MAJORITY VOTE CLASSIFIERS: THEORY AND APPLICATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Gareth James

May 1998

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Trevor Hastie
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Jerome Friedman

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Art Owen

Approved for the University Committee on Graduate Studies:

Abstract

The topic of this thesis is a special family of classifiers known as *Majority Vote Classifiers*. These work by producing a large number of classifications (often using a standard method such as a tree classifier) and classifying to the class receiving the largest number of votes or predictions. Some recent examples include Boosting (Freund and Schapire, 1996), Bagging (Breiman, 1996a) and Error Correcting Output Coding (ECOC) (Dietterich and Bakiri, 1995) . These classifiers have shown a great deal of promise but it is not fully understood how or why they work.

The thesis is split into two parts. In the first part we examine in detail the ECOC classifier. We show that it is in fact producing approximately unbiased probability estimates for each class and classifying to the maximum. It is therefore a method to approximate the Bayes Classifier. We also develop three extensions of ECOC and examine these new classifiers. Finally a detailed empirical study is made of nine different classifiers, including the ones mentioned above.

In the second part we examine in more generality why majority vote classifiers seem to work so well. Many theories have been suggested for the success of these classifiers but none provide a complete explanation. The theories tend to fall into one of two categories which we label *classical* and *modern*. The classical theories rely on generalizations of bias and variance ideas from regression theory. This is a very appealing concept. However, while this area still needs to be further explored, it is clear that the *nice* decompositions that arise from squared error loss do not hold for the 0-1 loss functions that are used for classification problems. As a result bias and variance ideas do not seem to be very useful.

The modern theories develop explanations that are more specific to classifiers. They work by defining a new quantity, call it M. An attempt is then made to relate a classifier's error rate to M. For example, a higher value of M might mean a lower error rate. In this case one would attempt to prove that certain classifiers work to increase M and hence reduce the error rate. The modern theories are still at an early stage and, as yet, have not been validated by any empirical results but seem to hold the potential to unlock some of the mystery surrounding these classifiers.

Acknowledgments

Where to start!

I am deeply grateful to my advisor Trevor Hastie. He provided the topic for this thesis as well as plenty of encouragement, many helpful suggestions and some dynamic debates! Trevor respected me enough to tell me when my ideas were bad as well as good. For that I would like to thank him.

Jerry Friedman and Art Owen deserve special mention for acting as members of my reading committee as well as providing useful suggestions throughout my time at Stanford. I would like to thank Richard Olshen for pointing out errors in my reasoning while still managing to imply I knew what I was talking about and no less importantly for helping me find a job.

I have many many reasons to thank my parents, Alison and Michael. Without them none of this would be possible. While we are on the subject I would like to thank the "department mother", Judi Davis. Judi was always happy to provide a topic of conversation that didn't involve statistics!

I have made several life long friends here. I know I echo the sentiments of many before me when I say how much I have enjoyed my time at Stanford. I am amazed that students ever choose to leave.

Catherine thank you for seeing me through the bad times as well as the good. This is not the end or even the beginning of the end but it is perhaps the end of the beginning.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Consider the digits in Figure 1.1 (Reprinted from Hastie and Tibshirani, 1994). These are scanned in images of hand written zip codes from the US Postal Service Zip Code Data Set. The US postal service wishes to design a system to read hand written zip codes on mail. In particular they want to :

1. scan in an image of a particular hand written digit,
2. convert the image into a pixel mapping (e.g. 16 by 16),
3. and use the pixel mapping to output a digit prediction between 0 and 9.

Of the above three steps, the third presents the most difficulty and is the general area that this thesis concerns.

1.1 Regression vs Classification

1.1.1 The Regression Problem

Suppose we observe pairs of observations $(x_i, y_i) \quad i = 1, \ldots, n$ where $x \in \mathcal{X} \subset \mathbb{R}^p$ and $y \in \mathcal{Y} \subset \mathbb{R}$. $\mathcal{X}$ is known as the predictor (or input) space and $\mathcal{Y}$ is the response (or output) space. The aim is to use these observations to estimate the relationship between X and Y and hence predict Y given X. Usually the relationship is denoted as follows

$$Y = f(X) + \epsilon$$

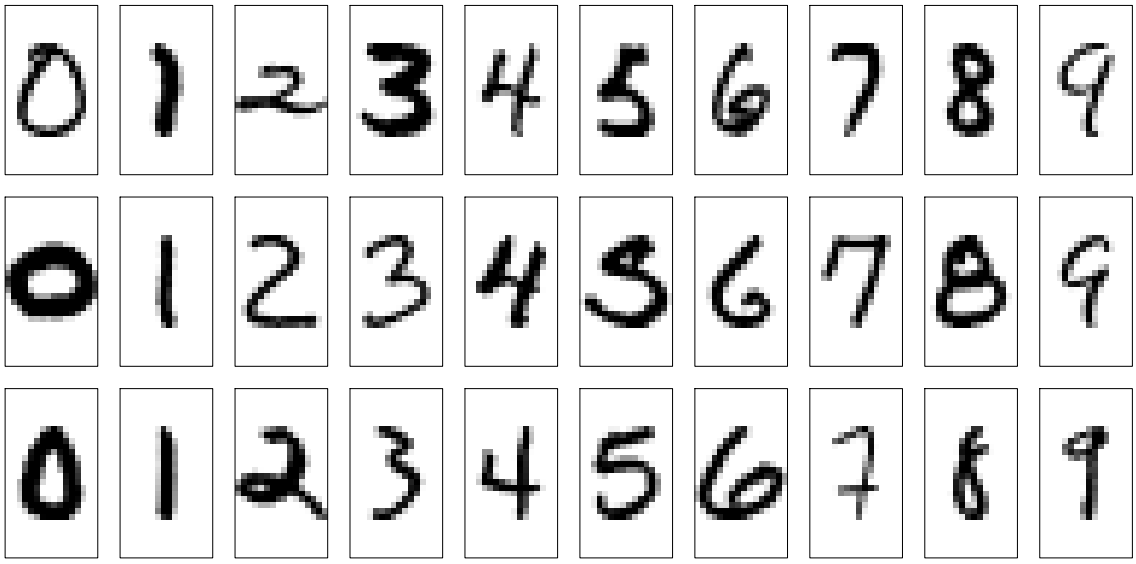

Figure 1.1: A plot of images from the US Postal Service Zip Code Data Set

where ϵ is a random variable with mean 0. So the problem of prediction reduces to one of estimating f based only on observing X and Y. This setting is known as a regression problem and is one of the most common and useful applications of statistics. There is a vast statistical literature in this area and for many situations the problem is essentially solved.

1.1.2 The Classification Problem

Now suppose we have an almost identical setup to that in Section 1.1.1. We again observe pairs of observations $(x_i, y_i) \quad i = 1, \ldots, n$ and wish to use these observations to form a prediction rule for Y given X. However, instead of $\mathcal{Y} \subset \mathbb{R}$ or even assuming Y is ordinal we now assume that Y is a categorical random variable i.e. Y has no ordering but simply denotes a *class* that the observation falls into. Examples include gender (male or female), region (North, South, East or West) or the letters of an alphabet (A,B, ... , Z). In statistics this situation is known as a *Classification Problem*, however, it is also known as *Pattern Recognition* in other fields.

Classification problems have an equally wide area of application to that of regression problems yet, in statistics, they have received much less attention and are far less well understood. Indeed it would be fair to say that statisticians play a fairly small role in the

work in this area. This is hard to understand because statistics has a lot to contribute to these problems.

Now let us briefly return to the zip code recognition example mentioned earlier. At first glance it is not obvious where statistics comes into this problem. However, each of the pixels (256 in the case of our 16 by 16 example) are coded as a number (usually ranging between 0 and 1) and can be treated as a predictor variable. We then have a 256 dimensional predictor space, $\mathcal{X} \subset \mathbb{R}^{256}$ and a response space containing 10 possible outcomes, $\mathcal{Y} = \{0, 1, \ldots, 9\}$. Our aim, then, is to use a set of *training data* (pairs of X and Y) to form a rule to predict Y (the digit) given X (the pixels). Such a prediction rule is known, in statistics, as a classifier. When the example is formulated in this manner it is clear that this is a classification problem to which we can apply all our statistical techniques.

In the next few sections we introduce some of the ideas and notation which are important in this area of statistics.

1.2 The Bayes Classifier

Suppose that Y takes on k different outcomes or *classes*. The task of predicting Y given X is known as a k class classification problem. We denote the conditional distribution of Y given X by q_i^X i.e.

$$q_i^X = Pr(Y = i|X)$$

Note, that $\mathbf{q}^X = (q_1^X, q_2^X, \ldots, q_k^X)$ is a function of X but the superscript X notation is dropped where there is likely to be no confusion.

Suppose that an oracle has given us $\mathbf{q}$ for every point X in the predictor space. What is the best possible classifier? The answer to this question depends on what we mean by *best*. A common definition of best is *lowest error rate* where the error rate is simply the percentage of *misclassifications* i.e. the fraction of the time that the classification is different from Y. This corresponds to a loss function which is 1 for a misclassification and 0 otherwise. Under this loss function the answer is to use the classifier, $C(X)$, which minimizes the expected

error rate.

$$\begin{aligned} E_X P(C(X) \neq Y|X) &= 1 - E_X P(C(X) = Y|X) \\ &= 1 - \sum_i E_X[P(C(X) = i)q_i^X] \end{aligned}$$

It is clear that

$$C(X) = \arg\max_i q_i^X$$

will minimize this quantity with the expected error rate equal to $E_X P(C(X) \neq Y|X) = 1 - E_X \max_i q_i^X$.

This classifier is known as the *Bayes Classifier* and the error rate it achieves is the *Bayes Error Rate*. Notice that the Bayes Error Rate is only 0 if $\max_i q_i^X = 1$ for all X. So if there is any randomness in Y given X the minimal achievable error rate will be greater than 0. This is equivalent to a regression problem where the minimum mean squared error between the prediction ($\hat{Y}$) and Y is equal to $Var(\epsilon)$ which is greater than 0 unless ϵ is identically equal to zero.

1.3 Some Standard Classifiers

We have shown that the Bayes Classifier is optimal in terms of achieving the minimum possible misclassification error. Unfortunately in practice we do not know $\mathbf{q}$ so can not produce this classifier. However the Bayes Classifier is still useful because it gives us a gold standard to aim for. Many classifiers attempt to approximate the Bayes Classifier by estimating $\mathbf{q}$ and then classifying to the maximum estimated probability. Notice that obtaining a *good* estimate of $\mathbf{q}$ is sufficient but not necessary to produce a good classifier.

Below we list a few examples of commonly used classifiers.

1.3.1 Tree Classifiers

Suppose all the variables in our predictor space are continuous. Then the first step for a tree classifier is to split the training data into two parts by choosing a single variable, say

x_i, and partitioning the data into parts where $x_i \leq t$ and $x_i > t$. The rule to determine both x_i and t is known as the splitting or partitioning rule. Next we examine these two sets of data and again split one of them into two parts. This partitioning continues until a stopping criterion is reached.

The procedure described above will result in a partitioning of the predictor space into hyper rectangles. These rectangles are often known as leaves or terminal nodes. We will refer to them as regions. We then examine the training data that falls in the same region as the test data point. A classification is made to the class with the largest number of training data points in this region. It is also possible to produce probability estimates by examining the proportion of training points from each class in the region.

There are several possible criterion to use for the splitting rule. One option is to choose the split that gives the largest reduction in deviance. The deviance of a tree is defined as

$$D = \sum_i D_i, \qquad D_i = -2 \sum_k n_{ik} \log p_{ik}$$

where n_{ik} are the number of observations from Class k in the ith terminal node and p_{ik} is the probability of an observation in the ith terminal node being from Class k. The partitioning ends when the reduction in error rate or deviance is below a threshold value.

It is also possible to construct regression trees when the response is continuous. There are various procedures for *pruning* the tree to reduce the number of terminal nodes. This tends to have the effect of reducing the variability at the expense of increased bias. A more detailed discussion of trees is given in Breiman et al., 1984.

1.3.2 LDA and QDA

Let π_Y denote the prior probabilities of the classes, and $p(X|Y)$ the densities of distributions of the observations for each class. Then the posterior distribution of the classes after observing X is :

$$p(Y|X) = \frac{\pi_Y p(X|Y)}{p(X)} \propto \pi_Y p(X|Y)$$

We know from Section 1.2 that the best classification is to the class with maximal $p(Y|X)$ or equivalently to the class with maximal $p(X|Y)\pi_Y$.

Now suppose the distribution for class Y is multivariate normal with mean μ_Y and covariance Σ_Y. By taking suitable transformations we see that the Bayes classification is to the class with minimum

$$\begin{aligned} Q_Y &= -2\log[p(X|Y)\pi_Y] \\ &= -2\log p(X|Y) - 2\log\pi_Y \\ &= (X-\mu_Y)\Sigma_Y^{-1}(X-\mu_Y)^T + \log|\Sigma_Y| - 2\log\pi_Y \end{aligned}$$

If we use the sample mean for each class to estimate μ_Y and the sample covariance matrix within each class to estimate Σ_Y then we can produce an estimate for Q_Y and classify to the class with lowest estimated Q_Y. The difference between the Q_Y for two classes is a quadratic function of X, so this method of classification is known as *Quadratic Discriminate Analysis* (QDA).

Further suppose that the classes have a common covariance matrix Σ. Differences in the Q_Y are then linear functions of X and we can maximize $-Q_Y/2$ or

$$L_Y = X\Sigma^{-1}\mu_Y^T - \mu_Y\Sigma^{-1}\mu_Y^T/2 + \log\pi_Y$$

This procedure is known as *Linear Discriminate Analysis* (LDA). For further details see Chapter 12 of Venables and Ripley, 1994.

1.3.3 K Nearest Neighbours

K Nearest Neighbours is an extremely simple classifier. To classify a new test point one simply finds the k closest training data points in the predictor space. One then classifies to the class which corresponds to the largest number of these training points. For example with 10 nearest neighbours one would find the 10 closest training points to each test point. If 5 of these points were from Class 2, 3 from Class 1 and 2 from Class 3 you would classify to Class 2.

In the event of a tie either a class can be chosen at random or no classification returned. It is common to use Euclidean distance to determine the closest training points though it is advisable to scale variables so that one direction does not dominate the classification. As k increases, the variability of the classification will tend to decrease at the expense of increased bias.

Although this is a very simple classifier, in practice it tends to work well on a large number of problems. We will use 1 nearest neighbour as a base line classifier to compare with the other methods suggested in this thesis.

1.4 Majority Vote Classifiers

Suppose for a certain classification problem we are given three different classification rules, $h_1(X)$, $h_2(X)$ and $h_3(X)$. Can we combine these three rules in such a way as to produce a classifier that is superior to any of the individual rules? The answer is yes under certain circumstances. A common way to combine these rules is to let

$$C(X) = \text{mode}\{h_1(X), h_2(X), h_3(X)\} \tag{1.1}$$

In other words, at each value of X classify to the class that receives the largest number of classifications (or *votes*). This family of classifiers are known as *Majority Vote Classifiers* or *Majority Vote Learners* (MaVLs pronounced Marvels).

As a simple example of the improvement that can be achieved using this method consider the following situation. In this example the predictor space, $\mathcal{X}$, is divided into three regions. In the first region h_1 and h_2 classify correctly but h_3 is incorrect, in the second h_1 and h_3 are correct but h_2 incorrect and in the last region h_2 and h_3 are correct but h_1 is incorrect. If a test point is equally likely to be in any of the three regions, each of the individual classifiers will be incorrect one third of the time. However, the combined classifier will always give the correct classification. Of course there is no guarantee that this will happen and it is possible (though uncommon) for the combined classifier to produce an inferior performance.

This procedure can be extended to any number of classifiers. It is also possible to put

more weight on certain classifiers. In general we define a majority vote classifier consisting of votes from rules $h_1, h_2, \ldots, h_B$ as follows

$$C(X) = \arg\max_i \sum_{j=1}^{B} w_j I(h_j(X) = i) \tag{1.2}$$

where $w_1, \ldots, w_B$ are weights that sum to 1 and $I(\cdot)$ is an indicator function. If the weights are set to $1/B$ this will give us the mode of $h_1, h_2, \ldots, h_B$ as in (1.1). A slightly different version can be obtained if the individual classifiers produce probability estimates,

$$C(X) = \arg\max_i \sum_{j=1}^{B} w_j \hat{p}_{ij} \tag{1.3}$$

where $\hat{p}_{ij}$ is the probability estimate from the jth classification rule for the ith class. We will refer to (1.2) as a Majority Vote Learner (MaVL) and (1.3) as a Semi Majority Vote Learner (Semi MaVL). Notice that MaVLs can be thought of as a special case of Semi MaVLs where all the probability estimates are either zero or one.

Majority Vote Classifiers are the central focus of this thesis. Over the last couple of years several classifiers falling into this family have demonstrated an ability to produce very accurate classification rules. As a result a lot of effort has been expended in trying to explain their success. In this thesis we survey some of the more successful MaVLs that have been developed as well as introducing a few new ones. We also discuss the theories that have been proposed and give some new insights of our own.

1.5 Plug in Classification Techniques

Plug in Classification Techniques (or PICTs) are algorithms that take a standard classifier, for example a tree classifier or LDA, and transforms it in some way to, hopefully, improve its accuracy. MaVLs which were introduced in the previous section are examples of PICTs. However, there are many PICTs which are not MaVLs. For example the *Pairwise Coupling* procedure, first suggested by Friedman (Friedman, 1996a) and later extended by Hastie and Tibshirani (Hastie and Tibshirani, 1996), is not a MaVL but is a PICT. Most of the classifiers introduced in Chapter 2 are examples of PICTs.

1.6 Summary of Chapters

In Chapter 2 we study a recently proposed classifier that is motivated by Error Correcting Output Coding ideas. While it is a PICT it does not quite fall into the family of MaVLs or Semi MaVLs. However, we demonstrate that it is an approximation to a classifier which we call the *Substitution PICT* and that this classifier is a Semi MaVL. As well as the Substitution PICT we also introduce two new classifiers, the *Regression* and *Centroid PICTs*, as well as two previously suggested MaVLs, *Bagging* and *Boosting*. At the end of the chapter we provide an experimental comparison between these alternative methods on several different data sets.

In Chapter 3 we explore theoretical explanations of MaVLs that rely on generalizations of the concepts of bias and variance from regression theory. We call this group of ideas *Classical*. It turns out that the best way to generalize these concepts is not obvious and as a consequence many alternative definitions have been proposed. We provide our own definitions as well as surveying the alternative suggestions. We give a case study where we implement some of the classical ideas on the Substitution PICT. All the definitions for bias and variance provide slightly different decompositions of the prediction error on any given data set so we provide an experimental comparison on several sets of data.

It turns out that there are some severe problems with attempting to generalize bias and variance to a 0-1 loss function. As a result Chapter 4 surveys a new class of theories which we call *Modern*. These modern ideas are more specifically tailored to classification problems and 0-1 loss functions. They involve producing bounds on the test error rate in terms of a quantity called the *Margin*. Unfortunately these bounds tend not to be tight so in practice they often work poorly. We suggest an alternative use of the margin called the Normal Model. This seems to produce superior performance on real data. The chapter concludes with a summary of the thesis.

In the appendix we give proofs for all the theorems contained in the thesis.

Chapter 2

Plug in Classification Techniques (PICTs)

In this chapter we will introduce some *standard* Majority Vote Classifiers, Bagging, Boosting and ECOC, as well as some new, Substitution, Regression and Centroid classifiers. The new classifiers are all adaptions of ECOC. A large proportion of the chapter is devoted to studying the ECOC PICT and its spin offs. However, Section 2.5 discusses the Bagging and Boosting algorithms and Section 2.6 provides experimental comparisons between the various classifiers.

2.1 The Error Correcting Output Coding Classifier (ECOC)

2.1.1 The Algorithm

Dietterich and Bakiri, 1995 suggested an algorithm, motivated by Error Correcting Output Coding Theory, for solving a k class classification problem using binary classifiers. We will refer to this classifier as the ECOC PICT.

ECOC Algorithm

- Produce a k by B (B large) binary coding matrix, i.e. a matrix of zeros and ones. We will denote this matrix by Z, its i,jth component by Z_{ij}, its ith row by $\mathbf{Z}_i$ and its jth column by $\mathbf{Z}^j$. Table 2.1 provides an example of a possible coding matrix for a 10 class problem.
- Use the first column of the coding matrix ($\mathbf{Z}^1$) to create two *super* groups by assigning all classes with a one in the corresponding element of $\mathbf{Z}^1$ to super group one and all other classes to super group zero. So for example, when using the coding matrix on page 12, one would assign classes 0, 2, 4, 6 and 8 to super group one and the others to super group zero.
- Train a Base Classifier on the new two class problem.
- Repeat the process for each of the B columns ($\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^B$) to produce B trained classifiers.
- To a new test point apply each of the B classifiers. Each classifier will produce $\hat{p}_j$ which is the estimated probability the test point comes from the jth super group one. This will produce a vector of probability estimates, $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_B)^T$.
- To classify the point calculate $L_i = \sum_{j=1}^{B} |\hat{p}_j - Z_{ij}|$ for each of the k classes (i.e. for i from 1 to k). This is the L1 distance between $\hat{\mathbf{p}}$ and $\mathbf{Z}_i$ (the ith row of Z). Classify to the class with lowest L1 distance or equivalently $\arg\min_i L_i$

This algorithm does not specify either how to create the coding matrix or what Base Classifier to use. These are obviously important components to the algorithm. The choice of coding matrix is discussed in detail in Section 2.2. Dietterich found that the best results were obtained by using a *tree* as the Base Classifier but in principle any binary classifier will do. All the experiments in this thesis use a standard tree classifier as the Base Classifier.

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$	$\mathbf{Z}^4$	$\mathbf{Z}^5$	$\mathbf{Z}^6$	$\cdots$	$\mathbf{Z}^{15}$
0	1	1	0	0	0	0	$\cdots$	1
1	0	0	1	1	1	1	$\cdots$	0
2	1	0	0	1	0	0	$\cdots$	1
3	0	0	1	1	0	1	$\cdots$	1
4	1	1	1	0	1	0	$\cdots$	0
5	0	1	0	0	1	1	$\cdots$	0
6	1	0	1	1	1	0	$\cdots$	1
7	0	0	0	1	1	1	$\cdots$	0
8	1	1	0	1	0	1	$\cdots$	1
9	0	1	1	1	0	0	$\cdots$	0

Table 2.1: A possible coding matrix for a 10 class problem

Note however that, unless otherwise stated, the theorems are general to any Base Classifier.

An Example of the ECOC Classification Step

As an example of how the final classification step works consider the following simplified scenario. Here we have only 4 classes and 5 columns in the coding matrix.

$Z =$

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$	$\mathbf{Z}^4$	$\mathbf{Z}^5$
1	1	1	0	0	0
2	0	0	1	1	0
3	1	0	0	1	1
4	1	1	1	1	0

We have trained the Base Classifier on the various super groupings and for a given test point of interest it has produced the following probability estimates.

$$\hat{\mathbf{p}} = (0.3, 0.2, 0.8, 0.9, 0.1)$$

One then calculates the L1 distance between each row and $\hat{\mathbf{p}}$. For example the L1 distance for the first row is $|0.3 - 1| + |0.2 - 1| + |0.8 - 0| + |0.9 - 0| + |0.1 - 0| = 3.3$.

If one continues this process for each class the following table is obtained.

Class	1	2	3	4
L1 Distance	3.3	0.9	2.7	1.9

The algorithm would then classify to Class 2 because it has the lowest L1 distance. Notice that $\hat{\mathbf{p}}$ looks far more similar to the second row than any of the others.

2.1.2 Original (Heuristic) Motivation

Dietterich's original motivation was roughly the following. Each row in the coding matrix corresponds to a unique (non-minimal) coding for the appropriate class. Now we would expect that if the correct class was in super group one then $\hat{p}_j$ would be close to 1 and if the correct class was in super group zero then $\hat{p}_j$ would be close to 0. Therefore we can think of $\hat{\mathbf{p}}$ as an approximation to the coding for the true class. So that we want to classify to the row or class that is closest to $\hat{\mathbf{p}}$ in some sense. Dietterich used the L1 metric as the distance measure.

Another way to think about this is that one would expect $|\hat{p}_j - Z_{ij}|$ to be low if i is the correct class. So that $L_i = \sum_{j=1}^{B} |\hat{p}_j - Z_{ij}|$ should be low if i is the correct class. Hence we classify to $\arg\min_i L_i$.

It is possible to produce a unique coding for each class provided $B \geq \log_2 k$. However if B is low, so there is no redundancy in the code, a misclassification by any single classifier (i.e. $|\hat{p}_j - Z_{ij}|$ close to 1 rather than 0) could cause the final classification to be incorrect. On the other hand if there is a redundancy built into the coding then it is possible to *correct* a certain number of mistakes and still classify to the correct class. It was Dietterich's belief that by using a matrix with a large degree of redundancy it would be possible to produce a classifier that made very few overall classification errors even if some of the individual Base Classifiers were incorrect.

2.1.3 The One vs Rest PICT

A simple but commonly used method (see for example Dietterich and Bakiri, 1995 or Nilsson, 1965) for handling a multi-class problem, when one has only a binary classifier, is to do the following.

One vs Rest Algorithm

- Compare Class 1 to the super group made up of all the other classes and produce a probability estimate that the test point comes from Class 1.
- Repeat this process for each of the k different classes.
- Classify to the class with the largest probability estimate.

We will call this the *One vs Rest PICT*. Notice that, if one uses the identity coding matrix for the ECOC PICT, the ECOC and One vs Rest PICTs are identical. So the One vs Rest PICT is a special case of the ECOC PICT. In Section 2.2.1 we will use this fact to examine the performance of the One vs Rest PICT by considering properties of the identity coding matrix.

2.1.4 An Alternative Way of Viewing the ECOC PICT

It is possible to view the ECOC PICT as performing a change of variables. For any point in the predictor space the ECOC algorithm will produce a B dimensional vector, $\hat{\mathbf{p}}$. This vector will be contained in a B dimensional unit cube. Each row of the coding matrix, corresponding to a vertex of the cube, will be a *target point*. For any given test point we would end up with a new point in the unit cube and classify to the closest target vertex, in L1 distance.

Figure 2.1 provides an illustration. Here we have a 3 dimensional cube. The solid circles represent target vertices and the open circles are training data. The "x" in the bottom left hand corner represents a test data point. Clearly this point is closest to the Class 1 vertex so the ECOC PICT would classify to that class. This alternative, geometric, way of viewing the procedure turns out to be useful. In particular, in Section 2.3.2, we will see that it leads to alternative classification procedures.

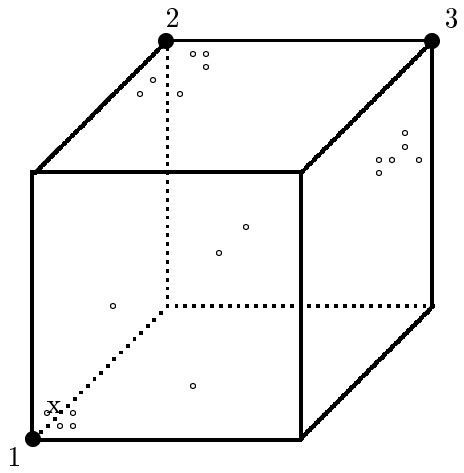

Figure 2.1: A multidimensional cube. The open circles are training points. The solid circles represent classes. A new point will be classified to the class corresponding to the closest vertex of the cube in L1 distance.

2.1.5 An Example

Figure 2.2 provides an illustration of the improvements in error rate that are possible by using ECOC. The plot shows test error rates on the Letter data set (see Sections 2.2.2 and 2.6 for descriptions of the data set) vs B, the number of columns in the coding matrix. Three classifiers are compared : the ECOC PICT, a Tree Classifier and 1 Nearest Neighbour. The ECOC classifier used a tree classifier as the Base Classifier to maintain comparability. It is clear from the plot that, provided B is large enough, the ECOC PICT produces large improvements over the tree classifier. It also produces a significant reduction over 1 nearest neighbour which is often a difficult classifier to *beat*. The full data from this experiment are given in Table 2.12 in Section 2.6.2.

2.2 Understanding the ECOC PICT

It is clear from Section 2.1.5 that the ECOC PICT can produce large reductions in the error rate. Section 2.6 as well as Dietterich and Bakiri, 1995 provide further evidence for the success of this procedure. However little is understood, beyond the vague motivation given in Section 2.1.2, about how the classifier works.

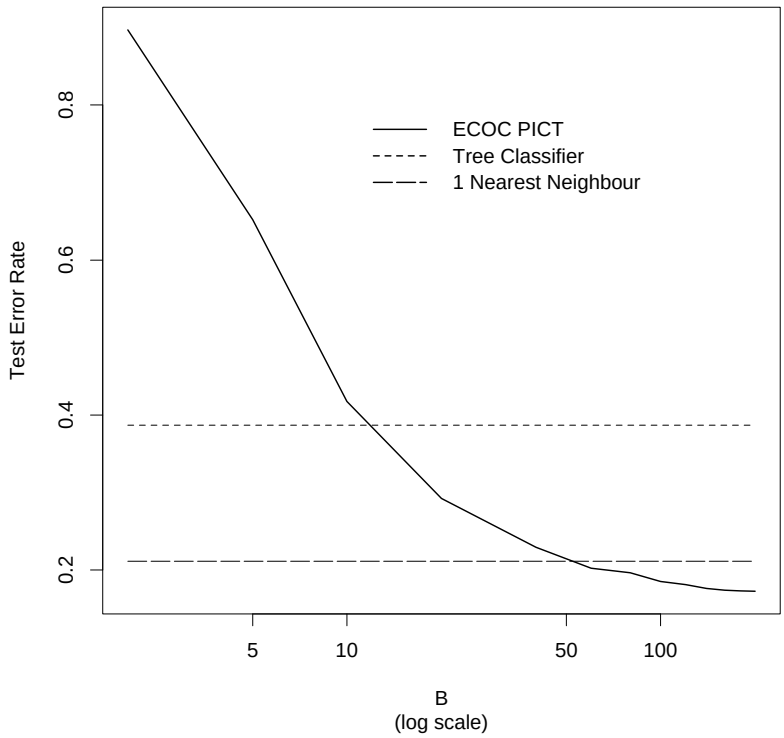

Figure 2.2: Test Error Rates on the Letter Data Set using ECOC, a Tree classifier and 1 nearest neighbour.

A key component in the performance of the ECOC PICT is the choice of the coding matrix. The coding matrix determines the arrangement of classes into super groups. One might imagine that certain arrangements are *better* than others so that one matrix may produce superior classifiers to another. In Section 2.1.1 a method for choosing the coding matrix was not provided. In fact there are at least three possible ways to produce such a matrix.

- **Deterministic**. This is the approach that has typically been used in the past. As a result of the original motivation for the ECOC PICT , a great deal of effort has been devoted to choosing an *optimal* coding matrix. In other words for any given values of k and B it was believed that there was one optimal matrix which could be chosen independently from the underlying data set. We call this the deterministic approach

because the matrix is fixed. Under this approach an attempt is made to produce a matrix with maximal Hamming distance between pairs of rows and pairs of columns. The Hamming distance between two binary vectors is the number of elements where the two vectors differ. The separation between rows is to allow as many errors, in individual classifiers, to be corrected as possible. The column separation is to produce as many different groupings as possible.

- **Random**. With this approach each element of the coding matrix is chosen to be zero or one with equal probability independently of any other element. In Section 2.2.2 we examine properties of the ECOC PICT with a random matrix in some detail. We present results which indicate that this approach will produce error rates that are close to those of the best deterministic matrix.

- **Trained**. It seems reasonable to suppose that certain groupings of classes make more sense than others and that it may be possible to learn this from the training data. In this last approach an attempt is made to use the training data to determine *good* groupings and hence a good coding matrix to use.

In the following three sections we examine the various approaches.

2.2.1 A Deterministic Coding Matrix

A simple example illustrates a disturbing flaw, with the ECOC PICT, when we use a deterministic coding matrix. In this example we have a 4 class problem ($k = 4$) and we use 3 columns in the coding matrix ($B = 3$).

$Z =$

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$
1	0	1	1
2	0	0	0
3	1	1	0
4	1	0	0

There is a redundancy in the coding because we only require 2 bits to uniquely identify 4 different classes. Now suppose at a given point in our predictor space the true distribution of posterior probabilities is as follows:

Class	1	2	3	4
q_i	0.4	0.1	0.3	0.2

It is clear that Class 1 is the Bayes Class, in as much as it has the largest posterior probability, so that the best possible classification is to this class. Indeed this is what the Bayes Classifier would give us. Now suppose our Base Classifier is estimating these probabilities perfectly. We would hope that, in this extremely favourable situation, the ECOC PICT would also classify to Class 1. In fact this is **not** the case.

If we combine this *perfect* classifier with the coding matrix it will produce the following vector of probabilities for super group one.

$$\hat{\mathbf{p}} = (0.5, 0.7, 0.4)$$

For example $\hat{p}_1 = q_3 + q_4 = 0.3 + 0.2 = 0.5$ because Classes 3 and 4 form super group one for the first column. Now if we calculate the L1 distance between $\hat{\mathbf{p}}$ and each row of Z we get the following:

Class	1	2	3	4
L_i	1.4	1.6	1.2	1.7

This means that the ECOC PICT would choose Class 3!

It turns out that the problem is not peculiar to this particular matrix . Theorem 1 gives very general conditions on the coding matrix under which it will always be possible to cause the ECOC PICT to produce an *incorrect* classification, even when the correct super group probabilities are used.

Bayes Consistency

Intuitively one would hope that any time you used a PICT classifier, such as ECOC, you could guarantee a good classification provided the Base Classifier was good enough. In other words one would like the PICT to produce the *Bayes Classification* whenever the Bayes Classifier is used as the Base Classifier. A PICT with this property is said to be Bayes Consistent.

Definition 1 *A PICT is said to be Bayes Consistent if, for any test set, it always classifies to the Bayes Class when the Base Classifier is the Bayes Classifier.*

Bayes Consistency implies a type of consistency in the PICT. Under continuity assumptions, it implies that, if the Base Classifier converges to the Bayes Classification rule, as for example, the training sample size increases, then so will the PICT.

Is the ECOC PICT, with a Deterministic Coding Matrix, Bayes Consistent?

The ECOC PICT will be Bayes Consistent iff

$$\underbrace{\arg\max_i q_i}_{\text{Bayes Classifier}} = \underbrace{\arg\min_i L_i}_{\text{ECOC Classifier}} \tag{2.1}$$

However Lemma 1 shows that each L_i can be written as a function of q, the posterior probabilities, and Z_{ij}, the individual elements of the coding matrix.

Lemma 1 *If one uses a deterministic coding matrix and the Bayes Classifier as the Base Classifier then*

$$L_i = \sum_{l \neq i} q_l \sum_{j=1}^{B} (Z_{lj} - Z_{ij})^2 \quad i = 1, \ldots, k$$

It is not clear from this expression why there should be any guarantee that (2.1) will hold. In fact Theorem 1 shows that only in very restricted circumstances will the ECOC PICT be Bayes Consistent.

Theorem 1 *The ECOC PICT is Bayes Consistent iff the Hamming distance between every pair of rows of the coding matrix is equal.*

For general B and k there is no known way to generate such a matrix. There are a couple of special cases that do fulfill this property. One is a matrix with all 2^k possible columns, for example.

$Z =$

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$	$\mathbf{Z}^4$	$\mathbf{Z}^5$	$\mathbf{Z}^6$	$\mathbf{Z}^7$	$\mathbf{Z}^8$
1	0	0	0	0	1	1	1	1
2	0	0	1	1	0	0	1	1
3	0	1	0	1	0	1	0	1

This matrix may work well but, if k is anything other than very small, computing 2^k possible classifiers will not be computationally feasible. For example one of the data sets used in Section 2.6 has 26 classes which would mean over 67 million possible columns!

Another matrix that fulfills this property is the identity matrix, for example.

$Z =$

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$	$\mathbf{Z}^4$
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

In Section 2.1.3 we saw that the ECOC PICT with an identity coding matrix is equivalent to the One vs Rest PICT. Therefore Theorem 1 implies that the the One vs Rest PICT is Bayes Consistent, indicating that it is a reasonable procedure to use if the Base Classifier is producing good probability estimates. However in practice the One vs Rest PICT tends to perform poorly because the coding matrix has too few columns, an uneven spread of classes and a low level of redundancy (the Hamming distance between pairs of rows is only 2). It is likely that the low number of columns is the largest problem with this matrix. In Section 2.6 it is clear that the ECOC PICT performs poorly unless $B >> k$.

Therefore we see that for any computationally feasible and practically useful deterministic matrix the ECOC PICT will not be Bayes Consistent.

2.2.2 A Random Coding Matrix

We have seen in the previous section that there are potential problems with using a deterministic matrix. Indeed it is not at all clear why a coding that is chosen independently from the underlying data should be optimal for every data distribution. Intuitively it seems that if we are going to choose the matrix independently from the training data then a random coding may work just as well as a designed coding. By random we mean choosing each element of the matrix as a zero or one with equal probability.

In fact, when we randomly generate the coding matrix, the ECOC PICT possesses a number of desirable properties. In Section 1.3 it was noted that a number of classifiers work

by estimating the posterior probabilities and then classifying to the maximum. It turns out that, provided a random coding matrix is used, the ECOC PICT is producing unbiased probability estimates and then classifying to the maximum of these.

Unbiased Probability Estimates

It is not obvious why this is the case. Certainly the L1 distances themselves are not unbiased probability estimates. In general they will increase with B. Besides the ECOC PICT is classifying to the minimum rather than maximum of these quantities. However, consider the following transformation of L_i.

$$\bar{D}_i = 1 - \frac{2L_i}{B} = \frac{1}{B}\sum_{j=1}^{B}(1 - 2|\hat{p}_j - Z_{ij}|) \quad \text{for} \quad i = 1, \ldots, k \tag{2.2}$$

Notice that $\bar{D}_i$ is simply a monotone decreasing transformation of L_i. As a result

$$\arg\min_i L_i = \arg\max_i \bar{D}_i$$

so classifying to the largest value of $\bar{D}_i$ produces identical classifications to the ECOC PICT e.g. consider the example in Section 2.1.1.

Class	1	2	3	4
L_i	3.3	0.9	2.7	1.9
$\bar{D}_i$	-0.32	0.64	-0.08	0.24

Notice that not only does Class 2 have the lowest value of L_i but it also has the largest value of $\bar{D}_i$ so under either approach a classification is made to that class.

Theorem 2 shows that $\bar{D}_i$ is in fact an unbiased estimate for q_i and therefore the ECOC PICT with a random coding matrix is performing an approximation to the Bayes Classifier.

Theorem 2 *Suppose that*

$$E_{\mathcal{T}}[\hat{p}_j \mid Z, X] = \sum_{i=1}^{k} Z_{ij} q_i = \mathbf{Z}^{j^T}\boldsymbol{q} \quad j = 1, \ldots, B \tag{2.3}$$

Then under this assumption for a randomly generated coding matrix

$$E_{\mathcal{T},Z}\bar{D}_i = q_i \quad i = 1, \ldots, k$$

$\mathcal{T}$ is the training set so $E_{\mathcal{T}}$ denotes expectation over all possible training sets of a fixed size and $E_{\mathcal{T},Z}$ denotes expectation over training sets and random matrices. Assumption 2.3 is an unbiasedness assumption. It states that on average the Base Classifier will estimate the probability of being in super group one correctly. In the experience of the author, the assumption seems to generally be good for non-parametric classifiers such as CART or other tree based procedures but less realistic for highly parametric methods such as LDA.

Theorem 2 provides the first theoretical evidence for the success of the ECOC PICT. It basicly tells us that, provided the Base Classifier is producing approximately unbiased probability estimates and the coding matrix is random, the ECOC PICT is classifying to the maximum among approximately unbiased probability estimates.

Limiting Properties of the ECOC PICT

Of course it is still possible for unbiased probability estimates to produce a bad classification rule. Therefore it is of interest to study the distribution of $\bar{D}_i$. While it is not easy to examine this directly, it is possible to characterize the limiting distribution. Let

$$\mu_i = E_Z(\bar{D}_i \mid \mathcal{T}) = E_Z(1 - 2|\hat{p}_1 - Z_{i1}| \mid \mathcal{T}) \tag{2.4}$$

Note that μ_i is the conditional expectation of $\bar{D}_i$. Then, conditional on $\mathcal{T}$, as B approaches infinity, $\sqrt{B}(\bar{D}_i - \mu_i)$ will converge to a normal random variable and $\bar{D}_i$ will converge to μ_i almost surely. This also implies that the ECOC PICT will converge to a limiting classifier consisting of classifying to $\arg\max_i \mu_i$. Theorem 3 provides a summary of these results.

Theorem 3 *Suppose that* $\arg\max_i \mu_i$ *is unique i.e. there are no ties in the* μ*s. Then for a random coding matrix, conditional on* $\mathcal{T}$*, the following results hold for any Base Classifier.*

1.

$$\sqrt{B}(\bar{D}_i - \mu_i) \Rightarrow N(0, \sigma_i^2) \quad i = 1, \ldots, k$$

2.

$$\bar{D}_i \to \mu_i \quad a.s. \quad i = 1, \ldots, k$$

3.

$$\lim_{B\to\infty} \arg\max_i \bar{D}_i = \arg\max_i \mu_i \quad a.s.$$

Notice that Theorem 2 along with (2.4) implies that

$$E_{\mathcal{T},Z}\bar{D}_i = E_{\mathcal{T}}\mu_i = q_i \quad i = 1, \ldots, k \tag{2.5}$$

(2.5), along with the final result of Theorem 3, mean that not only is the ECOC PICT classifying based on unbiased probability estimates but so is its limiting classifier.

These asymptotic results hold for any Base Classifier. It is interesting to note the effect if the Bayes Classifier is used as the Base Classifier.

Theorem 4 *When the Bayes Classifier is used as the Base Classifier*

$$\mu_i = q_i \tag{2.6}$$

By combining Theorems 3 and 4 we get Corollary 1.

Corollary 1 *For a random coding matrix the following results hold if the Bayes Classifier is used as the Base Classifier.*

1.

$$\sqrt{B}(\bar{D}_i - q_i) \Rightarrow N(0, \sigma_i^2) \quad i = 1, \ldots, K$$

2.

$$\bar{D}_i \to q_i \quad a.s.$$

3.

$$\lim_{B\to\infty} \arg\max_i \bar{D}_i = \arg\max_i q_i \quad a.s.$$

The final result of Corollary 1 implies that asymptotically, as B approaches infinity, the ECOC PICT will become Bayes Consistent, provided a random

coding matrix is used. This provides further motivation for using a random rather than deterministic matrix.

Rates of Convergence

The results from the previous section provide strong motivation, in the limit, for the ECOC PICT. We know that it is converging to a Bayes Consistent classifier and that the probability estimates are unbiased and converging to unbiased estimates (μ_i) of the posterior probabilities. These results guarantee good theoretical properties, *provided B is large enough.*

What we mean by *large* is highly dependent on the rate of convergence. Theorem 3 shows that $\sqrt{B}(\bar{D}_i - \mu_i)$ is converging to a normal random variable. This implies that $\bar{D}_i$ is converging to its mean at a rate of only $1/\sqrt{B}$. This is a fairly slow rate of convergence. However, we are not interested in the deviation of $\bar{D}_i$ from its mean. We are interested in the deviation of the error rate of ECOC, $\arg\max_i \bar{D}_i$, from the error rate of its limit, $\arg\max_i \mu_i$. Theorem 5 shows that the deviation between error rates approaches zero exponentially fast.

Theorem 5 *If the coding matrix is randomly chosen then, conditional on $\mathcal{T}$, for any fixed X*

$$\begin{aligned} |\textit{ECOC error rate} - \textit{Limiting error rate}| &\leq Pr_Z(\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i | \mathcal{T}) \\ &\leq (k-1)e^{-mB} \end{aligned}$$

where $m = (\mu_{(k)} - \mu_{(k-1)})/8$ and $\mu_{(i)}$ is the ith order statistic.

Note that Theorem 5 does not depend on Assumption 2.3. This tells us that the error rate for the ECOC PICT is equal to the error rate using $\arg\max_i \mu_i$ plus a term which decreases exponentially in the limit. The result can be proved using Hoeffding's inequality (Hoeffding, 1963).

As a simple corollary of Theorem 5 it is possible to show that, when the Bayes Classifier is used as the Base Classifier, the ECOC error rate approaches the Bayes error rate exponentially fast.

		B				
		10	50	100	200	1000
$\mu_{(k)} - \mu_{(k-1)}$	0.2	1	1	1	1	0.061
	0.5	1	1	0.395	0.017	0
	0.9	1	0.032	0	0	0

Table 2.2: Upper Bounds on the difference between ECOC and limiting classifier error rates for various combinations of B and $\mu_{(k)} - \mu_{(k-1)}$ on a 10 class problem ($k = 10$). Note the smallest of the 0's is actually 10^{-48}.

Corollary 2 *When the Bayes Classifier is the Base Classifier the following inequality holds*

$$|ECOC \text{ error rate} - \text{Bayes error rate}| \leq (k-1)e^{-mB}$$

where $m = (q_{(k)} - q_{(k-1)})/8$.

Corollary 2 shows that, while for any finite B the ECOC PICT will not be Bayes Consistent, its error rate will be exponentially close to that of a Bayes Consistent classifier.

Notice that the convergence will be fast if $\mu_{(k)} >> \mu_{(k-1)}$ but will be much slower if $\mu_{(k)} \approx \mu_{(k-1)}$. Table 2.2 gives the upper bounds on the difference in error rate for various values of $\mu_{(k)} - \mu_{(k-1)}$.

Of course Theorem 5 and Corollary 2 only give upper bounds on the error rate and do not necessarily indicate the behaviour for smaller values of B. If $\mu_{(k)} - \mu_{(k-1)}$ is very small it is possible that B would need to be large before the exponential convergence *kicks* in. Under certain conditions a Taylor expansion indicates that to first order

$$Pr(\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i) \approx 0.5 - m\sqrt{B}$$

for small values of $m\sqrt{B}$. So one might expect that for smaller values of B the error rate decreases as some power of B but that as B increases the change looks more and more exponential. If m is not too small the movement through the powers should be fast but if

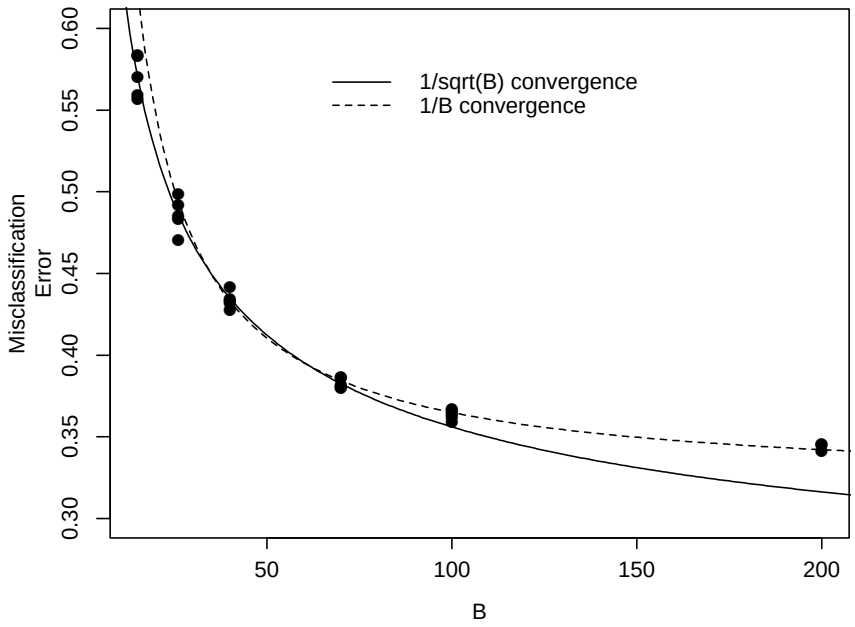

Figure 2.3: Best fit curves for rates $1/\sqrt{B}$ and $1/B$

m is small this transformation may require a large value of B.

Simulation Study

To evaluate how this theory works in practice a simulation study was performed on the Letter data set (available from the Irvine Repository of machine learning). This data set consists of 16 predictor variables and a 26 class response variable. The response variable consists of the letters of the English alphabet from A to Z. The predictor variables consist of various summary statistics. For example average height, average width, variance of height etc. For this study random matrices with differing row lengths were generated. The row lengths were $15, 26, 40, 70, 100, 200$. For each row length five matrices were created for a total of 30 matrices. 20 training sets were generated from the data by random sampling without replacement. Then for each of the 30 matrices the average error rate was calculated (on a separate test set of data) over the 20 training sets when the ECOC PICT was applied to that matrix. A tree classifier produced by Splus was used as the Base Classifier.

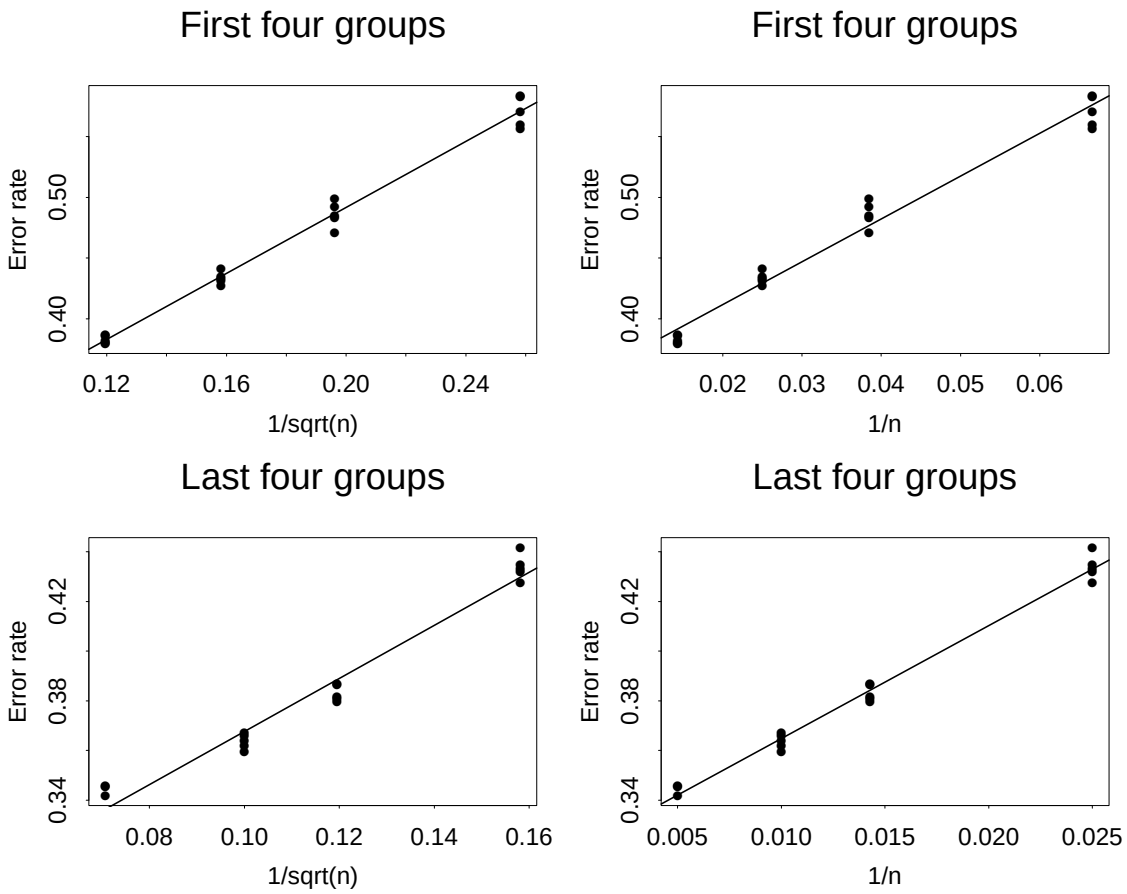

Figure 2.4: An alternative representation of Figure 2.3

Figure 2.3 illustrates the results. Each point is the averaged test error rate for one of the 30 random matrices. Here we have two curves. The lower curve is the best fit of $1/\sqrt{B}$ to the first four groups ($B = 15, 26, 40, 70$). It fits those groups well but under-predicts errors for the last two groups. The upper curve is the best fit of $1/B$ to the last four groups ($B = 40, 70, 100, 200$). It fits those groups well but over-predicts errors for the first two groups. This supports our hypothesis that the error rate is moving through the powers of B towards an exponential fit. We can see from the figure that even for relatively low values of B such as 100 the reduction in error rate has slowed substantially.

Figure 2.4 provides an alternative way of viewing these results. The first column shows plots of test error vs $1/\sqrt{B}$. The first plot illustrates that there is a strong linear relationship

for the first four groups. However, the second plot, for the last four groups, does not exhibit nearly such a linear relationship. The second column shows plots of test error rate vs $1/B$. Here the relationship is reversed. It is clear that for smaller values of B the error rate is declining at a rate of $1/\sqrt{B}$ but as B increases this has slowed to $1/B$ and we are rapidly reaching an exponential rate of convergence.

Random vs Deterministic Matrices

The coding matrix can be viewed as a method for sampling from a finite population $(1 - 2|\hat{p}_j - Z_{ij}|)$. Theorem 2 tells us that the mean of this population

$$\mu_i = E_Z(1 - 2|\hat{p}_1 - Z_{i1}| \mid \mathcal{T})$$

is an unbiased estimate for q_i. This implies we wish to choose a coding matrix which will produce the best possible estimate of μ_i. Theorem 5 as well as the simulation results from Section 2.2.2 show that the difference between the error rate from a random matrix and that of a perfect estimate will be exponentially small. No results have been given to indicate that a designed matrix will perform better than a random one and Theorem 5 shows that **at best** a designed matrix can only produce a very slight improvement. On the other hand, unless we are very careful, a designed sampling provides no guarantee of producing a reasonable estimate of μ_i or q_i. Therefore there seem to be clear potential problems with using such a matrix and little possible benefit over using a random coding.

Of course it may be possible to improve on a random sampling by using the training data to produce the coding matrix. This would allow the training data to influence the sampling procedure and hence estimate a different quantity.

2.2.3 Training the Coding Matrix

This is an area that has not been fully explored. Several researchers have attempted to gain reductions in the error rate by adaptively choosing the coding matrix according to groupings that the training data suggest are appropriate. To date these methods have met with limited success.

One of the possible reasons for this lack of success is that there is extra variability

introduced by using the training data to choose the coding matrix. It is possible that this extra variability could outweigh any gains from adaptively training the coding matrix.

2.3 The Regression and Centroid PICTs

Using L1 distance between $\hat{\mathbf{p}}$ and $\mathbf{Z}_i$ is one possible way to transform $\hat{\mathbf{p}}$ into a k class classification rule. However it is certainly not the only way. In this section we present two alternative classifiers, the *Regression* and *Centroid* PICTs. They both generate the vector $\hat{\mathbf{p}}$ in exactly the same manner as ECOC but use alternative methods to then produce the overall classification.

2.3.1 The Regression PICT

The best way to motivate this procedure is by using an example. Suppose that we have the following coding matrix

$Z =$

Class	$\mathbf{Z}^1$	$\mathbf{Z}^2$	$\mathbf{Z}^3$	$\mathbf{Z}^4$	$\mathbf{Z}^5$
1	0	1	1	0	0
2	0	0	0	1	1
3	1	1	0	0	1
4	1	0	0	0	0

and we assume that at a fixed point in the predictor space the distribution of classes is as follows :

Class	1	2	3	4
q_i	0.4	0.1	0.3	0.2

Now suppose we have a classifier, such as the Bayes Classifier, that can produce perfect probability estimates for super group one for each of the columns. Then we will get the following probability estimates.

Column	1	2	3	4	5
Super Group One Probability	0.5	0.7	0.4	0.1	0.4

Now the question becomes : *Can we use these super group probabilities to reproduce the individual class probabilities and hence derive the k class Bayes Classifier?* The answer to

this question is **yes**, provided we have at least k linearly independent columns in the coding matrix. This is because each column forms an equation with k unknowns, $q_1, \ldots q_k$ i.e.

$$\hat{p}_j = \sum_{i=1}^{k} Z_{ij} q_i \quad j = 1, \ldots, B \tag{2.7}$$

so, provided we have at least k independent equations, we can solve for the unknown variables. In this example we have the following simultaneous system of equations.

$$\begin{array}{rcrcrcl} & & & q_3 & + & q_4 & = & 0.5 \\ q_1 & & + & q_3 & & & = & 0.7 \\ q_1 & & & & & & = & 0.4 \\ & q_2 & & & & & = & 0.1 \\ & q_2 & + & q_3 & & & = & 0.4 \end{array}$$

By solving these equations it is possible to re derive the original k class probabilities.

Least Squares

Now of course in general the Base Classifier will not produce exactly correct probability estimates. The most that one can hope for is a classifier that produces unbiased probability estimates. In other words, a classifier such that

$$\hat{p}_j = \sum_{i=1}^{k} Z_{ij} q_i + \epsilon_j \quad j = 1, \ldots, B \tag{2.8}$$

where $E_{\mathcal{T}} \epsilon_j = 0$. Notice that Model 2.8 is linear in the unknown parameters q_i so a natural approach to estimating the probabilities would be to use least squares just as one would in a linear regression setting. Therefore one would choose q_i to minimize

$$R = \sum_{j=1}^{B} (\hat{p}_j - \sum_{i=1}^{k} Z_{ij} q_i)^2 = \sum_{j=1}^{B} \epsilon_j^2 \tag{2.9}$$

or in matrix terms one would like to minimize

$$\begin{aligned} R &= (\hat{\mathbf{p}} - Z^T \mathbf{q})^T (\hat{\mathbf{p}} - Z^T \mathbf{q}) \\ &= \hat{\mathbf{p}}^T \hat{\mathbf{p}} - 2\hat{\mathbf{p}}^T Z^T \mathbf{q} + \mathbf{q}^T Z Z^T \mathbf{q} \end{aligned}$$

To do this one would differentiate R with respect to $\mathbf{q}$ and set the derivative equal to zero to produce the least squares estimates.

$$\begin{aligned} \frac{\partial R}{\partial \mathbf{q}} &= -2Z\hat{\mathbf{p}} + 2ZZ^T\hat{\mathbf{q}} = 0 \\ \Rightarrow ZZ^T\hat{\mathbf{q}} &= Z\hat{\mathbf{p}} \\ \Rightarrow \hat{\mathbf{q}} &= (ZZ^T)^{-1}Z\hat{\mathbf{p}} \quad \text{provided } ZZ^T \text{ is invertible} \end{aligned}$$

Therefore the Regression PICT consists of the following algorithm :

Regression Algorithm

1. Produce a vector of super group probability estimates, $\hat{\mathbf{p}}$, as with the ECOC PICT.
2. Compute
$$\hat{\mathbf{q}} = (ZZ^T)^{-1}Z\hat{\mathbf{p}}$$
3. Classify to
$$\arg\max_i \hat{q}_i$$

Notice that the estimate,

$$\hat{\mathbf{q}} = (ZZ^T)^{-1}Z\hat{\mathbf{p}}$$

is simply the standard least squares solution except that the transpose of the coding matrix, Z^T, takes the place of the design matrix, X. Section 2.6 details results when this classifier is compared to ECOC as well as standard classifiers.

Theory

The Regression PICT seems to use a more justifiable procedure for combining the two class probabilities to produce a k class classification rule. Theorem 6 shows that it possesses a large theoretical advantage over the ECOC PICT.

Theorem 6 *The Regression PICT is Bayes Consistent for any coding matrix, provided ZZ^T is invertible. In other words if the Base Classifier is producing perfect two class*

probability estimates the Regression PICT will produce perfect k class probability estimates.

In contrast Theorem 1 tells us that the ECOC PICT will only be Bayes Consistent for a limited number of matrices.

Ridging

There is a serious practical limitation with the Regression PICT. The probability estimates can be highly variable when there are a small number of columns, B, relative to the number of classes, k. In fact if $B < k$ then it is not possible to produce a unique estimate for q_i. This is analogous to the situation in least squares regression where the number of data points is small compared to the number of predictor variables.

A common solution to this problem in regression is to use ridged least squares. This involves using an extra term which penalizes large parameter estimates. It works by finding estimates for β which minimize :

$$R = \sum_{i=1}^{n}(Y_i - \sum_j X_{ij}\beta_j)^2 + \lambda \sum_j \beta_j^2$$

In the classification setting this is equivalent to minimizing

$$\begin{aligned} R &= (\hat{\mathbf{p}} - Z^T\mathbf{q})^T(\hat{\mathbf{p}} - Z^T\mathbf{q}) + \lambda \mathbf{q}^T\mathbf{q} \\ &= \hat{\mathbf{p}}^T\hat{\mathbf{p}} - 2\hat{\mathbf{p}}^T Z^T\mathbf{q} + \mathbf{q}^T(\lambda I + ZZ^T)\mathbf{q} \end{aligned}$$

By differentiating with respect to $\mathbf{q}$, as in the ordinary least squares case, one gets the penalized least squares solution

$$\hat{\mathbf{q}} = (\lambda I + ZZ^T)^{-1} Z\hat{\mathbf{p}} \tag{2.10}$$

This reduces variance in the probability estimates at the expense of introducing bias. Of course in the classification setting ones only concern is the argument of the maximum probability estimate so bias is far less of a concern. A systematic bias in all the estimates will not change the argument of the maximum! Therefore ridging the Regression PICT could produce significant improvements.

The Ridged Regression algorithm can be expressed as follows :

Ridged Regression Algorithm

1. Produce a vector of super group probability estimates, $\hat{\mathbf{p}}$, as with the ECOC PICT.
2. Compute
$$\hat{\mathbf{q}} = (\lambda I + ZZ^T)^{-1} Z\hat{\mathbf{p}}$$
3. Classify to
$$\arg\max_i \hat{q}_i$$

To date preliminary results have been extremely promising with some large reductions in error rate achieved. However, further study is required. For example a procedure for choosing λ needs to be devised.

2.3.2 The Centroid PICT

In Section 2.1.4 we saw that it is possible to view the ECOC PICT as a procedure for projecting points into a B dimensional hypercube and then classifying to the class corresponding to the nearest target vertex.

Figure 2.5 illustrates a potential problem with the ECOC PICT that becomes apparent when this geometric view is used. Here we have the same situation as with Figure 2.1 except that now the training points associated with Class 3 have been systematically shifted towards those of Class 2. This may seem strange but in fact there is no guarantee that the transformed variables for a class need to be close to the vertex for that class. For example one of the Base Classifiers could be giving biased probability estimates. Never the less there is a clear separation between clusters of points so it should still be possible to form a good classification rule.

Unfortunately the ECOC PICT will not exploit this separation and will classify most

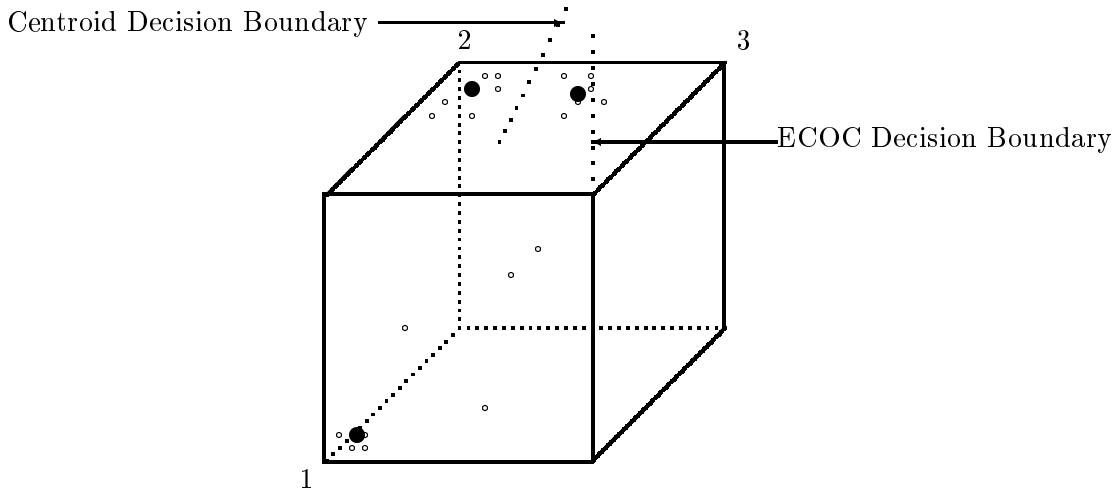

Figure 2.5: A multidimensional cube. The open circles are training points. The solid circles represent class centroids. A new point will be classified to the class corresponding to the closest vertex of the cube in L1 distance.

points from Class 3 to Class 2. This is because it will classify to the closest target vertex so the decision boundary will cut through the middle of the cube as indicated in the figure. Since most points in Class 3 seem to fall to the left of this boundary they will be classified as 2's.

It seems that a better *target* to aim for may be the center of the training data rather than an *arbitrary* point such as a vertex. This is the motivation behind the Centroid PICT. It performs a change of variables just as with the ECOC PICT but classifies to the closest training data centroid, in L2 distance. In other words it allows the data to determine the representation coding for each class. In Figure 2.5 the solid circles represent the training data centroids. The Centroid PICT will classify to the closest of these points. In this example there is a large change in the boundary between Classes 2 and 3 and the Class 3 points are now correctly classified.

Centroid Algorithm

1. For each training data point, produce a vector of super group probability estimates, $\hat{\mathbf{p}}$, as with the ECOC PICT.

2. For each class calculate the *class centroid* by taking the mean or median for all training data from that class. Call these centroids :

$$\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_k$$

3. Classify to

$$\arg\min_i ||\hat{\mathbf{p}} - \mathbf{c}_i||$$

Section 2.6 details results when this classifier is compared to ECOC as well as standard classifiers.

2.4 The Substitution PICT

While the ECOC classifier has a similar feel to a Majority Vote Classifier, it is not possible to formulate it as specified in either (1.2) or (1.3). However, in this section we introduce a Semi MaVL which we call the Substitution PICT. It is possible to show that, under certain conditions, the ECOC and Substitution PICTs are asymptotically (in B) identical and that in this sense the ECOC classifier is asymptotically a MaVL.

2.4.1 The Substitution PICT Algorithm

The Substitution PICT algorithm is as follows :

Substitution Algorithm

- Produce a random binary coding matrix as with the ECOC PICT.
- Use the first column of the coding matrix ($\mathbf{Z}^1$) to create two *super* groups by assigning all classes with a one in the corresponding element of $\mathbf{Z}^1$ to super group one and all other classes to super group zero.
- Train a tree classifier on the new two class problem and repeat the process for each of the B columns. Each tree will form a partitioning of the predictor space.
- Now retain the partitioning of the predictor space that each tree has produced. Feed back into the trees the original k class training data. Use the training data to form probability estimates, just as one would do for any tree classifier. The only difference here is the rule that has been used to create the partitioning.
- To a new test point apply each of the B classifiers. The jth classifier will produce a k class probability estimate, p_{ij}, which is the estimated probability the test point comes from the ith class.
- To classify the point calculate

$$p_i^S = \frac{1}{B}\sum_{j=1}^{B} p_{ij} \tag{2.11}$$

  and classify to $\arg\max_i p_i^S$

In summary, the Substitution PICT uses the coding matrix to form many different partitionings of the predictor space. Then, for each partitioning, it forms k class probability estimates by examining the proportions of each class, among the training data, that fall in the same region as the test point. The probability estimates are then combined by averaging over all the trees for each class. The final classification is to the maximum probability estimate.

2.4.2 Asymptotic Equivalence of ECOC and Substitution PICTs

Theorem 7 shows that under certain conditions the ECOC PICT can be thought of as an approximation to the Substitution PICT.

Theorem 7 *Suppose that p_{ij} is independent from $\mathbf{Z}^j$ (the jth column of Z), for all i and j. In other words the distribution of p_{ij} conditional on $\mathbf{Z}^j$ is identical to the unconditional distribution. Then*

$$E_Z[p_i^S \mid \mathcal{T}] = E_Z[\bar{D}_i \mid \mathcal{T}] = \mu_i$$

Therefore as B approaches infinity the ECOC PICT and Substitution PICT will converge for any given training set; i.e. they will give identical classification rules.

The theorem basicly states that under suitable conditions both p_i^S and $\bar{D}_i$ are unbiased estimates of μ_i and both will converge to μ_i almost surely.

It is unlikely the assumption of independence is realistic. However, empirically it is well known that trees are unstable and a small change in the training data can cause a large change in the structure of the tree so it may be reasonable to suppose that the correlation between p_{ij} and $\mathbf{Z}^j$ is low.

To test this empirically we ran the ECOC and Substitution PICTs on a simulated data set. The data set was composed of 26 classes. Each class was distributed as a bivariate normal with identity covariance matrix and means uniformly distributed in the range $[-5, 5]^2$. Each training data set consisted of 10 observations from each class. Figure 2.6 shows a plot of the estimated probabilities, for each method, for each of the 26 classes and 1040 test data points averaged over 10 training data sets. The probability estimates are calculated based on a matrix with 100 columns (i.e. $B = 100$). Only points where the true posterior probability is greater than 0.01 have been plotted since classes with insignificant probabilities are unlikely to affect the classification. If the two methods were producing identical estimates we would expect the data points to lie on the dotted 45 degree line. Clearly this is not the case. The Substitution PICT is systematically shrinking the probability estimates. However there is a very clear linear relationship ($R^2 \approx 95\%$) and since we are only interested in the

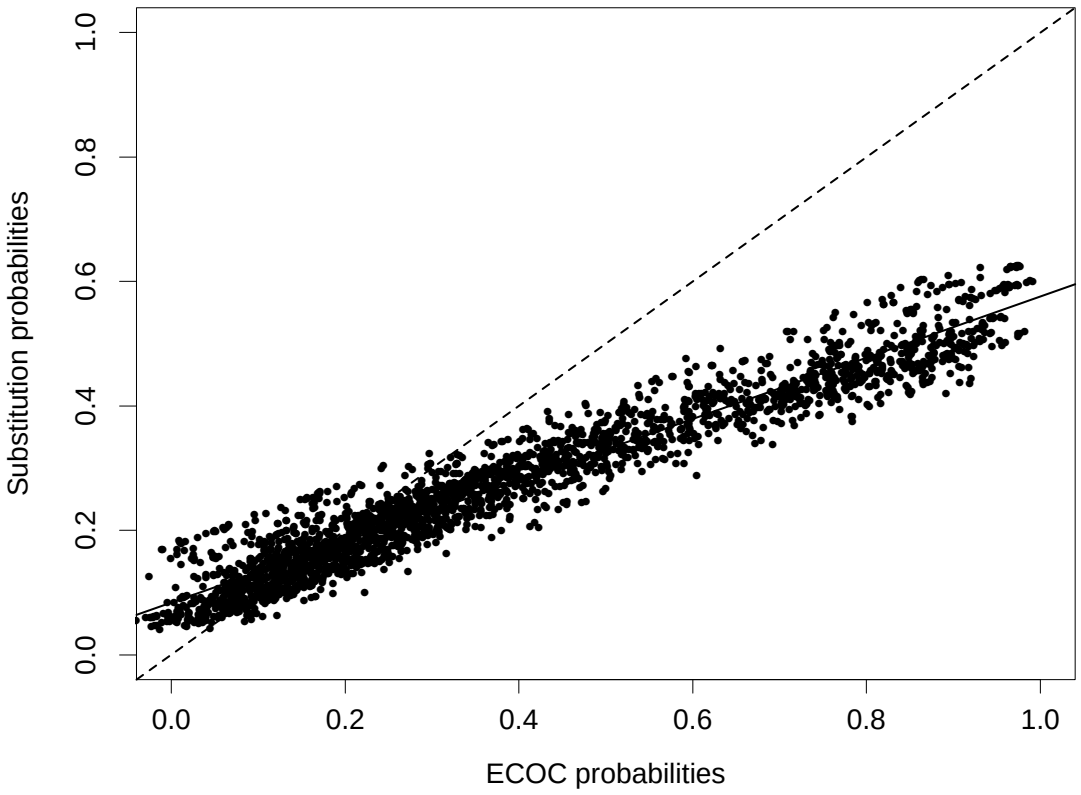

Figure 2.6: Probability estimates from both the ECOC and Substitution PICTs

arg max for each test point we might expect similar classifications. This is indeed the case. Fewer than 4% of points are correctly classified by one method but not the other.

2.4.3 The Substitution PICT for Low Values of B

The previous section provides theoretical as well as empirical motivation for the approximate equivalence of the ECOC and Substitution PICTs as B becomes large. Section 2.6 provides further illustration of this phenomenon. However, it is also apparent from the results in that section that the Substitution PICT provides vastly superior results for low values of B. This is fairly easy to explain.

Both p_i^S and $\bar{D}_i$ are averages over random variables. p_i^S is an average over p_{ij} and $\bar{D}_i$ is an average over $D_{ij} = 1 - 2|\hat{p}_i - Z_{ij}|$. Now under the assumptions in Theorem 7

$$\begin{aligned} Var_{\mathcal{T},Z}(p_i^S) &= Var_{\mathcal{T}}E_Z(p_i^S|\mathcal{T}) + E_{\mathcal{T}}Var_Z(p_i^S|\mathcal{T}) \\ &= Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}E_{\mathcal{T}}Var_Z(p_{i1}|\mathcal{T}) \\ &= Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}[Var_{\mathcal{T},Z}(p_{i1}) - Var_{\mathcal{T}}(\mu_i)] \\ &= (1 - \frac{1}{B})Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}Var_{\mathcal{T},Z}(p_{i1}) \end{aligned}$$

and similarly

$$\begin{aligned} Var_{\mathcal{T},Z}(\bar{D}_i) &= Var_{\mathcal{T}}E_Z(\bar{D}_i|\mathcal{T}) + E_{\mathcal{T}}Var_Z(\bar{D}_i|\mathcal{T}) \\ &= Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}E_{\mathcal{T}}Var_Z(D_{i1}|\mathcal{T}) \\ &= Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}[Var_{\mathcal{T},Z}(D_{i1}) - Var_{\mathcal{T}}(\mu_i)] \\ &= (1 - \frac{1}{B})Var_{\mathcal{T}}(\mu_i) + \frac{1}{B}Var_{\mathcal{T},Z}(D_{i1}) \end{aligned}$$

So we see that

$$Var_{\mathcal{T},Z}(\bar{D}_i) = Var_{\mathcal{T},Z}(p_i^S) + \frac{\epsilon}{B}$$

where $\epsilon = Var_{\mathcal{T},Z}(D_{i1}) - Var_{\mathcal{T},Z}(p_{i1})$.

In general the variance of p_{ij} is lower than that of D_{ij} so ϵ will be positive and the variance of $\bar{D}_i$ will be larger than that of p_i^S. As B becomes large the difference in variance will become negligible but for smaller values one would expect the increased variance to cause a deterioration in the classification.

2.5 The Bagging and Boosting PICTs

Bagging and Boosting are two of the most well known and successful examples of MaVLs. They both work by iteratively resampling the training data, producing a new classifier based on each resampled data set and then combining all the classifiers together using a majority vote procedure.

2.5.1 The Bagging PICT

Breiman, 1996a suggested the Bagging PICT (Bootstrap Aggregation). The algorithm consists of the following parts.

Bagging Algorithm

1. Resample observations from your training data set, with replacement, to produce a Bootstrapped data set.
2. Train a Base Classifier on this bootstrapped training data. Typically a tree classifier is used but in principle any classifier will work.
3. Repeat steps 1 and 2 B times where B is a pre chosen number. Typically convergence is obtained very fast so fewer than 50 iterations may be required.
4. Combine all B classifiers together into a single rule by taking a majority vote.

Breiman, 1996a provides motivation for the Bagging Algorithm. The Bagging Algorithm can also be applied to regression as well as classification problems.

2.5.2 The Boosting PICT

Boosting can be thought of as an extension of Bagging where the resampling weights do not remain constant but adapt to the data set. There are several different algorithms to implement the Boosting procedure. The most common one is known as AdaBoost (Freund and Schapire, 1996). It is possible to use AdaBoost in either a resampling or deterministic mode. We will describe here the resampling version.

AdaBoost Algorithm

Start with a set of n training observations

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

1. Set $w_1(x_i) = 1/n$ for $i = 1, \ldots, n$. Let $\mathbf{w}_1 = (w_1(x_1), \cdots, w_1(x_n))$.

2. At the t th step resample observations from the training data set, with replacement, according to the weighting induced by $\mathbf{w}_t$.

3. Train a Base Classifier on the resampled training data. Very simple classifiers such as a stump (two terminal node tree) classifier can be used at this step. Call this classifier C_t.

4. Let
$$\beta_t = \frac{\epsilon_{\mathbf{w}_t}(C_t)}{1 - \epsilon_{\mathbf{w}_t}(C_t)}$$
where $\epsilon_{\mathbf{w}_t}(C_t) = \sum_{i=1}^{n} w_t(x_i) I(C_t(x_i) \neq y_i)$. In other words $\epsilon_{\mathbf{w}_t}(C_t)$ is the error rate on the original training data, weighted by $\mathbf{w}_t$. Now let
$$w_{t+1}(x_i) = \begin{cases} \frac{1}{Z_t} w_t(x_i) & \text{if} \quad C_t(x_i) \neq y_i \\ \frac{1}{Z_t} \beta_t w_t(x_i) & \text{if} \quad C_t(x_i) = y_i \end{cases}$$
where Z_t is a normalizing constant.

5. Return to Step 2 and repeat the process B times.

6. Classify to $\arg\max_i \sum_{t=1}^{B} \alpha_t I(C_t(x) = i)$ where
$$\alpha_t = \frac{\log(1/\beta_t)}{\sum_s \log(1/\beta_s)}$$

The difference between the resampling and deterministic algorithms is that instead of resampling at Step 2 we simply train the Base Classifier at Step 3 on a weighted version of the original training data.

2.6 Experimental Comparisons

To provide a comparison of the different classifiers mentioned in this chapter we present experimental results on three different data sets. The first data set is simulated. It consists of two classes and the predictor space is in two dimensions. Figure 2.7 gives an illustration of one realization from the distribution. We used this data set to provide a comparison between random and deterministic weights in the AdaBoost classifier (see Section 2.6.1). The second is the Letter data set from the Irvine Repository of Machine Learning, and the third is the Vowel data set from the same location. The Letter data set is described in Section 2.2.2. For these experiments a test set of size 520 was used. The Vowel data set consists of 990 observations in 10 different dimensions. There are 11 different classes. The observations are split into a training set (528) and a test set (462).

On each of the last two data sets nine different classifiers were compared. They were
The ECOC PICT
The Substitution PICT
The Regression PICT
The Centroid PICT
Bagging
Boosting (Standard)
Boosting (Adapted)
Tree Classifier
1 Nearest Neighbour

The Adapted Boosting classifier works in the same way as standard boosting except that instead of calculating the error rate using

$$\epsilon_{\mathbf{w}_t}(C_t) = \sum_{i=1}^{n} w_t(x_i) I(C_t(x_i) \neq y_i)$$

one uses

$$\epsilon'_{\mathbf{w}_t}(C_t) = \text{Error rate on resampled training data}$$

The tree and 1 nearest neighbour classifiers provide a base line comparison to the MaVLs. Each of the first 7 classifiers uses a tree generated in Splus as the Base Classifier. See Section

1.3.1 for a general description of tree based classifiers. Splus uses deviance as its splitting criterion. The split producing the maximum reduction in deviance is chosen at each step in the tree growing process. The process terminates when either a pre specified number of terminal nodes are achieved or the number of cases in each leaf is small (by default $n_i < 5$ in Splus).

As noted in Section 2.5.2 there are two different ways to implement the Boosting algorithm, i.e. using random or deterministic weighting. Therefore, prior to a detailed comparison of the data sets it was desirable to determine if there was any significant difference between the two methods. Section 2.6.1 provides results from this comparison while Sections 2.6.2 and 2.6.3 provide results from the experiments on the other two data sets.

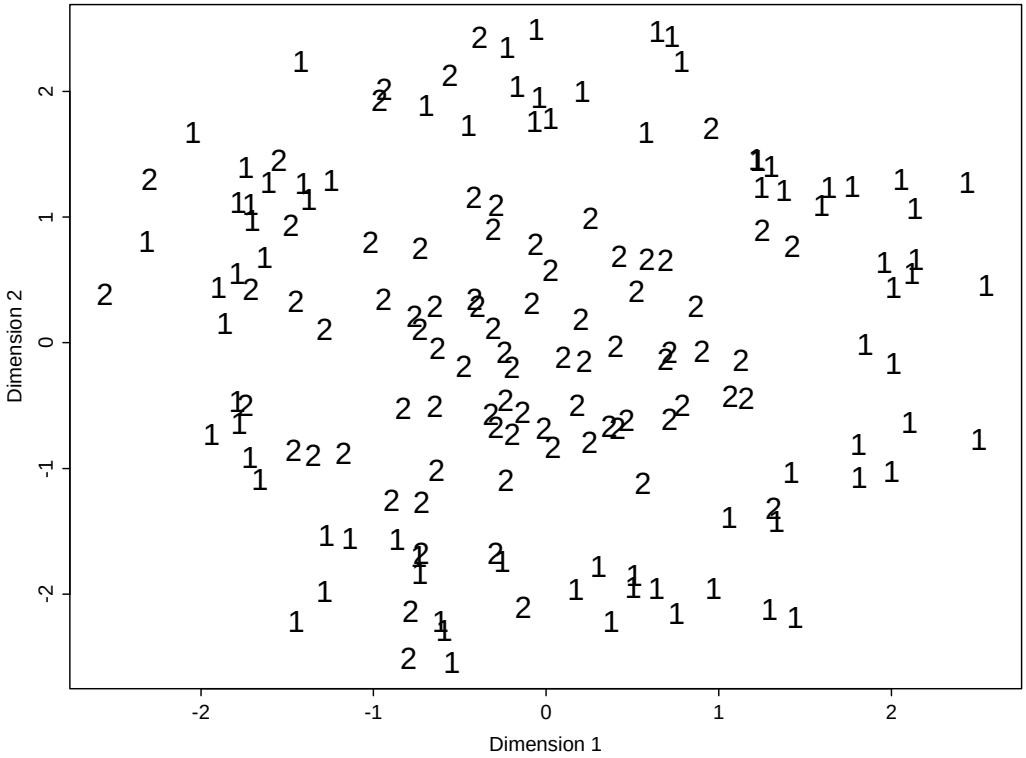

Figure 2.7: A single realization from the simulated distribution used in Section 2.6.1

B	2	10	20	40	60	80	100	120
Boosting (Det)	42.4	31.0	28.9	28.1	25.7	27.1	26.3	26.5
Boosting (Rand)	42.4	32.6	29.5	28.7	26.8	26.1	25.6	25.7
Tree	41.5	41.5	41.5	41.5	41.5	41.5	41.5	41.5
1NN	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2

Table 2.3: **Concentric Two Class data set** test error rates averaged over 10 training sets each of size 20 observations per class. 2 Terminal Nodes. Average standard error is 1.5%.

2.6.1 Random vs Deterministic Weightings

To perform a comparison between random and deterministic weightings we created a simulated 2 class distribution. Figure 2.7 provides an illustration. The data consists of a class in the center with a second class around the outside. We will call this the *Concentric Two Class* data set. Experiments were conducted with differing tree depths, i.e. controlling the complexity of the Base Classifier, and different training sample sizes.

Tables 2.3, 2.4 and 2.5 provide results for training sample sizes of 20 observations per class, with increasing degrees of tree complexity. While Tables 2.6, 2.7, 2.8 and 2.9 provide results for training sample sizes of 100 observations per class. Each table gives error rates for boosting with random and deterministic weighting along with a simple tree classifier and 1 nearest neighbour.

For the smaller training sample size it appears that a random weighting provides uniformly better error rates than the deterministic scheme. It seems that the difference increases with the tree complexity. For very simple trees (stumps) with only 2 terminal nodes there is very little difference but with more complex trees the difference is much larger. For the larger training sample size there is no clear trend with all four methods getting very similar error rates. Since a deterministic weighting scheme seemed never to outperform a random weighting, the experiments in the following sections use random weightings.

B	2	10	20	40	60	80	100	120
Boosting (Det)	32.4	30.1	31.7	31.8	31.7	31.8	31.8	32.1
Boosting (Rand)	35.0	27.9	26.9	27.2	26.8	26.9	26.7	26.9
Tree	34.1	34.1	34.1	34.1	34.1	34.1	34.1	34.1
1NN	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1

Table 2.4: **Concentric Two Class data set** test error rates averaged over 10 training sets each of size 20 observations per class. 5 Terminal Nodes. Average standard error is 1.2%.

B	2	10	20	40	60	80	100	120
Boosting (Det)	41.6	37.5	36.3	37.3	37.6	37.7	37.7	37.7
Boosting (Rand)	37.7	35.3	34.2	32.2	32.7	31.8	31.8	31.1
Tree	37.3	37.3	37.3	37.3	37.3	37.3	37.3	37.3
1NN	28.4	28.4	28.4	28.4	28.4	28.4	28.4	28.4

Table 2.5: **Concentric Two Class data set** test error rates averaged over 5 training sets each of size 20 observations per class. Default tree settings (approximately 7 terminal nodes). Average standard error is 1.7%.

B	2	10	20	40	60	80	100	120
Boosting (Det)	41.4	28.7	22.9	20.6	19.9	20.0	20.0	20.2
Boosting (Rand)	41.6	29.1	24.9	21.6	20.7	19.2	19.5	20.2
Tree	41.2	41.2	41.2	41.2	41.2	41.2	41.2	41.2
1NN	21.4	21.4	21.4	21.4	21.4	21.4	21.4	21.4

Table 2.6: **Concentric Two Class data set** test error rates averaged over 10 training sets each of size 100 observations per class. 2 Terminal Nodes. Average standard error is 0.96%.

B	2	10	20	40	60	80	100	120
Boosting (Det)	22.3	20.2	21.7	22.3	22.4	22.3	22.4	22.4
Boosting (Rand)	26.7	20.1	19.5	20.0	20.4	20.6	21.3	21.5
Tree	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4
1NN	21.9	21.9	21.9	21.9	21.9	21.9	21.9	21.9

Table 2.7: **Concentric Two Class data set** test error rates averaged over 5 training sets each of size 100 observations per class. 5 Terminal Nodes. Average standard error is 1.2%.

B	2	10	20	40	60	80	100	120
Boosting (Det)	20.8	21.3	21.0	21.9	22.1	22.3	22.1	22.1
Boosting (Rand)	23.9	20.2	20.8	21.3	21.6	21.6	21.8	21.6
Tree	20.8	20.8	20.8	20.8	20.8	20.8	20.8	20.8
1NN	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4

Table 2.8: **Concentric Two Class data set** test error rates for 2 class data averaged over 5 training sets each of size 100 observations per class. 10 Terminal Nodes. Average standard error is 0.69%.

B	2	10	20	40	60	80	100	120
Boosting (Det)	23.1	22.3	21.8	22.2	22.1	22.3	22.5	22.3
Boosting (Rand)	25.6	22.6	22.8	22.1	22.5	22.8	22.4	22.3
Tree	23.0	23.0	23.0	23.0	23.0	23.0	23.0	23.0
1NN	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4

Table 2.9: **Concentric Two Class data set** test error rates averaged over 5 training sets each of size 100 observations per class. 20 Terminal Nodes. Average standard error is 0.91%.

2.6.2 Letter Data Set

Experiments were carried out on this data set using two different training sample sizes, 10 observations per class and 50 observations per class. This was an attempt to examine the various classifiers under hard (10 per class) and easier (50 per class) training situations. We also examined the effect of different depths in the base tree classifier that was used for each PICT. Increasing the tree depth has the effect of producing a more highly trained classifier. Tables 2.10, 2.12, and 2.14 summarize the results for the various combinations of training sample size and tree depth. Tables 2.11, 2.13, and 2.15 provide the same information but all relative to 1 nearest neighbour. So for example 1 would mean it performed as well as nearest neighbours and 0.5 would mean it had half the error rate. Figures 2.8, 2.9 and 2.10 provide graphical representations of the error rates in Tables 2.10, 2.12, and 2.14

The ECOC, Regression, Centroid and (to a lesser extent) the Substitution PICTs produced encouraging results for all combinations of sample size and tree depth. They all produced significantly superior results to both standard tree and nearest neighbour classifiers. There was some improvement when the base tree classifier was forced to grow deeper trees (it seems that the default settings are under-training on the data) but it was not dramatic. They seemed to be relatively insensitive to the complexity of the Base Classifier.

On the other hand the Boosting PICTs seemed to be far more dependent on the complexity of the Base Classifier. With more training data and shallower base trees the Boosting PICTs did not perform very well. The adjusted method gave comparable error rates to that of nearest neighbours but the standard method was far worse. However, when the base tree classifier was forced to grow deeper trees they both improved dramatically (especially the standard method) and gave results that were as good as any of the other classifiers. For the harder problem, where the sample size was small, the adjusted method gave good results but the standard method performed poorly.

Bagging performed poorly on this data set relative to the other PICTs. It was also very dependent on the Base Classifier. It performed very poorly on the problem with large sample size and shallow trees. As with the Boosting it improved dramatically with deeper trees and performed satisfactorily on the harder problem with a smaller sample size.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	89.3	64.3	41.4	29.6	23.4	20.8	19.5	19.0	18.6	17.9	17.6
Regression	100.0	100.0	100.0	100.0	28.5	22.2	20.3	19.0	18.4	18.0	17.5
Centroid	86.7	60.1	39.6	28.5	22.5	20.7	19.5	19.2	18.9	18.3	18.1
Substitution	48.4	34.0	27.0	23.5	21.2	20.6	20.3	20.1	20.1	19.8	19.5
Bagging	41.9	36.5	34.3	33.1	32.1	31.8	31.7	31.7	31.4	31.2	31.4
Boosting (St)	48.6	41.3	37.9	37.2	37.3	36.8	37.8	37.4	37.2	36.7	37.1
Boosting (Ad)	47.9	36.5	30.8	26.6	23.5	22.0	21.2	20.8	20.6	20.2	20.1
1NN	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9
Tree	44.8	44.8	44.8	44.8	44.8	44.8	44.8	44.8	44.8	44.8	44.8

Table 2.10: **Letter data set** test error rates averaged over 20 training sets each of size 50 observations per class. Default tree settings (approximately 60 terminal nodes). Average standard error is 0.43%.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	4.49	3.23	2.08	1.49	1.18	1.04	0.98	0.96	0.93	0.90	0.89
Regression	5.03	5.03	5.03	5.03	1.43	1.12	1.02	0.96	0.93	0.90	0.88
Centroid	4.36	3.02	1.99	1.44	1.13	1.04	0.98	0.97	0.95	0.92	0.91
Substitution	2.43	1.71	1.36	1.18	1.07	1.04	1.02	1.01	1.01	0.99	0.98
Bagging	2.10	1.84	1.72	1.66	1.61	1.60	1.60	1.59	1.58	1.57	1.58
Boosting (St)	2.44	2.08	1.90	1.87	1.87	1.85	1.90	1.88	1.87	1.84	1.87
Boosting (Ad)	2.41	1.83	1.55	1.34	1.18	1.11	1.06	1.04	1.03	1.02	1.01
Tree	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25

Table 2.11: **Letter data set** test error rates relative to 1NN averaged over 20 training sets each of size 50 observations per class. Default tree settings (approximately 60 terminal nodes). Average standard error is 0.022.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	89.7	65.2	41.8	29.2	22.9	20.2	19.6	18.5	18.1	17.4	17.2
Regression	100.0	100.0	100.0	100.0	27.5	22.0	19.5	19.0	18.0	17.2	17.1
Centroid	87.0	61.2	40.4	28.5	22.6	20.2	19.5	18.9	18.3	17.6	17.3
Substitution	48.9	33.3	26.1	22.8	21.0	20.4	19.8	19.6	19.4	19.2	19.2
Bagging	36.9	30.6	27.4	26.0	24.8	24.1	24.0	24.0	24.0	23.8	23.9
Boosting (St)	42.6	29.8	25.0	21.6	19.3	18.9	18.3	18.1	18.3	17.9	17.9
Boosting (Ad)	43.4	32.2	25.7	21.7	19.6	18.4	18.7	18.2	17.9	17.5	17.6
1NN	21.1	21.1	21.1	21.1	21.1	21.1	21.1	21.1	21.1	21.1	21.1
Tree	38.7	38.7	38.7	38.7	38.7	38.7	38.7	38.7	38.7	38.7	38.7

Table 2.12: **Letter data set** test error rates averaged over 10 training sets each of size 50 observations per class. The Base Classifier has been forced to grow deeper trees (approximately 160 terminal nodes). Average standard error is 0.60%.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	4.25	3.09	1.98	1.38	1.09	0.96	0.93	0.88	0.86	0.82	0.82
Regression	4.73	4.73	4.73	4.73	1.30	1.04	0.92	0.90	0.85	0.81	0.81
Centroid	4.12	2.90	1.91	1.35	1.07	0.95	0.92	0.90	0.87	0.83	0.82
Substitution	2.32	1.58	1.24	1.08	0.99	0.97	0.94	0.93	0.92	0.91	0.91
Bagging	1.75	1.45	1.30	1.23	1.17	1.14	1.14	1.14	1.13	1.13	1.13
Boosting (St)	2.02	1.41	1.18	1.02	0.91	0.89	0.87	0.85	0.86	0.85	0.85
Boosting (Ad)	2.06	1.52	1.22	1.03	0.93	0.87	0.88	0.86	0.85	0.83	0.83
Tree	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83

Table 2.13: **Letter data set** test error rates relative to 1NN averaged over 10 training sets each of size 50 observations per class. The Base Classifier has been forced to grow deeper trees (approximately 160 terminal nodes). Average standard error is 0.028.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	91.7	78.5	64.0	51.7	42.5	39.3	38.1	36.8	36.2	34.9	34.3
Regression	87.3	74.0	60.9	49.0	40.7	38.8	37.5	36.1	35.7	34.7	34.0
Centroid	100.0	100.0	100.0	100.0	50.0	41.5	38.6	36.9	35.8	34.9	34.3
Substitution	66.5	53.7	46.3	41.9	38.1	37.3	37.0	36.9	37.0	36.9	36.8
Bagging	58.6	50.4	45.5	42.7	41.0	40.7	40.1	40.1	39.9	40.0	39.9
Boosting (St)	63.5	60.7	56.4	55.2	54.3	53.5	52.9	53.5	52.8	50.8	50.4
Boosting (Ad)	66.4	54.0	46.3	41.8	38.2	37.1	36.9	36.3	36.2	36.3	36.0
Tree	58.0	58.0	58.0	58.0	58.0	58.0	58.0	58.0	58.0	58.0	58.0
1NN	43.6	43.6	43.6	43.6	43.6	43.6	43.6	43.6	43.6	43.6	43.6

Table 2.14: **Letter data set** test error rates averaged over 20 training sets each of size 10 observations per class. Default tree settings (approximately 35 terminal nodes). Average standard error is 0.60%.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	2.10	1.80	1.47	1.19	0.98	0.90	0.87	0.84	0.83	0.80	0.79
Regression	2.29	2.29	2.29	2.29	1.15	0.95	0.88	0.85	0.82	0.80	0.79
Centroid	2.00	1.70	1.40	1.12	0.93	0.89	0.86	0.83	0.82	0.80	0.78
Substitution	1.52	1.23	1.06	0.96	0.87	0.86	0.85	0.85	0.85	0.85	0.84
Bagging	1.34	1.16	1.04	0.98	0.94	0.93	0.92	0.92	0.92	0.92	0.91
Boosting (St)	1.46	1.39	1.29	1.27	1.25	1.23	1.21	1.23	1.21	1.16	1.15
Boosting (Ad)	1.52	1.24	1.06	0.96	0.88	0.85	0.85	0.83	0.83	0.83	0.83
Tree	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33

Table 2.15: **Letter data set** test error rates relative to 1NN averaged over 20 training sets each of size 10 observations per class. Default tree settings (approximately 35 terminal nodes). Average standard error is 0.014.

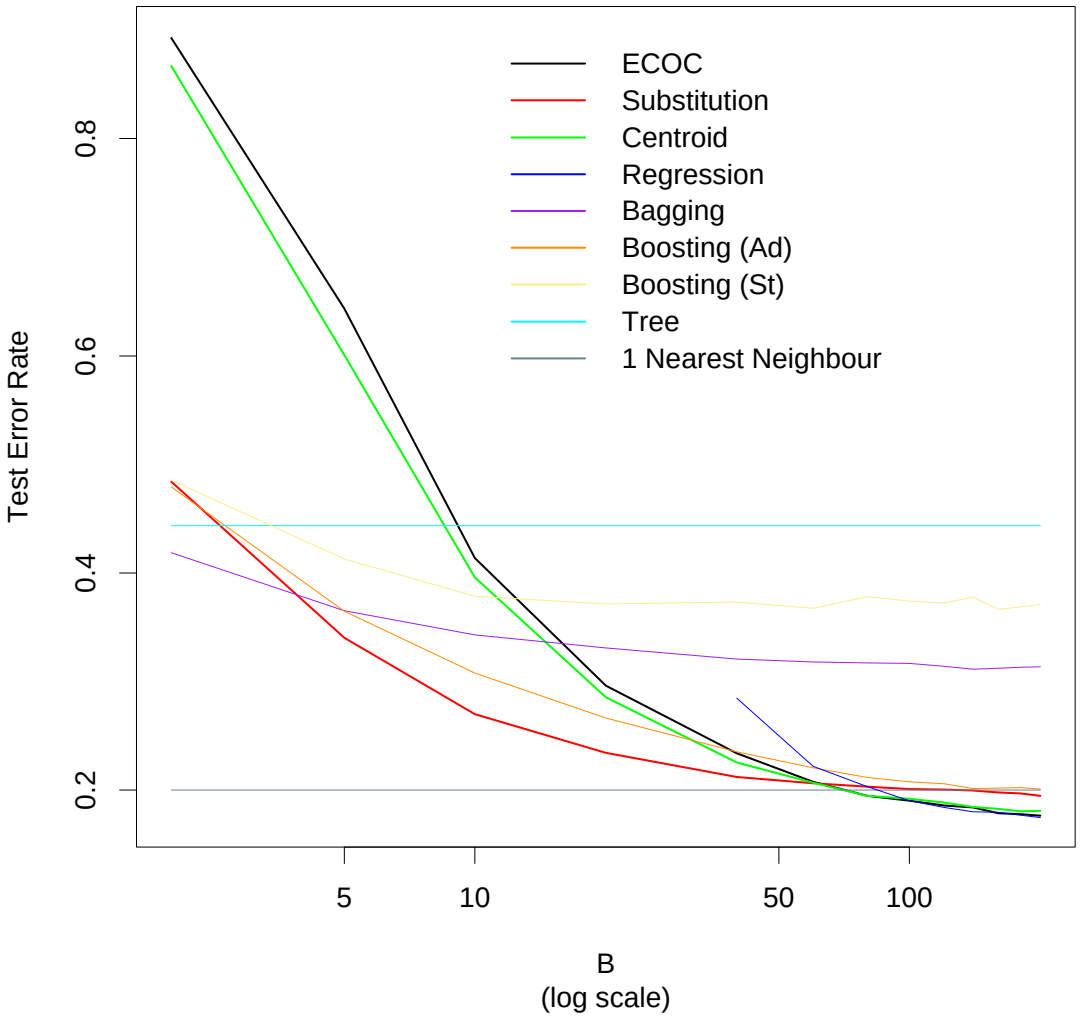

Figure 2.8: A plot of the results from Table 2.10 (Letter data set with 50 observations per class).

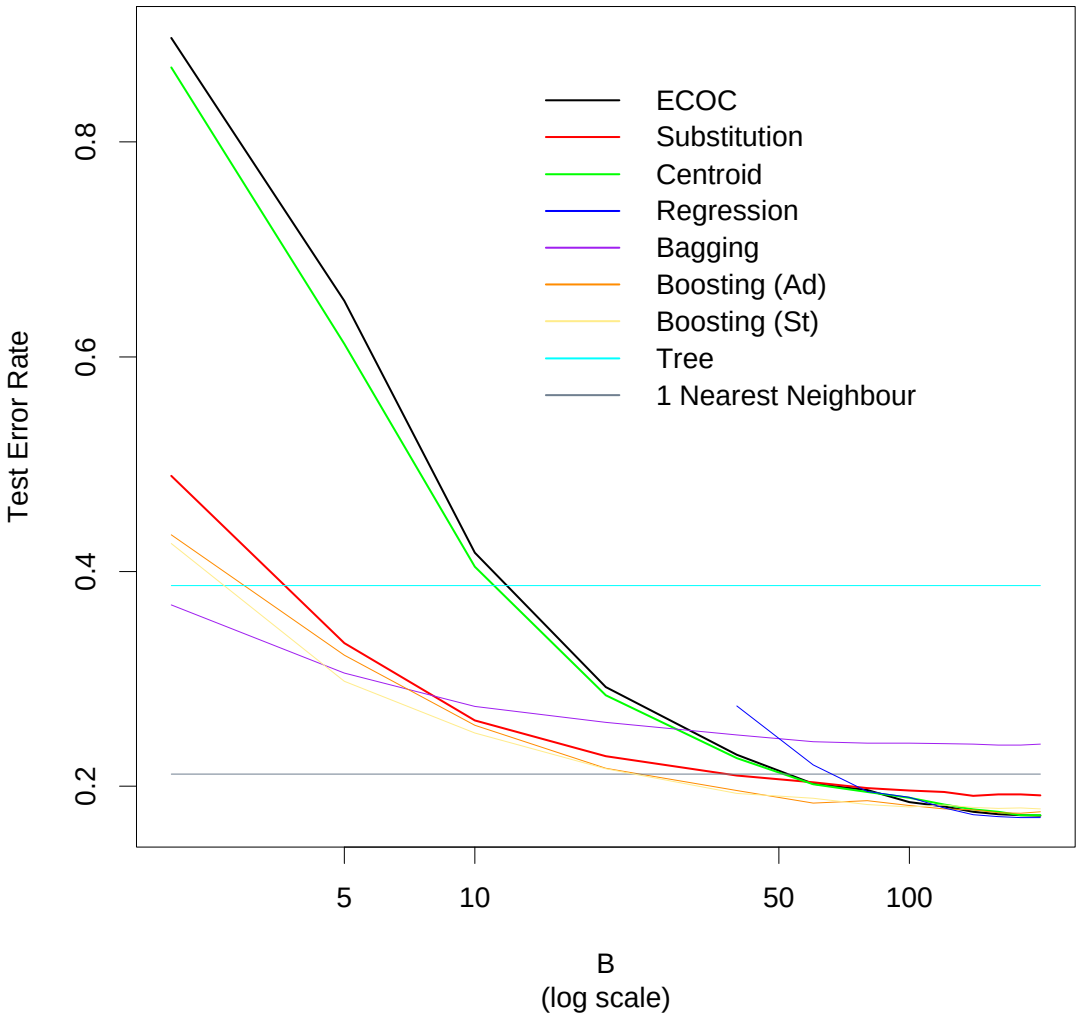

Figure 2.9: A plot of the results from Table 2.12 (Letter data set with 50 observations per class and deep trees).

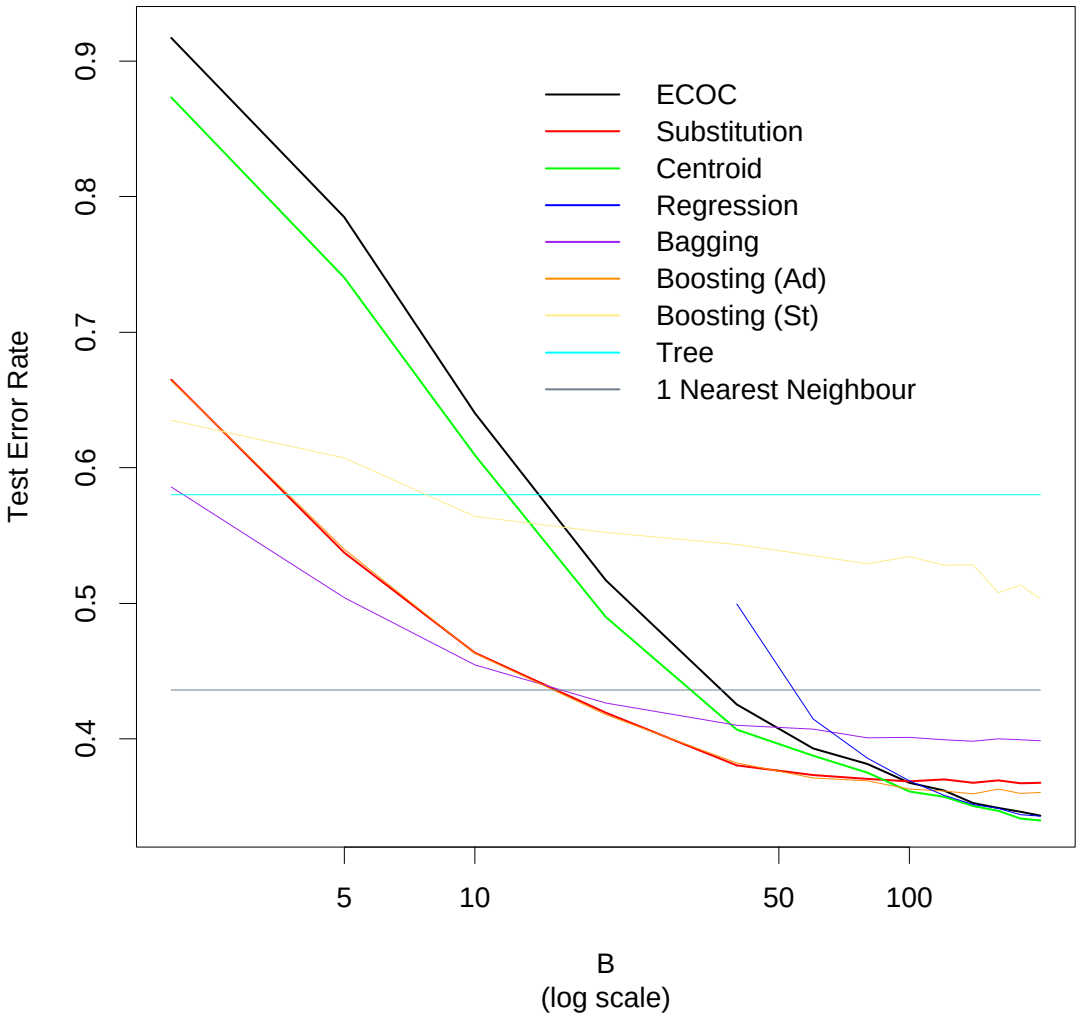

Figure 2.10: A plot of the results from Table 2.14 (Letter data set with 10 observations per class).

2.6.3 Vowel Data Set

This is a much more difficult data set and the various classifiers had a correspondingly more difficult time. The results are presented in Tables 2.16 through 2.19 and Figures 2.11 and 2.12. It is clear that nearest neighbours is a very effective classifier with this data. Of all the classifiers tested, the Substitution PICT was the only one that matched (and in one case exceeded) the performance of 1 nearest neighbour. With the easier problem, with 40 observations per class, nearest neighbours and the Substitution PICT were almost identical but with the more difficult problem where there were only 20 observations per class the Substitution PICT was noticeably better. None of the other classifiers could match the performance of nearest neighbours. However, it should be noted that they all gave significant improvements over using a tree classifier by itself.

2.6.4 Summary

The ECOC, Substitution, Centroid, Regression, Bagging, Boosting (St) and Boosting (Ad) PICTs all transformed a base tree classifier in some way. For all the data sets considered the PICTs gave large reductions in the error rate over using the tree classifier by itself. However, no one classifier was consistently better than any other for all data sets.

For the Vowel data set the Substitution PICT was significantly better than any of the other PICTs and marginally superior to 1 nearest neighbour. However, for the Letter data set it was slightly worse than the Centroid and ECOC PICTs. For the Letter data set the ECOC, Centroid and Regression PICTs performed best. This is consistent with results from other data sets we have considered, where no one classifier uniformly dominated.

The Boosting PICTs seemed to be much more sensitive to the degree of training of the tree classifier. The error rates for these two classifiers improved dramatically on the Letter data set when the trees were forced to include more terminal nodes. This suggests that Boosting may be better suited to less automated procedures where more effort can be devoted to fine tuning the parameters. The Adapted Boosting procedure appeared to provide a more robust classifier. This adaption to standard Boosting deserves further study. While the Bagging PICT gave consistent improvements over the tree classifier, in general it did not perform nearly as well as the other PICTs.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	80.1	69.1	59.0	52.8	48.3	47.6	48.0	47.6	46.9	46.1	46.1
Regression	100.0	100.0	100.0	55.1	48.6	47.4	47.5	47.5	46.7	45.9	45.9
Centroid	78.4	68.1	57.6	52.5	48.9	47.8	48.2	47.6	47.0	46.5	46.1
Substitution	61.7	53.2	48.6	45.5	43.0	43.3	43.7	43.5	43.0	43.1	43.1
Bagging	57.9	55.8	52.7	52.4	53.1	52.7	52.4	51.9	52.3	52.6	52.5
Boosting (Ad)	65.3	55.5	54.5	51.9	50.9	50.0	49.8	50.0	49.4	49.5	49.1
1NN	43.3	43.3	43.3	43.3	43.3	43.3	43.3	43.3	43.3	43.3	43.3
Tree	61.5	61.5	61.5	61.5	61.5	61.5	61.5	61.5	61.5	61.5	61.5

Table 2.16: **Vowel data set** test error rates averaged over 10 training sets each of size 40 observations per class. Default tree settings (approximately 32 terminal nodes). Average standard error is 0.97%.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	1.85	1.60	1.36	1.22	1.12	1.10	1.11	1.10	1.08	1.07	1.07
Regression	2.31	2.31	2.31	1.27	1.12	1.10	1.10	1.10	1.08	1.06	1.06
Centroid	1.81	1.57	1.33	1.21	1.13	1.10	1.11	1.10	1.09	1.07	1.07
Substitution	1.43	1.23	1.12	1.05	0.99	1.00	1.01	1.00	0.99	1.00	1.00
Bagging	1.34	1.29	1.22	1.21	1.23	1.22	1.21	1.20	1.21	1.22	1.21
Boosting (Ad)	1.51	1.28	1.26	1.20	1.18	1.16	1.15	1.16	1.14	1.14	1.14
Tree	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42

Table 2.17: **Vowel data set** test error rates relative to 1NN averaged over 10 training sets each of size 40 observations per class. Default tree settings (approximately 32 terminal nodes). Average standard error is 0.022.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	85.0	75.9	69.7	58.5	55.1	53.0	52.8	51.0	50.5	50.6	51.0
Regression	100.0	100.0	100.0	58.4	55.5	53.0	52.5	52.1	51.5	50.5	50.0
Centroid	80.9	75.5	66.8	56.3	54.5	52.6	53.0	52.5	50.5	51.9	51.7
Substitution	68.5	57.4	52.7	47.6	44.9	45.4	46.6	45.7	45.1	45.0	45.4
Bagging	59.4	57.6	55.6	55.0	55.7	56.6	56.5	56.5	55.6	56.3	55.5
Boosting (Ad)	67.0	58.9	57.6	56.2	54.0	54.2	53.8	55.5	55.1	55.5	55.3
Boosting (St)	63.2	56.6	55.5	55.2	56.0	55.4	55.0	54.6	54.5	54.5	54.1
Tree	61.2	61.2	61.2	61.2	61.2	61.2	61.2	61.2	61.2	61.2	61.2
1NN	47.5	47.5	47.5	47.5	47.5	47.5	47.5	47.5	47.5	47.5	47.5

Table 2.18: **Vowel data set** test error rates averaged over 5 training sets each of size 20 observations per class. Default tree settings (approximately 25 terminal nodes). Average standard error is 1.4%.

B	2	5	10	20	40	60	80	100	120	160	200
ECOC	1.79	1.60	1.47	1.23	1.16	1.12	1.11	1.07	1.07	1.07	1.07
Regression	2.11	2.11	2.11	1.23	1.17	1.12	1.11	1.10	1.08	1.07	1.05
Centroid	1.70	1.59	1.41	1.19	1.15	1.11	1.12	1.11	1.07	1.09	1.09
Substitution	1.44	1.21	1.11	1.00	0.95	0.96	0.98	0.96	0.95	0.95	0.96
Bagging	1.25	1.21	1.17	1.16	1.17	1.19	1.19	1.19	1.17	1.19	1.17
Boosting (Ad)	1.41	1.24	1.21	1.18	1.14	1.14	1.13	1.17	1.16	1.17	1.16
Boosting (St)	1.33	1.19	1.17	1.16	1.18	1.17	1.16	1.15	1.15	1.15	1.14
Tree	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29

Table 2.19: **Vowel data set** test error rates relative to 1NN averaged over 5 training sets each of size 20 observations per class. Default tree settings (approximately 25 terminal nodes). Average standard error is 0.030.

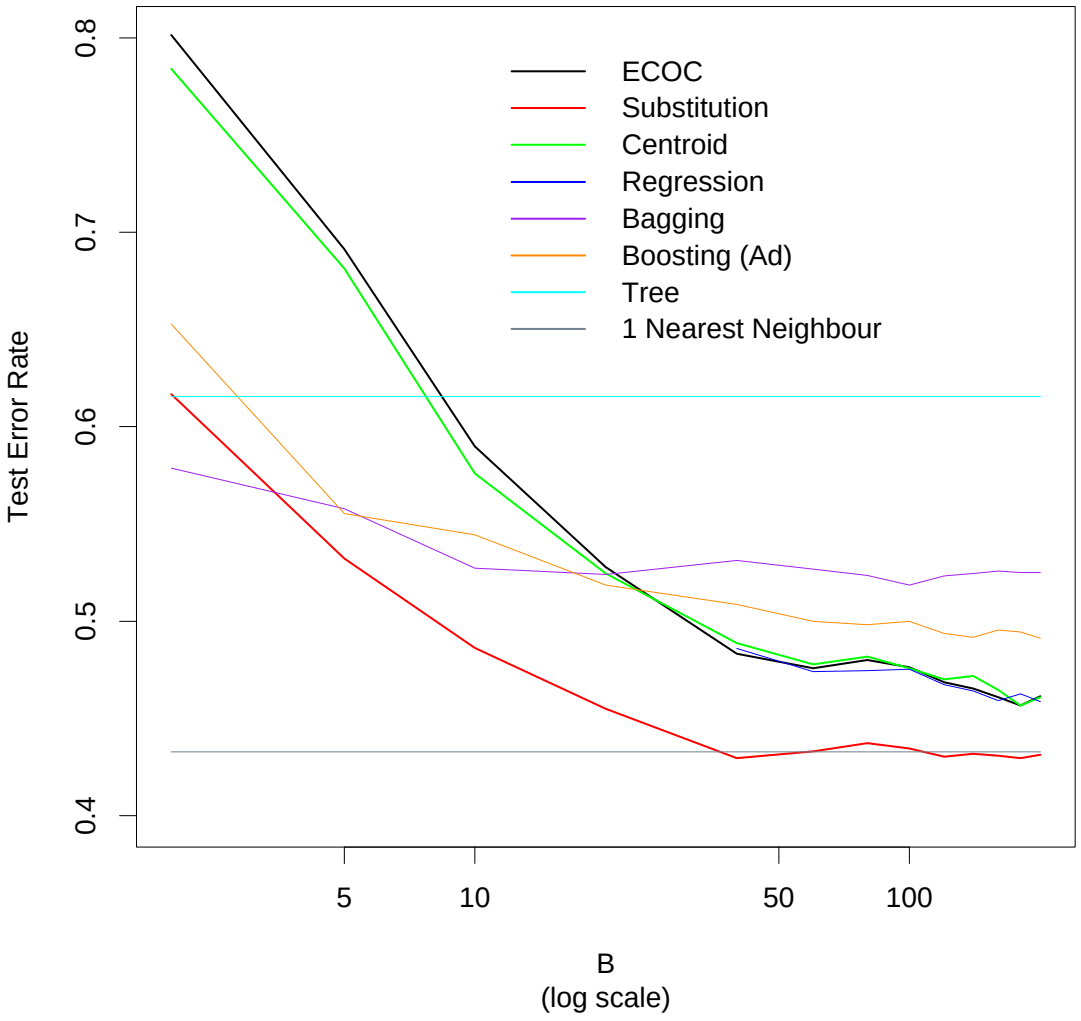

Figure 2.11: A plot of the results from Table 2.16 (Vowel data set with 40 observations per class).

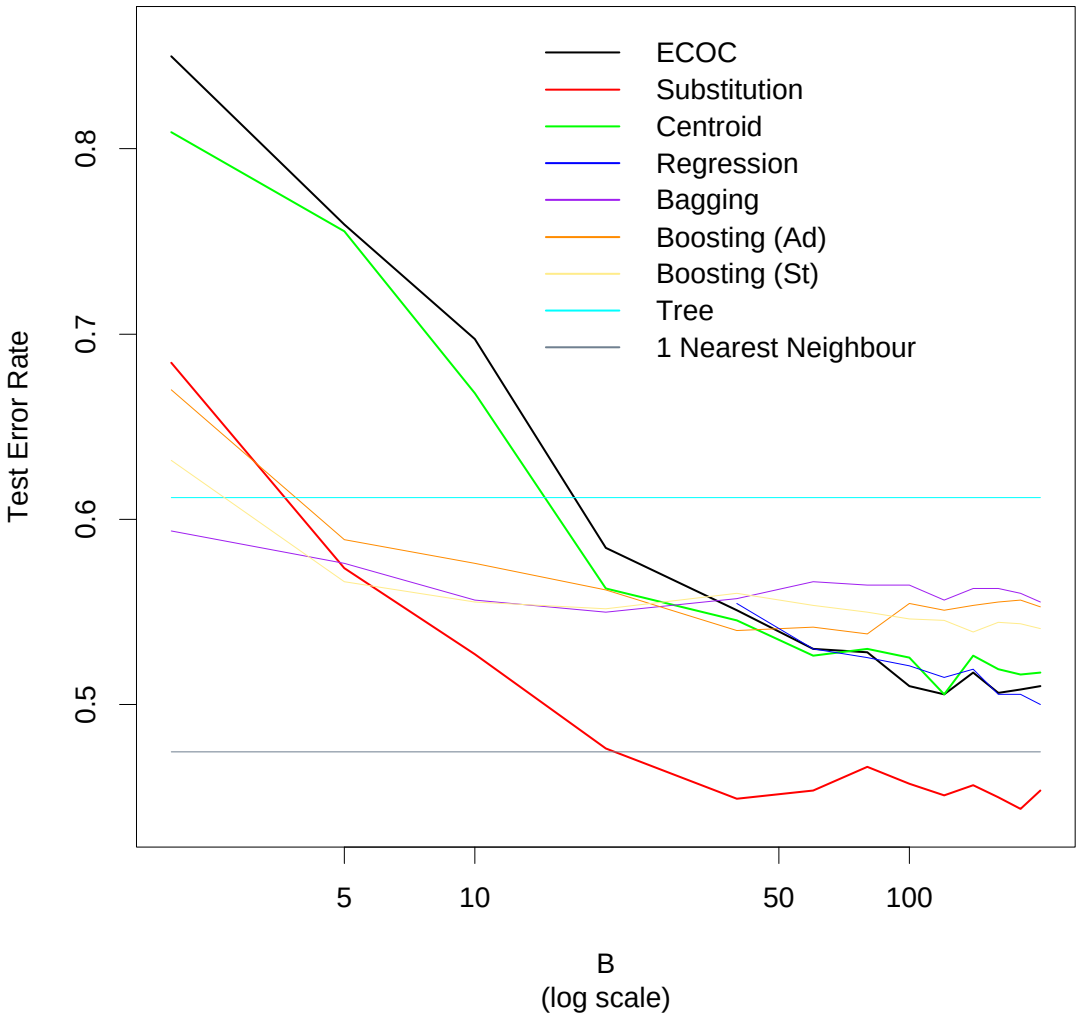

Figure 2.12: A plot of the results from Table 2.18 (Vowel data set with 20 observations per class).

Chapter 3

Classical Theories

In the previous chapter various examples of MaVLs were introduced. It is clear from Section 2.6 that members of this family of classifiers can give large improvements over using a single classifier. However, no clear theory was presented to explain why this might be the case. In the next two chapters we develop theoretical results which provide some insight into the success of this family. These theories generally fall into one of two categories which we call *Classical* and *Modern*. The Classical Theories will be discussed in this chapter and the Modern Theories in the next.

In Section 3.1 we present the basic motivation behind the classical theories. Section 3.2 develops generalizations of the concepts of bias and variance to general loss functions and general types of random variables i.e. continuous, ordinal or categorical. Section 3.3 shows how these generalizations can be used in specific situations such as classification problems. Section 3.4 provides a case study illustrating how classical ideas can be applied to the Substitution PICT. Section 3.5 gives a discussion of definitions that have recently been suggested in the literature for bias and variance and Section 3.6 provides experimental results comparing the various definitions on simulated data. The final section discusses some fundamental problems with the classical theories.

3.1 Extending Regression Theory to Classification Problems

The classical theories rely on a simple and appealing observation. Perhaps the fact that Majority Vote Classifiers are combining a large number of classifiers together is somehow

causing an averaging out of variance and this results in a reduction of the error rate, just as in a regression setting.

Recall that in a regression setting it is possible to decompose the prediction error, from using $\hat{Y}$ to estimate Y, in the following way :

$$E(\hat{Y} - Y)^2 = \underbrace{E(Y - EY)^2}_{VarY} + \underbrace{(EY - E\hat{Y})^2}_{bias(\hat{Y})^2} + \underbrace{E(\hat{Y} - E\hat{Y})^2}_{Var\hat{Y}} \tag{3.1}$$

(3.1) is an extremely useful decomposition. It can be used to prove many important results about prediction error. For example if

$$X_1, X_2, \ldots, X_n$$

are iid then it is easy to show that $Var\bar{X}_n$ must decrease as n increases and as a consequence of (3.1) the prediction error of $\bar{X}_n$ must also decrease. If $EX = EY$ then one can also show that the prediction error will approach $VarY$ as n grows large. These results are so well known and relatively simply proved that it is easy to forget how powerful they are. For example they guarantee that averaging random variables is always a good thing to do.

Unfortunately, this decomposition relies on the random variable of interest being real valued. It also makes an explicit assumption that the loss function is squared error. If one or both of these conditions fails to hold then the decomposition is no longer valid. In a classification problem the loss function is 0-1 and the random variable of interest is categorical so (3.1) will not hold. In fact, since the random variable is categorical, it is not even clear how to define variance, bias or expectation in this setting.

Therefore all classical theories have two general objectives :

1. develop definitions of bias and variance for a classification problem and
2. produce a decomposition of the error rate into bias and variance components.

The term *Classical* comes from the fact that the theories are attempting to provide generalizations of the classical regression ideas.

Section 3.2 introduces general definitions for any loss function and type of random variable and Section 3.3 shows how these definitions can be applied to a classification problem.

3.2 A Generalization of the Bias-Variance Decomposition

In this section we explore the concepts of variance and bias and develop a decomposition of the prediction error into functions of the systematic and variable parts of our predictor. In attempting this task two questions arise. Namely

- what do these quantities measure?
- and why are they useful?

3.2.1 Bias and Variance

In the regression setting the variance of an estimator $\hat{Y}$ is defined as $E(\hat{Y} - E\hat{Y})^2$. An equivalent definition is

$$Var(\hat{Y}) = \min_a E(\hat{Y} - a)^2$$

where a is non random. If we define

$$S\hat{Y} = \arg\min_a E(\hat{Y} - a)^2$$

then $Var(\hat{Y})$ is a measure of the *expected distance*, in terms of squared error loss, of the random quantity $(\hat{Y})$ from its nearest non random number $(S\hat{Y})$. We call $S\hat{Y}$ the systematic part of $\hat{Y}$ and use the notation $S\hat{Y}$ to emphasize that S is an operator acting on the distribution of $\hat{Y}$. In this case $S\hat{Y}$ will be equal to $E\hat{Y}$.

If we use $\hat{Y}$ to estimate a parameter θ then the bias of $\hat{Y}$ is defined as $S\hat{Y} - \theta$. The bias when using $\hat{Y}$ to predict Y, where $\hat{Y}$ and Y are independent random variables, is less well defined. However from the decomposition,

$$\begin{aligned} E(Y - \hat{Y})^2 &= E(Y - SY)^2 + E(\hat{Y} - SY)^2 \\ PE(Y, \hat{Y}) &= Var(Y) + MSE(\hat{Y}, SY) \end{aligned}$$

where $SY = \arg\min_a E(Y - a)^2$, we can see that the problem of predicting Y is equivalent

to one of estimating SY. This is because $Var(Y)$ is independent of $\hat{Y}$ so the mean squared error between $\hat{Y}$ and SY is the only quantity that we have control over. This motivates a definition of $(S\hat{Y} - SY)^2$ as the squared bias and means that we can think of bias as a measure of the distance between the systematic parts of $\hat{Y}$ and Y ($S\hat{Y}$ and SY).

By writing $\hat{Y} = S\hat{Y} + \epsilon$ we see that it is possible to decompose our random variable into systematic ($S\hat{Y}$) and random (ϵ) parts. Both parts contribute to any error we may make in estimation but their causes and cures can differ markedly.

3.2.2 Standard Prediction Error Decomposition

It is well known that we can decompose the expected squared error of $\hat{Y}$ from Y as follows

$$E(\hat{Y} - Y)^2 = \underbrace{Var(Y)}_{\text{irreducible error}} + \underbrace{bias^2(\hat{Y}, SY) + Var(\hat{Y})}_{\text{reducible error}} \tag{3.2}$$

so the *expected loss* of using $\hat{Y}$ is the sum of the variance of $\hat{Y}$ and Y plus the squared distance between their systematic components. The variance of Y is beyond our control and is thus known as the irreducible error. However the bias and variance of $\hat{Y}$ are functions of our estimator and can therefore potentially be reduced.

This shows us that $Var(\hat{Y})$ serves two purposes

1. it provides a measure of the variability of $\hat{Y}$ about $S\hat{Y}$
2. and it indicates the effect of this variance on the prediction error.

Similarly $bias(\hat{Y}, SY)$ serves two purposes

1. it provides a measure of the distance between the systematic components of Y and $\hat{Y}$
2. and by squaring it we see the effect of this bias on the prediction error.

This double role of both bias and variance is so automatic that we often fail to consider it. However when we extend these definitions to arbitrary loss functions it will not, in general, be possible to define one statistic to serve both purposes.

3.2.3 Generalizing the Definitions

Often squared error is a very convenient loss function to use. It possesses well known mathematical properties such as the bias/variance decomposition (3.2) that make it very attractive to use. However there are situations where squared error is clearly not the most appropriate loss function. This is especially true in classification problems where a loss function like 0-1 loss seems much more realistic.

Requirements for a Reasonable Generalization

So how might we extend these concepts of variance and bias to general loss functions? There is one obvious requirement that it seems natural for any generalization to fulfill.

❶ *When using squared error loss our general definitions must reduce to the standard ones.*

Unfortunately ❶ is not a strong enough requirement to ensure a unique generalization. This is a result of the large number of definitions for variance and bias that are equivalent for squared error but not for other loss functions.

For example the following definitions are all equivalent for squared error.

➤ $Var(\hat{Y}) = \min_a E(\hat{Y} - a)^2 = E(\hat{Y} - S\hat{Y})^2$

➤ $Var(\hat{Y}) = MSE(\hat{Y}, SY) - bias^2(\hat{Y}, SY) = E(\hat{Y} - SY)^2 - (S\hat{Y} - SY)^2$

➤ $Var(\hat{Y}) = PE(Y, \hat{Y}) - E(Y - S\hat{Y})^2 = E(Y - \hat{Y})^2 - E(Y - S\hat{Y})^2$

These lead naturally to three possible generalized definitions.

I. $Var(\hat{Y}) = \min_a EL(\hat{Y}, a) = EL(\hat{Y}, S\hat{Y})$

II. $Var(\hat{Y}) = EL(\hat{Y}, SY) - L(S\hat{Y}, SY)$

III. $Var(\hat{Y}) = EL(Y, \hat{Y}) - EL(Y, S\hat{Y})$

where L is a general loss function, $S\hat{Y} = \arg\min_a L(\hat{Y}, a)$ and $SY = \arg\min_a L(Y, a)$

For general loss functions these last three definitions certainly need not be consistent. This inconsistency accounts for some of the differences in the definitions that have been

proposed. For example Tibshirani, 1996 bases his definition of variance on I while Dietterich and Kong, 1995 base theirs more closely on III. We will see later that both I and III are useful for measuring different quantities.

What other requirements should a definition of variance fulfill? We can think of $\hat{Y} \sim g(F_{Tr})$ where g is a function that depends on the method used to obtain $\hat{Y}$ from the observations and F_{Tr} is the distribution of the observations or *training data (Tr)*. While $Y \sim F_{Te}$ where F_{Te} is the distribution of the *test data*. Often F_{Tr} and F_{Te} are assumed the same but in general they need not be. So we see that variance is a function of g and F_{Tr} but is not a function of F_{Te}. This is desirable because it allows us to compare estimators across different test sets (a low variance estimator for one test set will also be low variance for a second test set). So another natural requirement is

❷ *The variance must not be a function of the distribution of the test data, F_{Te}.*

This requirement rules out II and III given above which in general will be a function of F_{Te}.

There is a similar requirement on the bias. Since bias is a measure of the distance between the systematic components of $\hat{Y}$ and Y we require that

❸ *The bias must be a function of $\hat{Y}$ and Y through $S\hat{Y}$ and SY only (i.e. bias must be a function of $S\hat{Y}$ and SY).*

General Definitions of Bias and Variance

With ❶,❷ and ❸ in mind the most natural definitions of bias and variance are :

	Loss Function	
	Squared Error	General
Variance	$E(\hat{Y} - S\hat{Y})^2$ $S\hat{Y} = \arg\min_a E(\hat{Y} - a)^2$	$EL(\hat{Y}, S\hat{Y})$ $S\hat{Y} = \arg\min_a EL(\hat{Y}, a)$
Bias^2	$(SY - S\hat{Y})^2$	$L(SY, S\hat{Y})$

This definition of variance is identical to that given in Tibshirani, 1996 but my definition of bias differs from his. My definition of bias is equivalent to that of bias^2 for squared error. It should be noted that, even with the restrictions we have listed, these definitions by no

means represent a unique generalization of the concepts of bias and variance. However, as we will see in the next section, these statistics may not be our primary concern.

3.2.4 Bias and Variance Effect

While these definitions fulfill the stated conditions they have one major drawback. Namely in general there is no way to decompose the error into any function of bias and variance, as is the case for squared error loss (3.2). In fact it is possible to construct examples (see Section 3.3.2) where the variance and bias are constant but the reducible prediction error changes as we alter the test distribution.

Often we will be interested in the *effect* of bias and variance. For example it is possible to have an estimator with high variance but for this variance to have little impact on the error rate. It is even possible for increased variance to cause a lower error rate (see Section 3.3.1). We call the change in error caused by variance the *Variance Effect* (VE) and the change in error caused by bias the *Systematic Effect* (SE). For squared error the variance effect is just the variance and the systematic effect is the bias squared. However in general this will not be the case.

Recall in the standard situation we can decompose the expected squared error as follows,

$$E(Y-\hat{Y})^2 = \underbrace{Var(Y)}_{\text{irreducible error}} + \underbrace{bias^2(\hat{Y},SY) + Var(\hat{Y})}_{\text{reducible error}}$$

but note

$$\begin{aligned} Var(Y) &= E(Y-SY)^2 \\ bias^2(\hat{Y},SY) &= (SY-S\hat{Y})^2 \\ &= E[(Y-S\hat{Y})^2-(Y-SY)^2] \\ Var(\hat{Y}) &= E(\hat{Y}-S\hat{Y})^2 \\ &= E[(Y-\hat{Y})^2-(Y-S\hat{Y})^2] \end{aligned}$$

Remember for squared error $SY = EY$ and $S\hat{Y} = E\hat{Y}$. This gives the following decomposition

$$\underbrace{EL_S(Y,\hat{Y})}_{PE} = \underbrace{EL_S(Y,SY)}_{Var(Y)} + \underbrace{E[L_S(Y,S\hat{Y}) - L_S(Y,SY)]}_{bias^2(\hat{Y},SY)} + \underbrace{E[L_S(Y,\hat{Y}) - L_S(Y,S\hat{Y})]}_{Var(\hat{Y})}$$

where L_S is squared error loss. Note that everything is defined in terms of prediction error of Y with respect to L_S.

Notice that, in this formulation, $bias^2$ is simply the change in prediction error when using $S\hat{Y}$, instead of SY, to predict Y; in other words it is the change in prediction error caused by bias. This is exactly what we have defined as the systematic effect. Similarly $Var(\hat{Y})$ is the change in prediction error when using $\hat{Y}$, instead of $S\hat{Y}$, to predict Y; in other words the change in prediction error caused by variance. This is what we have defined as the variance effect.

This decomposition will hold for any loss function so in general we define

$$\begin{aligned} & SE(\hat{Y},Y) &=& E[L(Y,S\hat{Y}) - L(Y,SY)] \\ \text{and} \quad & VE(\hat{Y},Y) &=& E[L(Y,\hat{Y}) - L(Y,S\hat{Y})] \end{aligned}$$

Notice that the definition of VE corresponds to III in Section 3.2.3. We now have a decomposition of prediction error into errors caused by variability in Y ($Var(Y)$), bias between Y and $\hat{Y}$ ($SE(\hat{Y},Y)$) and variability in $\hat{Y}$ ($VE(\hat{Y},Y)$).

$$\begin{aligned} EL(Y,\hat{Y}) &= \underbrace{EL(Y,SY)}_{Var(Y)} + \underbrace{E[L(Y,S\hat{Y}) - L(Y,SY)]}_{SE(\hat{Y},Y)} + \underbrace{E[L(Y,\hat{Y}) - L(Y,S\hat{Y})]}_{VE(\hat{Y},Y)} \\ &= Var(Y) + SE(\hat{Y},Y) + VE(\hat{Y},Y) \end{aligned} \tag{3.3}$$

Now in general there is no reason for $Var(\hat{Y})$ to equal $VE(\hat{Y},Y)$ or for $bias(\hat{Y},SY)$ to equal $SE(\hat{Y},Y)$. *Often it will be the variance and bias effects that we are more interested in rather than the variance and bias itself.* One of the nice properties of squared error loss

is that $VE = Var$ so the variance effect, like the variance, is constant over test sets. In general this will not be the case.

Note that due to the fact L is a loss function, $Var(Y) \geq 0$, and by the definition of SY, $SE(\hat{Y}, Y) \geq 0$. However the only restriction on $VE(\hat{Y}, Y)$ is, $VE(\hat{Y}, Y) \geq -SE(\hat{Y}, Y)$. Indeed we will see examples where the variance effect is negative.

3.3 Applications of the Generalizations of Bias and Variance

All calculations in the following two examples are performed at a fixed input X. We have not included X in the notation to avoid confusion.

3.3.1 0-1 Loss

Suppose our loss function is $L(a, b) = I(a \neq b)$. We will now use the notation C and SC instead of $\hat{Y}$ and $S\hat{Y}$ to emphasize the fact that this loss function is normally used in classification problems so our predictor typically takes on categorical values: $C \in \{1, 2, \ldots, k\}$ for a k class problem.

Further define

$$\begin{aligned} & P_i^Y &=& Pr(Y = i) \\ \text{and} \quad & P_i^C &=& Pr(C = i) \end{aligned}$$

where i runs from 1 to k. Recall that $C \sim g(F_{Tr})$, and hence P_i^C are based on averages over training sets. With this loss function we see

$$\begin{aligned} SY &= \arg\min_i E(I(Y \neq i)) \\ &= \arg\min_i \sum_{j \neq i} P_j^Y \\ &= \arg\max_i P_i^Y \quad \text{i.e. the bayes classifier} \\ \text{and} \quad SC &= \arg\max_i P_i^C \quad \text{i.e. the mode of } C \end{aligned}$$

We now get

$$\begin{aligned}
VE(C,Y) &= E(I(Y \neq C) - I(Y \neq SC)) \\
&= P(Y \neq C) - P(Y \neq SC) \\
&= \sum_i P_i^Y (1 - P_i^C) - (1 - P_{SC}^Y) \\
Var(C) &= \min_a EI(C \neq a) \\
&= P(C \neq SC) \\
&= 1 - \max_i P_i^C \\
&= 1 - P_{SC}^C \\
SE(C,Y) &= E(I(Y \neq SC) - I(Y \neq SY)) \\
&= P(Y \neq SC) - P(Y \neq SY) \\
&= P_{SY}^Y - P_{SC}^Y \\
&= \max_i P_i^Y - P_{SC}^Y \\
bias(C,SY) &= I(SC \neq SY) \\
Var(Y) &= EI(Y \neq SY) \\
&= P(Y \neq SY) \\
&= 1 - \max_i P_i^Y \\
&= 1 - P_{SC}^Y
\end{aligned}$$

A simple example will provide some illumination. Suppose Y has the following distribution.

y	0	1	2
$Pr(Y = y)$	0.5	0.4	0.1

Now we compare two classifier random variables (at a fixed predictor X) with the following distributions :

c	0	1	2
$Pr(C_1 = c)$	0.4	0.5	0.1
$Pr(C_2 = c)$	0.1	0.5	0.4

$SY = 0$, $SC_1 = SC_2 = 1$ and $SE(C, Y)$ equals 0.1 for both classifiers. These two classifiers have identical distributions except for a permutation of the class labels. Since the labels have no ordering we would hope that both classifiers have the same variance. In fact $Var(C_1) = Var(C_2) = 1 - 0.5 = 0.5$. However the effect of this variance is certainly not the same for each classifier.

$$\begin{aligned} VE(C_1, Y) &= P(Y \neq C_1) - P(Y \neq SC_1) = 0.59 - 0.6 = -0.01 \\ VE(C_2, Y) &= P(Y \neq C_2) - P(Y \neq SC_2) = 0.71 - 0.6 = 0.11 \end{aligned}$$

The variance of C_1 has actually caused the error rate to decrease while the variance of C_2 has caused it to increase! This is because the variance in C_1 is a result of more classifications being made to 0 which is the bayes class while the variance in C_2 is a result of more classifications being made to 2 which is a very unlikely class to occur. Therefore, we see that it does not necessarily follow that increasing the variance of C_1 would cause a further reduction in $VE(C_1, Y)$ or that decreasing the variance of C_2 would cause a reduction in $VE(C_2, Y)$. Friedman, 1996b noted, that for 0-1 loss functions, increasing the variance can actually cause a reduction in the error rate as we have seen with this example.

3.3.2 Absolute Loss

Although the 0-1 loss function is of primary concern in this setting, it should be noted that these definitions can be applied to any loss function. To illustrate this we will consider the situation where the loss function is $L(a, b) = |a - b|$. What decomposition does this give?

$$\begin{aligned} EL(\hat{Y}, Y) &= Var(Y) + SE(\hat{Y}, Y) + VE(\hat{Y}, Y) \\ &= EL(Y, SY) + E[L(Y, S\hat{Y}) - L(Y, SY)] \\ &\quad + E[L(Y, \hat{Y}) - L(Y, S\hat{Y})] \\ \Rightarrow E|Y - \hat{Y}| &= E|Y - SY| + E(|Y - S\hat{Y}| - |Y - SY|) \\ &\quad + E(|Y - \hat{Y}| - |Y - S\hat{Y}|) \\ &= E|Y - med(Y)| + E(|Y - med(\hat{Y})| - |Y - med(Y)|) \\ &\quad + E(|Y - \hat{Y}| - |Y - med(\hat{Y})|) \end{aligned}$$

where $med(Y)$ is the median of Y.

This gives

$$\begin{aligned} VE(\hat{Y},Y) &= E(|Y-\hat{Y}|-|Y-med(\hat{Y})|) \\ Var(\hat{Y}) &= EL(\hat{Y},S\hat{Y}) = E|\hat{Y}-med(\hat{Y})| \\ SE(\hat{Y},Y) &= E(|Y-med(\hat{Y})|-|Y-med(Y)|) \\ bias(\hat{Y},SY) &= L(SY,S\hat{Y}) = |med(Y)-med(\hat{Y})| \\ Var(Y) &= \text{irreducible error} = E|Y-med(Y)| \end{aligned}$$

A simple example illustrates the concepts involved. Suppose Y is a random variable with the following distribution :

y	0	1	2
$Pr(Y=y)$	$a/4$	$1/2$	$(2-a)/4$

We will start with $a = 1$. Suppose our estimator is simply the constant $\hat{Y} = 2$. Then clearly $med(Y) = 1$ and $med(\hat{Y}) = 2$ so $bias(\hat{Y}, SY) = 1$. Note that both $Var(\hat{Y})$ and $VE(\hat{Y},Y)$ are zero so the systematic effect is the only relevant quantity in this case.

$$\begin{aligned} SE(\hat{Y},Y) &= E(|Y-med(\hat{Y})|-|Y-med(Y)|) \\ &= E(|Y-2|-|Y-1|) \\ &= 1-1/2 = 1/2 \end{aligned}$$

So the SE is not equal to the bias. We can show that SE is not a function of the bias by altering a. Notice that for $0 < a < 2$ the median of Y remains unchanged at 1. So the bias is also constant. However,

$$\begin{aligned} SE(\hat{Y},Y) &= E(|Y-med(\hat{Y})|-|Y-med(Y)|) \\ &= E(|Y-2|-|Y-1|) \\ &= 2\cdot\frac{a}{4}+1\cdot\frac{1}{2}+0\cdot\frac{2-a}{4}-(1\cdot\frac{a}{4}+0\cdot\frac{1}{2}+1\cdot\frac{2-a}{4}) \\ &= a/2 \end{aligned}$$

So as a approaches 0 so does the SE! In other words it is possible to have an estimator

that is systematically wrong but with an arbitrarily low reducible loss associated with it.

3.4 Case Study : The Substitution PICT

In this section we show how Classical Ideas can be used to provide insight into the success of the Substitution PICT and hence the ECOC PICT. The Substitution PICT is described in Section 2.4.

Recall that the probability estimates, p_i^S, in the Substitution PICT are formed by averaging over B different trees.

$$p_i^S = \frac{1}{B}\sum_{j=1}^{B} p_{ij}$$

The fact that p_i^S is an average of probability estimates suggests that a reduction in variability, without a complementary increase in bias, may be an explanation for the success of the Substitution PICT. This observation alone can not provide the answer, however, because it has been clearly demonstrated (see for example Friedman, 1996b) that a reduction in variance of the probability estimates does not necessarily correspond to a reduction in the error rate. The quantity that we are interested in is not the individual probabilities but $\arg\max_j p_j$. Now

$$i = \arg\max_j p_j \quad \text{iff} \quad p_i - p_j > 0 \quad \forall j \neq i$$

So what we are really interested in are the random variables $p_i - p_j$. However, even the variances of these variables are not enough because variance is not independent of scale. For example by dividing all the probabilities by 2 we could reduce the variance by a factor of 4 but the probability that $p_i - p_j > 0$ would remain unchanged. A better quantity to consider is the coefficient of variation,

$$CV(p_i - p_j) = \sqrt{\frac{Var(p_i - p_j)}{(E(p_i - p_j))^2}}$$

If the probability estimates are normally distributed there is a direct correspondence between $CV(p_i - p_j)$ and the probability that $p_i - p_j > 0$ i.e.

$$Pr(p_i - p_j > 0) = \Phi\left(E(p_i - p_j)/\sqrt{Var(p_i - p_j)}\right) = \Phi\left(1/CV(p_i - p_j)\right)$$

Notice that for a two class problem ($k = 2$) this implies a lower CV will give a lower error rate. An assumption of normality may not be too bad, but in any case we would expect a similar relationship for any *reasonable* distribution. For example, if p_i^T is the probability estimate for the ith class from an ordinary k class tree classifier, we might suppose that the Substitution PICT will have a superior performance provided

$$CV(p_i^S - p_j^S) < CV(p_i^T - p_j^T) \tag{3.4}$$

To examine when (3.4) might hold we use the following semi-parametric model for the probability estimates,

$$\begin{aligned} p_i^S &= \alpha_S f(q_i) + \sigma_S \epsilon_i^S \quad E\epsilon_i^S = 0 \\ p_i^T &= \alpha_T f(q_i) + \sigma_T \epsilon_i^T \quad E\epsilon_i^T = 0 \end{aligned}$$

where f is an arbitrary increasing function, α_S and α_T are positive constants and $\boldsymbol{\epsilon}^S = (\epsilon_1^S, \ldots, \epsilon_k^S)$ and $\boldsymbol{\epsilon}^T = (\epsilon_1^T, \ldots, \epsilon_k^T)$ have arbitrary but identical distributions. Recall that $q_i = P(G = i \mid X)$. This model makes few assumptions about the specific form of the probability estimates but does assume that the ratio Ep_i^S/Ep_i^T is constant and that the error terms ($\boldsymbol{\epsilon}^S$ and $\boldsymbol{\epsilon}^T$) have the same distribution.

Under this modeling assumption it can be shown that (3.4) holds iff

$$\frac{\sigma_S}{\alpha_S} < \frac{\sigma_T}{\alpha_T} \tag{3.5}$$

(3.5) states that the standardized variance of the Substitution PICT is less than that for the tree classifier. Note that (3.5) is also equivalent to the signal to noise ratio of the k class tree classifier being less than that of the Substitution PICT.

The question remains, under what conditions will (3.5) hold? The probability estimates from the Substitution PICT are formed from an average of B correlated random variables (p_{ij}) so we know that σ_S (which depends on B) will decrease to a positive limit as B increases. Intuitively this suggests that (3.5) will hold provided

I. B is large enough (so we are close to the limit),

II.
$$\gamma = \frac{Var(p_i^T/\alpha_T)}{Var(p_{i1}/\alpha_S)}$$
is large enough (so the standardized variance of p_{ij} is not too large relative to that of p_i^T),

III. and $\rho = Corr(p_{i1}, p_{i2})$ is low enough (so that a large enough reduction can be achieved by averaging).

Note that γ is the ratio of the noise to signal ratio (NSR) of the k class tree classifier to that of a *single tree* from the Substitution PICT. We assume γ is constant for all i. In fact we can formalize this intuition in the following theorem.

Theorem 8 *Under the previously stated semi-parametric model assumptions (3.4) and (3.5) will hold iff*

$$\rho < \gamma \qquad (\rho \text{ is small relative to } \gamma) \tag{3.6}$$

and

$$B \geq \frac{1-\rho}{\gamma-\rho} \qquad (B \text{ is large enough}) \tag{3.7}$$

Further more if $k = 2$ (there are only two classes) then (3.6) and (3.7) are sufficient to guarantee a reduction in the error rate.

Even in a situation where there are more than two classes it will often be the case that at any point in the predictor space there are effectively only two possible classes to choose between. Therefore, in practice (3.6) and (3.7) will often be sufficient to guarantee a lower error rate.

Now there is reason to believe that in general ρ will be small. This is a result of the empirical variability of tree classifiers. A small change in the training set can cause a large change in the structure of the tree and also the final probability estimates. So by changing the super group coding we might expect a probability estimate that is fairly unrelated to previous estimates and hence a low correlation.

To test the accuracy of this theory we examined the results from the simulation performed in Section 2.4. We wished to estimate γ and ρ. For this data it was clear that f could be well approximated by a linear function so our estimates for α_S and α_T were obtained using least squares. The following table summarizes our estimates for the variance and standardizing (α) terms from the simulated data set.

Classifier	$Var(p_i)$	α	$Var(p_i/\alpha)$
Substitution PICT	0.0515	0.3558	0.4068
Tree Method	0.0626	0.8225	0.0925

The table indicates that, when we account for the shrinkage in the Substitution PICT probability estimates ($\alpha_S = 0.3558$ vs $\alpha_T = 0.8225$), the NSR for a *single tree* from the Substitution PICT is over 4 times that of an ordinary k class tree (0.4068 vs 0.0925). In other words the estimate for γ is $\hat{\gamma} = 0.227$ so the signal to noise ratio of a single tree in the Substitution PICT is only about a quarter of that from an ordinary tree classifier. However, the estimate for ρ was very low at only $\hat{\rho} = 0.125$.

It is clear that ρ is less than γ so provided B is large enough we expect to see an improvement by using the Substitution PICT. From Theorem 8 we can estimate the required size of B as

$$B \geq \frac{1-\hat{\rho}}{\hat{\gamma}-\hat{\rho}} \approx 9$$

We see from Figure 3.1 that the Substitution error rate drops below that of the tree classifier at almost exactly this point, providing some validation for the theory. Together with Theorem 7 this result also provides further motivation for the success of the ECOC PICT.

3.5 Discussion of Recent Literature

Dietterich and Kong, 1995, Kohavi and Wolpert, 1996, Breiman, 1996b, Tibshirani, 1996, and Friedman, 1996b have all recently written papers on the topic of bias and variance for classification rules.

Kohavi and Wolpert

Kohavi and Wolpert, 1996 define bias and variance of a classifier in terms of the squared error when comparing P_i^C to P_i^Y. For a two class problem they define the squared bias as

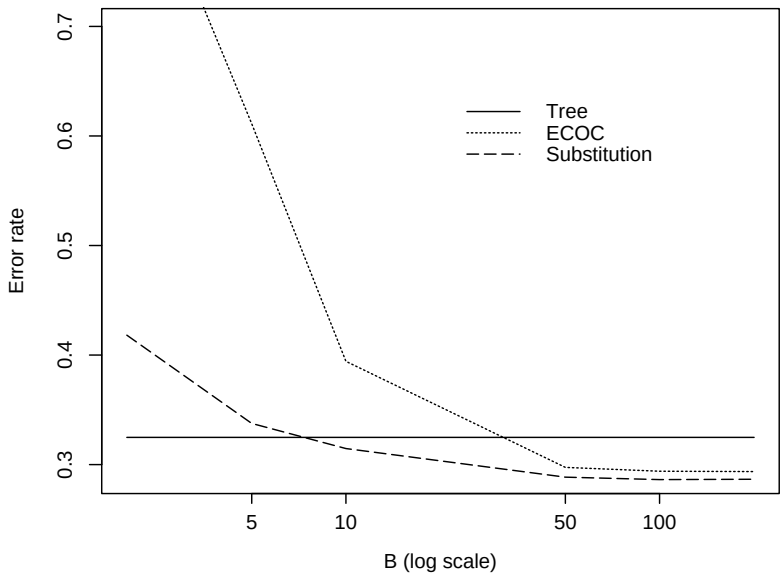

Figure 3.1: Error rates on the simulated data set for the tree method, Substitution PICT and ECOC PICT plotted against B (on log scale)

$(P_1^Y - P_1^C)^2$ and the variance as $P_1^C(1 - P_1^C)$ which are as one would expect for squared error. As a result the Bayes Classifier will have a positive squared bias unless $P_1^Y \in \{0, 1\}$.

Dietterich and Kong

The definitions of Dietterich and Kong, 1995, Breiman, 1996b, and Tibshirani, 1996 are more similar in spirit to those in this chapter. Dietterich and Kong, 1995 define $bias = I(Pr(C \neq Y) \geq 1/2)$ and $var = Pr(C \neq Y) - bias$. This gives the decomposition

$$Pr(C \neq Y) = var + bias$$

From these definitions we can note the following

- although not immediately apparent, this definition of bias coincides with ours ($I(SY = S\hat{Y})$) for the 2 class situation,
- for $k > 2$ the two definitions are not consistent which can be seen from the fact that for our definition of bias the Bayes Classifier will have zero bias while for Dietterich

and Kong's it is possible for the Bayes Classifier to have positive bias

- and the variance term will be negative whenever the bias is non zero.

Breiman

Breiman's definitions (Breiman, 1996b) are in terms of an "aggregated" classifier which is the equivalent of SC for a 0-1 loss function. He defines a classifier as unbiased at X if $SY = SC$ and lets U be the set of all X at which C is unbiased. He also defines the complement of U as the bias set and denotes it by B. He then defines the bias and variance over the entire test set as

$$bias(C) = P_X(C \neq Y, X \in B) - P_X(SY \neq Y, X \in B)$$

$$var(C) = P_X(C \neq Y, X \in U) - P_X(SY \neq Y, X \in U)$$

This is equivalent to defining bias and variance at a fixed X as

$$bias = \begin{cases} P(C \neq Y) - P(SY \neq Y) & SY \neq SC \\ 0 & SY = SC \end{cases}$$

$$var = \begin{cases} P(C \neq Y) - P(SY \neq Y) & SY = SC \\ 0 & SY \neq SC \end{cases}$$

This definition has the following appealing properties :

- Bias and variance are always non-negative.
- If C is deterministic then its variance is zero (hence SC has zero variance).
- The bias and variance of SY is zero.

However we note that at any fixed X the entire reducible error (total error rate minus bayes error rate) is either assigned to variance (if C is unbiased at X) or to bias (if C is biased at X). Certainly it seems reasonable to assign all the reducible error to variance if C is unbiased (if C were unbiased and did not vary it would be equal to the bayes classifier). However when C is biased it does not seem reasonable to assign all reducible errors to bias. Even when C is biased, variability can cause the error rate to increase or decrease (as illustrated in Section 3.3.1) and this is not reflected in the definition.

Tibshirani

Tibshirani, 1996 defines variance, bias and a prediction error decomposition for *classification rules (categorical data)*. Within this class of problems his definition of variance is identical to that given in this paper. He defines a quantity AE (Aggregation Effect), which is equal to the variance effect we have defined, and for most common loss functions his definition of bias will be equivalent to our systematic effect. This gives a decomposition of

$$Pr(C \neq Y) = Pr(Y \neq SY) + Bias(C) + AE(C)$$

which is identical to ours. However, it should be noted that although these definitions are generalizable to any loss function they do not easily extend beyond the class of "classification rules" to general random variables (e.g. real valued). It is comforting that when we restrict ourselves to this smaller class the two sets of definitions are almost identical.

Friedman, 1996b provides a good comparison of the different definitions.

3.6 Experimental Comparison of Different Definitions

To provide an experimental comparison of some of the definitions for variance and bias that have been suggested, we performed simulations using two artificial data sets.

First Data Set

The first data set consisted of 26 classes with the distribution of each class being a standard bivariate normal with identity covariance matrix. Many independent training sets with 10 observations per class were chosen. On each of these training sets 7 different classifiers were trained and their classifications, on each of 1040 test points, were recorded. This allowed $P(C = i)$ to be estimated for each test point. Since each class followed a normal distribution it was possible to calculate $P(Y = i)$. This in turn allowed estimates for bias and variance (under the various definitions) to be calculated for each of the classifiers. The 7 different classifiers were LDA, ECOC, Bagging, a Tree, 1 Nearest Neighbour, 5 Nearest Neighbour and 11 Nearest Neighbour. On the first 4 classifiers 100 training sets were used. However, it was discovered that the estimates of bias for Nearest Neighbours were inaccurate for this

number so 1000 training sets were used for the last 3 classifiers. Estimates for bias and variance were made using Dietterich and Breiman's definitions as well as those given in this chapter. The results are shown in Table 3.1.

The first thing we notice from these results is that LDA performs exceptionally well. This is not surprising because it can be shown that LDA is asymptotically optimal for mixtures of normals as we have in this case. Both Breiman's bias estimate and the systematic effect indicate no bias effect. This is comforting since we know that LDA has no bias on this data set. The James estimate of bias is not zero (1.6%). This is due to the relatively low number of training samples. It can be shown that this estimate will converge to zero as the number of training sets increases. On the other hand Dietterich's bias estimate is extremely high which makes less sense.

The next point to note is that Breiman's bias estimate is very similar to the systematic effect and his variance estimate is similar to the variance effect. His estimate of the bias contribution seems to be consistently below or equal to that of the systematic effect. This slight difference between the two definitions is due to the fact that, at any given test point, all the reducible error is attributed to either bias or variance (see Section 3.5).

On the other hand Dietterich's definitions produce quite different estimates. They tend to attribute almost all the error rate to bias rather than variance. This is partly due to the fact that no allowance is made for the positive Bayes Error (23.1%). However, even when the Bayes Error is subtracted off there are still some anomalies such as LDA having a negative bias.

If we examine the 3 Nearest Neighbour classifiers we can get an idea of the effect on variance and bias of increasing the number of neighbours. Notice that as the number of neighbours increases the variance (and variance effect) decreases which is as we would expect. However the bias estimate also decreases slightly which is not what we would expect. This happens with all the definitions. In fact the bias is not decreasing. There is a tendency to overestimate bias if it is very low or zero. 11 Nearest Neighbours averages each of its classifications over 11 points for each training data set so is using $11,000$ data points. This produces a good estimate for bias. However, 1 Nearest Neighbours is only using $1,000$ data

points which gives a higher estimate for bias. It is likely in both cases that the true bias is almost zero. This is evidenced by the fact that the systematic effect is zero.

The ECOC and Bagging PICTs are both constructed by combining 100 of the tree classifiers that are shown in column 4. Notice that both methods have reduced the variance (and variance effect) as the classical theories would predict. However, they have also reduced the bias (and systematic effects) which could not happen in a regression setting!

Lastly note that while in theory there need not be any relationship between bias and systematic effect and between variance and variance effect, in practice there is a strong relationship. So bias and variance are good predictors for the effects of these two quantities.

Second Data Set

The second data set is similar to the first except that in this one there were only 10 classes and only 5 training data points per class were used. For this data set eight classifiers were used. They were LDA, ECOC, Bagging, a tree classifier with 5 terminal nodes, a tree classifier with 8 terminal nodes, a tree classifier with 13 terminal nodes, 1 nearest neighbour and 11 nearest neighbour. The results are presented in Table 3.2.

Many of the conclusions from the first data set hold for the second. LDA performs extremely well again, with a very low bias. Also Breiman's definitions produce similar results to those of systematic and variance effect. Notice that Dietterich's definition can result in a negative variance. Also note that while in theory the variance effect can be negative it is not for any of the examples we examine.

As the number of terminal nodes in a tree increases we would expect the bias to decrease and the variance to increase. In this example the bias (and systematic effect) do both decrease. However, the variance (and variance effect) also both decrease. This is possible if, by increasing the number of terminal nodes, we average over lower variance data points.

Classifier	LDA	ECOC	Bagging	Tree	1NN	5NN	11NN
Bias (Dietterich)	21.5	27.4	27.7	32.2	27.7	25.3	24.1
Bias less Bayes Error	−1.6	4.3	4.6	9.1	4.6	2.2	1.0
Variance (Dietterich)	3.3	1.9	2.0	1.3	3.3	2.3	2.4
Bias (Breiman)	0.0	0.4	1.1	1.6	0.1	0.0	0.0
Variance (Breiman)	1.7	5.9	5.5	8.8	7.8	4.5	3.4
Bias	1.6	5.2	6.1	8.5	1.6	1.2	0.9
Variance	10.5	20.6	19.3	25.1	24.8	18.8	16.3
Systematic Effect	0.0	0.5	1.5	2.2	0.0	0.0	0.0
Variance Effect	1.7	5.8	5.1	8.2	7.9	4.5	3.5
Bayes Error	23.1	23.1	23.1	23.1	23.1	23.1	23.1
Prediction Error	24.8	29.3	29.7	33.5	31.0	27.6	26.5

Table 3.1: Bias and variance for various definitions calculated on a simulated data set with 26 classes.

Conclusions

Dietterich's definitions seem to assign far too high a proportion of the prediction error to bias rather than variance. His definitions do not take into account the Bayes Error. As the Bayes Error increases the bias will also tend to increase which does not seem sensible. His definition of bias may work better for a two class problem.

Both Breiman's definitions and those presented in this chapter seem to produce reasonable estimates with Breiman's tending to put slightly more weight on variance. However, it is difficult to get an accurate estimate of bias when the bias is low. There is a tendency to overestimate so a large number of training samples are required.

Classifier	LDA	ECOC	Bagging	Tree_5	Tree_8	Tree_{13}	1NN	11NN
Bias (Dietterich)	12.5	24.8	20.8	73.3	41.0	28.8	24.5	18.3
Bias less Bayes Error	−8.0	4.3	0.3	52.8	20.5	8.3	4.0	−2.2
Variance (Dietterich)	11.4	5.2	7.4	−16.1	−2.8	11.7	6.1	15.3
Bias (Breiman)	0.0	0.6	0.6	17.3	2.7	0.8	0.0	0.1
Variance (Breiman)	3.3	8.8	7.2	19.3	13.9	11.7	10.1	12.9
Bias	1.3	5.5	5.0	33.0	13.3	5.8	1.0	1.5
Variance	12.0	21.4	19.3	43.8	30.1	26.0	23.7	25.4
Systematic Effect	0.0	0.8	0.9	17.5	3.3	1.0	0.0	0.1
Variance Effect	3.3	8.7	7.0	19.1	13.4	11.5	10.1	12.9
Bayes Error	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5
Prediction Error	23.9	30.0	28.4	57.1	37.2	33.0	30.6	33.5

Table 3.2: Bias and variance for various definitions calculated on a simulated data set with 10 classes.

3.7 The Fundamental Problem with Classical Theories

Despite the appeal of generalizing regression ideas such as bias and variance to classification problems, there are some fundamental problems with this line of attack.

3.7.1 Inconsistent Definitions

The first is an inconsistency in the various definitions for variance and bias. It is clear from Section 3.5 that all of the various definitions that have been proposed in the literature are slightly different. In Section 3.6 we saw that these differences could provide quite different interpretations on which effect was dominating for any particular data set.

As was noted in Section 3.2.3 these differences arise from the fact that squared error loss allows many equivalent formulations which for general loss functions certainly will not be equivalent. In this chapter we have attempted to provide motivation for the definitions that

have been proposed. However, it can be argued, that once squared error loss is removed from consideration, there is no unique definition that clearly makes more sense than others.

3.7.2 Lower Variance DOES NOT Imply Lower Error Rate

It is possible that the problem of inconsistent definitions could be remedied. However, there is a second major problem that probably can not be overcome. In general there is no known relationship between the variance of a classifier (under any reasonable definition) and the effect of that variance on the error rate. In Section 3.3.1 we saw that it is possible for two different classifiers to have identical variances but for the effect of this variance to be quite different.

Recall that the aim of Classical Theories is two fold. The first is to provide a decomposition of the error rate into a function of variance and bias (for some definition of those two quantities). This decomposition would, for example, guarantee that a reduction in variance will cause a reduction in the error rate. The second aim is to provide results which will guarantee that, for example, MaVLs reduce variance, by what ever definition we use, and hence will reduce the error rate. It seems that it is possible to achieve either one of these aims but not both!

In Section 3.2.3 we saw that it is possible to produce general definitions of bias and variance. In Section 3.2.4 we saw that it is possible to provide a decomposition of the error rate into systematic and variance effects. However, it is not possible to decompose the error rate into a function of variance and bias since there is no provable relationship between the variance and variance effect.

The fact that it is possible to construct examples where variance and variance effect move in opposite directions led to the consideration of what we call Modern Theories, which we examine in the next chapter.

Chapter 4

Modern Theories

In the previous chapter we explored *Classical* theories where the concepts of Bias and Variance are generalized to Classification Problems. This approach has the advantage of being relatively intuitive and it has produced some interesting results. However, so far, it has failed to generate any concrete theories and Section 3.7 suggests that this may be a result of fundamental problems with the method.

In this chapter we explore a new set of ideas which we call *Modern*. These ideas are specifically intended for classifiers rather than just attempting to generalize regression theories. In Section 4.1 we detail the work of Schapire and others in defining a new quantity which they call the *Margin*. Based on this quantity they have proved bounds on the test error rate which, they claim, provides an explanation for the success of Boosting. In Section 4.2 we give an experimental study to evaluate the accuracy of the bound as an indicator of the test error rate. Section 4.3 details an alternative method of utilizing the margin which we call the Normal model. The final section provides a summary and conclusion.

4.1 Margins

In their paper *Boosting the Margin : A new explanation for the effectiveness of voting methods* (Schapire et al., 1997), Schapire et al. introduce a quantity which they call the *Margin*. Based on this quantity they prove a bound on the expected test error rate and then use this bound to develop a theory to explain the success of Boosting.

4.1.1 The Margin

The margin of a MaVL is defined, at a point x in the predictor space, as

$M(x)$ = Weighted proportion of classifications, at x, to the correct class
- Maximum weighted proportion of classifications, at x, to any of the other classes.

For example, suppose we have a 3 class problem and, at a point in the predictor space, Class 1 is the correct class. If the MaVL takes an unweighted majority vote over 100 classifiers and, 50 votes are to Class 1, 30 to Class 2 and 20 to Class 3, then

$$M(x) = \frac{50}{100} - \max(\frac{30}{100}, \frac{20}{100}) = 0.2$$

On the other hand if Class 2 were correct the margin would be -0.2 and if Class 3 were the correct class it would be -0.3.

Notice that the margin has two characteristics.

I. It is always between -1 and 1 and

II. a correct classification, at x, will be made iff

$$M(x) > 0.$$

A large positive margin can be interpreted as a confident classification.

We refer to the margin from a randomly chosen training data point as the *Training Margin* and the margin from a randomly chosen test data point as the *Test Margin*. The symbol $\mathcal{D}$ is used to indicate that the distribution is taken over test points (new data) and S to indicate that the distribution is over a randomly chosen training data point. So for example

$$P_S(M(X) \leq 0) = P(\text{Training Margin} \leq 0) = \text{Training Error Rate}$$

and

$$P_{\mathcal{D}}(M(X) \leq 0) = P(\text{Test Margin} \leq 0) = \text{Expected Test Error Rate.}$$

4.1.2 A Bound on the Expected Test Error Rate

Based on these definitions it is possible to prove a bound on the expected test error rate in terms of the training margin.

Theorem 9 *(Schapire et al., 1997) Let $\mathcal{D}$ be the test distribution of interest and let S be a sample of n elements drawn independently at random from $\mathcal{D}$. Assume that the Base Classifier used in the MaVL is finite i.e. it can only take on $|\mathcal{H}| < \infty$ different possible outcomes, and let $\delta > 0$. Then with probability at least $1 - \delta$ over the random choice of training sets S the following bound exists for all $\theta > 0$:*

$$\underbrace{P_{\mathcal{D}}(M(X) \leq 0)}_{\textit{Expected test error rate}} \leq P_S(M(X) \leq \theta) + O\left(\frac{1}{\sqrt{n}}\left(\frac{\log(nk)\log|\mathcal{H}|}{\theta^2} + \log(1/\delta)\right)^{1/2}\right)$$

See Schapire et al., 1997 for the proof.

This theorem tells us that with high probability the test error rate will be low, provided there is high certainty of the classifications on most of our training data ($P_S(M(X) \leq \theta)$ is low for moderate θ) and the Base Classifier is not too complex ($\log|\mathcal{H}|$ is not too large). Notice that the bound is independent of B, the number of votes, and the particular form of the Base Classifier. This bound is potentially useful for two reasons :

I. It is much easier to prove results about the training margin than it is for the expected test error rate. In Section 4.1.3 it is shown that AdaBoost can drive $P_S(M(X) \leq \theta)$ to 0. Therefore if a strong relationship between the training margin and the expected test error rate can be established this would be a large step towards explaining the success of Boosting and perhaps MaVLs in general. Theorem 9 provides a relationship between these two quantities but we will see empirical results in Section 4.2.3 that suggest this relationship is not all that strong in practice.

II. In general it will be possible to calculate the training margin but not the test error rate. If a relationship between the test error rate and training margin can be established this would allow us to predict the test error rate.

4.1.3 A Bound on the Training Margin

Schapire et al. also prove a bound on the training margin, when using AdaBoost.

Theorem 10 *(Schapire et al., 1997) Suppose the Base Classifier, when called by AdaBoost, generates classifiers with weighted training errors* $\epsilon_1, \ldots, \epsilon_B$. *Then for any* θ, *we have that*

$$P_S(M(X) \leq \theta) \leq 2^B \prod_{i=1}^{B} \sqrt{\epsilon_i^{1-\theta}(1-\epsilon_i)^{1+\theta}}$$

Further more, if, for all i, $\epsilon_i \leq 1/2 - \gamma$, *for some* $0 < \gamma \leq 1/2$, *then the bound simplifies to :*

$$P_S(M(X) \leq \theta) \leq \left(\sqrt{(1-2\gamma)^{1-\theta}(1+2\gamma)^{1+\theta}}\right)^B$$

Provided $\theta < \gamma$ *the expression inside the parentheses is less than* 1 *so that* $P_S(M(X) \leq \theta)$ *will decrease to zero exponentially fast in* B.

See Schapire et al., 1997 for the proof.

Together Theorems 9 and 10 provide strong motivation for the success of AdaBoost. Theorem 9 suggests that as $P_S(M(X) \leq \theta)$ decreases so will the test error rate and Theorem 10 shows that AdaBoost works to drive $P_S(M(X) \leq \theta)$ to 0.

4.1.4 Some Problems

There are at least two potential problems with the theory as it stands.

I. Theorem 9 only provides a bound on the expected test error rate in terms of the training margin. This means that there is no guarantee that reducing $P_S(M(X) \leq \theta)$ will cause a reduction in the test error rate.

II. Theorem 10 relies on the weighted training errors , ϵ_i, being bounded away from 1/2. This may not be the case in practice because AdaBoost tends to concentrate on more difficult points so the error rate may correspondingly increase.

Of these two problems the former is the potentially more serious one. In practice the error term in the bound can be very large, often greater than 1. This means that the bound is not even tight so, in theory, it is perfectly possible for $P_S(M(X) \leq \theta)$ to decrease but for the error rate to remain constant or even to increase!

4.2 How Well Does the Margin Bound Work?

We have seen in the previous section that the bound proved in Theorem 9 implies, but does not guarantee, a relationship between the test error rate and $P_S(M(X) \leq \theta)$. In this section we present experimental results to evaluate this relationship on *real data*.

4.2.1 The Schapire Model

Recall that the bound proved in Theorem 9 is of the form

$$\text{Expected text error rate} \leq P_S(M(X) \leq \theta) + O(\cdot)$$

We know that in practice the bound is not tight because $O(\cdot)$ is often large. However, the implication drawn in Schapire's paper is that even though the bound is not tight it still "gives correct qualitative predictions for the behaviour of the test error rate". This suggests the following relationship between the expected test error rate and the training margin.

$$E(\text{Test Error}|\mathcal{T}) = P_S(M(X) \leq \theta) + C_\theta \tag{4.1}$$

We call this the *Shapire Model*. The model has two unknown parameters i.e. θ and C_θ. If the model is correct, it states that although the bound is not tight there is a constant difference between the test error rate and $P_S(M(X) \leq \theta)$ so that as $P_S(M(X) \leq \theta)$ decreases the test errors will decrease at the same rate. The model implies that the objective in producing a classifier is to minimize $P_S(M(X) \leq \theta)$ for some positive value of θ.

4.2.2 The Training Model

Notice that a special case of the Schapire Model is achieved by setting $\theta = 0$. Since

$$P_S(M(X) \leq 0) = \text{Training error rate}$$

the Schapire Model reduces to :

$$E(\text{Test Error}|\mathcal{T}) = \text{Training error rate} + C_0 \tag{4.2}$$

We call (4.2) the *Training Model*. It is a one parameter model with C_0 being the only parameter. This is probably the simplest model you can imagine. It states that to reduce the test error rate we want to produce a classifier with low training error rate. We can think of this as the Null model.

If the bound in Theorem 9 does "give correct qualitative predictions for the behaviour of the test error rate" then we would expect the Schapire Model to produce significantly better predictions for the test error rate than using the Training Model. If this is not the case it would cast severe doubt on the practical usefulness of this bound.

4.2.3 An Experimental Comparison

In order to evaluate the accuracy of the two models introduced in Sections 4.2.1 and 4.2.2 an experimental study was performed. For the experiment we used the AdaBoost classifier on the Letter data set (see Section 2.2.2). From a randomly chosen training data set the test error rate was calculated for B between 1 and 100. Likewise the margin for each training data point was recorded for the same values of B. This allowed the training error rate to be computed for each value of B as well as $P_S(M(X) \leq \theta)$ for all θ and B. The aim was to produce the best fit of each model by choosing the parameters to minimize the squared difference between the test error rate and the model prediction over the 100 values of B.

Figure 4.1 shows a plot of the test error rate on this data set vs B. The red line is a smoothed version of the test error to give a clearer picture of the underlying trend. Figure 4.2 shows the best fit of the Training Model to the test data (also smoothed) i.e. choosing C_0 to minimize the squared discrepancy. It is clear that there are some significant departures of this model from the truth. The model vastly over predicts the error rate for small values of B. It then declines much faster than it should and levels out while the error rate is still declining. As one might expect, there is some relationship between training error and test error but it is not very strong.

Figure 4.3 is a plot of the best fit of the Shapire model to the test data (smoothed) i.e. choosing C_θ and θ to minimize the squared discrepancy. Unfortunately this model seems to possess many of the same problems as the Training Model. It is still over predicting errors for low values of B, declining too fast and then leveling off long before the test error rate

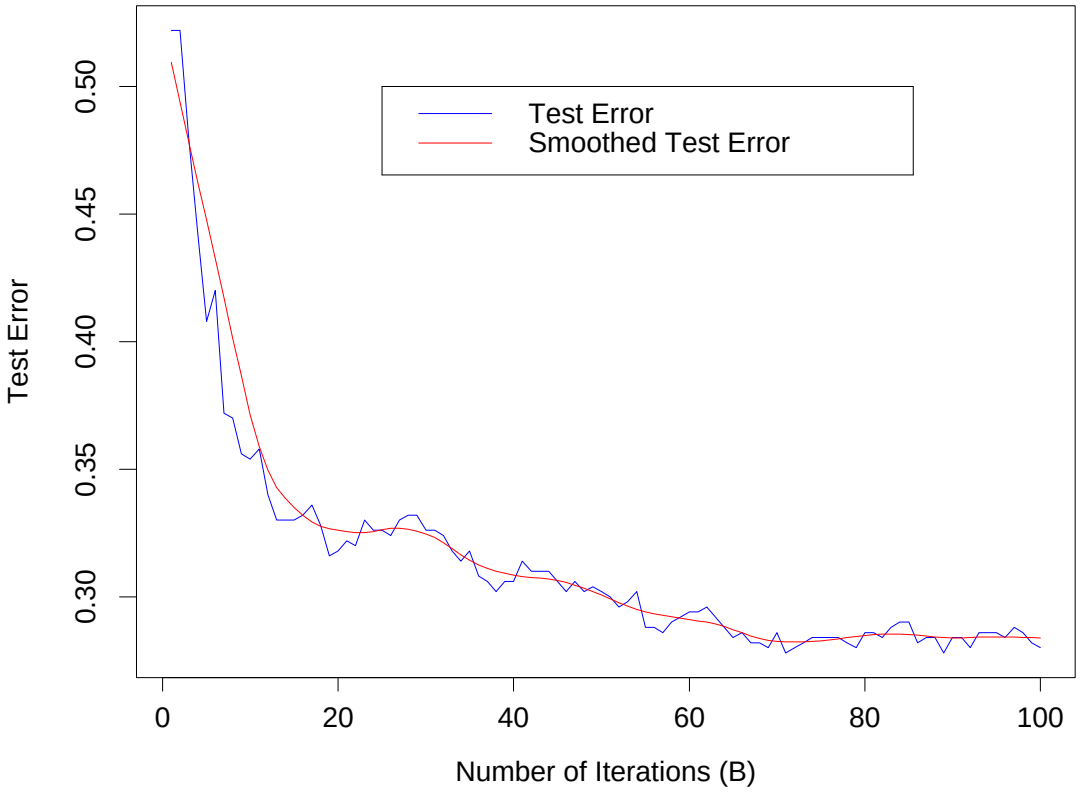

Figure 4.1: The actual test error rate and smoothed test error rate on the Letter data set.

does. Overall the fit looks a little better but it certainly doesn't seem to be a significant improvement over our null model of just using the training error.

In practice it seems that not only is the bound on the test error rate not tight, but it does not reflect the true behaviour of the test error!

4.3 The Normal Model

We have seen in Section 4.2.3 that in practice the test bound does not seem to match the actual behaviour of the test error rate which casts doubt on Schapire's hypothesis. However, the Schapire Model suggested by the bound in Theorem 9 is only one, fairly restrictive, use

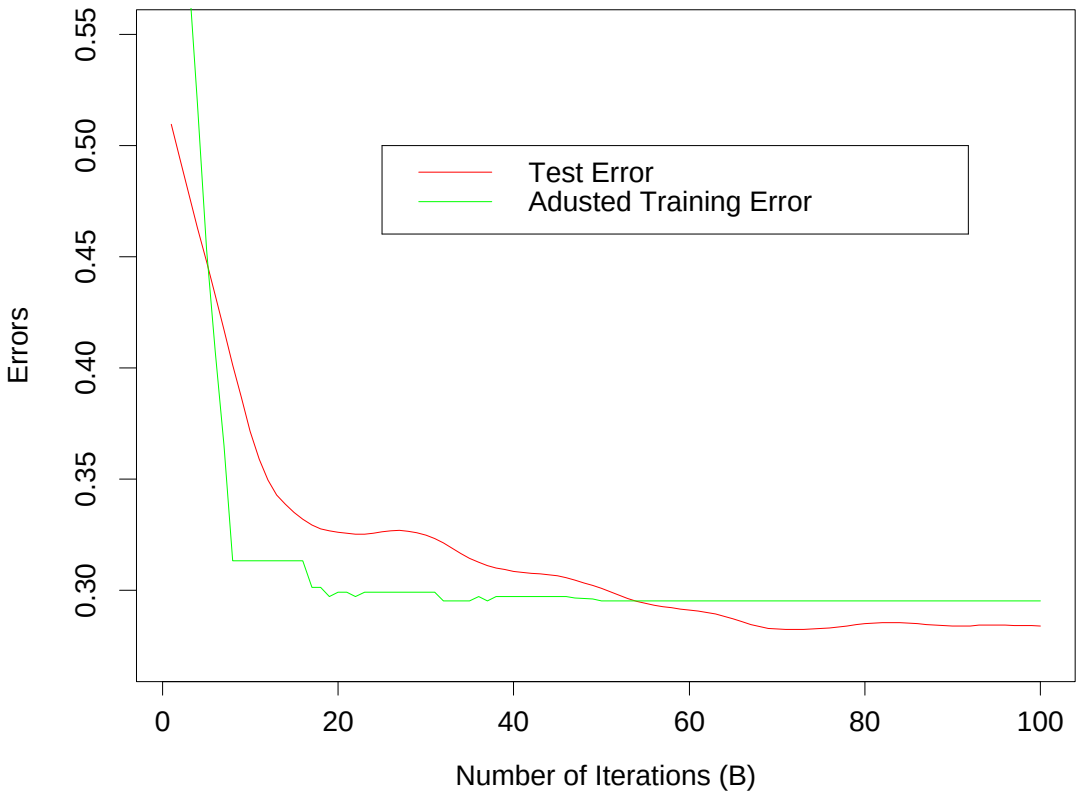

Figure 4.2: The test error rate and predictions from the Training Model, both smoothed.

of the training margin in predicting the test error rate. If we were to believe the Schapire Model it would suggest that there was some critical value, θ. If the margin of a training point is below that value it will increase the test error but if it is above θ then it will not. In reality this seems unlikely. It seems far more likely that there is some continuous mapping of the training margin distribution to the test error rate.

4.3.1 Developing the Normal Model

Let us more carefully examine the relationship betwee the expected test error rate and the margin. In order to simplify the notation, suppose we have n fixed test points $(x_1, g_1), \ldots,$

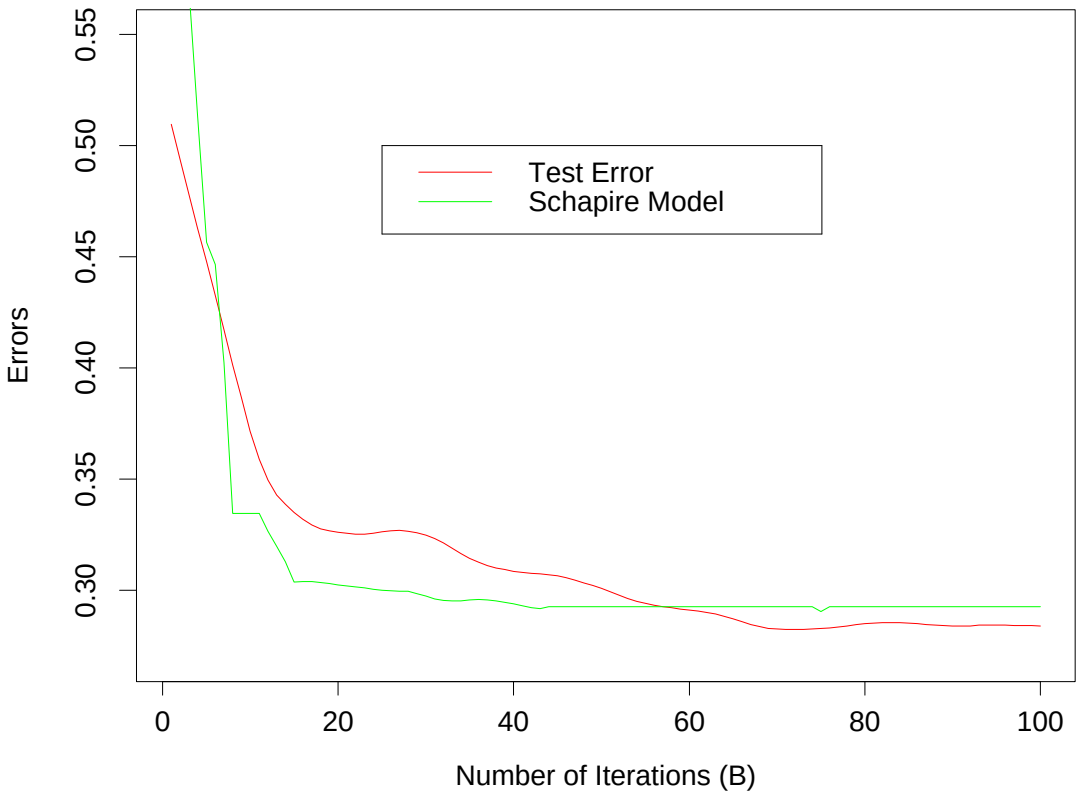

Figure 4.3: The test error rate and predictions from the Schapire Model, both smoothed.

(x_n, g_n). Then the expected test error is :

$$E(\text{Test Error}|\mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n} P(C(x_i) \neq g_i|\mathcal{T})$$

where the expectation and probability are over the randomness from the classifier. For example Bagging is a random classifier because it bootstraps the training data. Now let

$$M_B(x_i) = \text{margin at } x_i \text{ after } B \text{ iterations.}$$

Then

$$C(x_i) \neq g_i \quad \text{iff} \quad M_B(x_i) \leq 0$$

So

$$E(\text{Test Error}|\mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n} P(M_B(x_i) \leq 0|\mathcal{T})$$

If we knew the distribution of $M_B(x_i)$ we could, in principle, calculate the test error rate exactly. Of course in general this is a difficult problem but it is possible to rewrite the margin as an average of random variables.

$$M_B(x_i) = \frac{1}{B}\sum_{j=1}^{B} \delta_j(x_i)$$

where

$$\delta_j(x_i) = \begin{cases} 1 & \text{if the jth classifier classifies to the correct class} \\ -1 & \text{if the jth classifier classifies to the most popular of the other classes} \\ 0 & \text{otherwise} \end{cases}$$

For certain classifiers these random variables will be *iid*, conditional on the training data. A couple of examples are the Bagging and Substitution PICTs. In this case an application of the Central Limit Theorem tells us that

$$\sqrt{B}(M_B(x_i) - \mu_i) \Rightarrow N(0, \alpha_i) \tag{4.3}$$

where $\mu_i = E\delta_1(x_i)$ and $\alpha_i = Var[\delta_1(x_i)]$.

Even when the *iid* assumption is not met it often seems to be the case that the margin converges to a normal random variable. For example, experimental studies indicate that the margin from AdaBoost also converges to a normal distribution.

When (4.3) holds we know that :

$$E(\text{Test Error}|\mathcal{T}) \to \frac{1}{n}\sum_{i=1}^{n} \Phi\left(\frac{-\mu_i}{\sqrt{\alpha_i/B}}\right)$$

If one assumes that B is large enough, so that we are close to convergence, and α_i is approximately equal for all i, this leads to an alternative use of the margin to predict the

test error rate which we call the Normal Model.

$$E(\text{Test Error}|\mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n}\Phi\left(\frac{-\mu_i}{\sqrt{\alpha/B}}\right) \tag{4.4}$$

4.3.2 Relating the Training and Test Margins

Of course this model involves $\mu_i \quad i = 1, \ldots, n$ which are the means of the margins at each of the test points and in general these will be unknown. However, it is possible to use the margins at the training points to predict the means of the test margins.

There are many ways of doing this. For example, one might imagine that the margin of the test point x_i is related to the margin of the closest training point in the predictor space. We will denote the closest training point to x_i as y_i and the distance between them as $d_i = ||x_i - y_i||$. If d_i is small we would expect the training and test margins to be very similar but if d_i is comparatively large we would expect the test margin to be lower because it has not been used to train our classifier. So we could use the following model :

$$\mu_i = E[M(x_i)] = M(y_i) + \beta_0 - \beta_1 d_i$$

where β_1 is positive. However this model does not take into account the fact that we would expect a much larger decline in the test margin if $M(y_i)$ is close to 1 and a much smaller decline if $M(y_i)$ is close to -1. So instead we use the following model :

$$E\left(\frac{M(x_i) - M(y_i)}{M(y_i) + 1}\right) = \beta_0 - \beta_1 d_i \tag{4.5}$$

which gives the following estimate for the mean of the test margin

$$\hat{\mu}_i = M(y_i) + \hat{\beta}_0(1 + M(y_i)) - (\hat{\beta}_1 + \hat{\beta}_1 M(y_i))d_i \tag{4.6}$$

β_0 and β_1 are assumed constant for the entire test set and $\hat{\beta}_0$ and $\hat{\beta}_1$ are their least squares estimates. Thus the final form of the Normal Model is :

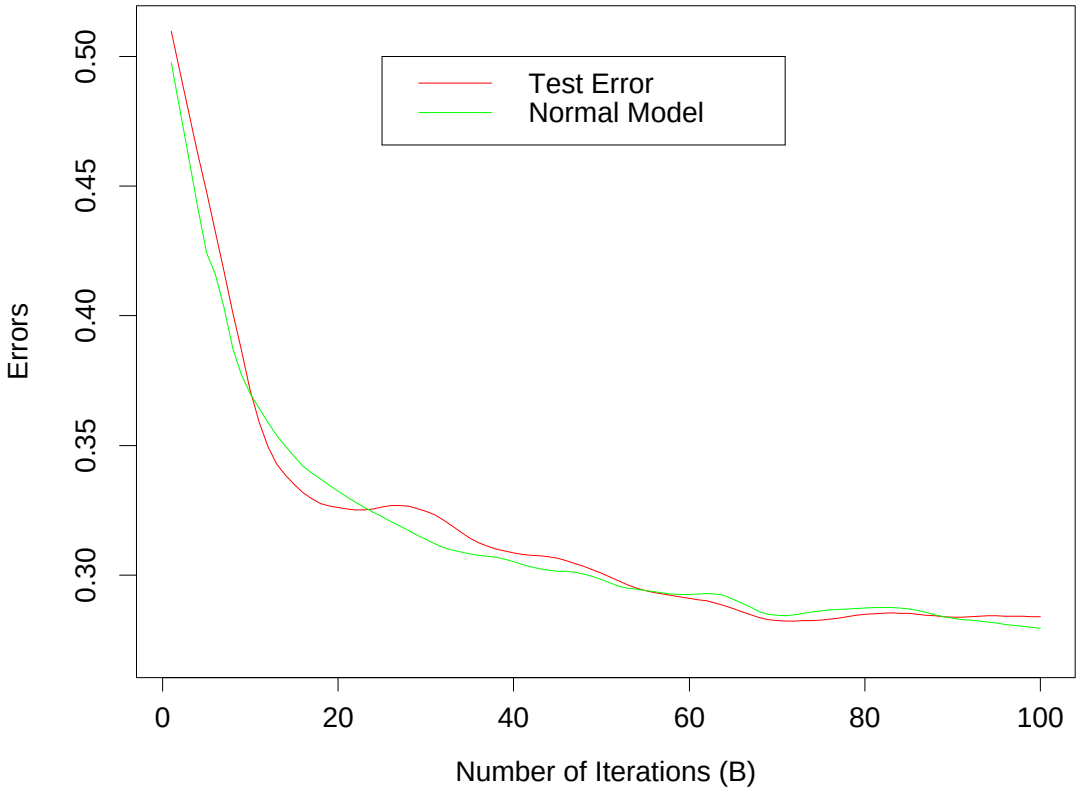

Figure 4.4: The test error rate and predictions from the Normal Model, both smoothed.

$$E(\text{Test Error}|\mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n}\Phi\left(\frac{-\hat{\mu}_i}{\sqrt{\alpha/B}}\right) \tag{4.7}$$

This model involves three unknown parameters, α, β_0 and β_1 as opposed to the Schapire Model which involved estimating two unknown parameters, θ and C_θ. Just as with the Schapire Model it is possible to optimize over these parameters to produce the best fit, in squared error terms, to the true test data. Figure 4.4 illustrates the best fit (smoothed). It is clear that this model is mapping the true test error with much more accuracy than either the Training Model or the Schapire Model even though only one extra parameter has been used.

4.3.3 Implications of the Normal Model

If we believe that the Normal Model is correct there are some immediate implications. Recall

$$E(\text{Test Error}|\mathcal{T}) = \frac{1}{n}\sum_{i=1}^{n}\Phi\left(\frac{\mu_i}{\sqrt{\alpha/B}}\right)$$

Now we can examine the effect on the expected test error rate as we change the margin at any particular test point by differentiating with respect to μ_i.

$$\begin{aligned}\frac{\partial E(\text{Test Error}|\mathcal{T})}{\partial \mu_i} &= -\sqrt{\frac{B}{\alpha}}\frac{1}{n}\phi\left(\frac{\mu_i}{\sqrt{\alpha/B}}\right) \\ &= -\sqrt{B}c_1\exp(-Bc_2\mu_i^2)\end{aligned}$$

where c_1 and c_2 are positive constants.

This means that :

I. The error rate will always decrease as we increase μ_i.

II. Increasing μ_i will cause a much larger decrease in the error rate if μ_i is close to zero rather than close to 1 or -1.

III. Increasing μ_is that are close to zero and decreasing μ_is that are close to 1 by the same amount (so the average margin is unchanged) will still reduce the error rate.

This last result is interesting because it has been demonstrated empirically that while AdaBoost works well at increasing low margins it tends to compensate by decreasing margins that are close to one (see Schapire et al., 1997). These results would explain why that is a good trade off to achieve in practice.

4.4 Conclusion

4.4.1 The Schapire Theories

Schapire et al.'s results proving bounds on the test error rate and also the training margin are very interesting. They have opened up a whole new approach in the search to understand the behaviour of MaVLs. However, they also have limitations. The largest of these limitations is

the lack of a tight bound on the test error. This means that, in theory at least, the behaviour of the test error rate does not need to match that suggested by the bound. In Section 4.2.3 we saw that in practice the test error rate does not seem to match the behaviour suggested by the bound.

4.4.2 The Normal Theory

In Section 4.3 it became apparent that it is possible to use the training margin to model the test error rate by using the Normal Model. This model seems to possess the necessary flexibility to match the behaviour of the test error without involving too many parameters. In Section 4.3.3 it was shown that the model also provides some explanations for the success of AdaBoost. However, there are also problems with the Normal Model :

I. While experimental observation seems to validate the model, there is a limited amount of theory to back it up. For many classifiers the model (4.4) will be asymptotically correct. However, the relationship between test margins and training margins given in (4.5) has no strong theoretical justification, even though empirically it seems to work well. It is an open problem as to whether a relationship similar to (4.5) can be proved. If so this would provide more theoretical motivation for the Normal Model.

II. The model implies that a decrease in

$$E\left[\Phi\left(\frac{-\mu_i}{\sqrt{\alpha/B}}\right)\right] \tag{4.8}$$

will cause a decrease in the test error rate. However, there is no theory to guarantee that a MaVL will cause (4.8) to decrease. The bound proved in Theorem 10 is not enough to guarantee that AdaBoost will reduce (4.8).

Despite these problems it seems clear that the Normal Model has potential which deserves further exploration.

4.4.3 Other Modern Theories

The *Modern* theories are still at an early stage. Breiman has written a couple of papers (Breiman, 1997 and Breiman, 1998) with a similar approach to that of Schapire's. He defines a quantity which he calls the *Edge*. The edge is equivalent to the margin for a two

class case but is slightly different for larger numbers of classes.

Breiman, 1997 also produces empirical results which seem to cast doubt on the usefulness, in practice, of Schapire's theories.

4.5 Thesis Summary and Conclusion

In Chapter 2 we surveyed a number of MaVLs and PICTs. It was clear from Section 2.6 that these classifiers can often produce significant reductions in error rates. We also provided motivation for the ECOC PICT in terms of an approximation to the Bayes Classifier. However, no explanation was given as to why this should be a good approximation or indeed why any of these classifiers should work as well as they do. Chapters 3 and 4 are devoted to attempting to answer this question.

In Chapter 3 an approach involving generalizations of bias and variance to classification problems is used. This method has a great deal of intuition to statisticians and potentially allows the use of the mountain of work that has been produced for regression problems. Unfortunately it seems that when an attempt is made to generalize bias and variance beyond squared error loss most of the regression results fail to generalize. In particular there is no clear decompostition of the prediction error into functions of bias and variance.

In Chapter 4 an alternative approach is used. Here a quantity called the margin is defined and bounds on the test error rates are proved in terms of the training margin. Based on these bounds a theory is developed to explain why AdaBoost produces reductions in error rates. Unfortunately the bounds are not tight and empirical results cast doubt on their practical usefulness. Another use of margins is also suggested which we call the Normal Model. This approach has the advantage that it appears to match the behaviour of the test error rate with high accuracy. However, it has less theoretical motivation.

Both the Classical and Modern theories produce useful insights into the success of MaVLs. However, it is our belief that, no individual approach provides a comprehensive explanation. It is still an open question as to why these methods work so well.

Appendix A

Theorems and Proofs

Lemma 1 *If one uses a deterministic coding matrix and the Bayes Classifier as the Base Classifier then*

$$L_i = \sum_{l \neq i} q_l \sum_{j=1}^{B} (Z_{lj} - Z_{ij})^2 \quad i = 1, \ldots, k$$

Proof

First note that

$$\hat{p}_j = \sum_{l=1}^{K} q_l Z_{lj}$$

when we use the the Bayes Classifier as the Base Classifier.

Now

$$\begin{aligned}
L_i &= \sum_{j=1}^{B} |\hat{p}_j - Z_{ij}| \\
&= \sum_{j=1}^{B} (Z_{ij}(1 - \hat{p}_j) + (1 - Z_{ij})\hat{p}_j) \\
&= \sum_{j=1}^{B} \left(Z_{ij}(1 - \sum_{l=1}^{K} q_l Z_{lj}) + (1 - Z_{ij}) \sum_{l=1}^{K} q_l Z_{lj} \right) \\
&= \sum_{j=1}^{B} \left(Z_{ij}(1 - q_i - \sum_{l \neq i} q_l Z_{lj}) + (1 - Z_{ij}) \sum_{l \neq i} q_l Z_{lj} \right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{B} \left(Z_{ij}(\sum_{l \neq i} q_l - \sum_{l \neq i} q_l Z_{lj}) + (1 - Z_{ij}) \sum_{l \neq i} q_l Z_{lj} \right) \\
&= \sum_{l \neq i} q_l \sum_{j=1}^{B} [Z_{ij} - 2 Z_{ij} Z_{lj} + Z_{lj}] \\
&= \sum_{l \neq i} q_l \sum_{j=1}^{B} (Z_{lj} - Z_{ij})^2 \quad \text{since } Z_{ij} = Z_{ij}^2
\end{aligned}
$$

Theorem 1 *The ECOC PaCT is Bayes Consistent iff the Hamming distance between every pair of rows of the coding matrix is equal.*

To prove this theorem we need to make use of the following Lemma.

Lemma 2 *Let* $\mathbf{d} = A\mathbf{q}$. *Consider the following situation*

$$\arg\min_i d_i = \arg\max_i q_i \quad \forall q_i \tag{A.1}$$

(A.1) will hold iff A is of the form

$$A \quad = \quad \begin{bmatrix} a_1 & a_2 + b & \cdots & a_k + b \\ a_1 + b & a_2 & \cdots & a_k + b \\ \vdots & \vdots & \ddots & \vdots \\ a_1 + b & a_2 + b & \cdots & a_k \end{bmatrix} \quad = \quad 1(\mathbf{a} + b)^T - bI$$

where b is a positive scalar and $\mathbf{a} = (a_1, a_2, \ldots, a_k)^T$ *is a vector of the diagonal elements of A.*

Proof of Lemma

First note that we can assume without loss of generality that every diagonal element of A is zero because subtracting a_i from the ith column of A simply subtracts $a_i q_i$ from every element of d and leaves (A.1) unaffected.

$(\Rightarrow) \quad \mathbf{d} = A\mathbf{q} = 1\underbrace{(a^T\mathbf{q} + b^T\mathbf{q})}_{scalar} - b\mathbf{q}$

$\therefore \quad d_i = \text{constant} - bq_i$

So (A.1) holds.

$(\Leftarrow)$ Now suppose (A.1) holds

Consider first the situation with $k = 2$

So we get

$$\begin{aligned} a_{12}q_2 &= d_1 \\ a_{21}q_1 &= d_2 \end{aligned}$$

If a_{12} is negative then (A.1) does not hold (eg $q_1 = 0$ and $q_2 = 1$ will violate (A.1)). Likewise if a_{21} is negative (A.1) will not hold. Also if a_{12} and a_{21} are both positive but $a_{12} > a_{21}$ then (A.1) does not hold (eg $q_1 = 1$ and $q_2 = 1/2(1 + \frac{a_{21}}{a_{12}})$). And similarly for $a_{12} < a_{21}$. Therefore (A.1) implies $a_{12} = a_{21} > 0$ so the lemma holds for $k = 2$

Next consider the situation with $k = 3$
We can reduce this back to the case $k = 2$ by setting one of the q's to zero e.g.

$$\begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ 0 \\ q_3 \end{bmatrix} = \begin{bmatrix} 0 & a_{13} \\ a_{31} & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_3 \end{bmatrix}$$

and we know for $k = 2$ the off diagonals are equal and positive so A must be positive and symmetric of the form

$$\begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{12} & 0 & a_{23} \\ a_{13} & a_{23} & 0 \end{bmatrix}$$

However we also know that the row sums must be equal. To see this note that if not by setting $q_i = 1$ for all i we get d_i not all equal. By taking the maximum d_i and adding ϵ to q_i we find that q_i is now the maximum but d_i is not the minimum! Therefore (A.1) implies the row sums must be equal.

Hence $a_{12} + a_{13} = a_{12} + a_{23} = a_{13} + a_{23}$ which means $a_{12} = a_{13} = a_{23} > 0$ so (A.1) holds for $k = 3$.

We are now ready to consider the situation for general k

First note that we can reduce any general k matrix to one with $k = 3$ by setting all but three of the q's to zero. So for example we can reduce A to the upper 3 by 3 matrix by setting $q_i = 0$ for $i > 3$. So from our previous work we know $a_{12} = a_{13} = a_{21} = a_{23} = a_{31} = a_{32} > 0$. Now we can repeat the same process by setting $q_1 = 0$ and $q_i = 0$ for $i > 4$ this gives us the 3 by 3 matrix formed from the 2'nd, 3'rd and 4'th rows and columns. So again we know $a_{23} = a_{24} = a_{32} = a_{34} = a_{42} = a_{43} > 0$. But $a_{12} = a_{13} = a_{23}$ etc so we know that all the elements in the first four rows and columns are equal and positive. By repeating this process for all combinations of rows we can see that every element on the off diagonal must be equal and positive.

Thus we have proved the Lemma.

Now we are ready to prove the theorem

Proof of Theorem

From the Lemma 1 we know

$$L_i = \sum_{l \neq i} q_l \sum_{j=1}^{B} (Z_{lj} - Z_{ij})^2$$

when we use the Bayes Classifier as the Base Classifier. So

$$L = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1K} \\ a_{21} & 0 & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K1} & a_{K2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_K \end{bmatrix}$$

where a_{il} is equal to $\sum_{j=1}^{B} (Z_{lj} - Z_{ij})^2$
Now from Lemma 2 we know that the ECOC PICT is Bayes Consistent iff a_{il} are equal and positive for all i, l. But note that $\sum_{j=1}^{n} (Z_{lj} - Z_{ij})^2$ is equal to the Hamming distance between row i and l of the coding matrix so we are done.

Theorem 2 *Suppose that*

$$E_{\mathcal{T}}[\hat{p}_j \mid Z, X] = \sum_{i=1}^{k} Z_{ij} q_i = \mathbf{Z}^{j^T} \boldsymbol{q} \quad j = 1, \ldots, B \tag{A.2}$$

Then under this assumption for a randomly generated coding matrix

$$E_{\mathcal{T},Z} \bar{D}_i = q_i \quad i = 1, \ldots, k$$

Proof

Let $D_{ij} = 1 - 2|\hat{p}_j - Z_{ij}|$ then

$$\bar{D}_i = \frac{1}{B} \sum_{j=1}^{B} D_{ij}$$

We only need to prove $E_{\mathcal{T},Z} D_{ij} = q_i$.

$$\begin{aligned} E_{\mathcal{T},Z} D_{ij} &= 1 - 2E_{\mathcal{T},Z} \mid \hat{p}_j - Z_{ij} \mid \\ &= 1 - 2E_{\mathbf{Z}_{l \neq i}}[E_{\mathcal{T},\mathbf{Z}_i}(\mid \hat{p}_j - Z_{ij} \mid \mid Z_{lj} : l \neq i)] \end{aligned}$$

but

$$\begin{aligned} & E_{\mathcal{T},\mathbf{Z}_i}(\mid \hat{p}_j - Z_{ij} \mid \mid Z_{lj} : l \neq i) \\ = \; & \frac{1}{2} E_{\mathcal{T}}(\hat{p}_j \mid Z_{ij} = 0, Z_{lj} : l \neq i) + \frac{1}{2} E_{\mathcal{T}}(1 - \hat{p}_j \mid Z_{ij} = 1, Z_{lj} : l \neq i) \qquad \text{by iid} \\ = \; & \frac{1}{2} \sum_{l \neq i} Z_{lj} q_l + \frac{1}{2}(1 - q_i - \sum_{l \neq i} Z_{lj} q_l) \qquad \text{from } (A.2) \\ = \; & \frac{1}{2}(1 - q_i) \end{aligned}$$

So

$$E_{\mathcal{T},Z} D_{ij} = 1 - 2(\frac{1}{2}(1 - q_i)) = q_i$$

Theorem 3 *Suppose that* $\arg\max_i \mu_i$ *is unique i.e. there are no ties in the* μ*s. Then for a random coding matrix, conditional on* $\mathcal{T}$*, the following results hold for any Base Classifier.*

I.

$$\sqrt{B}(\bar{D}_i - \mu_i) \Rightarrow N(0, \sigma_i^2) \quad i = 1, \ldots, k$$

II.

$$\bar{D}_i \to \mu_i \quad a.s. \quad i = 1, \ldots, k$$

III.

$$\lim_{B\to\infty} \arg\max_i \bar{D}_i = \arg\max_i \mu_i \quad a.s.$$

Proof

I. Conditional on $\mathcal{T}$, $\bar{D}_i$ is an average of iid random variables with mean μ_i and finite variance. Therefore the Central Limit Theorem gives us the result.

II. Conditional on $\mathcal{T}$, $\bar{D}_i$ is an average of iid random variables with mean μ_i and finite variance. Therefore the Strong Law of Large Numbers gives us the result.

III. Let A_i be the set of ω such that $\bar{D}_i(\omega) \to \mu_i$. From the previous theoem we know $P(A_i) = 1$. Let $A = A_1 \cap \cdots \cap A_K$. Then $P(A) = 1$ by basic prob theory results. Assume WLOG that μ_1 is the unique maximum μ and that $\mu_1 - \mu_i > \epsilon$ for all $i \neq 1$. Then for any $\omega \in A$ there exists B_0 such that for all $B > B_0$ $|\bar{D}_i(\omega) - \mu_i| < \epsilon/2$ for all i and hence $\arg\max_i \bar{D}_i(\omega) = \arg\max_i \mu_i$. Therefore since $P(A) = 1$ the result is proved.

Theorem 4 *When the Bayes Classifier is used as the Base Classifier*

$$\mu_i = q_i \tag{A.3}$$

Proof

To prove the above theorem we only need to prove

$$E_Z[D_{ij}|\mathcal{T}] = q_i$$

when we use the Bayes Classifier as the Base Classifier. First note that when we use the Bayes Classifier

$$\hat{p}_j = \sum_{i=1}^{k} q_i Z_{ij} \tag{A.4}$$

Now

$$\begin{aligned} E_Z[D_{ij}|\mathcal{T}] &= 1 - 2E_Z \mid \hat{p}_j - Z_{ij} \mid \\ &= 1 - 2E_{\mathbf{Z}_{l\neq i}}[E_{\mathbf{Z}_i}(\mid \hat{p}_j - Z_{ij} \mid \mid Z_{lj} : l \neq i)] \end{aligned}$$

but

$$\begin{aligned} & E_{\mathbf{Z}_i}(\mid \hat{p}_j - Z_{ij} \mid \mid Z_{lj} : l \neq i) \\ = \; & \frac{1}{2}E_{\mathbf{Z}_i}(\hat{p}_j \mid Z_{ij} = 0, Z_{lj} : l \neq i) + \frac{1}{2}E_{\mathbf{Z}_i}(1 - \hat{p}_j \mid Z_{ij} = 1, Z_{lj} : l \neq i) \qquad \text{by iid} \\ = \; & \frac{1}{2}\sum_{l\neq i} Z_{lj}q_l + \frac{1}{2}(1 - q_i - \sum_{l\neq i} Z_{lj}q_l) \qquad \text{from (A.4)} \\ = \; & \frac{1}{2}(1 - q_i) \end{aligned}$$

So

$$E_Z[D_{ij}|\mathcal{T}] = 1 - 2(\frac{1}{2}(1 - q_i)) = q_i$$

Corollary 1 is a consequence of Theorems 3 and 4.

Theorem 5 *If the coding matrix is randomly chosen then, conditional on $\mathcal{T}$, for any fixed X*

$$\begin{aligned} |\textit{ECOC error rate} - \textit{Limiting error rate}| &\leq Pr_Z(\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i|\mathcal{T}) \\ &\leq (k-1)e^{-mB} \end{aligned}$$

where $m = (\mu_{(k)} - \mu_{(k-1)})/8$ and $\mu_{(i)}$ is the ith order statistic.

Proof

$$\begin{aligned} & |\text{ECOC error rate} - \text{Limiting error rate}| \\ = \; & |E_Z[I\{\arg\max_i \bar{D}_i \neq Y\}|\mathcal{T}] - E_Z[I\{\arg\max_i \mu_i \neq Y\}|\mathcal{T}]| \\ \leq \; & E_Z[I\{\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i\}|\mathcal{T}] \\ = \; & Pr_Z(\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i|\mathcal{T}) \end{aligned}$$

Assume WLOG that $\arg\max_i \mu_i = 1$. Then

$$\begin{aligned} Pr(\arg\max_i \bar{D}_i \neq \arg\max_i \mu_i) &= 1 - Pr(\bar{D}_1 > \bar{D}_2, \cdots, \bar{D}_1 > \bar{D}_K) \\ &\leq \sum_{i=2}^{k} Pr(\bar{D}_1 < \bar{D}_i) \end{aligned}$$

So we only need to show $Pr(\bar{D}_1 < \bar{D}_i) \leq e^{-mB}$ for $i = 2, \ldots, k$ but

$$\begin{aligned} Pr(\bar{D}_1 < \bar{D}_i) &= Pr(\bar{D}_i - \bar{D}_1 - (\mu_i - \mu_1) > (\mu_1 - \mu_i)) \\ &= Pr(\frac{1}{B}\sum_{j=1}^{B}(D_{ij} - D_{1j}) - (\mu_i - \mu_1) > \mu_1 - \mu_i) \\ &\leq e^{-B(\mu_1 - \mu_i)/8} \quad \text{by Hoeffding's inequality} \\ &\leq e^{-mB} \end{aligned}$$

where $m = (\mu_{(k)} - \mu_{(k-1)})/8$. The second to last line follows becuase $-2 \leq D_{ij} - D_{1j} \leq 2$. See Theorem 2, page 16 of Hoeffding, 1963 for further details.

Corollary 2 follows directly from Theorems 4 and 5.

Theorem 6 *The Regression PICT is Bayes Consistent for any coding matrix, provided* ZZ^T *is invertible. In other words if the Base Classifier is producing perfect two class probability estimates the Regression PICT will produce perfect* k *class probability estimates.*

Proof

First note that when we use the Bayes Classifier as the Base Classifier

$$\hat{\mathbf{p}} = Z^T \mathbf{q}$$

The Regression PICT classifies to $\arg\max_i \hat{q}_i$ where

$$\hat{\mathbf{q}} = (ZZ^T)^{-1} Z \hat{\mathbf{p}}$$

So showing that $\hat{\mathbf{q}} = \mathbf{q}$ is sufficient. But

$$\hat{\mathbf{q}} = (ZZ^T)^{-1} ZZ^T \mathbf{q} = \mathbf{q}$$

so provided ZZ^T is invertible the Regression PICT is Bayes Consistent.

Theorem 7 *Suppose that p_{ij} is independent from $\mathbf{Z}^j$ (the jth column of Z), for all i and j. In other words the distribution of p_{ij} conditional on $\mathbf{Z}^j$ is identical to the unconditional distribution. Then*

$$E_Z[p_i^S \mid \mathcal{T}] = E_Z[\bar{D}_i \mid \mathcal{T}] = \mu_i$$

Therefore as B approaches infinity the ECOC PICT and Substitution PICT will converge for any given training set; i.e. they will give identical classification rules.

Proof

Since

$$\bar{D}_i = \frac{1}{B}\sum_{j=1}^{B} D_{ij} \quad \text{and} \quad p_i^S = \frac{1}{B}\sum_{j=1}^{B} p_{ij}$$

we just need to show

$$E_Z[D_{ij}|\mathcal{T}] - E_Z[p_{ij}|\mathcal{T}] = 0$$

$$\begin{aligned}
& E_Z[D_{ij}|\mathcal{T}] - E_Z[p_{ij}|\mathcal{T}] \\
= \;& E_Z[1 - 2|\hat{p}_j - Z_{ij}| - p_{ij}|\mathcal{T}] \\
= \;& E_Z[1 - 2(1 - \text{proportion in same super group as } i \text{ for column } j) - p_{ij}|\mathcal{T}] \\
= \;& E_Z[p_{ij} + 2\sum_{l\neq i}^{k} I\{l \text{ in same super group as } i \text{ for column } j\}p_{lj} - 1|\mathcal{T}] \\
= \;& E_Z[p_{ij}|\mathcal{T}] + 2\sum_{l\neq i}^{k} E_Z[I\{l \text{ in same super group as } i \text{ for column } j\}p_{lj}|\mathcal{T}] - 1 \\
= \;& E_Z[p_{ij}|\mathcal{T}] + 2\sum_{l\neq i}^{k} E_Z[I\{l \text{ in same super group as } i \text{ for column } j\}|\mathcal{T}]E[p_{lj}|\mathcal{T}] - 1 \\
& \text{(by independence)} \\
= \;& E_Z[p_{ij}|\mathcal{T}] + 2\sum_{l\neq i}^{k} \frac{1}{2}E_Z[p_{lj}|\mathcal{T}] - 1 \\
= \;& E_Z[p_{ij}|\mathcal{T}] + (1 - E_Z[p_{ij}|\mathcal{T}]) - 1 \\
= \;& 0
\end{aligned}$$

Theorem 8 *Under the previously stated semi-parametric model assumptions (3.4) and (3.5) will hold iff*

$$\rho < \gamma \qquad (\rho \text{ is small relative to } \gamma) \tag{A.5}$$

and

$$B \geq \frac{1-\rho}{\gamma-\rho} \qquad (B \text{ is large enough}) \tag{A.6}$$

Further more if $k = 2$ (there are only 2 classes) then (A.5) and (A.6) are sufficient to guarantee a reduction in the error rate.

First we show that (A.5) and (A.6) hold iff (3.5) holds.

$$\begin{aligned}
Var(p_i^S) &= Var(\frac{1}{B}\sum_{j=1}^{B} p_{ij}) \\
&= \frac{1}{B^2}[\sum_{j=1}^{B} Var(p_{ij}) + \sum\sum_{j\neq l} Cov(p_{ij}, p_{il})] \\
&= \frac{1}{B}[Var(p_{i1}) + (B-1)Cov(p_{i1}, p_{i2})] \\
&= \frac{1}{B}Var(p_{i1})(1 + (B-1)\rho)
\end{aligned}$$

So

$$\begin{aligned}
\frac{\sigma_S}{\alpha_S} &\leq \frac{\sigma_T}{\alpha_T} \\
\Leftrightarrow Var(p_i^T)\frac{{\alpha_S}^2}{{\alpha_T}^2} &\geq Var(p_i^S) \\
\Leftrightarrow Var(p_i^T)\frac{{\alpha_S}^2}{{\alpha_T}^2} &\geq \frac{1}{B}Var(p_i^1)(1 + (B-1)\rho) \\
\Leftrightarrow \gamma &\geq \frac{1}{B}(1 + (B-1)\rho) \\
\Leftrightarrow B &\geq \frac{1-\rho}{\gamma-\rho} \\
&\text{and} \quad \rho < \gamma
\end{aligned}$$

Next we show that (3.5) holds iff (3.4) holds.

$$\begin{aligned}
CV(p_i^S - p_j^S) &= \sqrt{\frac{Var(p_i^S - p_j^S)}{(E(p_i^S - p_j^S))^2}} \\
&= \sqrt{\frac{Var(\alpha_S[f(q_i) - f(q_j)] + \sigma_S[\epsilon_i^S - \epsilon_j^S])}{(E(\alpha_S[f(q_i) - f(q_j)] + \sigma_S[\epsilon_i^S - \epsilon_j^S]))^2}} \\
&= \frac{\sigma_S}{\alpha_S}\sqrt{\frac{Var(\epsilon_i^S - \epsilon_j^S)}{(E(f(q_i) - f(q_j)))^2}} \\
&\leq \frac{\sigma_T}{\alpha_T}\sqrt{\frac{Var(\epsilon_i^T - \epsilon_j^T)}{(E(f(q_i) - f(q_j)))^2}} \quad \text{by (3.5) and equailty of error distributions.} \\
&= CV(p_i^T - p_j^T) \quad \text{(just work backwards)}
\end{aligned}$$

Last we show that (3.5) implies a lower error rate for a two class problem. To do this we first show that (3.5) implies $Pr(\arg\max p_i^S = \arg\max q_i) \geq Pr(\arg\max p_i^T = \arg\max q_i)$ and then note that for a two class problem this guarantees a lower error rate. It seems likely that even if $k > 2$ this condition will also cause a lower error rate.

Assume WLOG that $\arg\max q_i = 1$

$$\begin{aligned}
& Pr(\arg\max p_i^S = \arg\max q_i) \\
= \;& Pr(p_1^S > p_2^S, \ldots, p_1^S > p_k^S) \\
= \;& Pr(p_1^S - p_2^S > 0, \ldots, p_1^S - p_k^S > 0) \\
= \;& Pr(\alpha_S(f(q_1) - f(q_2)) + \sigma_S(\epsilon_1^S - \epsilon_2^S) > 0, \ldots, \alpha_S(f(q_1) - f(q_k)) + \sigma_S(\epsilon_1^S - \epsilon_k^S) > 0) \\
= \;& Pr(\epsilon_1^S - \epsilon_2^S > -(f(q_1) - f(q_2))\frac{\alpha_S}{\sigma_S}, \ldots, \epsilon_1^S - \epsilon_k^S > -(f(q_1) - f(q_k))\frac{\alpha_S}{\sigma_S}) \\
= \;& Pr(\epsilon_1^T - \epsilon_2^T > -(f(q_1) - f(q_2))\frac{\alpha_S}{\sigma_S}, \ldots, \epsilon_1^T - \epsilon_k^T > -(f(q_1) - f(q_k))\frac{\alpha_S}{\sigma_S}) \\
& \text{(because } \boldsymbol{\epsilon}^T \text{ and } \boldsymbol{\epsilon}^S \text{ have identical distributions)} \\
\geq \;& Pr(\epsilon_1^T - \epsilon_2^T > -(f(q_1) - f(q_2))\frac{\alpha_T}{\sigma_T}, \ldots, \epsilon_1^T - \epsilon_k^T > -(f(q_1) - f(q_k))\frac{\alpha_T}{\sigma_T}) \\
& \text{(provided (3.5) holds)} \\
= \;& Pr(\arg\max p_i^T = \arg\max q_i) \quad \text{(just work backwards)}
\end{aligned}$$

Bibliography

Breiman, L. (1996a). Bagging predictors. *Machine Learning 26, No. 2*: 123–140.

Breiman, L. (1996b). Bias, variance, and arcing classifiers. *Technical Report 460, Statistics Department, University of California Berkeley*.

Breiman, L. (1997). Arcing the edge. *Unpublished*.

Breiman, L. (1998). Prediction games and arcing algorithms. *Unpublished*.

Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regresion Trees*. Wadsworth.

Dietterich, T. and G. Bakiri (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research 2*: 263–286.

Dietterich, T. G. and E. B. Kong (1995). Error-correcting output coding corrects bias and variance. *Proceedings of the 12th International Conference on Machine Learning*: 313–321 Morgan Kaufmann.

Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. *Machine Learning : Proceedings of the Thirteenth International Conference*.

Friedman, J. (1996a). Another approach to polychotomous classification. *Technical Report, Department of Statistics, Stanford University*.

Friedman, J. (1996b). On bias, variance, 0/1-loss, and the curse of dimensionality. *Technical Report, Department of Statistics, Stanford University*.

Hastie, T. and R. Tibshirani (1994). Handwritten digit recognition via deformable prototypes. *Technical Report, AT&T Bell Labs*.

Hastie, T. and R. Tibshirani (1996). Classification by pairwise coupling. *Technical Report, Department of Statistics, Stanford University.*

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association.*

Kohavi, R. and D. Wolpert (1996). Bias plus variance decomposition for zero-one loss functions. *Machine Learning : Proceedings of the Thirteenth International Conference.*

Nilsson, N. J. (1965). *Learning Machines*. McGraw-Hill, New York.

Schapire, R., Y. Freund, P. Bartlett, and W. Lee (1997). Boosting the margin : A new explanation for the effectiveness of voting methods. *(available at http://www.research.att.com/∼yoav).*

Tibshirani, R. (1996). Bias, variance and prediction error for classification rules. *Technical Report, Department of Statistics, University of Toronto.*

Venables, W. and B. Ripley (1994). *Modern Applied Statistics with S-Plus*. 1st edn., Springer-Verlag.