Assessment of Statistical Classification Rules: Implications for Computational Intelligence

By
Waleed Ahmed Yousef
M.Sc. in Computer Science, November 1999, Helwan University
B.Sc. in Electrical Engineering, June 1995, Ain Shams University

in partial satisfaction of the requirements for the degree of
Doctor of Science

A Dissertation submitted to
the Faculty of
The School of Engineering and Applied Science
of The George Washington University

January, 31, 2006

Dissertation directed by
Murray H. Loew, Ph.D.,
Professor of Engineering and Applied Science
Robert F. Wagner, Ph.D.,
Senior Biomedical Research Scientist at CDRH, FDA.

Abstract

The problem of binary classification is of great interest across many fields, including data mining, satellite imaging, and medical diagnostics. The performance of a classifier is, mostly, measured in terms of the error rate, i.e., the total probability of misclassification. A more general approach is to use the Receiver Operating Characteristic (ROC). The ROC is a plot of all the possible values of one type of errors versus the other one. Very practical and easy-to-interpret summary measures can be derived from such a curve, e.g., the Area Under the Curve (AUC), and the Partial Area Under the Curve (PAUC).

This dissertation studies the assessment of classification rules using the entire ROC space with no parametric assumptions. That is, the present approach requires having no knowledge about the distribution of the data. In addition, when data are scarce the classification rule should be designed and assessed from the single available data set. In the present regulatory setting for public-policy making, e.g., in the area of medical diagnostics, the available data set is required to be split into two disjoint sets, one for design and the other for assessment. In this dissertation, both strategies are studied. Moreover, the techniques developed in this dissertation assume no particular form for the classification rule to be assessed: the methodology is general across classical as well as novel modern architectures. The linear and quadratic discriminants used in the dissertation were selected, simply, for demonstration purposes.

The contemporary use of the expression *Computational Intelligence* refers to a number of rapidly maturing branches of the general field of artificial intelligence, including neural networks, fuzzy logic, evolutionary algorithms ...etc. Algorithms developed in these subfields to solve classification problems are, from a statistical point of view, nonparametric classification rules. This dissertation may, therefore, provide critical assessment tools for such algorithms when they must be developed within a setting in which data are scarce.

Acknowledgements

All praises are due to Allah, Most Gracious, Most Merciful. *"And He is Allah: there is no god but He. To Him be praise, at the first, and at the last: For Him is the command, and to Him shall ye (all) be brought back"*[1].

All respect and honour should be raised to those parents who dedicated their life in raising the author of this dissertation. *"Thy Lord hath decreed that ye worship none but Him, and that ye be kind to parents. Whether one or both of them attain old age in thy life, say not to them a word of contempt, nor repel them, but address them in terms of honour. And, out of kindness, lower to them the wing of humility, and say: My Lord! bestow on them thy mercy even as they cherished me in childhood."*[2]. I, hereby, express my gratitude to my mother, who dedicated her life to her children and enjoyed nothing but affording pleasure and happiness to them. I, hereby, express my gratitude to my father, who planted, in me, the seeds of loving excellence and success since my earliest scholarly life; therefore, now, this is the harvest. May I be, to them, the kind son as they were, to me, the great parents.

I want to express my thanks and respect to my supervisor Dr. Robert Wagner. I think that I cannot tailor a better character to supervise myself than him. His availability, tolerance and flexibility, support and caring, respect showing, and encouragement are not common to be all gathered in a supervisor. In parallel to this, I would like to express my thanks to my supervisor Dr. Murray Loew. His *Pattern Recognition* class opened up this broad field for me. In addition, he was who introduced me to this project and recommended me to this powerful research group at the Division of Imaging and Applied Mathematics (DIAM)\The Office of Science and Engineering Laboratories (OSEL)\Center for Devices and Radiological Health (CDRH)\Food and Drug Administration (FDA), in which I enjoyed scientific research; this project was supported in part by a CDRH Medical Device Fellowship and administered by the Oak Ridge Institute for Science and Education (ORISE). I would like to take this opportunity to thank David Brown Ph.D. and Kyle Myers Ph.D., directors of DIAM, for their support and my fellows at DIAM for providing such a thriving and respectful environment.

[1] The Holy Qur'ān, 28:70

[2] The Holy Qur'ān, 17:23–24

Dedication

I feel embarrassed to dedicate a work to someone who is, indeed, a partner in it. Should there be a coauthor in a dissertation, verily she is the one. Her encouragement, wisdom, patience, righteousness, dedication, and supportive partnership in this life are more than a scholarly coauthorship. These few words, in no way, can be an award; rather, they depict the love, gratitude, and respect I have towards such a person. To my wife, I dedicate this work; may I be to her more than what she deserves: *"The most perfect believers are the best in conduct. And the best of you are those who are the best to their wives."*[3]

[3] Allah's Messenger (may peace be upon him) said.

Contents

List of Tables

List of Figures

List of Symbols

In this list the symbol s is a generic symbol to refer to any metric that assesses a classification rule. Thus, s can be the Error Rate (Err), the Area Under the ROC Curve (AUC), or the Partial Area Under the ROC Curve ($PAUC$). References are mentioned for some symbols to indicate the first occurrence in the dissertation.

$F \longrightarrow \mathbf{t}$	A sample $\mathbf{t}$ sampled from a distribution F
$\widehat{F}$	The empirical distribution, also called MLE, of the distribution F, i.e., putting $1/n$ mass on every observation, where n is the sample size.
$\widehat{F} \longrightarrow \mathbf{t}^*$	Bootstrap sampling, from the empirical distribution $\widehat{F}$, by sampling with replacement.
$\mathbf{t}^{*b}$	The b^{th} bootstrap sample replicated from the sample $\mathbf{t}$.
$\eta_{\mathbf{t}}$	The classification rule η trained on the sample $\mathbf{t}$.
$\eta_{\mathbf{t}^{*b}}$	The classification rule η trained on the bootstrap sample $\mathbf{t}^{*b}$.
$\eta_{\mathbf{t}^{(i)}}$	The classification rule η trained on the sample $\mathbf{t}$ with exclusion of the observation t_i.
$h_{\mathbf{t}}$	The log-likelihood ratio of the rule η trained on the set $\mathbf{t}$.
E_F	Expectation over the population F.
$\mathrm{E}_{\widehat{F}}$	Expectation over the empirical distribution $\widehat{F}$ $(= \frac{1}{n}\sum_{i=1}^{n}(\cdot))$.
E_*	Expectation over bootstraps, which equals to $\lim_{B\to\infty}\frac{1}{B}\sum_{b=1}^{B}(\cdot)$.
E_{MC}	Expectation over Monte-Carlo (MC) trials, which equals to $\lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}(\cdot)$.
$s_{\mathbf{t}}$	The "true" performance of the classifier $\eta_{\mathbf{t}}$, which is trained on the training set $\mathbf{t}$, measured in the metric s. This is also referred to as the performance "conditional" on that particular training set.
$\mathrm{E}_{\mathbf{t}}(s_{\mathbf{t}})$	The mean, over training sets, of $s_{\mathbf{t}}$. The subscript $\mathbf{t}$ in $\mathrm{E}_{\mathbf{t}}$ is redundant, since the only source of variability for $s_{\mathbf{t}}$ is the training sets, and it is included for emphasis.
$\mathrm{Var}_{\mathbf{t}}(s_{\mathbf{t}})$	The variance, over training sets, of $s_{\mathbf{t}}$
$\widehat{s}_{\mathbf{t}}$	An estimator of $s_{\mathbf{t}}$. This estimator can be a function of only the training set $\mathbf{t}$, i.e., $\widehat{s}_{\mathbf{t}} = \widehat{s}_{\mathbf{t}}(\mathbf{t})$ (Chapters 2–5). It can also be a function of both the training and testing sets; in such a case, for notational clarity we change $\mathbf{t}$ to $\mathbf{tr}$ and write $\widehat{s}_{\mathbf{tr}} = \widehat{s}_{\mathbf{tr}}(\mathbf{tr}, \mathbf{ts})$ (Chapter 6).
$\widehat{\mathrm{E}_{\mathbf{t}}(s_{\mathbf{t}})}$	An estimator of $\mathrm{E}_{\mathbf{t}}(s_{\mathbf{t}})$
$\widehat{\mathrm{Var}_{\mathbf{t}}(s_{\mathbf{t}})}$	An estimator of $\mathrm{Var}_{\mathbf{t}}(s_{\mathbf{t}})$
$\mathrm{E}(\widehat{s}_{\mathbf{t}})$	The mean of the estimator $\widehat{s}_{\mathbf{t}}$. There is no subscript for the operator E, i.e., the expectation is taken over training and testing sets. When the expectation is taken over either the trainers, **tr**, or testers, **ts**, the operator E is subscripted.

$\mathrm{Var}(\widehat{s}_{\mathbf{t}})$	The variance of the estimator $\widehat{s}_{\mathbf{t}}$
$\mathrm{E}\left(\widehat{\mathrm{E}_{\mathbf{t}}\, s_{\mathbf{t}}}\right)$	The mean of the estimator $\widehat{\mathrm{E}_{\mathbf{t}}\, s_{\mathbf{t}}}$
$\mathrm{Var}\left(\widehat{\mathrm{E}_{\mathbf{t}}\, s_{\mathbf{t}}}\right)$	The variance of the estimator $\widehat{\mathrm{E}_{\mathbf{t}}\, s_{\mathbf{t}}}$
$\overline{s}$	Apparent estimator, i.e., the estimator obtained by training on the entire training set **tr** and then testing on the entire same set (Section 2.2.1).
$\widehat{s}^{(SB)}$	The simple bootstrap estimator for s (Section 2.2.4).
$\widehat{s}^{(RF)}$	The refined bootstrap estimator (Section 2.2.4.2).
$\widehat{s}_{\mathbf{t}^{*b}}\left(\widehat{F}^{(*)}\right)$	The metric s after training on the bootstrap replicate b and testing on all of the observations $\widehat{F}^{(*)}$ that did not appear in that replication.
$\widehat{s}^{(*)}$	A bootstrap estimator resulting from testing on the observations that did not appear in the bootstrap replication, i.e., $\widehat{s}^{(*)} = \mathrm{E}_{*}\left(\widehat{s}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)})\right)$.
$\widehat{s}^{(1)}$	The leave-one-out estimator, when s is the error rate (Section 2.2.4.1).
$\widehat{s}^{(1,1)}$	The leave-pair-out estimator, when s is either the AUC or the $PAUC$ (Section 3.3).
$\widehat{s}^{(.632)}$	The .632 bootstrap estimator (Section 2.2.4.3).
$\widehat{s}^{(.632+)}$	The .632+ bootstrap estimator (Section 2.2.4.4).
I_i^b	An indicator function that equals one when the observation i does not appear in the bootstrap b, and equals zero otherwise.

Preface

This preface serves as a road map to the dissertation. It introduces every chapter and indicates how the chapters are linked together.

Chapter 1 is a literature review for the general problem of statistical learning. It gives a brief account of the statistical decision theory necessary for understanding the general problem of regression and classification. It illustrates the unified principle of all classification rules, i.e., the estimation of the posterior probability of the class to be predicted. The end of the chapter introduces the ROC curve and the assessment problem.

Chapter 2 is an extension to the literature review chapter. It demonstrates different nonparametric techniques from the statistics literature in estimating the mean and the variance of a statistic. Hence, it is a good introduction to the nonparametric assessment of a classification rule, since it is a statistic that is function of two data sets, a training set and a testing set. It demonstrates, as well, the previous efforts in assessing classification rules in terms of the error rate, i.e., the probability of misclassification (PMC), as a performance metric.

Chapter 3 is an introduction to the work done in this dissertation, which is, mainly, represented in detail in the three subsequent chapters. This chapter defines the main problems that are considered in this dissertation. Section 3.2 gives an introduction to how the current methods of assessing classifiers in terms of the error rate can be extended to the area under the ROC curve (AUC). It illustrates, as well, the important issue of the bootstrap bias when resampling the training set. This chapter is the basis for the article Yousef, Wagner and Loew (2004).

Chapter 4 is a contribution to the literature on the use of the AUC as a performance metric. In particular, it provides a method for estimating the uncertainty of the estimator that estimates the mean performance of a classifier in terms of the AUC. The main statistical tool in this chapter is the influence function. This chapter is the basis for the article Yousef, Wagner and Loew (2005).

Chapter 5 introduces the Partial Area Under the ROC Curve (PAUC) as a proposed metric instead of the AUC, when some information is available on the testing environment. This chapter analyzes the PAUC and shows the pros and cons of such a metric. This chapter is the basis for the article Yousef (2013).

Chapter 6 concerns assessing classifiers from two independent data sets, as may be required in a public-policy-making regulatory setting. The assessment of classifiers in mean and variance is derived mathematically and checked by simulation studies. The methods developed in this chapter rely on the theory of U-statistics. This chapter is the basis for the article Yousef, Wagner and Loew (2006).

Chapter 7 provides a brief summary of the dissertation together with some final conclusions. It discusses, as well, possible future work and natural extensions to this dissertation.

CHAPTER 1

Classification and Regression: Literature Review

1.1. Introduction and Terminology

In the present chapter some basic concepts and terminology necessary for the sequel will be formally introduced. The world of variables can be categorized into two categories: deterministic variables and random variables. A deterministic variable takes a definite value; the same value will be the outcome if the experiment that yielded this value is rerun. On contrary, a random variable is a variable that takes a non-definite value with a probability value.

Definition **1.1**. A random variable X is a function from a sample space S into the real numbers $\Re$, that associates a real number, $x = X(s)$, with each possible outcome $s \in S$.

Details on the topic can be found in Casella and Berger (2002, Ch. 1). For more rigorous treatment of random variables based on measure theoretic approach see Billingsley (1995). Variables can be categorized as well, based on value, into: quantitative or metric, qualitative or categorical, and ordered categorical A quantitative variable takes a value on $\Re$ and it can be discrete or continuous. A qualitative or categorical variable does not necessarily take a numerical value; rather it takes a value from a finite set. E.g., the set $\mathcal{G} = \{Red, Green, Blue\}$ is a set of possible qualitative values that can be assigned to a color. An ordered categorical variable is a categorical variable with relative algebraic relations among the values. E.g., the set $\mathcal{G} = \{Small, Medium, Large\}$ includes ordered categorical values.

Variables in a particular process are related to each other in a certain manner. When variables are random the process is said to be stochastic, i.e., when the inputs of this process have some specified values there is no deterministic value for the output, rather a probabilistic one. The output in this case is a random variable.

We next consider the general problem of statistical learning algorithms. Consider a sample consisting of a number of cases—the words cases and observations may be used exchangeably—, where each case is composed of the set of inputs that will be given to the algorithm together with the corresponding output. Such a sample provides the means for the algorithm to learn during its so-called "design" stage. The goal of this learning or design stage is to understand as much as possible how the output is related to the inputs in these observations, so that when a new set of inputs is given in the future the algorithm will have some means of predicting the corresponding output. The above terminology has been borrowed from the field of machine learning. This problem is originally from the field of statistical decision theory, where the terminology is somewhat different. In the latter field, the inputs are called the predictors and the output is called the response. When the output is quantitative the learning algorithm is called regression; when the output is categorical or ordered categorical the learning algorithm is called classification. In the engineering communities that work on the pattern classification problem, the terms input features and output class are used respectively. The learning process in that setting is called training and the algorithm is called the classifier.

Definition **1.2**. Learning is the process of estimating an unknown input-output dependency or structure of a system using a limited number of observations.

Statistical learning is crucial to many applications. For example, In the medical imaging field, a tumor on a mammogram must be classified as malignant or benign. This is an example of prediction, regardless of whether it is done by a radiologist or by a computer algorithm (Computer Aided Diagnosis or CAD). In either case the prediction is done based on learning from previous mammograms. The features, i.e., predictors, in this case may be the size of the tumor, its density, various shape parameters, etc. The output, i.e., response, is a categorical one which belongs to the set: $\mathcal{G} = \{benign, malignant\}$. There are so many such examples in biology and medicine that it is almost a field unto itself, i.e., biostatistics. The task may be diagnostic as in the mammographic example, or prognostic where, for example, one estimates the probability of occurrence of a second heart attack for a particular patient who has had a previous one. All of these examples involve a prediction step based on previous learning. A wide range of commercial and military applications arises in the field of satellite imaging. Predictors in this case can be measures from the image spectrum, while the response can be the type of land or crop or vegetation of which the image was taken

Before going through some mathematical details, it is convenient to introduce some commonly used notation. A random variable—or a random vector—is referred to by an upper-case letter, e.g., X. An instance, or observation, of that variable is referred to by a lower-case letter, e.g., x. A collection of N observations for the p-dimensional random vector X is collected into an $N \times P$ matrix and represented by a bold upper-case $\mathbf{X}$. A lower-case bold letter $\mathbf{x}$ is reserved for describing a vector of any N-observations of a variable, even a tuple consisting of non-homogeneous types. The main notation in the sequel will be as follows: $\mathbf{t} : \{t_i = (x_i, y_i)\}$ represents an n-case training data set, i.e., one on which the learning mechanism will execute. Every

sample case t_i of this set represents a tuple of the predictors x_i represented in a p-dimensional vector, and the corresponding response variable y_i. All the N observations x_i's may be written in a single $N \times P$ matrix $\mathbf{X}$, while all the observations y_i may be written in a vector $\mathbf{y}$.

1.2. Statistical Decision Theory

This section provides an introduction to statistical decision theory, which serves as the foundation of statistical learning. If a random vector X and a random variable$Y$ have a joint probability density $f_{X,Y}(x,y)$, the problem is defined as follows: how to predict the variable Y from an observed value for the variable X. In this section we assume having a full knowledge of the joint density $f_{X,Y}$, so there is no learning yet (Definition 1.2). The prediction function $\eta(X)$ is required to have minimum average prediction error. The prediction error should be defined in terms of some loss function $L(Y,\eta(X))$ that penalizes for any deviation in the predicted value of the response from the correct value. Define the predicted value by:

$$\hat{Y} = \eta(X) \tag{1.1}$$

The risk of this prediction function is defined by the average loss, according to the defined loss function, for the case of prediction:

$$R(\eta) = E\left[L(Y,\hat{Y})\right] \tag{1.2}$$

For instance, some constraint will be imposed on the response Y by assuming it, e.g., to be a quantitative variable. This is the starting point of the statistical branch of regression, where (1.1) is the regression function. A form should be assumed for the loss function. A mathematically convenient and widely used form is the squared-error loss function:

$$L\left(Y,\eta(X)\right) = \left(Y-\eta(X)\right)^2 \tag{1.3}$$

In this case (1.2) becomes:

$$R(\eta) = \int \left(Y-\eta(X)\right)^2 dF_{X,Y}(X,Y) \tag{1.4}$$

$$= E_X\left[E_{Y|X}\left[\left(Y-\eta(X)\right)^2|X\right]\right] \tag{1.5}$$

hence, (1.5) is minimized by minimizing the inner expectation over every possible value for the variable X. Ordinary vector calculus solves the minimization for $\eta(X)$ and gives:

$$\eta(X) = \arg\min_{\eta(X)}\left(E_{Y|X}\left[\left(Y-\eta(X)\right)^2|X\right]\right) \tag{1.6}$$

$$= E_Y\left[Y|X\right] \tag{1.7}$$

This means that if the joint distribution for the response and predictor is known, the best regression function in the sense of minimizing the risk is the expectation of the response conditional on the predictor. In that case the risk of regression in (1.5) will be:

$$R_{\min}(\eta) = E_X\left[\mathrm{Var}\left[Y|X\right]\right]$$

Recalling (1.2), and lifting the constraint on the response being quantitative, and setting another constraint by assuming it to be a qualitative (or categorical) variable gives rise to the classification problem. Now the loss function cannot be the squared-error loss function defined in (1.3), since this has no meaning for categorical variables. Since Y may take now a qualitative value from a set of size k, (see Section 1.1), the loss function can be defined by the matrix

$$L(Y,\eta(X)) = \left(\left(c_{ij}\right)\right),\ 1 < i,\ j < k \tag{1.8}$$

where the non-negative element c_{ij} is the cost, the penalty or the price, paid for classifying an observation as y_j when it belongs to y_i. In the field of medical decision making this is often called the *utility matrix*. Under this assumption, the risk defined by (1.2) can be rewritten for the categorical variables to be:

$$R(\eta) = E_X E_{Y|X}\left[L\left(Y,\eta(X)\right)\right] \tag{1.9}$$

$$= E_X\left[\sum_{i=1}^{k} c_{ij}\Pr\left[Y=y_i|X\right]\right] \tag{1.10}$$

where $\Pr[Y|X]$ is the probability mass function for Y conditional on X. Then the conditional risk for decision y_j

$$R(j,\eta) = \sum_{i=1}^{k} c_{ij}\Pr\left[Y=y_i|X\right] \tag{1.11}$$

is the expected loss when classifying an observation as belonging to y_j and the expectation is taken over all the possible values of the response. Again, (1.10) can be minimized by minimizing the inner expectation to give:

$$\eta(X) = \arg\min_{j}\left[\sum_{i=1}^{k} c_{ij}\Pr\left[Y=y_i|X\right]\right] \tag{1.12}$$

Expressing the conditional probability of the response in terms of Bayes law and substituting in (1.12) gives:

$$\eta(X) = \arg\min_j \sum_{i=1}^{k} c_{ij} f_X\left(X|Y = y_i\right) \Pr\left[y_i\right] \tag{1.13}$$

$\Pr\left[y_i\right]$ is the prior probability for y_j while $\Pr\left[y_j|X\right]$ is the posterior probability, i.e., the probability that the observed case belongs to y_j, given the value of X. This is what statisticians call Bayes classification, or Bayes decision rule or alternatively, what engineers call the Bayes classifier.

Some special cases here may be of interest. The first case is when equal costs are assigned to all misclassifications and there is no cost for correct classification; this is called the 0-1 cost function. This reduces(1.12) to:

$$\eta(X) = \arg\min_j \left[1 - \Pr\left[Y = y_j|X\right]\right] \tag{1.14}$$

$$= \arg\max_j \left[\Pr\left[Y = y_j|X\right]\right] \tag{1.15}$$

The rule thus is to classify the sample case to the class having maximum posterior probability. Another special case of great interest is binary classification, i.e., the case of $k = 2$. In this case (1.12) reduces to:

$$\frac{\Pr\left[y_1|X\right]}{\Pr\left[y_2|X\right]} \underset{y_2}{\overset{y_1}{\gtrless}} \frac{(c_{22} - c_{21})}{(c_{11} - c_{12})} \tag{1.16}$$

Alternatively, this can be expressed as :

$$\frac{f_X(X = x|y_1)}{f_X(X = x|y_2)} \underset{y_2}{\overset{y_1}{\gtrless}} \frac{\Pr\left[y_2\right](c_{22} - c_{21})}{\Pr\left[y_1\right](c_{11} - c_{12})} \tag{1.17}$$

The decision taken in (1.12) has the minimum risk, which can be calculated by substituting back in (1.10) to give:

$$R_{\min}(\eta) = \sum_{i=1}^{k} \int_X c_{i,j(X)} \Pr\left[y_i\right] dF_X(X|y_i) \tag{1.18}$$

where $j(X)$ is the class decision $\eta(X)$. For binary classification and where there is no cost for a correct decision, i.e., $c_{11} = c_{22} = 0$, this reduces to:

$$R_{\min}(\eta) = c_{12} \Pr\left[y_1\right] \int_{R_2} dF_X(X|y_1) + c_{21} \Pr\left[y_2\right] \int_{R_1} dF_X(X|y_2) \tag{1.19}$$

where each of R_1 and R_2 is the predictor hyperspace over which the optimum decision (1.16) predicts as class 1 or class 2 respectively. The binary classification problem will be revisited in Chapter 3 for a more detailed treatment. Latter, the response variable Y may be referred to Ω in case of classification. To follow the notation of Section 1.1 the response of an observation is assigned a value ω_i, $i = 1, \ldots, k$ to express a certain class.

To recap, this section emphasizes the fact that there is no distinction between regression and classification from the conceptual point of view. Each minimizes the risk of predicting the response variable for an observation, i.e., a sample case with known predictor(s). If the joint probability distribution function for the response and predictors is known, it is just a matter of direct substitution in the above results. If the joint distribution is known but its parameters are not known, a learning process is used to estimate those parameters from a training sample **t** by methods of statistical inference. However, if the joint distribution is unknown, this gives rise to two different branches of prediction. These two branches are parametric regression (or classification)—where the regression or classification function is modeled and a training sample is used to build that model—and nonparametric regression (or classification), where no particular parametric model is assumed. Subsequent sections in this chapter give introductions to these techniques.

1.3. Parametric Regression and Classification

The prediction method introduced in Section 1.2 assumes, as indicated, that the joint density of the response and the predictor is known. If such knowledge exists, all the methods revolve around modeling the regression function (1.1) in the case of regression or the posterior probabilities in (1.12) in the case of classification.

1.3.1. Linear Models

In linear model (LM) theory, Y is assumed to be in the form:

$$Y = E[Y] + e \tag{1.20}$$

$$= \alpha + X'\beta + e \tag{1.21}$$

where the randomness of Y comes only from e, and it is assumed that the conditional expectation of Y is linear in the predictors X. The two basic assumptions in the theory are the zero mean and constant variance of the random error component e. The

regression function (1.1) is then written as:

$$\eta(X) = \alpha + X'\beta \tag{1.22}$$

More generally, still a linear model, it can be rewritten as:

$$\eta(X) = X'_{new}\beta, \tag{1.23}$$

$$X'_{new} = \left(f_1(X), \ldots, f_d(X)\right) \tag{1.24}$$

where the predictor X is replaced by a new d-dimensional vector, X_{new}, whose elements are scalar functions of the random vector X.

The intercept α in (1.22) may be modeled, if needed, in terms of (1.23) by setting $f_1(X) = 1$. Equation (1.23) can be seen as equivalent to (1.22), where X has been transformed to X_{new} which became the new predictor on which Y will be regressed.

Now β must be estimated, and this point estimation is done for some observed values of the predictor. Writing the equations for n observed values gives:

$$\mathbf{y} = \mathbf{X}'\beta + \mathbf{e} \tag{1.25}$$

If (1.25) is solved for β to give the least sum of squares for the components of error vector $\mathbf{e}$, this will give, as expected, the same result as if we approximated the conditional expectation of Y by the set of observations $\mathbf{y}$. Solving either way gives:

$$\widehat{\beta} = \left(\mathbf{X}\mathbf{X}'\right)^{-1}\mathbf{X}\mathbf{y} \tag{1.26}$$

Then the prediction of Y is done by estimating its expectation which is given by:

$$\widehat{\eta(X)} = \widehat{E[Y]} = X'\widehat{\beta} \tag{1.27}$$

For short notation we always write $\widehat{Y}$ instead of $\widehat{E[Y]}$.

Nothing up to this point involves statistical inference. This is just fitting a mathematical model using the squared-error loss function. Statistical inference starts when considering the random error vector $\mathbf{e}$ and the effect of that on the confidence interval for $\hat{\beta}$ and the confidence in predicted values of the response for particular predictor variable, or any other needed inference. All of these important questions are answered by the theory of linear models. Bowerman and O'Connell (1990) is a very good reference for an applied approach to linear models, without any mathematical proofs. For a theoretical approach and derivations the reader is referred to Christensen (2002), Graybill (1976), and Rencher (2000). It is remarkable that if the joint distribution for the response and the predictor is multinormal, the linear model assumption (1.21) is an exact expression for the random variable Y. This fact arises from the fact that the conditional expectation for the multinormal distribution is linear in the conditional variable. That is, by assuming that

$$\begin{pmatrix} Y \\ X \end{pmatrix} \sim N(\mu, \Sigma), \; where \tag{1.28}$$

$$\mu = \begin{pmatrix} \mu_Y \\ \mu_X \end{pmatrix}, \; \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \tag{1.29}$$

then the conditional expectation of Y on X is given by:

$$E[Y|X = x] = \mu_Y + \Sigma_{12}\Sigma_{22}^{-1}(x - \mu_X) \tag{1.30}$$

For more details on the multinormal properties, see Anderson (2003).

In the case of classification the classes are categorical variables but a dummy variable can be used as coding for the class labels. Then a linear regression is carried out for this dummy variable on the predictors. A drawback of this approach is what is called class masking, i.e., if more than two classes are used, one or more can be masked by others and they may not be assigned to any of the observations in prediction. For a clear example of masking see Hastie, Tibshirani and Friedman (2001, Sec. 4.2).

1.3.2. Generalized Linear Models

In linear models the response variable is directly related to the regression function by a linear expression of the form of (1.21). In many cases a model can be improved by indirectly relating the response to the predictor through a linear model—some times it is necessary as will be shown for the classification problem. This is done through a transformation or *link* function g by assuming:

$$g(E[Y]) = X'\beta \tag{1.31}$$

Now it is the transformed expectation that is modeled linearly. Hence, linear models are merely a special case of the generalized linear models when the link function is the identity function $g(E[Y]) = E[Y]$.

A very useful link function is the *logit* function defined by:

$$g(\mu) = \log\frac{\mu}{1-\mu}, \; 0 < \mu < 1 \tag{1.32}$$

Through this function the regression function is modeled in terms of the predictor as:

$$E[Y] = \frac{\exp(X'\beta)}{1+\exp(X'\beta)} \tag{1.33}$$

which is known as logistic regression. Equation (1.33) implies a constraint on the response Y, i.e., it must satisfy $0 < E[Y] < 1$, a feature that makes logistic regression an ideal approach for modeling the posterior probabilities in (1.12) for the classification problem. Equation (1.32) models the two-class problem, i.e., binary classification, by considering the new responses Y_1 and Y_2 to be defined in terms of the old responses ω_1 and ω_2, the classes, as:

$$Y_1 = \Pr[\omega_1|X], \tag{1.34}$$

$$Y_2 = \Pr[\omega_2|X] = 1 - \Pr[\omega_1|X] \tag{1.35}$$

The general case of the k-class problem can be modeled using $K-1$ equations, because of the constraint $\sum_i \Pr[\omega_i|X] = 1$, as:

$$\log \frac{\Pr[\omega_i|X=x]}{\Pr[\omega_k|X=x]} = x'\beta_i,\ i = 1,\ldots,K-1 \tag{1.36}$$

Alternatively, (1.36) can be rewritten as:

$$\Pr[\omega_i|X=x] = \frac{\exp(x'\beta_i)}{1+\sum_{j=1}^{K-1}\exp(x'\beta_j)},\ 1 \le i \le K-1, \tag{1.37}$$

$$\Pr[\omega_k|X=x] = \frac{1}{1+\sum_{j=1}^{K-1}\exp(x'\beta_j)} \tag{1.38}$$

The question now is how to estimate $\beta_i\ \forall\ i$. The multinomial distribution for modeling observations is appropriate here. For illustration, consider the case of binary classification; the log-likelihood for the n-observations can then be written as:

$$l(\beta) = \sum_{i=1}^{n}\{y_i \log \Pr[\omega_1|X_i,\beta] + (1-y_i)\log(1-\Pr[\omega_1|X_i,\beta]\} \tag{1.39}$$

$$= \sum_{i=1}^{n}\left\{y_i x_i'\beta - \log(1+e^{x_i'\beta})\right\} \tag{1.40}$$

To maximize this likelihood, the first derivative is set to zero to obtain:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{n} x_i\left(y_i - \frac{e^{x_i'\beta}}{1+e^{x_i'\beta}}\right) \stackrel{set}{=} 0 \tag{1.41}$$

This is a set of k equations, where the vector X can be the original predictor $(x_1,\ldots,x_p)'$ or any transformation $(f_1(X),\ldots,f_d(X))'$ as in (1.24). Equation (1.41) is a set of non-linear equations, and can by solved by iterative numerical methods like the Newton-Raphson algorithm. For more details with numerical examples see Hastie, Tibshirani and Friedman (2001, Sec. 4.4) or Casella and Berger (2002, Sec. 12.3).

It can be noted that (1.39) is valid under the assumption of the following general distribution:

$$f(X) = \phi(\theta_i,\gamma)h(X,\gamma)\exp(\theta_i' X) \tag{1.42}$$

with probability p_i, $i = 1,2$, $p_1 + p_2 = 1$, which is the exponential family. So logistic regression is no longer an approximation for the posterior class probability if the distribution belongs to the exponential family. For insightful comparison between logistic regression and the Bayes classifier under the multinormal assumption see Efron (1975).

It is very important to mention that logistic regression, and all subsequent classification methods, assume equal a priori probabilities. Then the ratio between the posterior probabilities will be the same as the ratio between the densities that appear in (1.13). Hence, the estimated posterior probabilities from any classification method are used in (1.13) as if they are the estimated densities.

1.3.3. Non-linear Models

The link function in the generalized linear models is modeled linearly in the predictors, (1.31). Consequently, the response variable is modeled as a non-linear function. In contrast to the linear models described in Section 1.3.1, in non-linear models the response can be modeled non-linearly right from the beginning, without the need for a link function.

1.4. Nonparametric Regression and Classification

In contrast to parametric regression, the regression function (1.1) is not modeled parametrically, i.e., there is no particular parametric form to be imposed on the function. Nonparametric regression is a versatile and flexible method of exploring the relationship of two variables. It may appear that this technique is more efficient than the linear models, but this is not the case. Linear models and nonparametric models can be thought of as two different techniques in the analyst's toolbox. If there is an a priori reason to believe that the data follow a parametric form, then linear models or parametric regression in general may provide an argument for an optimal choice. If there is no prior knowledge about the parametric form the data may follow or no prior information about the physical phenomenon that generated the data, there may be no choice other than nonparametric regression.

There are many nonparametric techniques proposed in the statistical literature. Some of these techniques have also been developed in the engineering community under different names, e.g., artificial neural networks. What was said above, when comparing parametric and nonparametric methods, can also be said when comparing nonparametric methods to each other. None can be preferred overall across all situations.

This section introduces some of the nonparametric regression and classification methods. The purpose is not to present a survey as much as to introduce the topic and show how it relates with the parametric methods to serve one purpose, predicting a response variable, categorical or quantitative. An excellent comprehensive source for regression and classification methods, with practical approaches and illustrative examples, is Hastie, Tibshirani and Friedman (2001).

1.4.1. Smoothing Techniques

Smoothing is a tool for summarizing in a nonparametric way a trend between a response and a predictor such that the resulting relationship is less variable than the original response, hence the name smoothing. When the predictor is unidimensional, the smoothing is called scatter-plot smoothing. In this section, some methods used in scatter-plot smoothing are considered. These smoothing methods do not succeed in higher dimensionality. This is one bad aspect of what is called the curse of dimensionality, which will be discussed in Section 1.7.

1.4.1.1. K-Nearest Neighbor

The regression function (1.1) is estimated in the K-nearest neighbor approach by:

$$\eta(x) = \frac{1}{n}\sum_{i=1}^{n} W_i(x) y_i, \tag{1.43}$$

$$W_i(x) = \begin{cases} n/k & i \in \jmath_x = \{i : x_i \in N_k(x)\} \\ 0 & otherwise \end{cases} \tag{1.44}$$

where $N_k(x)$ is the set consisting of the nearest k points to the point x. So in the case of regression, this technique approximates the conditional mean, i.e., the regression function that gives minimum risk, by local averaging for the response Y.

In the case of classification, the posterior probability is estimated by:

$$\Pr[\omega_j|x] = \frac{1}{n}\sum_{i=1}^{n} W_i(x) I_{\omega_i=\omega_j} \tag{1.45}$$

and I is the indicator function defined by:

$$I_{cond} = \begin{cases} 1 \; cond \; is \; True \\ 0 \; cond \; is \; False \end{cases} \tag{1.46}$$

That is, replacing the continuous response in (1.43) by an indicator function for each class given each point. So, the posterior probability is approximated by a frequency of occurrence in a k-point neighborhood.

1.4.1.2. Nearest Neighbor

This is a special case of the K-nearest neighbor method where $k = 1$. It can be thought of as narrowing the window W on which regression is carried out. In effect, this makes the regression function or the classifier more complex because it is trying to estimate the distribution at each point.

1.4.1.3. Kernel Smoothing

In this approach a kernel smoothing function is assumed. This means that a weighting and convolution (or mathematical smoothing) is carried out for the points in the neighborhood of the predicted point according to the chosen kernel function.

Formally this is expressed as:

$$\eta(x) = \sum_{i=1}^{n} y_i \left(\frac{K\left(\frac{x-x_i}{h_x}\right)}{\sum_{i'=1}^{n} K\left(\frac{x-x_{i'}}{h_x}\right)} \right) \tag{1.47}$$

Choosing the band-width h_x of the kernel function is not an easy task. Usually it is done numerically by cross-validation. It is worth remarking that K-nearest neighbor smoothing is nothing but a kernel smoothing for which the kernel function is an unsymmetrical flat window spanning the range of the K-nearest neighbors of the point x. The kernel (1.47) is called Nadaraya-Watson kernel.

Historically, Parzen (1962) first introduced the window method density function estimation; then his work was pioneered by Nadaraya (1964) and Watson (1964) in regression.

1.4.2. Additive Models

Recalling (1.23) and noticing that the function $f_i(X)$ is a scalar parametric function of the whole predictor shows that linear models are parametric additive models. By dropping the parametric assumption and letting each scalar function be a function of just one element of the predictor, i.e., X_i, allows defining a new nonparametric regression method, namely additive models, as:

$$\eta(x) = \alpha + \sum_{i=1}^{p} f_i(X_i) \tag{1.48}$$

where the predictor is of dimension p. The response variable itself, Y, is modeled as in (1.20) by assuming zero mean and constant variance for the random component e. Then, $f_i(X_i)$ is fit by any smoothing method defined in Section 1.4.1. Every function $f_i(X_i)$ fits the value of the response minus the contribution of the other $p-1$ functions from the previous iteration. This is called the back-fitting algorithm described in Hastie and Tibshirani (1990, Sec. 4.3)

1.4.3. Generalized Additive Models

Generalized additive models can be developed in a way analogous to how generalized linear models were developed above, i.e., by working with a transformation of the response variable, hence the name generalized additive models (GAM). Equation (1.48) describes the regression function as an additive model; alternatively it can be described through another link function:

$$g\left(\eta(x)\right) = \alpha + \sum_{i=1}^{p} f_i(X_i) \tag{1.49}$$

Again, if a *logit* function is used the model can be used for classification exactly as was done in the case of generalized linear models. Rewriting the score equations (1.41) for the GAM, using the posterior probabilities as the response variable, produces the nonparametric classification method using the GAM. Details of fitting the model can be found in Hastie and Tibshirani (1990, Sec. 4.5 and Ch. 6).

1.4.4. Projection Pursuit Regression

Projection Pursuit Regression (PPR), introduced by Friedman and Stuetzle (1981), is a direct attack on the dimensionality problem, since it considers the regression function as a summation of functions, each of which is a function of a projection of the whole predictor onto a direction (specified by some unit vector). Formally it is expressed as:

$$\eta(x) = \sum_{i=1}^{k} g_i(\alpha_i' x) \tag{1.50}$$

The function g_i for every selection for the direction α_i is to be fit by a smoother in the new single variable $\alpha_i' x$. It should be noted that (1.50) assumes that the function $g_i(\alpha_i' X)$, named the *ridge function*, is constant along any direction perpendicular to α_i. Fitting the model is done by iteratively finding the best directions α_i's that minimize(s) the residual sum square of errors, hence the name pursuit. Details of fitting the model and finding the best projection directions can be found in Friedman and Stuetzle (1981) and Hastie, Tibshirani and Friedman (2001).

In (1.50) by deliberately setting each unit vector α_i to have zero components except $\alpha_{ii} = 1$, reduces the projection pursuit method to additive models. Moreover, and interestingly as well, by introducing the *logit* link function to the regression function $\eta(x)$ in (1.50) suits the classification problem exactly as done in the GAM. This turns out to be exactly the same as the single-hidden-layer neural network, as will be presented in the next section.

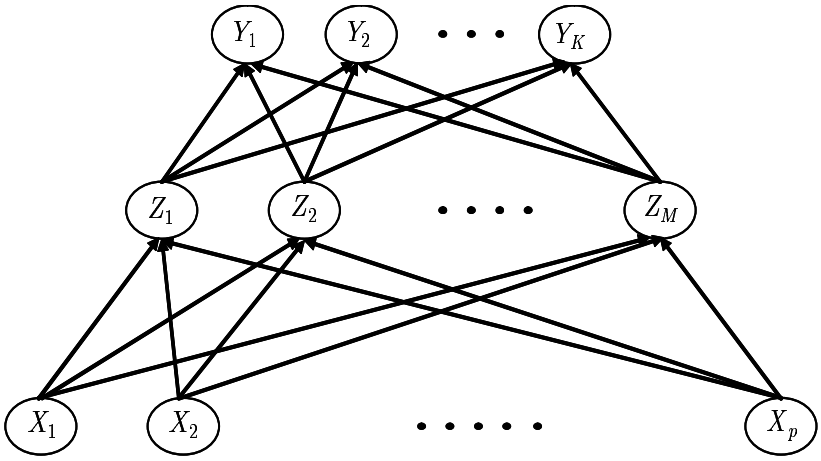

Figure 1.1. Schematic diagram for a single hidden layer neural network.

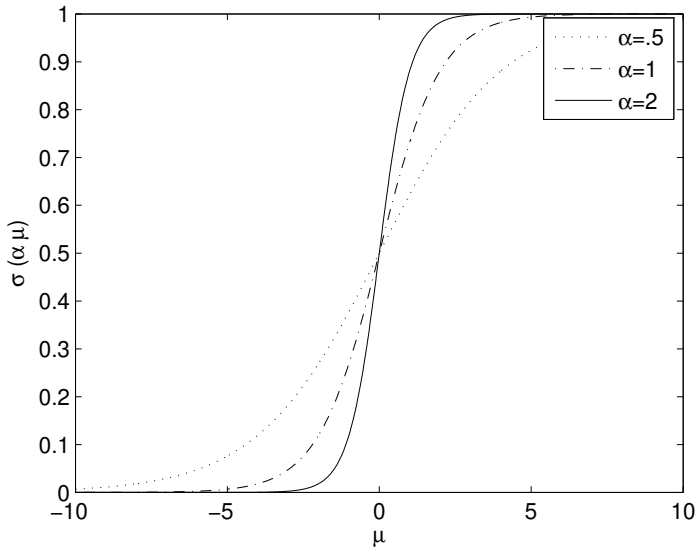

Figure 1.2. Sigmoid function under different learning rate α

1.4.5. Neural Networks

Neural Networks (NN) have evolved in the engineering community since the 1950s. As illustrated in Figure 1.1, a neural network can be considered as a process for modeling the output in terms of a linear combination of the inputs.

The set of p input features, i.e., the predictor components $X_1,\ldots,X_p$, are weighted linearly to form a new set of M arguments, $Z_1,\ldots,Z_M$, that go through the sigmoid function σ. The output of the sigmoid functions accounts for a hidden layer consisting of M intermediate values. Then these M hidden values are weighted linearly to form a new set of K arguments that go through the final output functions whose output is the response variables $Y_1,\ldots,Y_K$. This can be expressed mathematically in the form:

$$Z_m = \sigma(\alpha_{om} + \alpha'_m X),\ m = 1,2,\ldots,M, \tag{1.51}$$

$$Y_k = f_k\left(\beta_{0k} + \sum_{m=1}^{M} \beta_{mk} Z_m\right),\ k = 1,2,\ldots,K \tag{1.52}$$

Figure 1.2 shows the function under different values of α (called learning rate below).

The sigmoid function is defined by:

$$\sigma(\mu) = \frac{1}{1+e^{-\mu}} \tag{1.53}$$

Equation (1.52) shows that if the function f is chosen to be the identity function, i.e., $f(\mu) = \mu$, the neural network is simply a special case of the projection pursuit method defined in (1.50), where the sigmoid function has been explicitly imposed on the model rather than being developed by any smoothing mechanism as in PPR. This is what is done when the output of the network

is quantitative. When it is categorical, i.e., the case of classification, the contemporary trend is to model the function f as:

$$f_k(\mu_k) = \frac{e^{\mu_k}}{\sum_{k'=1}^{K} e^{\mu_{k'}}} \tag{1.54}$$

In that case each output node models the posterior probability $\Pr[\omega_k|X]$, which is exactly what is done by the multi-logistic regression link function defined in (1.32). Again, the model will be an extension to the generalized additive models as defined at the end of Section 1.4.4. Excellent references for neural networks are Bishop (1995) and Ripley (1996). We conclude this section by quoting the following statement from Hastie, Tibshirani and Friedman (2001):

> "There has been a great deal of hype surrounding neural networks, making them seem magical and mysterious. As we make clear in this section, they are just nonlinear statistical models, much like the projection pursuit regression model discussed above."

1.5. Computational Intelligence

The term computational intelligence was first coined by Bezdek (1992) and Bezdek (1994):

> "A system is computationally intelligent when it: deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI (Artificial Intelligence) sense; and additionally, when it (begins to) exhibit (i) computational adaptivity; (ii) computational fault tolerance; (iii) speed approaching human-like turnaround, and (iv) error rates that approximate human performance."

Since that time the term Computational Intelligence (CI) has been accepted as a generic term to the field that combine Neural Networks, Fuzzy Logic, and Evolutionary Algorithms; see Schwefel, Wegener and Weinert (2003) and Zimmermann et al. (2002). As a still-developing field, CI may incorporate other methodologies as a coherent part. In Engelbrecht (2002), the area of Swarm Detection is considered as a peer paradigm to the other three mentioned above.

In the spirit of what has been discussed in the preceding sections, these methods assume nothing about the data distributions; they try to approach the solution by merely dealing with the data, i.e., numbers (c.f. the definition above). Hence, the CI methods, from a purely statistical point of view, are considered as nonparametric methods. Sections 1.4.4 and 1.4.5 illustrated, mathematically, how Neural Networks, a basic building block in the CI field, is a special case of the projection pursuit, a nonparametric regression method.

1.6. No overall Winner among All Methods

This statement is important enough to be emphasized under a separate title, even though it has been touched upon throughout previous sections. If there is no prior information for the joint distribution between the response and the predictor, and if there is no prior information about the phenomenon to which that regression or classification will be applied, there is no overall winner among regression or classification techniques. If one classification method is found to outperform others in some application, this is likely to be limited to that very situation or that specific kind of problem; it may be beaten by other methods for other situations. In the engineering community, this concept is referred to as the *No-Free-Lunch* Theorem (see Duda, Hart and Stork, 2001, Sec. 9.2). This situation holds because each method makes different assumptions about the application or the process being modeled, and not all real-life applications are the same. If one or more of the assumptions are not satisfied in a given application, the performance will not be optimal in that setting.

1.7. Curse of Dimensionality and Dimensionality Reduction

In general, smoothing is difficult to implement in higher dimensions. This is because for a fixed number of observations available, the volume size needed to cover a particular percentage of the total number of observations increases by a power law, and thus exponentially, with dimensionality. This makes it prohibitive to include the same sufficient number of observations within a small neighborhood, or bandwidth, for a sample case to smooth the response. E.g., consider a unit hyper-cube in the p-dimensional subspace containing uniformly distributed observations; the percentage of the points located inside a hyper-cube with side length l is l^p. This means, if the suitable band-width for a certain smoother is l, the effective number of sample cases in the p-dimensional problem will go as the power $1/p$. This deteriorates the performance dramatically for p higher than 3. This is why the additive model, Section 1.4.2, and its variants are expressed as summation of functions of just one dimension. This single dimension may be just a component of the predictor or a linear combination.

A very crucial sub-field in statistical learning is dimensionality reduction; alternatively it is called feature selection in the engineering community. Qualitatively speaking, this means selecting those predictor components that best summarize the

relationship between the response and predictor. In real-life problems, some features are statistically dependent on others; this is referred to as multi-collinearity. On the other hand, there may also be some components that are statistically independent from the response. These add no additional information to the problem at all; thus they serve only as a source of noise

This is a rapidly maturing sub-field. A remarkable publication in the statistics literature in this regard is that by Li (1991). It introduces the Sliced Inverse Regression (SIR), in which each predictor component is regressed on the response; hence the name inverse regression. In that sense, the problem is reduced from regressing a single response on a p-dimensional predictor to regressing p-responses on a single-dimensional new predictor, which is far simpler than the former.

1.8. Unsupervised Learning

It should be noticed that the formal definition of the learning process, discussed thus far in the present chapter, assumed the existence of a training data set, name it, $\mathbf{t}: \{t_i = (x_i, y_i)\}$. Each element t_i, or sample case, in this set has an already known value for the response variable; this is what enables the learning process to develop the relationship between the predictor and the response. This is what is called supervised learning. On the contrary, in some applications the available data set is described by $\mathbf{t}: \{t_i = x_i\}$ without any additional information. This situation is called unsupervised learning. The objective in such a situation is to understand the structure of the data from the available empirical probability distribution of the points x_i. For the special case where the data come from different classes, the data will be represented in the hyper p-dimensional subspace , to some extent, as disjoint clouds of data. The task in this case is called clustering, i.e., trying to identify those classes that best describe, in some sense, the current available data. More formally, if the available data set is $\mathbf{X}$, the objective is to find the class vector $\Omega = [\omega_1, \ldots, \omega_k]'$ such that a criterion $\jmath(\mathbf{X}, \Omega)$ is minimized:

$$\Omega = \arg\min \jmath(\mathbf{X}, \Omega) \tag{1.55}$$

Different criteria give rise to different clustering algorithms. More discussion on unsupervised learning and clustering can be found in Duda, Hart and Stork (2001); Fukunaga (1990); Hastie, Tibshirani and Friedman (2001). This dissertation is concerned with the problem of supervised learning.

1.9. Performance of Classification Rules

From what has been discussed until now, there is not any conceptual difference between regression and classification for the problem of supervised learning. Abstractly, both aim to achieve the minimum risk under a certain loss function for predicting a response from a particular predictor. If the special case of classification is considered, there should be some metric to assess the performance of the classification rule. Said differently, if several classifiers are competing in the same problem, which is better? One natural answer is to consider the risk of each classifier, as was defined in (1.10).

A special case of classification, which is of great interest in many applications, is binary classification, where the number of classes is just two. In that case the risk of each classifier is reduced to (1.19), which can be rewritten as:

$$R_{\min} = c_{12} P_1 e_1 + c_{21} P_2 e_2 \tag{1.56}$$

where e_1 is the probability of classifying a case as belonging to class 2 when it belongs to class 1, and e_2 is vice versa.

In the feature subspace, the regions of classification have the dimensionality p, and it is very difficult to calculate the error components from multi-dimensional integration. It is easier to look at (1.17) as:

$$h(x) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} th, \; where \tag{1.57}$$

$$h(x) = \log \frac{f_X(X = x|\omega_1)}{f_X(X = x|\omega_2)}, \tag{1.58}$$

$$th = \log \frac{\Pr[\omega_1](c_{22} - c_{21})}{\Pr[\omega_2](c_{11} - c_{12})}, \tag{1.59}$$

and $h(X)$ is called the log-likelihood ratio. Now the log-likelihood ratio itself is a random variable whose variability comes from the feature vector X, and has a PDF conditional on the true class. This is shown in Figure 1.3. It can be easily shown that the two curves in Figure 1.3 cross at $h(X) = 0$, where the threshold is zero. In this case the two error components, appearing in (1.56), are written equivalently as:

$$e_1 = \int_{-\infty}^{th} f_h(h(x)|\omega_1)\, dh(x), \tag{1.60a}$$

$$e_2 = \int_{th}^{\infty} f_h(h(x)|\omega_2)\, dh(x) \tag{1.60b}$$

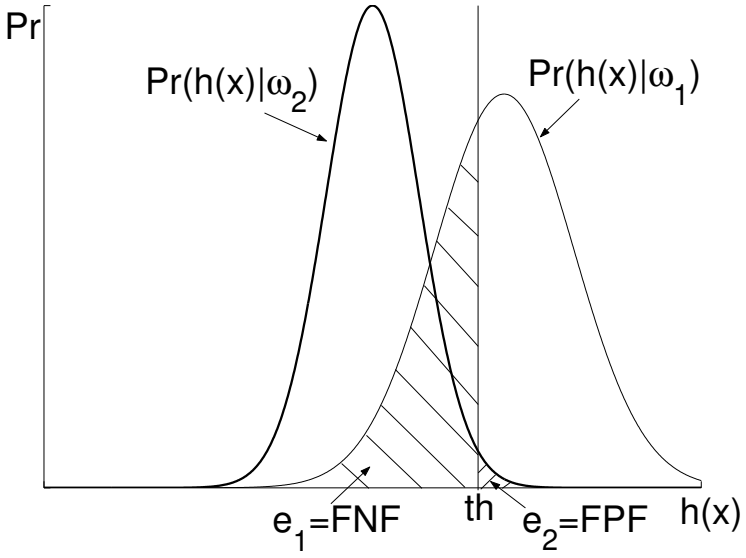

Figure 1.3. The probability of log-likelihood ratio conditional under each class. The two components of error are indicated as the FPF and FNF, the conventional terminology in medical imaging.

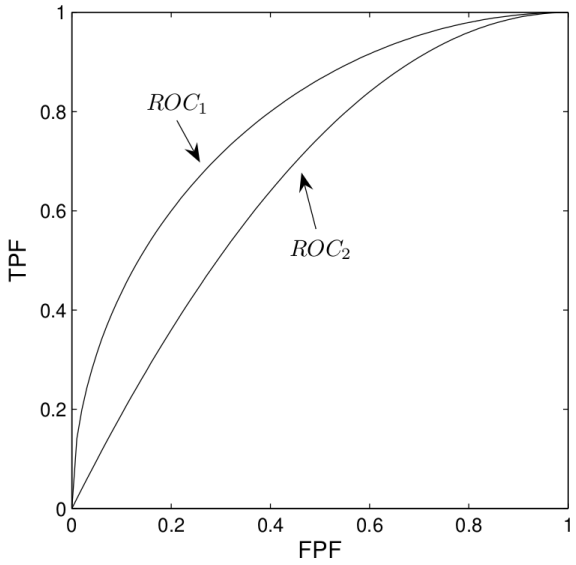

Figure 1.4. ROC curves for two different classifiers. ROC_1 is better than ROC_2, since for any error component value, the other component of classifier 1 is less than that one of classifier 2.

Now assume the classifier is trained under the condition of equal prevalence and cost, i.e., the threshold is zero. In other environments there will be different a priori probabilities yielding to different threshold values. The error is not a sufficient metric now, since it is function of a single fixed threshold. A more general way to assess a classifier is provided by the Receiver Operating Characteristic (ROC) curve. This is a plot for the two components of error, e_1 and e_2 under different threshold values. It is conventional in medical imaging to refer to e_1 as the False Negative Fraction (FNF), and e_2 as the False Positive Fraction (FPF). This is because diseased patients typically have a higher output value for a test than non-diseased patients. For example, a patient belonging to class 1 whose test output value is less than the threshold setting for the test will be called "test negative" while the patient is in fact in the diseased class. This is a false negative decision; hence the name FNF. The situation is reversed for the other error component.

Since the classification problem now can be seen in terms of the log-likelihood, it is apparent that the error components are integrals over a particular PDF. Therefore the resulting ROC is a monotonically non-decreasing function. A convention in medical imaging is to plot the $TPF = 1 - FNF$ vs. the FPF. In that case, the farther apart the two distributions of the log-likelihood function from each other, the higher the ROC curve and the larger the area under the curve (AUC). Figure 1.4 shows ROC curves for two different classifiers.

The first one performs better since it has a lower value of e_2 at each value of e_1. Thus, the first classifier unambiguously separates the two classes better than the second one. Also, the AUC for the first classifier is larger than that for the second one. AUC can be thought of as one summary metric for the ROC curve.

Formally the AUC is given by:

$$AUC = \int_0^1 TPF \; d(FPF) \tag{1.61}$$

If two ROC curves cross, this means each is better than the other for a certain range of the threshold setting, but it is worse in another range. In that case some other metric can be used, such as the partial area under the ROC curve in a specified region (Chapter 5).

The two components of error in (1.56), or the summary metric AUC in (1.61), are the parametric forms of these metrics. That is, these metrics can be calculated by these equations if the posterior probabilities are known parametrically, e.g., in the case of the Bayes classifier or by parametric regression techniques as in Section 1.3.

On the contrary, if the posterior probabilities are not known in a parametric form, the error rates can be estimated only numerically from a given data set, called the testing data set. This is done by assigning equal probability mass for each sample case, since this is the Maximum Likelihood Estimation (MLE) for the probability mass function under the nonparametric distribution. This can be proven by maximizing the likelihood function:

$$L(F) = \prod_{i=1}^{n} p_i \tag{1.62}$$

under the constraint $\Sigma_i p_i = 1$. The likelihood (1.62) can be rewritten, by considering this constraint, using a Lagrange multiplier as:

$$L(F) = \prod_{i=1}^{n} p_i + \lambda \left(\sum_{i=1}^{n} p_i - 1 \right) \tag{1.63}$$

The likelihood (1.63) is maximized by taking the first derivative and setting it to zero to obtain:

$$\frac{\partial L(F)}{\partial p_j} = \prod_{i \neq j} p_i + \lambda \stackrel{set}{=} 0, \; j = 1, \ldots, n \tag{1.64}$$

These n equations along with the constraint $\Sigma_i p_i = 1$ can be solved straightforwardly to give:

$$\hat{p}_i = \frac{1}{n}, \; i = 1, \ldots, n \tag{1.65}$$

That is, the nonparametric MLE of the distribution will be:

$$\hat{F} : mass \; \frac{1}{n} \; on \; t_i, \; i = 1, \ldots, n \tag{1.66}$$

where n is the size of the testing data set. In this case (1.2) will be reduced to:

$$\widehat{R(\eta)} = E_{\hat{F}} \left[L(Y, \eta(X)) \right] \tag{1.67}$$

$$= \frac{1}{n} \sum_{i=1}^{n} L\left(y_i, \eta(x_i) \right) \tag{1.68}$$

where the expectation has been taken over the empirical distribution $\hat{F}$ of the variable. In the case of classification, (1.67) can be reduced further to:

$$\widehat{R(\eta)} = \frac{1}{n} \sum_{i=1}^{n} c_{i,\eta(x_i)} \tag{1.69}$$

In the special case of zero loss for correct decisions in binary classification, (1.69) reduces further to:

$$\widehat{R(\eta)} = \frac{1}{n} \sum_{i=1}^{n} \left(c_{12} \, I_{\hat{h}(x_i|\omega_1) < th} + c_{21} \, I_{\hat{h}(x_i|\omega_2) > th} \right) \tag{1.70}$$

$$= \frac{1}{n} \left(c_{21} \, \widehat{e_1} \, n_1 + c_{21} \, \widehat{e_2} \, n_2 \right) \tag{1.71}$$

$$= c_{21} \, \widehat{FNF} \, \widehat{P_1} + c_{21} \, \widehat{FPF} \, \widehat{P_2} \tag{1.72}$$

which is the nonparametric approximation to (1.56) and (1.60). The indicator function I is defined in (1.46). The values n_1 and n_2 are the sizes of class-1 sample and class-2 samples respectively, and $\widehat{P_1}$ and $\widehat{P_2}$ are the estimated a priori probabilities. The function $\hat{h}(x_i)$ is the estimated log-likelihood ratio at case t_i obtained from estimating the posterior probabilities with any of the nonparametric classification methods (Section 1.4). In the case of $c_{12} = c_{21} = 1$, the so-called "0-1 loss function", the risk is called simply the error rate or (Probability of Misclassification (PMC).

The two components, $1 - \widehat{FNF}$ and $\widehat{FPF}$ give one point on the empirical (estimated) ROC curve. To draw the complete curve in the nonparametric situation, the estimated log-likelihood is calculated for each point of the available data set. Then

all possible thresholds are considered in turn, i.e., the threshold values between every two successive estimated log-likelihood values. At each threshold value a point on the ROC curve is calculated. Then the AUC can be calculated numerically from the empirical ROC curve using the trapezoidal rule:

$$\widehat{AUC} = \frac{1}{2}\sum_{i=2}^{n_{th}} (FNF_i - FNF_{i-1})(TPF_i + TPF_{i-1}) \tag{1.73}$$

where n_{th} is the number of threshold values taken over the data set. By plotting the empirical ROC curve, it is easy to see that the AUC obtained from the trapezoidal method is the same as the Mann-Whitney statistic—which is another form of the Wilcoxon rank-sum test (H\'{a}jek, \v{S}id\'{a}k and Sen, 1999, Ch.4)—defined by:

$$\widehat{AUC} = \frac{1}{n_1 n_2}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} \psi\left(\hat{h}(x_i|\omega_1), \hat{h}(x_j|\omega_2)\right), \tag{1.74}$$

$$\psi(a,b) = \begin{cases} 1 & a > b \\ 1/2 & a = b \\ 0 & a < b \end{cases} \tag{1.75}$$

The equivalence of the area under the empirical ROC and the Mann-Whitney-Wilcoxon statistic is the basis of its use in the assessment of diagnostic tests; see Hanley and McNeil (1982). Swets (1986) has recommended it as a natural summary measure of detection accuracy on the basis of signal-detection theory. Applications of this measure are widespread in the literature on human and computer-aided diagnosis in medical imaging, e.g., Jiang et al. (1999). In the field of machine learning, Bradley (1997) has recommended it as the preferred summary measure of accuracy when a single number is desired. These references also provide general background and access to the large literature on the subject.

It has been mentioned above that in the nonparametric situation these metrics are estimated from a single given data set, i.e., the testing data set or, less formally, the testers. But as long as the distribution is unknown it is not only impossible to calculate these metrics parametrically, but it is also impossible to generate, by simulation, testing data sets on which these metrics can be estimated. In that case the classifier might be trained and its performance metric estimated from the same training data set. This metric will be a random variable whose randomness comes from the finite training data set $\mathbf{t}$. That is, under different data sets even of the same size, the metric will vary. Therefore it is not sufficient to assess a classifier performance by estimating its mean, either error or AUC, without estimating the variability of that metric. The central content of this dissertation is about assessing the classifier performance in the mean and variance. Chapter 2 is and introduction to the different statistical methods available in the literature for nonparametric methods of estimating mean and variance.

CHAPTER 2

Nonparametric Estimation and Assessment: Literature Review

2.1. Nonparametric methods for Bias and Variance Estimation

In Section 1.9 it has been explained that in the nonparametric situation, the classifier performance must be estimated from a given data set, i.e., a data set available for testing. Now, the performance of the classifier is measured using a certain metric, e.g., the error rate or, alternatively, the AUC. Either of these metrics is a statistic since it is function of a sample from a distribution. The most accessible measures of the behavior of either of these statistics are the mean and variance. The present chapter introduces the main nonparametric methods for estimating the mean and variance of any statistic. Then the special case where the statistic is the error rate of a classification rule is discussed and the state-of-the-art methods in the literature are explained for that particular performance metric.

Assume that there is a statistic s that is a function of a data set $\mathbf{x}: \{x_i, i = 1,2,\ldots,n\}$, where $x_i \overset{i.i.d.}{\sim} F$. The statistic s is now a random variable and its variability comes from the variability of x_i. Assume that this statistic is used to estimate a real-valued parameter $\theta = f(F)$. Then $\hat{\theta} = s(\mathbf{x})$ has expected value $E[s(\mathbf{x})]$ and variance $\text{Var}[s(\mathbf{x})]$. The mean square error of the estimator $\hat{\theta}$ is defined as:

$$MSE_{\hat{\theta}} = E\left[\hat{\theta} - \theta\right]^2 \tag{2.1}$$

The bias of the estimator $\hat{\theta} = s(\mathbf{x})$ is defined by the difference between the true value of the parameter and the expectation of the estimator, i.e.,

$$bias_F = bias_F\left(\hat{\theta}, \theta\right) = E_F[s(\mathbf{x})] - f(F) \tag{2.2}$$

Then the MSE in (2.1) can be rewritten as:

$$MSE_{\hat{\theta}} = bias_F^2\left(\hat{\theta}\right) + \text{Var}\left[\hat{\theta}\right] \tag{2.3}$$

A critical question is whether the bias and variance of the statistic s in (2.3) may be estimated from the available data set and, if so, how?

2.1.1. Bootstrap Estimate

The bootstrap was introduced by Efron (1979) to estimate the standard error of a statistic. The bootstrap mechanism is implemented by treating the current data set $\mathbf{x}$ as a representation for the population distribution F; i.e., approximating the distribution F by the MLE defined in (1.66). Then B bootstrap samples are drawn from that empirical distribution. Each bootstrap replicate is of size n, the same size as $\mathbf{x}$, and is obtained by sampling with replacement. Then in a bootstrap replicate some case x_i, in general, will appear more than once at the expense of another x_j that will not appear. The original data set will be treated now as the population, and the replicates will be treated as samples from the population. This situation is illustrated in Figure 2.1. Therefore, the bootstrap estimate of bias is defined to be:

$$bias_{\hat{F}}(\hat{\theta}) = \hat{\theta}^*(\cdot) - \hat{\theta}, \tag{2.4}$$

$$\hat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^{*b}, \tag{2.5}$$

$$\hat{\theta}^{*b} = s(\mathbf{x}^{*b}), \tag{2.6}$$

$$\hat{\theta} = s(\mathbf{x}) \tag{2.7}$$

The bootstrap estimate of standard error of the statistic $\hat{\theta}(\mathbf{x})$ is defined by:

$$\widehat{SE}_B = \left\{\sum_{b=1}^{B}\left[\hat{\theta}^{*b} - \hat{\theta}^*(\cdot)\right]^2/(B-1)\right\}^{1/2} \tag{2.8}$$

Either in estimating the bias or the standard error, the larger the number of bootstraps the closer the estimate to the asymptotic value. Said differently:

$$\lim_{B\to\infty} \widehat{SE}_B(\hat{\theta}^*) = SE_{\hat{F}}(\hat{\theta}^*) \tag{2.9}$$

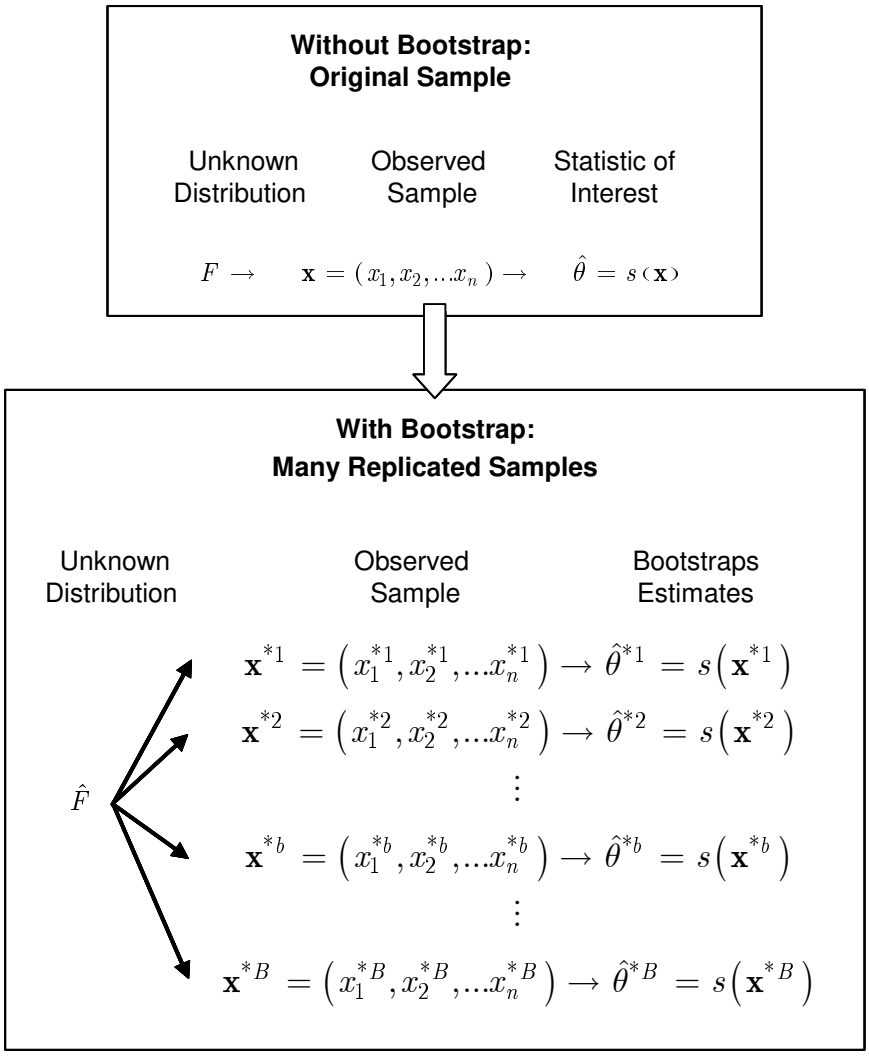

Figure 2.1. Bootstrap mechanism: B bootstrap replicates are withdrawn from the original sample. From each replicate the statistic is calculated.

For more details and some examples the reader is referred to Efron and Tibshirani (1993, Ch. 6, 7, and 10)

2.1.2. Jackknife Estimate

Instead of replicating from the original data set, a new set $\mathbf{x}_{(i)} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ is created by removing the case x_i from the data set. Then the jackknife samples are defined by:

$$\mathbf{x}_{(i)} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), i = 1, \ldots, n \tag{2.10}$$

and the n-jackknife replications of the statistic $\hat{\theta}$ are:

$$\hat{\theta}_{(i)} = s(\mathbf{x}_{(i)}),\ i = 1, \ldots, n \tag{2.11}$$

The jackknife estimates of bias and standard error are defined by:

$$\widehat{bias}_{jack} = (n-1)(\hat{\theta}(\cdot) - \hat{\theta}) \tag{2.12}$$

$$\widehat{SE}_{jack} = \left[\frac{n-1}{n}\sum_{i=1}^{n}(\hat{\theta}_{(i)} - \hat{\theta}(\cdot))^2\right]^{1/2}, \tag{2.13}$$

$$\hat{\theta}(\cdot) = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_{(i)} \tag{2.14}$$

For motivation behind the factors $(n-1)$ and $(n-1)/n$ in (2.12) see Efron and Tibshirani (1993, Ch. 11). The jackknife estimate of variance is discussed in detail in Efron (1981) and Efron and Stein (1981).

2.1.3. Bootstrap vs. Jackknife

Usually it requires up to 200 bootstraps to yield acceptable bootstrap estimates (in special situations like estimating the uncertainty in classifier performance it may take up to thousands of bootstraps). Hence, this requires calculating the statistic $\hat{\theta}$ the same number of times B. In the case of the jackknife, it requires only n calculations as shown in (2.11). If the sample size is smaller than the required number of bootstraps, the jackknife is more economical in terms of computational cost.

In terms of accuracy, the jackknife can be seen to be an approximation to the bootstrap when estimating the standard error of a statistic; see Efron and Tibshirani (1993, Ch. 20). Thus, if the statistic is linear they almost give the same result (The bootstrap gives the jackknife estimate multiplied by $[(n-1)/n]^{1/2}$. A statistic $s(\mathbf{x})$ is said to be linear if:

$$s(\mathbf{x}) = \mu + \frac{1}{n}\sum_{i=1}^{n}\alpha(x_i), \tag{2.15}$$

where μ is a constant and $\alpha(\cdot)$ is a function. This also can be viewed as having one data point at a time in the argument of the function α. Similarly, the jackknife can be seen as an approximation to the bootstrap when estimating the bias. If the statistic is quadratic, they almost agree except in a normalizing factor . A statistic $s(\mathbf{x})$ is quadratic if:

$$s(\mathbf{x}) = \mu + \frac{1}{n}\sum_{1\le i\le n}\alpha(x_i) + \frac{1}{n^2}\sum_{1\le i<j\le n}\beta(x_i,x_j) \tag{2.16}$$

An in-depth treatment of the bootstrap and jackknife and their relation to each other in mathematical detail is provided by Efron (1982, Ch. 1-5).

If the statistic is not smooth the jackknife will fail. Informally speaking, a statistic is said to be smooth if a small change in the data leads to a small change in the statistic. An example of a non-smooth statistic is the median. If the sample cases are ranked and the median is calculated, it will not change when a sample case changes unless this sample case bypasses the median value. An example of a smooth statistic is the sample mean.

2.1.4. Influence Function, Infinitesimal Jackknife, and Estimate of Variance

The infinitesimal jackknife was introduced by Jaeckel (1972). The concept of the influence curve was introduced later by Hampel (1974). In the present context and for pedagogical purposes, the influence curve will be explained before the infinitesimal jackknife, since the former can be understood as the basis for the latter.

Following Hampel (1974), let $\Re$ be the real line and s be a real-valued functional defined on the distribution F which is defined on $\Re$. The distribution F can be perturbed by adding some probability measure (mass) on a point x. This should be balanced by a decrement in F elsewhere, resulting in a new probability distribution $G_{\varepsilon,x}$ defined by:

$$G_{\varepsilon,x} = (1-\varepsilon)F + \varepsilon\delta_x,\ x \in \Re \tag{2.17}$$

Then, the influence curve $IC_{s,F}(\cdot)$ is defined by:

$$IC_{s,F}(x) = \lim_{\varepsilon\to 0^+}\frac{s\left((1-\varepsilon)F + \varepsilon\delta_x\right) - s(F)}{\varepsilon} \tag{2.18}$$

It should be noted that F does not have to be a discrete distribution. A simple example of applying the influence curve concept is to consider the expectation $s = \int x\, dF(x) = \mu$. Substituting back in (2.18) gives:

$$IC_{s,F}(x) = x - \mu \tag{2.19}$$

The meaning of this formula is the following: the rate of change of the functional s with the probability measure at a point x is $x-\mu$. This is how the point x influences the function s.

The influence curve can be used to linearly approximate a functional s; this is similar to taking up to only the first-order term in a Taylor series expansion. Assume that there is a distribution G near to the distribution F; then under some regularity conditions (see, e.g., Huber, 1996, Ch. 2) a functional s can be approximated as:

$$s(G) \approx s(F) + \int IC_{s,F}(x)\, dG(x) \tag{2.20}$$

The residual error can be neglected since it is of a small order in probability. Some properties of (2.20) are:

$$\int IC_{T,F}(x)\, dF(x) = 0 \tag{2.21}$$

and the asymptotic variance of $s(F)$ under F, following from (2.21), is given by:

$$\mathrm{Var}_F[s(F)] \simeq \int \left\{IC_{T,F}(x)\right\}^2\, dF(x) \tag{2.22}$$

which can be considered as an approximation to the variance under a distribution G near to F. Now, assume that the functional s is a functional statistic in the data set $\mathbf{x} = \{x_i : x_i \sim F, i = 1,2,\cdots,n\}$. In that case the influence curve (2.18) is defined for each

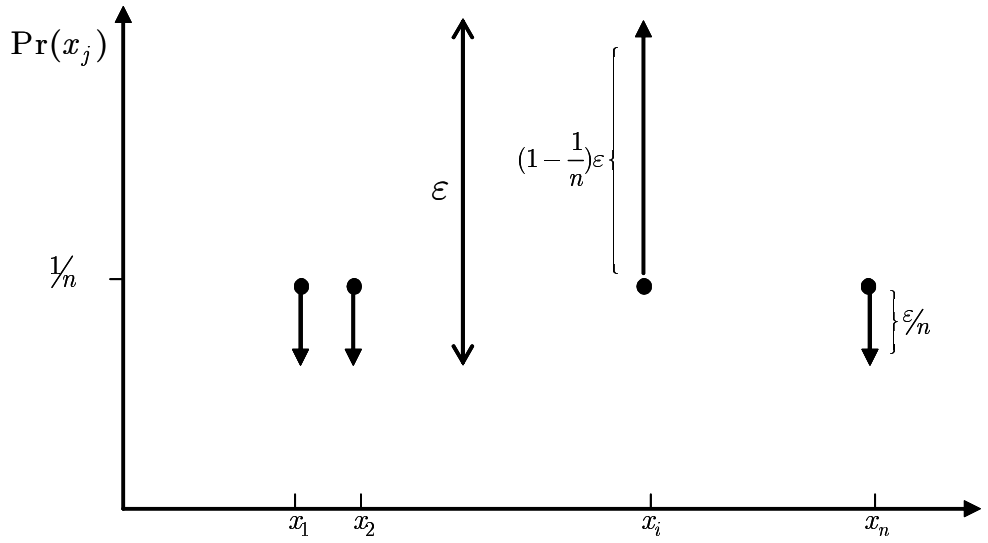

Figure 2.2. The new probability masses for the data set X under a perturbation at sample case x_i obtained by letting the new probability at x_i exceed the new probability at any other case x_i by ε

sample case x_i, under the true distribution F as:

$$U_i(s,F) = \lim_{\varepsilon\to 0} \frac{s(F_{\varepsilon,i}) - s(F)}{\varepsilon} = \left.\frac{\partial s(F_{\varepsilon,i})}{\partial \varepsilon}\right|_{\varepsilon=0}, \tag{2.23}$$

where $F_{\varepsilon,i}$ is the distribution under the perturbation at observation x_i. In the sequel (2.23) will be called the influence function. If the distribution F is not known, the MLE $\hat{F}$ of the distribution F is given by (1.66), and as an approximation $\hat{F}$ may substitute for F in (2.23); the result may then be called the empirical influence function, Mallows (1974), or infinitesimal jackknife Jaeckel (1972). In such an approximation, the perturbation defined in (2.17) can be rewritten as:

$$\hat{F}_{\varepsilon,i} = (1-\varepsilon)\hat{F} + \varepsilon\delta_{x_i},\ x_i \in \mathbf{x},\ i = 1,\ldots,n \tag{2.24}$$

This kind of perturbation is illustrated in Figure 2.2. It will often be useful to write the probability mass function of (2.24) as:

$$\hat{f}_{\varepsilon,i}(x_j) = \begin{cases} \frac{1-\varepsilon}{n} + \varepsilon & j = i \\ \frac{1-\varepsilon}{n} & j \neq i \end{cases} \tag{2.25}$$

Substituting $\hat{F}$ for G in (2.20) and combining the result with (2.23) gives the influence-function approximation for any functional statistic under the empirical distribution $\hat{F}$. The result is:

$$s(\hat{F}) = s(F) + \frac{1}{n}\sum_{i=1}^{n} U_i(s,F) + O_p(n^{-1}) \tag{2.26}$$

$$\approx s(F) + \frac{1}{n}\sum_{i=1}^{n} U_i(s,F) \tag{2.27}$$

The term $O_p(n^{-1})$ reads "big-O of order $1/n$ in probability". In general, $U_n = O_p(d_n)$ if U_n/d_n is bounded in probability, i.e., $\Pr\{|U_n|/d_n < k_\varepsilon\} > 1-\varepsilon\ \forall\ \varepsilon > 0$. This concept can be found in Barndorff-Nielsen and Cox (1989, Ch. 2). Then the asymptotic variance expressed in (2.22) can be given for $s(F)$ by:

$$\mathrm{Var}_F[s] = \frac{1}{n} E_F\left[U^2(x_i,F)\right] \tag{2.28}$$

which can be approximated under the empirical distribution $\hat{F}$ by:

$$\widehat{\mathrm{Var}}_{\hat{F}}[s] = \frac{1}{n^2}\sum_{i=1}^{n} U_i^2(x_i,\hat{F}) \tag{2.29}$$

It is important to state here that s should be a functional in $\hat{F}$ that is an approximation to F, as was initially assumed in (2.18). If for example the value of the statistic s changes if every sample case x_i is duplicated, i.e., repeated twice, this is not a functional statistic. An example of a functional statistic is the biased version of the variance estimate $\Sigma_i(x_i-\bar{x}_i)^2/n$, while the unbiased version $\Sigma_i(x_i-\bar{x}_i)^2/(n-1)$ is not a functional statistic. Generally, any approximation $s(\hat{F})$ to the functional $s(F)$, by approximating F by the MLE $\hat{F}$, obviously will be functional. In such a case the statistic $s(\hat{F})$ is called the plug-in estimate of the functional $s(F)$. Moreover, the influence function method for variance estimation is applicable only to those functional statistics whose derivative (2.23) exists. If that derivative exists, the statistic is called a smooth statistic; i.e., a small change in the data set leads

a small change in the statistic. For instance, the median is a functional statistic in the sense that duplicating any sample case will result in the same value of the median. On the other hand it is not smooth as described at the end of Section 2.1.3. A key reference for the influence function is Hampel (1986).

Equation (2.29) gives the nonparametric estimate of variance for a statistic s under the empirical distribution $\hat{F}$; this equation will be the basis of subsequent work in this dissertation. A very interesting case arises from (2.25) if $-1/(n+1)$ is substituted for ε. In this case the new probability mass assigned to the point $x_{j=i}$ in (2.25) will be zero. This value of ε simply generates the jackknife estimate discussed in Section 2.1.2 where the whole point is removed from the data set.

2.2. Estimating the Mean Performance of a Classification Rule

In the previous section the statistic, or generally speaking the functional, was a function of just one data set. For a non-fixed design, i.e., the predictors for the testing set do not have to be the same as the predictors of the training set, a slight clarification for the previous notations is needed. The classification rule trained on the training data set $\mathbf{t}$ will be denoted as $\eta_{\mathbf{t}}$. Any new observation that does not belong to $\mathbf{t}$ will be denoted by $t_0 = (x_o, y_o)$. Therefore the loss due to classification is given by $L(y_0, \eta_{\mathbf{t}}(x_0))$. Any metric conditional on that training data set will be similarly subscripted. Thus, the risk (1.56), the error rate whose two components are (1.60), and the area under the curve (1.61) should be denoted by $R_{\mathbf{t}}$, $Err_{\mathbf{t}}$, and $AUC_{\mathbf{t}}$, respectively. In the rest of the present chapter, for simplicity and without loss in generality, the 0-1 loss function will be used. In such a case the conditional error rate will be given by:

$$Err_{\mathbf{t}} = E_{0F}\left[L\left(y_0, \eta_{\mathbf{t}}(x_0)\right)\right], \left(x_0, y_0\right) \sim F \tag{2.30}$$

The expectation E_{0F} is subscripted so to emphasize that it is taken over the observations $t_0 \notin \mathbf{t}$. If the performance is measured in the error rate and we are interested in the mean performance, not the conditional one, then it is given by:

$$Err = E_{\mathbf{t}}\left[Err_{\mathbf{t}}\right] \tag{2.31}$$

where $E_{\mathbf{t}}$ is the expectation over the training set $\mathbf{t}$, which would be the same if we had written E_F; for notation clarity the former is chosen.

This section now picks up where Section 1.9 ended and assumes the existence of a classification rule already trained on a training data set, $\eta_{\mathbf{t}}$. A natural next question is, given that there is just a single data set available, how to use this data set in assessing the classifier performance as well? Said differently, how should one estimate, using only the available data set, the classification performance of a classification rule in predicting new observations; these observations are different from those on which the rule was trained. In this section the principal methods in the literature for estimating the mean and variance of the performance of a classification rule are introduced. The performance metric here will be the error rate. Different estimators are proposed in the literature. Later, in Chapter 3, estimating the classification performance using the AUC as a metric will be explained in detail, since it is the contribution of this dissertation.

2.2.1. Apparent Error

The apparent error, or residual error in regression, is the error of the fitted model when it is tested on the same training data. Of course it is downward biased with respect to the true error rate since it results from testing on the same information used in training (Efron, 1986). The apparent error is defined by:

$$\overline{Err}_{\mathbf{t}} = E_{\hat{F}} L(y, \eta_{\mathbf{t}}(x)), (x, y) \in \mathbf{t} \tag{2.32}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(I_{\hat{h}_{\mathbf{t}}(x_i|\omega_1)<th} + I_{\hat{h}_{\mathbf{t}}(x_i|\omega_2)>th}\right) \tag{2.33}$$

Over-designing a classifier to minimize the apparent error is not the goal. The goal is to minimize the true error rate (2.30).

2.2.2. Variance-Bias Trade-off

Over-training, over-designing, or overfitting are all synonyms. The more trained the classifier, conditional on the same data set size, the more complex it is. This can be better understood in terms of the smoothing techniques discussed in Section 1.4.1. The smaller the window size of the smoother, the more complex it will be. An extreme example of an over-trained classifier is the 1-nearest neighbor classifier. In this case, the window size tends to zero. Hastie and Tibshirani (1990, Ch. 3) review a measure of the complexity of smoothing functions in terms of an effective number of degrees of freedom. This overfitting decreases the apparent error; but what is its effect on the true error rate?

There is always a trade-off between the bias and variance of the measure of performance of the classification rule. Over-training a particular classifier decreases its bias and increases its variance; and vice versa. This can be best understood if the

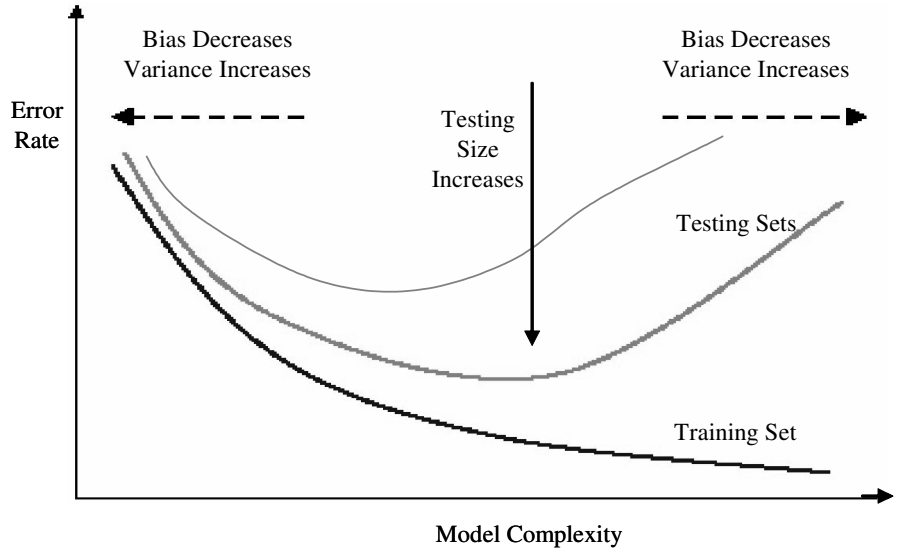

Figure 2.3. True error rate versus model complexity (or overtraining). The apparent error rate improves with the model complexity. The true error rate is minimized at a corresponding optimal model complexity. The two metrics asymptote with the training set size to the same value.

k-NN smoother is considered (Section 1.4.1.1). At point x_i the prediction is $\Sigma_{j\in N_K(x_i)} y_j/k$. The expectation of this regression function is $\Sigma_{j\in N_K(x_i)} E[y_j]/k$, while the variance will be σ^2/k, where the response is assumed to have constant variance σ^2 with the predictor. If the window size of this rule is squeezed to produce a more complex rule, i.e., k is decreased, the variance will increase. But the bias will decrease since $\Sigma_{j\in N_K(x_i)} E[y_j]/k$ tends to approach $E[y_i]$. On the contrary, increasing k obviously decreases the variance, while incorporating many data points whose expectations will be very likely to vary from $E[y_i]$, hence the bias increases. The relationship between the model complexity and the error rate is illustrated in Figure 2.3. Hughes (1968) first carried out the required computations displayed in that figure.

2.2.3. Cross Validation

The basic concept of cross validation (CV) has been proposed in different articles since the mid-1930s. The concept simply leans on splitting the data into two parts; the first part is used in design without any involvement of the second part. Then the second part is used to test the designed procedure; this is to test how the designed procedure will behave for new data sets. Stone (1974) is a key reference for CV that proposes different criteria for optimization.

Cross-validation can be used to assess the prediction error of a model or in model selection. In this section the former is discussed, since assessing classifiers is the interest of this work rather than designing classifiers. The true error rate in (2.30) is the expected error rate for a classification rule if tested on the population, conditional on a particular training data set $\mathbf{t}$. This metric can be approximated by leave-one-out cross-validation by:

$$\widehat{Err}_{\mathbf{t}}^{cv1} = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{(i)}}(x_i)\right),\ (x_i, y_i) \in \mathbf{t} \tag{2.34}$$

This is done by training the classification rule on the data set $\mathbf{t}^{(i)}$ that does not include the case t_i; then testing the trained rule on that omitted case. This proceeds in "round-robin" fashion until all cases have contributed one at a time to the error rate. There is a hidden assumption in this mechanism: the training set $\mathbf{t}$ will not change very much by omitting a single case. Therefore, testing on the omitted points one at a time accounts for testing approximately the same trained rule on n new cases, all different from each other and different from those the classifier has been trained on. Besides this leave-one-out cross-validation, there are other versions named k-fold (or leave-n/k-out). In such versions the whole data set is split into k roughly equal-sized subsets, each of which contains approximately n/k observations. The classifier is trained on $k-1$ subsets and tested on the left-out one; hence we have k iterations.

It is of interest to assess this estimator to see if it estimates the conditional true error with small mean square error (MSE) $E\left[\widehat{Err}_{\mathbf{t}}^{cv1} - Err_{\mathbf{t}}\right]^2$. Many simulation results, e.g., Efron (1983), show that there is only a very weak correlation between the cross validation estimator and the conditional true error rate $\widehat{Err}_{\mathbf{t}}$. This issue is discussed in mathematical detail in the excellent paper by Zhang (1995). Other estimators to be discussed below are shown to have this same attribute. This very interesting (and perhaps surprising) result will be revisited in more detail in Chapter 3.

2.2.4. Bootstrap Methods for Estimation of Error Rate

The prediction error in (2.30) is a function of the training data set $\mathbf{t}$ and the testing population F. Bootstrap estimation can be implemented here by treating the empirical distribution $\hat{F}$ as an approximation to the actual population distribution F; by replicating from that distribution one can simulate the case of many training data sets $\mathbf{t}_b$, $b = 1,\ldots,B$, the total number of bootstraps. For every replicated training data set the classifier will be trained and then tested on the original data set $\mathbf{t}$. This is the simple bootstrap estimator approach (Efron and Tibshirani, 1993, Sec. 17.6) defined by:

$$\widehat{Err}_{\mathbf{t}}^{SB} = E_* \sum_{i=1}^{n} L(y_i, \eta_{\mathbf{t}^*}(x_i))/n,\ \hat{F} \rightarrow \mathbf{t}^* \tag{2.35}$$

It should be noted that this estimator no longer estimates the true error rate (2.30) because the expectation taken over the bootstraps mimics an expectation taken over the population of trainers, i.e., it is not conditional on a particular training set. Rather, the estimator (2.35) estimates the expected performance of the classifier $E_F Err_{\mathbf{t}}$, which is a constant metric, not a random variable any more. For a finite number of bootstraps the expectation (2.35) can be approximated by:

$$\widehat{Err}_{\mathbf{t}}^{SB} = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/n \tag{2.36}$$

2.2.4.1. Leave-One-Out Bootstrap

The last estimator is obviously biased since the original data set $\mathbf{t}$ used for testing includes part of the training data in every bootstrap replicate. Efron (1983) proposed that, after training the classifier on every bootstrap replicate, it is tested on those cases in the set $\mathbf{t}$ that are not included in the training; this concept can be developed as follows. Equation (2.36) can be rewritten by interchanging the order of the double summation to give:

$$\widehat{Err}_{\mathbf{t}}^{SB} = \frac{1}{n}\sum_{i=1}^{n}\sum_{b=1}^{B} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/B \tag{2.37}$$

This equation is formally identical to (2.36) but it expresses a different mechanism for evaluating the same quantity. It says that, for a given point, the average performance over the bootstrap replicates is calculated; then this performance is averaged over all the n cases. Now, if every case t_i is tested only from those bootstraps that did not include it in the training, a slight modification of the previous expression yields the leave-one-out bootstrap estimator:

$$\widehat{Err}_{\mathbf{t}}^{(1)} = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{b=1}^{B} I_i^b L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/\sum_{b'=1}^{B} I_i^{b'}\right] \tag{2.38}$$

where the indicator function I_i^b equals one when the case t_i is not included in the training replicate b, and zero otherwise. To simplify notation, the error $L(y_i, \eta_{\mathbf{t}^{*b}}(x_i))$ may be denoted by L_i^b. Efron and Tibshirani (1997) emphasized a critical point about the difference between this bootstrap estimator and leave-one-out cross validation. Cross-validation tests on a given sample case t_i, having been trained just once on the remaining data set. By contrast, the leave-one-out bootstrap tests on a given sample case t_i using a large number of classifiers that result from a large number of bootstrap replicates that do not contain that sample. This results in a smoothed cross-validation-like estimator. To see how the conventional cross-validation estimator is unsmooth, consider the decision surface in the feature subspace that results from training the classifier on the data set that remains after leaving out the case t_i. Whenever the predictor x_i changes, the loss function will not change unless the predictor passes across the decision surface. Training on many data sets results in many decision surfaces and then whenever the predictor x_i changes it will tend to cross some of the surfaces, yielding a smoother estimator, rather than the discontinuous one that results from cross-validation.

2.2.4.2. The Refined Bootstrap

The simple bootstrap and the leave-one-out bootstrap can be shown to estimate the mean true error rate for a classifier. This mean is with respect to the population of all training data sets. For estimating the true error rate of a classifier, conditional on a particular training data set, Efron (1983) proposed to correct for the downward biased estimator $\overline{Err}_{\mathbf{t}}$. Since the true error rate $Err_{\mathbf{t}}$ can be written as $\overline{Err}_{\mathbf{t}} + (Err_{\mathbf{t}} - \overline{Err}_{\mathbf{t}})$, then it can be approximated by $\overline{Err}_{\mathbf{t}} + E_F(Err_{\mathbf{t}} - \overline{Err}_{\mathbf{t}})$. The term $Err_{\mathbf{t}} - \overline{Err}_{\mathbf{t}}$ is called the optimism. The expectation of the optimism can be approximated over the bootstrap population. Finally the refined bootstrap approach, as named in Efron and Tibshirani (1993, Sec. 17.6), gives the estimator:

$$\widehat{Err}_{\mathbf{t}}^{RF} = \overline{Err}_{\mathbf{t}} + E_*(Err_{\mathbf{t}*}(\hat{F}) - \overline{Err}_{\mathbf{t}*}) \tag{2.39}$$

where $Err_{\mathbf{t}*}(\hat{F})$ represents the error rate obtained from training the classifier on the bootstrap replicate $\mathbf{t}^*$ and testing on the empirical distribution $\hat{F}$. This can be approximated for a limited number of bootstraps by:

$$\widehat{Err}_{\mathbf{t}}^{RF} = \overline{Err}_{\mathbf{t}} + \frac{1}{B}\sum_{b=1}^{B}\left\{\sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/n - \sum_{i=1}^{n} L\left(y_{ib}^*, \eta_{\mathbf{t}^{*b}}(x_{ib}^*)\right)/n\right\} \tag{2.40}$$

2.2.4.3. The 0.632 Bootstrap

If the concept used in developing the leave-one-out bootstrap estimator, i.e., testing on cases not included in training, is used again in estimating the optimism described above, this gives the 0.632 bootstrap estimator. Since the probability of including a case t_i in the bootstrap $\mathbf{t}^{*b}$ is given by:

$$\Pr(t_i \in \mathbf{t}^{*b}) = 1 - (1 - 1/n)^n \tag{2.41}$$

$$\approx 1 - e^{-1} = 0.632 \tag{2.42}$$

the effective number of sample cases contributing to a bootstrap replicate is approximately 0.632 of the size of the training data set. Efron (1983) introduced the concept of a *distance* between a point and a sample set in terms of a probability. Having trained on a bootstrap replicate, testing on those cases in the original data set not included in the bootstrap replicate accounts for testing on a set far from the training one, i.e., the bootstrap replicate. This is because every sample case in the testing set has zero probability of belonging to the training set, i.e., very distant from the training set. This is a reason for why the leave-one-out is upward biased estimator. Efron (1983) showed roughly that :

$$E_F\left\{Err_{\mathbf{t}} - \overline{Err}_{\mathbf{t}}\right\} \approx 0.632\, E_F\left\{\widehat{Err}_{\mathbf{t}}^{(1)} - \overline{Err}_{\mathbf{t}}\right\} \tag{2.43}$$

Substituting back in (2.39) gives the 0.632 estimator:

$$\widehat{Err}_{\mathbf{t}}^{(.632)} = .368\,\overline{Err}_{\mathbf{t}} + .632\,\widehat{Err}_{\mathbf{t}}^{(1)} \tag{2.44}$$

The proof of the above results can be found in Efron (1983) and Efron and Tibshirani (1993, Sec. 6).

The motivation behind this estimator as stated earlier is to correct for the downward biased apparent error by adding a piece of the upward biased leave-one-out-bootstrap estimator. But an increase in variance should be expected as a result of adding this piece of the relatively variable apparent error. Moreover, this new estimator is no longer smooth since the apparent error itself is unsmooth.

2.2.4.4. The 0.632+ Bootstrap Estimator

The .632 estimator reduces the bias of the apparent error. But for over-trained classifiers, i.e., those whose apparent error tends to be zero, the .632 estimator is still downward biased. Breiman et al. (1984) provided the example of an over-fitted rule, like 1-nearest neighbor where the apparent error is zero. If, however, the class labels are assigned randomly to the predictors the true error rate will obviously be 0.5. But substituting in (2.44) gives the .632 estimate of .632 × .5 = .316. To account for this bias for such over-fitted classifiers, Efron and Tibshirani (1997) defined the *no-information error rate* γ by:

$$\gamma = E_{oF_{ind}}\left[L\left(y_0, \eta_{\mathbf{t}}(x_0)\right)\right] \tag{2.45}$$

where F_{ind} means that x_0 and y_0 are distributed marginally as F but they are independent. Or said differently, the label is assigned randomly to the predictor. Then for a training sample $\mathbf{t}$ γ can be estimated by:

$$\hat{\gamma} = \sum_{i=1}^{n}\sum_{j=1}^{n} L\left(y_i, \eta_{\mathbf{t}}(x_j)\right)/n^2 \tag{2.46}$$

This means that the n predictors have been permuted with the n responses to produce n^2 non-informative cases. In the special case of binary classification, let $\hat{p}_1$ be the proportion of the response classified as belonging to class 1. Also, let $\hat{q}_1$ be the proportion of the responses classified as belonging to class 1. Then (2.46) reduces to:

$$\hat{\gamma} = \hat{p}_1(1 - \hat{q}_1) + (1 - \hat{p}_1)\hat{q}_1 \tag{2.47}$$

Also define the *relative overfitting rate*:

$$\hat{R} = \frac{\widehat{Err}_{\mathbf{t}}^{(1)} - \widehat{Err}_{\mathbf{t}}}{\hat{\gamma} - \widehat{Err}_{\mathbf{t}}} \tag{2.48}$$

Efron and Tibshirani (1997) showed that the bias of the .632 estimator for the case of over-fitted classifiers is alleviated by using a renormalized version of that estimator:

$$\widehat{Err}_{\mathbf{t}}^{(.632+)} = (1 - \hat{w})\overline{Err}_{\mathbf{t}} + \hat{w}\widehat{Err}_{\mathbf{t}}^{(1)}, \tag{2.49}$$

$$\hat{w} = \frac{.632}{1 - .368\hat{R}} \tag{2.50}$$

It is useful to express the .632+ estimator in terms of its predecessor, the .632 estimator. Combining (2.44), (2.47), and (2.48) then substituting in (2.49) yields:

$$\widehat{Err}_{\mathbf{t}}^{(.632+)} = \widehat{Err}_{\mathbf{t}}^{(.632)} + (\widehat{Err}_{\mathbf{t}}^{(1)} - \overline{Err}_{\mathbf{t}})\frac{.368 \cdot .632 \cdot \hat{R}}{1 - .368\hat{R}} \tag{2.51}$$

Efron and Tibshirani (1997) consider the possibility that $\hat{R}$ lies out of the region $[0,1]$. This leads to their proposal of defining:

$$\widehat{Err}_{\mathbf{t}}^{(1)\prime} = \min(\widehat{Err}_{\mathbf{t}}^{(1)}, \hat{\gamma}), \tag{2.52}$$

$$\hat{R}' = \begin{cases} (\widehat{Err}_{\mathbf{t}}^{(1)} - \overline{Err}_{\mathbf{t}})/(\hat{\gamma} - \overline{Err}_{\mathbf{t}}) & \overline{Err}_{\mathbf{t}} < \widehat{Err}_{\mathbf{t}}^{(1)} < \gamma \\ 0 & otherwise \end{cases} \tag{2.53}$$

to obtain a modification to (2.51) that becomes:

$$\widehat{Err}_{\mathbf{t}}^{(.632+)} = \widehat{Err}_{\mathbf{t}}^{(.632)} + (\widehat{Err}_{\mathbf{t}}^{(1)\prime} - \overline{Err}_{\mathbf{t}}) \frac{.368 \cdot .632 \cdot \hat{R}'}{1 - .368\hat{R}'} \tag{2.54}$$

2.3. Estimating the Standard Error of $\widehat{Err}_{\mathbf{t}}^{(1)}$

What have been discussed above are different methods to estimate the error rate of a trained classification rule, e.g., cross validation, .632, .632+, conditional on that training set; alternatively, to estimate the mean error rate, as an expectation over the population of training data sets, like the leave-one-out bootstrap estimator. Regardless of what the estimator is designed to estimate, it is still a function of the current data set $\mathbf{t}$, i.e., it is a random variable. If $\widehat{Err}_{\mathbf{t}}^{(1)}$ is considered, it estimates a constant real-valued parameter $E_{0F}E_F L(y_0, \eta_{\mathbf{t}}(x_0))$ with expectation taken over all the trainers and then over all the testers, respectively; this is the overall mean error rate. Yet, $\widehat{Err}_{\mathbf{t}}^{(1)}$ is a random variable whose variability comes from the finite size of the available data set. If the classifier is trained and tested on a very large number of observations, this would approximate training and testing on the entire population, and the variability would shrink to zero. This also applies for any metric other than the error rate. Chapter 3 introduces new work in which the AUC is used as the summary metric of performance, where all the concepts and different estimators mentioned in the present chapter are applied.

Efron and Tibshirani (1997) proposed using the influence function method, see Section 2.1.4, to estimate the uncertainty (variability) in $\widehat{Err}_{\mathbf{t}}^{(1)}$. The reader is alerted that estimators that incorporate a piece of the apparent error are not suitable for the influence function method. Such estimators are not smooth because the apparent error is not smooth. By recalling the definitions of Section 2.1.4, $\widehat{Err}_{\mathbf{t}}^{(1)}$ is now the statistic $s(\hat{F})$. Define N_i^b to be the number of times the case t_i is included in the bootstrap b. Also, define the following notation:

$$l_\cdot^b = \frac{1}{n}\sum_{i=1}^{n} I_i^b L_i^b, \tag{2.55}$$

It has been proven in Efron and Tibshirani (1995) that the influence function of such an estimator is given by:

$$\left.\frac{\partial s(\hat{F}_{\varepsilon,i})}{\partial \varepsilon}\right|_{\varepsilon=0} = (2 + \frac{1}{n-1})(\hat{E}_i - \widehat{Err}_{\mathbf{t}}^{(1)}) + \frac{n\sum_{b=1}^{B}(N_i^b - \bar{N}_i)I_i^b}{\sum_{b=1}^{B} I_i^b} \tag{2.56}$$

Combining (2.29) and (2.56) give an estimation to the uncertainty in $\widehat{Err}_{\mathbf{t}}^{(1)}$. A very similar, but complete, proof will be given in Chapter 3 when estimating the uncertainty in the AUC. Critical comments and details are deferred to that chapter.

2.4. Comparative Study for Proposed Estimators

Efron (1983); Efron and Tibshirani (1997) provide comparisons of some performance measure for the proposed estimators. The latter reference shows an example of how a single estimate of the influence function agreed well with the uncertainty of the $\widehat{Err}_{\mathbf{t}}^{(1)}$ estimator obtained from Monte-Carlo (MC) simulation. In the same reference the authors ran many simulations considering a variety of classifiers and data distributions, as well as real data sets. The $\widehat{Err}_{\mathbf{t}}^{(.632+)}$ estimator is the least biased among them all, achieving its design goal. But it is also necessary to consider the RMS error, defined by Efron as:

$$RMS = E_{MC}\left\{\widehat{Err}_{\mathbf{t}} - Err_{\mathbf{t}}\right\}^2 \tag{2.57}$$

$$= \frac{1}{G}\sum_{g=1}^{G}\left\{\widehat{Err}_{\mathbf{t}_g} - Err_{\mathbf{t}_g}\right\}^2 \tag{2.58}$$

where $\widehat{Err}_{\mathbf{t}_g}$ is the estimator (any estimator) conditional on a training data set $\mathbf{t}_g$. $Err_{\mathbf{t}_g}$ is the true prediction error conditional on the same training data set. The number of MC trials, G, in his experiments was 200. The following statement is quoted from Efron and Tibshirani (1997): *"The results vary considerably from experiment to experiment, but in terms of RMS error the .632+ rule is an overall winner."* An extension to this study is done in this dissertation by considering the AUC as the summary metric, and a comment on the above statement will be given in Chapter 3.

CHAPTER 3

Introduction to the Work Done In This Dissertation: Nonparametric Approach of Classifier Assessment in Terms of ROC Curve

3.1. Introduction

The purpose of the present chapter is to introduce the work done in this dissertation, the nonparametric assessment in terms of the ROC curve. The new summary metric to be considered in the present chapter, and beyond, is the AUC (see Section 1.9). Before delving into our objective in this dissertation, i.e., the assessment task, it may be instructive to examine, analytically, what the log-likelihood ratio looks under the popular data population, the multinormal distribution. Consider two different classes, ω_1 and ω_2, whose p-dimensional feature vectors have the multinormal distributions F_1 and F_2 respectively, described by the PDFs:

$$f_X(x|\omega_i) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right],\ i=1,2 \tag{3.1}$$

The Bayes classifier is the optimal one; it has the minimum risk (see Section 1.2). The training for the Bayes classifier in the multinormal case requires only the estimation of the mean vectors μ_i's and the covariance matrices Σ_i's. Assume that the training set for ω_1 is $\mathbf{t}_1 = \{t_i : t_i = (x_i, \omega_1)\},\ i = 1,\ldots,n_1$, and the training set for ω_2 is $\mathbf{t}_2 = \{t_i : t_i = (x_i, \omega_2)\},\ i = 1,\ldots,n_2$. The estimates of the population parameters are given by:

$$\hat{\Sigma}_i = \frac{1}{n_i - 1}\left[\sum_{j=1}^{n_i}(x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)'\right], \tag{3.2}$$

$$\hat{\mu}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j,\ where\ x_j \in \omega_i \tag{3.3}$$

Anderson (2003). The log-likelihood function (1.57), in combination with the estimated parameters (3.2), assuming equal prevalence, for the two classes, and equal costs for the two kinds of errors, can be written as:

$$h(X) = -\frac{1}{2}\left[(X-\hat{\mu}_1)'\hat{\Sigma}_1^{-1}(X-\hat{\mu}_1) - (X-\hat{\mu}_2)'\hat{\Sigma}_2^{-1}(X-\hat{\mu}_2)\right] - \frac{1}{2}\ln\frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_2|} \tag{3.4}$$

It should be noted that if the a priori probabilities and costs, which form a particular threshold value of the testing environment, are known they should be included in the log-likelihood ratio, i.e., the log of the R.H.S. of inequality (1.17) should be added to the R.H.S. of (3.4). In that case, the classifier is designed to be used in this environment having that threshold. For demonstration, consider the population parameters to take the following values:

$$\mu_1 = \begin{pmatrix}2\\2\end{pmatrix},\ \Sigma_1 = \begin{pmatrix}1 & .2\\ .2 & 1\end{pmatrix}, \tag{3.5}$$

$$\mu_2 = \begin{pmatrix}1\\1\end{pmatrix},\ \Sigma_2 = \begin{pmatrix}.3 & .1\\ .1 & .3\end{pmatrix} \tag{3.6}$$

Consider that we train on a very large size of observations such that the estimated parameters are almost the same as the true ones. Under these parameters the two PDFs (3.1) of the two classes are shown in Figure 3.1. Two simulated data sets, one set for each class with 10,000 observations per class, are simulated from binormal distributions with the above parameters and illustrated in Figure 3.2. Under these parameter values, the log-likelihood ratio in (3.4) is plotted in Figure 3.3, while its contour plot is given in Figure 3.4. The locus separating the two classes in Figure 3.2 is obtained by solving $h(X) = 0$ in (3.4).

The log-likelihood ratio $h(X)$ in (3.4) is a function of the random vector $X = (x_1, x_2)'$. Consider the transformation $T : X \longrightarrow (h, d)$, where d is a dummy variable and set

$$d = x_1 \tag{3.7}$$

The dummy variable here is introduced just for clarity and we could have written $T : X \longrightarrow (h, x_1)$. We shall, no longer, refer to d; rather, we refer to its genuine x_1. This transformation is $1:2$, this produces two values of the new vector (h, x_1) at every value of the vector X. In other words, solving (3.4) and (3.7) for x_1 and x_2 gives two solutions. By adding these two solutions, and calculating the Jacobian of the transformation it can be shown that the joint PDF of $h(X)$ and x_1, under the assumption that $X \sim F_1$, is given by:

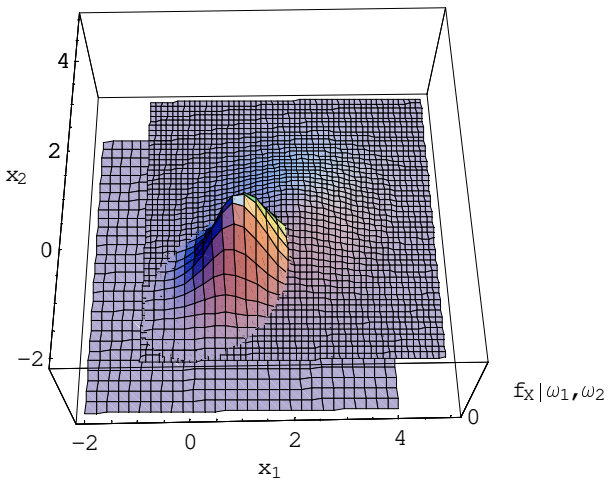

Figure 3.1. A 3-D illustration of Probability Density Function (PDF) of two binormal distributions

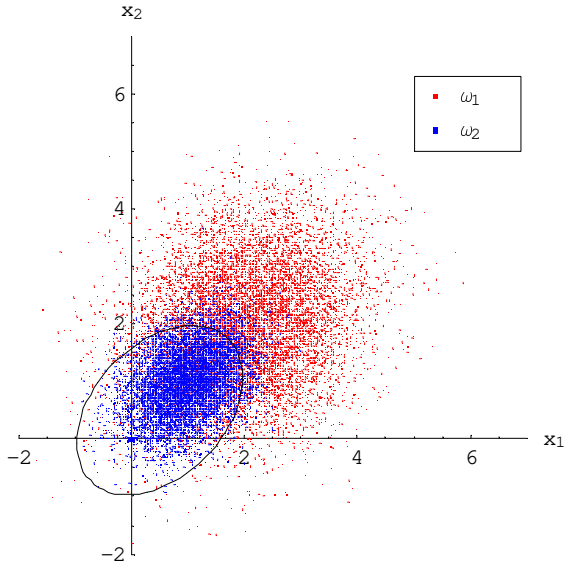

Figure 3.2. Two simulated data sets from two binormal distributions. The number of observations per class is $10,000$.

$$f(h,x_1|\omega_1) = \exp[-.385h - .074x_1^2 + 1.805x_1 - 1.243\sqrt{r}]\times$$
$$\frac{.00157}{\sqrt{|r_1|}}(\exp[.178x_1\sqrt{r}] + \exp[2.49\sqrt{r} - .178x_1\sqrt{r}]) \quad (3.8)$$

and, under the assumption that $X \sim F_2$, is given by:

$$f(h,x_1|\omega_2) = \exp[-1.385h - .074x_1^2 + 1.805x_1 - 1.243\sqrt{r}]\times$$
$$\frac{.00157}{\sqrt{|r_1|}}(\exp[.178x_1\sqrt{r}] + \exp[2.49\sqrt{r} - .178x_1\sqrt{r}]), \quad (3.9)$$

where

$$r_1 = 1.91 + .866h + x_1 - x_1^2 \quad (3.10)$$

The conditional joint density functions (3.8) and (3.9) are defined only on a parabolic area determined in the $h - x_1$ space by the parabola r_1 in (3.10). In general, under different values of the mean vectors and covariance matrices (3.5) this area is a conic section.

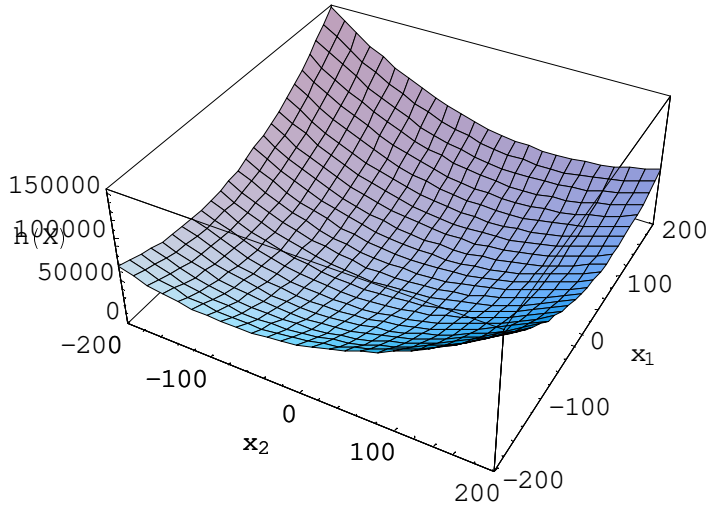

Figure 3.3. A 3-D representation of the log-likelihood ratio function of two features x_1 and x_2.

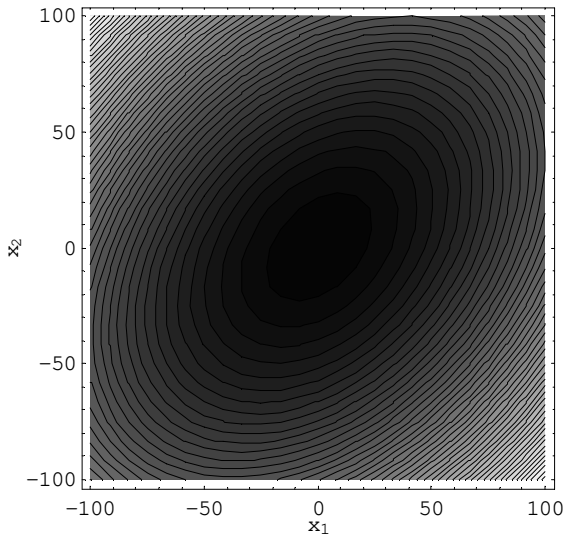

Figure 3.4. Contour plot for the log-likelihood ratio function of two features x_1 and x_2.

Unfortunately, a closed form integration for (3.8) and (3.9) over x_1, to obtain the marginal probabilities $f(h|\omega_1)$ and $f(h|\omega_2)$, is not available. However, we always can obtain a numerical solution to the problem by carrying out the integration over x_1 for every desired value of h. The marginal PDFs of h, conditional on ω_1 and ω_2, are obtained by the described technique and illustrated in Figures 3.5 and 3.6. In addition, these two figures show the histograms of h obtained from simulating testing observations from the distributions F_1 and F_2 and obtaining the log-likelihood ratio h for every observation. The figures show how well both, the histogram and the mathematical solution, are highly consistent. Figure 3.7 shows the two PDFs, together, for the classification purpose.

It is extremely important to comment on the result illustrated in Figure 3.7. Although the data are coming from binormal distributions the log-likelihood ratio is not distributed as normal distribution. Moreover, $f_{h|\omega_2}$, in no way, can be approximated to a normal distribution. It has an abrupt behavior that makes it resemble more the exponential distribution. This simple example provides an important caveat to the exaggerated use of normality for the log-likelihood ratio.

For completeness, the following should be mentioned. One could have simultaneously diagonalized the matrices Σ_1 and Σ_2—said differently, transform x_1 and x_2 to x_1' and x_2' such that the new variables have diagonalized covariance matrices—to get rid of the cross terms in the new space of $h - x_1'$. For the topic of simultaneous diagonalization the reader may be referred

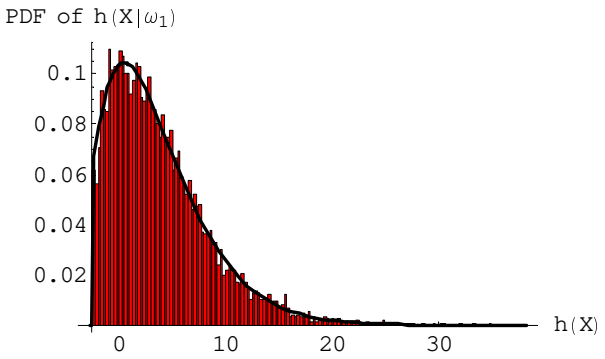

Figure 3.5. The PDF of the log-likelihood ratio under ω_1 obtained from mathematical analysis, along with its histogram obtained from a simulation study.

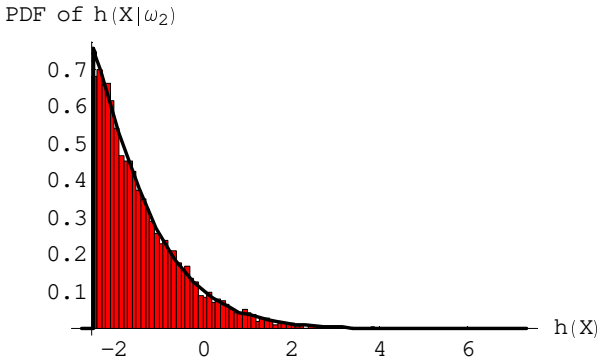

Figure 3.6. The PDF of the log-likelihood ratio under ω_2 obtained from mathematical analysis, along with its histogram obtained from a simulation study.

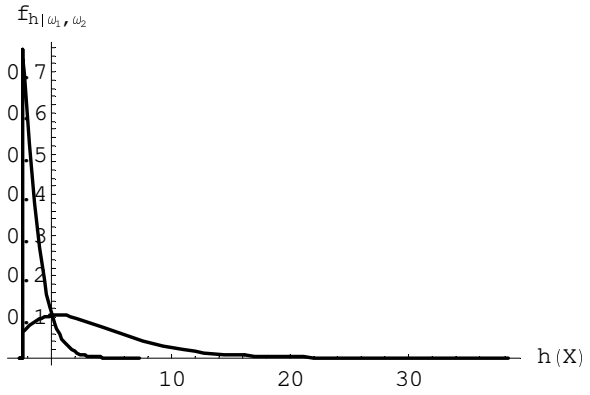

Figure 3.7. The two PDFs of the log-likelihood ratio and ω_1 and ω_2.

to Fukunaga (1990) or for more rigorous analysis to Schott (2005) or Searle (1982). After performing the simultaneous diagonalization to the matrices Σ_1 and Σ_2 and proceeding as described above, the joint density function of h and x_1' can be shown to be

$$f_{H,X_1'}(h,x_1'|\omega_1) = \frac{.00232\exp[-.33h-.166x_1'^2+2.15x_1']}{\sqrt{|r_2|}} \tag{3.11}$$

$$f_{H,X_1'}(h,x_1'|\omega_2) = \frac{.00232\exp[-1.33h-.166x_1'^2+2.15x_1']}{\sqrt{|r_2|}}, \tag{3.12}$$

where

$$r_2 = 2.08+h+1.29x_1'-x_1'^2, \tag{3.13}$$

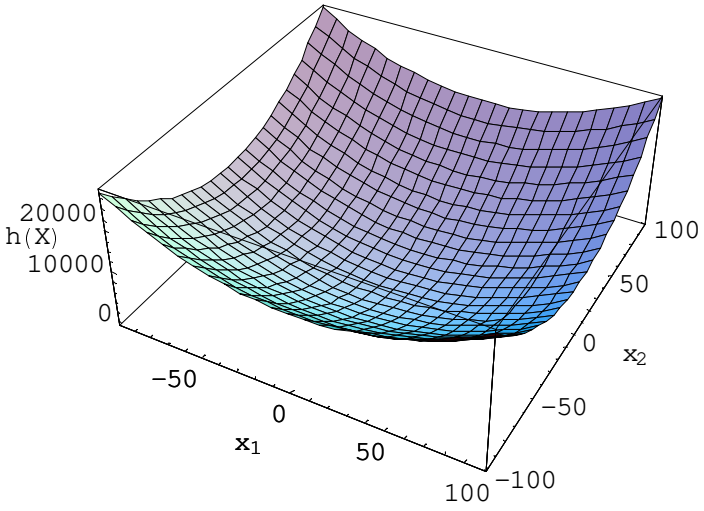

Figure 3.8. A 3-D representation of the log-likelihood ratio function of x_1 and x_2 after simultaneous diagonalization for the two covariance matrices Σ_1 and Σ_2.

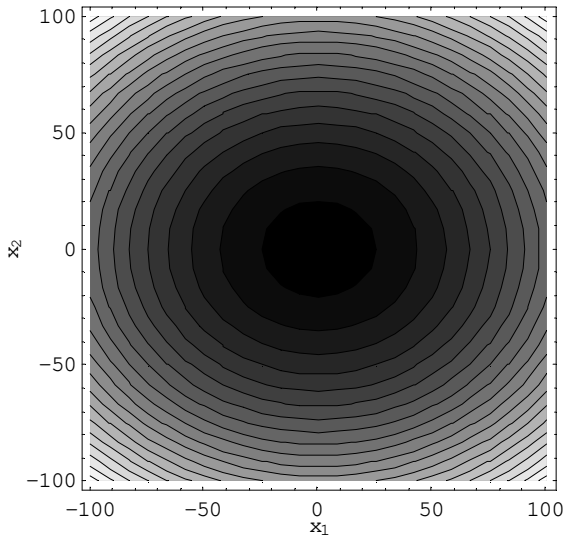

Figure 3.9. A contour plot of the log-likelihood ratio function of x_1 and x_2 after simultaneous diagonalization for the two covariance matrices Σ_1 and Σ_2.

where r_2 is the region on which the joint density of $h - x_1'$ is defined. The log-likelihood ratio (3.4), after simultaneous diagonalization, is illustrated in 3-D, as well as, contour plot in Figures 3.8 and 3.9.

The virtue of simultaneous diagonalization is obvious from Equations 3.11, 3.12, and 3.13.

It deserves mentioning that under the special case of normal distribution for the log-likelihood ratio $h_{\mathbf{t}}(\cdot)$, which is not the case above, the ROC curve can be expressed, using the inverse error function transformation, as:

$$\phi^{-1}(TPF) = \frac{(\mu_1 - \mu_2)}{\sigma_1} + (\frac{\sigma_2}{\sigma_1})\phi^{-1}(FPF) \tag{3.14}$$

This means that the whole ROC curve can be summarized in just two parameters: the intercept a, and the slope b; this is shown in Figure 3.10. We frequently see the Central Limit Theorem at work in higher dimensions driving the ROC curve toward this condition.

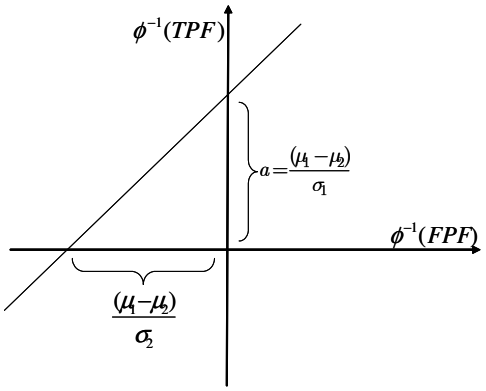

Figure 3.10. The double-normal-deviate plot for the ROC under the normal assumption for the log-likelihood ratio is a straight line.

3.2. Comparison of Nonparametric Methods for Assessing Classifier Performance in Terms of ROC Parameters

For a particular classification rule, $\eta_{\mathbf{t}}$, trained on the training data set $\mathbf{t}$, the true value of the AUC for that rule is given by (1.61). For notational clarity, this may be rewritten as:

$$AUC_{\mathbf{t}} = \int_0^1 TPF_{\mathbf{t}}\, d(FPF_{\mathbf{t}}) \tag{3.15}$$

We can redefine the true AUC (3.15) as the expected value, over the population of testers, of the Mann-Whitney statistics, introduced in (1.74). That is,

$$AUC_{\mathbf{t}} = E_{F_1} E_{F_2} \left[\psi\left(\widehat{h}_{\mathbf{t}}(x|\omega_1), \widehat{h}_{\mathbf{t}}(x|\omega_2) \right) \right], \tag{3.16}$$

$$\psi(a,b) = \begin{cases} 1 & a > b \\ 1/2 & a = b \\ 0 & a < b \end{cases} \tag{3.17}$$

where F_1 and F_2 represent the population of class 1 and class 2 respectively. Even if the two distributions are identical, except in parameter values, they are distinguished here by two different subscripts to indicate that the expectation is carried over two different testing data sets, one for each class. In the nonparametric situation, the expectation (3.16) can be approximated numerically by expectation over the empirical distribution $\widehat{F}_1$ and $\widehat{F}_2$; this is given by (1.74). If the distributions of the log-likelihood ratio $\widehat{h}_{\mathbf{t}}$ are continuous then the condition $a = b$, in the equation above, occurs with probability zero. and we can write:

$$\psi(a,b) = \begin{cases} 1 & a > b \\ 0 & a < b \end{cases} \tag{3.18}$$

The definition (3.16) is nothing but the probability that a random variable with distribution F_2 is larger than a random variable with distribution F_1. The equivalence of (3.15) and (3.16) will be derived in Chapter 5.

3.2.1. Mean of Classifier Performance vs. Training Set Size

Since any classifier is designed using a finite-size data set, its true performance—the performance obtained from expectation over the population—is dominated by the size of this set—assuming fixing the distribution of the data. When the classifier is re-designed using different training set size, the expected performance will vary. This is because the limited size training set has some, not all, of the information represented in the population. In addition, if it is re-designed, using another data set having the same size, the performance will vary as well. This is because the performance metric is a function of the training data set; hence it is a random variable. An illustration of this can be provided by the following simulation study. If we consider the case, e.g., when the distribution is given by (3.1), where the dimensionality $p = 11$, $\Sigma_1 = \Sigma_2 = \mathbf{I}$, the identity matrix, $\mu_1 = \mathbf{0}$, the zero vector, and $\mu_2 = 0.27 \times \mathbf{1}$, where $\mathbf{1}$ is the vector all of whose components are ones then Figure 3.11 shows the variability in the AUC vs. the inverse of the training set size.

To exhibit the basic structure of the problem under the practical limitation of a finite-training set, we carried out simulations inspired by Chan et al. (1999) and the work of Fukunaga and Hayes (1989*a*,*b*). In our simulation, we assume that the feature vector has the multinormal distribution with the following parameters: $\mu_1 = 0, \mu_2 = c\mathbf{1}$, and $\Sigma_1 = \Sigma_2 = \mathbf{I}$ where 0 is the vector all of whose components are zeros, $\mathbf{1}$ is the vector all of whose components are ones, $\mathbf{I}$ is the identity matrix, and c is a constant. A fundamental metric is the Mahalanobis distance between the mean vectors of the two classes: it is defined as:

$$\Delta = \left[(\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2) \right]^{1/2} \tag{3.19}$$

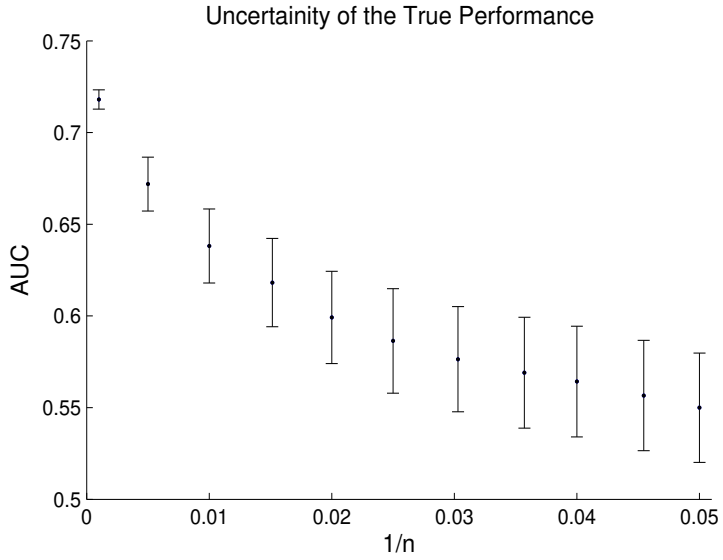

Figure 3.11. Uncertainty (variance) around the mean performance of the Bayes classifier, for 11 features, vs. the size of the training data set. Asymptotically, the variability vanishes.

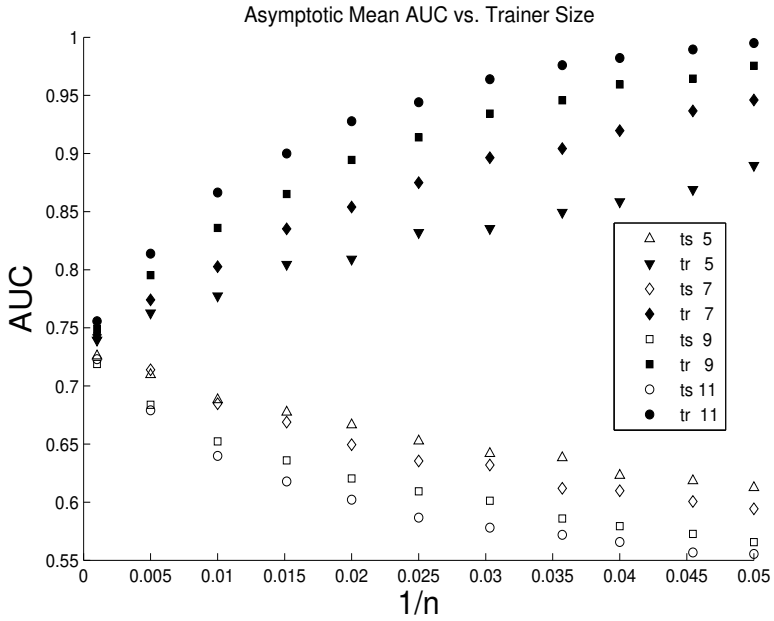

Figure 3.12. Mean AUC of the Bayes classifier. For every training sample size n, the classifier is tested on pseudo-infinite testers (represented as "ts") and tested as well on the same training sample (represented as "tr"). Each curve shows the average performance over 100 MC trials. The numbers in the legend are the dimensionalities of the feature vectors.

It expresses how these two vectors are separated from each other with respect to the spread Σ. In the simulation of the present example, the Mahalanobis distance is $c^2 p$. In this simulation, illustrated in Figure 3.12, the value c is adjusted for every dimensionality to obtain the same asymptotic AUC. This allows us to isolate the effect of the variation in training set sizes. Typically, the simulations described in this context used a value of 0.8 for Δ. For the time being, it is assumed that $n_1 = n_2 = n$, which is referred to as the training set size per class. For a particular dimensionality, and for particular data set size n, two training data sets are generated using the above parameters and distributions. When the classifier is trained, it will be tested on a pseudo-infinite test set, here 1000 cases per class, to obtain a very good approximation to the true AUC for the classifier trained on this very training data set; this is called a single realization or a Monte-Carlo (MC) trial.

Many realizations of the training data sets with same n are generated over MC simulation to study the mean and variance of the AUC for the Bayes classifier under this training set size. The number of MC trials used is 100.

Several important observations can be made from these results. As was expected, for training size n the mean apparent AUC, i.e., coming from testing on the same training data set, is upwardly biased from the true AUC. It should be cautioned that this is on the average, i.e., over the population of all training sets; it is possible that for a single data set (single realization) the apparent performance can be better or worse than the true one. In addition, the classifier had the same asymptotic performance, approximately 0.74, for all dimensionalities in the simulation (by design as above).

3.2.2. Nonparametric Inference for the AUC

In the present section, we extend the study carried out in Efron and Tibshirani (1997), and summarized in Section 2.2, to include the AUC as the performance metric. Similar work has been done by considering the .632 bootstrap and the leave-one-out cross validation in Sahiner et al. (2001).

3.2.2.1. Mathematical Definitions

Analogously to what has been presented in Section 2.2, and we will follow the same notation, we can extend the literature to consider the AUC as the summary performance metric. The term *design* may be used interchangeably with *train* since the training phase involves procedures like cross-validation for model or parameter selection to optimize some performance metric. If only one data set is available for design, and neither the data distribution nor a parametric model is available this is referred to as the nonparametric situation. In this case, the mean performance of the classifier has to be estimated from the same training data set.

Before switching to the AUC some more elaboration on Section 2.2 is needed. The simple bootstrap estimator (2.35) can be rewritten as:

$$\widehat{Err}^{SB} = E_* E_F \left[L(\eta_{\mathbf{t}^*}(x), y) | \mathbf{t}^* \right] \tag{3.20}$$

Since there would be some observation overlap between the $\mathbf{t}$ and $\mathbf{t}^*$ this approach suffers an obvious bias as was introduced in that section. This was the motivation behind interchanging the expectations and defining the leave-one-out bootstrap (Section 2.2.4.1). Alternatively, we could have left the order of the expectation but with testing on only those observations in $\mathbf{t}$ that do not appear in the bootstrap replication $\mathbf{t}^*$, i.e., the distribution $\hat{F}^{(*)}$. We call the resulting estimator $\widehat{Err}^{(*)}$, which is given formally by:

$$\widehat{Err}^{(*)} = E_* E_{\hat{F}^{(*)}} \left[L(\eta_{\mathbf{t}^*}(x), y) | \mathbf{t}^* \right] \tag{3.21}$$

$$= \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{i=1}^{N} I_i^b L(\eta_{\mathbf{t}^{*b}}(x_i), y_i) / \sum_{i'=1}^{N} I_{i'}^b \right] \tag{3.22}$$

where the indicator I_i^b equals one if the observation t_i is excluded from the bootstrap replication $\mathbf{t}^{*b}$, and equals zero otherwise. The inner expectation in (3.22) is taken over those observations not included in the bootstrap replication $\mathbf{t}^*$, while the outer expectation is taken over all the bootstrap replications.

Analogously to Section 2.2 and to what has been introduced above we can define several bootstrap estimators for the AUC. The start is the simple bootstrap estimate which can be written as:

$$\widehat{AUC}_{\mathbf{t}}^{SB} = E_* \left[AUC_{\mathbf{t}^*}(\widehat{F}) \right] \tag{3.23}$$

$$= E_* \left[\frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \psi(\hat{h}_{\mathbf{t}^*}(x_i), \hat{h}_{\mathbf{t}^*}(x_j)) \right], \tag{3.24}$$

$$where\ \widehat{F} \rightarrow \mathbf{t}^*,\ x_i \in \omega_1,\ and\ x_j \in \omega_2 \tag{3.25}$$

This averages the Mann-Whitney statistic over the bootstraps, where $AUC_{\mathbf{t}^*}(\widehat{F})$ refers to the AUC obtained from training the classifier on the bootstrap replicate $\mathbf{t}^*$ and testing it on the empirical distribution $\widehat{F}$. In the approach used here, the bootstrap replicate $\mathbf{t}^*$ preserves the ratio between n_1 and n_2. That is, the training sample $\mathbf{t}$ is treated as $\mathbf{t} = \mathbf{t}_1 \cup \mathbf{t}_2, \mathbf{t}_1 \in \omega_1, and\, \mathbf{t}_2 \in \omega_2$ then n_1 cases are replicated from the first-class sample and n_2 cases are replicated from the second-class sample to produce $\mathbf{t}_1^*$ and $\mathbf{t}_2^*$ respectively, where $\mathbf{t}^* = \mathbf{t}_1^* \cup \mathbf{t}_2^*$; this was not needed when the performance metric was the error rate. This is because error rate is a statistic that does not operate simultaneously on two different data sets as the Mann-Whitney statistic does (Mann-Whitney statistic will be illustrated in subsequent chapters as a two-sample U-statistic). For a limited number of bootstraps the expectation (3.23) is approximated by:

$$\widehat{AUC}_{\mathbf{t}}^{SB} = \frac{1}{B} \sum_{b=1}^{B} \left[AUC_{\mathbf{t}^{*b}}(\widehat{F}) \right] \tag{3.26}$$

i.e., averaging over the B bootstraps for the AUC obtained from training the classifier on the bootstrap replicate $\mathbf{t}^{*b}$ and testing it on the original data set $\mathbf{t}$.

The same motivation behind the estimator (2.38) can be applied here, i.e., testing only on those cases in $\mathbf{t}$ that are not included in the training set $\mathbf{t}^{*b}$ in order to reduce the bias. This can be carried out in (3.26) without interchanging the summation order. The new estimator is named $\widehat{AUC}_{\mathbf{t}}^{(*)}$, where the parenthesis notation $(*)$ refers to the exclusion, in the testing stage, of the training cases that were generated from the bootstrap replication. Formally, this is written as:

$$\widehat{AUC}_{\mathbf{t}}^{(*)} = \frac{1}{B}\sum_{b=1}^{B}\left[AUC_{\mathbf{t}^{*b}}(\hat{F}^{(*)})\right] \tag{3.27}$$

$$= \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\psi(\hat{h}_{\mathbf{t}^*}(x_i),\hat{h}_{\mathbf{t}^*}(x_j))I_i^b I_j^b / \sum_{i'=1}^{n_1} I_{i'}^b \sum_{j'=1}^{n_2} I_{j'}^b\right] \tag{3.28}$$

The 0.632 estimator can be introduced here in the same way it was used for the true error rate (see Section 2.2.4.3). The true AUC for the classifier if trained on a particular training data set $\mathbf{t}$ can be written as:

$$\widehat{AUC}_{\mathbf{t}} = \overline{AUC}_{\mathbf{t}} + E_*(AUC_{\mathbf{t}*}(\hat{F}) - \overline{AUC}_{\mathbf{t}*}) \tag{3.29}$$

This is the same approach developed in Section 2.2.4.2 for the error rate. If testing is carried out on cases excluded from the bootstraps, then (3.29) can be approximated analogously to what was done in Section 2.2.4.3. This gives rise to the 0.632 AUC estimator:

$$\widehat{AUC}_{\mathbf{t}}^{(.632)} = .368\,\overline{AUC}_{\mathbf{t}} + .632\,\widehat{AUC}_{\mathbf{t}}^{(*)} \tag{3.30}$$

It should be noted that this estimator is designed to estimate the true AUC for a classifier trained on the data set $\mathbf{t}$ (the classifier performance conditional on the training data set $\mathbf{t}$). This is on contrary to the estimator (3.27) that estimates the mean performance of the classifier (this is the expectation over the training set population for the conditional performance).

The 0.632+ estimator, $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$, develops from $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ in the same way as $\widehat{Err}_{\mathbf{t}}^{(.632+)}$ developed from $\widehat{Err}_{\mathbf{t}}^{(.632)}$ in Section 2.2.4.4. There are two modifications to the details; the first regards the *no-information error rate* γ, and the second regards to the definitions (2.52). The *no-information* $AUC\gamma_{AUC}$, an analogue to the *no-information error rate* γ, is given by (1.61) but with TPF and FPF given under the *no-information* distribution E_{0F} (see Section 2.2.4.4). To estimate γ_{AUC} assume that there are n_1 cases from class ω_1 and n_2 cases from class ω_2 as described above. Assume also for fixed threshold th the two metrics that define the error rate for this threshold value are TPF and FPF. Also, assume that the sample cases are tested by the classifier and each sample has been assigned a decision value (log-likelihood ratio). Under the *no-information* distribution, consider the following. For every decision value $\hat{h}_{\mathbf{t}}(x_i)$ assigned for the case $t_i = (x_i, y_i)$, create new $n_1 + n_2 - 1$ cases; all of them have the same decision value $\hat{h}_{\mathbf{t}}(x_i)$, while their responses are equal to the responses of the rest $n_1 + n_2 - 1$ cases t_j $j \neq i$. Under this new sample that consists of $(n_1 + n_2)^2$ cases, it is quite easy to see that the new TPF and FPF for the same threshold th are given by:

$$FPF_{0\hat{F},th} = TPF_{0\hat{F},th} = \frac{TPF \cdot n_1 + FPF \cdot n_2}{(n_1 + n_2)} \tag{3.31}$$

This means that the ROC curve under the *no-information* rate is a straight line with slope equal to one; this directly gives:

$$\gamma_{AUC} = 0.5 \tag{3.32}$$

Regarding the definitions (2.52) they should be modified to accommodate the AUC. The new definitions are given by:

$$\widehat{AUC}_{\mathbf{t}}^{(.632+)} = \widehat{AUC}_{\mathbf{t}}^{(.632)} + (\widehat{AUC}_{\mathbf{t}}^{(*)\prime} - \overline{AUC}_{\mathbf{t}})\frac{.368 \cdot .632 \cdot \hat{R}'}{1 - .368\hat{R}'} \tag{3.33}$$

where

$$\widehat{AUC}_{\mathbf{t}}^{(*)\prime} = \max(\widehat{AUC}_{\mathbf{t}}^{(*)}, \gamma_{AUC}), \tag{3.34}$$

$$\hat{R}' = \begin{cases} (\widehat{AUC}_{\mathbf{t}}^{(*)} - \overline{AUC}_{\mathbf{t}})/(\gamma_{AUC} - \overline{AUC}_{\mathbf{t}}) & if\ \overline{AUC}_{\mathbf{t}} > \widehat{AUC}_{\mathbf{t}}^{(*)} > \gamma_{AUC} \\ 0 & otherwise \end{cases} \tag{3.35}$$

We note that there are several points of view regarding the relative utility of measuring the "true performance", i.e., the performance conditional on a given training data set, versus estimating the mean performance over the population of training sets. Some users might argue that the conditional performance is the most appropriate, claiming that they will *freeze* the trainers. However, this does not really correspond to the practical world in which practitioners up-date the training as more data becomes available; in that case the target would be the expected performance over the population of trainers.

3.2.2.2. Experimental Results

Different experiments have been carried out to compare these three bootstrap-based estimators, considering different dimensionalities, different parameter values, and training set sizes, all based on the multinormal assumption for the feature vector. We use the same experiments described in Section 3.2.1. Here in this section we illustrate the results when the dimensionality p was five. The number of trainer groups per point (the number of MC trials) is 1000 and the number of bootstraps is 100.

It is apparent from Figure 3.13 that the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ is downward biased. This is a natural opposite of the upward bias observed in Efron and Tibshirani (1997) when the metric was the true error rate as a measure of incorrectness, by contrast with the true AUC

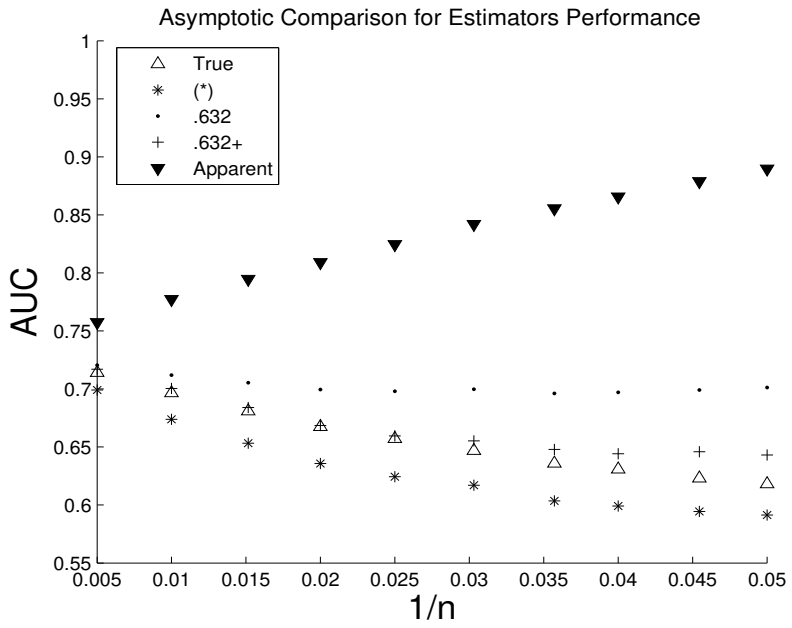

Figure 3.13. Comparison of the three bootstrap estimators, $\widehat{AUC}_{\mathbf{t}}^{(*)}$, $\widehat{AUC}_{\mathbf{t}}^{(.632)}$, and $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ for 5-feature predictor. The $\widehat{AUC}_{\mathbf{t}}^{(*)}$ is downward biased, while the $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ is an over correction for that bias. $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ is almost the unbiased version of the$\widehat{AUC}_{\mathbf{t}}^{(.632)}$.

as a measure of correctness. The $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ is designed as a correction for $\widehat{AUC}_{\mathbf{t}}^{(*)}$; it appears in the figure to correct for that but with an over-shoot. The correct adjustment for the remaining bias is almost achieved by the estimator $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$. The $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ estimator can be seen as an attempt to balance between the two extreme biased estimators, $\widehat{AUC}_{\mathbf{t}}^{(*)}$ and $\overline{AUC}_{\mathbf{t}}$.

Table 3.1 gives a comparison for the different estimators in terms of the RMS and $\text{RMS}_{\text{AroundMean}}$ values. The RMS is defined in the present context as the root of the mean squared difference between an estimate and the population mean, i.e., the mean over all possible training sets. More details about the definitions of both of them are given in Section 3.2.3.

Estimator	Mean	SD	RMS	RMS around mean	Corr. Coef.	Size
$AUC_{\mathbf{t}}$	0.6181	0.0434	0	0.0434	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.5914	0.0947	0.0973	0.0984	0.2553	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.7012	0.0749	0.1128	0.1119	0.2559	20
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6431	0.0858	0.0906	0.0894	0.2218	
$\overline{AUC}_{\mathbf{t}}$	0.8897	0.0475	0.2774	0.2757	0.2231	
$AUC_{\mathbf{t}}$	0.6231	0.0410	0	0.0410	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.5945	0.0947	0.0956	0.0990	0.2993	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6991	0.0763	0.1066	0.1077	0.3070	22
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6459	0.0846	0.0863	0.0876	0.2726	
$\overline{AUC}_{\mathbf{t}}$	0.8788	0.0499	0.2615	0.2606	0.2991	
$AUC_{\mathbf{t}}$	0.6308	0.0400	0	0.0400	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.5991	0.0865	0.0897	0.0922	0.2946	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6971	0.0701	0.0961	0.0965	0.2997	25
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6442	0.0817	0.0815	0.0828	0.2758	
$\overline{AUC}_{\mathbf{t}}$	0.8656	0.0471	0.2406	0.2395	0.2833	
$AUC_{\mathbf{t}}$	0.6359	0.0358	0	0.0358	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6035	0.0840	0.0874	0.0901	0.2904	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6962	0.0688	0.0906	0.0915	0.2934	28
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6479	0.0792	0.0785	0.0802	0.2719	
$\overline{AUC}_{\mathbf{t}}$	0.8554	0.0472	0.2253	0.2246	0.2747	
$AUC_{\mathbf{t}}$	0.6469	0.0343	0	0.0343	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6170	0.0750	0.0792	0.0807	0.2746	

Table 3.1. *continued*

Estimator	Mean	SD	RMS	RMS around mean	Corr. Coef.	Size
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6997	0.0623	0.0818	0.0817	0.2722	33
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6553	0.0761	0.0752	0.0766	0.2656	
$\overline{AUC}_{\mathbf{t}}$	0.8419	0.0439	0.2010	0.1999	0.2434	
$AUC_{\mathbf{t}}$	0.6571	0.0308	0	0.0308	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6244	0.0711	0.0753	0.0783	0.3185	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6981	0.0598	0.0710	0.0725	0.3167	40
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6595	0.0739	0.0707	0.0739	0.3092	
$\overline{AUC}_{\mathbf{t}}$	0.8246	0.0431	0.1735	0.1730	0.2923	
$AUC_{\mathbf{t}}$	0.6674	0.0271	0	0.0271	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6357	0.0654	0.0690	0.0727	0.3534	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.6995	0.0556	0.0615	0.0642	0.3570	50
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6685	0.0690	0.0646	0.0690	0.3522	
$\overline{AUC}_{\mathbf{t}}$	0.8091	0.0406	0.1473	0.1474	0.3517	
$AUC_{\mathbf{t}}$	0.6808	0.0217	0	0.0217	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6533	0.0546	0.0602	0.0611	0.2451	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.7053	0.0471	0.0527	0.0531	0.2488	66
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.6840	0.0568	0.0556	0.0569	0.2477	
$\overline{AUC}_{\mathbf{t}}$	0.7946	0.0355	0.1195	0.1192	0.2499	
$AUC_{\mathbf{t}}$	0.6965	0.0158	0	0.0158	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6738	0.0454	0.0483	0.0507	0.3422	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.7119	0.0399	0.0405	0.0428	0.3492	100
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.7004	0.0452	0.0426	0.0453	0.3448	
$\overline{AUC}_{\mathbf{t}}$	0.7772	0.0312	0.0860	0.0866	0.3596	
$AUC_{\mathbf{t}}$	0.7141	0.0090	0	0.0090	1.0000	
$\widehat{AUC}_{\mathbf{t}}^{(*)}$	0.6991	0.0298	0.0327	0.0334	0.2288	
$\widehat{AUC}_{\mathbf{t}}^{(.632)}$	0.7205	0.0272	0.0273	0.0279	0.2291	200
$\widehat{AUC}_{\mathbf{t}}^{(.632+)}$	0.7170	0.0285	0.0279	0.0286	0.2294	
$\overline{AUC}_{\mathbf{t}}$	0.7573	0.0228	0.0487	0.0489	0.2277	

Table 3.1. Comparison of the different bootstrap-based estimators of the AUC. they are comparable to each other in the RMS sense, $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ is almost unbiased, and all are weakly correlated with the true conditional performance $AUC_{\mathbf{t}}$.

3.2.2.3. Remarks

As shown by Efron and Tibshirani (1997), the $\widehat{Err}_{\mathbf{t}}^{(1)}$ estimator is a smoothed version of the leave-one-out cross validation, since for every test sample case the classifier is trained on many bootstrap replicates. This reduces the variability of the cross-validation based estimator. On the other hand, the effective number of cases included in the bootstrap replicates is .632 of the total sample size n. This accounts for training on a less effective data set size; this makes the leave-one-out bootstrap estimator $\widehat{Err}^{(1)}$ more biased than the leave-one-out cross-validation. This bias issue is observed Sahiner et al. (2001), as well, when the performance metric was the AUC. This fact is illustrated in Figure 3.14 for $\widehat{AUC}_{\mathbf{t}}^{(*)}$. At every sample size n the true value of the AUC is plotted. The estimated value $\widehat{AUC}_{\mathbf{t}}^{(*)}$ at data sizes of $n/.632$ and $n/.5$ are plotted as well. It is obvious that these values are lower and higher than the true value respectively, which supports the discussion of whether the leave-one-out bootstrap is supported on 0.632 of the cases or 0.5 of the cases (as mentioned in Efron and Tibshirani (1997)) or, as here, something in-between.

The estimators studied here are used to estimate the mean performance (AUC) of the classifier. However, the basic motivation for the $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ and $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ is to estimate the AUC conditional on the given data set $\mathbf{t}$. This is the analogue of $\widehat{Err}_{\mathbf{t}}^{(.632)}$ and $\widehat{Err}_{\mathbf{t}}^{(.632+)}$. Nevertheless, as mentioned in Efron and Tibshirani (1997) and detailed in Zhang (1995) the cross-validation, the basic ingredient of the bootstrap based estimators, is weakly correlated with the true performance on a sample by sample basis. This means that no estimator has a preference in estimating the conditional performance. This fact is elaborated in the following section.

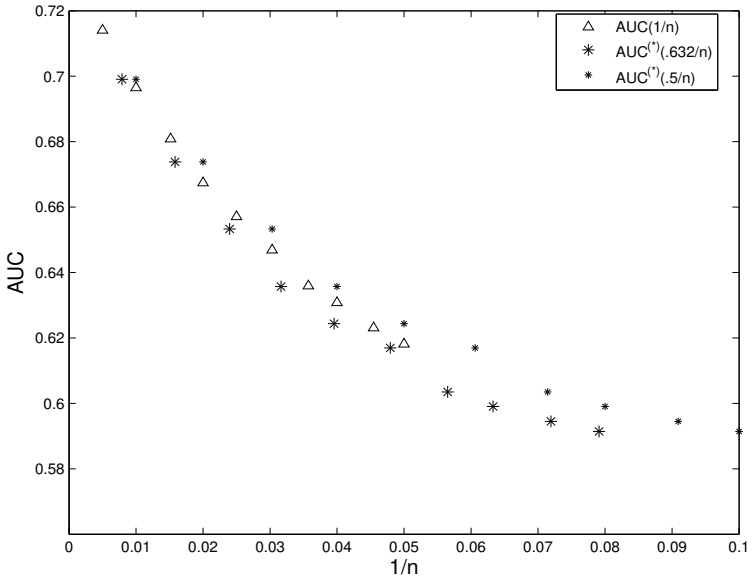

Figure 3.14. The true AUC and rescaled version of the bootstrap estimator $\widehat{AUC}_{\mathbf{t}}^{(*)}$. At every sample size n the true AUC is shown along with the value of the estimator $\widehat{AUC}_{\mathbf{t}}^{(*)}$ at $n/.632$ and $n/.5$.

3.2.3. Components of Variance of Performance Estimators and Weak Correlation

Section 3.2 shows how the different estimators estimate the mean performance. This estimation is a random variable whose randomness comes from the training data set. The bias is not enough to judge estimator efficiency. Rather the square root of the mean square error (MSE) should be used; the MSE is defined in (2.3). Efron and Tibshirani (1997) suggested to use the difference between every estimated value from a training data set and the true performance of the classifier trained on this data set. The following expression and its illustration in Figure 3.15 is very instructive in understanding the different aspects of the new work undertaken in this dissertation. Formally, this can be written as:

$$
\begin{aligned}
RMS^2 = MSE(\hat{S}_{\mathbf{t}_i}, S_{\mathbf{t}_i}) &= E_{MC}\left\{\hat{S}_{\mathbf{t}_i} - S_{\mathbf{t}_i}\right\}^2 \\
&= \underbrace{\underbrace{E_{MC}^2\{\hat{S}_{\mathbf{t}_i} - \overline{S}\}}_{Bias^2(\hat{S}_{\mathbf{t}_i},\overline{S})} + \underbrace{E_{MC}\{\hat{S}_{\mathbf{t}_i} - \overline{\hat{S}}\}^2}_{\mathrm{Var}[\hat{S}_{\mathbf{t}_i}]}}_{RMS(\hat{S}_{\mathbf{t}_i},\overline{S}) \equiv RMS\, around\, the\, mean} + \underbrace{\underbrace{E_{MC}\{S_{\mathbf{t}_i} - \overline{S}\}^2}_{\mathrm{Var}[S_{\mathbf{t}_i}]} - 2\,\mathrm{Cov}(\hat{S}_{\mathbf{t}_i}, S_{\mathbf{t}_i})}_{small}, \\
where: \overline{S} &= E_{MC}\left\{S_{\mathbf{t}_i}\right\},\ \overline{\hat{S}} = E_{MC}\left\{\hat{S}_{\mathbf{t}_i}\right\}
\end{aligned}
\tag{3.36}
$$

where $S_{\mathbf{t}_i}$ is the true performance (Err or AUC), and $\hat{S}_{\mathbf{t}_i}$ is any estimate function of the training data set $\mathbf{t}_i$. E_{MC} represents the expectation approximated by averaging over the MC trials, i.e., $E_{MC} \equiv \Sigma_{i=1}^{G}(\cdot)/G$. This definition of the RMS implies that the estimators will be treated as if they estimate the true performance $S_{\mathbf{t}_i}$ of the classifier conditional on the data set $\mathbf{t}_i$. This is exactly what the .632 and .632+ estimators were designed for, but not the $(*)$ estimator. Equation (3.36) show the decomposition of this RMS into four components. The first two components are the bias-squared and the variance for the estimator $\hat{S}_{\mathbf{t}_i}$ as if it estimates S, the mean performance, not the true performance conditional on a particular data set $\mathbf{t}_i$. The third component is the variability of the conditional true performance $S_{\mathbf{t}_i}$, e.g., see Figure 3.11. The fourth component is the covariance between the estimator and the conditional true performance. These four components are shown in Table 3.1.

The estimators are comparable in RMS with a little superiority of $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ for higher set size; it is almost unbiased as well. However, Efron and Tibshirani (1997) ran many experiments considering different classifiers and distributions and their conclusion was quoted in Section 2.4. A comment on their conclusion is in order. As they stated, the results vary considerably from experiment to experiment, i.e., in some experiments the $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ estimator wins while in others the $\widehat{AUC}_{\mathbf{t}}^{(.632)}$ or even the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ does. They concluded:

> "*...but in terms of RMS error the .632+ rule is an overall winner*"

This conclusion was without proposing a metric for deciding the *overall winner*. It was apparent that the .632+ rule is the winner in terms of the bias—as was designed for. However, an average for the RMS of every estimator, across all the 24 experiments they ran, is shown in Table 3.2. The estimators $\widehat{AUC}_{\mathbf{t}}^{(*)}$ and $\widehat{AUC}_{\mathbf{t}}^{(.632+)}$ are quite comparable to each other.

Estimator	Average over 24 Exp.
$Err_{\mathbf{t}}$	0
$\widehat{Err}_{\mathbf{t}}^{(1)}$	.083
$\widehat{Err}_{\mathbf{t}}^{(.632)}$	.101
$\widehat{Err}_{\mathbf{t}}^{(.632+)}$	.081
$\overline{Err}_{\mathbf{t}}$	.224

Table 3.2. Average of RMS error of each estimator over 24 experiments run by Efron and Tibshirani (1997). The estimator $\widehat{Err}_{\mathbf{t}}^{(1)}$ is quite comparable to $\widehat{Err}_{\mathbf{t}}^{(.632+)}$.

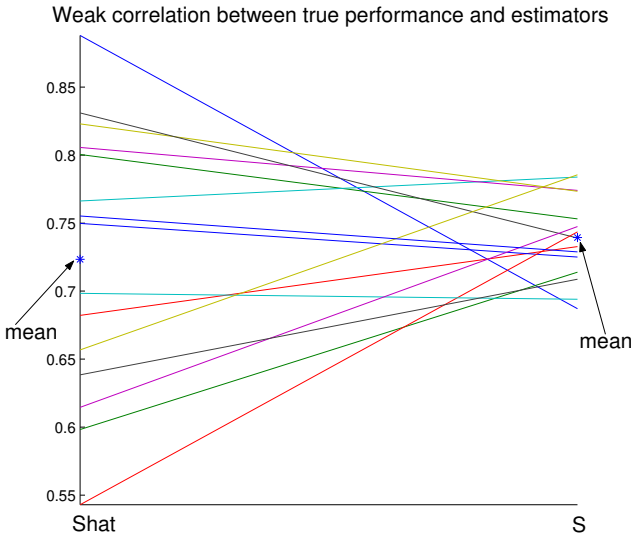

Figure 3.15. The lack of correlation (or the very weak correlation) between the bootstrap-based estimators and the true conditional performance. Every line connects the true performance of the classifier trained on a data set $\mathbf{t}_i$ and the estimated value. The figure represents 15 trials of the 1000 MC trials. Two nearby values of true performance may correspond to two widely separated estimates on different sides of the mean.

A crucial comment must be made on the results of Table 3.1 in the light of the uncertainty components (3.36). The results show that the RMS and $RMS_{AroundMean}$ are very close to each other. This means that the last two components of (3.36) are negligible. It is quite interesting to conclude the following: even if the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ estimator was designed to estimate the mean performance and others were designed to estimate the true performance conditional on a particular data set, they agreed in both objectives. More surprisingly, the last component in (3.36), $\mathrm{Cov}(\widehat{S}_{\mathbf{t}_i}, S_{\mathbf{t}_i})$, is very small. This not because of the relative comparison between that term and other uncertainty components but because of the lack of correlation between all of the estimators and the true conditional performance; see the correlation coefficient in Table 3.1. This fact was observed in simulations by others, e.g., Efron (1983); Efron and Tibshirani (1997). An excellent mathematical treatment, Zhang (1995), shows that the cross-validation estimator should not be used to estimate the true error rate of a classification rule conditional on a particular training data set because they are uncorrelated. This can shed some light on the weak correlation in Table 3.1, since these bootstrap estimators have an underlying layer of cross-validation (see Section 2.2.4.1).

The point of the previous paragraph is somewhat subtle; the reader may find the illustration in Figure 3.15 helpful for understanding it.

This figure shows 15 realizations of the 1000 MC trials of same experiment. On the right are the true AUC values of the classifier when trained on these different 15 training sets. On the left are the corresponding 15 estimated values of the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ estimator. The lines provide links between the true values and the corresponding estimates. This figure shows that two nearby true values for the AUC are likely to have two widely separated estimated values on different sides of the mean. This visually illustrates the lack of correlation (or the weak correlation) between the estimators and the true conditional performance.

3.3. Estimating the Variability of the Performance Estimators

The results mentioned above mean that if a training data set is available with no information about the distribution, it is possible to obtain good estimates of the mean AUC of that classifier from this training data set using any of the estimators discussed above. However, each estimate has an associated variability (shown by the MC simulation). Unfortunately, in a practical setting, it is not possible to generate different data sets to know the variability of any particular estimator. The next question then is, having estimated the mean performance of a classifier, what is the associated uncertainty of this estimate, i.e., can an estimate of the variance of this estimator be obtained from the same training data set? The answer of this critical question is provided by the method of the influence function analogously to what Efron and Tibshirani (1997) proposed for estimating the uncertainty in $\widehat{Err}_{\mathbf{t}}^{(1)}$. The only estimator suitable for such a method among those discussed until now is the $\widehat{AUC}_{\mathbf{t}}^{(*)}$, since it is the only smooth estimator, as will be detailed below.

Before proceeding, a very careful investigation of some critical issues is needed. The $\widehat{AUC}_{\mathbf{t}}^{(*)}$, defined in (3.27), is the expectation over the bootstraps for the AUC that come from training on a bootstrap replicate and testing on only those cases not included in that bootstrap training sample. The concept of the influence function (Section 2.1.4) can be implemented by perturbing a sample case and studying its effect on the variability of the estimator. This perturbation propagates through to the probability masses of the bootstrap replicates as well. It can be easily shown that (see Efron, 1992) the bootstrap b includes the case t_i N_i^b times with probability $g_{\varepsilon,i}^b$ given by (4.4). The estimator $\widehat{AUC}_{\mathbf{t}}^{(*)}$, after perturbation, is evaluated as:

$$\widehat{AUC}_{\mathbf{t}}^{(*)}(\hat{F}_{\varepsilon,i}) = \sum_b g_{\varepsilon,i}^b \, AUC_{\mathbf{t}^{*b}}(\hat{F}_{\varepsilon,i}^{(*)}) \tag{3.37}$$

The reader should note that if there is no perturbation, i.e., ε is set to zero, (3.37) is merely reduced to an averaging over the bootstraps. Details are deferred to Chapter 4.

There is a critical point related to the smoothness of the estimator to be used. This detailed discussion is mentioned here, in this section, not in 4. Applying the influence function the $\widehat{AUC}_{\mathbf{t}}^{(*)}$ statistic enforces distributing the differential operator $\partial/\partial\varepsilon$ over the summation to be encountered by the unsmooth statistic $AUC_{\mathbf{t}^{*b}}(\hat{F}_{\varepsilon,i}^{(*)})$ in (3.37). It is unsmooth since the classifier is trained on just one data set (very similar to a single iteration of the cross-validation). For better understanding for the smoothness issue, consider the very simple case where there are just two features and the classifier is designed as (3.2) but with assuming equal covariance matrices. This is called linear discriminant analysis since the decision surface $h(X) = 0$ will be a straight line in the bi-feature plane (generally, it will be a hyper-plane in the p-dimensional feature space). Also for simplicity, consider the true error as the metric of interest; then the analogue to (3.37) is:

$$\widehat{Err}_{\mathbf{t}}^{(*)}(\hat{F}_{\varepsilon,i}) = \sum_b g_{\varepsilon,i}^b \, Err_{\mathbf{t}^{*b}}(\hat{F}_{\varepsilon,i}^{(*)}) \tag{3.38}$$

A simple simulation in this bi-feature problem was carried out using 1000 bootstraps. The decision surfaces obtained from the first five bootstrap replicates are shown in Figure 3.16.

A sample is generated from each of two classes and is represented in the figure together with the decision surface obtained by training on this sample. The decision surfaces obtained from training on the first five bootstraps (one at a time) are drawn as well. Each decision surface trained on the bootstrap replicate $\mathbf{t}^{*b}$ and tested on the sample cases not included in the training produces an estimate $Err_{\mathbf{t}^{*b}}(\hat{F}_{\varepsilon,i}^{(*)})$, which is clearly unsmooth. This is because the estimate does not change with a change in a feature value, e.g., X_1, unless this change allows X_i to cross the decision surface. This lack of smoothness leads to the conclusion that the differential operator of the influence function is suitable neither for $\widehat{Err}_{\mathbf{t}}^{(*)}$ nor $\widehat{AUC}_{\mathbf{t}}^{(*)}$.

The other way to define the estimated true error rate is the leave-one-out bootstrap defined in (2.38). The two estimators are very close in their estimated values, in particular asymptotically. In addition, both are smooth, yet, $\widehat{Err}_{\mathbf{t}}^{(1)}$ has an inner summation, as in (2.37), which is a smooth function too. This is so since any change in a sample case will cross many bootstrap-based decision surfaces (some extreme violations to this fact may occur under particular classifiers). The smoothness of these two estimators along with the non-smooth component $Err_{\mathbf{t}^{*b}}(\hat{F}_{\varepsilon,i}^{(*)})$ is shown in Figure 3.17.

In brief $\widehat{Err}_{\mathbf{t}}^{(1)}$ and $\widehat{Err}_{\mathbf{t}}^{(*)}$ almost give the same estimated value and both are smooth. However, the former has a smooth inner summation, which makes it suitable for using the differential operator of the influence function. On the contrary, the latter has a non-smooth inner summation, which is not suitable for the differential operator of the influence function.

The above discussion suggests introducing an analogue to $\widehat{Err}_{\mathbf{t}}^{(1)}$ for measuring the performance in AUC. This new estimator is motivated from (3.23) the same way the estimator $\widehat{Err}_{\mathbf{t}}^{(1)}$ was motivated from (2.37). The simple bootstrap estimator (3.23) can be rewritten as:

$$\widehat{AUC}_{\mathbf{t}}^{SB} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} E_* \left[\psi(\hat{h}_{\mathbf{t}^*}(x_i), \hat{h}_{\mathbf{t}^*}(x_j)) \right] \tag{3.39}$$

$$= \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \sum_{b=1}^{B} \left[\psi(\hat{h}_{\mathbf{t}^{*b}}(x_i), \hat{h}_{\mathbf{t}^{*b}}(x_j))/B \right] \tag{3.40}$$

In words, the procedure is to select a pair (one observation from each class) and calculate for that pair the mean—over many bootstrap replications and training—of the Mann-Whitney kernel. Then, average over all possible pairs. This procedure will be

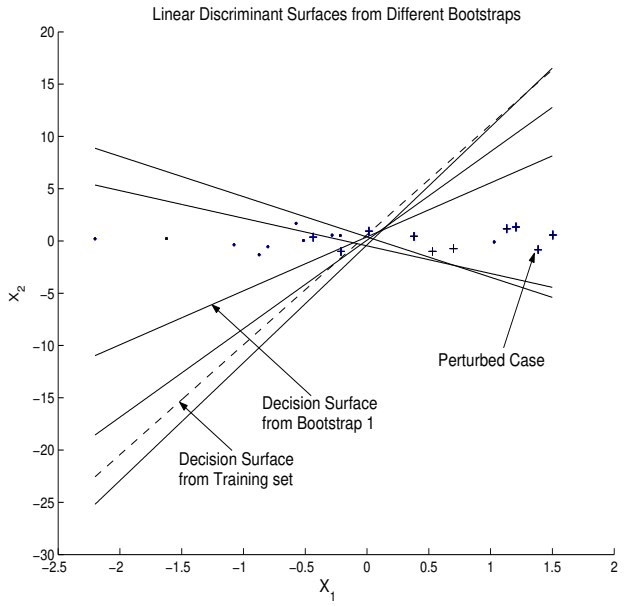

Figure 3.16. Different linear decision surfaces obtained by training on different bootstrap replicates from the same training data set. The first case from class 1 is chosen for perturbation. Changing a feature, e.g., X_1, has no change on the decision value of a single surface unless the case crosses that surface.

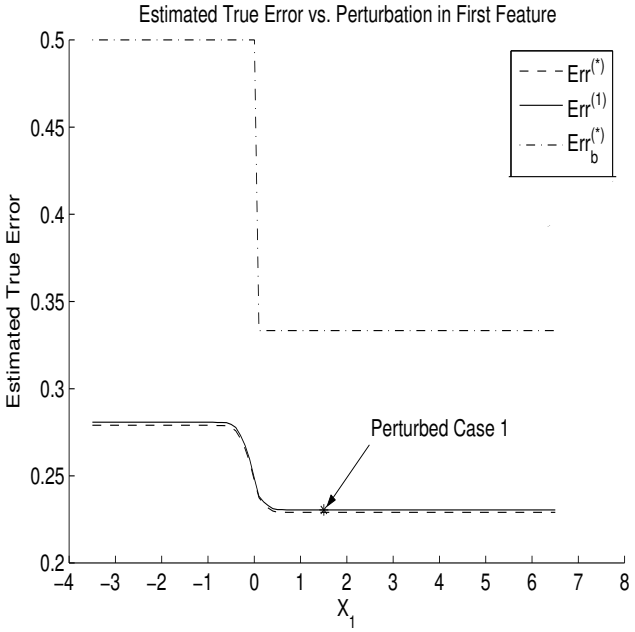

Figure 3.17. The two estimators $\widehat{Err}_{\mathbf{t}}^{(*)}$, $\widehat{Err}_{\mathbf{t}}^{(1)}$, and the component $Err_{\mathbf{t}^{*b}}(\widehat{F}_{\varepsilon,i}^{(*)})$ estimated after training on the first bootstrap replicate. The first two are smooth while the third is not. The estimated true error is plotted vs. change in the value of the first feature.

optimistically biased because sometimes the testers will be the same as the trainers. To eliminate that bias, the inner bootstrap expectation should be taken only over those bootstrap replications that do not include the pair (t_i, t_j) in the training. Under that

Metric M	LDA	QDA	Diff.
$\mathrm{E}\, M_{\mathbf{t}}$	.7706	.7163	.0543
$\mathrm{SD}\, M_{\mathbf{t}}$	.0313	.0442	.0343
$\mathrm{E}\, \widehat{M}^{(1,1)}$	.7437	.6679	.0758
$\mathrm{SD}\, \widehat{M}^{(1,1)}$	.0879	.0944	.0533
$\mathrm{E}\mathrm{SD}\, \widehat{M}^{(1,1)}$	.0898	.1003	.0708
$\mathrm{SD}\mathrm{SD}\, \widehat{M}^{(1,1)}$	.0192	.0163	.0228

Table 3.3. Estimating the uncertainty in the estimator that estimates the difference in performance of two competing classifiers, the LDA and the QDA. The metric M represents AUC_1 for LDA, AUC_2 for QDA, and Δ for the difference.

constraint, the estimator (3.39) becomes the leave-pair-out bootstrap estimator:

$$\widehat{AUC}^{(1,1)} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \widehat{AUC}_{i,j}, where \tag{3.41}$$

$$\widehat{AUC}_{i,j} = \sum_{b=1}^{B} I_j^b I_i^b \psi(\hat{h}_{\mathbf{t}^{*b}}(x_i), \hat{h}_{\mathbf{t}^{*b}}(x_j)) / \sum_{b'=1}^{B} I_j^{b'} I_i^{b'} \tag{3.42}$$

The two estimators $\widehat{AUC}^{(*)}$ and $\widehat{AUC}^{(1,1)}$ produce very similar results; this is expected since they both estimate the same thing, i.e., the mean AUC. However, the inner component $\widehat{AUC}_{i,j}$ of the estimator $\widehat{AUC}^{(1,1)}$ also enjoys the smoothness property of $\widehat{Err}^{(1)}$ discussed above. Chapter 4 discusses how to estimate the uncertainty of the estimator $\widehat{AUC}^{(1,1)}$ using the influence function.

3.4. Two Competing Classifiers

If the assessment problem is how to compare two classifiers, then the metric to be used is the conditional difference

$$\Delta_{\mathbf{t}} = AUC_{1_{\mathbf{t}}} - AUC_{2_{\mathbf{t}}}, \tag{3.43}$$

or the mean, unconditional, difference

$$\Delta = \mathrm{E}\, \Delta_{\mathbf{t}} = \mathrm{E}\left[AUC_{1_{\mathbf{t}}} - AUC_{2_{\mathbf{t}}}\right] \tag{3.44}$$

Then it is obvious that there is nothing new in the estimation task, i.e., it is merely the difference of the performance estimate of each classifier, i.e.,

$$\widehat{\Delta} = \widehat{\mathrm{E}\, AUC_{1_{\mathbf{t}}}} - \widehat{\mathrm{E}\, AUC_{2_{\mathbf{t}}}}, \tag{3.45}$$

where each of the two estimators in (3.45) is obtained by any of the estimators discussed in section 3.2. A natural candidate, from the point of view of the present dissertation is the leave-pair-out estimator $\widehat{AUC}^{(1,1)}$—because of the weak correlation and the smoothness issues discussed in the current chapter. Then two questions arise:

First, how to estimate the variance of $\Delta_{\mathbf{t}}$. That is, if $\Delta_{\mathbf{t}} > 0$ then how uncertain are we when saying classifier 1 is better than classifier 2—or vice versa. This problem is the same as estimating the variance of one classifier $\mathrm{Var}\, AUC_{\mathbf{t}}$. It can be pursued by adopting either the one-data-set or the two-data-set framework. The former is the framework that has been used, so far, in this dissertation for estimating $AUC_{\mathbf{t}}$, $\mathrm{E}\, AUC_{\mathbf{t}}$, and $\mathrm{Var}\, \widehat{AUC}_{\mathbf{t}}$, but not yet $\mathrm{Var}\, AUC_{\mathbf{t}}$, which is one of the future work proposals. The latter is the framework considered in Chapter 6 to estimate the same metrics as well as $\mathrm{Var}\, AUC_{\mathbf{t}}$ and other important metrics.

Second, how to estimate the uncertainty of $\widehat{\Delta}$. This is very similar to estimating the variance in $\widehat{\mathrm{E}\, AUC_{\mathbf{t}}}$, which is discussed in Chapter 4. There is nothing new in estimating $\mathrm{Var}\, \widehat{\Delta}$. It is obtained by replacing $\widehat{AUC}^{(1,1)}$, in Chapter 4, by the statistic $\widehat{\Delta}$ in (3.45). Typical values are given in Table 3.3, for demonstration, when comparing the linear and quadratic discriminants, where the training set size per class is 20 and number of features is 4.

Estimating the uncertainty in $\widehat{\Delta}$ should in fact be a central point for the field of statistical pattern recognition, or even computational intelligence in general. Most practitioners simply provide simple estimates of the conditional performance of their favorite classifier, and similarly for a competing classifier. It is rare to see estimates of the uncertainty of measures of classifier performance, and especially rare to see estimates of the uncertainty in the difference of measures of performance of competing classifiers.

3.5. The Partial Area Under the ROC Curve

All of what have been discussed, i.e., assessing classifiers under the ROC analysis, concerns the total area under the curve, i.e., the AUC. In that sense a classifier is superior to another if it has a larger AUC. However, in the ROC space, two ROC curves,

for two different classifiers, may cross each other. Hence, they interchange the superiority region, i.e., the higher ROC before the crossing point will be the lower after it. If some information is available on the operating region, then it will not be utilized if the AUC is used as a summary measure. This proposes the Partial Area Under the Curve (PAUC), which is discussed in Chapter 5.

3.6. Assessing Classifiers From Two Independent Data Sets

All the former matter was regarding assessing classifiers, in terms of AUC or PAUC, from a single data set. That is, the available data are considered to be a single data set without deliberate separation between the trainers and testers. This enabled us to utilize the most available information for training and testing by training on bootstrap replications from the available data then testing on the remaining observations not included in the bootstrap replications. In some situations, e.g., in a regulatory setting, it is required to separate, before hand, the training set from the testing set. In such a paradigm, no contamination or mixing of information across the independent sets is allowed; training is pursued on a completely separate set of the testing set. This principle is called the data hygiene regulation. Chapter 6 considers the mentioned problem and introduces a formal solution.

CHAPTER 4

Estimating the Uncertainty in the Estimated Mean Area Under the ROC Curve of a Classifier

4.1. Introduction

In Chapter 3 we found that the three bootstrap-based estimators proposed by Efron and Tibshirani (1997), namely, the (∗), the .632, and the .632+ bootstrap, showed comparable performance in terms of a mean-square error measure. In the present chapter, we focus on the leave-pair-out bootstrap estimator $\widehat{AUC}^{(1,1)}$ and estimate its uncertainty in terms of the standard error measure. We adopt this approach because all of the estimates are only weakly correlated with the corresponding true values, i.e., conditional on the given training set (see Section 3.2.3). The .632 and .632+ estimators have a component of the apparent performance which is unsmooth, while the (∗) estimator has an unsmooth inner component (see Section 3.3).

4.2. Influence Function and Estimating the Variance of $\widehat{AUC}^{(1,1)}$

When the distribution of the data is known, the optimal classifier is the Bayes one; then, in principle, the population parameters, e.g., class means, covariances, etc., can be obtained in closed form, or at least numerically if the closed form is prohibitive. For mathematical analysis of classifier performance under the multinormal assumption, see Fukunaga (1990). When either the underlying form of the distribution is unknown, or the form is known but its parameters must be estimated, any estimator of the mean performance of the classifier is itself a random variable and a function of the design data set. There are many approaches to estimating the variability of a statistic nonparametrically, e.g., jackknife, bootstrap, influence function, etc. (see Chapter 2). In our case the statistic of interest, i.e., the leave-pair-out estimator (3.41), is a bootstrap-based estimator. Bootstrapping again to estimate the uncertainty is, at the best, not computationally efficient; in many situations it will not be feasible since the time for even one pass of training is typically very long. This is one of the situations where the method of the influence function offers a practical solution.

Equation (2.29) gives the nonparametric estimate of variance for a statistic s under the empirical distribution $\hat{F}$. Efron and Tibshirani (1997) used this powerful approach to estimate the standard error of the estimator $\widehat{Err}^{(1)}$. In the present chapter, we extend their study to the task of estimating the standard error of the estimator $\widehat{AUC}^{(1,1)}$.

Assume that the available data set $\mathbf{t}$ is comprised of two data sets, one for each class, i.e. $\mathbf{t} = \mathbf{t}_1 \cup \mathbf{t}_2$, $\mathbf{t}_1 \in \omega_1$, *and* $\mathbf{t}_2 \in \omega_2$. The sizes of the sets are $n_1 + n_2 = N$. We assume the two sets are independent, and $\mathbf{t}_1 \sim F_1$ and $\mathbf{t}_2 \sim F_2$. The functional s now will be our metric $\widehat{AUC}^{(1,1)}$; then there is no covariance term and it is easy to see that:

$$\widehat{sd} = \sqrt{\frac{1}{n_1^2}\sum_{i=1}^{n_1}\hat{U}_{1_i}^2 + \frac{1}{n_2^2}\sum_{j=1}^{n_2}\hat{U}_{2_j}^2}, \quad where \tag{4.1}$$

$$\hat{U}_{k_i} = \left.\frac{\partial\widehat{AUC}^{(1,1)}(\hat{F}_{k_{\varepsilon,i}})}{\partial\varepsilon}\right|_{\varepsilon=0}, \; t_i \in \mathbf{t}_k, \; k = 1,2 \tag{4.2}$$

The perturbation (2.25) when applied, independently, for $\hat{F}_1$ and $\hat{F}_2$ gives:

$$\hat{f}_{k_{\varepsilon,i}}(x_j) = \begin{cases} \left[1 + (n_k\delta_{ij} - 1)\varepsilon\right]/n_k, & x_j \in \omega_k \\ 1/n_{3-k}, & x_j \in \omega_{3-k} \end{cases} \tag{4.3}$$

where $k = 1,2$, and the subscript $3-k$ accounts for class 2 and 1 respectively. This affects the probability mass of a bootstrap replication b. A straightforward extension of the lemma of Efron (1992) allows us to obtain the probability $g_{k_{\varepsilon,i}}(b)$ of the bootstrap b that includes the observation $t_i \in \omega_k \, (k = 1,2) \quad N_i^b$ times. It is easy to show that:

$$g_{k_{\varepsilon,i}}(b) = (1-\varepsilon)^{n_k}\left(1 + \frac{n_k\varepsilon}{1-\varepsilon}\right)^{N_i^b}(1/n_1)^{n_1}(1/n_2)^{n_2} \tag{4.4}$$

A similar formula was given originally in Efron (1992) for one distribution F. Under $\hat{F}_{1_{\varepsilon,i}}$ and $\hat{F}_{2_{\varepsilon,i}}$ the estimator $\widehat{AUC}^{(1,1)}$ is given by:

$$\widehat{AUC}^{(1,1)}_{\varepsilon,i}\left(\hat{F}_{k_{\varepsilon,i}}\right)=\sum_{j_2=1}^{n_2}\sum_{j_1=1}^{n_1}\hat{f}_{1_{\varepsilon,i}}(j_1)\hat{f}_{2_{\varepsilon,i}}(j_2)\frac{\sum_{b=1}^{B}\psi(\hat{h}_b(j_1),\hat{h}_b(j_2))I^b_{j_1}I^b_{j_2}g_{k_{\varepsilon,i}}(b)}{\sum_{b=1}^{B}I^b_{j_1}I^b_{j_2}g_{k_{\varepsilon,i}}(b)} \tag{4.5}$$

The derivative $\partial\widehat{AUC}^{(1,1)}(\hat{F}_{k_{\varepsilon,i}})/\partial\varepsilon$ in (4.2) can be obtained formally, similarly to what was done in Efron and Tibshirani (1995), by writing (4.5) as:

$$\widehat{AUC}^{(1,1)}_{\varepsilon,i}=\sum_{j_2=1}^{n_2}\sum_{j_1=1}^{n_1}A(\varepsilon)\frac{B(\varepsilon)}{C(\varepsilon)} \tag{4.6}$$

where $A(\varepsilon)=f_{1_{\varepsilon,i}}(j_1)f_{2_{\varepsilon,i}}(j_2)$, $B(\varepsilon)=\Sigma_b\psi\left(\hat{h}_b(j_1),\hat{h}_b(j_2)\right)I^b_{j_1}I^b_{j_2}g_{k_{\varepsilon,i}}(b)$, and $C(\varepsilon)=\Sigma_b I^b_{j_1}I^b_{j_2}g_{k_{\varepsilon,i}}(b)$. Then the derivative $\hat{U}_{k_i}$ can be written as:

$$\hat{U}_{k_i}=I+II+III \tag{4.7}$$

where,

$$\begin{aligned}I&=\Sigma_{j_2}\Sigma_{j_1}A'(0)B(0)/C(0),\\ II&=\Sigma_{j_2}\Sigma_{j_1}A(0)B'(0)/C(0),\\ III&=-\Sigma_{j_2}\Sigma_{j_1}A(0)C'(0)B(0)/C^2(0).\end{aligned} \tag{4.8}$$

To simplify the notation, if $t_i\in\mathbf{t}_1$ then the derivatives of (4.3) and (4.4) are given by:

$$\frac{\partial f_{1_{\varepsilon,i}}(j_1)}{\partial\varepsilon}=\delta_{ij_1}-1/n_1, \tag{4.9}$$

$$\frac{\partial g_{1_{\varepsilon,i}}(b)}{\partial\varepsilon}=n_1(N^b_i-1)(1/n_1)^{n_1}(1/n_2)^{n_2} \tag{4.10}$$

Then the three terms above are given by:

$$I=\widehat{AUC}_i-\widehat{AUC}^{(1,1)}, \tag{4.11}$$

$$II=\sum_{j_1}\sum_{j_2}\frac{1}{n_2}\frac{\sum_b\psi(\hat{h}_b(j_1),\hat{h}_b(j_2))I^b_{j_1}I^b_{j_2}(N^b_i-1)}{\sum_{b'}I^{b'}_{j_1}I^{b'}_{j_2}}, \tag{4.12}$$

$$III=-\frac{1}{E_*\left[I^b_{j_1}I^b_{j_2}\right]}\frac{1}{n_2}\sum_{j_2}\sum_{j_1}AUC_{j_1j_2}\,\mathrm{Cov}_*\left[I^b_{j_1}I^b_{j_2},N^b_i\right], \tag{4.13}$$

$$\widehat{AUC}_i=\frac{1}{n_2}\sum_{j_2}\frac{\sum_b\psi(\hat{h}_b(i),\hat{h}_b(j_2)I^b_iI^b_{j_2}}{\sum_{b'}I^{b'}_iI^{b'}_{j_2}}, \tag{4.14}$$

$$\widehat{AUC}_{j_1j_2}=\frac{\sum_b\psi(\hat{h}_b(j_1),\hat{h}_b(j_2))I^b_{j_1}I^b_{j_2}}{\sum_{b'}I^{b'}_{j_1}I^{b'}_{j_2}} \tag{4.15}$$

where the expectation E_* and the covariance Cov_* are taken over all possible combinations of bootstrap replications, i.e., $n_1^{n_1}\cdot n_2^{n_2}$ replications. That is ,$E_*[X]=\Sigma_bX_b/B$, and $\mathrm{Cov}_*[x,y]=E_*[x(y-E_*y)]$. Then it is not hard to show that:

$$\mathrm{Cov}_*\left[I^b_{j_1}I^b_{j_2},N^b_i\right]=E_*\left[I^b_iI^b_{j_2}\right]/(n_1-1) \tag{4.16}$$

We use in our simulations the balanced bootstrap mechanism. This is implemented by simulating a string consisting of indices ranging from 1 to n and copying this string B times. The $n\cdot B$ indexes are shuffled randomly then repartitioned into B strings. This mechanism is proposed by Davison, Hinkley and Schechtman (1986) to speed up convergence towards the asymptotic bootstrap expectation. Just as was noticed by Efron and Tibshirani (1997) for the true error simulations, it has little effect on the results here. In that case we have:

$$E_*N^b_i=1 \tag{4.17}$$

Combining (4.11) through (4.17) and substituting back in (4.7), a little algebra yields:

$$\begin{aligned}\hat{U}_{1_i}=&\left(2+\frac{1}{n_1-1}\right)(\widehat{AUC}_i-\widehat{AUC}^{(1,1)})\\ &+\sum_b\frac{1}{n_2}(N^b_i-1)\sum_{j_2}\sum_{j_2}\frac{\psi(\hat{h}_b(j_1),\hat{h}_b(j_2))I^b_{j_1}I^b_{j_2}}{\sum_{b'}I^{b'}_{j_1}I^{b'}_{j_2}}\end{aligned} \tag{4.18}$$

.1225	.1097	.1027	.1211	.1458	.1007
.1211	.0834	.1021	.1003	.1064	.1042
.0997	.1170	.1069	.0913	.0996	.1288
.1082	.0983	.1177	.1189	.1159	.1021
.1268	.0815	.1031	.1147	.0933	.1113
.0953	.1109	.1073	.1029	.1259	.1120
.1098	.1132	.0856	.1021	.1319	.0941
.0979	.1071	.1126	.0992	.1091	.1055
.1156	.1154	.1140	.1258	.1065	.1157
.1125	.0874	.0932	.0997	.1008	.1025
.1138	.1333	.1159	.1139	.1195	.1149
.1147	.1074	.1007	.1017	.1102	.1054
.1093	.1330	.1133	.1133	.1161	.0989
.1157	.1212	.1100	.1055	.1107	.0947
.1171	.1121	.0952	.1015	.1192	.1078

Table 4.1. Ninety estimates of the variance of $AUC^{(1,1)}$ using the method of the influence function. Each estimate is obtained from a single data set, and the entire process is repeated over independent MC trials. These 90 values have an average of .1090 (compare to 0.0930, the true standard deviation obtained from MC) with standard deviation of .0114.

For $t_i \in \mathbf{t}_2$ (4.18) will be the same for $\hat{U}_{2_i}$, but with exchange of n_1 and n_2.

The derivative given in (4.1) can also be evaluated numerically for (4.5) by substituting a very small value for ε, typically 10^{-3} to 10^{-4} was adequate in our simulations.

4.3. Simulation Results

The concept of the influence function is completely nonparametric. To demonstrate it, however, it is most straightforward to simulate data from a particular distribution. Consider the same experiments of Section 3.2. We assume, for simplicity, $n_1 = n_2 = n$. Monte-Carlo (MC) simulation was carried out, typically with 10,000 trials, to closely approximate the true variance of the finite-sample estimator $\widehat{AUC}^{(1,1)}$. In each trial n observations are sampled from the underlying distribution and over the 1000 trials the true variability is calculated. From each trial, the true variability is also estimated using the influence function method. The results show that—for the case of linear and quadratic classifiers studied here—the true variability can be estimated with very little bias (the error is always on the order of one standard deviation) over a wide range of dimensionalities (from 2 to 15) and sample sizes (from 6 to 100 cases per class). Table 4.1 shows the standard error estimate of the first 90 trials obtained from an experiment with $p = 5$, $n = 20$, $c = .5477$ and $B = 5000$. With these parameters, the true mean AUC of the classifier was 0.7575. The values should be compared to the true standard error obtained from MC, which was .0930. The 90 estimation values have an average of .1090 with standard deviation of .0114.

Table 4.2 presents results from a variety of experiments with different combinations of classifier, dimensionality, and sample size. The LDF classifier is the linear discriminant function under the assumption that the covariance matrices of the two classes are equal. The QDF is the quadratic one, where each covariance matrix is estimated separately. In all experiments, c is set to equal to $\sqrt{1.5/p}$ which gives a Mahalanobis distance of 1.5. The true mean and standard deviation of the estimator $\widehat{AUC}^{(1,1)}$ is obtained over 10,000 trials of MC simulation. However, the mean and standard deviation for the influence-function-based estimator, $\widehat{sd}$, is obtained from only the first 100 trials because of the limitation of execution time.

4.4. Experiments With Real Data

Another experiment is carried out on a large real-world data set obtained from the UCI repository of machine learning databases Newman and Asuncion (2007). The data set used is the "Adult" data of Ron Kohavi and Barry Becker. We used four of the continuous features with 1000 testers per class and 10,687 trainers per class. In order to challenge the small-sample performance of our estimators—and obtain Monte Carlo estimates of the corresponding population quantities—the trainers are divided into 534 groups (the number of MC trials), each with 20 observations per class. The classifier used was the linear discriminant classifier and gave a true mean AUC of .7589. The AUC obtained over the MC simulations had a "true" standard deviation of .0902. Table 4.3 shows the estimation of the standard deviation obtained from the influence function method when applied to the first 90 trials. These 90 estimates have an average of .0963 (vs. the true value .0902) with standard deviation of .0188.

Classifier	p	n	$MC = 10000$		$MC = 100$	
			$E_{MC}(\widehat{AUC}^{(1,1)})$	$sd_{MC}(\widehat{AUC}^{(1,1)})$	$E_{MC}(\widehat{sd})$	$sd_{MC}(\widehat{sd})$
LDA	15	40	.7012	.0654	.0734	.0080
LDA	15	25	.6567	.0815	.0925	.0120
LDA	10	40	.7296	.0641	.0685	.0083
LDA	10	25	.6951	.0826	.0884	.0124
LDA	10	15	.6429	.1048	.1091	.0196
LDA	5	40	.7664	.0589	.0631	.0095
LDA	5	25	.7441	.0796	.0795	.0148
LDA	5	15	.7097	.1089	.1102	.0226
LDA	2	25	.7868	.0693	.0696	.0156
LDA	2	15	.7710	.0986	.0956	.0288
LDA	2	6	.7129	.1768	.1571	.0666
QDA	2	25	.7580	.0792	.0789	.0164
QDA	2	15	.7257	.1088	.1064	.0273

Table 4.2. Different experiments under different dimensionality p, sample size n, and two different classifiers. The true mean and standard deviation for the estimator $AUC^{(1,1)}$ are obtained from 10, 000 MC trials. The mean and standard deviation of the estimated standard deviation, sd, are obtained from 100 MC trials. Notice the closeness of the mean of the estimates, $E_{MC}(sd)$, to the MC results.

.1306	0943	.1018	.1083	.1075	.0829
.1232	0783	.1161	.0837	.0790	.1075
.0883	1029	.0824	.1132	.1566	.0884
.1345	1023	.0892	.0813	.1050	.1140
.0864	0828	.1065	.0684	.0726	.0904
.1021	0724	.1171	.0885	.0797	.0954
.0671	0928	.1331	.1042	.0725	.0603
.0786	0849	.1053	.0884	.1017	.0813
.1303	0876	.1089	.0750	.0797	.0648
.1002	1038	.0716	.1105	.0605	.0879
.1101	0929	.0911	.1041	.0682	.1277
.1119	0862	.1271	.0909	.0867	.1022
.0929	1294	.1102	.0785	.0855	.1080
.1005	0728	.0979	.0926	.0944	.0804
.1306	0943	.1018	.1083	.1075	.0829

Table 4.3. Ninety estimates of the variance of $AUC^{(1,1)}$ using the method of the influence function on the real data set experiment. Each estimate is obtained from a single data set, and the entire process is repeated over independent MC trials. These 90 values have an average of .0963 (compare to .0902, the true standard deviation obtained from MC) with standard deviation of .0188.

4.5. Chapter Summary

Our study for the bootstrap-based estimators indicates that the estimator $\widehat{AUC}^{(*)}$ discussed in Chapter 3 and the new estimator $\widehat{AUC}^{(1,1)}$ defined here estimate the same metric, the mean AUC of a classification rule, that is, not conditional on a particular training set. The $\widehat{AUC}^{(1,1)}$ estimator enjoys the smoothness property of $\widehat{Err}^{(1)}$ and leads to a powerful method for estimating the standard error of AUC estimates using the same bootstrap samples used for the mean estimate.

A key feature of this method of estimating the standard error of the AUC estimates is that it reflects the finite size of samples used to train the classifier as well as the finite size of samples used to assess its performance. Methods of assessing the uncertainty of performance estimates based on conventional cross-validation, e.g., Bradley (1997), do not incorporate the variability inherent in the finite training sample because the training sets in the various partitions are similar to one another. Such methods are therefore essentially conditional on the given training set.

Application of the approach to the linear classifier over a range of dimensionalities and finite-sample sizes produced results with very small bias.

CHAPTER 5

The Partial Area under the ROC Curve: Its Properties and Nonparametric Estimation for Assessing Classifier Performance

5.1. Introduction

When two competing ROC curves cross, as in Figure 5.1, the AUC is no longer an unambiguous summary measure of performance. In addition, if it is known a priori that the classifier will be used only over a narrow range of environments, a different summary measure of performance is desirable. The most commonly used distinction in medical testing, for example, is that between the screening environment where the disease prevalence is typically low and that of the diagnostic work-up environment where the disease prevalence is typically higher. Thus, it may be of interest in the screening environment to restrict consideration of the ROC curve to the low false-positive region (in which case, classifier 1 in the figure would be superior) and in the diagnostic environment to restrict consideration of the ROC curve to the high true-positive region (in which case, classifier 2 would be superior). These considerations suggest the use of a summary measure of performance that embraces only the partial area under the ROC curve (PAUC) in a particular operating region of interest.

In Section 5.2, we will give a formal definition of the PAUC and derive several of its properties. In Section 5.3.1, we estimate the PAUC nonparametrically (cf. parametric applications in Jiang, Metz and Nishikawa, 1996; McClish, 1989) and estimate, as well, the uncertainty in the estimation. Section 5.4 provides experimental and simulation results.

5.2. The Partial Area under the Curve (PAUC)

5.2.1. Definitions and properties in the mean

In this section we will give a formal definition of a nonparametric statistic that is a generalization of the Mann-Whitney version of the Wilcoxon statistic and show that this statistic corresponds to what is intuitively understood as the PAUC. We then derive several properties of this statistic in the mean. By $\eta_{\mathbf{t}}$ we denote a classifier conditional on a particular training set $\mathbf{t}$. As described in Chapter 3, the resulting performance metrics thus are said to be conditional on $\mathbf{t}$; they become random variables if one conceives of replicating a given experiment with independent training sets drawn from the population.

Definition **5.1**. Assume we are given a classification rule $\eta_{\mathbf{t}}$ and the corresponding log-likelihood ratio $\hat{h}_{\mathbf{t}}$. Assume that it is known from the testing environment that only threshold or cut-off values th greater than a specified threshold th_c are of interest. Then, an appropriate metric for measuring the ability of the classifier to separate the two classes is the separability function $SP_{\mathbf{t}}(th_c)$ defined by:

$$SP_{\mathbf{t}}(th_c) = \Pr\left[\hat{h}_{\mathbf{t}}(X|\omega_1) > \hat{h}_{\mathbf{t}}(X|\omega_2) > th_c\right] \tag{5.1}$$

Theorem **5.2**. The metric defined in (5.1) is equal to the partial area under the ROC curve (PAUC) given by:

$$\begin{aligned} SP_{\mathbf{t}}(th_c) &= \int_0^c TPF_{\mathbf{t}}\, dFPF_{\mathbf{t}} \\ &= PAUC_{\mathbf{t}}(th_c),\ where \\ c &= \int_{th_c}^{\infty} dF_{\hat{h}_{\mathbf{t}}(X|\omega_2)} \end{aligned} \tag{5.2}$$

Proof. For ease of notation, we name the random variables $\hat{h}_{\mathbf{t}}(X|\omega_i)$, $i = 1,2$ by X and Y respectively, where, for simplicity, F_X and F_Y will be referred to as the left and right distributions and F_{XY} as the joint distribution, similar to the diseased and

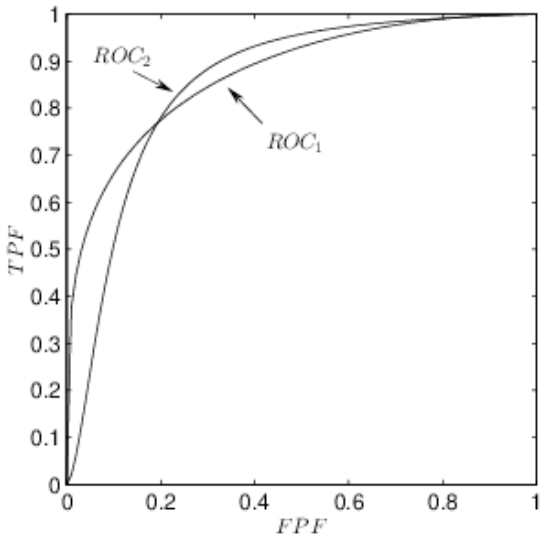

Figure 5.1. Two ROC curves for two different classifiers. Classifier 1 outperforms classifier 2 at the lower scale of the FPF; then classifier 2 is superior to classifier 1 until the end of the FPF scale.

nondiseased distributions in diagnostic testing (see Figure 1.3); then:

$$\begin{aligned} SP_{\mathbf{t}}(th_c) &= \int_{th_c}^{\infty} dx \int_{x}^{\infty} dy\, f_{XY}(x,y) \\ &= \int_{th_c}^{\infty} dx\, f_X(x) \int_{x}^{\infty} dy\, f_y(y) \\ &= \int_{\infty}^{th_c} -dx\, f_X(x) \int_{x}^{\infty} dy\, f_y(y) \\ &= \int_0^c dFPF_{\mathbf{t}}(x) TPF_{\mathbf{t}}(x),\ where \\ c = FPF_{\mathbf{t}}(th_c) &= \int_{th_c}^{\infty} f_X(x)dx \end{aligned} \tag{5.3}$$

Comparing this with (3.15) shows that $SP_{\mathbf{t}}(-\infty)$ is the conventional $AUC_{\mathbf{t}}$ and $PAUC_{\mathbf{t}}(\infty) = 0$. ■

Theorem **5.3**. The conditional performance of a classification rule $\eta_{\mathbf{t}}$ trained on the training data set $\mathbf{t}$ and measured in terms of the PAUC is a monotonically nonincreasing function of the cutoff threshold th_c.

Proof. Assume that $th_{c_1} < th_{c_2}$, then by Theorem 5.2 we have:

$$\begin{aligned} PAUC_{\mathbf{t}}(th_{c_2}) &= PAUC_{\mathbf{t}}(th_{c_1}) - (\Pr_{\mathbf{t}}[Y > X > th_{c_1}] - \Pr_{\mathbf{t}}[Y > X > th_{c_2}]) \\ &= PAUC_{\mathbf{t}}(th_{c_1}) - \Delta, \end{aligned} \tag{5.4}$$

where Δ is the probability measure, in the two-dimensional probability subspace, of the set $(Y > X) \cap (th_{c_1} < X < th_{c_2})$—a plot in the X–Y probability space analogous to Figure 5.2 makes this visually obvious. Hence, $\Delta \geq 0$ and $PAUC_{\mathbf{t}}(th_{c_2}) \leq PAUC_{\mathbf{t}}(th_{c_1})$. ■

Corollary **5.4**. The expected value of $PAUC_{\mathbf{t}}(th_c)$, i.e., $E_{\mathbf{t}}[PAUC_{\mathbf{t}}(th_c)]$, over the population of training sets t is a monotonically nonincreasing function of the cutoff threshold th_c.

Proof. The proof follows immediately by taking the expectation of both sides of (5.4). ■

5.2.2. Definitions and properties of the variance

The next critical issue when assessing a classifier is the variance, i.e., the uncertainty of the performance metric under variation of the training sets. One might expect from the monotonicity of the mean of the PAUC with the threshold that the variance of the PAUC might behave qualitatively in the same manner. Our simulation results, however, reviewed below in Section 5.4, show that the variance of the classifier performance, measured in terms of the PAUC over the population of trainers, increases

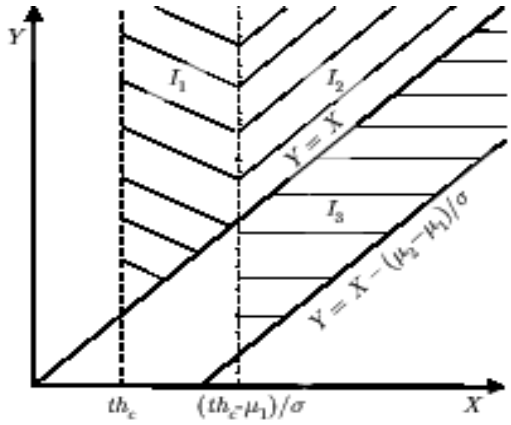

Figure 5.2. The different areas of integration, I_1, I_2, and I_3, in the X–Y space.

with the threshold value until it reaches a peak, and then it decays towards zero. This counter-intuitive result led us to explore the issue theoretically for some special cases that are tractable. The following theorem gives an interpretation for the observed phenomenon under the assumption that the distributions for the log-likelihood ratio come from the location-scale family; see Casella and Berger (2002, pp. 241)

Theorem **5.5**. Assume that the two distributions of the log-likelihood ratios under the variation of the training data set come from the location-scale families, with location parameters $\theta_1(\mathbf{t})$, $\theta_2(\mathbf{t})$, and a common scale parameter $\theta_3(\mathbf{t})$, whose means and covariances are given by m_i, σ_{ij}, $i, j = 1,2,3$ respectively. These random parameters depend on the particular training sample. The first-order Taylor-series approximation for the variance of $PAUC_{\mathbf{t}}(th_c)$ of a classifier is given by the quadratic form:

$$\mathrm{Var}\left[PAUC_{\mathbf{t}}\left(th_c\right)\right] \approx \mathbf{d}'\Sigma\mathbf{d}, \tag{5.5}$$

$$\mathbf{d}' = \left(\frac{(A_1 - A_2)}{m_3}, -A_2, \frac{\left(\left(th_c - m_1\right)A_1 - \left(m_1 - m_2\right)A_2\right)}{m_3^2}\right),$$

$$A_1 = f_X\left(\frac{th_c - m_1}{m_3}\right)\left(1 - F_Y\left(\frac{th_c - m_2}{m_3}\right)\right),$$

$$A_2 = \int_{\frac{th_c - m_1}{m_3}}^{\infty} f_X\left(x\right) f_Y\left(x + \frac{m_1 - m_2}{m_3}\right) dx$$

Proof. For any set of random variables T_i, $1 \le i \le k$ define the random vector $T = (T_1, \ldots, T_k)$. Any scalar differentiable function $g(T)$ can be approximated using the first-order Taylor-series expansion by:

$$g(t) \approx g\left(E\left[T\right]\right) + \sum_{i=1}^{k} g^{(i)}\left(E\left[T\right]\right)\left(t_i - E\left[T_i\right]\right), \tag{5.6}$$

where $g^{(i)}\left(E\left[T\right]\right) = \partial g\left(t\right)/\partial t_i|_{t=E[T]}$. Then the variance $\mathrm{Var}\left[g\left(T\right)\right]$ can be approximated by:

$$\mathrm{Var}\left[g\right] \approx \sum_{i=1}^{k}\left[g^{(i)}\left(E\left[T\right]\right)\right]^2 \mathrm{Var}\left[T_i\right] + 2\sum_{i>j} g^{(i)}\left(E\left[T\right]\right) g^{(j)}\left(E\left[T\right]\right) \mathrm{Cov}\left(T_i, T_j\right) \tag{5.7}$$

It is convenient to write (5.7) in more compact vector-matrix notation as:

$$\mathrm{Var}\left[g\right] = \mathbf{d}'\Sigma\mathbf{d}, \tag{5.8}$$

where $\mathbf{d} = \left(g^{(1)}, \ldots, g^{(k)}\right) = \nabla g$, where $\nabla = \left(\partial/\partial t_1, \ldots, \partial/\partial t_k\right)$; $\Sigma = ((\sigma_{ij}))$, $\sigma_{ij} = \mathrm{Cov}\left(T_i, T_j\right)$. For our problem, under the location-scale-family assumption, the two distributions are given by $f_X\left(\left(x - \theta_1\right)/\theta_3\right)/\theta_3$ and $f_Y\left(\left(y - \theta_2\right)/\theta_3\right)/\theta_3$ respectively. By replacing g in the above equations by $PAUC_{\mathbf{t}}(th_c)$ and T by $\left(\theta_1\left(\mathbf{t}\right), \theta_2\left(\mathbf{t}\right), \theta_3\left(\mathbf{t}\right)\right)$, then using (5.3) and straightforward calculus, (5.5) follows. ∎

Corollary **5.6**. The variance of the PAUC, unlike the conditional and the mean PAUC, is not necessarily a monotonically decreasing function of the threshold.

Proof. The first derivative of (5.8) is given by the bilinear form:

$$\frac{\partial \mathrm{Var}\left[PAUC_{\mathbf{t}}(th_c)\right]}{\partial th_c} = 2\mathbf{d}'\Sigma\frac{\partial \mathbf{d}}{\partial th_c},\ where \tag{5.9}$$

$$\begin{aligned}
\frac{\partial d_1}{\partial th_c} &= \frac{1}{m_3^2} f'_X\left(\frac{th_c - m_1}{m_3}\right)\left(1 - F_Y\left(\frac{th_c - m_2}{m_3}\right)\right),\\
\frac{\partial d_2}{\partial th_c} &= -\frac{1}{m_3^2} f_X\left(\frac{th_c - m_1}{m_3}\right) f_Y\left(\frac{th_c - m_2}{m_3}\right),\\
\frac{\partial d_3}{\partial th_c} &= (th_c - m_1)\frac{\partial d_1}{\partial th_c} + \frac{1}{m_3}(th_c - m_2)\frac{\partial d_2}{\partial th_c}\\
&\quad + \frac{1}{m_3^2}\left(1 - F_Y\left(\frac{th_c - m_2}{m_3}\right) f_X\left(\frac{th_c - m_1}{m_3}\right)\right)
\end{aligned}$$

It is beyond the scope of the present work to use this integral-differential equation to find the maxima and minima using (5.9). It is sufficient for our present purposes to show that there are conditions under which the variance can exhibit a maximum. Consider the region where the left tail of the right density function has almost decayed, i.e., $f_Y((th_c - m_2)/m_3) \approx 0$, then the first derivative in (5.9) will be zero at the points of maxima and minima of f_X if at those points the following condition is satisfied:

$$\begin{aligned}
&f_X\left(\frac{th_c - m_1}{m_3}\right)\left(\sigma_{13} + \frac{th_c - m_1}{m_3}\sigma_{33}\right)/A_2 + \sigma_{33}\left(\frac{m_1}{m_3}\right) - \sigma_{13} \\
&= \sigma_{23} + m_2\left(\frac{\sigma_{33}}{m_3}\right)
\end{aligned} \tag{5.10}$$

Note that $\mathrm{Var}\left[PAUC_{\mathbf{t}}(-\infty)\right] \approx 0$ and $\mathrm{Var}\left[PAUC_{\mathbf{t}}(\infty)\right] = 0$ since $PAUC_{\mathbf{t}}(\infty) = 0$; then if (5.10) is satisfied this will guarantee that $\mathrm{Var}\left[PAUC\right]$ will exhibit a local maximum. Since the L.H.S. of (5.10) is a function of the distribution, while the R.H.S. is function only in the parameters, then for any value given by the L.H.S. there will always be values of the parameters that satisfy the equality. Hence, the proof is complete. ∎

A more clear interpretation for (5.10) is available by considering a special case of the location-family, not scale, distribution where μ_1 and μ_2 are uncorrelated, i.e., $\sigma_{12} = \sigma_{13} = \sigma_{33} = 0$. In such a case, (5.10) will always be satisfied and the variance of the PAUC will always exhibit a maximum at the peak of the left distribution providing the right distribution has decayed sufficiently. This is intuitively clear since in the region of log-likelihood space where the left tail of the right distribution is zero, (5.1) becomes $SP_{\mathbf{t}}(th_c) = \Pr\left[\hat{h}(X|\omega_2) > th_c\right]$, which can be calculated, if we have a sufficiently large number of testers, by counting how many observations achieve this inequality. If th_c occurs at a maximum of f_X, which is a very dense neighborhood of the random variable $\hat{h}(X|\omega_2)$, any sampling variation in the training set will be reflected in a variation of the density function f_X, especially around th_c the dense region, and this variation will be propagated into the fraction of times the inequality is satisfied.

In addition, still under that special case of having $\sigma_{12} = \sigma_{13} = \sigma_{33} = 0$, the PAUC can be seen, approximately, as $\frac{1}{n_1} \times Binomial(n_1, p)$; where n_1 is the number of testers belonging to distribution 1 and having $th_c < \widehat{h}$ and $p = \Pr\left[\hat{h}(X|\omega_2) > th_c\right]$. In such a case the peak of the variance has the value of $p(1-p)/n_1$.

5.3. Nonparametric Estimation

5.3.1. Estimation of Mean Performance

A nonparametric bootstrap-based estimator for the PAUC can be defined by an extension of the estimator proposed in Section 3.2 for estimating the AUC. By defining the kernel:

$$\psi(x, y, th_c) = \begin{cases} 1 & th_c < x < y \\ 0 & otherwise \end{cases} \tag{5.11}$$

where continuous distributions are assumed, the mean PAUC of a classifier will be estimated by:

$$\widehat{PAUC}^{(1,1)}(th_c) = \frac{1}{n_1 n_2}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\widehat{PAUC}_{i,j}(th_c),\ where \tag{5.12}$$

$$\widehat{PAUC}_{i,j}(th_c) = \sum_{b=1}^{B} I_j^b I_i^b \psi\left(\hat{h}_{\mathbf{t}^{*b}}(x_i), \hat{h}_{\mathbf{t}^{*b}}(x_j), th_c\right) / \sum_{b'=1}^{B} I_j^{b'} I_i^{b'}$$

Here, I_i^b is the indicator function that equals one when the bootstrap replication b does not include the observation i, and zero otherwise. The component $\widehat{PAUC}_{i,j}(th_c)$ is the average of the kernel $\psi(i, j, th_c)$ over all bootstrap replications that do not include the observations i and j. This averaging approximates the expectation over the population of training sets. The average in equation (5.12) approximates the expectation over the population of testers.

It is expected that the leave-pair-out estimator will greatly attenuate the increase-in-variance phenomenon described in Corollary 5.6 because of the averaging or smoothing effect of the bootstrap summation in (5.12). This will be demonstrated in the example simulations in Section 5.4.

It is typical in the literature on classifier performance assessment to review the so-called *apparent error* or *training error*. It is always demonstrated that, for the case of a finite sample size, the apparent error is biased downward, or corresponding measures of *goodness* such as the AUC are biased upward, compared to the conservative approach of using independent testers, or even compared to the mean performance over the population. We have noticed that the PAUC can break this pattern for a small range of values of the threshold. We thus formalize this observation with the following theorem.

Theorem **5.7** (Crossover of apparent and true PAUC). It is not necessary that the apparent PAUC, as an estimator of the true PAUC, be upward biased. In particular, it can be downward biased if the apparent distribution of the log-likelihood ratio (i.e., when the classifier is trained and tested on the same finite samples) is a location-scaled version of the true distribution.

Proof. Assume that the true distribution of the log-likelihood ratio, i.e., the distribution when the trained classifier is tested on the entire population of testers, is given by f_X and f_Y, for the left-hand and right-hand distributions respectively. Then under the assumption of the theorem, the apparent distribution will be given by: $f_X\left((x-\mu_1)/\sigma\right)$ and $f_Y\left((y-\mu_2)/\sigma\right)$ respectively, with $\mu_1 < 0$ and $\mu_2 > 0$ to satisfy the expectation that training and testing on the same observations will generate more class separability than will be found in the population when trainers and testers are independently sampled. Figure 5.2 displays the simplicity of the following equations. The PAUC as given by (5.3) can then be decomposed to:

$$PAUC_{\mathbf{t}}(th_c) = I_1(th_c) + I_2(th_c), \tag{5.13}$$

$$I_1(th_c) = \int_{th_c}^{(th_c-\mu_1)/\sigma} \int_x^{\infty} f_{XY}(x,y)\,dx\,dy$$

$$I_2(th_c) = \int_{(th_c-\mu_1)/\sigma}^{\infty} \int_x^{\infty} f_{XY}(x,y)\,dx\,dy$$

The apparent PAUC, $\overline{PAUC}(th_c)$, at the same threshold th_c is given by:

$$\overline{PAUC}(th_c) = \int_{x'=th_c}^{\infty} \int_{y'=x'}^{\infty} f_X\left(\frac{x'-\mu_1}{\sigma}\right) f_Y\left(\frac{y'-\mu_2}{\sigma}\right) dy'dx', \tag{5.14}$$

which, with a change of variables, can be decomposed to:

$$\overline{PAUC}(th_c) = I_2(th_c) + I_3(th_c), \tag{5.15}$$

$$I_3(th_c) = \int_{(th_c-\mu_1)/\sigma}^{\infty} \int_{x-(\mu_2-\mu_1)/\sigma}^{x} f_{XY}(x,y)\,dx\,dy$$

It is clear that $I_1(-\infty) = 0$, while $I_3(-\infty) > 0$. This shows that the apparent AUC, i.e., $\overline{PAUC}(-\infty)$, is upward biased w.r.t. the AUC. Now, since $I_1(\infty) = I_1(-\infty) = 0$, and I_1 is a continuous function in (th_c) since f_{XY} is positive and defined everywhere in the X–Y subspace, this assures that I_1 exhibits a maximum at some threshold value. However, I_3 is a monotonically nonincreasing function in th_c, and moreover $I_3(\infty) = 0$. Since we can always artificially construct a distribution where the probability measure given by I_1 at some value of th_c is larger than the one given by I_3, then at such a threshold $\overline{PAUC}(th_c) < PAUC(th_c)$ and $\overline{PAUC}(th_c)$ is downward biased with respect to $PAUC_{\mathbf{t}}$. ∎

Remark **5.1**. It seems remarkable that the apparent performance at some values of a metric parameter, i.e., th_c, would be worse than the true performance. This runs against the conventional wisdom in the field that the apparent error rate, as a measure of "badness", i.e., shortcoming, is downward biased w.r.t. the true error, i.e., optimistically biased; and the apparent AUC, a measure of "goodness" is upward biased w.r.t. the true AUC, i.e., optimistically biased as well. In the new metric, i.e., the $PAUC_{\mathbf{t}}(th_c)$, the apparent performance is optimistically biased if the threshold value is lower than a particular crossover value $th_{c.o.}$; after this value the apparent performance may be pessimistically biased. This theorem is illustrated in section 5.4 by simulation results.

The location-scale assumption in theorems 5.5 and 5.7 is not very vulnerable to criticism, at least under the nonparametric assumption, i.e., when no parametric form is known for the distributions. It can be considered a first-order analysis where the available information for the distributions is simply that they will change their location and scale under variation of the training data sets (Theorem 5.5), or under testing on the same training data set (Theorem 5.7). The virtue of Theorem 5.5 is to raise the attention of the user when the PAUC is used as a performance metric. That is, it is possible to choose threshold values where the true performance of the classifier will be very variable. Theorem 5.7 breaks the expected pattern that the apparent performance, where testing is on the same sample from which learning has been developed, is better than the true performance, where testing is carried out on an independent population.

5.3.2. Estimation of Uncertainty in the Estimated Mean Performance

The estimator (5.12) is designed to estimate, from a single available data set, the mean performance of the classifier over the population of training sets (keeping the same set size), i.e., $E_{\mathbf{t}}(PAUC_{\mathbf{t}})$. This estimator will have a variance over the population of training data sets, which can be observed in Monte Carlo trials. This observable variance can also be estimated from a single available data set using a variational approach based on the influence function (introduced in Section 2.1.4 and utilized in Chapter 4). The previous chapter is thus straightforward to extend to the present problem by simply modifying the kernel used for the AUC to the kernel used for the PAUC. The result is an estimate of the variance of the estimated mean performance directly from the original bootstrap samples

5.4. Results With Simulated and Real Data Sets

In this section we show the simulation results that compare, in mean and variance, the true performance, $PAUC_{\mathbf{t}}(th_c)$, the apparent performance, $\overline{PAUC}(th_c)$, and the bootstrap-based estimator $\widehat{\overline{PAUC}}^{(1,1)}(th_c)$, versus th_c. Estimation of the standard error of the estimator from a single available data set using the influence function will be compared to the true standard error obtained from Monte Carlo (MC) trials. Both synthetic data and real data sets will be used in these simulations.

For the present work on the PAUC problem we continue with both, the same simulation parameters and the same real data set of the previous chapters, where the metric was the AUC, i.e., $th_c = -\infty$. MC simulation, typically with 1000 trials, was carried out to closely approximate the true mean and variance of $PAUC_{\mathbf{t}}$, $\widehat{\overline{PAUC}}^{(1,1)}$, and $\overline{PAUC}$. In each trial n observations are sampled from the underlying distributions and over the 1000 trials the true variability is calculated. The two classifiers used in this simulation study are the linear and quadratic discriminant classifiers, LDA and QDA, respectively.

Figures 5.3–5.5 are plots for the mean of $PAUC_{\mathbf{t}}$, $\widehat{\overline{PAUC}}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c; they all illustrate the monotonicity of $PAUC(th_c)$ (5.2.2). The estimator $\widehat{\overline{PAUC}}^{(1,1)}$ shows an observable bias in the figures. This due to the fact mentioned before that—See Section 3.2.2—the effective size of the bootstrap-based training sets is .632 of the original set size. These figures also illustrate the crossover phenomenon explained in Theorem 5.7. This only takes place in a small region of the parameter space.

Figures 5.6–5.8 are plots of the standard error of the same three measures; they illustrate how the variance may increase with the threshold while the $PAUC_{\mathbf{t}}(th_c)$ decreases (Corollary 5.6). These figures also demonstrate the smoothing effect of the bootstrap sampling on the peaking of the variance; see section 5.3.1. Moreover, in some cases the standard error appears as a monotonically decreasing function of the threshold; e.g., see Figure 5.8.

We note that in practice the region where the variance experiences the peaking phenomenon does not correspond to a typical region of interest for the threshold setting. For example, in the screening environment where the prevalence of the target condition is low, one is usually interested in the region of low false-positive fractions, corresponding to somewhat higher threshold regions. For the diagnostic environment where the prevalence of the target condition is high, one solves the problem that is the mirror image to the one solved here; i.e., the inequalities and order of parameters are reversed. In that case, one will also find that the most relevant threshold setting is outside of the peaking region. However, these statements may not necessarily correspond to all situations of interest to practitioners, so users are cautioned to keep Figures 5.6–5.8 in mind.

The nonparametric estimation, using the influence function, for the standard error of $\widehat{\overline{PAUC}}^{(1,1)}$ is illustrated in Table 5.1 for a range of threshold values for every experiment, described above and displayed in the figures. The columns labeled $E_{MC}(\widehat{\overline{PAUC}}^{(1,1)})$ and $sd_{MC}(\widehat{\overline{PAUC}}^{(1,1)})$ show the true mean and true standard error of the estimator, measured over 1000 MC trials. The column $E_{MC}(\widehat{sd})$ gives the mean of $\widehat{sd}(\widehat{\overline{PAUC}}^{(1,1)})$, which is the estimation of $sd(\widehat{\overline{PAUC}}^{(1,1)})$ from one available data set, over 100 MC trials. The last column is the standard deviation of this estimation from the 100 MC trials. Note that the mean estimates of the standard error are all within one standard deviation of the MC population results.

5.5. Chapter Summary

A natural generalization of the area under the ROC curve has been introduced to assess classifiers, i.e., the separability function or the PAUC. The metric is essential for assessing classifiers that are used in an environment whose threshold falls within an a priori known limited range. Some caution should be taken when choosing the threshold value, since at some values the performance is intrinsically (i.e., aside from issues of estimation) very variable under the variability of the training data set. Some mathematical properties of the new metric have been stated and proven.

An estimator was proposed, the leave-pair-out estimator, to estimate the mean of the PAUC. The estimator can be downwards (or upward) biased as was the case with its predecessors which estimated the mean AUC (or mean error). The influence function approach was used to estimate the uncertainty of that estimator from the available set. This uncertainty reflects the finite size of the sample available to train as well as to test the performance of the classifier.

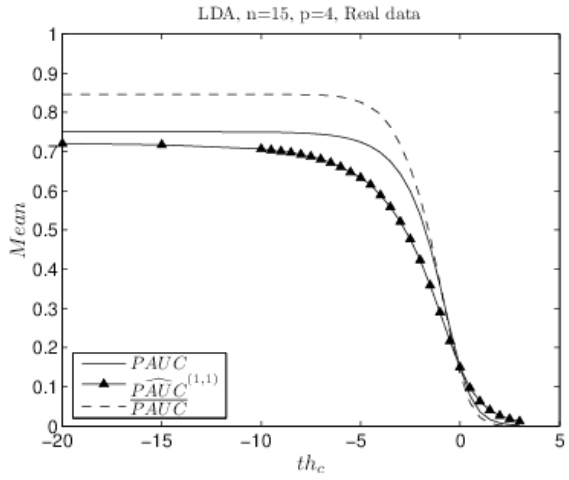

Figure 5.3. Mean of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using real data set with LDA, $n = 15$, and $p = 4$.

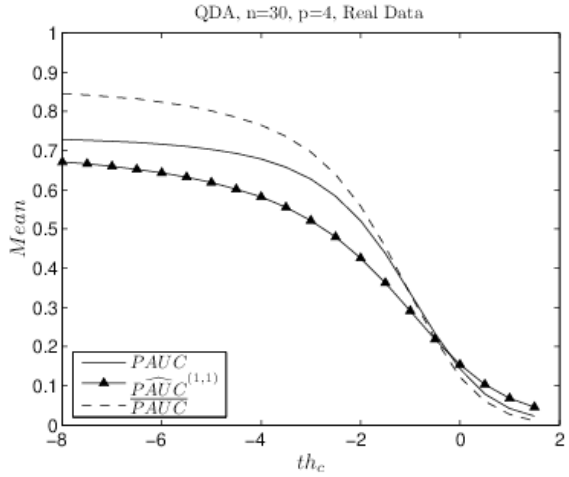

Figure 5.4. Mean of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using real data set with QDA, $n = 30$, and $p = 4$.

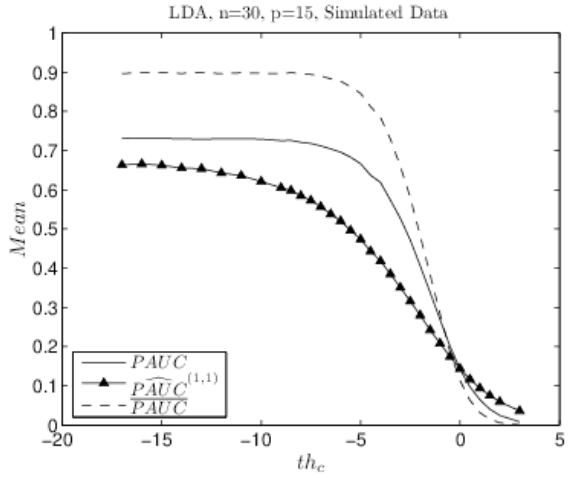

Figure 5.5. Mean of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using multinormal-simulated data set with LDA, $n = 30$, and $p = 15$.

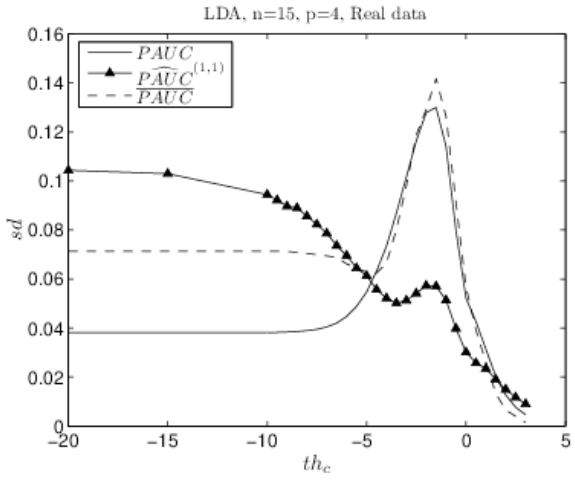

Figure 5.6. Standard error of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using real data set with LDA, $n = 15$, and $p = 4$.

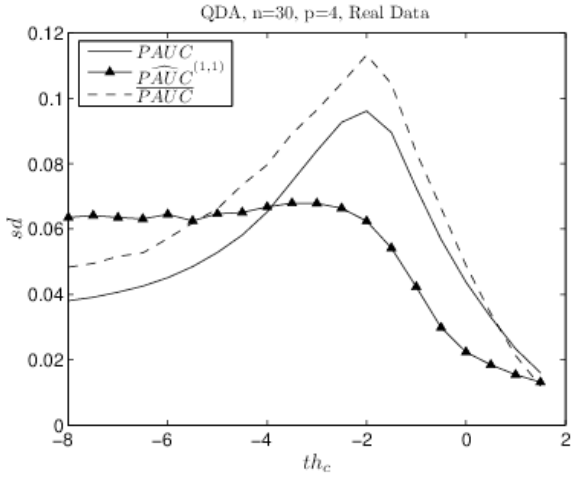

Figure 5.7. Standard error of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using real data set with QDA, $n = 30$, and $p = 4$.

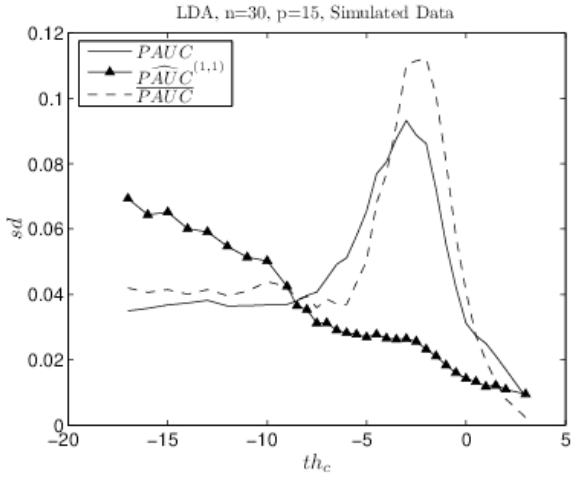

Figure 5.8. Standard error of $PAUC$, $\widehat{PAUC}^{(1,1)}$, and $\overline{PAUC}$ vs. th_c using multinormal-simulated data set with LDA, $n = 30$, and $p = 15$.

Classifier	th_c	$MC = 1000$		$MC = 100$	
		$E_{MC}(\widehat{PAUC}^{(1,1)})$	$sd_{MC}(\widehat{PAUC}^{(1,1)})$	$E_{MC}(\widehat{sd})$	$sd_{MC}(\widehat{sd})$
LDA	-4	.5884	.05218	.0731	.0295
$P = 4$	-2	.4232	.05733	.0621	.0214
$N = 15$	-1	.2898	.05135	.0550	.0145
Real Data	0	.1499	.03012	.0374	.0077
	1	.06278	.02346	.0229	.0060
QDA	-4	.5818	.06674	.0709	.0111
$P = 4$	-2	.4256	.06242	.0623	.0119
$N = 30$	-1	.2911	.04228	.0475	.0076
Real Data	0	.1541	.02234	.0288	.0051
	1	.06815	.01541	.0169	.0033
	-4	.4181	.02655	.0347	.0077
LDA	-2	.2798	.02315	.0294	.0039
$p = 15$	-1	.2083	.01829	.0252	.0037
$n = 30$	0	.1440	.0142	.0208	.0030
	1	.09407	.01175	.0163	.0024

Table 5.1. Different experiments and different n-p-th_c combinations. for the estimator $PAUC^{(1,1)}$ measured over 1000 MC trial. The last two columns are the mean and standard error for the influence function estimator measured over 100 MC trials.

CHAPTER 6

Assessing Classifiers From Two Independent Data Sets Using ROC Analysis: a Nonparametric Approach

6.1. Introduction

From the definition of the PAUC in Chapter 5 the AUC, as a special case of the PAUC, can be written as:

$$AUC_{\mathbf{tr}} = \Pr(\widehat{h}_{\mathbf{tr}}(X|\omega_2) < \widehat{h}_{\mathbf{tr}}(X|\omega_1)) \tag{6.1}$$

We will focus here on the AUC as the metric for the assessment of classifier performance. It is straightforward to extend this treatment to other summary measures of performance such as the PAUC. In the present chapter the training and testing sets will be separate and independent; hence we will denote them by **tr** and **ts** respectively. The fundamental population parameters of this random variable are the following: The true performance $AUC_{\mathbf{tr}}$ conditional on a particular training data set **tr** of a specified size but over the population of testers; the expectation of this performance over the population of training data sets $\mathrm{E}_{\mathbf{tr}}\, AUC_{\mathbf{tr}}$; and the measure of variability of this performance over the population of training data sets, namely, $\mathrm{Var}_{\mathbf{tr}}\, AUC_{\mathbf{tr}}$. Estimators of these parameters, respectively, $\widehat{AUC_{\mathbf{tr}}}$, $\mathrm{E}_{\mathbf{tr}}\widehat{AUC}_{\mathbf{tr}}$, and $\mathrm{Var}_{\mathbf{tr}}\widehat{AUC}_{\mathbf{tr}}$, can be obtained in several ways. Parametric estimates can be obtained by modeling the underlying distributions of the samples, e.g., as in Fukunaga (1990).

The present work addresses the case where the distributions of the samples are either unknown or not readily modeled; that is, we address the problem of nonparametric estimators of these population parameters. There are several traditional approaches to using the available data in this estimation task. One approach is to have a common data set that is used for training and testing; this approach often includes various resampling strategies, including cross-validation and bootstrapping. This approach is what has been considered in this dissertation up to this point; the first two of these estimators, $\widehat{AUC_{\mathbf{tr}}}$ and $\mathrm{E}_{\mathbf{tr}}\widehat{AUC}_{\mathbf{tr}}$, were discussed, along with their variances, in previous chapters.

Another approach is to maintain what might be called the traditional *data hygiene* of two independent data sets, one for training and one for testing. There are some situations, e.g., in several public-policy-making or regulatory settings, in which it could be highly recommended, or even mandatory. In the present chapter we analyze the problem in this context. The approach will be completely nonparametric. We will derive closed-form expressions for the three estimators listed above, and also for $\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\,\widehat{AUC_{\mathbf{tr}}}$ (the variance, over the trainers and testers, of the estimator $\widehat{AUC_{\mathbf{tr}}}$). Note that $AUC_{\mathbf{tr}}$ is a population parameter conditional on a particular training set, which becomes a random variable when a population of training sets is considered; however, $\widehat{AUC_{\mathbf{tr}}}$ is an estimate of this parameter whose randomness comes from both the finite training set **tr** and the finite testing set **ts**. All of the proposed estimators are functions only in these two data sets.

In Section 6.2 we give a brief account of the theory of U-statistics that will establish the framework of the estimators discussed in the present chapter. In Section 6.3 we derive the U-statistic-based estimators of the population parameters discussed above. In Section 6.4 we provide examples and results of simulations of estimators and their properties over some specified populations.

6.2. Nonparametric Point Estimation

In nonparametric estimation we assume no knowledge about the distribution of the available data. Any population parameter must then be estimated from the information in the available data without further assumptions. U-statistics are, by construction, a class of nonparametric unbiased estimators. All of the estimators, of the means and variances, discussed above are natural candidates for this approach. We will therefore provide in the present section the fundamental definitions and concepts underlying the U-statistic theory, which will be fully utilized in Section 6.3. We follow the terminology used in Randles and Wolfe (1979).

Definition **6.1**. For any distribution $F \in \mathcal{F}$, where $\mathcal{F}$ is a family of distributions, a population parameter γ is said to be estimable of degree r if r is the smallest sample size for which $\exists$ a function $k^*(a_1,\ldots,a_r) \ni$

$$\mathrm{E}_F\left[k^*\left(A_1,\ldots,A_r\right)\right] = \gamma \tag{6.2}$$

The function k^* is called the U-statistic kernel of the parameter γ. We can always create a symmetric function k from that kernel by averaging over the permutations:

$$k\left(a_1,\ldots,a_r\right) = \frac{1}{r!}\sum_{\alpha\in\mathcal{A}} k^*\left(a_{\alpha_1},\ldots,a_{\alpha_r}\right), \tag{6.3}$$

where $\mathcal{A} = \{\alpha : \alpha \text{ is a permutation of the integers } 1,\dots,r\}$.

Definition **6.2**. The one-sample U-statistic for the estimable population parameter γ, with degree r, is constructed from a sample with size n as:

$$U(a_1,\dots,a_n) = \frac{1}{\binom{m}{r}} \sum_{\xi\in\mathcal{S}} k(a_{\xi_1},\dots,a_{\xi_r}), \tag{6.4}$$

where $\mathcal{S} = \{\xi : \xi \text{ is one of the } \binom{n}{r} \text{ unordered subsets of } r \text{ integers chosen without replacement from the set } \{1,\dots,n\}\}$.

If the population parameter of interest is a function of two distributions then the U-statistic is called a two-sample statistic (the concept can be generalized to any number of distributions). The utility of symmetrization is to reduce the variance of the U-statistic, as Lemma 6.7 below states. Aside from any rigorous proof, this symmetrization utilizes all the available information in a set of r observations and thus reduces the variance of the estimator. Consider, e.g., the kernel $x_1^2 - x_1x_2$, where $r = 2$, whose expectation is $\operatorname{Var} X$. We have not yet utilized all the available information in the two observations x_1 and x_2 until we consider the other permutation $x_2^2 - x_2x_1$.

Lemma **6.3**. The variance of any one-sample U-statistic is given by (see Randles and Wolfe, 1979, Sec. 3.1):

$$\operatorname{Var} U(A_1,\dots,A_m) = \frac{1}{\binom{m}{r}} \sum_{c=1}^{r} \binom{r}{c}\binom{m-r}{r-c}\xi_c, \tag{6.5}$$

where

$$\begin{aligned}\xi_c &= \operatorname{Cov}\left[k(A_1,\dots,A_c,A_{c+1},\dots,A_r), k(A_1,\dots,A_c,A_{r+1},\dots,A_{2r-c})\right] \\ &= \mathrm{E}\left[k(A_1,\dots,A_c,A_{c+1},\dots,A_r)k(A_1,\dots,A_c,A_{r+1},\dots,A_{2r-c})\right] - \gamma^2\end{aligned} \tag{6.6}$$

Proof. From (6.4) we can write the variance (6.5) as:

$$\begin{aligned}\operatorname{Var} U(A_1,\dots,A_n) &= \mathrm{E}\left[\left\{\frac{1}{\binom{m}{r}}\sum_{\xi\in\mathcal{S}}\left[k(A_{\xi_1},\dots,A_{\xi_r}) - \gamma\right]\right\}^2\right] \\ &= \frac{1}{\binom{m}{r}^2}\sum_{\xi\in\mathcal{S}}\sum_{\xi'\in\mathcal{S}} \mathrm{E}\left[\left\{k(A_{\xi_1},\dots,A_{\xi_r}) - \gamma\right\}\left\{k\left(A_{\xi'_1},\dots,A_{\xi'_r}\right) - \gamma\right\}\right] \\ &= \frac{1}{\binom{m}{r}^2}\sum_{\xi\in\mathcal{S}}\sum_{\xi'\in\mathcal{S}} \operatorname{Cov}\left[k(A_{\xi_1},\dots,A_{\xi_r}), k(A_{\xi'_1},\dots,A_{\xi'_r})\right]\end{aligned} \tag{6.7}$$

Since the kernel k is symmetric any covariance term in (6.7), having exactly c common observations, will be ξ_c. There are $\binom{m}{r}$ ways to split m to two groups, r and $m-r$. If we want both kernels, in a covariance pair, to have c observations in common and $r-c$ observations not in common, then we have to choose $r-c$ from the first group that contain r observations—yielding the binomial coefficient $\binom{r}{c}$—and $r-c$ from the second group containing $m-r$ observations—yielding the binomial coefficient $\binom{m-r}{r-c}$. Then the total number of permutations for building the two kernels with common c observations is $\binom{m}{r}\binom{r}{c}\binom{m-r}{r-c}$. Thus, (6.7) can be rewritten as:

$$\begin{aligned}\operatorname{Var} U(A_1,\dots,A_n) &= \frac{1}{\binom{m}{r}^2}\sum_{\xi\in\mathcal{S}}\sum_{\xi'\in\mathcal{S}}\binom{m}{r}\binom{r}{c}\binom{m-r}{r-c}\xi_c \\ &= \frac{1}{\binom{m}{r}}\sum_{c=1}^{r}\binom{r}{c}\binom{m-r}{r-c}\xi_c,\end{aligned} \tag{6.8}$$

and we start the summation from 1, not from 0, since the ξ_0, the covariance between two independent random variables, is equal to 0. ∎

Definition **6.4**. For any two distributions F and G in the family $\mathcal{F}$, a parameter γ is said to be estimable of degree (r,s), respectively, if $\exists$ a function $k^*(a_1,\dots,a_r;\, b_1,\dots,b_s) \ni$

$$\mathrm{E}_{F,G}\left[k^*(A_1,\dots,A_r;\, B_1,\dots,B_s)\right] = \gamma \tag{6.9}$$

Again, by averaging over permutations, we can make the kernel k^* symmetric as done above by defining:

$$k(a_1,\dots,a_r;\, b_1,\dots,b_s) = \frac{1}{r!s!}\sum_{\alpha\in\mathcal{A}}\sum_{\beta\in\mathcal{B}} k^*(a_{\alpha_1},\dots,a_{\alpha_r};\, b_{\beta_1},\dots,b_{\beta_s}), \tag{6.10}$$

where $\mathcal{A} = \{\alpha : \alpha \text{ is a permutation of the integers } 1,\dots,r\}$ and $\mathcal{B} = \{\beta : \beta \text{ is a permutation of the integers } 1,\dots,s\}$.

Definition **6.5**. The two-sample U-statistic for the estimable population parameter γ, with degree (r,s), is constructed from two samples of sizes m and n respectively by:

$$U(a_1,\dots,a_m;\, b_1,\dots,b_n) = \frac{1}{\binom{m}{r}\binom{n}{s}}\sum_{\xi\in\mathcal{S}_A}\sum_{\zeta\in\mathcal{S}_B} k(a_{\xi_1},\dots,a_{\xi_r};\, b_{\zeta_1},\dots,b_{\zeta_s}), \tag{6.11}$$

where $\mathcal{S}_A$ and $\mathcal{S}_B$ are collections of all subsets of r integers chosen from m integers, and s integers chosen from n integers, without replacement.

Lemma **6.6**. The variance of any two-sample U-statistic is given by (see Randles and Wolfe, 1979, Sec. 3.4):

$$\mathrm{Var}\,U\left(a_1,\ldots,a_m;\ b_1,\ldots,b_n\right)=\frac{1}{\binom{m}{r}\binom{n}{s}}\sum_{d=0}^{s}\sum_{c=0}^{r}\binom{r}{c}\binom{m-r}{r-c}\binom{s}{d}\binom{n-s}{s-d}\xi_{c,d}, \tag{6.12}$$

where

$$\begin{aligned}\xi_{c,d}=\mathrm{Cov}\big[&k(A_1,\ldots,A_c,A_{c+1},\ldots,A_r;B_1,\ldots,B_d,B_{d+1},\ldots,B_s),\\ &k(A_1,\ldots,A_c,A_{r+1},\ldots,A_{2r-c};B_1,\ldots,B_d,B_{s+1},\ldots,B_{2s-d})\big]\end{aligned} \tag{6.13}$$

i.e., the covariance between the two symmetric kernels which have the first c and d observations, respectively, common from the two samples.

Proof. By splitting the first sample to r and $m-r$ and the second to s and $n-s$, then proceeding as above when proving Lemma 6.3, the proof is immediate. It is clear that $\xi_{0,0}=0$ since it is a covariance between two independent kernels. ∎

Lemma **6.7**. If $\mathcal{F}$ includes all continuous distributions, then the U-statistic is the Unique Minimum Variance Unbiased Estimator (UMVUE); see Randles and Wolfe (1979, Exercises 3.1.10 and 3.1.11).

Proof. Without loss of any generality we will speak in terms of the one-sample U-statistic. By sorting the n observations $a_1,\ldots,a_n$ such that we have $a_{(1)}<\cdots<a_{(n)}$, then $a_{(j)}$ is called the j'th order statistic. It is not hard to see that any estimator $\delta(A_1,\ldots,A_n)$ is a function of the order statistics if and only if it is symmetric in its arguments. Since the U-statistic is symmetric, by construction, hence it is a function of the order statistics $A_{(1)},\ldots,A_{(n)}$. Since the order statistics are complete and sufficient for the nonparametric distribution family $\mathcal{F}$—see Lehmann and Romano (2005, Ex. 2.4.1)—then the U-statistic estimator is a function of a complete and sufficient statistic. This property is what makes it the UMVUE. The latter follows from the Lehmann-Scheffé theorem; a good discussion of the concept of UMVUE is provided in Lehmann and Casella (1998, Ch. 2). A full account of complete and sufficient statistics is given in Casella and Berger (2002, Ch. 6) or Lehmann and Casella (1998, Sec. 1.6). ∎

Lemma **6.8**. if U_1 and U_2 are U-statistic estimators for γ_1 and γ_2 respectively and the degree(s) of $\gamma_1+\gamma_2$ is the maximum of the degree(s) of γ_1 and γ_2, then U_1+U_2 is the U-statistic estimator for $\gamma_1+\gamma_2$ Randles and Wolfe (see 1979, Ex. 3.1.2). This Lemma applies to one- and two-sample U-statistics as well.

6.3. Analyzing the AUC

The present problem is a member of a class of problems that involve several sources of randomness that contribute to the outcome. An important example for the field of diagnostic medicine has been that of assessing medical imaging systems. In that problem two major sources of randomness are the variability of patient cases and the variability of the radiologists who read their images. A large literature has evolved to address this problem and a unifying framework based on linear components-of-variance models has been provided in Roe and Metz (1997*a*) and Roe and Metz (1997*b*). The authors in Beiden, Maloof and Wagner (2003) discussed the correspondence between the random effects of readers and cases in imaging and the random effects of finite training sets and finite test sets in the field of statistical pattern recognition and showed how some methods in the former field apply to the latter. The authors of Barrett, Kupinski and Clarkson (2005) questioned the approach of basing these solutions on the linear statistical models and provided a formulation based on general principles of multivariate probability theory. Practical implementations of the literature cited here have depended on methods of statistical resampling. In Gallas (2006) it has been shown that the approach to the medical imaging problem addressed in Barrett, Kupinski and Clarkson (2005) can be implemented without statistical resampling. The authors in DeLong, DeLong and Clarke-Pearson (1988) started analyzing the problem from the U-statistic theoretic approach, yet they relied on the resampling techniques in deriving their estimators. The present work addresses the problem of assessing classifiers in the field of pattern recognition, and our point of departure is to provide a rigorous analysis under the theory of nonparametric estimation.

6.3.1. Variance Decomposition

We begin by analyzing the true conditional AUC, Eq. (6.1), and its estimator in mean and variance. We consider the setting where we are required to maintain two independent data sets, one for training and one for testing. In the previous chapters this restriction was not imposed. We denote the training data set $\mathbf{tr}=\mathbf{tr}_1\cup\mathbf{tr}_2$, where $\mathbf{tr}_c=\{t_i: t_i=(x_i,y_i),\ i=1,\ldots,n_{\mathbf{tr}_c},\ y_i=\omega_c,\ c=1,2\}$, and a testing data set $\mathbf{ts}=\mathbf{ts}_1\cup\mathbf{ts}_2$, where $\mathbf{ts}_c=\{t_i: t_i=(x_i,y_i),\ i=1,\ldots,n_{\mathbf{ts}_c},\ y_i=\omega_c,\ c=1,2\}$. A classifier trained on the training set $\mathbf{tr}$ and tested on the testing set $\mathbf{ts}$ estimates the true AUC $AUC_{\mathbf{tr}}$ by $\widehat{AUC}_{\mathbf{tr}}$. The latter is a function of $\mathbf{tr}$ and $\mathbf{ts}$, while the former is only a function of $\mathbf{tr}$. For simplicity of notation we will refer to $AUC_{\mathbf{tr}}$ as γ and to $\widehat{AUC}_{\mathbf{tr}}$ as $\widehat{\gamma}$, where $\widehat{\gamma}=\widehat{\gamma}(\mathbf{tr},\mathbf{ts})$. The two sets $\mathbf{tr}$ and $\mathbf{ts}$ comprise, respectively, $n_{\mathbf{tr}}$ and $n_{\mathbf{ts}}$ observations each of which is p-dimensional vector; hence they are $(p\times n_{\mathbf{tr}})$- and $(p\times n_{\mathbf{ts}})$-dimensional vectors. Consequently, $\widehat{\gamma}(\mathbf{tr},\mathbf{ts})$ is a function of two random variables and its variance can be

decomposed to (see Casella and Berger, 2002, Sec. 4.4):

$$\begin{aligned}\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\,\widehat{\gamma} &= \mathrm{E}_{\mathbf{tr},\mathbf{ts}}[\widehat{\gamma}^2] - \left(\mathrm{E}_{\mathbf{tr},\mathbf{ts}}[\widehat{\gamma}]\right)^2 \\ &= \mathrm{E}_{\mathbf{tr}}[\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma}] + \mathrm{Var}_{\mathbf{tr}}[\mathrm{E}_{\mathbf{ts}}\,\widehat{\gamma}],\end{aligned} \tag{6.14}$$

where the subscripts **tr** and **ts** indicate over which random variable the expectation and the variance are taken. E.g., $\mathrm{E}_{\mathbf{ts}}\,\widehat{\gamma}$ will be a random variable whose randomness comes only from **tr**. In Section 6.5 we write (6.14) in a different form to comment on some conventional wisdom in the field.

Eq. (6.14) provides the decomposition of the variance of an estimator $\widehat{\gamma}$. We require first the estimator $\widehat{\gamma}$ itself. We simplify the notation by letting $a_i = \widehat{h}_{\mathbf{tr}}(x_i|\omega_1)$ represent an observation of the random variable A; in medical diagnostics this represents the abnormal class, i.e., the class with the higher average log-likelihood values. Also let $b_j = \widehat{h}_{\mathbf{tr}}(x_j|\omega_2)$ represent an observation from the random variable B; in medical diagnostics this represents the normal class, i.e., the class with the lower average log-likelihood values. The subscript **tr** indicates that the log-likelihood ratio $\widehat{h}$ is obtained from a particular training data set **tr**. The U-statistic estimator for γ is obtained by defining the kernel

$$k_1^*(a_i;b_j) = \begin{cases} 1 & b_j < a_i \\ 0 & otherwise \end{cases} \tag{6.15}$$

It is clear that expectation of k_1^* with respect to (w.r.t.) the set of b_j's and a_i's is the AUC (6.1). It is a two-sample kernel with degrees $r = s = 1$. Since this kernel is already symmetric, i.e., $k_1 = k_1^*$ from (6.10), the U-statistic estimator $\widehat{\gamma}$ is immediately:

$$\widehat{\gamma} = \frac{1}{n_{ts_1} n_{ts_2}} \sum_{j=1}^{n_{ts_2}} \sum_{i=1}^{n_{ts_1}} k_1(a_i;b_j), \tag{6.16}$$

which is the well-known Mann-Whitney statistic, a version of the Wilcoxon statistic, the UMVUE for the population parameter $\gamma = \Pr(B < A)$. The random variables A and B are distributed as $\widehat{h}_{\mathbf{tr}}(X|\omega_1)$ and $\widehat{h}_{\mathbf{tr}}(X|\omega_2)$ respectively.

Lemma **6.9** ($\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma}$). The variance of $\widehat{\gamma}$ over the population of testers **ts**, conditional on a particular training set **tr**, is given by:

$$\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma} = \frac{1}{n_{ts_1} n_{ts_2}} \left[(n_{ts_1} - 1)p_{12} + (n_{ts_2} - 1)p_{21} + \gamma - (n_{ts} - 1)\gamma^2\right], \tag{6.17}$$

where

$$p_{12} = \Pr\left[B < \min(A_1, A_2)\right], \tag{6.18a}$$

$$p_{21} = \Pr\left[\max(B_1, B_2) < A\right], \tag{6.18b}$$

where all A_1, A_2, and A are i.i.d. and so are B_1, B_2 and B.

Proof. Since $r = s = 1$, then it follows directly from (6.12) that

$$\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma} = \frac{1}{n_{ts_1} n_{ts_2}} [(n_{ts_1} - 1)\xi_{0,1} + (n_{ts_2} - 1)\xi_{1,0} + \xi_{1,1}] \tag{6.19}$$

Direct application to (6.13) shows that

$$\begin{aligned}\xi_{0,1} &= \mathrm{E}[k_1(a_1;b_1)k_1(a_2;b_1)] - \mathrm{E}[k_1(a_1;b_1)]\,\mathrm{E}[k_1(a_2;b_1)] \\ &= \Pr\left[B < \min(A_1, A_2)\right] - \gamma^2,\end{aligned} \tag{6.20a}$$

$$\begin{aligned}\xi_{1,0} &= \mathrm{E}[k_1(a_1;b_1)k_1(a_1;b_2)] - \mathrm{E}[k_1(a_1;b_1)]\,\mathrm{E}[k_1(a_1;b_2)] \\ &= \Pr\left[\max(B_1, B_2) < A\right] - \gamma^2,\end{aligned} \tag{6.20b}$$

$$\begin{aligned}\xi_{1,1} &= \mathrm{E}[k_1(a_1;b_1)k_1(a_1;b_1)] - \mathrm{E}[k_1(a_1;b_1)]\,\mathrm{E}[k_1(a_1;b_1)] \\ &= \gamma - \gamma^2,\end{aligned} \tag{6.20c}$$

where $k_1(a_1;b_1)^2 = k_1(a_1;b_1)$ is used (check the definition (6.15)). By substituting back into (6.19), (6.17) follows directly. ∎

An analogue of (6.17) was previously given without derivation in Campbell, Douglas and Bailey (1988) for the case of a simple diagnostic test. In the present chapter we essentially investigated the variance of $\widehat{\gamma}$ when the only random effect was from the finite testing set, i.e., $\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma}$. In the present chapter, this variance represents the bracketed quantity in the first term in (6.14).

Theorem **6.10** ($\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\,\widehat{\gamma}$). The variance of the Mann-Whitney estimator is given by:

$$\begin{aligned}\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\,\widehat{\gamma} = {} & \frac{1}{n_{ts_1} n_{ts_2}} \mathrm{E}_{\mathbf{tr}}\left[\gamma + \gamma^2 - p_{12} - p_{21}\right] + \frac{1}{n_{ts_2}} \mathrm{E}_{\mathbf{tr}}\left[p_{12} - \gamma^2\right] + \frac{1}{n_{ts_1}} \mathrm{E}_{\mathbf{tr}}\left[p_{21} - \gamma^2\right] \\ & + \mathrm{E}_{\mathbf{tr}}\,\gamma^2 - (\mathrm{E}_{\mathbf{tr}}\,\gamma)^2\end{aligned} \tag{6.21}$$

Proof. Since $\mathrm{E}_{\mathbf{ts}}\widehat{\gamma} = \gamma$, from the unbiasedness of the U-statistic, then

$$\begin{aligned}\mathrm{Var}_{\mathbf{tr}}[\mathrm{E}_{\mathbf{ts}}\widehat{\gamma}] &= \mathrm{Var}_{\mathbf{tr}}\,\gamma \\ &= \mathrm{E}_{\mathbf{tr}}\,\gamma^2 - (\mathrm{E}_{\mathbf{tr}}\,\gamma)^2\end{aligned} \tag{6.22}$$

That is, the variance of the true conditional AUC is a component of the variance of the estimator that estimates that AUC. The proof is completed by combining this equation with the result from Lemma 6.9, then substituting back into (6.14) and combining the common terms for n_{ts_1} and n_{ts_2}. ∎

Before moving on to the nonparametric estimation, we have to comment on the last theorem. A classifier trained on the training set **tr** has the true conditional performance AUC, i.e., γ. The classifier has the mean performance $\mathrm{E}_{\mathbf{tr}}\,\gamma$, which is not a random variable any more; the classifier also has the variance $\mathrm{Var}_{\mathbf{tr}}\,\gamma$ over the population of training sets. These very important parameters are parts of the variance of the statistic $\widehat{\gamma}$ that estimates, from only one training and one testing set, the true conditional performance γ. Estimating these parameters, from the same training and testing sets, is not only desirable as a means for estimating the variance of a statistic, i.e., $\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\,\widehat{\gamma}$, but it is valuable for its own sake, since they are performance parameters of the classifier itself.

6.3.2. Nonparametric Estimation

We start by estimating p_{12} and p_{21} in (6.18). Define the kernel

$$k_2^*(a_i, a_{i'}; b_j) = \begin{cases} 1 & b_j < \min(a_i, a_{i'}) \\ 0 & otherwise \end{cases}, \tag{6.23}$$

where $\mathrm{E}_{\mathbf{ts}}\,k_2^* = p_{12}, r = 2$, and $s = 1$; then make it symmetric as explained in (6.10):

$$k_2(a_i, a_{i'}; b_j) = \frac{1}{2}\left[k_2^*(a_i, a_{i'}; b_j) + k_2^*(a_{i'}, a_i; b_j)\right] \tag{6.24}$$

Then the U-statistic estimator for p_{12} is given by

$$\widehat{p}_{12} = \frac{1}{\binom{n_{ts_1}}{2}\binom{n_{ts_2}}{1}} \sum_{i=1}^{n_{ts_1}} \sum_{i'>i} \sum_{j=1}^{n_{ts_2}} k_2(a_i, a_{i'}; b_j) \tag{6.25}$$

Analogously, p_{21} is estimated by defining the kernel

$$k_3^*(a_i; b_j, b_{j'}) = \begin{cases} 1 & \max(b_j, b_{j'}) < a_i \\ 0 & otherwise \end{cases}, \tag{6.26}$$

where $\mathrm{E}_{\mathbf{ts}}\,k_3^* = p_{12}, r = 1$, and $s = 2$; then make it symmetric by defining

$$k_3(a_i; b_j, b_{j'}) = \frac{1}{2}\left[k_3^*(a_i; b_j, b_{j'}) + k_3^*(a_i; b_{j'}, b_j)\right] \tag{6.27}$$

Then the U-statistic estimator for p_{21} is given by

$$\widehat{p}_{21} = \frac{1}{\binom{n_{ts_1}}{1}\binom{n_{ts_2}}{2}} \sum_{i=1}^{n_{ts_1}} \sum_{j=1}^{n_{ts_2}} \sum_{j'>j} k_3(a_i; b_j, b_{j'}) \tag{6.28}$$

In Campbell, Douglas and Bailey (1988), the authors estimated γ^2 by $\widehat{\gamma}^2$. Although this has the correct asymptotic behavior, it will be biased for finite sample sizes. Rather, γ^2 can be estimated anew by proposing kernel

$$\begin{aligned} k_4^*(a_i, a_{i'}; b_j, b_{j'}) &= \begin{cases} 1 & b_j < a_i \;\&\; b_{j'} < a_{i'} \\ 0 & otherwise \end{cases} \\ &= k_1^*(a_i; b_j) k_1^*(a_{i'}; b_{j'}) \end{aligned} \tag{6.29}$$

Then it is clear that $r = s = 2$ and $\mathrm{E}_{\mathbf{ts}}\,k_4^* = \gamma^2$. The symmetric kernel is obtained as

$$k_4(a_i, a_{i'}; b_j, b_{j'}) = \frac{1}{2}\left[k_4^*(a_i, a_{i'}; b_j, b_{j'}) + k_4^*(a_{i'}, a_i; b_j, b_{j'})\right] \tag{6.30}$$

Then the U-statistic estimator for γ^2 is given by

$$\widehat{\gamma^2} = \frac{1}{\binom{n_{ts_1}}{2}\binom{n_{ts_2}}{2}} \sum_{i=1}^{n_{ts_1}} \sum_{i'>i} \sum_{j=1}^{n_{ts_2}} \sum_{j'>j} k_4(a_i, a_{i'}; b_j, b_{j'}) \tag{6.31}$$

The estimators discussed in this section, thus far, are the estimators of the components of $\mathrm{Var}_{\mathbf{ts}}\,\widehat{\gamma}$ that are required for (6.21). The total expression in that equation for the uncertainty from both the training and testing sets requires that we take the expectation over the population of training sets, $\mathrm{E}_{\mathbf{tr}}$. If we had the availability of multiple training sets, this expectation could be

estimated by averaging over the multiple training sets, i.e.,

$$\widehat{\mathrm{E}_{\mathbf{tr}}(\widehat{\mathrm{E}_{\mathbf{ts}}\hat{s}})} = \frac{1}{B}\sum_{b=1}^{B}\hat{s}(\mathbf{ts},\mathbf{tr}_b), \tag{6.32}$$

where $\mathbf{tr}_b$ is the training set b, where $\hat{s}$ is either $\hat{\gamma}$, $\widehat{\gamma^2}$, $\hat{p}_{12}$, or $\hat{p}_{21}$. That is, from every training data set we train the classifier and obtain an estimate $\hat{s}(\mathbf{ts},\mathbf{tr}_b)$ by testing $\mathbf{ts}$, then take the average over the different training sets. We committed ourselves from the beginning to the restriction of having only one available training and testing sets. In such a situation, the estimator (6.32) can be approximated by averaging over bootstrap replications from the available training set, i.e.,

$$\widehat{\mathrm{E}_{\mathbf{tr}}(\widehat{\mathrm{E}_{\mathbf{ts}}\hat{s}})} \approx \frac{1}{B}\sum_{b=1}^{B}\hat{s}(\mathbf{ts},\mathbf{tr}_b^*), \tag{6.33}$$

where $\mathbf{tr}_b^*$ is the b^{th} bootstrap replication from $\mathbf{tr}$. This estimator, however, can be biased (see Section 3.2.2).

The last component to be estimated is $(\mathrm{E}_{\mathbf{tr}}\gamma)^2$. In contrast to simply proposing the intuitive estimator $\widehat{(\mathrm{E}_{\mathbf{tr}}\gamma)}^2$, which is biased, implementing a U-statistic estimator, following the definitions in Section 6.2, requires defining a kernel over two different training sets; so we should keep the full notation $h_b(x_i|\omega_c)$ rather than a_i or b_j to distinguish a's and b's obtained from the same testing set $\mathbf{ts}$ but from different training sets $\mathbf{tr}_b$ and $\mathbf{tr}_{b'}$. For more simple notation, we drop the class ω_c and let x_i and $x_{i'}$ represent two observations from $\mathbf{ts}_1$, while z_j and $z_{j'}$ represent two observations from $\mathbf{ts}_2$. Then define the kernel

$$k_5^*(x_i,x_{i'};z_j,z_{j'};\mathbf{tr}_b,\mathbf{tr}_{b'}) = k_1^*\left(h_1(x_i);h_1(z_j)\right)k_1^*\left(h_2(x_{i'});h_2(z_{j'})\right) \tag{6.34}$$

This kernel equals one if, for two pairs, the ω_1 observation has a higher decision value than the ω_2 observation for every pair after training on two different training sets. This kernel is a three-sample U-statistic kernel with $r=s=k=2$, where $\mathrm{E}_{\mathbf{tr},\mathbf{ts}}k_5^* = (\mathrm{E}_{\mathbf{tr}}\gamma)^2$. This kernel can be symmetrized by

$$\begin{aligned} k_5(x_i,x_{i'};z_j,z_{j'};\mathbf{tr}_b,\mathbf{tr}_{b'}) = \frac{1}{4}[& \\ & k_1^*\left(h_1(x_i);h_1(z_j)\right)k_1^*\left(h_2(x_{i'});h_2(z_{j'})\right) + k_1^*\left(h_1(x_i);h_1(z_{j'})\right)k_1^*\left(h_2(x_{i'});h_2(z_j)\right) \\ & + k_1^*\left(h_1(x_{i'});h_1(z_j)\right)k_1^*\left(h_2(x_i);h_2(z_{j'})\right) + k_1^*\left(h_1(x_{i'});h_1(z_{j'})\right)k_1^*\left(h_2(x_i);h_2(z_j)\right)] \end{aligned} \tag{6.35}$$

The U-statistic estimator is then given by

$$\widehat{(\mathrm{E}_{\mathbf{tr}}\gamma)^2} = \frac{1}{\binom{n_{ts_1}}{2}\binom{n_{ts_2}}{2}\binom{B}{2}}\sum_{b=1}^{B}\sum_{b'>b}\sum_{i=1}^{n_{ts_1}}\sum_{i'>i}\sum_{j=1}^{n_{ts_2}}\sum_{j'>j}k_5(x_i,x_{i'};z_j,z_{j'};\mathbf{tr}_b,\mathbf{tr}_{b'}) \tag{6.36}$$

Once again, the summation over the different training sets B can be approximated by summation over bootstrap replications as in (6.33).

Now we have estimated all the components of (6.21). Every component is a U-statistic estimator with some degree. It is easy to see that the overall degree of the expression (6.21) cannot be reduced below the maximum degree of its components, i.e., $r=s=k=2$. Then by Lemma 6.8 the summation of the above estimated components is the U-statistic estimator for the whole expression; hence it is the UMVUE for $\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\hat{\gamma}$ over $\mathcal{F}$, the class of all continuous distributions for $\mathbf{ts}$ and $\mathbf{tr}$. After approximating the summation (the averaging) over different training sets by the bootstrap replications from one training set the estimator is no longer the UMVUE; rather, it is an approximation to it.

6.4. Simulation Results

In this section we illustrate the approach discussed above with application to several experiments. The classifiers used are the Linear and Quadratic Discriminant Analysis. We use the same experiment parameters described in Section 3.2.1.The first rows of Table 6.1 lists the experiments with their parameters.

The true population parameters of interest are the following: the mean performance $\mathrm{E}_{\mathbf{tr}}\gamma$; the square of the mean performance $\left(\mathrm{E}_{\mathbf{tr}}\gamma\right)^2$; the mean of the squared mean performance $\mathrm{E}_{\mathbf{tr}}\gamma^2$; the variance of the true performance $\mathrm{Var}_{\mathbf{tr}}\gamma$; the mean of variance of the estimator $\hat{\gamma}$, i.e., $\mathrm{E}_{\mathbf{tr}}[\mathrm{Var}_{\mathbf{ts}}\hat{\gamma}]$; and the variance of the estimator $\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\hat{\gamma}$. Table 6.1 illustrates the true value of these parameters, obtained from a large number of Monte-Carlo (MC) trials, for the different experiments shown at the head of the table. Each one of these parameters is estimated from one simulated training set and an independent simulated testing set using the estimators derived in Section 6.3.2. MC trials are also carried out to study the mean and variance of these estimators over many training and testing sets. Every population parameter in the table is followed by the mean and the standard error (obtained over 1000 MC trials) of its estimate. The estimators have some bias, as anticipated, from the nature of the reduction in the effective number of training samples when bootstrapping trainers (cf. Section 6.3.2).

The table shows how the different parameters are estimated almost without bias. However, some bias is observable and attributable back to the bootstrap effect, which accounts for reducing the training set size. The component $\mathrm{Var}_{\mathbf{tr}}\gamma$ is much

Parameters	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**	**Exp. 6**
Classifier	LDA	LDA	QDA	QDA	QDA	LDA
p	3	3	3	5	5	20
$n_{\mathbf{tr}}/2$	7	50	50	20	30	30
$n_{\mathbf{ts}}/2$	70	10	10	20	40	10
$\mathrm{E}_{\mathbf{tr}}\gamma$	.7342	.7983	.7828	.6921	.7197	.7021
$\mathrm{E}_M\widehat{\mathrm{E}_{\mathbf{tr}}\gamma}$	.6970	.7884	.7599	.6480	.6820	.6452
$\mathrm{SD}_M\widehat{\mathrm{E}_{\mathbf{tr}}\gamma}$	.0764	.0991	.1002	.0723	.0580	.0924
$\left(\mathrm{E}_{\mathbf{tr}}\gamma\right)^2$	.5390	.6372	.6128	.4790	.5180	.4930
$\mathrm{E}_M\widehat{\left(\mathrm{E}_{\mathbf{tr}}\gamma\right)^2}$	.4904	.6220	.5783	.4213	.4664	.4173
$\mathrm{SD}_M\widehat{\left(\mathrm{E}_{\mathbf{tr}}\gamma\right)^2}$	.1006	.1564	.1518	.0940	.0789	.1203
$\mathrm{E}_{\mathbf{tr}}\gamma^2$	.5456	.6373	.6130	.4810	.5191	.4943
$\mathrm{E}_M\widehat{\mathrm{E}_{\mathbf{tr}}\gamma^2}$	.4980	.6222	.5789	.4235	.4679	.4192
$\mathrm{SD}_M\widehat{\mathrm{E}_{\mathbf{tr}}\gamma^2}$	.0964	.1564	.1518	.0940	.0787	.1204
$\mathrm{Var}_{\mathbf{tr}}\gamma$	.0065	.0001	.0003	.0021	.0011	.0014
$\mathrm{E}_M\widehat{\mathrm{Var}_{\mathbf{tr}}\gamma}$	.0076	.0003	.0006	.0022	.0015	.0019
$\mathrm{SD}_M\widehat{\mathrm{Var}_{\mathbf{tr}}\gamma}$	.0075	.0013	.0017	.0022	.0011	.0039
$\mathrm{E}_{\mathbf{tr}}[\mathrm{Var}_{\mathbf{ts}}\widehat{\gamma}]$	.0017	.0102	.0110	.0069	.0032	.0140
$\mathrm{E}_M\widehat{\mathrm{E}_{\mathbf{tr}}\mathrm{Var}_{\mathbf{ts}}\widehat{\gamma}}$	.0018	.0106	.0118	.0075	.0035	.0155
$\mathrm{SD}_M\widehat{\mathrm{E}_{\mathbf{tr}}\mathrm{Var}_{\mathbf{ts}}\widehat{\gamma}}$	.0003	.0046	.0042	.0009	.0004	.0025
$\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\widehat{\gamma}$	.0083	.0102	.0112	.0090	.0043	.0153
$\mathrm{E}_M\widehat{\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\widehat{\gamma}}$	.0094	.0109	.0124	.0097	.0050	.0173
$\mathrm{SD}_M\widehat{\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\widehat{\gamma}}$	.0076	.0048	.0046	.0023	.0012	.0043

Table 6.1. Different experiments with different parameters. p is the dimensionality of every problem, $n_{\mathbf{tr}}/2$ and $n_{\mathbf{ts}}/2$ are the training and testing set sizes per class. The true population parameters (obtained from MC experiments); each parameter is followed by the mean and the standard deviation of its estimate (also obtained from repeated MC trials), where each estimate was obtained from a single training and a single testing set. A small bias is traceable to the reduction in support of the training set size under bootstrapping.

influenced in some experiments, e.g., Exp. 2 and 3, with the reduction in the effective sample size coming from bootstrapping. The bias observed for such a component in these experiments has negligible effect on estimating the variance $\mathrm{Var}\widehat{\gamma}$ since the other component, $\mathrm{E}_{\mathbf{tr}}[\mathrm{Var}_{\mathbf{ts}}\widehat{\gamma}]$, dominates. It is remarkable that the results in Exp. 6 are typically well within the mean standard error despite the high dimensionality ($p = 20$).

6.5. Discussion and Remarks

Remark **6.1**. We can examine the connection of the present work with relevant contemporary literature on similar problems, see Roe and Metz (1997*a*) and Roe and Metz (1997*b*), by rewriting (6.14) as

$$\begin{aligned}\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\widehat{\gamma} &= \mathrm{Var}_{\mathbf{tr}}[\mathrm{E}_{\mathbf{ts}}\widehat{\gamma}] + \mathrm{Var}_{\mathbf{ts}}[\mathrm{E}_{\mathbf{tr}}\widehat{\gamma}] - \mathrm{Var}_{\mathbf{ts}}[\mathrm{E}_{\mathbf{tr}}\widehat{\gamma}] + \mathrm{E}_{\mathbf{tr}}[\mathrm{Var}_{\mathbf{ts}}\widehat{\gamma}] \\ &= \underbrace{\mathrm{Var}_{\mathbf{tr}}[\mathrm{E}_{\mathbf{ts}}\widehat{\gamma}]}_{\sigma^2_{\mathbf{tr}}} + \underbrace{\mathrm{Var}_{\mathbf{ts}}[\mathrm{E}_{\mathbf{tr}}\widehat{\gamma}]}_{\sigma^2_{\mathbf{ts}}} + \underbrace{\mathrm{E}\widehat{\gamma}^2 + (\mathrm{E}\widehat{\gamma})^2 - \mathrm{E}_{\mathbf{tr}}[\mathrm{E}_{\mathbf{ts}}\widehat{\gamma}]^2 - \mathrm{E}_{\mathbf{ts}}[\mathrm{E}_{\mathbf{tr}}\widehat{\gamma}]^2}_{C_{\mathbf{tr},\mathbf{ts}}}\end{aligned} \tag{6.37}$$

This problem was previously modeled, see Beiden, Maloof and Wagner (2003), in terms of a linear components-of-variance model as

$$\mathrm{Var}_{\mathbf{tr},\mathbf{ts}}\widehat{\gamma} = \sigma^2_{\mathbf{tr}} + \sigma^2_{\mathbf{ts}} + \sigma^2_{\mathbf{tr},\mathbf{ts}}, \tag{6.38}$$

The similarity and difference of these two results is worthy of comment. It is reasonable to identify the first term of (6.37) with the first term of (6.38), at least conceptually. In the language of the components-of-variance models, they are the pure random effect of the finite size of the training sets; i.e., they are what remains in the variance when the number of testers goes to infinity. Similarly, it is reasonable to match up the second term of both equations. They would be considered the pure random effect of the finite size of the test sets; i.e., they are what remains in the variance when the number of trainers goes to infinity. The third term in the components-of-variance model of (6.38) is referred to as the trainer-tester interaction Beiden, Maloof and Wagner (2003); it would vanish if either the number of trainers or the number of testers goes to infinity. This kind of term is included in these models to allow for flexible variance structures; e.g., to explicitly allow the range of difficulty in the test set to depend on the range of difficulty in the training set. In these models, all of the terms are variances and are thus always nonnegative. The last

term in (6.37), however, can have either sign; thus, it cannot strictly be a variance. This can be proven by two simple examples. If we assume that $\widehat{\gamma} = f_1(\mathbf{tr}) + f_2(\mathbf{ts})$ then it is straightforward to see that $C_{\mathbf{tr},\mathbf{ts}} = 2\,\mathrm{Cov}(f_1, f_2)$, which can have both signs. While if $\widehat{\gamma} = f_1(\mathbf{tr}) \cdot f_2(\mathbf{ts})$ then $C_{\mathbf{tr},\mathbf{ts}} = \mathrm{Var}\, f_1 \cdot \mathrm{Var}\, f_2$, which is always positive.

Nevertheless, one can argue from a practical point of view that it is expected that the range of difficulty in the test set would have a positive covariance with the range of difficulty in the training set, but not vice versa. We conclude that the components-of-variance model is not an unreasonable point of departure for understanding the roles of multiple random effects. We also see that it is not necessary for the solution of the current problem.

Remark **6.2**. The most time consuming estimation is Eq. 6.36. This is $O(B^2 \cdot n_{ts}^4)$, i.e., if it takes one hour on a specific machine it would take 16 hours if the testing set size is doubled. This was a time constraint factor not to extend the number of experiments to include more testing set sizes. However, we took advantage of the sorting techniques and reduced the complexity to $O(B^2 \cdot n_{ts}^3)$. It does not seem possible to us to reduce it below $O(B^2 \cdot n_{ts}^2 \log n_{ts})$. Experiment 1, when 100 bootstraps are used, took 30 hours on a $P4$–2.4 GHz machine.

6.6. Chapter Summary

The present chapter considered assessing classifiers from independent training and testing sets; the metric was the AUC; however it can be immediately applicable to the PAUC by replacing the kernel (6.15) by (5.11). The analysis here assumed no particular distribution, i.e., nonparametric assessment. A closed form expression was derived for the variance of the estimator that estimates the true conditional AUC. The components of that expression are population parameters which are, themselves, important metrics for the classifier, e.g., mean and variance of the AUC from different training sets. U-statistic estimators are derived for these components and for the whole expression. Simulation results show how the proposed methodology is successful even in high dimensionality. The present chapter is very important for those who construct classification rules—as companies submitting new medical testing devices—and have neither enough data to test the new classifier nor parametric knowledge of the data distribution. It is important, as well, for any regulatory-decision makers to assess any new submitted product.

CHAPTER 7

Conclusions, Contributions, and Future Work

This dissertation addressed the classical problem of the assessment of statistical classification rules. The emphasis was on assessing classifiers in terms of ROC analysis. This more general approach is useful in applications where the prevalences of the classes as well as the relative costs of the two kinds of correct and incorrect classifications may vary from one environment to another, leading to the need to consider a range of threshold settings. The contributions of this work are the following:

- Nonparametric estimation for the conditional and the mean performance of classifiers in terms of the Area Under the ROC Curve (AUC) from one data set using the various bootstrap methods. This is an extension of methods defined and used by Efron (1983); Efron and Tibshirani (1997) for estimating the error rate of a classification rule. Until now, the error rate has been the historically predominant performance measure in the literature on statistical pattern recognition.
- Using the method of the influence function to estimate the uncertainty of the estimator above. This is an extension of Efron and Tibshirani (1997) where they used the same method to estimate the uncertainty of the estimator that estimates the mean error rate. The present work required defining the smooth leave-pair-out estimator, which is a two-sample statistic. This is in contrast to Efron's leave-one-out estimator, which is a one-sample statistic for estimating the error rate.
- Proposing the natural extension of the Mann-Whitney kernel to the task of nonparametric estimation of the Partial Area Under the ROC Curve (PAUC), analyzing the properties of the PAUC, and estimating the mean PAUC and the variance of that estimate using the methods summarized above. All of the present estimates are nonparametric; there was no need to make the common parametric assumption of binomial statistics as in medical diagnostics. Although the concept of the partial area precedes the present work, e.g., the parametric version introduced in the field of medical diagnostic testing, there have been no previous developments of the concept of PAUC in the field of statistical pattern recognition where there are two random effects, namely, training and testing. Several new features of the present work are the derivations of the properties of the "true" PAUC, i.e., the PAUC conditional on a particular training data set. The work uncovered several surprising properties of the PAUC.
- Establishing the mathematical nonparametric treatment for assessing classification rules in terms of both the AUC and PAUC from two independent data sets. This approach is a direct application of U-statistics to obtain nonparametric UMVU estimators for different population parameters, e.g., the conditional performance, the mean performance, and the performance variance.

The new solutions sketched above are documented in four publications, Yousef, Wagner and Loew (2004, 2005); Yousef (2013); Yousef, Wagner and Loew (2006) respectively. The fundamental set of problems addressed in this dissertation serves many fields where the general problem of binary classification arises. These fields include medical diagnostics, automatic target recognition, satellite imaging, and data mining. The assessment task is very important for both classifier designers and regulatory agencies that review submitted proposals. When data are scarce and no parametric formulation is possible, the nonparametric assessment techniques discussed in this dissertation have critical relevance.

It is worth mentioning that the techniques specified in this dissertation are independent of the particular classification rule to be used. This is so since all of these techniques are functions of the numerical value of the modeled log-likelihood ratio of every observation in the data sample, no matter which classifier generated this value.

An obvious next step is to apply the techniques developed here to a wide range of classification rules. This will serve not only as a check on the claim made above of the independence of the present methods to the particulars of a classification rule, but also provide a practical demonstration of the generality of this work. It will also serve to show whether there are classes of classification rules and types of data distributions that reveal limitations of those techniques.

Another extension to the dissertation is the application of the techniques in Chapter 6 to the one-data-set paradigm to estimate the variability of the classifier itself. It is expected that the estimation will be better than the one obtained by splitting the available data into two disjoint data sets, since this maximizes data utilization.

A third piece of work is to examine the behavior of different kinds of nonparametric classifiers, e.g., classification and regression trees (CART) and some computational intelligence techniques such as neural networks and fuzzy logic, under situations where data are scarce. A particular limitation in that case is that there are no available data on which to run Monte-Carlo trials

to test the classifier, e.g., as in the case of clinical studies. In such a situation, the overall approach and methods proposed and developed in this dissertation will be the natural candidates for the assessment task.

Bibliography

Anderson, T W. 2003. *An introduction to multivariate statistical analysis*. 3rd ed. Hoboken, N.J.: Wiley-Interscience.

Barndorff-Nielsen, O E and D R Cox. 1989. *Asymptotic techniques for use in statistics*. London; New York: Chapman and Hall.

Barrett, Harrison H, Matthew A Kupinski and Eric Clarkson. 2005. Probabilistic foundations of the {MRMC} method. In *Medical Imaging 2005: Image Perception, Observer Performance, and Technology Assessment*. Vol. 5749 San Diego, CA, USA: SPIE p. 21.

Beiden, S V, M A Maloof and R F Wagner. 2003. "A General Model for Finite-Sample Effects in Training and Testing of Competing classifiers." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25(12):1569.

Bezdek, J C. 1994. What is computational intelligence? In *Computational intelligence : imitating life*, ed. Jacek M Zurada, Robert J Marks and Charles J Robinson. New York: pp. 1–12.

Bezdek, James C. 1992. "On the Relationship Between Neural Networks, Pattern Recognition and Intelligence." *The International Journal of Approximate Reasoning* 6:85–107.

Billingsley, Patrick. 1995. *Probability and measure*. 3rd ed. New York: Wiley.

Bishop, Christopher M. 1995. *Neural networks for pattern recognition*. Oxford; New York: Clarendon Press; Oxford University Press.

Bowerman, Bruce L and Richard T O'Connell. 1990. *Linear statistical models : an applied approach*. 2nd ed. Boston: PWS-Kent Pub. Co.

Bradley, Andrew P. 1997. "The Use of the Area Under the {ROC} Curve in the Evaluation of Machine Learning algorithms." *Pattern Recognition* 30(7):1145.

Breiman, Leo, Jerome Friedman, Richard Olshen and Charles Stone. 1984. *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group.

Campbell, G, M A Douglas and J J Bailey. 1988. Nonparametric comparison of two tests of cardiac function on the same patient population using the entire {ROC} curve. In *Computers in Cardiology. IEEE Proceedings*. p. 267.

Casella, George and Roger L Berger. 2002. *Statistical inference*. 2nd ed. Australia ; Pacific Grove, CA: Duxbury/Thomson Learning.

Chan, H P, B Sahiner, R F Wagner and N Petrick. 1999. "Classifier Design for Computer-Aided Diagnosis: Effects of Finite Sample Size on the Mean Performance of Classical and Neural Network classifiers." *Medical Physics* 26(12):2654–2668.

Christensen, Ronald. 2002. *Plane answers to complex questions : the theory of linear models*. 3rd ed. New York: Springer.

Davison, A C, D V Hinkley and E Schechtman. 1986. "Efficient Bootstrap simulation." *Biometrika* 73(3):555–566.

DeLong, Elizabeth R, David M DeLong and Daniel L Clarke-Pearson. 1988. "Comparing the Areas Under Two Or More Correlated Receiver Operating Characteristic Curves: a Nonparametric approach." *Biometrics* 44(3):837–845.

Duda, Richard O, Peter E Hart and David G Stork. 2001. *Pattern classification*. 2nd ed. New York: Wiley.

Efron, B and C Stein. 1981. "The Jackknife Estimate of Variance." *The Annals of Statistics* 9(3):586–596.

Efron, Bradley. 1975. "The Efficiency of Logistic Regression Compared To Normal Discriminant Analysis." *Journal of the American Statistical Association* 70(352):892–898.

Efron, Bradley. 1979. "Bootstrap Methods: Another Look At the Jackknife." *The Annals of Statistics* 7(1):1–26.

Efron, Bradley. 1981. "Nonparametric Estimates of Standard Error: the Jackknife, the Bootstrap and Other Methods." *Biometrika* 68(3):589–599.

Efron, Bradley. 1982. *The jackknife, the bootstrap, and other resampling plans*. Philadelphia, Pa.: Society for Industrial and Applied Mathematics.

Efron, Bradley. 1983. "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation." *Journal of the American Statistical Association* 78(382):316–331.

Efron, Bradley. 1986. "How Biased Is the Apparent Error Rate of a Prediction Rule?" *Journal of the American Statistical Association* 81(394):461–470.

Efron, Bradley. 1992. "Jackknife-After-Bootstrap Standard Errors and Influence Functions." *Journal of the Royal Statistical Society. Series B (Methodological)* 54(1):83–127.

Efron, Bradley and Robert Tibshirani. 1993. *An introduction to the bootstrap*. New York: Chapman and Hall.

Efron, Bradley and Robert Tibshirani. 1995. "Cross Validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule." *Technical Report 176, Stanford University, Department of Statistics* .

Efron, Bradley and Robert Tibshirani. 1997. "Improvements on Cross-Validation: the .632+ Bootstrap Method." *Journal of the American Statistical Association* 92(438):548–560.

Engelbrecht, Andries P. 2002. *Computational intelligence : an introduction*. Chichester, England ; Hoboken, N.J.: J. Wiley \& Sons.

Friedman, Jerome H and Werner Stuetzle. 1981. "Projection Pursuit Regression." *Journal of the American Statistical Association* 76(376):817–823.

Fukunaga, K and R R Hayes. 1989*a*. "Effects of Sample Size in Classifier design." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 11(8):873–885.

Fukunaga, K and R R Hayes. 1989*b*. "Estimation of Classifier performance." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* - 11(- 10):– 1101.

Fukunaga, Keinosuke. 1990. *Introduction to statistical pattern recognition*. 2nd ed. Boston: Academic Press.

Gallas, Brandon D. 2006. "One-Shot Estimate of {MRMC} Variance: {AUC}." *Academic Radiology* 13(3):353.

Graybill, Franklin A. 1976. *Theory and application of the linear model*. North Scituate, Mass.: Duxbury Press.

Hampel, Frank R. 1974. "The Influence Curve and Its Role in Robust Estimation." *Journal of the American Statistical Association* 69(346):383–393.

Hampel, Frank R. 1986. *Robust statistics : the approach based on influence functions*. New York: Wiley.

Hanley, J A and B J McNeil. 1982. "The Meaning and Use of the Area Under a Receiver Operating Characteristic ({ROC}) curve." *Radiology* 143(1):29–36.

Hastie, Trevor and Robert Tibshirani. 1990. *Generalized additive models*. 1st ed. London ; New York: Chapman and Hall.

Hastie, Trevor, Robert Tibshirani and J H Friedman. 2001. *The elements of statistical learning : data mining, inference, and prediction*. New York: Springer.

H\'{a}jek, Jaroslav, Zbyn\v{e}k \v{S}id\'{a}k and Pranab Kumar Sen. 1999. *Theory of rank tests*. 2nd ed. San Diego, Calif.: Academic Press.

Huber, Peter J. 1996. *Robust statistical procedures*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics.

Hughes, Gordon F. 1968. "On the Mean Accuracy of Statistical Pattern recognizers." *Information Theory, IEEE Transactions on* 14(1):55.

Jaeckel, L. 1972. "The Infinitesimal jackknife." *Memorandum, MM 72-1215-11, Bell Lab. Murray Hill, N.J.* .

Jiang, Y, C E Metz and R M Nishikawa. 1996. "A Receiver Operating Characteristic Partial Area Index for Highly Sensitive Diagnostic tests." *Radiology* 201(3):745–750.

Jiang, Y, R M Nishikawa, R A Schmidt, C E Metz, M L Giger and K Doi. 1999. "Improving Breast Cancer Diagnosis With Computer-Aided diagnosis." *Academic Radiology* 6(1):22–33.

Lehmann, E L and George Casella. 1998. *Theory of point estimation*. 2nd ed. New York: Springer.

Lehmann, E L and Joseph P Romano. 2005. *Testing statistical hypotheses*. 3rd ed. New York: Springer.

Li, Ker-Chau. 1991. "Sliced Inverse Regression for Dimension Reduction." *Journal of the American Statistical Association* 86(414):316–327.

Mallows, C. 1974. "On Some Topics in robustness." *Memorandum, MM 72-1215-11, Bell Lab. Murray Hill, N.J.* .

McClish, D K. 1989. "Analyzing a Portion of the {ROC} curve." *Med Decis Making* 9(3):190–195.

Nadaraya, Elizbar A. 1964. "On Estimating Regression." *Theory of Probability and Its Applications* 9(1):141–142.

Newman, D J and A Asuncion. 2007. "UCI Machine Learning Repository." *University of California, Irvine, Dept. of Information and Computer Sciences* .

Parzen, Emanuel. 1962. "On Estimation of a Probability Density Function and Mode." *The Annals of Mathematical Statistics* 33(3):1065–1076.

Randles, Ronald H and Douglas A Wolfe. 1979. *Introduction to the theory of nonparametric statistics*. New York: Wiley.

Rencher, Alvin C. 2000. *Linear models in statistics*. New York: Wiley.

Ripley, Brian D. 1996. *Pattern recognition and neural networks*. Cambridge ; New York: Cambridge University Press.

Roe, C A and C E Metz. 1997*a*. "Dorfman-Berbaum-Metz Method for Statistical Analysis of Multireader, Multimodality Receiver Operating Characteristic Data: Validation With Computer simulation." *Academic Radiology* 4(4):298–303.

Roe, C A and C E Metz. 1997*b*. "Variance-Component Modeling in the Analysis of Receiver Operating Characteristic Index estimates." *Academic Radiology* 4(8):587–600.

Sahiner, B, H P Chan, N Petrick, L Hadjiiski, S Paquerault and M N Gurcan. 2001. "Resampling Schemes for Estimating the Accuracy of a Classifier Designed With a Limited Data Set." *Medical Image Perception Conference IX, Airlie Conference Center, Warrenton VA, 20-23* .

Schott, James R. 2005. *Matrix analysis for statistics*. 2nd ed. Hoboken, N.J.: Wiley.

Schwefel, Hans-Paul, Ingo Wegener and Klaus Weinert. 2003. “Advances in computational intelligence : theory and practice.”.

Searle, S R. 1982. *Matrix algebra useful for statistics*. New York: Wiley.

Stone, M. 1974. “Cross-Validatory Choice and Assessment of Statistical Predictions.” *Journal of the Royal Statistical Society. Series B (Methodological)* 36(2):111–147.

Swets, J A. 1986. “Indices of Discrimination Or Diagnostic Accuracy: Their {ROC}s and Implied Models.” *Psychological Bulletin* 99:100–117.

Watson, Eoffrey S. 1964. “Smooth Regression Analysis.” *Sankhy\={a}: The Indian Journal of Statistics* Series A,:359–372.

Yousef, W A, R F Wagner and M H Loew. 2004. Comparison of Non-Parametric Methods for Assessing Classifier Performance in Terms of {ROC} Parameters. In *Applied Imagery Pattern Recognition Workshop, 2004. Proceedings. 33rd; IEEE Computer Society*. pp. 190–195.

Yousef, W A, R F Wagner and M H Loew. 2006. “Assessing Classifiers From Two Independent Data Sets Using {ROC} Analysis: a Nonparametric Approach.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(11):1809–1817.

Yousef, Waleed A. 2013. “Assessing Classifiers in Terms of the Partial Area Under the Roc curve.” *Computational Statistics & Data Analysis* 64(0):51–70.
URL: *https://doi.org/10.1016/j.csda.2013.02.032*

Yousef, Waleed A, Robert F Wagner and Murray H Loew. 2005. “Estimating the Uncertainty in the Estimated Mean Area Under the {ROC} Curve of a Classifier.” *Pattern Recognition Letters* 26(16):2600–2610.

Zhang, P. 1995. “Assessing Prediction Error in Nonparametric Regression.” *Scandinavian Journal Of Statistics* 22(1):83–94.

Zimmermann, Hans-J\”{u}rgen, Georgios Tselentis, Maarten van Someren and Deorgios Dounias. 2002. “Advances in Computational Intelligence and Learning : Methods and Applications.”.