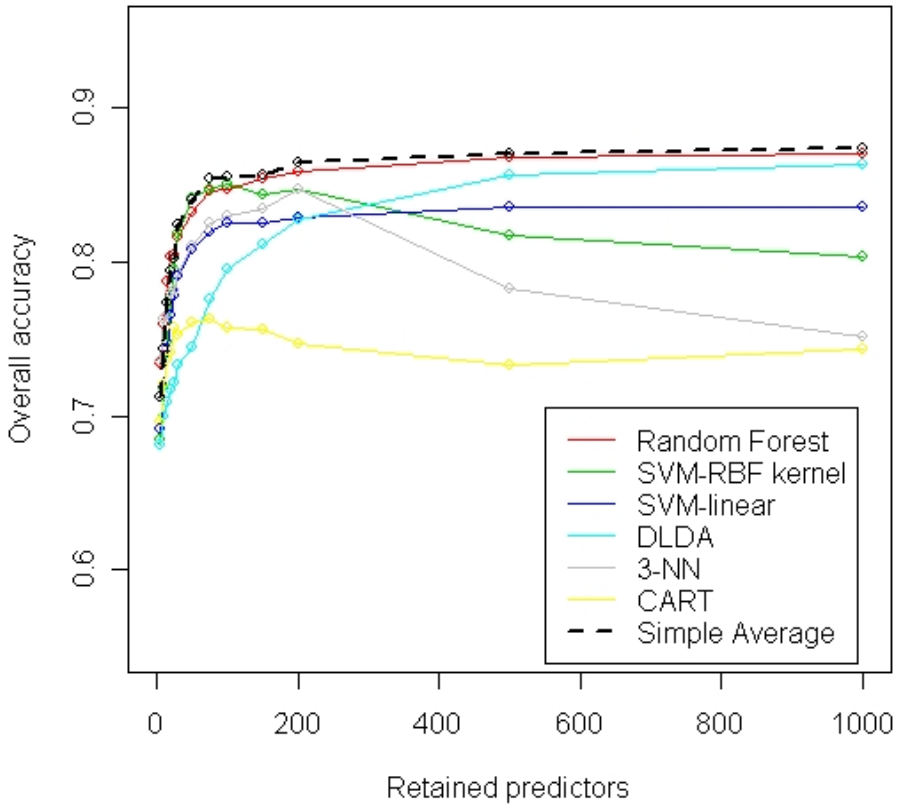



Figure 10: Imprinting Data Set. Individual Classifier Accuracy for p=5 to 1,000

The reduction in accuracy for both SVM-RBF and 3NN as dimension is increased was explored. The initial conclusion was that the high dimension coupled with the complexity of the classification yielded high variance. However this is not the case. Figure 11 shows the average bias for each classifier. As it is clear, more than 90% of the error is accounted for by bias. Even after accounting for the fact that some of the bias term is noise (we bundle noise and bias together in this process), the bias is generally increasing as dimension increases. For SVM-linear, RF, CART, and DLDA, the level of

bias remains quite flat across the range of p, after the initial steep transition from the under-fitted models (high bias) when p is low. For SVM-RBF and 3-NN, bias is low when the dimension is reasonable, yet increases after about 200 predictors.

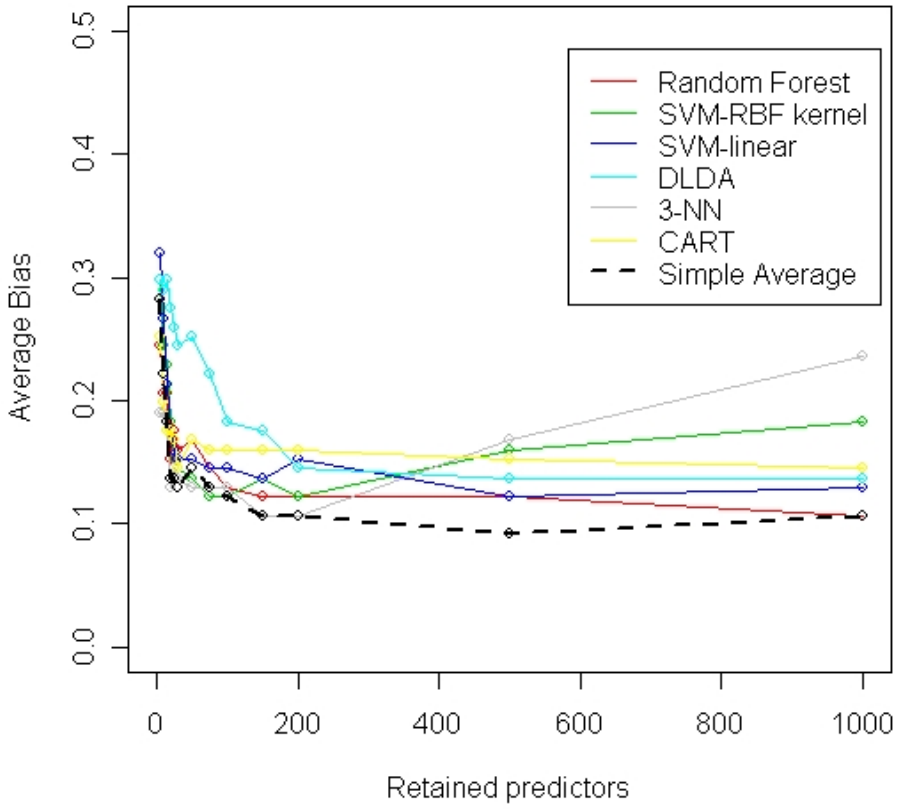



Figure 11: Classifier Average Bias for the Imprinting Data

The RBF kernel used in SVM may be sensitive to dimension for many reasons. First, the form of the RBF kernel is essentially squared Euclidean distance, so the curse of dimensionality is an issue. The curse of dimensionality means that the observations in high dimensional space are extremely sparse. The distance between the two closest observations is not very different than the distance from the two furthest observations when the dimension is high, so the concept of closeness has less meaning. The same phenomenon occurs with kNN- another measure that uses Euclidean distance. Second, at

a fixed level of gamma (which controls the spread of influence of each support vector), an increasing dimension means that the control of each support vector diminishes. Third, as dimension increases, there is likely a large increase in the amount of noise directions. Since SVM and kNN consider the entire space, noisy and meaningless directions could obscure more informative directions. The impact of this high dimension is that the bias term of both classifiers increases as p increases. This can also be illustrated by examining the average number of support vectors used in the training set. For p=100, there are about 55 support vectors on average, for p=1000, this number almost doubles to 95. This means that the dependence on the training set is very high when p is large. However this does not result in increased variance, as expected with increased dependence on the training set. What appears to happen is that as dimension increases, the concept of distance becomes meaningless, and the support vectors do not have the reach of influence for many of the points. SVM does not ignore any noisy or irrelevant directions in the data. Thus, the decision for a subset of observations is close to zero (no decision), resulting in bias. Trials to improve this by changing the gamma or cost parameters (large C should minimize bias) for p=1,000 did not alter the results in any meaningful way, and this bias remains the dominant factor in test error. There is no making up for the large bias through tuning in high dimensions. Therefore over-fitting the training set with the RBF kernel through too small a width parameter can cause high variance in test sets, but high bias may also occur in high dimensions due to the local nature of the kernel.

Evaluation of the variance yields less information, since the errors are mostly derived from bias terms. Of interest, is the fact that SVM-RBF is observed to have similar net variance (though slightly higher) to Random Forest. Since Random Forest is an ensemble of classifiers, the low bias and low variance achieved by SVM-RBF makes SVM-RBF, with proper variable screening, a very competitive classifier. The lower variance is due to the internal variance minimization – namely the margin maximization that occurs in the procedure.

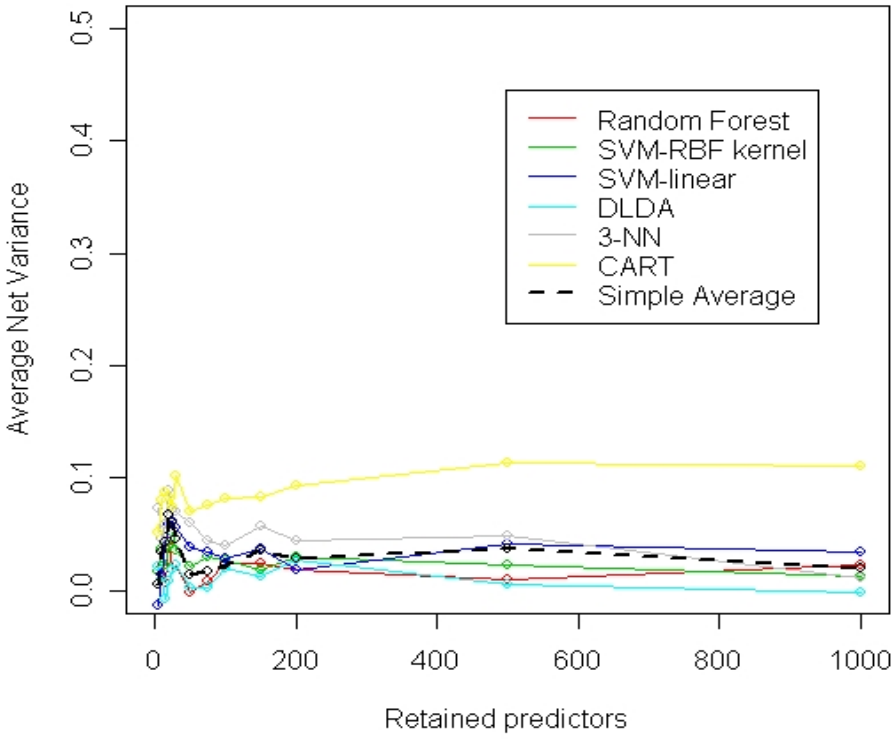



Figure 12: Net Variance of each classifier for the Imprinting Data. Net variance added to bias is equal to classifier error.

The training set error is a highly biased estimator of the generalization performance. As dimension increases, for most classifiers the training error tends to get closer and closer to 1.0. By adding information to the classifier, we expect to never lose accuracy with respect to the training points. However, if we fit the training set too well, we are finely tuning our classifier to the nuances of those specific observations. This has the tendency to over-fit the data, and thus performs poorly in new testing sets. Figure 13 illustrates the training minus test error for a few classifiers of interest. The DLDA classifier has a training set accuracy that increases as p increases and then flattens out, indicating no new information. The test accuracy continues to climb steadily, indicating that the classifier continues to learn and thus bias continues to drop. The difference between accuracies (training and test) is an indication of the amount of optimism in the

in-sample estimate. SVM-RBF on the other hand, has training accuracy that rises and stabilizes to 1.0, meaning that the hyper-plane in expanded feature space fits the training set perfectly. However, as p increases past about 200 predictors, the testing accuracy decreases, meaning that the classifier is over-fitting in the higher dimensional spaces. Linear SVM, on the other hand, has a flat training and testing accuracy, which indicates that this classifier does not increasingly over-fit as p increases. It is interesting to examine the spread between training and testing accuracy. DLDA has a narrow difference, while SVM has a much larger band of difference.

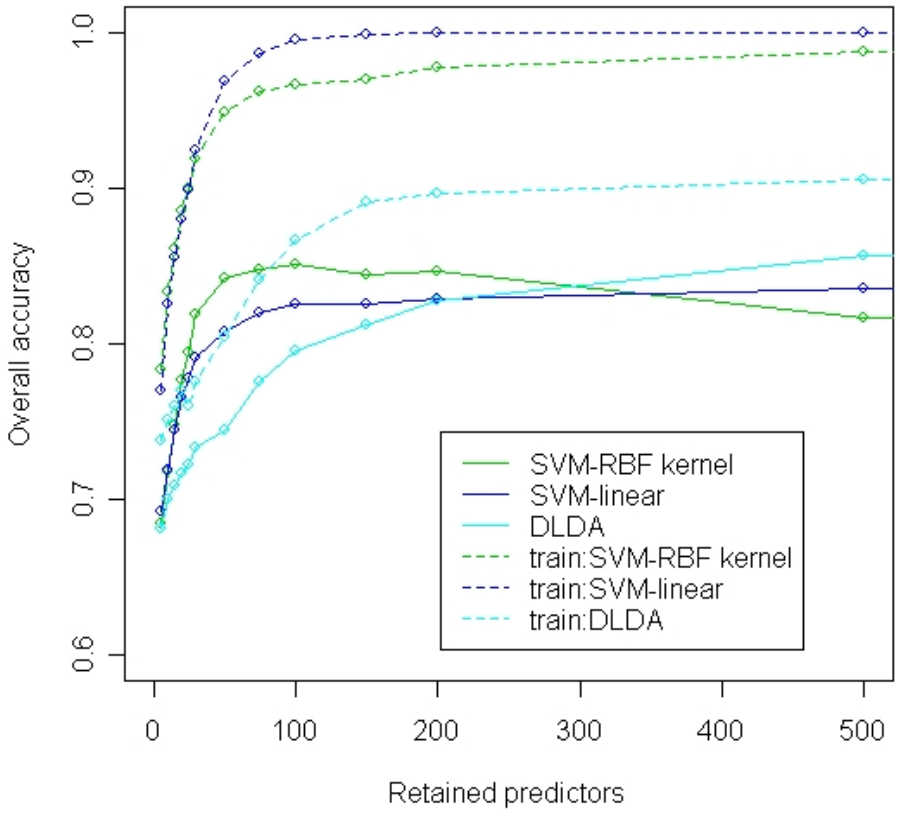



Figure 13: Classifier training vs. test accuracy in the Imprinting Data

## 7.2 Colon data

The colon data set has some interesting aspects. First, DLDA is again extremely sensitive to the number of predictors retained. The optimal number of predictors for this classifier appears to be about 25. Initially, there is a steep increase in accuracy for DLDA from p=5 to 25 predictors, showing the information gained by taking the weighted vote of a larger set of features. After p=25, the accuracy appears to stabilize for many of the classifiers. For DLDA, however, the increasing dimension is detrimental and its accuracy steadily drops. As the dimension grows larger, the performance of this classifier declines rapidly without stabilizing. The other classifiers appear quite immune to increasing dimension, or even display slightly increasing accuracies for large p (SVM-linear and 3NN). Linear SVM suffers from higher bias, causing the overall accuracy to be lower than its more flexible counterparts.

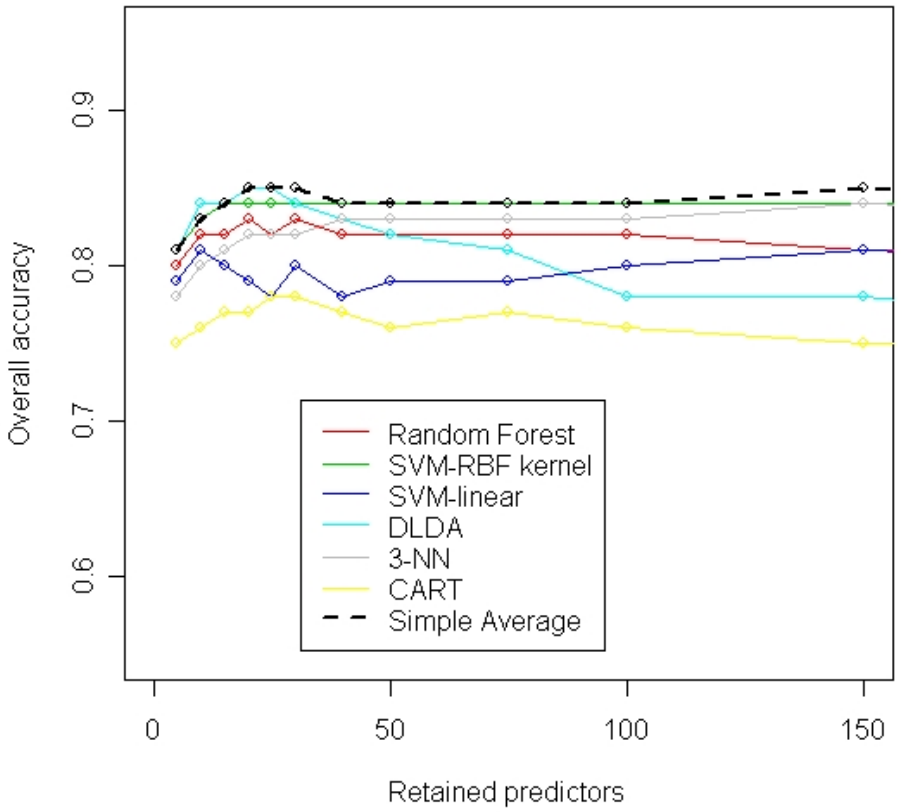



Figure 14: Colon data set. Classifier accuracy across varying dimensions

Examination of the bias (again, the dominating term in the error) yields a noisy trend for all of the classifiers (Figure 15). The bias term does not decrease monotonically with each new predictor being added, which may be an indication that the univariate ranking is not reflective of the multivariable importance of each predictor. DLDA has the lowest bias at p=15-25 predictors, but then displays increased bias as p is increased.

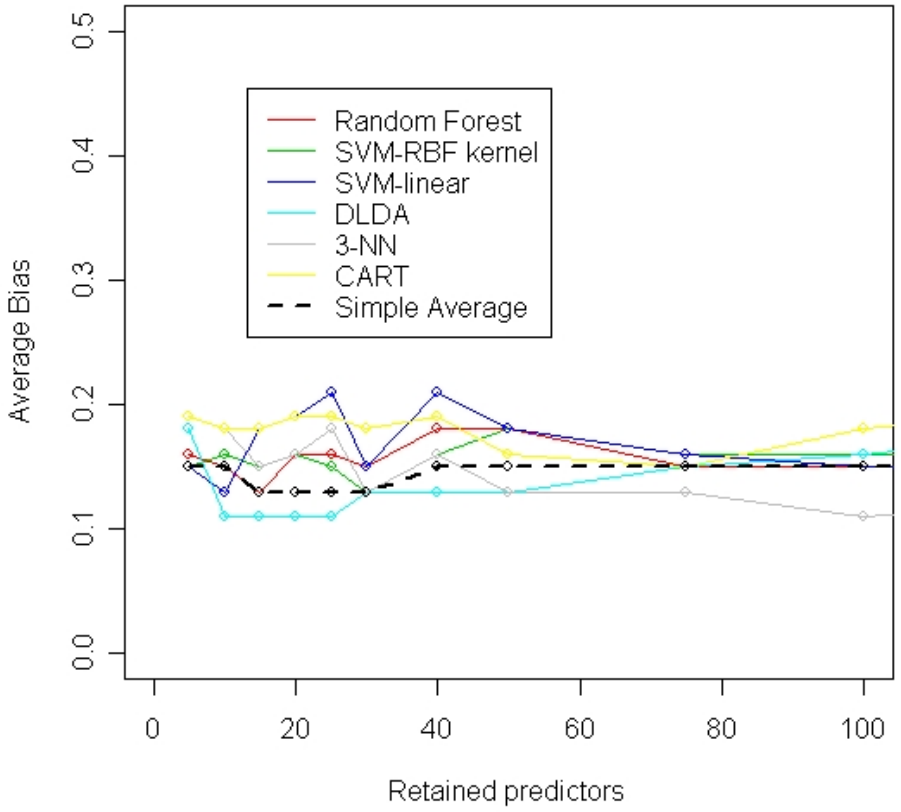



Figure 15: Colon data set. Average Bias across varying dimensions

In Figure 16 the reason for the bias in DLDA becomes clearer – as p is increased past p=30, the training error actually *increases* – there is no learning taking place. The DLDA classifier is doing a worse job fitting the training set. As a result, the test set is becoming increasingly biased. Again, the band between test and training accuracy is narrower than the band corresponding to the SVM classifiers.

SVM with RBF kernel has a smaller difference between test and training set accuracies when compared with SVM-linear. This is an indication that SVM is resistant to over-fitting in some situations, even when a more complex feature space is used. CART is a poor classifier in this data – it displays both high bias and high net variance. Compared to Random Forest, which are fully grown CART trees, the bias for CART

(with minor regularization since tree splits are stopped when the node size reaches 10) is 0.19 (vs. RF's bias at 0.16). Of course we may attribute this to the fact that the trees in RF are fully grown.

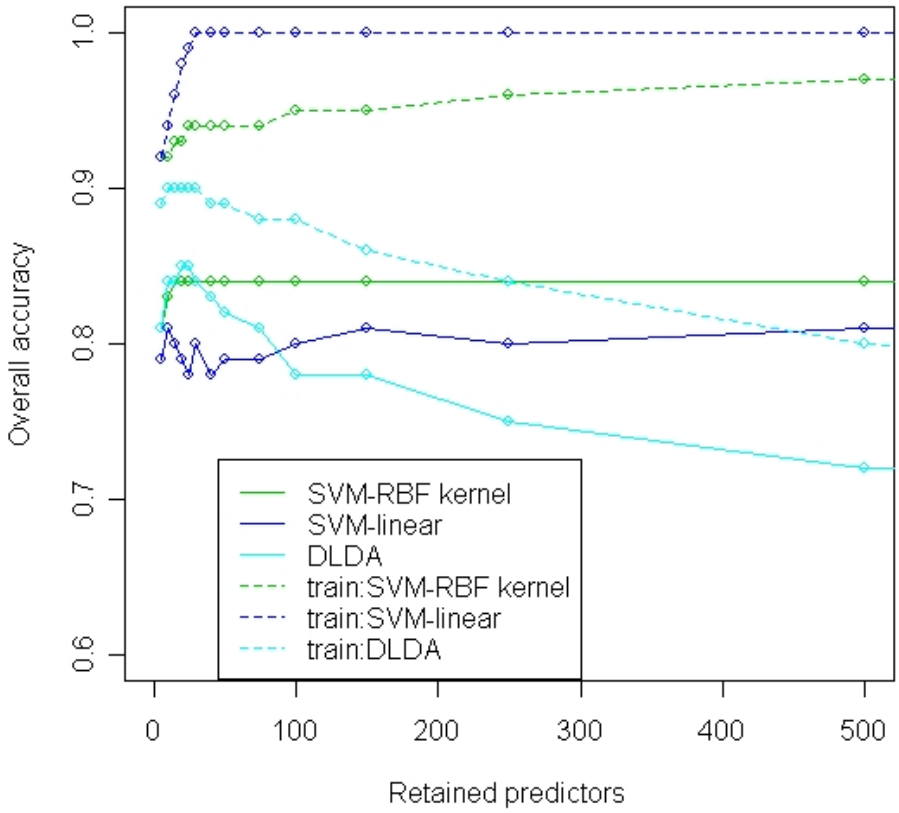



Figure 16: Colon data set. Training and test accuracy across varying dimensions

## 7.3 Estrogen data

The estrogen data set has 332 predictors in total therefore it is representative of a much smaller data set than the other studies. However, similar to the others, there is a leveling off of accuracy after about 75-100 predictors – most of the information gain comes with the addition of the first 10-20 highest ranked features (Figure 17). For the non-linear classifiers, the increase in accuracy and decrease in bias is most steep in the

first 50 predictors added, while the same set of predictors does not yield the same information gain for the more rigid classifiers such as DLDA and linear SVM (Figure 18). This indicates that the decision surface may be more complex, and the linear classifiers are not representing it as well as the more flexible ones. For all classifiers, the bias dominates the error term, though CART shows a high level of unbiased variance, which makes it both a high bias and high variance classifier in this situation. The simple average of classifiers shows extremely low bias across all levels of p, and appears to be much smoother and more stable when dimension is changed. Since the bias is lowered, the good variance is lowered accordingly, while the bad variance (unbiased observations varying around the true class) is in the middle range of all of the classifiers.

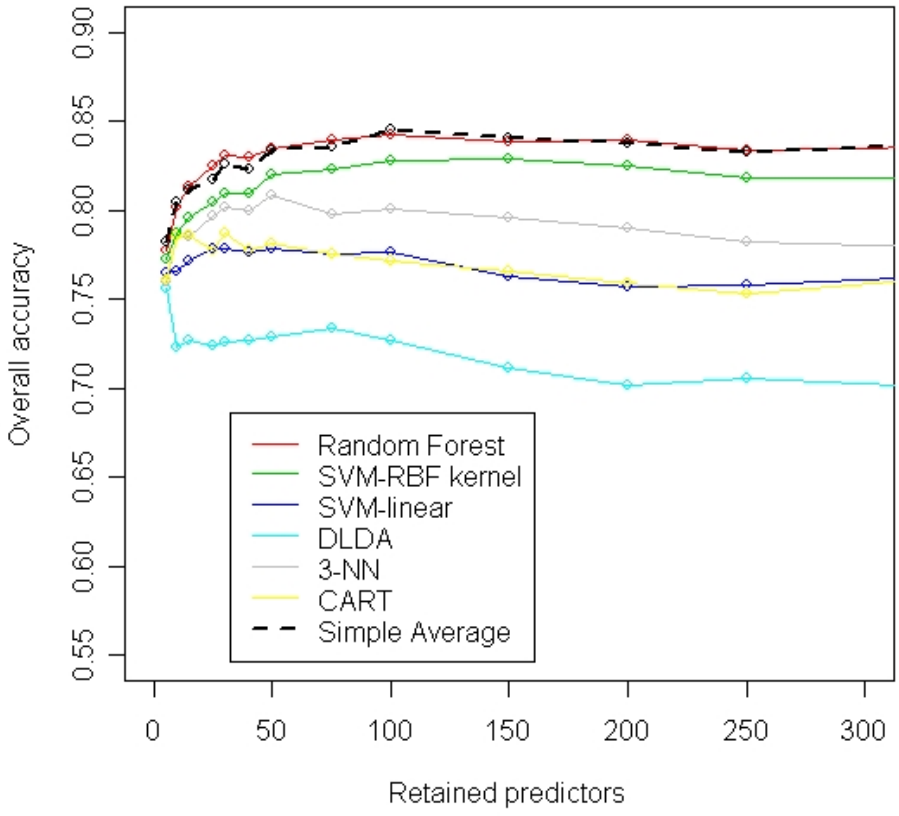



Figure 17: Estrogen data set. Classifier accuracy across varying dimensions

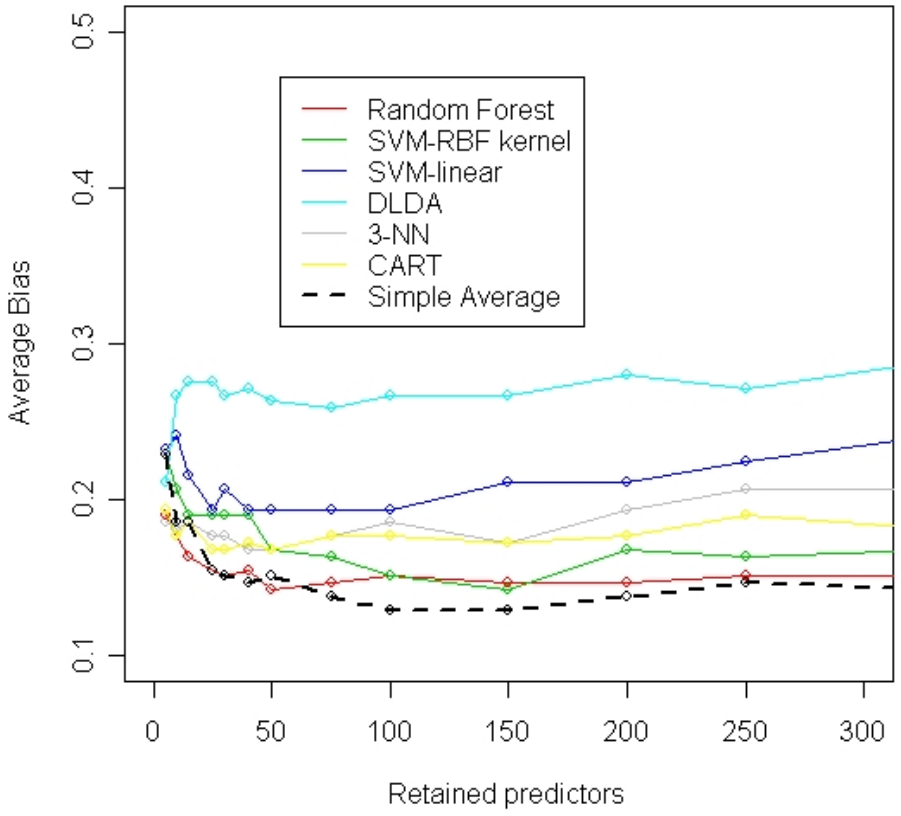



Figure 18: Estrogen data set. Average Bias across varying dimensions

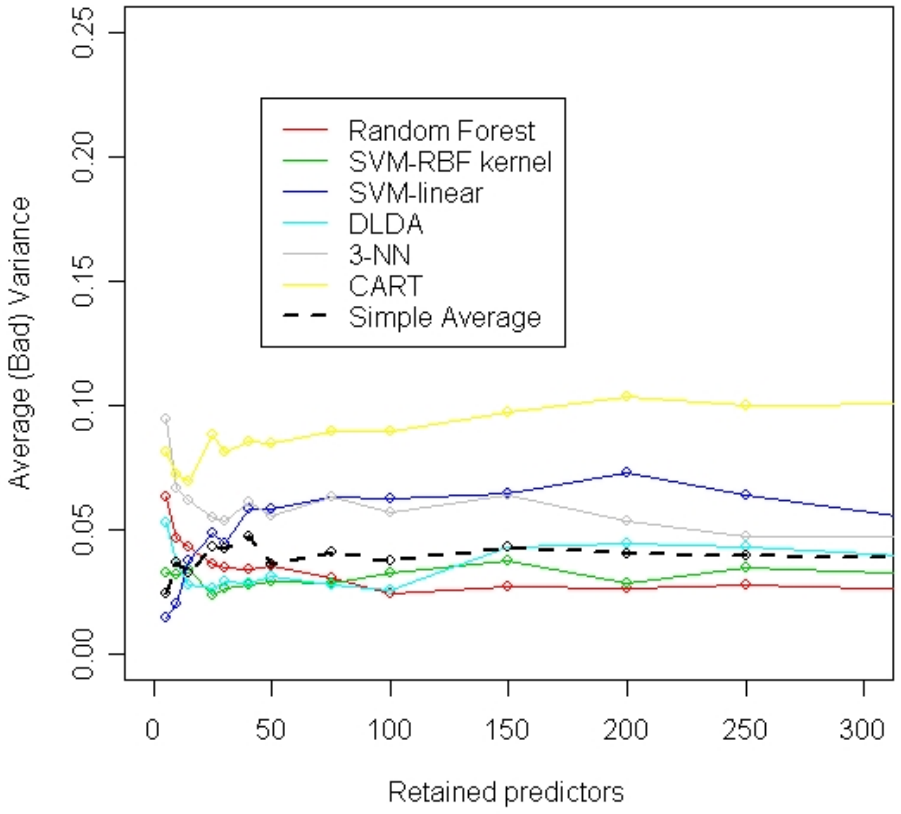



Figure 19: Estrogen data set. Unbiased variance across varying dimensions

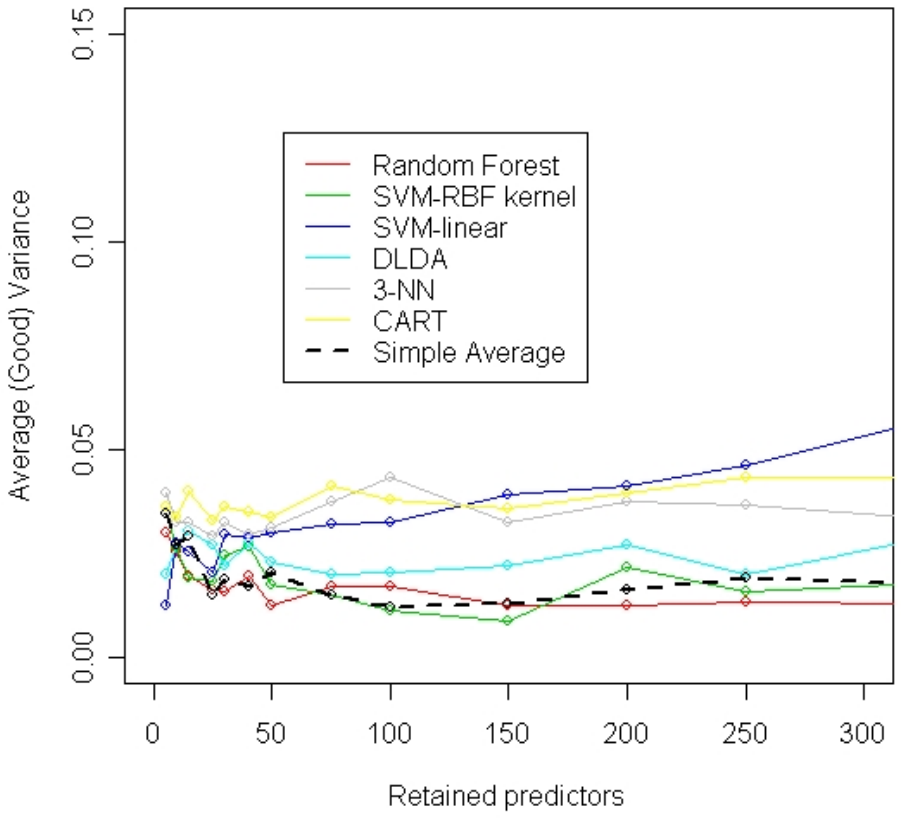



Figure 20: Estrogen data set. Biased variance across varying dimensions

Figure 21 depicts the effect of dimension on the training and testing accuracies for each classifier in the Estrogen data set. DLDA shows a small difference in accuracies between the training and test estimates, which indicates a low level of over-fitting. Since the bias in the DLDA classifier dominates, this is expected. The linear SVM classifier seems to have a slight increase in over-fitting as p is increased. The SVM-RBF classifier has low levels of over-fitting, thus evidence that the regularization that is internal to SVM can provide some protection against over-fitting.

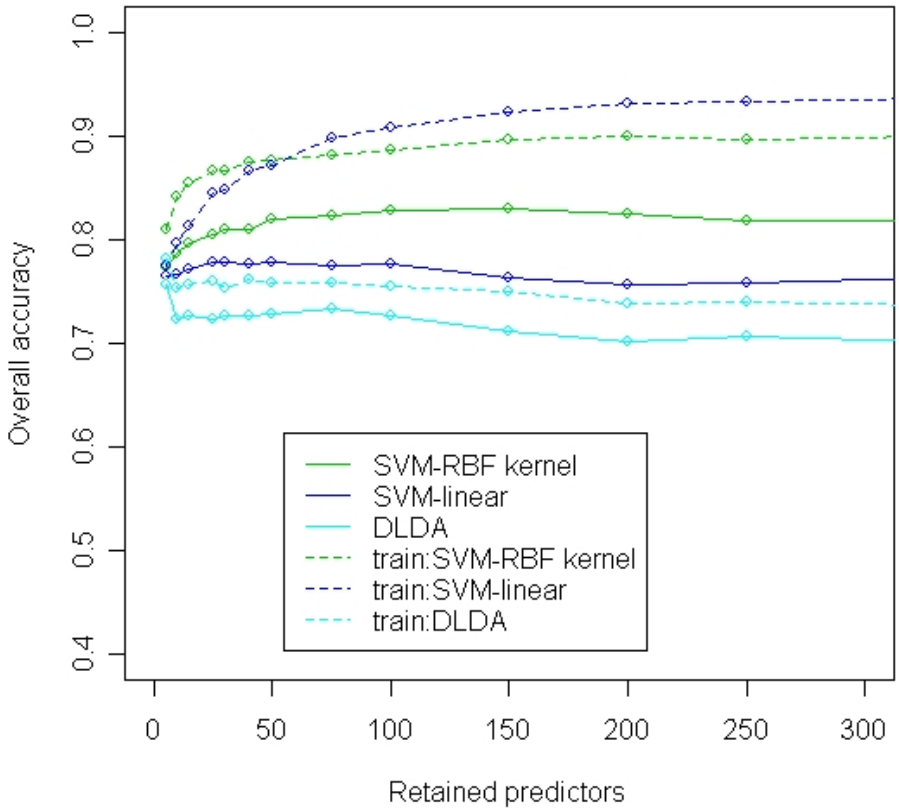



Figure 21: Estrogen data set. Training and test accuracy across varying dimensions

The difference between training and testing accuracies is once again very narrow across p for DLDA, and wider for SVM. All classifiers exhibit a lack of over-fitting that increases as p increases.

## 7.4 Prostate data

The prostate data has similar trends as the other data sets. DLDA again proves to be extremely sensitive to dimension, with a sharply reduced accuracy after 100 predictors. Examination of the training set performance yields the same diminished performance in high dimensions, which indicates that the classifier is too rigid as p is increased – it cannot learn the decision function under the strict assumptions of uncorrelated features and equal variances. For the other classifiers, performance is quite flat across the range of p and there appears to be little over-fitting in the data, even as p gets very large. Linear SVM appears to have a high unbiased variance (bad variance) in lower dimensions, causing lower accuracy, but then achieves high accuracy in higher dimensions as this variance is reduced and bias is also lowered. Thus linear SVM is a low variance and low bias classifier in this application.

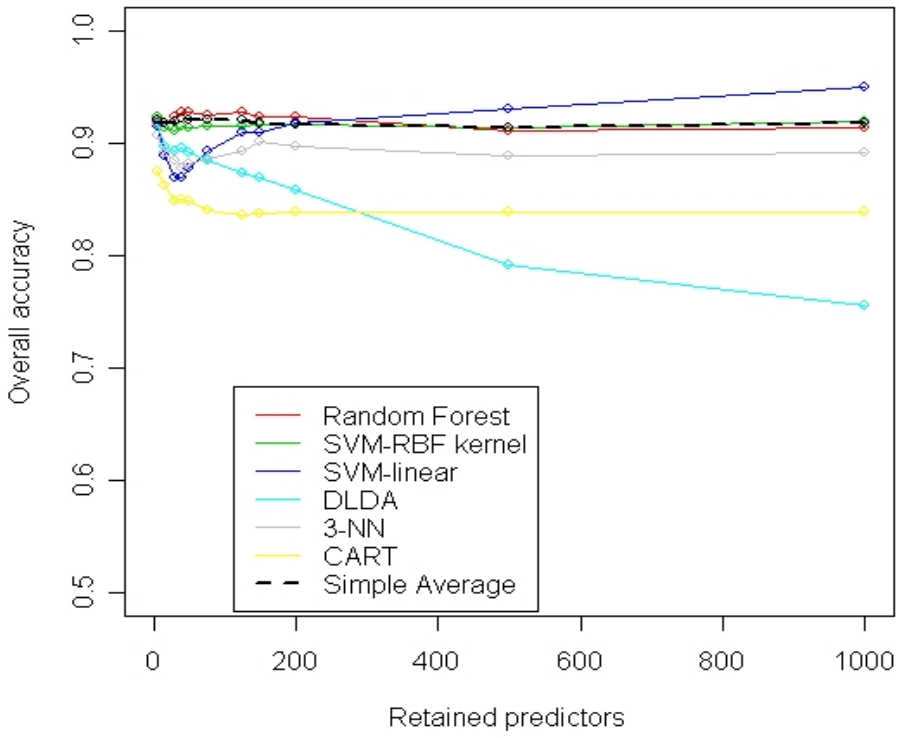



Figure 22: Prostate data set. Classifier accuracy across varying dimension

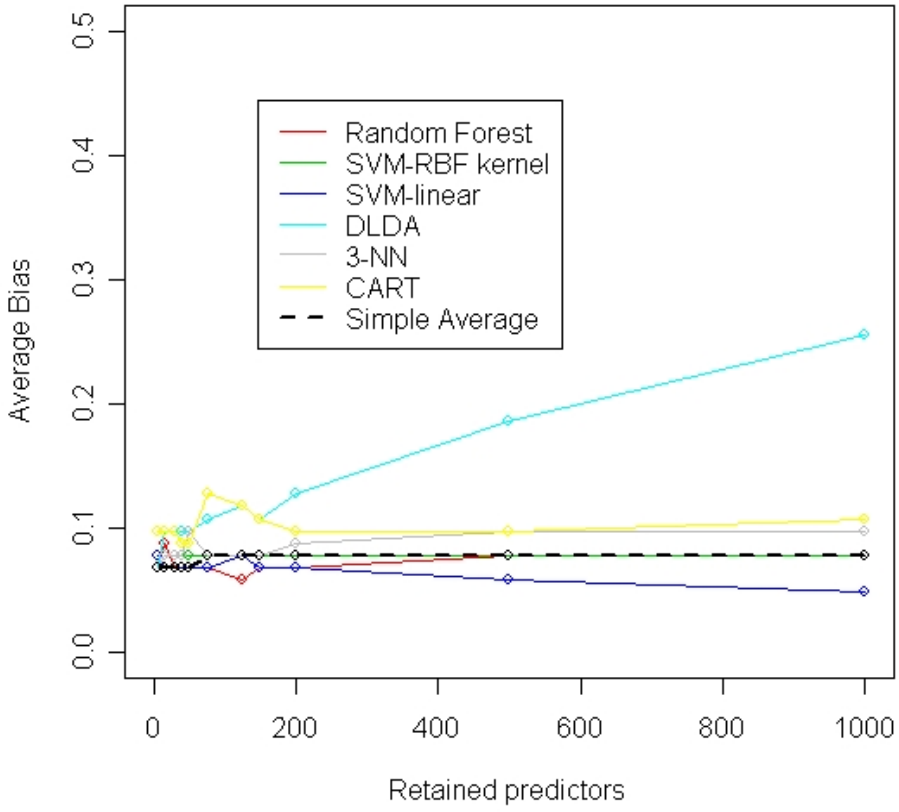



Figure 23: Prostate: Average Bias across varying dimensions

The bias term is again the dominant contributor to error. DLDA has a sharply rising bias, while the other classifiers are flat or have slightly lowered bias across the range of p. For linear SVM, the bias is low at small values of p, however the variance of the unbiased observations is large, thus causing error to be markedly higher in this range of p. As more variables are added, this variance decreases and the overall error thus decreases.

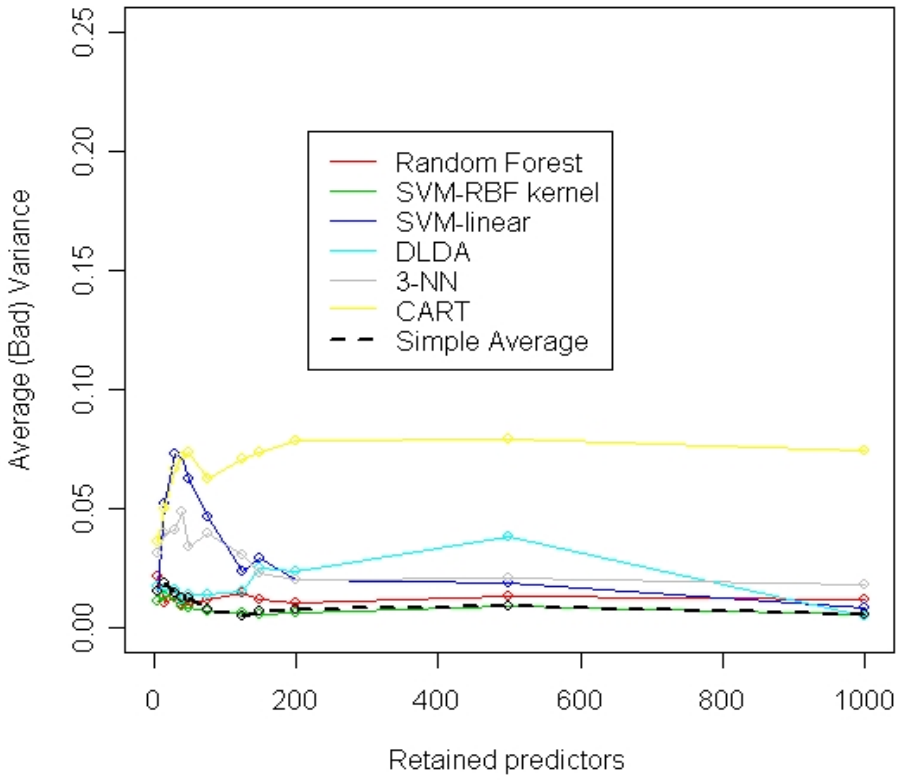



Figure 24: Prostate data set. Unbiased variance across varying dimensions

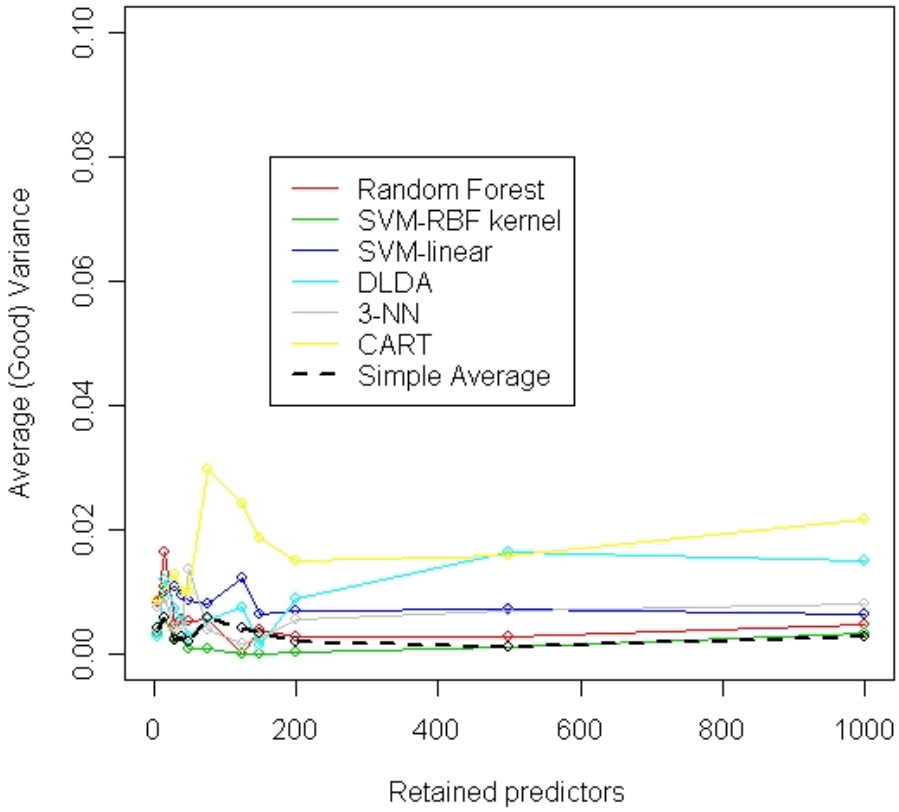



Figure 25: Prostate data set. Biased variance across varying dimensions

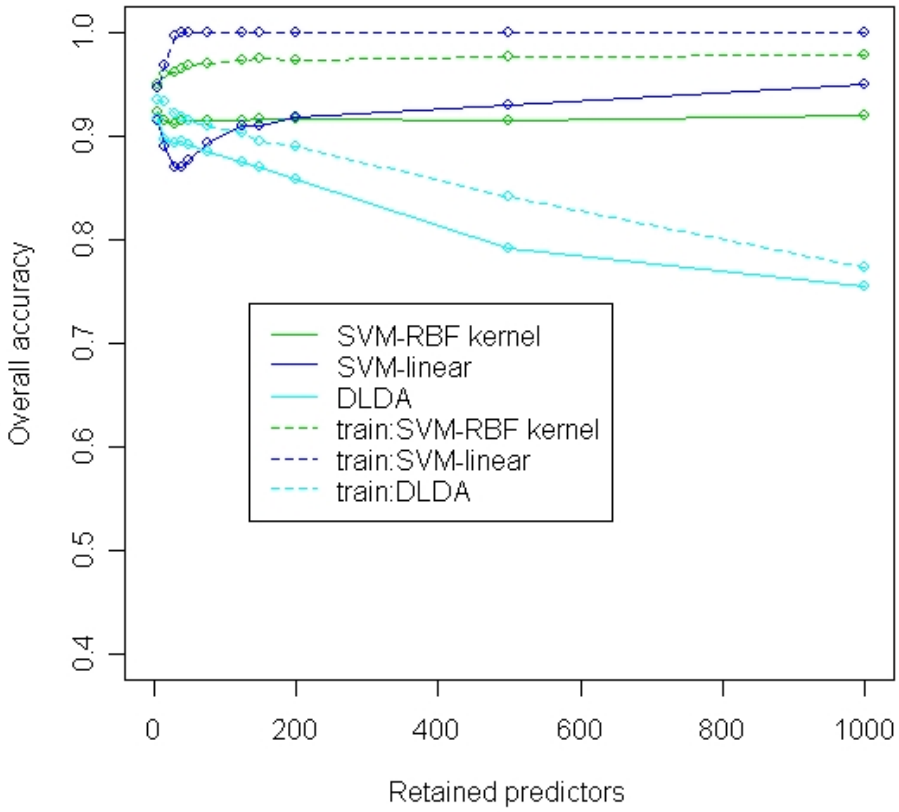



Figure 26: Prostate: training and test accuracy across varying dimensions

# Chapter 8

# Ensemble Results

The performances achieved in the five data sets are examined, and general patterns observed in each of the data sets. As detailed in Chapter 3, variable screening was performed in order to reduce the data set to a subset of the most informative predictors. The average number of variables retained was p=175 (Imprinting), p=58 (Estrogen), p=127 (Colon), p=460 (Prostate) and p=268 (Lymphoma). This average number was computed by taking the average numbers of variables retained in the informative set across all iterations. Features were retained if they had a BW rank higher than that of any artificial variable. Examination of the accuracy curves in Chapter 7 for each data set gives empirical evidence that this approach is satisfactory, given the flattening of each curve at or close to this retained number, with a few exceptions. For the Colon data set, the retained number appears to be too large, occurring at a flat portion of the curve for all of the classifiers. Despite retaining too many predictors, it appears to have not harmed accuracy for any of the classifiers except DLDA.

All data set performance is based on an ensemble containing the five individual classifiers described in section 3.6.1. Within each loop of the cross validation, each classifier was presented with the same set of pre-screened predictors (the informative set), based on the variable screening method described in Chapter 3. The screening was performed on the designated training set within each CV loop to obtain accuracy

estimates that are as close to unbiased as possible. Although classifier performance may improve slightly with a feature set selected optimally for each method, the primary comparison of interest is the ensemble performance vs. that of the individual classifiers. In addition, the set of gene selected reflects those genes that appear to have some discriminatory information in a univariate sense. Optimization of each classifier may be performed with respect to number of predictors, as well as many other factors such as tuning parameters, therefore the results presented may differ slightly from the highest attainable performance.

## 8.1 Imprinting data

### Individual classifiers

The best individual classifier is SVM-RBF. It has the lowest bias across classifiers, as expected given the variable reduction and flexibility of the classifier. It also achieves reasonably low variance, due to the regularization internal to SVM. CART is the worst performing classifier, with relatively both high bias and variance.

### Ensembles

The results of the imprinting data set show that all of the ensemble methods tend to produce slightly better accuracies compared to individual classifiers. If we take the selection of the best classifier into account (section 3.6.1), the *best* classifier has an accuracy of 0.84, compared with an ensemble accuracy of 0.86. Both the simple (SA) and weighted average (GW) produce similar results, and the local average follows closely, but has slightly higher estimated net variance. This is to be expected, since the estimation of locally derived weights from the training set is likely to increase variance. There appears to be little gain in defining the locality of the test point, indicating that the classifiers likely perform similarly to the global performance throughout the input space.

Random Forest also performs well, with slightly higher bias, and lower variance.

Table 7: Summary of Performances: Imprinting Data Set. SD(acc) denotes the standard deviation of the accuracy estimate.

| | Individual members | | | | | Ensembles | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **CART** | **SVM-RBF** | **SVM-linear** | **DLDA** | **3-NN** | **SA** | **GW** | **local** | **RF** |
| **accuracy** | 0.75 | 0.85 | 0.83 | 0.81 | 0.83 | 0.86 | 0.86 | 0.85 | 0.86 |
| **SD(acc)** | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 |
| **sensitivity** | 0.61 | 0.73 | 0.67 | 0.65 | 0.69 | 0.70 | 0.68 | 0.68 | 0.67 |
| **specificity** | 0.82 | 0.91 | 0.91 | 0.89 | 0.91 | 0.94 | 0.94 | 0.94 | 0.95 |
| | | | | | | | | | |
| **Bias** | 0.18 | 0.11 | 0.14 | 0.16 | 0.11 | 0.11 | 0.11 | 0.11 | 0.12 |
| **Vu** | 0.13 | 0.06 | 0.07 | 0.07 | 0.08 | 0.05 | 0.05 | 0.05 | 0.04 |
| **Vb** | 0.06 | 0.02 | 0.03 | 0.04 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 |
| **Net Var** | 0.07 | 0.04 | 0.04 | 0.03 | 0.05 | 0.03 | 0.03 | 0.03 | 0.02 |

## 8.2 Colon data

### Individual classifiers

The accuracy of the *best* classifier was 0.83, which was usually selected to be SVM-RBF or k-NN a vast majority of the time. The best classifiers were SVM-RBF and nearest neighbors. The difference between linear and RBF kernel-based SVM indicates that there is some non-linearity that is better approximated by more local methods. Even the more flexible CART does not perform well in this data set. DLDA ended up doing poorly given the variable selection process. Although the number of variables selected was on average quite small (about 27 predictors were retained), any variability in this number retained would cause large swings in performance for DLDA, given the observed trends as the number of predictors increased (see Chapter 7).

## Ensembles

Random Forest does not perform as well as the other ensemble classifiers, despite its flexibility. Since the other ensembles are based on a mix of diverse classifiers, they are able to fit the data well, and as a result perform well. The weighted average performs best out of the three, indicating that the higher weights of SVM-RBF and k-NN act in reducing the bias, as well as maintaining a low net variance. It appears that the main bias reduction comes from the minority class.

Table 8: Summary of Performances: Colon Data Set. SD(acc) denotes the standard deviation of the accuracy estimate.

| | **Individual members** | | | | | **Ensembles** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **CART** | **SVM -RBF** | **SVM-linear** | **DLDA** | **3-NN** | **SA** | **GW** | **local** | **RF** |
| **accuracy** | 0.74 | 0.84 | 0.79 | 0.78 | 0.83 | 0.84 | 0.85 | 0.84 | 0.81 |
| **SD(acc)** | 0.05 | 0.03 | 0.04 | 0.06 | 0.04 | 0.03 | 0.03 | 0.04 | 0.03 |
| **sensitivity** | 0.81 | 0.88 | 0.87 | 0.80 | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 |
| **specificity** | 0.62 | 0.75 | 0.66 | 0.75 | 0.72 | 0.76 | 0.78 | 0.76 | 0.69 |
| | | | | | | | | | |
| **Bias** | 0.21 | 0.18 | 0.15 | 0.18 | 0.13 | 0.13 | 0.13 | 0.13 | 0.18 |
| **Vu** | 0.11 | 0.02 | 0.09 | 0.07 | 0.06 | 0.05 | 0.04 | 0.05 | 0.04 |
| **Vb** | 0.06 | 0.03 | 0.02 | 0.03 | 0.01 | 0.01 | 0.01 | 0.01 | 0.03 |
| **Net Var** | 0.05 | -0.01 | 0.06 | 0.04 | 0.05 | 0.03 | 0.02 | 0.03 | 0.01 |

## 8.3 Estrogen data

### Individual classifiers

The *best* classifier had a performance of 0.79, with a higher level of variability in which classifier was selected to be the best. SVM-RBF performs well again in the estrogen data set, along with k-NN. There are close performances by CART and linear SVM, but DLDA lags behind in performance due to its sensitivity to the number of features selected. SVM-RBF achieves a very low bias, but also has a low net variance, making it a very good classifier for this data set.

### Ensembles

On average, all of the ensemble methods out-performed individual classification. Random Forest was the best ensemble, achieving similar bias reduction when compared to the three other ensemble methods, along with a low net variance. Weighted averaging performs slightly better than non-weighted since it takes into account the low bias performances of SVM-RBF and k-NN.

Table 9: Summary of Performances: Estrogen Data Set. SD(acc) denotes the standard deviation of the accuracy estimate.

| | **Individual members** | | | | | **Ensembles** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **CART** | **SVM -RBF** | **SVM-linear** | **DLDA** | **3-NN** | **SA** | **GW** | **local** | **RF** |
| **accuracy** | 0.78 | 0.81 | 0.77 | 0.73 | 0.80 | 0.82 | 0.83 | 0.82 | 0.84 |
| **SD(acc)** | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.01 | 0.02 |
| **sensitivity** | 0.83 | 0.87 | 0.88 | 0.76 | 0.85 | 0.89 | 0.89 | 0.88 | 0.89 |
| **specificity** | 0.72 | 0.74 | 0.64 | 0.68 | 0.73 | 0.73 | 0.74 | 0.75 | 0.77 |
| | | | | | | | | | |
| **Bias** | 0.19 | 0.16 | 0.20 | 0.25 | 0.18 | 0.16 | 0.15 | 0.15 | 0.15 |
| **Vu** | 0.07 | 0.04 | 0.06 | 0.04 | 0.06 | 0.04 | 0.05 | 0.04 | 0.03 |
| **Vb** | 0.04 | 0.02 | 0.03 | 0.03 | 0.04 | 0.02 | 0.02 | 0.02 | 0.02 |
| **Net Var** | 0.03 | 0.02 | 0.03 | 0.02 | 0.02 | 0.02 | 0.03 | 0.03 | 0.01 |

## 8.4 Prostate data

### Individual classifiers

The *best* classifier had an overall accuracy of 0.93, and was usually selected to be a support vector machine. All of the classifiers display low net variance, indicating that the bias observed in a good classifier such as SVM may be dominated by the noise in the system or observations that are hard-to-classify. The linear SVM was the best performer, achieving a remarkably low bias in comparison to the others.

### Ensembles

Due to the higher bias in many of the other classifiers, the ensembles performed slightly worse than the individual linear SVM. There is no apparent gain in using an ensemble, since the linear SVM appears to be consistent across all perturbations of the data set.

Table 10: Summary of Performances: Prostate Data Set. SD(acc) denotes the standard deviation of the accuracy estimate

| | **Individual members** | | | | | **Ensembles** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **CART** | **SVM -RBF** | **SVM-linear** | **DLDA** | **3-NN** | **SA** | **GW** | **local** | **RF** |
| **accuracy** | 0.85 | 0.91 | 0.94 | 0.78 | 0.89 | 0.92 | 0.92 | 0.91 | 0.92 |
| **SD(acc)** | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 |
| **sensitivity** | 0.88 | 0.90 | 0.93 | 0.82 | 0.86 | 0.90 | 0.90 | 0.89 | 0.89 |
| **specificity** | 0.81 | 0.92 | 0.95 | 0.73 | 0.92 | 0.93 | 0.93 | 0.94 | 0.94 |
| | | | | | | | | | |
| **Bias** | 0.13 | 0.09 | 0.05 | 0.24 | 0.10 | 0.09 | 0.08 | 0.08 | 0.08 |
| **Vu** | 0.06 | 0.00 | 0.02 | 0.02 | 0.02 | 0.00 | 0.01 | 0.01 | 0.01 |
| **Vb** | 0.03 | 0.00 | 0.00 | 0.03 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Net Var** | 0.02 | 0.00 | 0.01 | -0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.01 |

## 8.5 Lymphoma data

### Individual classifiers

There were many good classifiers for this data set, with DLDA having the highest accuracy and the *best* classifier also achieving an accuracy of 0.96. This is likely the easiest data set to fit, however CART is observed to have significant problems with extremely high unbiased variance, and as a result, high net variance.

## Ensembles

The three ensembles based on the aggregation of the individual classifiers out-perform Random Forest by having a lower bias, though slightly higher net variance. Random Forest seems to have the same bias issues as observed in the CART procedure.

Table 11: Summary of Performances: Lymphoma Data Set. SD(acc) denotes the standard deviation of the accuracy estimate

| | Individual members | | | | | Ensembles | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **CART** | **SVM-RBF** | **SVM-linear** | **DLDA** | **3-NN** | **SA** | **GW** | **local** | **RF** |
| **accuracy** | 0.75 | 0.96 | 0.96 | 0.97 | 0.89 | 0.96 | 0.96 | 0.96 | 0.94 |
| **SD(acc)** | 0.06 | 0.02 | 0.03 | 0.02 | 0.04 | 0.02 | 0.02 | 0.03 | 0.02 |
| **sensitivity** | 0.73 | 0.97 | 0.97 | 0.99 | 0.84 | 0.97 | 0.98 | 0.98 | 0.97 |
| **specificity** | 0.78 | 0.96 | 0.95 | 0.94 | 0.95 | 0.95 | 0.95 | 0.95 | 0.91 |
| | | | | | | | | | |
| **Bias** | 0.09 | 0.02 | 0.02 | 0.02 | 0.09 | 0.02 | 0.02 | 0.02 | 0.06 |
| **Vu** | 0.17 | 0.02 | 0.02 | 0.02 | 0.04 | 0.02 | 0.02 | 0.02 | 0.02 |
| **Vb** | 0.01 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 |
| **Net Var** | 0.16 | 0.02 | 0.02 | 0.01 | 0.02 | 0.02 | 0.02 | 0.02 | 0.00 |

## 8.6 Summary of results

In this small, but representative, set of genomic data sets, it is clear that classifier performance varies, and that there is no dominant classifier on all data sets in this domain. Issues such as dimension and tuning affect results greatly, making a valid comparison of classifiers difficult for methods which are sensitive to these parameters. We attempted to fix the set of predictors under consideration by applying a simple threshold based on the univariate ranking. Although this may cause the performance of some classifiers to be slightly reduced compared to the absolute optimal performance, we feel that a comparison based on a fixed set is justified given that this reduced list represents an informative set for the problem at hand. Based on this fixed set, classifiers were evaluated and several observations were made.

The first is that the support vector machine classifier is an extremely powerful and consistent classifier. Depending on the kernel used, the results were always comparable or superior to other methods of classification. The RBF kernel appears to be the best performing across the range of genomic data sets. Despite its complexities, SVM appears to be quite a low variance classifier, with less over-fitting of the training set and good generalization capabilities. As examined in Chapter 7, the success of this classifier depends upon the feature set used. Although it generally performs well across a wide range of dimensions due to internal regularization, there are cases where the dimensionality may cause poor results, such as in the imprinting data set. DLDA has a varied performance, due to its high level of dependence on the number of retained features. In the same data set (Colon), DLDA achieved both the best and worst performances. In the imprinting data set, the DLDA classifier went from one of the worst, to the best individual classifier. This observation is slightly different than that of Dudoit [20], who found DLDA to be consistently the best performer in a small set of microarray data sets. The difference may lie in the fact that Dudoit et al. pre-selected a small set of predictors (p=30 or 40), therefore DLDA's sensitivity to dimension was not observed.

Random Forest was shown to be a good ensemble-based classifier across all of the data sets, consistent with other reports [10, 13]. It is obviously superior to using a single CART tree, achieving low net variance and a lowered bias. Some of the bias reduction may come from the fully-grown trees used in RF, compared to the CART tree, which was stopped from growing too large by requiring a minimum size of 10 in each node to split. There are likely more successful methods to prune or regularize CART to prevent over-fitting, thus the results achieved here are likely to not represent the maximum possible accuracy. Despite this, the results observed in this study are highly consistent with those observed in others on the same data sets [1].

Table 12: Summary of Performances across five representative genomic data sets, rounded to the closest 0.50, taking ties into account.

**Individual classifiers:**

| **Classifier** | **Average Rank across data sets** | **(min, max)** |
|---|---|---|
| CART | 4.4 | (3,5) |
| SVM-RBF | 1.5 | (1,2) |
| SVM-Linear | 2.5 | (1,4) |
| DLDA | 4.0 | (2,5) |
| 3-NN | 2.5 | (2,4) |

**Ensembles:**

| | | |
|---|---|---|
| SA | 2.5 | (1,3) |
| GW | 2.0 | (1,2) |
| Local (locally optimal) | 3.0 | (2,4) |
| RF | 2.5 | (1,4) |

The combining of classifiers, either through a simple or weighted average proved to be a successful approach for these applications. The reduction in error was mainly bias-related, therefore it is likely that the decisions of each classifier were likely quite stable (low variance) across training sets, but for a subset were stably *incorrect*. Through the consideration of multiple diverse representations, this bias was reduced. The poor prediction of a single classifier was mitigated by the good predictions of the others. The weighted average was beneficial in some cases; however the simple average also performed very well, despite the presence of some poor individual classifiers. Comparing the weighted and simple average performances to Random Forest, we find that both all three are successful strategies; however the weighted average appears to be more stably successful across data sets.

The locally-derived weighting scheme does not improve the predictive performance of the ensemble. Although the computed weights are diverse across observations, signifying that there are some classifiers that perform better for specific observations, the identification of the nearest neighbor to the testing point, as well as the estimation of the proper weights has obvious limitations. It is, however, a method that should be explored with larger data sets. The idea that different classifiers may be selected and combined uniquely for a particular observation is akin to applying different models in different parts of the space – a common application.

# Chapter 9

# Variable and feature set importance

In most genomic studies there are two goals. The first goal is the accurate prediction of new observations. How we predict is often not as important as the results of the prediction (i.e. accuracy). Therefore issues like dimensionality, multicollinearity, and magnitude of effects are usually secondary to obtaining a good fit to the data. The second goal, which requires a much more in-depth understanding of the decision function, is the examination of how features contribute to the overall fit. Understanding the contributors to the chosen classifier can allow construction of new hypotheses, as well as confirmation existing theories.

The most common method of assessing variable importance is a simple univariate ranking. This is a highly informative process, yielding a set of features that each exhibits a strong association with class status or response. Usually, there are clusters within the rankings, representing groups of highly correlated variables. Therefore, one of the strengths of a univariate importance measure is that we retain the entire set of informative genes, since all genes within a given cluster will have similar test statistic magnitudes. The main drawback to this approach is that we fail to consider multivariable or non-linear aspects to the data. In addition, the optimal set of singular predictors may not be the optimal multivariable set for a given classifier. Some genes are likely to work jointly-yielding marginal or uninformative associations alone, but highly significant associations

with class status when considered together. We may exclude such variables, thus losing information and classifier performance without even knowing it.

Multivariable methods are often performed via a greedy search, such as is done in stepwise selection, backwards selection or similar. It is computationally prohibitive to search exhaustively for many genomic-based applications. These greedy selection strategies have been around for a long time in traditional regression modeling, and have found a new importance given the large numbers of features being considered and the computational efficiency they possess in screening large numbers of variables. In SVM for example, new variable importance measures are based on a backwards deletion of features that have little weight or negligible impact on the margin when removed one at a time, or in groups [30, 75]. Alternatively, Random Forest determines variable importance via sensitivity analysis by perturbing the data one-by-one for each feature and observing the impact on purity or accuracy [10, 16]. Both types of methods may easily be applied to many types of classifiers.

As we have shown, most classifiers will benefit when used in conjunction with variable screening. The removal of noisy and unreliable directions will allow proper evaluation of the classifier within a more informative subspace. If we examine the figures in Chapter 7 we observe that as the number of features are increased from smaller sets to much larger sets, there is a consistent leveling off of information across all data sets considered. This leveling off may signify that all of the informative genes are being captured successfully by the top univariate ranks. However, this leveling off may also be an indication that we are at the limit with what may be captured given the sample sizes we are using; subtleties in the data are unable to be clearly learned with so few observations. Although classifier performance does not currently seem to be hampered with the use of the top features from the univariate list, it is informative to compare the univariate rankings with those determined by multivariable methods, and summarize across both methods.

Random Forest outputs a variable importance ranking that is based on the change in accuracy or Gini purity index when the variable in question is randomly permuted in all trees it is present in. If the variable is in many trees, and is consistently close to the root node, the impact of perturbing the data will be large. If the feature is in few trees, or

close the bottom leaves, then the impact is likely minor. The advantage to using this importance measure is clear – we are considering importance in a multivariable setting, as well as over hundreds of trees based on perturbed data. The only drawback to this method is that there is a reported bias inherent in the RF procedure, biasing the greedy selection process towards features with larger numbers of potential splits [58]. If there is a selection bias in how features are added to trees, then there will be a bias in their importance using a sensitivity analysis. In our comparison, the set of important features in RF is defined to be those that are ranked higher than any artificial variables.

We compare the rankings based on the univariate BW statistic and Random Forest VI measure in Table 13. The top ranking features according to RF importance are relatively far down the univariate list. L2 and AT rich counts are considered most informative, followed by CR1, ALU, and MER type features. The difference in ranking, particularly for L2 and AT rich is striking, yet we have likely captured many of these predictors for the multivariable classifiers since they are, on average, ranked before the noise features. Therefore, in terms of inclusion into our informative set, there is a high level of consistency between the rankings. In the extremes, the lowest ranking features in Random Forest are very low ranking in the BW based test, or not present at all in the pre-screened variables.

Table 13: Variable importance measures for the imprinting data set. RF importance is based on the RF analysis of importance and BW rank is based on the univariate summary of information

| Feature | RF - importance | BW rank |
|---|---|---|
| L2_DNSC500 | 1.452736606 | 201 |
| L2_DNSC250 | 1.31952056 | 70 |
| ATRICH_UPSC250 | 1.270920053 | 65 |
| ATRICH_DNSS500 | 1.25945717 | 94 |
| CR1_DNES500 | 1.243351189 | 27 |
| **ALU_DNSS500** | **1.23086399** | **1** |
| MER1T_DNES250 | 1.200909678 | 233 |
| **ALU_DNSC500** | **1.191346081** | **3** |
| **ALU_DNES250** | **1.188149396** | **11** |
| MER1T_DNEC250 | 1.141197417 | 47 |
| s15 | 1.137938788 | 22 |
| ATRICH_UPSS250 | 1.128641761 | 88 |
| **ALU_DNSC250** | **1.128432538** | **8** |
| CR1_DNSS500 | 1.126541131 | 40 |
| CR1_DNEC500 | 1.1152653 | 23 |
| **ALU_DNSS100** | **1.107512375** | **6** |
| MIR_DNSS250 | 1.085808748 | 48 |
| ATRICH_UPSC500 | 1.054211927 | 81 |
| **ALU_DNEC500** | **1.029981591** | **14** |
| **ALU_DNES500** | **1.02388921** | **12** |
| L2_DNSS500 | 1.020285586 | 226 |
| MER1T_DNSC500 | 1.009059126 | 50 |
| ATRICH_DNSC500 | 0.99575449 | 78 |
| L2_DNSS250 | 0.986175526 | 117 |
| L2_DNES500 | 0.982621392 | NA |
| ATRICH_DNES500 | 0.978675985 | 128 |
| MIR_DNSC500 | 0.969180705 | 55 |
| **ALU_DNSC100** | **0.955719243** | **9** |
| **ALU_DNSS250** | **0.94614827** | **5** |
| CR1_DNSC500 | 0.92875167 | 19 |
| MER1_UPSS500 | 0.927121818 | NA |
| CR1_DNES250 | 0.924897973 | 128 |
| GCRICH_DNSS100 | 0.912136915 | 235 |
| **ALU_DNSS25** | **0.902461446** | **4** |
| CPG_GENEBODY | 0.885623011 | 73 |
| **ALU_DNEC250** | **0.88201224** | **13** |
| MIR_DNEC500 | 0.868004997 | 76 |
| ALU_DNES50 | 0.867248669 | 46 |
| ATRICH_DNSS250 | 0.866467553 | 210 |
| SIMREP_UPSS250 | 0.86336915 | 21 |

Table 13: continued from previous page

| Feature | RF – importance | BW rank |
|---|---|---|
| MER1_UPSS250 | 0.853078796 | NA |
| **ALU_DNSS50** | **0.851263804** | **2** |
| CR1_DNSS250 | 0.829445103 | 156 |
| ATRICH_UPSS500 | 0.827885448 | 119 |
| MIR_DNSS500 | 0.812599464 | 186 |
| MIR_DNSC250 | 0.803829331 | 26 |
| MER1T_DNEC500 | 0.790945645 | 56 |
| **ALU_UPSC50** | **0.787591518** | **16** |
| MIR_DNEC250 | 0.78395724 | 34 |
| ALU_DNES100 | 0.775813025 | 35 |
| MER1T_DNSS500 | 0.74250387 | 146 |
| ALU_UPSC250 | 0.735940549 | 44 |
| ALU_DNSC25 | 0.733974368 | 10 |
| ATRICH_DNEC500 | 0.733878249 | 90 |
| TC2_DNSS500 | 0.727501212 | 101 |
| ATRICH_DNES250 | 0.724648649 | NA |
| ALU_UPSC100 | 0.724179575 | 25 |
| ALU_DNEC100 | 0.722369666 | 32 |
| MER1T_UPSC500 | 0.713028127 | 86 |
| L2_DNEC250 | 0.710601406 | 164 |
| TC2_UPSC500 | 0.709844168 | 54 |
| **ALU_UPSS50** | **0.698891726** | **15** |
| **ALU_DNSC50** | **0.68962069** | **7** |
| CR1_DNSS100 | 0.685598774 | 232 |
| CR1_UPSS500 | 0.680529 | 58 |
| ALU_DNES25 | 0.679608997 | 33 |
| L1_UPSC250 | 0.670963852 | NA |
| MER1T_DNSS250 | 0.660013958 | NA |
| MIR_DNSS100 | 0.652591515 | 131 |
| ATRICH_DNEC250 | 0.641577678 | 225 |
| GCRICH_DNSS50 | 0.6233838 | 115 |
| MER1T_DNES500 | 0.61878625 | 193 |
| L2_UPSS500 | 0.603230923 | NA |
| ALU_DNEC50 | 0.598314865 | 45 |
| GCRICH_BDYS10 | 0.593465103 | 92 |
| ALU_UPSS250 | 0.592305492 | 37 |
| **ALU_UPSC25** | **0.584142318** | **18** |
| ALU_UPSS5 | 0.582773082 | 36 |
| TC2_UPSS500 | 0.581624745 | 49 |
| SIMREP_UPSS500 | 0.566384117 | 71 |
| TC2_DNES500 | 0.558064892 | 91 |
| CR1_DNEC250 | 0.556785638 | 66 |

Table 13: continued from previous page

| | | |
|---|---|---|
| MIR_DNES100 | 0.554526442 | 182 |
| CR1_DNSC250 | 0.552058714 | 89 |
| MIR_UPSC250 | 0.545406827 | 95 |
| **ALU_UPSS25** | **0.544317804** | **17** |
| GCRICH_DNSC250 | 0.543782305 | NA |
| ERVL_UPSS250 | 0.543573622 | NA |
| ATRICH_UPSC100 | 0.539443243 | 80 |
| **ALU_UPSS100** | **0.537602715** | **20** |
| ALU_DNSS10 | 0.535002748 | 30 |
| GCRICH_UPSS10 | 0.530668512 | 230 |
| s13 | 0.527777218 | 57 |
| L2_DNSC25 | 0.519420229 | NA |
| MIR_DNSC100 | 0.516374352 | 53 |
| PIGGY_DNSS500 | 0.51468268 | 135 |
| PIGGY_DNES500 | 0.511875571 | 130 |
| ALU_DNES10 | 0.510338355 | 60 |
| TC2_UPSC250 | 0.510286949 | 116 |
| MIR_UPSS250 | 0.50872554 | 157 |
| MIR_UPSS50 | 0.492483432 | 243 |
| ALU_UPSS10 | 0.487853988 | 43 |
| CR1_DNES100 | 0.482068086 | 220 |
| SIMREP_UPSS100 | 0.477446489 | 59 |
| MER1T_UPSS500 | 0.472842077 | 238 |
| CRICH_UPSS250 | 0.471893924 | NA |
| GNUM_DST250 | 0.47113947 | NA |
| L1_DNES50 | 0.460398017 | NA |
| L2_UPSC250 | 0.458204776 | NA |
| ALU_DNEC10 | 0.457028686 | 68 |
| GCRICH_BDYC10 | 0.45623144 | 110 |
| CPG_DST25 | 0.454216966 | NA |
| TIP100_DNSC500 | 0.450380776 | NA |
| ATRICH_DNSS50 | 0.446572287 | 159 |
| ALU_UPSC500 | 0.443806782 | 114 |
| m1 | 0.442242118 | NA |
| ALU_UPSC10 | 0.440106884 | 39 |
| L2_UPSC500 | 0.438766249 | NA |
| CPG_DST10 | 0.432596948 | 132 |
| CR1_UPSC500 | 0.432273059 | 41 |
| L2_DNEC500 | 0.429468016 | 245 |
| CRICH_UPSS50 | 0.428076117 | 224 |
| MIR_DNSS10 | 0.425234422 | 154 |
| CRICH_DNES100 | 0.424492338 | NA |
| CR1_UPSC250 | 0.419092454 | 108 |

If we examine features individually, it is clear that many highly ranking features in RF are lower on the list in the BW set. This is due mainly to the large clusters of correlated predictors that dominate the tops spots on the univariate rankings. The ALU cluster (highlighted) represents such a cluster. These predictors are all highly related, and come from the larger ALU windows. Retaining all of these features is generally not harmful with respect to any classification procedure, since accurate prediction tends not to be hampered by collinearity in many methods of classification.

A descriptive approach to understanding likely clusters in the set of informative features is to use principal components analysis (PCA) and examine those features with high loadings (>0.60) on unique factors. Given the set of components, we are able to examine the univariate rankings to gauge cluster importance (Table 14). The ALU cluster is clearly the dominating cluster, with the highest average rankings, as well as a high visibility in the screened subset. AT rich elements, ranked highly on the RF list, are also ranked well in the cluster rankings, though it is much less dominant of a feature in univariate ranks when compared with ALU. CR1 sequence features are related to LINES – they are known as "L3's" and represent an ancient feature of DNA that is shared with birds (CR=chicken repeats). This is observed in the data as CR1 and L1 elements share a common component. Both ranked lists show that CR1 and L1s are important features. There is a bigger presence of L1 elements in the ranked list (11% vs. 6%), and they occupy a higher average univariate-based rank than CR1 elements (156 vs. 116) which is also observed in the RF rankings. The s15 and s16 motifs are also biologically related motifs and are ranking highly in both lists.

Although L2-type features are highly ranked in the RF list, with the most important feature being a large window L2 count, it ranks only 201 in the univariate rankings. In fact, the first appearance of L2 in the univariate list occurs at rank=70, followed by another at rank=117. At first glance this appears to be quite a large difference between ranking systems; however L2 elements are associated with MIR elements, which are highly ranked via the BW statistic. L2 and MIR elements are biologically related since MIR elements die off when L2 elements do. A factor that should be considered is that Random Forest has an element of randomness to it, given that it relies on bootstrapped data and random selection of features. Even with 1,000 trees under consideration, the

important features will vary based on run and criterion used (Gini index vs. accuracy reduction). The high level of consistency between the univariate and multivariable rankings gives more stability and reliability to which features are important for this data set. This consistency does not mean that we can ignore relationships between variables (recall the poorer performance of DLDA in many examples) and non-linear aspects to the data. It does indicate that selecting a good-sized informative set based on univariate ranks is a practical approach in variable screening for data sets of this size and type.

Table 14: Components of sequence features using PCA and their average ranking based on univariate ranks. Proportion of screened set is relevant for features with equal numbers of initial features under study and represents the proportion of times the element (over varying window and count/size) appears in the screened set.

| Cluster | Cluster Rank | Proportion of screened set* |
|---|---|---|
| **ALU** | **1** | **15%** |
| **AT rich** | **2** | **5%** |
| **S15/s16 motifs** | **3** | **--** |
| **MIR/L2** | **4** | **10% / 3%** |
| **MER** | **5** | **4%** |
| **ERV/CR1/L1/Piggy/Tc2** | **6** | **3%/ 6%/ 11%/ 13%/ 11%** |

## Variable importance using support vector machines

As a secondary approach to multivariable feature importance, we use a backwards selection approach for SVM to assess which variables are considered important for this classifier. Backwards selection has been a common screening approach in the regression literature, and the process used in SVM by the machine learning community suffers from the same issues and strengths. There are two (quite similar) approaches that are currently most used, SVM-RFE (recursive feature elimination) and Recursive SVM [30, 74]. R-SVM assesses variable importance through evaluation of the individual weight coefficient in the SVM classifier, but adjusts this weight by the difference in class means. This has the effect of being more robust to outliers and noise, since all of the training data are used in computing the mean differences (instead of just the support vectors). SVM-RFE uses the square of the weight corresponding to each feature, and is based completely on the support vectors, rather than the average observation. This approach (which uses only those observations on the boundary, rather than the "average" observation) can also be advantageous in finding informative predictors; however the best method is, as always, data-dependent. The main goal in either method is to derive a set of informative features, with the understanding that the top ranked features within the retained subset may not be most important outside of the set. Both feature elimination methods will be impacted by multicollinearity since it is expected that correlated features will have a diminished ranking due to their non-uniqueness. However, it is expected that at least one representative feature from each cluster will be present in many moderate-sized informative sets. In order to arrive at a single informative set, we ran several iterations of elimination until the size of the informative set was reasonable.

## General Overview of Recursive Feature selection in SVM

1. Fit the SVM classifier using the full set of predictors
2. Compute the weights, $w_i$ for each predictor

$$w_i = \sum_{i=1}^{n} \alpha_i y_i x_i$$

where:

$\alpha_i$ is non-zero for all support vectors

$y_i$ is the class status of obs. i

$x_i$ is the feature measurements for obs. i

3. SVM-RFE: Rank features by $wi^2$
4. R-SVM: Rank features by $w_i$ ($m_1$-$m_0$), where $m_1$ and $m_0$ are the class means of the two classes.
5. Remove the bottom 25% of features.
6. Repeat analysis on reduced feature set.

We repeated this process eight times, ending up with a list of 240 informative features. Stopping was based on the size of the data set, rather than the accuracy of the resulting model since the goal of this analysis is to explore the most important features. Table 15 gives the top 25 features obtained from backwards selection of a SVM with RBF kernel using the squared weight as the ranking criterion. There was little difference between the two ranking criteria in this data set. As it is clear, the set of informative features is again comprised of ALU, CR1, MIR and L2 elements. This is supportive of both the RF variable importance results and the univariate results, giving a clear indication that the same set of predictors is informative over a multitude of different classification strategies. If we allow the process to run for 10 iterations, we end up with a reduced subset of 135 predictors. Figure 16 shows the top 25 features in this informative set. The relative positions have changed, with CR1 and MIR elements having a higher rank than ALU, as well as a bigger presence of L2 elements. The redundancy of the ALU elements is likely resulting in more of these important, but overlapping elements being removed from the

informative set. Consistent with univariate and RF-based measures, the large windows (250-500) are the dominant window size observed in the RFE-based set. In using SVM, this result is particularly striking.

Table 15: Variable importance based on recursive SVM selection. The final rank is based on 8 iterations of backwards selection based on the criterion used in R-SVM

| Feature | Rank |
|---|---|
| CR1_DNEC250 | 1 |
| CR1_DNEC500 | 2 |
| ALU_DNSS500 | 3 |
| ALU_DNES500 | 4 |
| ALU_DNSC500 | 5 |
| CR1_UPSC500 | 6 |
| CR1_DNSC250 | 7 |
| ALU_DNES250 | 8 |
| ALU_DNEC250 | 9 |
| ALU_DNSS250 | 10 |
| MIR_DNSC250 | 11 |
| CR1_DNSS500 | 12 |
| CR1_UPSS500 | 13 |
| CR1_UPSC250 | 14 |
| MIR_DNSS250 | 15 |
| MIR_DNEC250 | 16 |
| MIR_DNSS500 | 17 |
| CR1_UPSS250 | 18 |
| CR1_DNSS250 | 19 |
| L2_DNSC250 | 20 |
| L2_DNSC500 | 21 |
| MIR_DNES500 | 22 |
| MIR_UPSS250 | 23 |
| MIR_UPSC250 | 24 |
| L2_DNEC250 | 25 |

Table 16: Variable importance based on recursive SVM selection. The final rank is based on 10 iterations of backwards selection based on the criterion used in R-SVM

| Feature | Rank |
|---|---|
| CR1_DNSC500 | 1 |
| MIR_DNSC250 | 2 |
| CR1_DNEC250 | 3 |
| MIR_DNSS250 | 4 |
| MIR_DNEC250 | 5 |
| ALU_DNES250 | 6 |
| ALU_DNSS250 | 7 |
| CR1_UPSS500 | 8 |
| L2_DNSC500 | 9 |
| L2_DNSC250 | 10 |
| L2_DNEC250 | 11 |
| MIR_UPSS250 | 12 |
| MIR_UPSC250 | 13 |
| CR1_DNSS250 | 14 |
| MIR_UPSS500 | 15 |
| L2_DNSS250 | 16 |
| ERV1_DNES250 | 17 |
| L1_DNSS250 | 18 |
| L2_DNES250 | 19 |
| ALU_UPSS250 | 20 |
| ALU_UPSC250 | 21 |
| SIMREP_UPSS250 | 22 |
| L2_UPSC500 | 23 |
| ERVL_DNSC500 | 24 |
| L2_UPSS250 | 25 |

The results obtained are based on the RBF kernel. If we instead use a linear kernel for the support vector machine, we will end up with a different set of ranks, based on the relative impact of each feature on the margin. However, it is observed that many of the same elements are present in both sets of informative features.

## Window Size

The most informative window size appears to be in the range of 250-500, as opposed to smaller, more local influences captured by using narrow windows flanking the gene. In considering the entire informative set based on either univariate ranks or Random Forest importance, Simple Repeat elements and to a lesser extent ALU and MIR elements, have a larger range of informative widow sizes (5 to 500kb), meaning that the narrowly defined windows are part of the informative set, in conjunction with the information in the larger windows. For these elements, the local aspects of the gene appear to be informative in addition to more global patterns. For other important sequence features, the informative window sizes are primarily those that are in the range of 250 to 500kb. This is true of MER, L2, CR1, and AT rich elements. If we consider the recursive process using SVM, we observe the same trends. As we further refine the list based on more iterations of backwards elimination, the informative set increasingly becomes based on windows of 250-500kb only. This provides important insights with respect to local vs. global control of the gene.

## Stability of Predictions across classifiers

One advantage to having ensembles of classifiers is that we may create subtypes of observations within outcome class. If we use the stability of each prediction At X=x estimated via bootstrap or cross validation, we may take the median stability as a summary measure. When we use several classifiers, we are able to further assess the stability of prediction across classifiers. Observations that are easy for any classifier to predict consistently may be robustly labeled as "easy" observations. Observations that cause disagreement between classifiers (the diverse predictions that ensembles benefit), and those with low margins, are considered "boundary" observations. Finally, those observations that are consistently misclassified by all systems across all training samples are labeled as "hard" observations.

If an observation is far from the theoretical decision boundary, it is expected that most, if not all, classifiers will correctly classify this observation most of the time. In other words, the stability at X=x for each classifier will be close to 1.0. If an observation is close to the theoretical decision boundary then the predictions of most classifiers will be noisy or consistently incorrect (for a biased classifier). Given this, observations may be re-evaluated based on the ease of prediction. In the imprinting data set, there are 7 hard-to-classify cases (MEST1, NNAT, PLAGL1, COPG2, PEG10, ZNF215, CPA4). Of these, 5 out of 7 (71%) are paternally expressed genes (ZNF215 and CPA4 are maternally expressed), many (4/7) residing on chromosome 7. Chromosome 7 currently contains 3 imprinted gene clusters – one of which is comprised of CPA4, COPG2 and MEST1 on the q arm (7q32). Therefore, there is some evidence that this specific cluster has sequence characteristics that are different than the other imprinted genes, thus making these member genes harder to classify. In future analyses, it may be preferable to remove this cluster. In addition, NNAT is a paternally expressed gene that is located within a non-imprinted gene (BLCAP). It therefore is likely that the regions flanking NNAT are highly characteristic of a non-imprinted gene.

# Chapter 10

# Conclusions and Future Work

There are many classifiers, and tens of variants of each classifier, all with the same general goal: to provide a classification scheme with high levels of accuracy across all testing sets. Classification differs from estimation in one important regard – the usual additive bias and variance breakdowns in estimation error do not always hold in classification. Traditional high bias estimators can become powerful classifiers, since high estimation bias, such as that caused by over-smoothing, does not always result in a loss of classification efficacy. As Friedman points out [22], all we need to have is the observation be on the correct side of the decision boundary. Therefore, it has been observed that simple classifiers such as DLDA or naïve Bayes will perform as well as complex methods such as SVM. This relative similarity between simple and complex has been characterized by Hand [30] as an "illusion of progress", meaning that the novel classifiers being constructed are no better than the set of standard classifiers that existed decades ago. Despite the success of simple classifiers across many applications, there is a growing body of work to suggest that the newer methods are also highly successful classifiers, as well as strong estimators. Methods such as Support Vector Machines and Ensemble-based classifiers have been shown to complement, and sometimes offer a substantial advantage to, standard methods, as illustrated in this work. The regularization offered internally through the SVM process is shown to be highly successful in the

genomics domain. Although these methods are more "black box" than a simple DLDA, new ways to measure variable importance are emerging, and using multiple representations of the data via ensembles provides informative summaries of the data.

This dissertation attempts to evaluate some of the most common classifiers with respect to error, and focuses on the decomposition of this error into bias and variance. The strength of this decomposition is that it offers a hint of why certain classifiers fail, and the impact of high dimension with respect to error. Although bias and variance have been evaluated previously [17, 38], it was studied to provide empirical evidence concerning variance reduction in bagging and bias reduction in boosting - there is little to date that studies this decomposition to evaluate individual classifiers with respect to increasing dimension and complexity. Although high error, whatever the source, is detrimental, knowledge of whether this error occurs due to high variance as opposed to high bias is informative for many reasons. First, high variance is typically associated with over-fitting, which is usually attributed to a large number of parameters being estimated. The classifier fits the training data so well that it produces highly varying predictions on new data. On average, the classifier is correct since the true signal is being modeled correctly along with the noise from the training set. High variance is able to be corrected – either through simplification of the model, or larger sample sizes, or through model averaging.

High bias comes from lack of representation, which is attributed to under-fitting the data, either with fewer parameters than needed (for example: missing an important predictor in the model) or a too rigid classifier (linear models instead of non-linear). We show that Support Vector Machines may exhibit high bias when the dimension is too large and as a result the support vectors have no reach or influence except in their immediate area. The classification boundary imposed by the classifier is off from the true boundary. Bias may be corrected by expanding the model complexity or adding new features, however it is a harder problem to fix. Assuming that there is no noise term, bias represents the classifier's inability to correctly and consistently classify a set of observations within the larger data set.

We show the breakdown of bias and variance for each classifier and find some surprising results. First, the simple classifier DLDA is highly sensitive to dimension,

more so than any of the more complex classifiers. Also, DLDA suffers, as expected, from high bias in many situations, but also can display a larger than expected net variance relative to other classifiers. Support vector machines, on the other hand, can display low bias and variance, despite their complexities. However, contrary to many reports, SVM's are not resistant to the curse of dimensionality. As we observed, kernel methods which use Euclidean distance may be adversely affected by high dimension. In such dimensions, the decision function may consist of spikes corresponding to the area close to the support vectors, and a flat decision surface elsewhere. This will result in a high training accuracy, but a highly biased classification of new data. SVMs often benefit greatly from variable reduction, and this reduction may explain why linear SVM is often found to be similar in performance to the RBF kernel. If linear SVM can handle high dimensions as well or better than RBF kernels (which are more local in nature), then studies where all features are included may be biased towards linear kernels. In addition, in high dimensions the data are likely separable, thus the regularization achieved using a linear kernel may result in better generalization. High dimension also appears to affect the simple classifiers. DLDA has a high sensitivity to increasing dimension. In cases where added dimension is detrimental, it is often the bias term that increases, rather than the variance. The rigidity of this classifier that often allows low variance and low bias with fewer predictors, may change into a high bias classifier when more variables are considered as the impact of this rigidity becomes more pronounced.

Ensembles are the next step in classification progress. It has been shown, both theoretically and empirically that combining predictions based on perturbations of a fixed classifier is largely a variance reduction technique, assuming that the biases of the classification do not change. We explore several alternative ensembles that use different classifiers, rather than different versions of the training sets, such as in bagging. This method, although intuitive and quite simple, has not been explored fully in the bioinformatics or machine learning literature. The main advantage to combining a set of diverse classifiers appears to be in bias reduction, as opposed to variance reduction. The diversity introduced by the different classifiers created a majority unbiased classifier at X=x, resulting in lower bias. How to exploit this lowered bias further is another research topic. Although the location-specific weighting used in this study proves to not confer

much advantage over simple averaging or global weighting, it should be explored with larger samples sizes and more variation in how location is assessed in future studies.

In conclusion, ensembles offer a way to consistently combine a set of classifiers to yield a comparable or superior accuracy. Even a simple average of five classifiers offered a stable and robust method to aggregate decisions. The weighted average was observed to be the best combination method in this study, as it offers some protection again a few poor classifiers in the set. Using bootstrap-based estimates of accuracy, rather training estimates likely prevented over-fitting, therefore this is the recommended approach for weight construction. Although we used a fixed set of predictors on each classifier, future work will allow diverse platforms (such as gene expression and protein abundance) to be integrated, evaluated separately, and then combined. The added diversity stemming from independent predictor sets is the main idea of CERP, and will likely enhance the success of combining diverse classifiers based on different or combined sets of features. It is likely that ensembles will benefit from base member selection as much as models benefit from variable selection, a future research direction in this area [34].

Variable importance is a vital aspect of building classifiers, especially given the sensitivity of some methods to dimension. In addition, understanding which variables are the largest contributors to the decision allows new hypotheses to be made regarding disease, potential gene targets to be explored for treatment, or more simply which features to measure and focus on in future studies. Variable importance was briefly examined in the imprinting study, with the goal of understanding the multivariable rankings obtained by Random Forest and SVM in the context of the univariate rankings via the BW statistic. While it is hard to completely characterize the relationships of features to response given the complexities of the classifiers involved, we may begin to examine the differences in ranking through careful examination and *a priori* knowledge of feature relationships. We found that in the imprinting data set, each method was telling a similar story, and that the set of informative features mainly consisted of the patterns of ALU, MIR, L2, and CR1 sequence features in more far-reaching areas around the gene. Research, already underway, will take these individual importance measures

and combine them to allow multiple representations to be summarized to gain a wider picture of importance for a particular problem.

It is also possible to use bootstrapped estimates of stability in conjunction with stability over a set of classifiers to determine which observations sit closer to the boundary and are, therefore, predicted with less certainty across many forms of classifier and over many training sets. This is informative for many reasons. First, there may be new observations that cannot be predicted with much certainty. If we simply output a class label, we fail to understand whether the assignment has been made with confidence. Examining the prediction across multiple classifiers and across multiple perturbations of the data gives a better idea of whether it is a hard to fit observation, or simply the wrong classifier was used to predict.

In summary, this dissertation made the following contributions:

- A comprehensive bias-variance breakdown for many standard classifiers across varying dimension to examine the effects of using large feature sets
- Novel approaches in ensemble methodology were explored. A locally optimal weighted ensemble was created.
- An ensemble of classifiers that combines via a simple or weighted average can improve or maintain accuracy, with some additional stability with respect to performance across and within data sets
- Using diverse classifiers can lower bias, as opposed to the variance reduction observed in bagging procedures.
- A classifier for genomic imprinting based on the ensemble of five diverse classifiers was built
- An understanding of the important features with respect to imprinting
- A comparison of Random Forest's variable ranking, SVM-based feature elimination method, and a univariate approach on the imprinting data set.

## Limitations

The limitations of this work are similar to those present in any comparative analysis of classifiers. The success of any comparison rests on the minimization of all biases that favor one classifier over another. The performance of each classifier was maximized with: 1) the selection of variables based on information content to reduce noise, 2) a good understanding of how each method works, and 3) appropriate tuning parameters and regularization. Despite this, there may be some inadvertent biases towards some methods over others, especially given that some classifiers require more effort in achieving maximum performance. Support vector machines are an example of such a classifier, where user experience is a big contributing factor to its success. Random Forest, on the other hand, is a plug-and-go classifier, requiring no experience other than simple data input.

Given this important issue, we point out the following features of the analyses performed in this work. First, the main comparison in Chapter 7 is primarily an examination of dimension within each classifier, rather than a comparison between classifiers. Any observations made regarding relative classifier performance were based on general patterns observed. In Chapter 8, the new ensemble methods examined all depended completely upon the performances of the individual classifiers. Therefore, any bias in any of the classifiers would impact the results of the ensembles under investigation equally.

A second limitation was in the estimation of bias and variance. We folded the noise term (the unavoidable variance) into the bias term due to its difficulties in estimation from small data sets. This likely inflated the bias term, perhaps considerably. However, the bias estimated from each classifier under consideration was limited by the same noise term. The impact of a zero noise term is thus on the absolute level of bias estimated. The dominance of bias over variance observed in each data set may be lessened, however it is unlikely that the conclusions of this work will change.

A third limitation is the size of the data sets and range of problems considered. Although the five data sets under investigation were considered representative of genomics-based problems, they are only a small fraction of the types of data sets that may

be encountered. Therefore, all results based on these data should be interpreted as valid for the data sets on which they were ascertained. The generalization of results should be treated with caution, since each user will have their own biases and each data set will have its own nuances.

Finally, the set of classifiers used in the ensemble is, of course, arbitrary. This line of research is not dependent upon any particular fixed set of classifiers, yet the decision of which methods to include in this work had a solid basis. First, the included classifiers were felt to represent some of the best, most diverse, and most commonly used state-of-the-art classifiers. Each can be considered a stand-alone classifier, as illustrated by their individual performances. Support vector machines and Random Forest are relatively new classifiers, but are widely considered to be important and successful innovations. DLDA was the simple, yet successful, classifier highlighted in Dudoit. Nearest neighbor is an instance-based classification technique, and CART is a flexible tree-based approach that serves as the base member in Random Forest.

# References


1. Ahn, H., Moon, H., Fazzari, MJ, Lim, N., Chen, J. J. and Kodell, R. L. Classification by Ensembles of Random Partitions. *Journal of Computational Statistics and Data Analysis 2007;*51:In Press
2. Alexandre LA, Campillo AC, Kamel M. Combining Independent and Unbiased Classifiers using Weighted Average. *Proceedings of the International Conference on Pattern Recognition,* 2000.
3. Ali K. On the link between error correlation and error reduction in decision tree ensembles. UCI Technical Report 1995.
4. Alizadeh, A., Eishen, M., Davis, E., Ma, C. & et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 2000; 403: 503–511.
5. Alon,A., Barkai,N., Notterman,D.A., Gish,K., Ybarra,S., Mack,D., and Levine,A.J. (1999) Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. *Proc. Natl. Acad. Sci. 199*9; 96:6745-6750.
6. Austin, PC. A Comparison of regression trees, logistic regression, generalized additive models, and multivariate additive regression splines for predicting AMI mortality. *Statistics in medicine* 2007; In press
7. Blair RM, Fang H, Branham WS, Hass BS, Dial SL , Moland CL, Tong W, Shi L, Perkins R, Sheehan DM. The Estrogen Receptor Relative Binding Affinities of 188 Natural and Xenochemicals: Structural Diversity of Ligands *Toxicol. Sci* 2000*;* 54: 138-153.
8. Breiman,L. Bagging predictors. *Machine Learning* 1996; 24: 123–140
9. Brieman L. Arcing classifiers. *Annals of Statistics 1998;* 26(3):801-849
10. Breiman,L. Random forests. *Machine Learning 2001;* 45: 5–32
11. Burnham, Kenneth P. and David R. Anderson. Model Selection and Multimodel Inference: A Practical Information-Theoretical Approach. 2nd ed. New York: Springer-Verlag, 2002.

12. Capriotti E, Calabrese R, Casadio R. Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. *Bioinformatics.* 2006 Nov 15; 22(22):2729-34.
13. Chen XW, Lieu M. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics 2*005 Dec 15;21(24):4394-400
14. Condorcet, M. Sur les elections par scrutiny. Histoire de l'Academie Royale des Sciences, 31-34, 1781.
15. Del Rio M, Molina F, Bascoul-Mollevi C, Copois V, Bibeau F, Chalbos P, Bareil C, Kramar A, Salvetat N, Fraslon C, Conseiller E, Granci V, Leblanc B, Pau B, Martineau P, Ychou M. Gene expression signature in advanced colorectal cancer patients select drugs and response for the use of leucovorin, fluorouracil, and irinotecan. J Clin Oncol. 2007 Mar 1;25(7):773-80
16. Diaz-Uriarte R, Alvarez de Andres S. Variable selection from random forests: application to gene expression data. Tech. report. http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html
17. Dietterich,T.G. Ensemble methods in machine learning. In Kittler,J. and Roli,F. (eds), *First Intl. Workshop on Multiple Classifier Systems*, *Lecture Notes in Computer Science.* Springer Verlag, New York, 2000; 1–15
18. Domchek SM, Eisen A, Calzone K, Stopfer J, Blackwood A, Weber BL. Application of Breast Cancer Risk Prediction Models in Clinical Practice. *Journal of Clinical Oncology 2003;* 21(4):593-601
19. Domingos, P. A Unified Bias-Variance Decomposition and its Applications *Proc. 17th International Conf. on Machine Learning, 2001.*
20. Dudoit S, Fridyland JF, Speed TP: Comparison of discrimination methods for tumor classification based on microarray data. *JASA* 2002; 97:77-87.
21. Falls JG, Pulford DJ, Wylie AA, Jirtle, RL. Genomic Imprinting: Implications for Human Disease. *American Journal of Pathology* 1999; 154(3):635-647.
22. Friedman J. On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery* 1997, 1(1):

23. Freund Y., and Schapire, R. E. Experiments with a new boosting algorithm, In *Proc. 13th International Conference on Machine Learning* 1997; pp. 148-146. San Francisco
24. Fumera G and Roli F. A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems. IEEE Trans. Pattern Anal. Mach. Intell 2005; 27(6): 942-956
25. Fumera G and Roli F. Linear Combiners for Classifier Fusion: Some Theoretical and Experimental Results. In Proc. Int. Workshop on Multiple Classifier Systems 2003; pp:74-83
26. Gail MH, Brinton LA, Byar DP, Corle DK, Green SB, Schairer C, Mulvihill JJ. Projecting individualized probabilities of developing breast cancer for white females who are being examined annually. *J Natl Cancer Inst.* 1989; 81:1879-1886.
27. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. Molecular classification of cancer: class discovery and class prediction by gene expression profiling. *Science* 1999; 286:531-537
28. Greally JM. Short Interspersed transposable elements (SINEs) are excluded from imprinted regions in the human genome. *Proc Natl. Acad.* Sci. 2002; 99:327-332
29. Greally, JM. *Personal communication*
30. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. *Machine Learning,* 2000.
31. Hand DJ. Classifier technology and the Illusion of Progress. *Statistical Science.* 2006; 21(1):1-14
32. Hastie, T., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning*: *Data Mining, Inference and Prediction*. Springer Verlag, 2002.
33. Ho, TK . The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1998; 20 (8): 832-844
34. Ho TK. Multiple classifier combination: Lessons and next steps. *Hybrid Methods in Pattern Recognition*, World Scientific 2002; Kandel, A and Bunke H (eds.).

35. Huang YL, Chen DR. Support vector machines in sonography: application to decision making in the diagnosis of breast cancer. *Clin Imaging.* 2005 May-Jun;29(3):179-84.
36. Izmerlian G. Application of the random forest classification algorithm to a SELDI-TOF proteomics study in the setting of a cancer prevention trial. Ann N Y Acad Sci. 2004;1020:154-74.
37. James, G (2003). Variance and Bias for General Loss Functions. *Machine Learning* 51, 115-135
38. James, G., and Hastie, T. Generalizations of the Bias/Variance Decomposition for Prediction Error, Technical Report, Department of Statistics, Stanford University 1997
39. Kohavi, R and Wolpert, DH. Bias Plus Variance Decomposition for Zero-One Loss Functions. *Proceedings of the Thirteenth International Conference Machine Learning* 1996.
40. Kuncheva L.I., S.T. Hadjitodorov, Using Diversity in Cluster Ensembles, *Proc. IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands,* 2004; 1214-1219
41. KunchevaL.I. Diversity in multiple classifier systems (Ed.), *Information Fusion* 2005; 6 (1): 3-4
42. Kuncheva, LI, Whitaker, CJ, and Shipp, CA. Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis and Applications* 2003; 6:22-31
43. L. Lam and C. Y. Suen. Optimal combinations of pattern classifers. *Pattern Recognition Letters* 1995, 16(9):945-954.
44. Liaw A. and Wiener M. Classification and Regression by randomForest. R News, 2(3):18-22, December 2002
45. Lim N, Ahn H, Moon H. Classification by Ensembles of Random partitions of Logistic Regression models. *Journal of Computational Statistics and Data Analysis. Submitted.*
46. Luedi PP, Hartmink AJ, Jirtle, RL. Genome-wide prediction of imprinted murine genes. *Genome Research* 2005; 15:875-884.
47. Michiels S, Koscielny S, Hill C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet 2005*. 365(9458):488-92.

48. Miller, A. Subset Selection in Regression. 2nd Edition. Chapman and Hall/CRC 2002
49. Opitz, D. and Maclin, R. *Popular Ensemble Methods: An Empirical Study* 1999; 11:169-198
50. Pepe, MS, Tinaxi C, and Longton, G. Combining Predictors for Classification Using the Area under the Receiver Operating Characteristic Curve. *Biometrics* 2006; 62:221-229.
51. Pittman J, Huang E, Dressman H, Horng CF, Cheng SH, Tsou MH, Chen CM, Bild A, Iversen ES, Huang AT, Nevins JR, West M. Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes. *Proc Natl Acad Sci* 2004; 101(22): 8431–8436
52. Prentice, R.L. (1986). Correlated Binary Regression Using an Extended Beta-Binomial Distribution, with Discussion of Correlation Included by Covariate Measurement Error. *Journal of the American Statistical Association* 81:321–327.
53. Ripley BD. Pattern Recognition and Neural Networks, Cambridge University Press, 1996
54. Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S. Boosting the Margin: a new explanation for the effectiveness of Voting methods. *Annals of statistics* 1998, 26, 1651-1686.
55. Shatland ES, Kleitman K, Cain EM. A new strategy of model building in Proc Logistic with automatic variable selection, validation, shrinkage and model averaging. 2004; SUGI '29 Proceedings. Cary, NC: SAS Institute, Inc.
56. Simon R. When is a genomic classifier ready for prime time? *Nature Clinical practice Oncology* 2004; 1(1):4:5.
57. Singh,D., Febbo,P.G., Ross,K., Jackson,D.G., Manola,J., Ladd,C., Tamayo,P., Renshaw,A.A., D'Amico,A.V., Richie,J.P., Lander,E.S., Loda,M., Kantoff,P.W., Golub,T.R. and Sellers,W.R. Gene expression correlates of clinical prostate cancer behavior, *Cancer Cell* 2002; 1(2):203-209.
58. Strobl C, Bolesteiux, A, Zeilies A, Hothorn T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 2007, 8:25

59. Su, JQ and Liu, JS. Linear Combinations of Multiple Diagnostic Markers. *Journal of the American Statistical Association*. Vol. 88, No. 424. Dec 1993
60. Surowiecki, J. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*, 2004. Little, Brown ISBN 0-316-86173-1
61. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., and Feuston, B.P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Model 2003;* 43 (6):1947-1958
62. Tibshirani R, Hastie T, Narasimhan B, Chu G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS* 2002; 99(10):6567-6572
63. Tilstone C. DNA microarrays: Vital statistics. Nature 2003; 424:610-612
64. Tumer K and Ghosh J. Error Correlation and Error Reduction in Ensemble Classifiers. *Connection Science* 1996*;* 8(3/4):385–404
65. Tumer, K and Ghosh, J. Analysis of Decision Boundaries in linearly Combined Neural Classifiers. *Pattern Recognition* 1996; 29(2):341-348
66. Tumer K and Ghosh J. *Linear and order statistics combiners for pattern classification*. In A.J.C. Sharkey, editor, Combining Artificial Neural Nets, pages 127--161. Springer-Verlag, London, 1999.
67. Valentini G and Dietterich TG. Low Bias Bagged Support Vector Machines. In *Proc. ICML* 2003.
68. Valentini G and Dietterich TG. Bias variance analysis of Support Vector machines for the Development of SVM-based ensemble systems. *Journal of Machine learning Research.* 2004; 5, 725-775.
69. V. Vapnik and A. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities." *Theory of Probability and its Applications*, 16(2):264--280, 1971.
70. Vapnik, V. and Cortes, C. Support vector networks *Machine Learning* 1995; 20, 273–293.
71. Webb G.I. and Conilione P. Estimating Bias and Variance from data. Tech Report, 2006. Available at http://www.csse.monash.edu.au/~webb/

72. Wolpert DH. and Macready WG. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1997.
73. Wood AJ, Oakey RJ. Genomic Imprinting in Mammals: Emerging Themes and Established Theories. *PLoS Genet*. 2006; 2(11): e147
74. Woods K, Kegelmeyer, WP, and Bowyer, K. Combination of Multiple Classifiers Using Local Accuracy Estimates. *IEEE transactions on Pattern Analysis and Machine Intelligence* 1997; 19(4)
75. Zhang X, Lu X, Shi Q, Xu X, Leung HE, Harris LN, Iglehart JD, Miron A, Liu JS, Wong WH. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics* 2006. 7:197

# Appendix

## Appendix 1. Publicly available genomic data sets

| | Data set | web-site |
|---|---|---|
| 1. | Colon Data (Alon) | http://microarray.princeton.edu/oncology/ |
| 2. | Estrogen Data (Blair) | www.ams.sunysb.edu/~hahn/research/ |
| 3. | Prostate Data (Singh): | www.broad.mit.edu/cgi-bin/cancer/publications |
| 4. | Lymphoma data (Alizadeh): | http://llmpp.nih.gov/lymphoma/data.shtml |

## Appendix 2. Outline of procedure used to examine individual classifier performance (Chapter 7)

1. The data are split into a training set and test set using 4-fold cross validation. The test set consists of ¼ of the data, and the training set ¾.

*For each of the 4 folds of the CV:*

2. On the training set only, the variables are pre-screened using the BW statistic as defined in section 3.7. The top M ranked predictors are retained.
3. The classifiers discussed in section 3.6 using the packages detailed in section 3.8 are used to fit the training data based on the M predictors retained in step 2.

*On the test set:*

4. The classifiers are used to predict the class of the test set and the individual accuracies are estimated
5. Steps 1-4 are repeated 50 times (50 iterations of four-fold CV)
6. The mode of the predicted classes for classifier k (k=1,2,…,K) is determined. The observation is biased for classifier k if is the mode is not equal to the true class.
7. The variance is the proportion of times the prediction varies from the mode
8. The accuracy, variance and bias are averaged across all observations

# Appendix 3. Outline of procedure used to examine ensemble performance (Chapter 8)

1. The data are split into a training set and test set using 4-fold cross validation. The test set consists of ¼ of the data, and the training set ¾.

*For each of the 4 folds of the CV:*

2. On the training set only, the variables are pre-screened using the BW statistic. All features with a higher BW score than the best artificial variable are retained as part of the informative set (section 3.7)
3. The classifiers discussed in section 3.6 using the packages detailed in section 3.8 are used to fit the training data based on the M predictors retained in step 2.
4. A nested bootstrap analysis is performed within the training set to estimate the accuracy of each classifier to be used in construction of the combining weights (section 6.1)
5. For training observation i, all OOB bootstrap samples are used to compute accuracy.
6. Local weights (section 6.3) are determined by determining the nearest neighbor to the test point from the training set and using the weights from that observation based on bootstrap estimated accuracies.
7. Global weights are determined by averaging the bootstrap-based accuracy estimates over all observations

*On the test set:*

8. The classifiers are used to predict test set class and the individual accuracies are estimated
9. The ensemble-based prediction is computed using the prediction of each classifier at the test point and the weights derived by the training set.
10. Steps 1-4 are repeated 50 times (50 iterations of four-fold CV)
11. The mode of the predicted classes for classifier k (k=1,2,…,K) is determined. The observation is biased for classifier k if is the mode is not equal to the true class.
12. The variance is the proportion of times the prediction varies from the mode
13. The accuracy, variance and bias are averaged across all observations