Title
Considerations in using Diagnostic Tests for Disease Classification

Author
Huynh, Dat

Publication Date
2014

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Considerations in using Diagnostic Tests for Disease Classification

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Biostatistics

by

Dat Thanh Huynh

2014

Abstract of the Dissertation

Considerations in using Diagnostic Tests for Disease Classification

by

Dat Thanh Huynh

Doctor of Philosophy in Biostatistics

University of California, Los Angeles, 2014

Professor Ron Brookmeyer, Chair

In this thesis, we consider statistical issues in classification for disease using diagnostic testing. We discuss two aspects of disease classification: the generation of testing algorithms to combine multiple diagnostic tests that address both accuracy and cost considerations and the application of an imperfect diagnostic test to determine cases in a case-control study.

Motivated by the problem of combining multiple biomarkers to identify recent HIV infection (< 1 year), we first develop methods for identifying "serial testing algorithms" to reduce the cost of diagnosis. These "serial testing algorithms" are characterized by the ability to make a classification determination before all diagnostic markers are acquired. These algorithms are able to maintain accuracy while controlling costs of the diagnostic testing.

We present two approaches to this problem. A logic regression approach in which serial testing algorithms are developed by means of logical combinations of dichotomous tests. Testing costs are optimized through a permutation algorithm on the logical rule. We also develop a serial risk score classification approach. In this method, we establish multiple ordered stages of classification determined by a risk score model. In each stage, one or more diagnostic tests are added to the risk

score model from the previous stage and each observation is either determined to continue on for further testing or classified as positive or negative.

The methods are studied in simulations and compared with logistic regression. We applied the methods to data from HIV cohort studies to identify HIV infected individuals who are recently infected (< 1 year) by testing with assays for multiple biomarkers. The biomarkers that we used as part of the classification rule were the CD4 count, viral load, BED assay and avidity assay. We find that serial testing algorithms can maintain accuracy while achieving a reduction in cost compared to testing all individuals with all assays.

We then investigate the application of a non-gold standard test to a case control study. This work was motivated by case-control studies for risk factors associated with recent (<1 year) HIV infection when the duration of infection cannot be directly observed. In this type of study, recently (< 1 year) and chronically (> 1 year) infected people represent two types of cases. When the case type is misclassified, the usual standard estimates for an odds ratio associated with one of the case types can be biased. We discuss methods to adjust the odds ratio from a case control study using the performance characteristics of a classification rule. In particular, we discuss a matrix adjustment method to adjust the observed counts of each case type, and an adjustment method based on a multinomial logistic regression model. These methods have shown to reduce bias in the estimation of the odds ratio.

We conclude with a discussion of the described methods in disease classification that were motivated by problems in HIV research. These problems included the cost of diagnostic tests and the fact that dates of infection cannot often be determined. The methods we developed may also have application to other settings especially when the costs of diagnostic testing is high and there are multiple types of cases that cannot be distinguished with complete accuracy.

The dissertation of Dat Thanh Huynh is approved.

William G. Cumberland

Pamina Gorbach

Catherine Sugar

Ron Brookmeyer, Committee Chair

University of California, Los Angeles

2014

I would like to dedicate this thesis to my mother and father who have supported me all these years.

Thank you to my brother Cong and sister Thao who inspire me and have helped me achieve all of my goals.

Table of Contents

List of Figures . . . . . **ix**

List of Tables . . . . . **x**

1 Introduction **1**

1.1 Preliminary Concepts 4

1.1.1 Notation 4

1.2 Summary of Dissertation 6

2 Serial Testing Algorithm Optimization **7**

2.1 Introduction 7

2.2 Preliminary Concepts 8

2.2.1 Definitions 8

2.2.2 Optimality Criterion 11

2.3 Classification Methods 12

2.3.1 Risk score Estimation 12

2.3.2 Logic Regression 13

2.3.3 Serial Risk Score Algorithm 17

2.3.4 Methods and Estimation of a Serial Risk Score Algorithm 21

2.3.5 Evaluation of Methods 24

2.4 Application to HIV Data 33

2.4.1 Description of Data 33

2.4.2 Application of Logistic Regression to HIV data 35

2.4.3 Application of Logic Regression to HIV data 36

2.4.4 Application of Serial Risk Score to HIV data 38
2.5 Discussion 46

3 Estimation of the Odds Ratio under Misclassification of Cases 48
3.1 Introduction 48
3.1.1 Preliminary Concepts 49
3.2 Estimators of Odds Ratio 51
3.2.1 Naïve Estimator 51
3.2.2 Matrix-adjusted Odds Ratio 55
3.2.3 Limitations of Matrix Adjustment Method 58
3.3 Multinomial Logistic Regression Estimator 60
3.3.1 Multinomial Logistic Regression Under Cross-sectional Sampling 61
3.3.2 Multinomial Logistic Regression Under Case-control Sampling 61
3.3.3 Identifiability Conditions 62
3.3.4 Simulation 63
3.4 Discussion 67

4 Discussion 68

5 Bibliography 71

Appendix A Comparison of Tolerance Values in Serial Risk Score Simulation 76

Appendix B Technical Details for the Adjusted Multinomial Logistic Model 79

Appendix C R Source Code Listings 82

List of Figures

2.1 Decision tree representations of logical rules 10

2.2 Illustration of swaps of logical operands in a decision tree 16

2.3 Illustration plot where a logical rule may fail 17

2.4 Illustration of serial risk score classifier 20

2.5 Smoothed density curves of the biomarkers 36

2.6 Scatter plot of biomarkers 37

2.7 ROC curves using logic regression classification algorithm 39

2.8 ROC curves using serial risk score classification algorithm 41

3.1 Percent error of $\bar{\omega_1}$ by ω_1/ω_2 54

3.2 Odds ratio estimates (OR) with the matrix adjusted estimator and the naïve estimator with increasing sample size 57

List of Tables

2.1 Translation table of a logical rule to a serial testing rule 15

2.2 Monte Carlo simulation accuracy estimates of classification algorithms from logistic regression, logic regression and serial risk score classification with model parameters $\rho = 0.5$, $\beta_1 = 0.69, \beta_2 = 0.37$ 29

2.3 Monte Carlo simulation accuracy estimates of classification algorithms from logistic regression, logic regression and serial risk score classification with model parameters $\rho = 0.5$, $\beta_1 = 0, \beta_2 = 0.37$ . . 30

2.4 Monte Carlo simulation accuracy estimates of classification algorithms from logistic regression, logic regression and serial risk score classification with model parameters $\rho = 0$, $\beta_1 = 0.69, \beta_2 = 0.37$. 31

2.5 Monte Carlo simulation accuracy estimates of classification algorithms from logistic regression, logic regression and serial risk score classification with model parameters $\rho = 0$, $\beta_1 = 0, \beta_2 = .37$ 32

2.6 Biomarker means and standard deviations of chronic and recently infected observations. 35

2.7 Estimated coefficients of logistic risk score model for HIV data . . 38

2.8 Biomarker threshold values by biomarker 42

2.9 Rules generated from logic regression classification method 43

2.10 Sensitivity, specificity, and average cost savings of serial risk score classification and logistic regression applied to dataset described in Section 2.4 44

2.11 Estimated regression coefficients to serial risk score rule chosen for $p = 0.30$ when $t = 0.10$ 45

3.1 Sample counts divided into cells indicating classification of disease stage and exposure E 50

3.2 Example data for limiting case of matrix adjustment 59

3.3 Monte Carlo simulation results with cross-sectional sampling comparing the naïve and likelihood adjusted multinomial regression estimators of the odds ratio 65

3.4 Monte Carlo simulation results with case-control sampling comparing the naïve and likelihood adjusted multinomial regression estimators of the odds ratio 66

A.1 Monte Carlo simulation[1] estimates for sensitivity (Sens.), specificity (Spec.) and average cost savings (ACS) over a range of tolerance values. Here we assume correlated biomarkers ($\rho = .5$) and two models for the data as specified in Section 2.3.5.3. 77

A.2 Monte Carlo simulation[1] estimates for sensitivity (Sens.), specificity (Spec.) and average cost savings (ACS) over a range of tolerance values. Here we assume uncorrelated biomarkers ($\rho = 0.0$) and two models for the data as specified in Section 2.3.5.3. 78

Acknowledgments

I would like to acknowledge and express all of my gratitude for all of those who have helped me in the research and writing of this thesis.

First and foremost, Dr. Ronald Brookmeyer for his guidance and patience throughout the writing of this research. I would also like to thank my committee members: Dr. William G. Cumberland, Dr. Pamina Gorbach, and Dr. Catherine Sugar for their encouragement and support without which, this work would not be possible.

Finally, I thank the NIH grant 5-T32-AI007370 Biostatistical Training in AIDS for financial support.

Vita

2003	B.S. (Computer Science and Engineering), University of California, Los Angeles.
2005–2010	Programmer/Analyst, Human Genetics Department, UCLA.
2010	M.S. (Biostatistics), University of California, Los Angeles.

Publications and Presentations

Huynh, D. and Brookmeyer R. (July 2012). "*Optimizing Serial Diagnostic Testing to Reduce Cost.*". Presented at the Joint Statistical Meeting, San Diego, CA

Huynh, D. and Brookmeyer R. (July 2013). "*Adjusting Odds Ratios For Misdiagnosis Of Cases In Case Control Studies.*" Presented at the Joint Statistical Meeting, Montreal, QC

Huynh, D., Laeyendecker, O., and Brookmeyer, R. (in review) "*A Serial Risk Score Approach to Disease Classification that Accounts for Accuracy and Cost,*" Submitted to Biometrics, March 2014.

CHAPTER 1

Introduction

Diagnostic testing is the process of determining the disease state of a person. It can be thought of as a specific kind of a classification problem where one decides if a person has a certain condition or does not. It can be characterized by the use of biomarkers such as immunological assays to inform classification determinations. Procedures for diagnostic testing can be both costly and time consuming. Furthermore, a diagnostic test may not have perfect classification ability which may affect the inferences made as a result.

In this thesis, we discuss considerations in using diagnostic tests for disease classification. First, we discuss considerations in cost for diagnostic testing. Multiple diagnostic tests are often used to make an accurate diagnosis and with each diagnostic test, costs will accumulate. A serial testing algorithm can reduce the cost of diagnosis by classifying some subjects with fewer tests. A serial testing algorithm is a classification method where tests are performed in a specified order, one at a time. The costs associated with diagnosis can be reduced if classification can occur before all tests are performed.

To make this concrete, consider the following simple example. Suppose there are 2 diagnostic tests for a disease, B_1 and B_2 (e.g. assays for two different biomarkers). These tests can be combined into a classification rule such as '*either positive*' or '*both positive*' where the former refers to a positive diagnosis of disease if either test is positive and the latter referring to positive diagnosis if both tests are positive. If we use an '*either positive*' rule, and B_1 is positive, there is no need

to test with B_2. Similarly, if we use a '*both positive*' strategy and B_1 is negative, there again is no need to test with B_2.

Serial testing algorithms have been used in the Ivory Coast specifically to reduce the cost of HIV testing with great success. In 1999, Nkengasong and colleagues compared the performance of using a combination of three ELISA assays (ICE 1.0.2, Enzygnost, and Vironostika) for comparison to a standard algorithm (Peptilav and p24 antigen assay) (Nkengasong et al., 1999). The algorithm tested the sera with Enzygnost test and considered non-reactive sera as true HIV-negative. If the sera was reactive with the Enzygnost test, the ICE 1.0.2 assay was used and positive sera from the ICE 1.0.2 assay was considered true HIV-positive. If there were discordant results, the Vironostika assay was used and the outcome of that assay was considered true. These algorithms resulted in 100% sensitivity and 99.95% specificity when compared to the standard. The advantage is that the serial algorithm cost US$ 23,432 overall (US$ 2.80 per sample), while the the standard Peptilav algorithm cost US$ 77,975 (US$ 9.50 per sample). The result was 70% cost savings for the serial algorithm compared to the Peptilav algorithm.

These algorithms can be applied to defining cases in a case-control study. Consider a case-control study examining the incidence of HIV. A simple comparison of risk factors between sexually active HIV+ and HIV- groups (unaware of status) may dilute the effect of risk factors. The reason for this dilution can be because of a combination of two factors. The first factor is the long incubation time of HIV where behavior changes can occur. The second factor is that the behavior of a person can tend towards less risky behavior over time. For example, a 25 year old HIV+ person who was infected at age 20 may have altered risk behavior patterns such as increased condom usage or reduced the number of sex partners during the 5 year period of infection.

Many studies have found associations between age and incidence of HIV infection. Studies of MSM populations in Western countries have shown younger

populations (≤ 30) having greater frequencies of unprotected anal intercourse and increased incidence of HIV (Mansergh and Marks, 1998; Crepaz et al., 2000). Studies have shown that South African youth lack an understanding of the nature of HIV and perceive a low risk of infection. A recent survey among South African youth revealed 61% of HIV positive and 73% of HIV negative youth reporting that they thought they were at no risk at all or had a small chance of getting HIV (Eaton et al., 2003; Pettifor et al., 2005).

In addition, Pines et al. (2013) identified three sexual risk trajectories (low, moderate and high-risk) among MSM in the United States which exemplifies the transient nature of HIV risk behavior. Minimal changes in probability of engaging in high-risk behaviors were reported over time for the low and high risk trajectory groups, but the moderate risk trajectory group showed a strong decline of 29% to 17%. They also identified temporary attributes such as depression and "seasons of risk" that may also play a role in increased risk behavior as well.

To resolve this issue, we suggest a comparison of recently infected (<1 year) HIV+ and HIV- groups to describe the leading edge of HIV transmission risk. The problem is that while we can easily identify infection status, it is difficult to separate those with chronic (>1 year) infections from recent infections. The disease history of each person may be unknown, so other methods for classification must be used. An early method in HIV research uses detuned assays to identify recent infections (Janssen et al., 1998). Another recent method has been described to use serological biomarkers to a determine disease status (Laeyendecker et al., 2013). Neither algorithm provided perfectly accurate classification.

The serial testing algorithms we describe in Chapter 2 can be used to distinguish between recently and chronically infected individuals with reasonable but imperfect accuracy and a reduced cost; however, misclassification of the cases will tend to bias the odds ratio towards the null and reduce the power of the study. In Chapter 3, we discuss adjustments to the odds ratio based on knowledge of

the misclassification rates to reduce the bias that is induced from the imperfect classification.

1.1 Preliminary Concepts

To set the stage for our discussion, we describe preliminary concepts that will be used throughout this thesis in this section.

1.1.1 Notation

Consider a sample study population of size N where each person is either diseased or not diseased. Within the diseased group, a further division can be made into two types: "case I" and "case II". Let Y_i denote the disease status of the individual i in the study population. Let $Y_i = 0$ indicate that the individual is uninfected and is in the control group. Let $Y_i = j$ (for $j = +1, -1$) indicate an infected individual in case group j where case I is indicated by $+1$ and case II is indicated by -1. Let $\widehat{Y}_i$ denote the observed disease status that is obtained from a classification algorithm. In addition, assume that an exposure E can be determined for each person. We restrict the exposure of interest to a dichotomous exposure that is accurately classified.

A classification algorithm combines multiple biomarkers in $\mathbf{X}_i = \{X_{i1}, ..., X_{ip}\}$ and provides the predicted outcome $\widehat{Y}_i$ for the true classification indicator Y_i. Classification methods will "search" over the classification algorithm space and find one with optimal performance, say most accurate or most specific. In this thesis, we assume $Y_i = 0$ to be perfectly classified through external means. The discussion will focus on only the classification between the two case types. For a sample of size N, we define the quantities: number of false positives (FP) and

number of false negatives (FN) for a algorithm r applied to a certain sample,

$$FP(r) = \sum_{\{i|Y_i=-1\}} I(\widehat{Y}_i = +1) \text{ and } FN(r) = \sum_{\{i|Y_i=1\}} I(\widehat{Y}_i = -1).$$

We define the number of true positives (TP) and true negatives (TN) similarly,

$$TP(r) = \sum_{\{i|Y_i=+1\}} I(\widehat{Y}_i = +1) \text{ and } TN(r) = \sum_{\{i|Y_i=-1\}} I(\widehat{Y}_i = -1).$$

The sensitivity and specificity of a classification algorithm are commonly used in epidemiological settings to describe classification accuracy. The sensitivity (Se) of a classification algorithm describes its ability to identify a positive subject. We define the population sensitivity for a rule to be a conditional probability

$$Se = Pr(\widehat{Y} = -1|Y = -1)$$

which we estimate from a sample with

$$\overline{Se} = TP/(TP + FN).$$

We refer to $\overline{Se}$ as the *apparent* sensitivity. The specificity (Sp) of a classification algorithm describes its ability to identify a negative subject. The population specificity is defined as

$$Sp = Pr(\widehat{Y} = 0|Y = 0)$$

which we estimate using

$$\overline{Sp} = TN/(TN + FP).$$

We refer to $\overline{Sp}$ as the *apparent* specificity.

For a given classification algorithm, the receiver operating characteristic (ROC) curve is a plot of the sensitivity vs. (1 - specificity) when changing an optimality criteria. The ROC curve allows us compare our classification methods over various threshold settings as well as over several optimal rules. We use the area under the ROC curve as an overall comparison measure of the performance of our classification methods under study without choosing an explicit optimality criterion.

1.2 Summary of Dissertation

This thesis is organized as follows. In Chapter 2, we will discuss methods for generating serial testing algorithms to save diagnostic costs while combining multiple diagnostic tests to make accurate classifications. We describe two classification methods, one based on logic regression and another novel approach called serial risk score classification. In Chapter 3, we develop a method for adjusting the estimate of an odds ratio between a case and control group in the aforementioned setting where case types can be misclassified. Lastly, we provide final remarks and a discussion of extensions to the discussed methods in Chapter 4.

CHAPTER 2

Serial Testing Algorithm Optimization

2.1 Introduction

As previously mentioned, diagnostic testing for classifying a person into one of two states (e.g. disease or no disease) can be costly, especially if multiple diagnostic tests are required to make accurate diagnoses. While increasing the number of biomarkers may increase accuracy, that comes at the price of increased costs of the assays. The trade-off between accuracy and costs in diagnostic classification is of concern particularly in countries with limited resources where adequate screening is costly (Parpia et al., 2010).

In this chapter, we examine serial testing algorithms used for classifying HIV infected persons as to whether their infection occurred recently or not (e.g., within the previous year or not). The problem of accurately identifying recently occurring infections is important for several reasons. It has been shown that individuals who are recently infected have higher rates of HIV transmission (Wawer et al., 2005). It is important to identify sexual partners of those recently infected so they can seek testing and care. Treatment decisions may also depend on the recency of infections (Sáez-Cirión et al., 2013). Additionally, knowledge of recency of infection is useful in studies of risk factors for HIV infection and may help identify subgroups of the population where HIV incidence is growing rapidly.

Our work builds upon a large body of research in biomarker-based HIV incidence estimates that began with work by Brookmeyer and Quinn in 1995. They

developed an approach to estimate incidence of infections using the number of seronegative samples in a population that tested positive for HIV p24 antigen; however, cost of the p24 antigen test and the short window period between detection of the p24 antigen and seropositivity suggested that this method was only suitable for large samples in population with high incidence (Brookmeyer and Quinn, 1995). Concurrent work by Janssen and colleagues published in 1998 used a 'detuned', or less sensitive, assay to discriminate recent and chronic infections (Janssen et al., 1998). These early studies ultimately led to the development of a variety of different serological assays for testing recent infection.

The testing algorithms presented in this paper combine 4 biomarker assays with well known trajectories over the course of HIV infection. In the next section, we discuss preliminary concepts important for the understanding of the methodology discussed in this paper. In section 2.3, we define two classification methods that we apply to a serial testing framework. One of the methods is based on an existing procedure called logic regression. The other is a novel method to combine a serial testing algorithm with logistic regression to improve discrimination performance versus current methods. We then evaluate and compare our methods when applied to a cohort data in section 2.4.

2.2 Preliminary Concepts

2.2.1 Definitions

A continuous biomarker can be transformed into a dichotomous test with a discrimination threshold or cut off. The threshold defines the line dividing a positive classification region in the biomarker space and the negative classification region. Usually the test is represented as $T = (X > c)$ where X is the continuous biomarker and c is the threshold value.

We can combine multiple tests using a logical rule. Here we define a logical rule to be a combination of tests with binary operators $\wedge$ (and), $\vee$ (or), $\neg$ (not). The logical rule can form a basis for classification as in the following example:

$$Y = \begin{cases} +1, & \text{if } r(X_1, X_2) = \text{true} \\ -1 & \text{otherwise} \end{cases} \tag{2.1}$$

where

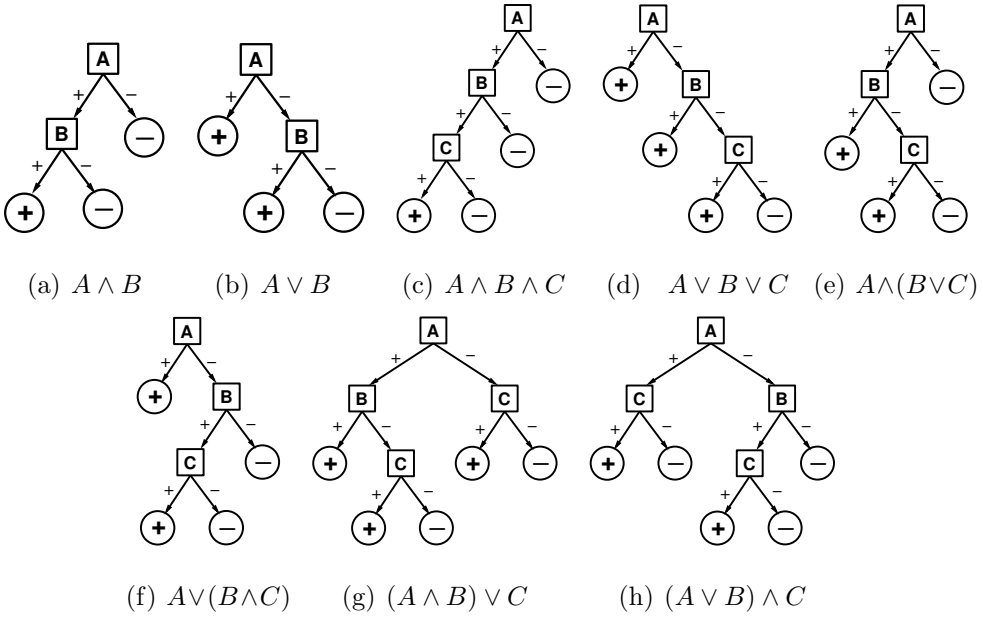
$$r(X_1, X_2) = (X_1 > c_1) \vee (\neg(X_1 > c_1) \wedge (X_1 > c_2) \wedge (X_2 > c_3)).$$

The operator $\wedge$ results in a positive classification if both the operands are positive and negative otherwise. The operator $\vee$ results in a positive classification if either of the operands are positive and negative otherwise. The operator $\neg$ results in the complement of the result of the operand. The parentheses are used in a normal fashion and define precedence when combining more than two tests.

A serial testing algorithm is a decision tree representation of a logical rule. Define a stage to be a node on the decision tree where a classification for an individual can occur. For some stage s, the stage index i is defined by the total number of stages + 1 that is traversed in a path to s starting from the root. In each serial testing algorithm, we have M maximum number of stages. A serial testing algorithm is distinct from a logical rule in that there is a specified ordering. The ordering of tests in a serial testing algorithm is very important as it impacts cost.

A graphical representation of some serial testing algorithms is given in Figure 2.1. The squares labeled A, B and C are representations of tests. The arrows labeled + and - are the resulting outcomes from each test where indicates a true result and - indicates a false result. The circles are the overall classification outcome for a given rule. In figure 2.1(c), we see the representation of $A \wedge B \wedge C$. Following the diagram, if one were to implement this algorithm for testing an observation, if test A corresponding to the observation has a positive result, one

continues to test B, otherwise the observation is classified as negative and testing stops. If testing continues to test B and the result of test B is negative, then the observation is classified as negative, otherwise testing continues to test C, in which case final determination must be made.



(a) $A \wedge B$ (b) $A \vee B$ (c) $A \wedge B \wedge C$ (d) $A \vee B \vee C$ (e) $A \wedge (B \vee C)$

(f) $A \vee (B \wedge C)$ (g) $(A \wedge B) \vee C$ (h) $(A \vee B) \wedge C$

Figure 2.1: Decision tree representation of logical classification rules combining tests A, B, C. Lines represent outcomes from a test. Circles represent resulting classification from combined tests. Cost savings occur at intermediate nodes below the bottom level of each tree.

2.2.2 Optimality Criterion

We define an optimal algorithm r^* to be an algorithm in the algorithm space $\mathbf{R}$ defined by the biomarker set $\boldsymbol{X}$ that minimizes a loss function $L(\cdot)$. The set of optimal algorithms form a subset of $\mathbf{R}$, $\mathbf{R}^* = \{r^*\}, \mathbf{R}^* \subset \mathbf{R}$.

One example of a classification loss function L uses a weighted average of false negatives and false positives, i.e. $L(r, p) = pFP(r) + FN(r)$ where p is the relative contribution to the loss of a false positive (FP) compared to a false negative (FN). Larger values for p will indicate a preference for lower rates of false positives to the classification algorithm i.e. a preference for specificity. Similarly, smaller values of p indicate a preference for lower rates of false negatives and sensitivity. This loss function was adopted to generate the logic rule ROC curve (Etzioni et al., 2003). We continue the use of this loss function for consistency when drawing the ROC curves presented in the next section.

The total cost of an algorithm is defined by the number of individuals who are tested with a diagnostic test and the cost of the diagnostic test itself. Let Z be the set of a diagnostic tests used in a classification algorithm. Let b_i be the cost for diagnostic test z_i. Let n_{zi} be the number of individuals tested with z_i in an algorithm. For example, suppose an algorithm uses diagnostic test z_i in the first stage, then in this case, $n_{zi} = N$. We define the total cost of an algorithm by:

$$\text{cost} = \sum_{z_i}^{Z} b_i n_{zi}.$$

We define the average cost savings (ACS) of an algorithm by the "discount" of using the serial testing algorithm versus a classification algorithm that uses all diagnostic tests with all individuals.

$$\text{ACS} = \sum_{z_i}^{Z} b_i (N - n_{zi})/N).$$

We can expand the loss function to include the cost of diagnosis

$$L_r(p_1, p_2, cost) = p_1 FP + p_2 FN + h(cost). \tag{2.2}$$

where $h(\cdot)$ is a function of the diagnostic cost; however, the parameters p_1, p_2 and the function $h(\cdot)$ depend greatly on the study population, the biomarkers required in the classification routine, and the objectives of the study. It may not be clear what parameters are appropriate for a specific application as arbitrary selection may emphasize one particular feature of the loss function when a balance is appropriate. An alternative approach to balancing diagnostic accuracy and cost is to first minimize the loss function $L(p)$ and then consider cost among candidate algorithms that have the same minimal value.

2.3 Classification Methods

2.3.1 Risk score Estimation

Methods for classification using multiple diagnostic tests have been extensively developed. A fundamental approach is based on estimating risk scores, $S(\mathbf{X}) = P(Y = +1|\mathbf{X})$ where $\mathbf{X}$ is a vector of predictors (e.g., diagnostic tests or assays for biomarkers) and $Y = +1$ if a person has the disease and $Y = -1$ otherwise (Pepe, 2003).

A standard tool for risk score modeling for binary classification is logistic regression. In order to estimate the risk score for each of the subjects, we use the biomarkers as predictors in the logistic regression model and the true classification of the subject as the response variable. More explicitly, we fit models of the form

$$log\left(\frac{RS(\mathbf{X})}{1 - RS(\mathbf{X})}\right) = \beta_0 + h(\beta, \mathbf{X}). \tag{2.3}$$

In this approach, a person is classified as positive with disease if $RS(\mathbf{X}) > c$ and otherwise negative without disease, where c is the discrimination threshold and is chosen according to optimality constraints (i.e. minimizing the loss function).

In our discussion, we only use linear first-order terms with no interaction terms; this is a controversial choice as it has been shown that using linear rules can be too restrictive (McIntosh and Pepe, 2002). In practice, a more complex model would be appropriate, but we use the linear terms here for comparison with our other models. Prediction of the risk score will require that we have all of the biomarker information available at time of classification. The total cost of logistic regression will be the cost of all biomarkers multiplied by the number of people. We discuss a serial testing classification method in the following sections to reduce the total cost.

2.3.2 Logic Regression

An advantage of risk score modeling is that it uses all the biomarker (or diagnostic test) information, and can account for biomarker data that is either continuous or categorical; however, it does require that all the diagnostic tests be performed on all persons which can be expensive. On the other hand, serial testing algorithms can be cost effective because not all persons are necessarily evaluated with all diagnostic tests. Given a set of tests and a condition that we seek to classify, our problem is finding the right classification algorithm that minimizes our loss function. We can attempt to search through the entire space of combinations of predictors, thresholds and binary operators, but the problem becomes intractable with an increasing number of predictors and thresholds. Instead, we use an approach that was developed by Ruczinski et al. (2003) to find a set of optimal rules using a combination of biomarkers. We employ a classification algorithm to generate optimal classification rules called logic regression.

Logic regression is a regression methodology that can be used when the covariates in the data are binary or dichotomous (Ruczinski et al., 2003). The goal of logic regression is to find predictors that can be combined using boolean operators to model the response variable. Given $B_1, B_2, ..., B_k$ binary predictors, and

a response variable Y, logic regression fits models of the form

$$g(E[Y]) = \beta_0 + \beta_1 M_1 + ... + \beta_n M_n,$$

where M_j is a boolean expression of the predictors. For example, $M_1 = B_1 \wedge B_2 \vee B_3$. In our case, the binary predictors are all of the tests generated from the biomarkers and thresholds i.e. $B_{1j} = (X_1 < c_{1j})$.

Since a binary outcome can be fully expressed with M_j, the link function $g(\cdot)$ is the identity, $g(\cdot) = \cdot$, and the regression model collapses into

$$E[D] = +1 \cdot M_1.$$

Because the space of classification rules is computationally intractable to search over, logic regression minimizes the loss function $L(\cdot)$ through a iterative heuristic procedure called simulated annealing (Ruczinski et al., 2003).

We apply logic regression to combining multiple continuous biomarkers in the following manner. We begin with m continuous biomarkers. We choose a set of k thresholds $\{c_{1j}, c_{2j}, ..., c_{kj}\}$ for each biomarker j. Then we form boolean predictors of biomarker j by forming k tests $(X_j < c_{1j}), (X_j < c_{2j}), ..., (X_j < c_{kj})$. The $k \times m$ tests of all the biomarkers are entered into a logic regression model. The logic regression algorithm then provides us with an optimal rule based on those predictors that minimizes the loss function L.

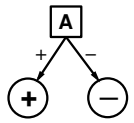
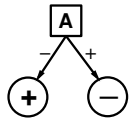
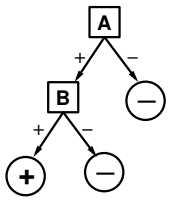
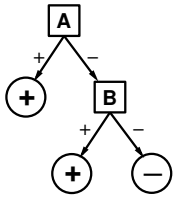
An example of a model from a logic regression procedure is

$$E[Y] = +1 \cdot (B_1 \wedge B_2 \vee B_3) \tag{2.4}$$

where $B_1 = (X_1 < c_1), B_2 = (X_2 < c_2), B_3 = (X_3 < c_3)$

This model fits into one of the serial testing algorithms we have considered, namely, figure 2.1(e) if we map B_1 to A, B_2 to B and B_3 to C. The cost savings in this particular model occur when B_1 evaluates to false. According to the logic given by equation 2.4, if B_1 evaluates to false, there is no need to evaluate B_2 and B_3. No matter what B_2 and B_3 evaluate to, $E[Y]$ will be false.

Table 2.1: Translation table of a logical rule to a serial testing rule. Here the binary variables A and B represent a diagnostic test or another logical rule. The table is applied recursively to a logical rule by order of precedence.

Logical Rule	Decision Tree
(A)	
not A	
$A \wedge B$	
$A \vee B$	

As a result, in the context of biomarkers, X_1, X_2, X_3, with respective costs r_1, r_2, r_3 of obtaining the biomarker, if an optimal rule follows the model given by equation 2.4, all subjects with $(X_1 > c_1)$ only need a fraction of the total cost $r_1/(r_1+r_2+r_3)$ to make a diagnosis. This may be even more apparent in graphical representations of rules given by figure 2.1. Any $\oplus$ or $\ominus$ circle above the bottom level of the classification tree represents a savings in diagnostic cost versus testing with all biomarkers. We provide a translation table for logical rules to a serial testing rule in Table 2.1.

We incorporate diagnostic cost into logic regression by extending the method to look at permutations of equivalent logic rules. Logic rules have well known commutative properties over $\wedge$ and $\vee$. We exploit this to reorder the serial testing

algorithm to change the diagnostic costs for a given algorithm. By performing swaps of the logical operands of $\wedge$ and $\vee$, we find equivalent classification rules that assign the same classification to the same predictors but at different costs. This is illustrated with an example in Figure 2.2. Among the set of equivalent logic rules $\mathbf{R}$, we find the permutation that minimizes the classification cost in the training sample.

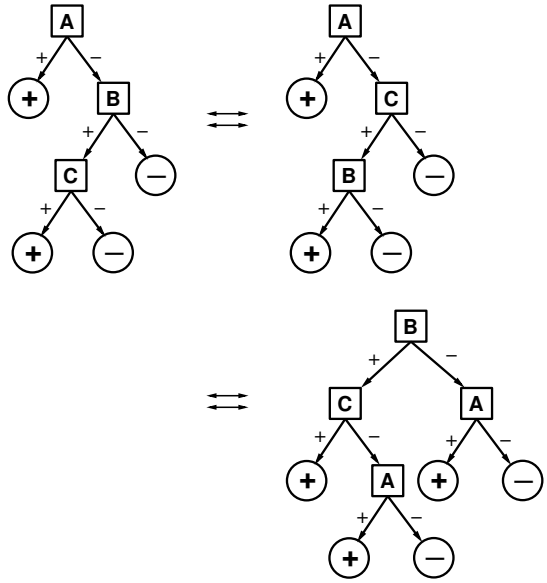


Figure 2.2: Illustration of swaps of logical operands in a decision tree. The tree in the left hand column represents the rule $A \vee (B \wedge C)$. From this rule, logically equivalent rules $A \vee (C \wedge B)$ and $(C \wedge B) \vee A$ are spawned.

In order to compute $L_r(p, cost)$, each rule considered during the simulated annealing portion of the logic regression procedure needs to perform the above procedure to find the minimal cost of the proposed rule. While this method will work in the logic regression framework, this procedure is computationally intensive and quickly becomes intractable with the large set of predictors. For reasons of computation, we approximate the optimal classification rule r^* which minimizes $L_r(\mathbf{p}, cost)$ with the optimal rules r that minimize $L(p)$. Among those rules, we find the permutation which minimizes cost using the method as described above.

2.3.3 Serial Risk Score Algorithm

2.3.3.1 Overview

The disadvantage of a serial testing rule when compared with risk score modeling is that there is a restriction on the type of rules you can create with logical operators, biomarkers and thresholds especially with limits on the complexity of the rule. A rule based on a linear combination of two biomarkers, for instance, would not be available. As an illustration, consider a two dimensional plotting of biomarkers X_1 and X_2. All combinations of logical rules would generate rectangular regions that represent positive classification. If the true classification is $(X_1 < X_2)$, it would be difficult this simple rule without a fairly complex model. Figure 2.3 illustrates this example.

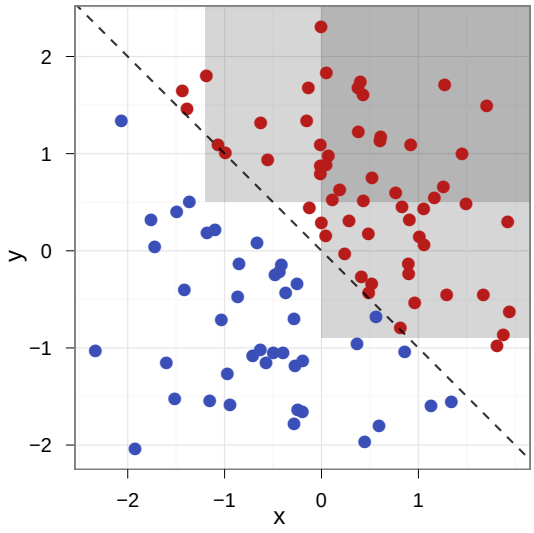


Figure 2.3: Illustration plot where a logical rule may fail. Two biomarkers X and Y, the logical rules form rectangles in the biomarker space, but the true classification is formed from a linear combination.

Here we outline a strategy that combines aspects of a risk score modeling approach and a serial testing approach to use all available information while min-

imizing costs. We call the method serial risk score classification. Our approach introduces several stages of classification where a person may either be classified as positive for disease, negative for disease, or neutral (i.e. undecided) when there is not enough evidence for either a positive or negative classification. Neutral classification was originally suggested to keep misclassification rates under a specified level (Rao, 1947) and has also been described in engineering literature as a *reject*, an event when a recognition system withholds its recognition decision (Chow, 1970). Most recently, Jeske et al. (2007) formalized a framework for a Bayesian neutral zone classifier using posterior class probabilities to define a neutral zone.

A general description of our strategy is as follows. Suppose we have a total of M different diagnostic tests that can be used for disease classification. At the first stage of diagnostic testing, each person is evaluated with m_1 diagnostic tests. The m_1 diagnostic tests are a subset of the M diagnostic tests. Based on those m_1 test results, we classify each person into one of three categories: positive for disease ($\widehat{Y} = +1$), negative for disease ($\widehat{Y} = -1$), or neutral (undecided; $\widehat{Y} = \varnothing$). Persons classified as neutral proceed to the second stage. At the second stage, an additional diagnostic test is performed on persons who were classified neutral at the first stage. We then reclassify those neutral persons from the first stage as either positive, negative, or still neutral using all $m_1 + 1$ diagnostic test results available at the second stage (i.e, the m_1 tests from the first stage together with the additional test performed at the second stage). We proceed similarly through each stage. In general, at the i^{th} stage, we reclassify those persons classified as neutral at the $(i-1)$ stage using all the m_i diagnostic tests that are available (i.e, m_{i-1} tests from the preceding $(i-1)$ stages plus the additional test performed at the i^{th} stage). The classification at the i^{th} stage is based on risk scores, determined, for example, from logistic regression. Risk score thresholds are determined at each stage, such that if the score is below (above) a lower (upper) threshold, the person is classified as negative (positive) for disease, and otherwise the person

is classified as neutral. We allow a maximum of K stages where $K \leq M$. All persons are classified at or before the K^{th} stage because we do not allow for neutral classification at the last K^{th} stage. The serial risk score classification approach can be less costly than testing all persons with all M diagnostic tests because some persons are classified as positive or negative before undergoing all M diagnostic tests.s

The strategy described above define a serial testing algorithm. An example illustration of the strategy is given in Figure 2.4. Figure 2.4 is an illustration of our strategy with 4 stages and 4 diagnostic tests which are labeled z_1, z_2, z_3 and z_4. At the first stage, a single diagnostic test z_1 is performed ($m_1 = 1$). Risk scores $S(z_1)$ are estimated from a risk score model (e.g., logistic regression) and persons are classified as negative if their scores are below a lower threshold c_{1l}, as positive if their risk scores are above an upper threshold c_{1u}, or as neutral if their scores lie between the lower and upper thresholds. At the second stage, an additional diagnostic test z_2 is performed only on those persons who were classified as neutral at the first stage. These persons are assigned updated risk scores $S(z_1, z_2)$ from a risk score model using both diagnostic tests z_1 and z_2 and they are then reclassified (using these updated risk scores) based on updated thresholds c_{2l} and c_{u2}. Similarly, at the third stage, an additional diagnostic test z_3 is performed on those persons still neutral at the second stage. At the fourth and final stage, the last diagnostic test z_4 is performed on those persons neutral at the third stage, and a classification (positive or negative) is finally made for these persons.

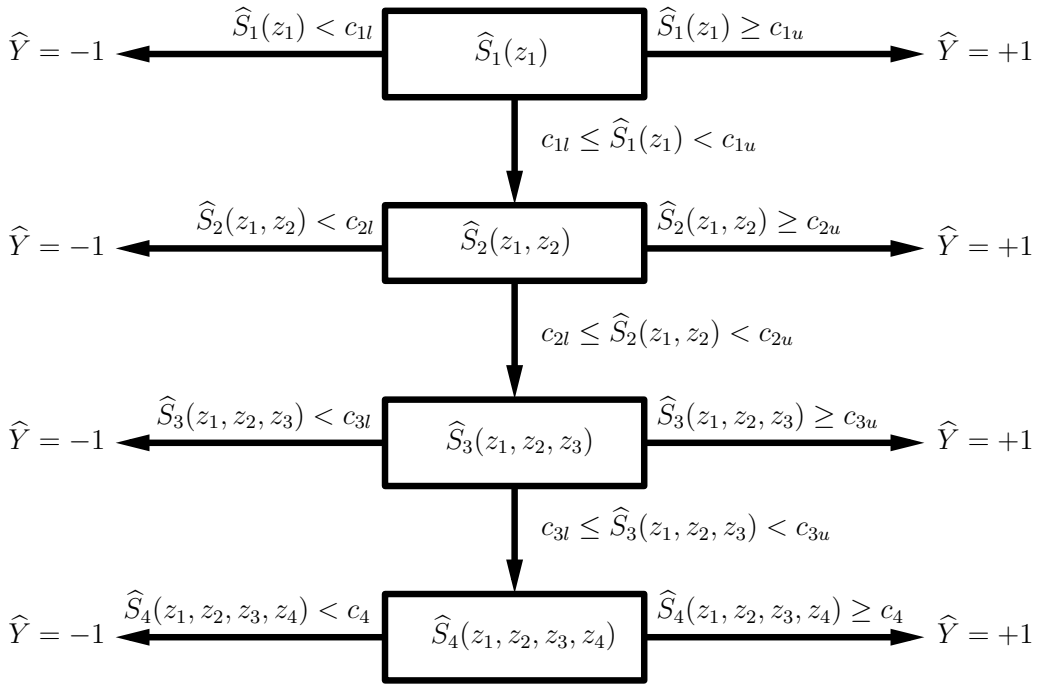


Figure 2.4: Illustration of serial risk score classifier with a maximum of $K = 4$ stages. At each level of the classifier, a risk score model is estimated and thresholds are defined. The threshold values determine whether an observation continues onto further testing or is classified as positive or negative. Four diagnostic tests are shown which are labeled z_1, z_2, z_3 and z_4.

2.3.4 Methods and Estimation of a Serial Risk Score Algorithm

In this section, we detail and formalize the approach outlined in 2.3.3.1. We assume we have available a training dataset that includes Y and all M diagnostic test results for each person in the dataset. In this section, we describe the methods for determining the risk score thresholds and the sequential order to add particular diagnostic tests at each successive stage.

Suppose we have a training data set of N subjects, of whom n_+ have disease ($Y = +1$) and n_- do not have disease ($Y = -1$). The total number of diagnostic tests is M which are named $z_1, ..., z_M$. Each of the M diagnostic tests are performed on each person. Let $\mathbf{X}_i$ be a vector of m_i diagnostic tests available at the ith stage for risk score modeling. The vector $\mathbf{X}_i$ includes all the diagnostic tests in $\mathbf{X}_{i-1}$ plus one additional diagnostic test that is performed at stage i, and thus $m_i = m_{i-1} + 1$.

To be clear, the tests included in earlier stages are available for analysis at later stages, that is, the diagnostic tests in $\mathbf{X}_i$ are a subset of the diagnostic tests in $\mathbf{X}_j$ for $i < j$. For example, in figure 2.4, $\mathbf{X}_1 = (z_1)$, $\mathbf{X}_2 = (z_1, z_2)$, $\mathbf{X}_3 = (z_1, z_2, z_3)$, and $\mathbf{X}_4 = (z_1, z_2, z_3, z_4)$. The $\mathbf{X}_i$ vectors are not fixed or predetermined. Rather, we use our methods described below to determine the sequential order of tests to perform at each stage that balances accuracy and cost considerations.

At the ith stage, let $\widehat{S}_i(\mathbf{X}_i)$ be a model for the risk score, $P(Y = +1|\mathbf{X}_i)$. For example, when using a logistic regression model, we have $\widehat{S}_i(\mathbf{X}_i) = 1/\{1 + \exp(-\mathbf{X}_i^T \boldsymbol{\beta}_i)\}$ where $\boldsymbol{\beta}_i$ are estimated regression coefficients.

For risk score models $\widehat{S}_i(\mathbf{X}_i)$, $1 \leq i < k-1$, we divide the risk score space into "positive", "negative", and "neutral" regions using the lower and upper thresholds

c_{il} and c_{iu}, where $c_{il} \leq c_{iu}$, and both lie between 0 and 1, so that

$$\widehat{Y}_i = \begin{cases} -1, & \text{if } \widehat{S}_i(\mathbf{X}_i) < c_{il} \\ \varnothing & \text{if } c_{il} \leq \widehat{S}_i(\mathbf{X}_i) < c_{iu} \\ +1, & \text{if } \widehat{S}_i(\mathbf{X}_i) \geq c_{iu} \end{cases}$$

where 0 indicates neutral or no classification in stage i. At the last stage, we do not assign a neutral zone and use a single threshold c_k to divide the risk score space, i.e.

$$\widehat{Y}_k = \begin{cases} -1, & \text{if } \widehat{S}_k(\mathbf{X}_k) < c_k \\ +1, & \text{if } \widehat{S}_k(\mathbf{X}_k) \geq c_k. \end{cases}$$

For a person assigned to the neutral zone in the first $j-1$ stages, the final classification of that individual is $\widehat{Y}_j$.

We now present a formal definition for a *serial risk score classification algorithm*. It is a classification algorithm where the prediction $\widehat{Y}$ ($\widehat{Y} = 1$ if classified positive for disease and -1 otherwise) is based on a logical rule r (or algorithm) of the following form (where $\wedge$ is a logical AND operator and $\vee$ is a logical OR operator):

$$\begin{aligned} r = &\{\widehat{S}_1(\mathbf{X}_1) \geq c_{1u}\} \\ &\vee [\{\widehat{S}_2(\mathbf{X}_2) \geq c_{2u}\} \wedge \{\widehat{S}_1(\mathbf{X}_1) \geq c_{1l}\}] \\ &\vee [\{\widehat{S}_3(\mathbf{X}_3) \geq c_{3u}\} \wedge \{\widehat{S}_1(\mathbf{X}_1) \geq c_{1l}\} \wedge \{\widehat{S}_2(\mathbf{X}_2) \geq c_{2l}\}] \\ &\vdots \\ &\vee [\{\widehat{S}_k(\mathbf{X}_k) \geq c_k\} \wedge \{\widehat{S}_1(\mathbf{X}_1) \geq c_{1l}\} \wedge \cdots \wedge \{\widehat{S}_{k-1}(\mathbf{X}_{k-1}) \geq c_{(k-1)l}\}] \end{aligned} \quad (2.5)$$

where

$$\widehat{Y} = \begin{cases} +1, & \text{if } r = true \\ -1, & \text{if } r = false. \end{cases}$$

In equation 2.5, the first line refers to persons classified as positive at the end of stage 1; the second line refers to persons neutral at the end of stage 1 but

subsequently classified as positive at the end of stage 2; and so on. The set of algorithms given by (2.5) define a space of algorithms $\mathbf{R}$ which is generated by the various sequential orders of adding diagnostic tests and choices for the risk score thresholds.

For a specified value of p, we could simply find the minimal value, $L_{min}(p)$, of the loss function that is achievable with any algorithms in the space of algorithms $\mathbf{R}$. However we also want to balance accuracy with cost considerations. The algorithm that corresponds to $L_{min}(p)$ may not have the lowest total cost. To help achieve balance between both cost and accuracy considerations, we adopt the following strategy. We find the collection of algorithms that have accuracy loss functions "close" to the minimal value. Among these algorithms that are "nearly" equivalent with regard to accuracy, we find the one with minimal total cost. To quantify the meaning of "close" with respect to accuracy, we introduce a tolerance t on the accuracy loss function. Consider the set of algorithms $\mathbf{R}(t, p)$ that give values of the loss function within a percentage $t \times 100\%$ of $L_{min}(p)$ so that

$$\mathbf{R}(t, p) = \{r : L_{min}(p) \leq L(r, p) < (1 + t) \cdot L_{min}(p)\}.$$

That is, $\mathbf{R}(t, p)$ is a set of algorithms that are nearly equivalent with respect to accuracy as calibrated by the tolerance t. We define an optimal algorithm r_p as an algorithm in the set $\mathbf{R}(t, p)$ with minimal total cost. It is possible that multiple algorithms satisfy this optimality criteria. For a chosen set of $p \in \mathbf{P}$, these optimal algorithms form a subset of $\mathbf{R}$, $\mathbf{R_P}(t) = \{r_p : \exists\ p \in \mathbf{P}, r_p = \arg\min_{cost} \mathbf{R}(t, p)\}$. The set $\mathbf{R_P}(t)$ is used to construct a ROC curve. As t decreases more weight is given to accuracy compared to cost for determining the optimal algorithm.

The space of algorithms $\mathbf{R}$ includes an algorithm of performing all M diagnostic tests at the first stage and not allowing any person to be classified as neutral, that is, $m_1 = M$ and $c_{1l} = c_{1u}$. Such an algorithm is equivalent to a risk score model using all diagnostic tests. Thus, if the tolerance t is set to 0, then the

serial risk score classification algorithm can achieve the same accuracy as standard logistic regression. If there is another algorithm which has a lower or equal value of the accuracy loss function as that achieved by logistic regression using all diagnostic tests and has lower total cost, then the serial risk score classification algorithm will be as accurate but less expensive.

2.3.5 Evaluation of Methods

2.3.5.1 Bootstrapping misclassification rates and cost

Here we describe a method for bootstrapping the sensitivity (Se), and specificity (Sp) in order to characterize the prediction ability of the algorithm with a future unobserved subject. When the same data is used to both construct and evaluate an algorithm, the apparent sensitivity and specificity can tend to be over optimistic (Efron, 1983). The reason is that the classifiers are chosen specifically to minimize the loss function with respect to the training data. In this section, we describe a bootstrapping method to estimate this optimism, or bias, and how we adjust for it.

In the following discussion, as it is understood that Se and Sp will depend on a particular p and optimal algorithm r_p, we notationally omit this dependency for brevity.

We define the biases in the apparent sensitivity and specificity respectively by $e_{se} = \overline{Se} - Se$ with expectation ϵ_{se}, and $e_{sp} = \overline{Sp} - Sp$ with expectation ϵ_{sp}. If ϵ_{se} was known, then $\widehat{Se} = \overline{Se} - \epsilon_{se}$ would be an unbiased estimate for Se. Similarly if ϵ_{se} was known, $\widehat{Sp} = \overline{Sp} - \epsilon_{sp}$ is an unbiased estimate for Sp as well.

Our strategy is to estimate ϵ_{se} and ϵ_{sp} using a bootstrap method. Once an estimate is found, we can adjust our apparent estimates by the errors to reduce the bias incurred from overfitting. The diagnostic costs can be adjusted by the bootstrap error in the same fashion. Let $\overline{Cost}$ be the apparent total cost of an

algorithm found from (2.2.2). The error of the total cost is $e_{cost} = Cost - \overline{Cost}$ where $Cost$ is the true cost for an optimal algorithm with some p. If we let ϵ_{cost} be the expectation of this error, then the adjusted cost $\widehat{Cost} = \overline{Cost} + \epsilon_{cost}$.

For a training data set $T = \{Y, \mathbf{X}\}$ where $\mathbf{X}$ is a $N \times M$ matrix of diagnostic tests and a chosen p, the bootstrap procedure is described below.

1. Draw a bootstrap sample T^* from T with replacement.

2. Use the serial risk score classification method to find the optimal algorithm r_p^* from T^*.

3. Use the optimal algorithm r_p^*, applied to the bootstrap sample, T^*, to calculate the apparent sensitivity, specificity and cost, which we denote respectively by, $\overline{Se}(r_p^*|T^*)$, $\overline{Sp}(r_p^*|T^*)$, $\overline{Cost}(r_p^*|T^*)$.

4. Use the optimal algorithm r_p^*, applied to the original training data set, T, to calculate the sensitivity, specificity, and cost, which we denote respectively by, $\overline{Se}(r_p^*|T)$, $\overline{Sp}(r_p^*|T)$, $\overline{Cost}(r_p^*|T)$.

5. Calculate $e_{sp}^* = \overline{Se}(r_p^*|T^*) - \overline{Se}(r_p^*|T)$, $e_{sp}^* = \overline{Sp}(r_p^*|T^*) - \overline{Sp}(r_p^*|T)$, $e_{cost}^* = \overline{Cost}(r_p^*|T^*) - \overline{Cost}(r_p^*|T)$.

6. Repeat 1 - 5 for B times. Estimate $\widehat{\epsilon}_{se} = 1/B \sum_{b=1}^{B} e_{se_b}^*$, $\widehat{\epsilon}_{sp} = 1/B \sum_{b=1}^{B} e_{sp_b}^*$, $\widehat{\epsilon}_{cost} = 1/B \sum_{b=1}^{B} e_{cost_b}^*$.

Following the bootstrap procedure, we estimate the adjusted sensitivity $\widehat{Se} = \overline{Se} - \widehat{\epsilon}_{se}$, specificity $\widehat{Sp} = \overline{Sp} - \widehat{\epsilon}_{sp}$, and cost, $\widehat{Cost} = \overline{Cost} + \epsilon_{cost}$.

The bootstrap samples allow the construction of a $(1-2\alpha)\%$ confidence interval for Se and Sp. For each p, we find the interval $(\widehat{Se}_{\alpha}, \widehat{Se}_{1-\alpha})$ where $\widehat{Se}_{\alpha}$ and $\widehat{Se}_{1-\alpha}$ are the α and $1-\alpha$ percentiles of the empirical distribution of the bootstrapped sensitivity (Efron and Tibshirani, 1993). The percentile-based bootstrap confidence interval is constructed similarly for Sp.

2.3.5.2 Simulation Procedure

We evaluated the performance of the logic regression and serial risk score classification method and bootstrap procedures through a series of simulations. Each simulation was comprised of 1000 simulated datasets. In each dataset, we generated data for 500 subjects with two diagnostic markers z_1 and z_2 from a truncated bivariate normal distribution where $\mu = [2, 9]^T$ and Σ is set so that z_1 and z_2 have a standard deviation of 1 and 3 respectively and a correlation of ρ. We modeled the outcome Y using a logistic regression model.

We considered two situations for the correlation coefficient ρ between the two markers: $\rho = 0.5$ and $\rho = 0.0$. We also considered 2 cases for the coefficients in the logistic regression model: (Model I) where the coefficients β_1 and β_2 are chosen so that a 1 standard deviation increase in z_1 doubles the risk of $Y = 1$ and 1 standard deviation increase in z_2 triples the risk; and (Model II) where 1 standard deviation increase in z_2 triples the risk and $\beta_1 = 0$. Lastly, we assumed the cost of z_2 was 10 times that of z_1.

The resulting values for each simulation are averaged over the 1000 simulations to generate the Monte Carlo estimates for sensitivity, specificity and average cost savings for each classification method and for each specified value of p in the loss function $L(r, p) = pFP(r) + FN(r)$.

We computed bootstrap adjusted estimates for both the logic regression method and serial risk score classification using 500 bootstrap samples. We also compared the sensitivity and specificity estimates of each classification method as determined by our bootstrapping procedure with an independent data set of 500 values drawn from the same distribution for each simulation.

2.3.5.3 Simulation Results

We present the results for correlated diagnostic markers in Tables 2.2 and 2.3. In each simulation, we compared the logic regression and serial risk score classification method to logistic regression with backward elimination (non-significant terms at the $\alpha = 0.05$ level were eliminated).

Table 2.2 corresponds to the (Model I) logistic model specified in section 2.3.5.2. We find that both serial testing methods presented have slightly diminished accuracy performance as measured by the AUC (area under curve) compared to logistic regression, but the cost savings are substantial. This makes intuitive sense because both diagnostic markers have information about disease risk, so both diagnostic markers are not necessary to make a determination. While z_2 has a stronger effect size on the disease risk than z_1, we find z_1 can determine classifications on its own without diminishing the sensitivity and specificity of the algorithm. The serial risk score algorithms show greater average cost savings when compared to logic regression algorithms while maintaining a practically equivalent AUC.

In Table 2.3, the results of simulation under the (Model II) logistic model is presented. We find smaller cost savings and accuracy performance compared to (Model I) indicating a larger proportion of the sample using both tests for classification. There are still significant cost savings in (Model II) because of the correlation between the two diagnostic tests. The correlation allows the less expensive diagnostic test to be used as a proxy for the significantly more expensive test. Some information from one diagnostic test about disease risk is included in the other, and thus the less expensive diagnostic test can take the place of the more expensive one to some extent.

Table 2.4 and 2.5 presents the results for a simulation using uncorrelated diagnostic tests ($\rho = 0.0$) for logistic models (Model I) and (Model II) respectively.

In (Model I), we find a markedly reduced cost savings with uncorrelated (Table 2.4) compared to correlated diagnostic tests (Table 2.2) with a minor reduction in accuracy. Because the diagnostic tests are no longer correlated, both tests are more often necessary for correct classification.

In (Model II), the cost savings are further reduced, and only very minor cost savings are achieved. We note that at large values of p, we show dramatic cost savings for the serial risk score algorithm. This may indicate an improper random determination of classification for a large portion of the sample. It should be noted that the cost savings we report in this simulation are dependent on the disease prevalence and different results will be obtained with different prevalence rates.

The bootstrap adjusted SRS estimates of the performance measures appear in good agreement with the estimates found when the algorithms are applied to the independent data sets. We believe this demonstrates the bias reduction from the bootstrap procedure is effective. We also performed the simulation reported in section 2.3.5.2 with only 100 bootstraps and obtained very similar results as that reported here with 500 bootstraps. When the serial risk score algorithm is applied to data where the diagnostic tests have no discrimination ability, we find an AUC of .50 as expected.

The serial risk score classification procedure was performed at different levels of tolerance ($t = 0.0, 0.05$, and 0.10). We find the tolerance t calibrates the trade-off between accuracy and cost. Increasing t puts more priority on decreasing costs at the expense of accuracy. A comparison table for the different values of t is presented in Appendix A.

Table 2.2: Monte Carlo simulation[1] estimates for sensitivity (sens.), specificity (spec.), and the average cost savings versus logistic regression (ACS) for classification algorithms generated from logistic regression, logic regression and serial risk score classification. Here we assume correlated biomarkers ($\rho = .5$) and model parameters $\beta_1 = .69, \beta_2 = .37$

	Logistic Regression		Logic Regression							Serial Risk Score[3]						
			Apparent		Validation		Bootstrap [2]			Apparent		Validation		Bootstrap [2]		
p	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS
0.03	0.99	0.28	1.00	0.34	0.94	0.33	0.95	0.33	44%	0.99	0.27	0.96	0.26	0.97	0.26	46%
0.07	0.94	0.47	0.96	0.49	0.89	0.47	0.90	0.47	48%	0.95	0.44	0.91	0.43	0.92	0.44	50%
0.10	0.91	0.56	0.93	0.57	0.86	0.56	0.86	0.56	49%	0.91	0.53	0.87	0.53	0.87	0.53	51%
0.30	0.69	0.82	0.72	0.82	0.64	0.81	0.64	0.81	48%	0.66	0.82	0.61	0.81	0.62	0.81	58%
0.70	0.44	0.94	0.47	0.94	0.40	0.93	0.39	0.93	36%	0.37	0.95	0.34	0.94	0.33	0.94	72%
1.00	0.34	0.97	0.36	0.97	0.30	0.95	0.29	0.96	30%	0.27	0.97	0.24	0.97	0.24	0.97	77%
AUC =	0.84		0.86		0.79		0.79			0.82		0.78		0.79		

[1] On each simulation we generated 500 values $\mathbf{Z} = [z_1, z_2]^T \sim$ truncated $MVN(\mu, \Sigma)$, where $\mu = [2, 9]^T$ and Σ is set so that z_1 and z_2 have a standard deviation of 1 and 3 respectively and a correlation of .5. The distribution is truncated using rejection sampling so that $z_1 > 0,$ and $z_2 > 0$. The outcome $Y \sim \text{Bern}(\pi)$ where $\pi = 1/(1 + exp(-(\beta_0 + \beta_1 z_1 + \beta_2 z_2))$. The empirical proportion where $Y = 1$ was 17.6%. Parameters β_1 and β_2 chosen so that a 1 standard deviation increase in β_1 doubles the risk and 1 standard deviation increase in β_2 triples the risk.

[2] Bootstrap adjusted estimates details are based on methods described in Section 2.3.5.1 with 500 bootstraps.

[3] Serial risk score classification algorithms generated with tolerance $t = 10\%$.

Table 2.3: Monte Carlo simulation[1] estimates for sensitivity (sens.), specificity (spec.), and the average cost savings versus logistic regression (ACS) for classification algorithms generated from logistic regression, logic regression and serial risk score classification. Here we assume correlated biomarkers ($\rho = .5$) and model parameters $\beta_1 = 0, \beta_2 = .37$

	Logistic Regression		Logic Regression							Serial Risk Score[3]						
			Apparent		Validation		Bootstrap [2]			Apparent		Validation		Bootstrap [2]		
p	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS
0.03	0.99	0.20	1.00	0.28	0.92	0.27	0.94	0.27	31%	0.99	0.17	0.96	0.16	0.97	0.16	39%
0.07	0.93	0.41	0.95	0.44	0.86	0.42	0.87	0.42	29%	0.93	0.37	0.88	0.36	0.89	0.37	34%
0.10	0.86	0.53	0.89	0.55	0.79	0.53	0.80	0.53	28%	0.87	0.50	0.81	0.49	0.82	0.49	30%
0.30	0.52	0.86	0.55	0.86	0.45	0.85	0.45	0.85	27%	0.45	0.87	0.39	0.87	0.40	0.87	45%
0.70	0.24	0.97	0.27	0.97	0.19	0.96	0.19	0.96	31%	0.14	0.98	0.11	0.98	0.10	0.98	77%
1.00	0.15	0.99	0.19	0.98	0.13	0.97	0.13	0.97	38%	0.06	0.99	0.05	0.99	0.03	0.99	84%
AUC =	0.78		0.81		0.72		0.73			0.76		0.7		0.71		

[1] On each simulation we generated 500 values $\mathbf{Z} = [z_1, z_2]^T \sim$ truncated $MVN(\mu, \Sigma)$, where $\mu = [2, 9]^T$ and Σ is set so that z_1 and z_2 have a standard deviation of 1 and 3 respectively and a correlation of .5. The distribution is truncated using rejection sampling so that $z_1 > 0,$ and $z_2 > 0$. The outcome $Y \sim \text{Bern}(\pi)$ where $\pi = 1/(1 + exp(-(\beta_0 + \beta_1 z_1 + \beta_2 z_2))$. The empirical proportion where $Y = 1$ was 14.2%. Parameter β_2 chosen so that a 1 standard deviation increase in β_2 triples the risk.

[2] Bootstrap adjusted estimates details are based on methods described in Section 2.3.5.1 with 500 bootstraps.

[3] Serial risk score classification algorithms generated with tolerance $t = 10\%$.

Table 2.4: Monte Carlo simulation[1] estimates for sensitivity (sens.), specificity (spec.), and the average cost savings versus logistic regression (ACS) for classification algorithms generated from logistic regression, logic regression and serial risk score classification. Here we assume uncorrelated biomarkers ($\rho = 0$) and model parameters $\beta_1 = .69, \beta_2 = .37$

	Logistic Regression		Logic Regression							Serial Risk Score[3]						
			Apparent		Validation		Bootstrap [2]			Apparent		Validation		Bootstrap [2]		
p	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS
0.03	0.99	0.24	1.00	0.29	0.94	0.28	0.95	0.29	42%	0.99	0.19	0.97	0.19	0.97	0.19	38%
0.07	0.94	0.43	0.96	0.45	0.88	0.43	0.89	0.44	46%	0.94	0.38	0.90	0.37	0.91	0.37	34%
0.10	0.88	0.54	0.91	0.54	0.82	0.53	0.83	0.53	48%	0.90	0.49	0.85	0.48	0.85	0.48	32%
0.30	0.60	0.84	0.62	0.84	0.53	0.83	0.53	0.83	47%	0.55	0.85	0.49	0.84	0.50	0.84	41%
0.70	0.33	0.96	0.34	0.96	0.27	0.95	0.27	0.95	39%	0.22	0.97	0.18	0.96	0.18	0.96	69%
1.00	0.23	0.98	0.25	0.98	0.19	0.97	0.19	0.97	38%	0.13	0.99	0.11	0.98	0.10	0.98	77%
AUC =	0.81		0.83		0.75		0.75			0.78		0.74		0.74		

[1] On each simulation we generated 500 values $\mathbf{Z} = [z_1, z_2]^T \sim$ truncated $MVN(\mu, \Sigma)$, where $\mu = [2, 9]^T$ and Σ is set so that z_1 and z_2 have a standard deviation of 1 and 3 respectively and zero correlation. The distribution is truncated using rejection sampling so that $z_1 > 0,$ and $z_2 > 0$. The outcome $Y \sim \text{Bern}(\pi)$ where $\pi = 1/(1 + exp(-(\beta_0 + \beta_1 z_1 + \beta_2 z_2))$. The empirical proportion where $Y = 1$ was 15.7%. Parameters β_1 and β_2 chosen so that a 1 standard deviation increase in β_1 doubles the risk and 1 standard deviation increase in β_2 triples the risk.

[2] Bootstrap adjusted estimates details are based on methods described in Section 2.3.5.1 with 500 bootstraps.

[3] Serial risk score classification algorithms generated with tolerance $t = 10\%$.

Table 2.5: Monte Carlo simulation[1] estimates for sensitivity (sens.), specificity (spec.), and the average cost savings versus logistic regression (ACS) for classification algorithms generated from logistic regression, logic regression and serial risk score classification. Here we assume uncorrelated biomarkers ($\rho = 0$) and model parameters $\beta_1 = 0, \beta_2 = .37$

	Logistic Regression		Logic Regression							Serial Risk Score[3]						
			Apparent		Validation		Bootstrap [2]			Apparent		Validation		Bootstrap [2]		
p	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS
0.03	0.99	0.21	1.00	0.28	0.93	0.27	0.94	0.27	28%	0.99	0.16	0.97	0.16	0.97	0.16	32%
0.07	0.92	0.41	0.95	0.44	0.86	0.42	0.87	0.43	21%	0.93	0.37	0.88	0.37	0.89	0.37	20%
0.10	0.86	0.53	0.89	0.55	0.79	0.54	0.80	0.54	18%	0.86	0.51	0.81	0.50	0.81	0.51	15%
0.30	0.51	0.87	0.54	0.87	0.44	0.85	0.44	0.85	22%	0.45	0.88	0.39	0.87	0.39	0.87	23%
0.70	0.23	0.97	0.27	0.97	0.19	0.96	0.19	0.96	40%	0.13	0.98	0.10	0.98	0.09	0.98	64%
1.00	0.15	0.99	0.19	0.98	0.13	0.97	0.13	0.97	49%	0.07	0.99	0.05	0.99	0.03	0.99	75%
AUC =	0.78		0.81		0.72		0.73			0.76		0.71		0.71		

[1] On each simulation we generated 500 values $\mathbf{Z} = [z_1, z_2]^T \sim$ truncated $MVN(\mu, \Sigma)$, where $\mu = [2, 9]^T$ and Σ is set so that z_1 and z_2 have a standard deviation of 1 and 3 respectively and zero correlation. The distribution is truncated using rejection sampling so that $z_1 > 0,$ and $z_2 > 0$. The outcome $Y \sim \text{Bern}(\pi)$ where $\pi = 1/(1 + exp(-(\beta_0 + \beta_1 z_1 + \beta_2 z_2))$. The empirical proportion where $Y = 1$ was 14.0%. Parameter β_2 chosen so that a 1 standard deviation increase in β_2 triples the risk.

[2] Bootstrap adjusted estimates details are based on methods described in Section 2.3.5.1 with 500 bootstraps.

[3] Serial risk score classification algorithms generated with tolerance $t = 10\%$.

2.4 Application to HIV Data

2.4.1 Description of Data

The dataset consisted of 1782 samples from three cohort studies: a vaccine preparedness study (HIVNET001, men and women with different risk factors for HIV infection, (Celum et al., 2001)), a cohort study of intravenous drug users (the AIDS Linked to Intravenous Experience (ALIVE) cohort, (Vlahov et al., 1991)), and a cohort study of men who have sex with men (the Multicenter AIDS Cohort Study (MACS), (Kaslow et al., 1987)).

The data has repeated measures for each subject; however, for this analysis, we treated each as independent observations to study our classification procedures. We estimated seroconversion date using the midpoint between the time when a subject first tested positive and when the subject last tested negative. We excluded observations without all 4 biomarkers and which had a time between first positive test and last negative test of 1.5 years. Originally the data set contained 2106 observations, but the complete case analysis reduced the data set to 1782.

Recent infection is defined to be infected for < 1 year. We define chronic infection by as infected > 1 year. Only the HIVNET001 cohort had recently infected observations, but we use information from all cohorts to train the classification rules. The complete case data set contained 310 observations that were classified as "recent infection." (17.4% of the observations classified as recent.) The data provided the 4 biomarkers: CD4, Avidity, BED and Viral Load. The laboratory costs for performing the BED-CEIA, avidity, CD4, and viral load assays were, relative to the BED assay, 1,2,5 and 10 respectively. Table 2.6 presents the means and standard deviations of the biomarkers in the recently infected population and chronically infected population.

CD4 is the CD4 blood cell count in a cubic millimeter volume of blood. The

CD4 blood cell count will drop over the course of the disease. A normal uninfected individual will have CD4 blood cell counts between 500 to 1200. Figure 2.5(a) displays smoothed density curves of CD4 stratified by recent infection. CD4 is higher ($C\bar{D}4 = 610.4(228.8)$) on average for recently infected observations than observations with chronic infection ($C\bar{D}4 = 484.5(284.2)$), but the figure displays relatively poor separation from this biomarker alone.

Avidity is a measure of the strength of the binding between immunoglobulin G (IgG) antibodies and the corresponding antigen, a property that increases over a period of months in newly acquired infections. In past studies, avidity indices (AI) of ≤ 80 reproducibly identified seroconversion within the previous 142 days (Chawla et al., 2007). In figure 2.5(b), avidity indices for recently infected observations are spread relativity evenly across avidity indices from 10 to 100. Of the chronically infected observations, the majority of the avidity indices take on the value of 100. In figure 2.6(b), the pattern is more obvious; as time from seroconversion increases, the avidity values tend to take on avidity indices of around 100.

The BED (BED-CEIA) assay detects levels of anti-HIV IgG relative to total IgG and is based on the observation that the ratio of anti-HIV IgG to total IgG increases with time after HIV infection. It has been used with the serologic testing algorithm for recent HIV seroconversion (STARHS) to estimate incidence rates, but can overestimate recent infection (Hargrove et al., 2008). In figure 2.5(c), recently infected observations clearly show a lower BED values on average than chronically infected observations. Figure 2.6(c) shows the trend for BED to increase over time, and it appears to level out over 2 years from seroconversion.

Viral load measures the number of virus copies per ml in the blood sera. In recently infected individuals, viral load is high and gradually drops to an equilibrium level through the course of the disease if untreated in the natural course of infection. Onset of AIDS or other factors can make the viral load rise again

which can confound the usage of this biomarker for recent infection. In the data set, viral load ranges from 25 to almost 3 million with a median of 11220. We used a log transformation to alleviate the skewness of the variable.

Table 2.6: Biomarkers means and standard deviations of chronic (infected > 1 year) and recently (infected < 1 year) infected observations. Means and standard deviations calculated from complete case data only.

	Chronically Infected (n = 1472)	Recently Infected (n = 310)
Avidity	96.46 (11.51)	72.81 (28.73)
BED	2.03 (1.12)	1.17 (0.92)
CD4	484.50 (284.2)	610.40 (228.8)
log Viral Load	8.54 (2.86)	9.17 (2.33)

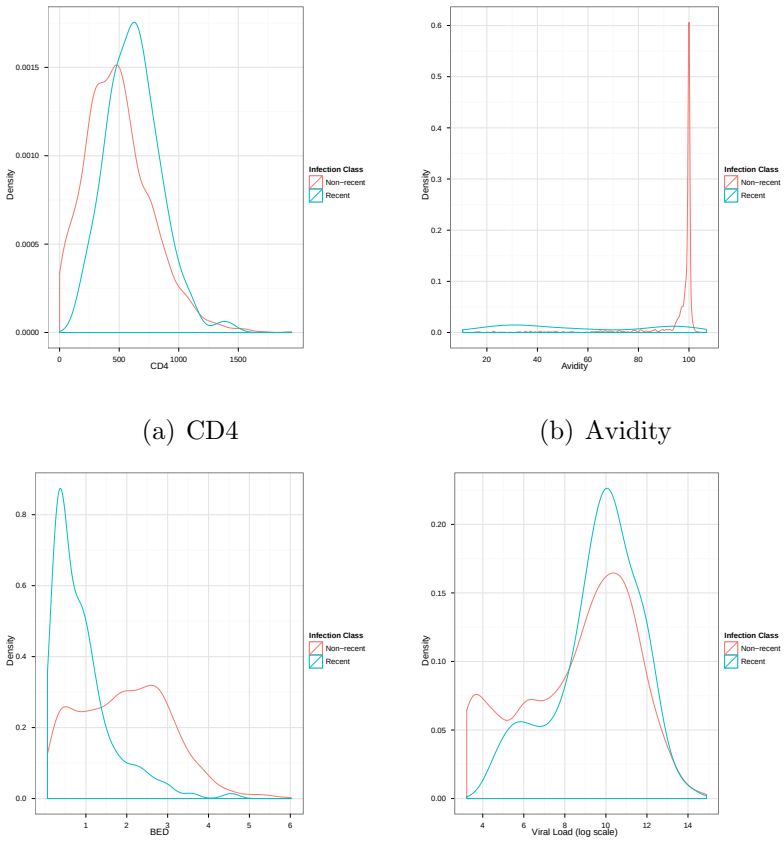
2.4.2 Application of Logistic Regression to HIV data

Using a risk score modeling approach, we fit, using maximum likelihood, the model

$$logit(RS(X)) = \beta_0 + \beta_1 \text{Avidity} + \beta_2 \text{CD4} + \beta_3 \text{BED} + \beta_4 log(\text{Viral Load}) \quad (2.6)$$

Using the dataset as described in the previous section (n = 1782), we present the estimated coefficients of the risk score model fitted by the `glm` function in R using a logit link function in Table 2.7. All of the biomarker coefficients are significant at the $\alpha = 0.05$ level. As expected, the coefficients for Avidity and BED are negative which suggests an increase in these predictors leads to a decrease in odds of being recently infected. Conversely, higher values of CD4 and viral load indicate higher odds of being recently infected.

For a selected value of p, an appropriate threshold c of the risk score $RS(X)$ that minimizes $L(p)$ is found to generate a classifier ($RS(X) > c$).



(a) CD4 (b) Avidity

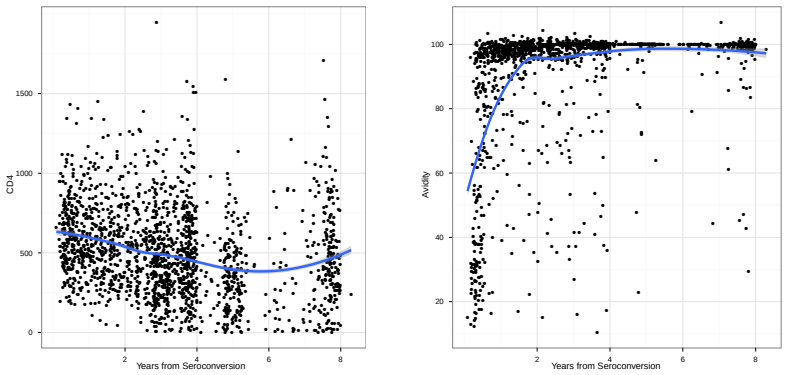
(c) BED (d) Viral Load

Figure 2.5: Estimated densities of the biomarkers used to predict recent infection using kernel smoothing. Densities of recent infections (infected < 1 yr.) in blue and chronic infections (infected > 1 yr.) in red, estimated separately.

2.4.3 Application of Logic Regression to HIV data

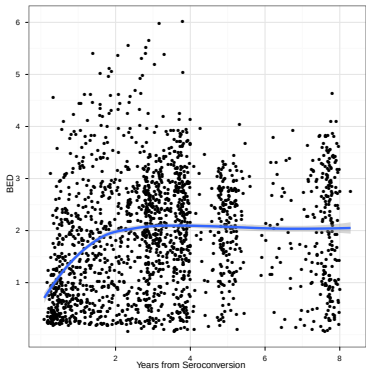
The thresholds in Table 2.8 were applied to the biomarkers to form the logical predictors in the logic regression routine. Logic regression models were limited to using a maximum of 4 "leaves" or predictors to avoid overfitting. Although this might seem too restrictive, the method still displays good discrimination ability. This restriction also allows a researcher or clinician to apply the rule more readily in the field.

In Table 2.9, we present the bootstrap adjusted sensitivity, specificity, ACS

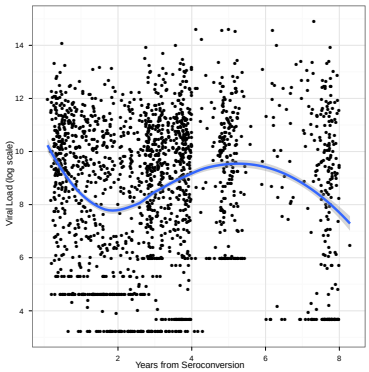


(a) CD4

(b) Avidity



(c) BED



(d) Viral Load

Figure 2.6: Scatter plot of each biomarker against time since infection among untreated individuals.
A smooth loess curve has been added.

and the generated logical rule for selected values of p. The generated logical rule is presented in the ordering that produced the minimal cost; that is, if read from right to left and using the parenthesis to indicate precedence. We can see from the tables that increasing p, the weight of false positives relative to false negatives, decreases the sensitivity of the classification rule, but increases the specificity.

The cost of classification varies between choices of the penalty factor but the classification algorithms provide an average cost savings of at least 69% versus using all biomarkers. The avidity biomarker is used first in 8 out of the 9 rules presented.

Table 2.7: Estimated coefficients of logistic risk score model given by (2.6). All biomarker coefficients are highly significant at $\alpha = .05$ level. Each biomarker has been scaled as shown in the first column.

	Estimate	Std. Error	Z score	p-value
(Intercept)	-0.83	0.49	-1.69	0.091
Avidity / 100	-4.46	0.42	-10.60	< 0.001
BED /10	-5.86	0.89	-6.60	< 0.001
CD4 /1000	2.26	0.29	7.86	< 0.001
log(Viral Load) /10	3.32	0.35	9.50	< 0.001

We present logic rule ROC curves for adjusted and unadjusted sensitivity and specificity from serial testing algorithms generated from the logic regression in figures 2.7. Each point on the ROC curve represents a different algorithm corresponding to a particular p selected by minimization of the loss $L(p)$. For each point, the 95% lower and upper confidence limits of $\widehat{Se}$, and $\widehat{Sp}$ are plotted to form the confidence intervals shown on the graph. The AUC of the bootstrap adjusted ROC curves presented is 0.87. We employed the bootstrap procedure described in Section 2.3.5.1 to compute adjusted estimates for sensitivity and specificity and cost using 500 bootstrap samples. We have compared results using 100 bootstraps and 500 bootstraps and similar results were obtained.

2.4.4 Application of Serial Risk Score to HIV data

The serial risk score classification method was applied to the HIV data described in section 2.4.1. Because of the computational constraints of brute force searching, we limited the risk score thresholds $\{c_{1l} \cdots c_{(k-1)l}\}$, $\{c_{1u} \cdots c_{(k-1)u}\}$, and c_k to using only the deciles of the risk score space at each stage. This approximate algorithm set $\mathbf{R}_{approx}$ was used to find the $\mathbf{R_P}(t)$, an optimal algorithm set with the tolerance

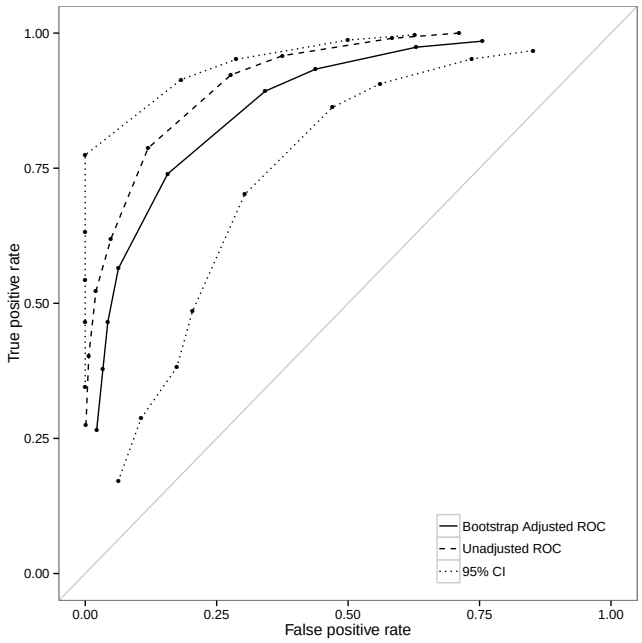


Figure 2.7: ROC curves generated using logic regression classification procedure. The bootstrap adjusted ROC curve shown as a solid black line was generated from the algorithm specified in Section 2.3.5.1. The adjusted ROC curve represents the expected performance of the algorithm. The unadjusted ROC curve is constructed from the unadjusted (apparent) error rates given by $\overline{Sp}$ and $\overline{Se}$. The 95% confidence interval curves are constructed using a percentile bootstrap confidence interval.

$t = 0.10$.

In table 2.10, we present a comparison table of the adjusted and unadjusted sensitivities and specificities for optimal algorithms generated by logistic regression, logic regression and the serial risk score classification. As expected, the adjusted bootstrap estimates have reduced the upward bias of the apparent estimates for sensitivity and specificity. For each of these algorithms, the cost savings for the values of p shown is decreased by at least 62% versus a classification method using all biomarkers (e.g. logistic regression). The AUC of the three methods are very similar with both logic regression and serial risk score classification performing better than logistic regression.

We present ROC curves for adjusted and unadjusted sensitivity and specificity from serial testing algorithms generated from the serial risk score classification algorithm in figure 2.8. For each point, the 95% lower and upper confidence limits of $\widehat{Se}$, and $\widehat{Sp}$ are plotted to form the confidence intervals shown on the graph. A comparison of the figures 2.8 and 2.7 shows wider bootstrap confidence intervals with logic regression for $\widehat{Se}$ and $\widehat{Sp}$. This suggests greater uncertainty in estimating misclassification rates with logic regression than with serial risk score classification.

In table 2.11, we present an example optimal algorithm chosen when $p = .3$. In each column, the risk score model coefficients for $\widehat{S}(X_i)$ for classification stage i are displayed. The decision rules for recent classification are also displayed under the corresponding risk score model coefficients. We also show the cumulative percentage that are classified in each stage as well as cumulative percentage of total cost in each stage. From the table, we can see that the majority of the observations are classified with only 2 tests.

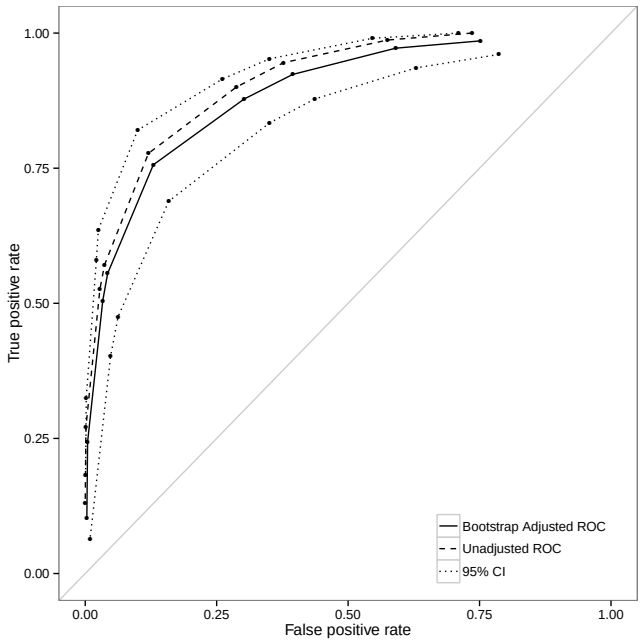


Figure 2.8: ROC curves generated using serial risk score classification procedure. The bootstrap adjusted ROC curve shown as a solid black line was generated from the algorithm specified in Section 2.3.5.1. The adjusted ROC curve represents the expected performance of the algorithm. The unadjusted ROC curve is constructed from the unadjusted (apparent) error rates given by $\overline{Sp}$ and $\overline{Se}$. The 95% confidence interval curves are constructed using a percentile bootstrap confidence interval.

Table 2.8: Biomarker threshold values used to generate predictors for logic regression procedure. Threshold values were generated from {5th, 10th, 15th ...} percentiles of the empirical distribution of each biomarker as noted in column 1; that is, for a biomarker X with value x and percentile p in the table, $P(X < x) \leq p$ and $P(X \geq x) \leq 1 - p$

Percentile (%)	Avidity	BED	CD4	log(Viral Load)
0	10.4	0.06	0	3.2
5	40.2	0.26	83	3.7
10	71.5	0.38	166	4.4
15	88.7	0.54	227	4.7
20	95.3	0.73	268	6.0
25	96.8	0.92	302	6.5
30	97.8	1.09	340	7.3
35	98.6	1.28	376	8.0
40	99.1	1.49	412	8.5
45	99.5	1.69	449	9.0
50	99.8	1.86	479	9.3
55	100.0	2.02	508	9.6
60	-	2.20	542	9.9
65	-	2.39	581	10.2
70	-	2.54	620	10.5
75	-	2.72	672	10.8
80	-	2.85	729	11.1
85	-	3.06	787	11.4
90	-	3.31	869	11.8
95	100.5	3.79	1006	12.4
100	106.9	6.02	1948	14.9

Table 2.9: Rules generated from logic regression classification method

p	sens.	spec.	ACS	Rule
0.01	0.99	0.24	98%	((not Avidity<100.5) or ((not BED<3.81) or (Avidity<100 or (not CD4<483))))
0.03	0.97	0.37	96%	(((not Avidity<100.5) or Avidity<100) or ((not logVL<9.33) and (not CD4<483)))
0.07	0.93	0.56	69%	((not logVL<4.07) and ((not CD4<168) and (Avidity<100.7 or (not Avidity<100.5))))
0.10	0.89	0.66	72%	(((Avidity<99.47 or BED<0.91) and (not logVL<4.07)) and (not CD4<168))
0.30	0.74	0.85	90%	((Avidity<97.69 and (not CD4<168)) and ((not BED<2.72) or (not logVL<4.61)))
0.70	0.56	0.94	80%	((Avidity<95.2 and ((not BED<2.38) or (not logVL<4.61))) and (not CD4<270))
1.00	0.47	0.96	79%	((Avidity<95.2 and (not CD4<270)) and (Avidity<39.54 or (not logVL<6.47)))
3.00	0.38	0.97	78%	((Avidity<95.2 and (not CD4<381)) and ((not logVL<6.47) and BED<2.38))
7.00	0.27	0.98	85%	(((Avidity<87.73 and (not logVL<7.3)) and BED<1.48) and (not CD4<381))
AUC =	0.87			

Table 2.10: Sensitivity (sens.), specificity (spec.), and average cost savings (ACS) of serial risk score classification and logistic regression applied to dataset described in Section 2.4. ACS refers to the average cost decrease v.s. the cost of all diagnostic tests.

	Logistic Regression		Logic Regression					Serial Risk Score				
			Apparent		Bootstrap			Apparent		Bootstrap		
p	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	ACS	Sens.	Spec.	Sens.	Spec.	ACS
0.03	1.00	0.04	0.99	0.42	0.97	0.37	96%	0.99	0.43	0.97	0.41	74%
0.07	0.90	0.56	0.96	0.62	0.93	0.56	69%	0.95	0.62	0.92	0.61	62%
0.10	0.88	0.61	0.92	0.72	0.89	0.66	72%	0.90	0.71	0.88	0.70	82%
0.30	0.65	0.89	0.79	0.88	0.74	0.85	90%	0.78	0.88	0.76	0.87	76%
0.70	0.43	0.96	0.62	0.95	0.56	0.94	80%	0.57	0.96	0.56	0.96	81%
1.00	0.41	0.97	0.52	0.98	0.47	0.96	79%	0.53	0.97	0.50	0.97	81%
AUC =	0.84		0.92		0.87			0.91		0.89		

Logistic regression model including all diagnostic tests: avidity, BED, CD4, and viral load. Individually each diagnostic test has an AUC of 0.85, 0.73, 0.65 and 0.56 respectively.

Table 2.11: Estimated regression coefficients to serial risk score rule chosen for $p = 0.30$ when $t = 0.10$. The decision rules for classification are displayed under the corresponding risk score model coefficients. Cumulative % classified refers to the percentage of observations classified ($\widehat{Y} = +1$ or -1) up to and including the specified stage. Cumulative % of total cost is cost expenditure up to and including the specified stage divided by the total cost at the end of stage 4.

	Stage			
	1	2	3	4
Reg. Coefficients				
Intercept	3.27	51.61	-72.73	-21.37
Avidity Index	-0.05	-0.54	0.71	0.21
BED-CEIA		-0.28	0.06	-0.14
Log Viral Load			0.41	0.01
CD4				2.7×10^{-3}
Decision Rule				
$\widehat{Y} = -1$	$\widehat{S} < 0$	$\widehat{S} < 0.18$	$\widehat{S} < 0.27$	$\widehat{S} < 0.33$
$\widehat{Y} = 0$	$0 \leq \widehat{S} < 0.13$	$0.18 \leq \widehat{S} < 0.51$	$0.27 \leq \widehat{S} < 1.00$	—
$\widehat{Y} = +1$	$\widehat{S} \geq 0.13$	$\widehat{S} \geq 0.51$	$\widehat{S} \geq 1.00$	$\widehat{S} \geq 0.33$
Cumulative % Classified	20.1	92.0	96.0	100
Cumulative % of Total Cost	10.6	67.5	81.1	100

The serial risk score algorithm presented here is based on 4 diagnostic biomarkers: avidity, BED-CEIA, CD4 and viral load. Avidity is shown measured as the avidity index (in %). BED capture immunoassay (BED-CEIA) is shown measured in normalized optical density (OD-n). The viral load assay is shown measured in copies/ml. CD4 is shown measured in cells/mm^3. For each decision rule in stage i, $\widehat{S}_i = 1/(1 + e^{-X_i\beta_i})$ where β_i is given in the i^{th} column of the regression coefficients.

2.5 Discussion

In this chapter, we described methods for generating serial testing algorithms for diagnostic testing and classification with considerations for cost. The first is logic regression with an extension for cost optimization which generates logic rules that fit into a serial testing framework. The second is serial risk score classification which uses risk score modeling in a serial testing structure. These two methods create two non overlapping sets of algorithms for diagnostic testing.[1]

The serial testing approach is designed to make classifications before proceeding to the next stage and incurring additional costs associated with additional diagnostic tests. Using this methodology, we were able to realize at least a 62% reduction in costs for identifying recently HIV infected persons (<1 year) using up to 4 diagnostic tests compared to testing all persons with all 4 diagnostic tests using standard risk score modeling based on logistic regression. In addition, the algorithms created by both logic regression and serial risk score classification had similar accuracy to logistic regression.

A challenge with the serial risk score classification approach we have proposed is the computational intensity of the optimization. Each additional biomarker exponentially increases the computation time as well as memory requirements to store the generated algorithms. The bootstrapping adjustment procedure further complicates the method, as each bootstrap sample requires a new set of generated algorithms. Fortunately, the approach is capable of being parallelized in a computation cluster.

The implementation of serial risk score classification described in the application was limited in a number of ways. The number of risk score thresholds for consideration at each stage had been limited to deciles of the risk score space. It would be useful to examine the sensitivity of the accuracy to this choice. We

[1]For instance, one rule not reproduceable by serial risk score classification is $(B_1 \wedge B_2) \vee (B_3 \wedge B_4)$. We address a rule not possible with logic regression in Section 2.3.4.

might consider introducing refinements to the risk score model such as interactions or polynomial terms. We used logistic regression models to estimate the risk score, but alternative models could be used such as a probit or a Bayesian probability classifier (Kim, 2013). There may also be restrictions in the ordering of the assays that may require consideration. For example, it may be logistically important to perform the CD4 assay first, because of the difficulty associated with cryo preserving viable samples (Brookmeyer et al., 2013b; Laeyendecker et al., 2013). In addition, there are considerations in choosing the tolerance of the loss function. Using a larger tolerance may lower costs further but may diminish accuracy. A smaller tolerance may increase costs, but increase accuracy. Approaches for choosing the tolerance could be studied. Finally, McIntosh and Pepe (2002), have discussed modeling considerations for combining diagnostic tests which could be incorporated into our risk score estimation within each stage. We discuss further extensions for research in Chapter 4.

The method we have presented describes an approach to develop algorithms for diagnostic testing that mediates the trade-off between cost and accuracy. The method has shown to be accurate but with considerably less cost than logistic regression in many situations of interest. We believe the serial testing algorithms proposed can be a useful approach for screening populations in resource-limited settings.

CHAPTER 3

Estimation of the Odds Ratio under Misclassification of Cases

3.1 Introduction

In this chapter, we examine the case-control study design previously mentioned to examine risk factors for disease where the case group may be misclassified. Our focus is on the estimation of an odds ratio, ω_y, for a specific case type y in exposed and unexposed persons. The populations of interest here are the control population, the true case population (called case I), and a population which can be misclassified as cases (called case II). We assume an imperfect rule is employed to separate the two case groups. If the risk profiles of the two case groups are very different, ω_y will be biased by the misclassification. We show that an adjustment of the odds ratio estimate can be made using knowledge of the misclassification rates of a particular algorithm to reduce the bias.

Our problem is motivated by studies where the interest is in examining risk factors for incident infection in HIV. The control group in this type of study are uninfected persons, whereas the cases are recently infected. The remarks concerning behavior changes over time in Chapter 1 apply here. As a result, the recently infected represent the leading edge of incidence risk and are most informative in discovering contributing risk factors for infection.

The adjustment for misclassification between cases and controls and exposure has been described previously (Barron, 1977; Greenland et al., 1983). The setting

we describe in this paper is distinguished from prior work in that misclassification occurs between two different types of cases.

3.1.1 Preliminary Concepts

3.1.1.1 Notation

In this chapter, we redefine Y and $\hat{Y}$ to take on values from $\{0, 1, 2\}$, corresponding to the disease statuses: uninfected, case I and case II respectively to reduce confusion in the notation. We will consider two kinds of study designs in this chapter. The first in which the study population is randomly sampled in a cross-sectional study. The second in which a random sample is drawn from each disease status group in a case-control study.

Define the vector $\boldsymbol{n} = (n_{00}, n_{01}, n_{02}, n_{10}, n_{11}, n_{12})^\intercal$ where n_{ey} is the number of observations where the dichotomous exposure variable $E = e$ and the disease status classification $\hat{Y} = y$. Define the single subscript notation $\boldsymbol{n}_k$ to represent the element in the vector $\boldsymbol{n}$ at the index k. A sample of size N is tabulated in the 2×3 contingency table shown in Table 3.1 where the counts of all combinations of disease status and exposure populate the cells of the table. The subscript $\cdot$ indicates either the sum over a column or a row, so that $n_{i\bullet}$ denotes the row totals and $n_{\bullet j}$ denotes the column totals. We assume N is large and there are no zero counts in any of the table cells.

Dependent on the sampling design, the counts in $\boldsymbol{n}$ may represent samples from the joint probability $Pr(E, \widehat{Y})$ in a random sampling setting or a conditional probability $Pr(E|\hat{Y})$ as in a case control setting. Let $\boldsymbol{p} = (p_{00}, p_{01}, p_{02}, p_{10}, p_{11}, p_{12})^\intercal$ define a probability distribution for $\boldsymbol{n}$ so that $\boldsymbol{n} \sim \text{Multinomial}(N, \boldsymbol{p})$.

Define the odds ratio ω_y of a case of type y and the control group as,

Table 3.1: Sample counts divided into cells indicating classification of disease stage and exposure E. n_{ey} represents the counts of persons who have been classified into the table cell (e, y) using an imperfect classification rule.

	Not Diseased	Case I	Case II	
	$(\widehat{Y}=0)$	$(\widehat{Y}=1)$	$(\widehat{Y}=2)$	Total
$(E=0)$	n_{00}	n_{01}	n_{02}	$n_{0\bullet}$
$(E=1)$	n_{10}	n_{11}	n_{12}	$n_{1\bullet}$
Total	$n_{\bullet 0}$	$n_{\bullet 1}$	$n_{\bullet 2}$	N

$$\omega_y = \frac{Pr(E_i = 1|Y_i = y)}{Pr(E_i = 0|Y_i = y)} \frac{Pr(E_i = 0|Y_i = 0)}{Pr(E_i = 1|Y_i = 0)} \tag{3.1}$$

Note that an application of the Bayes rule to the conditional probabilities in equation 3.1 leads to an equivalent statement in terms of the joint probabilities.

$$\omega_y = \frac{Pr(E = 1, Y = y)}{Pr(E = 0, Y = y)} \frac{Pr(E = 0, Y_i = 0)}{Pr(E = 1, Y_i = 0)} \tag{3.2}$$

3.1.1.2 Assumptions of Misclassification

In the general case of misclassification, we define a 6×6 misclassification probability matrix $\boldsymbol{Q} = (q_{ij}),\ i, j = (1 \dots 6)$, where q_{ij} represents the probability that an observation is classified into the count $\mathbf{n}_i$ when the its true classification is in $\mathbf{n}_j$. We make the assumption that each observed misclassification event is independent; therefore, $\boldsymbol{p} = \boldsymbol{Q}\boldsymbol{\pi}$ where $\boldsymbol{\pi}$ indicates the latent true probability vector for $\boldsymbol{n}$ under no misclassification. In addition, we assume non-differential misclassification between case I and case II, so that the probability of misclassification is independent of the exposure E.

For misclassification of disease status, Y, the matrix $\boldsymbol{Q}$ is specified by the sensitivity and specificity of the classification algorithm in a given sample. Exogenous

estimates for sensitivity and specificity of a rule can be applied; however, misspecification of these parameters may bias the estimates (Rothman et al., 2008).

Let Se and Sp denote the known sensitivity and specificity of a classification algorithm for the sample under study. Then with no misclassification of the control group and non-differential misclassification between the two cases, the misclassification matrix $\boldsymbol{Q}$ is block diagonal and is specified,

$$\boldsymbol{Q} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & Se & (1-Sp) \\ 0 & (1-Se) & Sp \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

For a cross-sectional random sample of a given population, Se and Sp may be specified using general population-based estimates; however, these estimates will not apply to a case-control sample of the same population due to the oversampling of cases. (Greenland et al., 1983). It is suggested a validation substudy be conducted to gather these rates (Greenland et al., 1983). For now, we assume that $\boldsymbol{Q}$ is correctly specified for the sample under study.

A classification algorithm where $Se + Sp < 1$ lies below the line of no-discrimination in a ROC curve and would be considered a very poor classifier. We make the weak assumption that classification algorithms in our discussion have sensitivities and specificities where $Se + Sp > 1$.

3.2 Estimators of Odds Ratio

3.2.1 Naïve Estimator

A naïve estimate of the odds ratio that does not account for misclassification is derived by assuming $Y = \widehat{Y}$. Using MLE estimates for $Pr(E|Y)$ in equation 3.1,

we define the naïve odds ratio estimate,

$$\bar{\omega}_y = \frac{\widehat{Pr}(E=1|\hat{Y}_i=y)}{\widehat{Pr}(E=0|\hat{Y}_i=y)}\frac{\widehat{Pr}(E=0|\hat{Y}_i=0)}{\widehat{Pr}(E=1|\hat{Y}_i=0)} = \frac{n_{1y}/n_{\bullet y}}{n_{0y}/n_{\bullet y}}\frac{n_{00}/n_{\bullet 0}}{n_{10}/n_{\bullet 0}} = \frac{n_{1y}}{n_{0y}}\frac{n_{00}}{n_{10}}.$$

We note that the estimate is the same for a cross-sectional or case-control study. An alternative formulation of the naïve estimator is given by a logistic model using dummy indicator variables for disease status, e.g. $\mathcal{I}(\hat{Y}_i = y) = 1$ if $\hat{Y}_i = y$. Define the vector $\hat{\boldsymbol{Y}}_i = [1, \mathcal{I}(\hat{Y}_i = 1), \mathcal{I}(\hat{Y}_i = 2)]^\intercal$. Then,

$$log\left(\frac{Pr(E_i=1|\hat{\boldsymbol{Y}}_i)}{1-Pr(E_i=1|\hat{\boldsymbol{Y}}_i)}\right) = \beta_0 + \beta_1\mathcal{I}(\hat{Y}_i=1) + \beta_2\mathcal{I}(\hat{Y}_i=2) = \boldsymbol{\beta}^\intercal\hat{\boldsymbol{Y}}_i \qquad (3.3)$$

We also write the above model as

$$P(E_i=1|\hat{\boldsymbol{Y}}_i) = \frac{exp(\boldsymbol{\beta}^\intercal\hat{\boldsymbol{Y}}_i)}{1+exp(\boldsymbol{\beta}^\intercal\hat{\boldsymbol{Y}}_i)} := \mathcal{C}(\boldsymbol{\beta}^\intercal\hat{\boldsymbol{Y}}_i)$$

In this formulation, $\bar{\omega}_y = exp(\beta_y)$. Under no misclassification, it has been shown that $\bar{\omega}_y$ computed through maximum likelihood yields consistent results for the odds ratio ω_y for both cross-sectional and case-control samples (Prentice and Pyke, 1979). Furthermore, the covariance matrix for $\boldsymbol{\beta}$ is valid for both study designs and at worst conservative under case-control (Carroll et al., 1995).

To examine the misclassification bias of this model, we first enumerate the possible values of $\mathcal{C}(Y_i = y; \boldsymbol{\beta}) = c_{1y}$ over the range of Y_i and let $c_{0y} = 1 - c_{1y}$,

$$c_{00} = \frac{1}{1+exp(\beta_0)} \qquad c_{10} = \frac{exp(\beta_0)}{1+exp(\beta_0)}$$
$$c_{01} = \frac{1}{1+exp(\beta_0+\beta_1)} \qquad c_{11} = \frac{exp(\beta_0+\beta_1)}{1+exp(\beta_0+\beta_1)}$$
$$c_{02} = \frac{1}{1+exp(\beta_0+\beta_2)} \qquad c_{12} = \frac{exp(\beta_0+\beta_2)}{1+exp(\beta_0+\beta_2)}$$

Solving for the β's, we find the relationships,

$$\beta_0 = log(c_{10}/c_{00})$$
$$\beta_1 = log(\frac{c_{11}c_{00}}{c_{01}c_{10}})$$
$$\beta_2 = log(\frac{c_{12}c_{00}}{c_{02}c_{10}})$$

Let β^* be the true coefficients for the model under no misclassification. We define the bias of β_1, $\Delta_1(Se, Sp) = \beta^* - \beta$.

$$\begin{aligned}\Delta_1(Se, Sp) &= \beta^* - \beta \\ &= log\left(\frac{Se + (1 - Sp)\gamma_{02}/\gamma_{01}}{Se + (1 - Sp)\gamma_{12}/\gamma_{11}}\right)\end{aligned}$$

An examination of this equation reveals conditions for unbiasedness, namely, if specificity is equal to one or if the odds ratio between case I and case II is 1. This is demonstrated in the Figure 3.1, a plot of the percent error of the naïve estimate across three misclassification rates by ω_1/ω_2. It is shown that the magnitude of the percent error increases as the ratio ω_1/ω_2 diverges from 1, that is, as the risk profiles of the two case groups diverge, bias of the naïve estimator increases. In addition, as specificity increases the percent error decreases towards the line of no error.

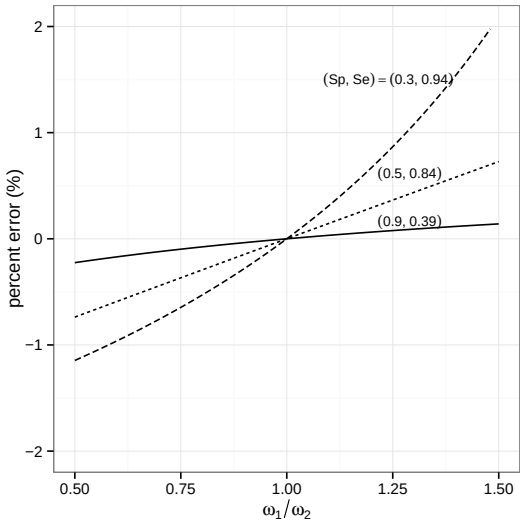


Figure 3.1: Percent error of $\bar{\omega_1}$ by ω_1/ω_2. Percent error is defined by $100\% \cdot (\bar{\omega_1} - \omega_1)/\omega_1$. Each line corresponds to a different rule with parameters sp, se labeled above the line.

3.2.2 Matrix-adjusted Odds Ratio

One method for adjusting the odds ratio is to adjust the counts in each cell by inverting a misclassification matrix $\boldsymbol{Q}$. The expectation of counts of subjects in each cell of table 3.1 can be expressed as a function of the latent true counts and the misclassification rates. Here we describe a matrix-adjustment formulation described previously by Greenland, Ericson, and Kleinbaum (1983). The matrix adjustment draws on the assumption that $E\boldsymbol{n} = \boldsymbol{Q}\boldsymbol{\pi}N$ given by the multinomial. This expression leads to an adjusted odds ratio estimate based on adjusted counts, $\hat{\boldsymbol{n}}$. Using $\boldsymbol{n}$ in place for $E\boldsymbol{n}$, we estimate the true counts for the sample to be

$$\hat{\boldsymbol{n}} = \boldsymbol{Q}^{-1}\boldsymbol{n},$$

3.2.2.1 Matrix-adjusted Odds Ratio under Random Sampling

The adjustment matrix $\boldsymbol{Q}$ is block diagonal so the inverse is easily specified,

$$\boldsymbol{Q}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{Sp}{Se+Sp-1} & \frac{-(1-Sp)}{Se+Sp-1} \\ 0 & \frac{-(1-Se)}{Se+Sp-1} & \frac{Se}{Se+Sp-1} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The misclassification matrix $\boldsymbol{Q}$ is singular and not invertible if $Se + Sp = 1$, which corresponds to a completely random classifier. Under random sampling, we define the adjusted estimate for $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{n}}/N = \boldsymbol{Q}^{-1}(\boldsymbol{n}/N).$$

This estimate is a consistent estimator for $\boldsymbol{\theta}$ if the misclassification rates are known. Since $\boldsymbol{n}/N \rightarrow_p \boldsymbol{p}$ and $\boldsymbol{Q}^{-1}\boldsymbol{p} = \boldsymbol{\theta}$, then $\hat{\boldsymbol{\theta}} \rightarrow_p \boldsymbol{\theta}$. Using the adjusted estimates for θ_{ij}, we estimate ω_y by

$$\hat{\omega}_y = \frac{\hat{\theta}_{1y}}{\hat{\theta}_{0y}} \frac{\hat{\theta}_{00}}{\hat{\theta}_{10}} \tag{3.4}$$

Hence, the matrix adjusted estimator for $\hat{\omega}_1$ and $\hat{\omega}_2$ are given by the equations,

$$\hat{\omega}_1 = \left(\frac{Sp \cdot n_{11} - (1 - Sp)n_{12}}{Sp \cdot n_{01} - (1 - Sp)n_{02}} \right) \frac{n_{00}}{n_{10}}. \tag{3.5}$$

$$\hat{\omega}_2 = \left(\frac{Se \cdot n_{12} - (1 - Se)n_{11}}{Se \cdot n_{02} - (1 - Se)n_{01}} \right) \frac{n_{00}}{n_{10}}. \tag{3.6}$$

By a delta method approximation, the *log* matrix adjusted odds ratio has an approximate asymptotic normal distribution where

$$log(\hat{\omega}_y) \sim \mathcal{N}(log(\omega_y), \sigma/\sqrt{N}),$$

where

$$\sigma^2 = \frac{1}{p_{00}} + \frac{1}{p_{10}} + \frac{Sp^2 p_{01} + (1 - Sp)^2 p_{02}}{(Sp\ p_{01} - (1 - Sp)p_{02})^2} + \frac{Sp^2 p_{11} + (1 - Sp)^2 p_{12}}{(Sp\ p_{11} - (1 - Sp)p_{12})^2} - 4$$

We define a Wald confidence interval by the large sample normality of $log(\hat{\omega}_y)$ to be

$$log(\hat{\omega}_y) \pm z_{\alpha/2} \hat{\sigma}(log(\hat{\omega}_y))$$

where

$$\hat{\sigma}(log(\hat{\omega}_y)) = \sqrt{\frac{1}{n_{00}} + \frac{1}{n_{10}} + \frac{Sp^2 n_{01} + (1-Sp)^2 n_{02}}{(Sp\ n_{01} - (1-Sp)n_{02})^2} + \frac{Sp^2 n_{11} + (1-Sp)^2 n_{12}}{(Sp\ n_{11} - (1-Sp)n_{12})^2} - 4/N}.$$

The consistency of this matrix adjusted estimator is examined and compared to the naïve estimator in Figure 3.2. The estimator appears to converge to the true odds ratio with increasing sample size, whereas the naïve estimator is biased.

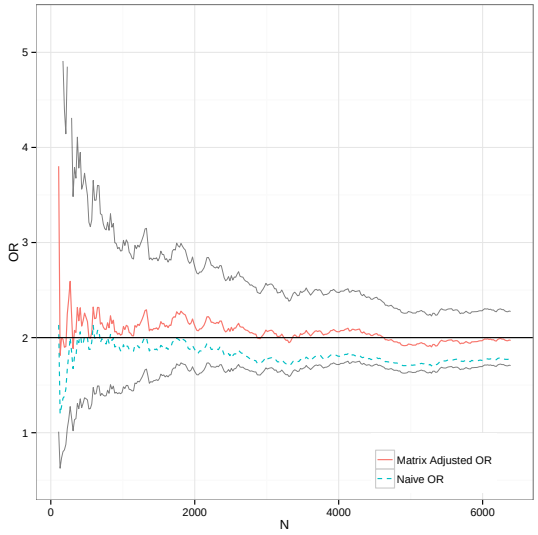


Figure 3.2: Odds ratio estimates (OR) with the matrix adjusted estimator and the naïve estimator with increasing sample size. Here the true odds ratio of X for case I and controls is given by the dark horizontal line at $OR = 2$. The odds ratio of X for case II and controls was set to 1. The classification algorithm has a sensitivity of 0.84 and a specificity of 0.7.

3.2.2.2 Matrix-adjusted Odds Ratio under Case-Control Sampling

If $\boldsymbol{Q}$ is known for the specific case-control sample, then estimates for ω_y have the same form as given in Section 3.2.2.1.

Alternatively, an estimator for the odds ratio under case-control sampling may be derived by adjusting for the oversampling of the cases. Let $\rho_j^* = Pr(\hat{Y}_i = j)$ be the population prevalences of each disease status classification. Since under case-control sampling the sample data arises from the conditional distribution $Pr(X = i|\hat{Y} = j)$, we reduce the problem to the one described in Section 3.2.2.1 by multiplying the conditional and marginal distributions to obtain the joint distribution $Pr(X = i, \hat{Y} = j)$, i.e. $Pr(X = i, \hat{Y} = j) = Pr(X = i|\hat{Y} = j)Pr(\hat{Y} = j)$.

Thus, the matrix adjusted estimator for $\hat{\omega}_1$ and $\hat{\omega}_2$ are given by the equations,

$$\hat{\omega}_1 = \left(\frac{\rho_1^* Sp \cdot n_{11} - \rho_2^*(1-Sp)n_{12}}{\rho_1^* Sp \cdot n_{01} - \rho_2^*(1-Sp)n_{02}} \right) \frac{n_{00}}{n_{10}}. \tag{3.7}$$

$$\hat{\omega}_2 = \left(\frac{\rho_2^* Se \cdot n_{12} - \rho_1^*(1-Se)n_{11}}{\rho_2^* Se \cdot n_{02} - \rho_1^*(1-Se)n_{01}} \right) \frac{n_{00}}{n_{10}}. \tag{3.8}$$

Again, the *log* matrix adjusted odds ratio has an approximate asymptotic normal distribution where

$$log(\hat{\omega}_y) \sim N(log(\omega_y), \sigma/\sqrt{n}),$$

where

$$\begin{aligned} \sigma^2 =& \frac{1}{p_{00}} + \frac{1}{p_{10}} + \frac{(\rho_1^*)^2 Sp^2 p_{01} + (\rho_2^*)^2 (1-Sp)^2 p_{02}}{(\rho_1^* Sp\ p_{01} - \rho_2^*(1-Sp)p_{02})^2} \\ &+ \frac{(\rho_1^*)^2 Sp^2 p_{11} + (\rho_2^*)^2 (1-Sp)^2 p_{12}}{(\rho_1^* Sp\ p_{11} - \rho_2^*(1-Sp)p_{12})^2} - 4 \end{aligned}$$

A confidence interval is defined similarly as in the previous section by substitution of n_{ij}/N for p_{ij} in σ^2.

3.2.3 Limitations of Matrix Adjustment Method

A limitation of this simple matrix adjustment method in our misclassification framework is that it may lead to inadmissible estimates where $\hat{\pi}_j \notin (0,1)$ for some $j = (1 \ldots 6)$ (Viana, 1994). Since misclassification occurs only between cases, attention is focused on probabilities p_{e1} and p_{e2} for exposure e.

According to our assumptions of misclassification, the relationship between p_{e1}, p_{e2} and the π_{e1}, π_{e2} is stated thusly,

$$\begin{bmatrix} p_{e1} \\ p_{e2} \end{bmatrix} = \begin{bmatrix} Se & (1-Sp) \\ (1-Se) & Sp \end{bmatrix} \begin{bmatrix} \pi_{e1} \\ \pi_{e2} \end{bmatrix}.$$

Conditioning on $Y_i \in (1,2)$ and $E = e$, we find

$$\begin{bmatrix} p_{e1}/(p_{e1}+p_{e2}) \\ p_{e2}/(p_{e1}+p_{e2}) \end{bmatrix} = \begin{bmatrix} Se & (1-Sp) \\ (1-Se) & Sp \end{bmatrix} \begin{bmatrix} \pi_{e1}/(\pi_{e1}+\pi_{e2}) \\ \pi_{e2}/(\pi_{e1}+\pi_{e2}) \end{bmatrix}. \tag{3.9}$$

where $p_{e1} + p_{e2} = \pi_{e1} + \pi_{e2}$. From (3.9), we find that $p_{e1}/(p_{e1} + p_{e2})$ is a weighted average of Se and $(1-Sp)$ where $\pi_{e1}/(\pi_{e1}+\pi_{e2})$ and $\pi_{e2}/(\pi_{e1}+\pi_{e2})$ act as weights. Likewise, $p_{e2}/(p_{e1}+p_{e2})$ is a weighted average of $(1-Se)$ and Sp with $\pi_{e1}/(\pi_{e1}+\pi_{e2})$ and $\pi_{e2}/(\pi_{e1} + \pi_{e2})$ as weights. Therefore, for matrix adjusted estimates of the odds ratio to be valid, p_{e1} and p_{e2} must be subject to the constraints,

$$\min(Se, (1 - Sp)) \leq p_{e1}/(p_{e1} + p_{e2}) \leq \max(Se, (1 - Sp))$$

and

$$\min((1 - Se), Sp) \leq p_{e2}/(p_{e1} + p_{e2}) \leq \max((1 - Se), Sp).$$

A similar logic can be applied in the case control situation for the conditional probabilities. If any of the empirical proportions, $\hat{p}_{e1} = n_{e1}/N$, $\hat{p}_{e2} = n_{e2}/N$ fall outside of these constraints, we may be subject to inadmissible estimates.

The following example can provide insight into this issue. Consider the cross-sectional sample of 2000 observations presented in Table 3.2.3 where case I and case II were classified with an algorithm with specificity of 90% and sensitivity of 60%. This sample fails the condition where $p_{e1}/(p_{e1} + p_{e2}) > \min(Se, (1 - Sp))$, $(.095 > .1)$. If we continue with the matrix adjusted estimator to estimate ω_1, we find an odds ratio of -35 which is not valid.

Table 3.2: Example data for limiting case of matrix adjustment

	Not Diseased	Case (I)	Case (II)	
	$(\widehat{Y} = 0)$	$(\widehat{Y} = 1)$	$(\widehat{Y} = 2)$	Total
$(E = 0)$	951	6	57	1014
$(E = 1)$	873	21	92	986
Total	1824	27	149	2000

3.3 Multinomial Logistic Regression Estimator

Here we develop a multinomial logistic regression model that is based on a maximum likelihood method and not subject to the admissibility constraints of the matrix adjusted estimator. This method can also extend our estimation discussion beyond a dichotomous risk factor. The multinomial regression model assumes the log odds of either case I or case II is linear in E and is given by,

$$log\frac{P(Y_i = j|E_i)}{P(Y_i = 0|E_i)} = \beta_{j0} + \beta_{j1}E_i, \text{ for } j = 1, 2. \tag{3.10}$$

Solving for $P(Y_i = y|E_i) := \Phi_y(E_i; \boldsymbol{\beta})$, we find an expression for the probabilities for each disease status defined by the model.

$$\Phi_0(E_i; \boldsymbol{\beta}) = \frac{1}{1 + exp(\beta_{10} + \beta_{11}E_i) + exp(\beta_{20} + \beta_{21}E_i)}$$

$$\Phi_1(E_i; \boldsymbol{\beta}) = \frac{exp(\beta_{10} + \beta_{11}E_i)}{1 + exp(\beta_{10} + \beta_{11}E_i) + exp(\beta_{20} + \beta_{21}E_i)}$$

$$\Phi_2(E_i; \boldsymbol{\beta}) = \frac{exp(\beta_{20} + \beta_{21}E_i)}{1 + exp(\beta_{10} + \beta_{11}E_i) + exp(\beta_{20} + \beta_{21}E_i)}$$

From this model definition, we find the following result,

$$\begin{aligned}\beta_{j1} &= \log\left(\frac{P(Y_i = 1|E_i = 1)}{P(Y_i = 0|E_i = 1)} - log\frac{P(Y_i = 1|E_i = 0)}{P(Y_i = 0|E_i = 0)}\right) \\ &= \log\left(\frac{P(Y_i = 1|E_i = 1)P(Y_i = 0|E_i = 0)}{P(Y_i = 1|E_i = 0)P(Y_i = 0|E_i = 1)}\right).\end{aligned}$$

Hence with a dichotomous predictor, the interpretation of the coefficient β_{j1} for $j = 1, 2$ is the log odds ratio of case type (j) to the uninfected group. With a continuous predictor, β_{j1} describes the log odds ratio for case type (j) vs. the uninfected group or the change in log odds associated with a one unit increase in E.

3.3.1 Multinomial Logistic Regression Under Cross-sectional Sampling

In a cross-sectional study with misclassification, the sampling model yields the following conditional log likelihood,

$$logL = \sum_{i=1}^{N} \sum_{j=0}^{2} \mathcal{I}(\hat{Y}_i = j) \log P(\hat{Y}_i = j | E_i = e) \tag{3.11}$$

To adjust for misclassification, we derive $P(Y_i^* = j | E_i = e)$ as a function of the misclassification rates and the true classification model.

$$\begin{aligned} P(\hat{Y}_i = j | E_i = e) &= \sum_{y=0}^{2} P(\hat{Y}_i = j | Y_i = y, E_i = e) P(Y_i = y | E_i) \\ &= \sum_{y=0}^{2} P(\hat{Y}_i = j | Y_i = y) P(Y_i = y | E_i) \end{aligned} \tag{3.12}$$

This allows us to write the expression for the conditional likelihood incorporating the misclassification rates.

$$logL = \sum_{i=1}^{N} \sum_{j=0}^{2} \mathcal{I}(\hat{Y}_i = j) \log \sum_{y=0}^{2} q_{j(y+1)} \Phi_y(E_i; \boldsymbol{\beta})$$

where $q_{j(y+1)}$ is the element at the jth row and $(y+1)$st column of the misclassification matrix $\boldsymbol{Q}$[1]. The β parameters are solved through maximum likelihood methods. Estimating equations for this model are given by the partial derivatives of the likelihood with respect to β. They are listed in Appendix B.

3.3.2 Multinomial Logistic Regression Under Case-control Sampling

The model 3.10 based on $P(Y_i|E_i)$ can be applied to a case-control setting to produce consistent non-intercept parameters (Prentice and Pyke, 1979); however, the formulation of the likelihood is altered to include an inclusion variable. We introduce a probability of inclusion in the study to account for oversampling of

[1]Here this notation exploits the fact that $\boldsymbol{Q}$ is composed of two identical blocks diagonally.

either of the cases. We introduce a variable Z which represents if an individual is selected for the study or not. The log likelihood now is conditional on E and the inclusion indicator Z and equation 3.11 becomes

$$logL = \sum_{i=1}^{N} \sum_{j=0}^{2} \mathcal{I}(\hat{Y}_i = j) \log P(\hat{Y}_i = j | E_i = e, Z).$$

Define $\eta_j = Pr(\hat{Y}_i = j | Z_i = 1) = n_j/N$ where n_j is the sample size of the group with disease status j and N is the total sample size. Also as before define

$$\rho_j^* = Pr(\hat{Y}_i = j) = \sum_i Pr(\hat{Y}_i = j | Y_i = i) Pr(Y_i = i).$$

We derive the new sampling probability in the following way,

$$\begin{aligned} Pr(\hat{Y}_i = y_i | E_i, Z_i) &= \frac{Pr(Z_i = 1 | \hat{Y}_i) Pr(\hat{Y}_i | E_i)}{\sum_{y=0}^{2} Pr(Z_i = 1 | \hat{Y}_i = y) \cdot Pr(\hat{Y}_i = y | E_i)} \\ &= \frac{(\eta_{y_i}/\rho_{y_i}^*) Pr(\hat{Y}_i = y_i | E_i)}{\sum_{y=0}^{2} (\eta_j/\rho_j^*) \cdot Pr(\hat{Y}_i = j | E_i)} \end{aligned}$$

This assumes conditional independence between the E and the sampling indicator Z given Y. Thus in a case-control sampling setting the likelihood is specified by,

$$logL = \sum_{i=1}^{N} \sum_{j=0}^{2} \mathcal{I}(\hat{Y}_i = j) \log \frac{(\eta_{y_i}/\rho_{y_i}^*) \sum_{y=0}^{2} q_{j(y+1)} \Phi_y(E; \boldsymbol{\beta})}{\sum_{y=0}^{2} (\eta_j/\rho_j^*) \cdot \sum_{y=0}^{2} q_{j(y+1)} \Phi_y(E; \boldsymbol{\beta})}$$

where $q_{j(y+1)}$ is the element at the jth row and $(y+1)$st column of the misclassification matrix $\boldsymbol{Q}$. Estimation continues in the usual fashion through maximum likelihood.

3.3.3 Identifiability Conditions

Identifiability of the β_1 and β_2 parameters in this model can be shown using a similar argument for the misclassified probit model case given by Hausman et al. (1998). The argument hinges upon monotonicity conditions on $f_y(E; \beta) =$

$P_\beta(\hat{Y}_i|E_i)$. We present partial derivatives of $f_y(E;\beta)$ in Appendix B. It is clear from the partial derivatives of $f_0(E;\beta)$ that it is monotonically decreasing. For $f_1(E;\beta)$ and $f_2(E;\beta)$ to be monotonically increasing, inspection of the partial derivatives of each reveal two conditions that must be fulfilled.

$$Se(1-\Phi_1(E;\boldsymbol{\beta})) - (1-Sp)\Phi_2(E;\boldsymbol{\beta}) > 0 \tag{3.13}$$

$$(1-Sp)(1-\Phi_2(E;\boldsymbol{\beta})) - Se\Phi_1(E;\boldsymbol{\beta}) > 0 \tag{3.14}$$

The expression 3.13 can also be written as

$$\frac{Se}{1-Sp} > \frac{\Phi_2(E;\boldsymbol{\beta})}{\Phi_0(E;\boldsymbol{\beta}) + \Phi_2(E;\boldsymbol{\beta})}$$

which will be always satisfied when $Se + Sp > 1$.

Similarly, expression 3.14 is

$$\frac{Se}{1-Sp} < \frac{\Phi_0(E;\boldsymbol{\beta}) + \Phi_1(E;\boldsymbol{\beta})}{\Phi_1(X;\boldsymbol{\beta})}$$

which will be always satisfied when $Se + Sp > 1$.

Both consistency and asymptotic normality of the log odds ratio ω_y follow from identifiability of the β from standard maximum likelihood theorems (Newey and McFadden, 1994).

3.3.4 Simulation

We evaluated the performance of the odds ratio estimators with misclassification using a simulation of 1000 data sets. In each simulation, we have modeled the latent counts $\boldsymbol{n}^*$ using a multinomial distribution where the probability vector is given by the model (3.10). The observed counts $\boldsymbol{n}$ are modeled using a Bernoulli random variable conditional on selected sensitivities and specificities.

We generated data for the simulations using the logistic model (3.10) where coefficients in the logistic model were specified, $\beta_j = [b_j \ log(w_j)]$ for $j = (1, 2)$.

The b_j parameter will affect the prevalence of each case type in the population. We define the b_j parameters so that the prevalence of case I is set to 10%, the prevalence of case II is set to 20% and control prevalence is set to 30%. The classification of each case type was determined by $\hat{Y}_i|Y_i \sim Bernoulli_{\hat{Y}|Y}(p)$ where $p = Se$ if $Y = 1$ and $p = Sp$ if $Y = 2$.

We considered two sampling situations: (1) the situation a sample of size N = 2000 is drawn randomly from the above model and (2) a case-control sampling situation. For case-control sampling, we specified equal allocation of 500 persons for the samples of control and case types. We sampled each from a large pool of $\gg 1500$ persons from the logistic model specified above. Prevalences of each classified case type was given and assumed known for each estimation method.

In Table 3.3, we provide the results of the simulation of a cross-sectional study specified by situation (1). We find that the naïve estimator can provide consistent results when the $\omega_2 = \omega_1$, i.e. the risk profiles between the two case types are the same. Otherwise, the naïve estimates are predictably biased downward if $\omega_2 < \omega_1$ and upward if $\omega_2 > \omega_1$. The multinomial regression estimator produces estimates that are fairly close to the true values and the coverage of confidence interval is close to the expected 95%. Matrix adjusted estimates of ω_1 were nearly identical to the presented results for the multinomial estimator and we omit the results for display.

In Table 3.4, we display results of the simulation of a case control study. The adjusted estimates are close to the true values and the coverage of the confidence interval is again close to the expected 95%. We note that in the case of when $\omega_1 = \omega_1$, the naïve estimator performs well with a narrower confidence interval than the other method. Considerations should be made as to the whether handling misclassification is necessary as there is a penalty in the variance when adjusting for misclassification.

Table 3.3: Monte Carlo simulation results [1] with cross-sectional sampling comparing the naïve and likelihood adjusted multinomial regression estimators for the odds ratio, $\omega_1 = 2$, of case I ($Y = 1$) to control group ($Y = 0$). Results presented for each estimation method are estimates for the odds ratio ω_1, standard error (SE) and the coverage probability of the 95% confidence interval (%In).

			Naive Estimator			Multinomial Regression		
ω_2	Sp	Se	$\bar{\omega}_1$	SE	%In	$\hat{\omega}_1$	SE	%In
1.00	0.50	0.93	1.39	0.12	11.1%	2.19	0.35	96.9%
	0.70	0.84	1.49	0.13	37.5%	2.11	0.29	95.9%
	0.90	0.59	1.69	0.17	79.7%	2.09	0.27	95.8%
1.33	0.50	0.93	1.63	0.12	55.3%	2.14	0.33	97.2%
	0.70	0.84	1.71	0.13	73.4%	2.13	0.28	95.5%
	0.90	0.59	1.83	0.17	90.2%	2.09	0.26	96.0%
2.00	0.50	0.93	2.03	0.12	94.0%	2.13	0.31	96.9%
	0.70	0.84	2.03	0.13	94.9%	2.10	0.26	96.1%
	0.90	0.59	2.05	0.17	95.0%	2.11	0.25	95.3%
4.00	0.50	0.93	2.83	0.12	19.0%	2.09	0.29	97.3%
	0.70	0.84	2.66	0.14	49.1%	2.07	0.25	95.4%
	0.90	0.59	2.41	0.18	83.9%	2.09	0.25	95.6%

[1]Here we performed 1000 simulations. On each simulation, we sampled 500 values for the control and each case group (N = 1500) from 10^6 generated values from a multinomial where for $Y_i \in (0, 1, 2)$, and $E_i \in (0, 1)$ $P(Y_i = y|E_i) = 1/\sum_{i=0}^{2} exp((\boldsymbol{\beta}_i - \boldsymbol{\beta}_y)^\intercal \vec{\boldsymbol{E}}_i)$, $(\beta_0 = \mathbf{0}), \beta_j = [\alpha_j \; log(w_j)]$ for $j = (1, 2)$; α_j was chosen for the prevalences of case I and case II to be 10% and 20% respectively. The classification of case type was determined $\hat{Y}|Y \sim Bernoulli_{\hat{Y}|Y}(p)$ where $p = Se$ if $Y = 1$ and $p = Sp$ if $Y = 2$. Table cells represent average values over 1000 simulations.

Table 3.4: Monte Carlo simulation results [1] with case-control sampling comparing the naïve and likelihood adjusted multinomial regression estimators for the odds ratio, $\omega_1 = 2$, of case I ($Y = 1$) to control group ($Y = 0$). Results presented for each estimation method are estimates for the odds ratio ω_1, standard error (SE) and the coverage probability of the 95% confidence interval (%In).

			Naive Estimator			Multinomial Regression		
ω_2	Sp	Se	$\bar{\omega}_1$	SE	%In	$\hat{\omega}_1$	SE	%In
1.00	0.93	0.50	1.40	0.13	17.1%	2.09	0.28	95.2%
	0.84	0.70	1.50	0.13	35.2%	2.07	0.23	96.1%
	0.59	0.90	1.69	0.13	71.1%	2.04	0.17	95.3%
1.33	0.93	0.50	1.63	0.13	62.8%	2.11	0.27	96.9%
	0.84	0.70	1.70	0.13	73.3%	2.06	0.22	95.0%
	0.59	0.90	1.82	0.13	87.9%	2.04	0.17	95.0%
2.00	0.93	0.50	2.00	0.13	93.3%	2.05	0.25	94.8%
	0.84	0.70	2.03	0.13	94.6%	2.07	0.21	95.4%
	0.59	0.90	2.03	0.13	95.2%	2.04	0.17	95.5%
4.00	0.93	0.50	2.83	0.13	25.9%	2.07	0.24	96.5%
	0.84	0.70	2.64	0.13	44.8%	2.04	0.21	96.6%
	0.59	0.90	2.38	0.13	76.7%	2.03	0.16	96.1%

[1]Here we performed 1000 simulations. On each simulation, we sampled 500 values for the control and each case group (N = 1500) from 10^6 generated values from a multinomial where for $Y_i \in (0,1,2)$, and $E_i \in (0,1)$ $P(Y_i = y|E_i) = 1/\sum_{i=0}^{2} exp((\boldsymbol{\beta}_i - \boldsymbol{\beta}_y)^\intercal \vec{\boldsymbol{E}}_i)$, $(\beta_0 = \mathbf{0}), \beta_j = [\alpha_j \; log(w_j)]$ for $j = (1,2)$; α_j was chosen for the prevalences of case I and case II to be 10% and 20% respectively. The classification of case type was determined $\hat{Y}|Y \sim Bernoulli_{\hat{Y}|Y}(p)$ where $p = Se$ if $Y = 1$ and $p = Sp$ if $Y = 2$. Table cells represent average values over 1000 simulations.

3.4 Discussion

In this chapter we discussed methods to adjust the odds ratio under misclassification in a cross-sectional and case-control study design. These methods are useful in a case control setting where there are two types of cases. This can be important in HIV research where the recently infected population can characterize risk factors for infections that are happening right now rather than some time in the past.

We presented two methods for reducing the bias of misclassification. One method is based on a matrix adjustment on the expected counts of the exposure and disease status. The second method, based on multinomial regression, is a likelihood based method that is constrained in the estimation to avoid the problems of the matrix adjustment. These estimators have shown good performance in simulations of a cross-sectional study and in a case-control study.

There are several extensions to this work presented in this chapter. Covariates can be introduced into the multinomial logistic regression model to control for confounding or interaction effects as well as increase power. This misclassification rates can also be dependent on risk factors or other covariates as well. In addition, the multinomial regression likelihood extends to allow for the estimation of misclassification rates as well under certain regularity and identifiability conditions (Lewbel, 2000).

We note that these methods produce more conservative confidence intervals than the naïve estimator; therefore, in situations where the specificity of a classifier is extremely high (with a suitable sensitivity as well), or when the risk profiles of the two case groups are equivalent, it may be advisable to use the naïve estimator instead of one that accounts for misclassification.

CHAPTER 4

Discussion

The goal of this dissertation was to examine problems in diagnostic testing arising out of current problems in HIV research. We have presented two novel approaches for developing "serial testing algorithms" for diagnostic testing while controlling costs: logic regression and serial risk score classification. These approaches were motivated by screening for recent HIV infection using multiple biomarkers. We have also discussed methods for adjusting an odds ratio in a case control study using an imperfect classification algorithm. Taken together, these approaches can be applied to examine risk factors for incident infections of HIV.

The serial testing algorithms we have described in Chapter 2 have shown to produce a great reduction of cost (at least 62%) in testing a cohort of HIV infected persons while maintaining the accuracy of a logistic regression approach. The logic regression method presented the highest cost savings among the studied methods, but the bootstrap confidence intervals suggest a larger variance in estimation of sensitivity and specificity when compared to serial risk score classification.

These classification methods could be extended in a number of directions. It would be useful to incorporate covariates into the risk score modeling performed at each stage of a serial risk score algorithm. For example, in our HIV application, it may be helpful to incorporate covariates such as HIV transmission group (e.g., intravenous drug use, men who have sex with men) and age. Furthermore, the modeling procedures we implemented tended to overfit the data in each stage. A simple variable selection procedure computed in each stage should help reduce this

effect and produce algorithms that are transferable and cross validation methods could be applied during optimization to reduce the variability of the accuracy estimates and overfitting.

Our problem is motivated by studies in HIV where the interest is in examining risk factors for incident infection. The control group in this type of study are uninfected persons, whereas the cases are recently infected. The recently infected represent the leading edge of incidence and are most informative in discovering contributing risk factors for infection. The issue is that while we can easily identify infection status, it is difficult to separate long-standing (chronic) infections from recent infections. The disease history of each person may be unknown, so other methods for classification must be used. An early method in HIV research used detuned assays to identify recent infections (Janssen et al., 1998). Another recent method has been described to use serological biomarkers to a determine disease status (Laeyendecker et al., 2013). Neither algorithm provided perfectly accurate classification.

The adjusted odds ratios that are discussed in Chapter 3 are shown to provide consistent estimates under misclassification. One limitation of the estimation of the adjusted odds ratio is the assumption of known misclassification rates. In practice, these rates can vary between populations and may not be known and should be accounted for in the variance of our estimates. One can avoid this problem by estimating the misclassification matrix using a validation substudy. This would allow for direct estimation of misclassification rates in a sample without knowledge of the prevalence rates and would allow for incorporation of the sampling error of the misclassification. In addition, one method for accounting for the extra variance is to use a bootstrapping method on a validation subsample to determine a bootstrap joint distribution for the misclassification rates. This distribution can be incorporated into the estimates to account for the extra variance.

It may also be useful to account for errors in the formulation of the true disease

classification Y. For example, in our HIV application, the calendar date of infection was based on midpoint imputation of the interval of seroconversion, which would introduce error into Y. Methods have been suggested for incorporating uncertainties in the dates of infection in HIV studies, and their extensions to our methods would be useful (Brookmeyer et al., 2013a; Sweeting et al., 2010).

The methods we have discussed were presented with a focus on problems motivated by HIV research but can be applied in other contexts. We believe our methods can be a useful approach in settings where diagnostic screening procedures are cost prohibitive in resource limited areas or even in cases where the gold standard test is painful or time-intensive.

CHAPTER 5

Bibliography

Barron, B. A. (1977), "The effects of misclassification on the estimation of relative risk," *Biometrics*, 414–418.

Brookmeyer, R., Konikoff, J., Laeyendecker, O., and Eshleman, S. H. (2013a), "Estimation of HIV incidence using multiple biomarkers," *American Journal of Epidemiology*, 177, 264–272.

Brookmeyer, R., Laeyendecker, O., Donnell, D., and Eshleman, S. H. (2013b), "Cross-Sectional HIV Incidence Estimation in HIV Prevention Research," *Journal of Acquired Immune Deficiency Syndromes*, 63, S233–S239.

Brookmeyer, R. and Quinn, T. C. (1995), "Estimation of current human immunodeficiency virus incidence rates from a cross-sectional survey using early diagnostic tests," *American Journal of Epidemiology*, 141, 166–172.

Carroll, R., Wang, S., and Wang, C. (1995), "Prospective analysis of logistic case-control studies," *Journal of the American Statistical Association*, 90, 157–169.

Celum, C. L., Buchbinder, S. P., Donnell, D., Douglas, J. M., Mayer, K., Koblin, B., Marmor, M., Bozeman, S., Grant, R. M., Flores, J., et al. (2001), "Early human immunodeficiency virus (HIV) infection in the HIV Network for Prevention Trials Vaccine Preparedness Cohort: risk behaviors, symptoms, and early plasma and genital tract virus load," *Journal of Infectious Diseases*, 183, 23–35.

Chawla, A., Murphy, G., Donnelly, C., Booth, C., Johnson, M., Parry, J., Phillips, A., and Geretti, A. (2007), "Human immunodeficiency virus (HIV) antibody

avidity testing to identify recent infection in newly diagnosed HIV type 1 (HIV-1)-seropositive persons infected with diverse HIV-1 subtypes," *Journal of clinical microbiology*, 45, 415–420.

Chow, C. (1970), "On optimum recognition error and reject tradeoff," *Information Theory, IEEE Transactions on*, 16, 41–46.

Crepaz, N., Marks, G., Mansergh, G., Murphy, S., Miller, L. C., Appleby, P. R., et al. (2000), "Age-related risk for HIV infection in men who have sex with men: examination of behavioral, relationship, and serostatus variables," *AIDS Education and Prevention*, 12, 405–415.

Eaton, L., Flisher, A. J., and Aarø, L. E. (2003), "Unsafe sexual behaviour in South African youth," *Social Science & Medicine*, 56, 149–165.

Efron, B. (1983), "Estimating the error rate of a prediction rule: improvement on cross-validation," *Journal of the American Statistical Association*, 78, 316–331.

Efron, B. and Tibshirani, R. (1993), *An introduction to the bootstrap*, vol. 57, CRC press.

Etzioni, R., Kooperberg, C., Pepe, M., Smith, R., and Gann, P. (2003), "Combining biomarkers to detect disease with application to prostate cancer," *Biostatistics*, 4, 523–538.

Greenland, S., Ericson, C., and Kleinbaum, D. G. (1983), "Correcting for misclassification in two-way tables and matched-pair studies," *International Journal of Epidemiology*, 12, 93–97.

Hargrove, J., Humphrey, J., Mutasa, K., Parekh, B., McDougal, J., Ntozini, R., Chidawanyika, H., Moulton, L., Ward, B., Nathoo, K., et al. (2008), "Improved HIV-1 incidence estimates using the BED capture enzyme immunoassay," *AIDS*, 22, 511.

Hausman, J. A., Abrevaya, J., and Scott-Morton, F. M. (1998), "Misclassification of the dependent variable in a discrete-response setting," *Journal of Econometrics*, 87, 239–269.

Janssen, R. S., Satten, G. A., Stramer, S. L., Rawal, B. D., O'Brien, T. R., Weiblen, B. J., Hecht, F. M., Jack, N., Cleghorn, F. R., Kahn, J. O., et al. (1998), "New testing strategy to detect early HIV-1 infection for use in incidence estimates and for clinical and prevention purposes," *JAMA: the journal of the American Medical Association*, 280, 42–48.

Jeske, D. R., Liu, Z., Bent, E., and Borneman, J. (2007), "Classification rules that include neutral zones and their application to microbial community profiling," *Communications in Statistics − Theory and Methods*, 36, 1965–1980.

Kaslow, R. A., Ostrow, D. G., Detels, R., Phair, J. P., Polk, B. F., Rinaldo, C. R., et al. (1987), "The Multicenter AIDS Cohort Study: rationale, organization, and selected characteristics of the participants," *American Journal of Epidemiology*, 126, 310–318.

Kim, H. (2013), "Sequential Neutral Zone Classifier," Unpublished paper presented at Quality and Productivity Research Conference 2013.

Laeyendecker, O., Brookmeyer, R., Cousins, M. M., Mullis, C. E., Konikoff, J., Donnell, D., Celum, C., Buchbinder, S. P., Seage, G. R., Kirk, G. D., et al. (2013), "HIV incidence determination in the United States: a multiassay approach," *Journal of Infectious Diseases*, 207, 232–239.

Lewbel, A. (2000), "Identification of the binary choice model with misclassification," *Econometric Theory*, 16, 603–609.

Mansergh, G. and Marks, G. (1998), "Age and risk of HIV infection in men who have sex with men," *AIDS*, 12, 1119–1128.

McIntosh, M. W. and Pepe, M. S. (2002), "Combining Several Screening Tests: Optimality of the Risk Score," *Biometrics*, 58, pp. 657–664.

Newey, W. K. and McFadden, D. (1994), "Large sample estimation and hypothesis testing," *Handbook of Econometrics*, 4, 2111–2245.

Nkengasong, J. N., Maurice, C., Koblavi, S., Kalou, M., Yavo, D., Maran, M., Bile, C., Nguessan, K., Kouadio, J., Bony, S., Wiktor, S. Z., and Greenberg, A. E. (1999), "Evaluation of HIV serial and parallel serologic testing algorithms in Abidjan, Cote d'Ivoire," *AIDS*, 13, 109–17.

Parpia, Z. A., Elghanian, R., Nabatiyan, A., Hardie, D. R., and Kelso, D. M. (2010), "p24 antigen rapid test for diagnosis of acute pediatric HIV infection," *Journal of Acquired Immune Deficiency Syndromes*, 55, 413.

Pepe, M. S. (2003), *The statistical evaluation of medical tests for classification and prediction.*, Oxford, United Kingdom: Oxford University Press.

Pettifor, A. E., Rees, H. V., Kleinschmidt, I., Steffenson, A. E., MacPhail, C., Hlongwa-Madikizela, L., Vermaak, K., and Padian, N. S. (2005), "Young people's sexual health in South Africa: HIV prevalence and sexual behaviors from a nationally representative household survey," *AIDS*, 19, 1525–1534.

Pines, H. A., Gorbach, P. M., Weiss, R. E., Shoptaw, S., Landovitz, R. J., Javanbakht, M., Ostrow, D. G., Stall, R. D., and Plankey, M. (2013), "Sexual risk trajectories among MSM in the United States: implications for pre-exposure prophylaxis delivery," *Journal of Acquired Immune Deficiency Syndromes (1999).*

Prentice, R. L. and Pyke, R. (1979), "Logistic disease incidence models and case-control studies," *Biometrika*, 66, 403–411.

Rao, R. C. (1947), "The Problem Of Classification And Distance Between Two Populations," *Nature*, 159, 30–31.

Rothman, K. J., Greenland, S., and Lash, T. L. (2008), *Modern Epidemiology*, Wolters Kluwer Health.

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003), "Logic Regression," *Journal of Computational and Graphical Statistics*, 12, 475–511.

Sáez-Cirión, A., Bacchus, C., Hocqueloux, L., Avettand-Fenoel, V., Girault, I., Lecuroux, C., Potard, V., Versmisse, P., Melard, A., Prazuck, T., et al. (2013), "Post-treatment HIV-1 controllers with a long-term virological remission after the interruption of early initiated antiretroviral therapy ANRS VISCONTI Study," *PLoS Pathogens*, 9, e1003211.

Sweeting, M. J., De Angelis, D., Parry, J., and Suligoi, B. (2010), "Estimating the distribution of the window period for recent HIV infections: a comparison of statistical methods," *Statistics in Medicine*, 29, 3194–3202.

Viana, M. (1994), "Bayesian small-sample estimation of misclassified multinomial data," *Biometrics*, 50, 237–243.

Vlahov, D., Anthony, J., Munoz, A., Margolick, J., Nelson, K., Celentano, D., Solomon, L., and Polk, B. (1991), "The ALIVE study, a longitudinal study of HIV-1 infection in intravenous drug users: description of methods and characteristics of participants." *NIDA research monograph*, 109, 75.

Wawer, M. J., Gray, R. H., Sewankambo, N. K., Serwadda, D., Li, X., Laeyendecker, O., Kiwanuka, N., Kigozi, G., Kiddugavu, M., Lutalo, T., et al. (2005), "Rates of HIV-1 transmission per coital act, by stage of HIV-1 infection, in Rakai, Uganda," *Journal of Infectious Diseases*, 191, 1403–1409.

Appendix A

Comparison of Tolerance Values in Serial Risk Score Simulation

Here we provide a comparison of the sensitivity and specificity and ACS over different tolerance values from the simulation in Section 2.3.5.3. We see from these tables that using a larger tolerance may lower costs further but may diminish accuracy.

Table A.1: Monte Carlo simulation[1] estimates for sensitivity (Sens.), specificity (Spec.) and average cost savings (ACS) over a range of tolerance values. Here we assume correlated biomarkers ($\rho = .5$) and two models for the data as specified in Section 2.3.5.3.

		t = 0			t = .05			t = .10			t = .20		
	p	Sens.	Spec.	ACS	Sens.	Spec.	ACS	Sens.	Spec.	ACS	Sens.	Spec.	ACS
(Model 1)	0.03	0.97	0.31	19%	0.97	0.29	35%	0.97	0.26	46%	0.96	0.25	55%
	0.07	0.92	0.47	17%	0.92	0.46	34%	0.92	0.44	50%	0.90	0.43	65%
	0.10	0.88	0.56	15%	0.88	0.54	35%	0.87	0.53	51%	0.86	0.52	69%
	0.30	0.65	0.81	16%	0.64	0.81	41%	0.62	0.81	58%	0.57	0.82	76%
	0.70	0.40	0.93	23%	0.38	0.94	57%	0.33	0.94	72%	0.26	0.95	83%
	1.00	0.30	0.96	32%	0.28	0.96	65%	0.24	0.97	77%	0.16	0.97	86%
	AUC =	0.81			0.80			0.79			0.77		
(Model 2)	0.03	0.95	0.24	15%	0.96	0.21	28%	0.97	0.16	39%	0.97	0.12	48%
	0.07	0.88	0.43	10%	0.88	0.41	21%	0.89	0.37	34%	0.89	0.32	52%
	0.10	0.82	0.54	8%	0.82	0.53	17%	0.82	0.49	30%	0.81	0.45	52%
	0.30	0.48	0.85	11%	0.45	0.86	27%	0.40	0.87	45%	0.28	0.90	69%
	0.70	0.19	0.97	24%	0.16	0.97	58%	0.10	0.98	77%	0.01	0.99	89%
	1.00	0.13	0.98	35%	0.09	0.98	68%	0.03	0.99	84%	0.00	1.00	90%
	AUC =	0.75			0.73			0.71			0.67		

[1] See Table 2.2 for simulation details.

Table A.2: Monte Carlo simulation[1] estimates for sensitivity (Sens.), specificity (Spec.) and average cost savings (ACS) over a range of tolerance values. Here we assume uncorrelated biomarkers ($\rho = 0.0$) and two models for the data as specified in Section 2.3.5.3.

		t = 0			t = .05			t = .10			t = .20		
	p	Sens.	Spec.	ACS	Sens.	Spec.	ACS	Sens.	Spec.	ACS	Sens.	Spec.	ACS
(Model 1)	0.03	0.96	0.26	14%	0.96	0.23	27%	0.97	0.19	38%	0.97	0.14	48%
	0.07	0.90	0.43	9%	0.90	0.41	21%	0.91	0.37	34%	0.90	0.33	51%
	0.10	0.85	0.54	8%	0.85	0.52	18%	0.85	0.48	32%	0.84	0.45	51%
	0.30	0.56	0.83	9%	0.53	0.83	25%	0.50	0.84	41%	0.40	0.86	62%
	0.70	0.28	0.95	19%	0.24	0.96	50%	0.18	0.96	69%	0.08	0.98	84%
	1.00	0.19	0.97	26%	0.16	0.98	59%	0.10	0.98	77%	0.02	0.99	88%
	AUC =	0.77			0.76			0.74			0.71		
(Model 2)	0.03	0.95	0.25	10%	0.96	0.21	20%	0.97	0.16	32%	0.98	0.07	43%
	0.07	0.88	0.43	7%	0.88	0.42	12%	0.89	0.37	20%	0.91	0.25	38%
	0.10	0.82	0.54	6%	0.81	0.54	10%	0.81	0.51	15%	0.84	0.39	32%
	0.30	0.47	0.86	7%	0.44	0.86	13%	0.39	0.87	23%	0.27	0.90	49%
	0.70	0.19	0.97	13%	0.17	0.97	36%	0.09	0.98	64%	0.00	0.99	87%
	1.00	0.12	0.98	19%	0.10	0.98	49%	0.03	0.99	75%	0.00	1.00	90%
	AUC =	0.75			0.73			0.71			0.66		

[1] See Table 2.2 for simulation details.

Appendix B

Technical Details for the Adjusted Multinomial Logistic Model

In this appendix, we derive the estimating equations for the misclassification adjusted multinomial logistic model. We list the partial derivatives of the log likelihood with respect to each β parameter below. Estimating equations are formed by setting each partial derivative to zero.

$$\begin{aligned}\frac{\partial}{\partial\beta_{00}}L = \sum_i^N & \mathcal{I}(\hat{Y}_i = 1)\frac{Se\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - (1-Sp)\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})}{Se\Phi_1(X_i;\boldsymbol{\beta}) + (1-Sp)\Phi_2(X_i;\boldsymbol{\beta})} \\ & + \mathcal{I}(\hat{Y}_i = 2)\frac{(1-Se)\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - Sp\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})}{(1-Se)\Phi_1(X_i;\boldsymbol{\beta}) + (Sp)\Phi_2(X_i;\boldsymbol{\beta})} \\ & - \mathcal{I}(\hat{Y}_i = 0)\Phi_1(X_i;\boldsymbol{\beta})\end{aligned} \tag{B.1}$$

$$\begin{aligned}\frac{\partial}{\partial\beta_{01}}L = \sum_i^N & \mathcal{I}(\hat{Y}_i = 1)\frac{X_iSe\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - X_i(1-Sp)\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})}{Se\Phi_1(X_i;\boldsymbol{\beta}) + (1-Sp)\Phi_2(X_i;\boldsymbol{\beta})} \\ & + \mathcal{I}(\hat{Y}_i = 2)\frac{X_i(1-Se)\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - X_iSp\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})}{(1-Se)\Phi_1(X_i;\boldsymbol{\beta}) + (Sp)\Phi_2(X_i;\boldsymbol{\beta})} \\ & - \mathcal{I}(\hat{Y}_i = 0)X_i\Phi_1(X_i;\boldsymbol{\beta})\end{aligned} \tag{B.2}$$

$$\frac{\partial}{\partial\beta_{10}}L = \sum_i^N \mathcal{I}(\hat{Y}_i = 1)\frac{(1-Sp)\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - Se\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})}{Se\Phi_1(X;\boldsymbol{\beta}) + (1-Sp)\Phi_2(X;\boldsymbol{\beta})}$$
$$+\mathcal{I}(\hat{Y}_i = 2)\frac{Sp\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - (1-Se)\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})}{(1-Se)\Phi_1(X;\boldsymbol{\beta}) + (Sp)\Phi_2(X;\boldsymbol{\beta})}$$
$$-\mathcal{I}(\hat{Y}_i = 0)\Phi_2(X_i;\boldsymbol{\beta}) \tag{B.3}$$

$$\frac{\partial}{\partial\beta_{11}}L = \sum_i^N \mathcal{I}(\hat{Y}_i = 1)\frac{X(1-Sp)\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - XSe\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})}{Se\Phi_1(X;\boldsymbol{\beta}) + (1-Sp)\Phi_2(X;\boldsymbol{\beta})}$$
$$+\mathcal{I}(\hat{Y}_i = 2)\frac{XSp\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - X(1-Se)\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})}{(1-Se)\Phi_1(X;\boldsymbol{\beta}) + (Sp)\Phi_2(X;\boldsymbol{\beta})}$$
$$-\mathcal{I}(\hat{Y}_i = 0)X\Phi_2(X;\boldsymbol{\beta}) \tag{B.4}$$

We also derive partial derivatives of the functions $f_y := P(\widehat{Y} = y|X)$. Monotonicity of these functions will imply identification of the multinomial logistic regression likelihood specified in Section 3.3.

$$f_0(X;\beta) = \Phi_0(X;\boldsymbol{\beta})$$
$$f_1(X;\beta) = Se\Phi_1(X;\boldsymbol{\beta}) - (1-Sp)\Phi_2(X;\boldsymbol{\beta})$$
$$f_2(X;\beta) = (1-Se)\Phi_1(X;\boldsymbol{\beta}) - (Sp)\Phi_2(X;\boldsymbol{\beta})$$

$$\begin{aligned}
\frac{\partial}{\partial\beta_{10}}f_0(X;\beta) &= -\Phi_0\Phi_1 \\
\frac{\partial}{\partial\beta_{11}}f_0(X;\beta) &= -X\Phi_0\Phi_1 \\
\frac{\partial}{\partial\beta_{20}}f_0(X;\beta) &= -\Phi_0\Phi_2 \\
\frac{\partial}{\partial\beta_{21}}f_0(X;\beta) &= -X\Phi_0\Phi_2
\end{aligned} \tag{B.5}$$

$$\frac{\partial}{\partial\beta_{00}}f_1(X;\beta) = Se\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - (1-Sp)\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{01}}f_1(X;\beta) = X_iSe\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - X_i(1-Sp)\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{10}}f_1(X;\beta) = (1-Sp)\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - Se\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{11}}f_1(X;\beta) = X(1-Sp)\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - XSe\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta}) \tag{B.6}$$
$$\frac{\partial}{\partial\beta_{00}}f_2(X;\beta) = (1-Se)\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - Sp\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{01}}f_2(X;\beta) = X_i(1-Se)\Phi_1(X_i;\boldsymbol{\beta})(1-\Phi_1(X_i;\boldsymbol{\beta})) - X_iSp\Phi_1(X_i;\boldsymbol{\beta})\Phi_2(X_i;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{10}}f_2(X;\beta) = Sp\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - (1-Se)\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta})$$
$$\frac{\partial}{\partial\beta_{11}}f_2(X;\beta) = XSp\Phi_2(X;\boldsymbol{\beta})(1-\Phi_2(X;\boldsymbol{\beta})) - X(1-Se)\Phi_1(X;\boldsymbol{\beta})\Phi_2(X;\boldsymbol{\beta}) \tag{B.7}$$

Appendix C

R Source Code Listings

The following code is provided to generate a serial risk score classification rule as well as make predictions for a serial risk score classification rule.

```
library(gtools) # permutations
library(caTools)

sensspec <- function( true, guess ) {
# Description: Calculate sensitivity, specificity,
##             true positives (tp), true negatives (tn), false
##             positives (fp) and false negatives (fn)
# Usage: sensspec( true, guess )
# Arguments: true: Vector of true values in binary test
##          guess: Vector of estimated values of binary test

# Output: List of sensitivity (sens) and specificity (spec) and
## a vector of true positives (tp), true negatives (tn),
## false positives (fp) and false negatives (fn).

  sens <- sum( as.integer( true == 1 & guess == 1) ) / sum( as.
     integer( true == 1 ))
  spec <- sum( as.integer( true == 0 & guess == 0) ) / sum( as.
     integer( true == 0 ))
  tp <- sum( as.integer( true == 1 & guess == 1) )
  tn <- sum( as.integer( true == 0 & guess == 0) )
  fp <- sum( as.integer( true == 0 & guess == 1) )
  fn <- sum( as.integer( true == 1 & guess == 0) )
```

```
  list( sens = sens, spec = spec, c = c(tp,tn,fp,fn) )
}



srsc <- local({
# Description: Generate a serial risk score classification rule set
##             from training data. This function writes a function
##             for some number of biomarkers and then evaluates the
##             function

#Usage:
## srsc( data, outcome, markers, costs, outputfile )

#Arguments:
## data: Data frame from which to train rules
## outcome: String with name of the outcome variable in data
## markers: A string vector contain the names of the markers in data
## costs: A vector of costs corresponding to markers
## outputfile: name of file to write set of rules

##Output:
# A set of serial risk score classification rules are written to
# outputfile

buildFunction <- function(numMarkers ) {
  strmodels <- paste( paste( rep("model.", numMarkers),
                seq( 1, numMarkers), sep = "" ), collapse = "," )
  strthresholds <- paste( rep("crs.", numMarkers - 1),
                    seq( 1, numMarkers - 1 ), sep = "" )
  strthresholds <-  c( strthresholds, strthresholds)
  strthresholds <- strthresholds[ order( strthresholds ) ]
  strthresholds <- paste( c( paste( strthresholds, c(".1", ".2"),
    sep = "" ), paste("crs.", numMarkers, sep = "")), collapse = ",")
```

```
  strsetup <- paste( "
    numcols <- length(markers)
    numclass <- rep( 0, numcols )
    perms <- permutations(numcols,numcols)
    divisions <- 10
    cat(\"id,sens,spec,tp,tn,fp,fn,avgcost,", strmodels,",",
      strthresholds, "\\n\", file = outputfile, append = FALSE)
    data$class <- 0
    neutral.0 <- data
    for(i in 1:nrow(perms) ) {
  ", sep = "")
  strbody <- recurseFunction( numMarkers, numMarkers )
  strend <- "}"
  paste( strsetup, strbody, strend, sep = "")
}

recurseFunction <- function( numMarkers, totalMarkers  ) {
  if( numMarkers < 2) {
    strbegin <- "
      df.%d <- get( paste( \"neutral.\", %d - 1, sep = \"\") )
      y <- with(  df.%d, get( outcome ))
      testlist <- list()
      for( testidx in seq_len( %d ) ) {
        testlist[[ testidx ]] <- with( df.%d, get(
          markers[ perms[i, testidx] ] ))
      }
      logt.formula <- as.formula(paste(\"y ~ \",
        paste( \"testlist[[\", seq( 1, %d ) , \"]]\",
          collapse = \"+\" )))
      if( nrow( df.%d) - 2 > %d ) {
        logt.model <- tryCatch( glm( logt.formula,
            family=binomial(link=\"logit\"), na.action=na.omit ),
          error=function(e) e, warning=function(w) w)
```

```
    if( is( logt.model, \"warning\") ) {
      rs.%d <- rep( 0, nrow( df.%d ))
      thresholdset.%d <- c(0,1)
      coefs.%d <- c(1)
    } else {
      rs.%d <- fitted.values( logt.model )
      coefs.%d <- coef( logt.model )
      thresholdset.%d <- as.numeric(c(0, quantile( rs.%d,
        probs = seq(0,1,1/divisions) ),1))
    }
  } else {
        rs.%d <- rep( 0, nrow( df.%d ))
        thresholdset.%d <- c(0,1)
        coefs.%d <- c(1)
  }
  model.%d <- base64encode( unname( coefs.%d ) )
  for(i.%d in seq_len( length(thresholdset.%d)) ) {
    crs.%d <- thresholdset.%d[i.%d]

    classneg.%d <- df.%d[(rs.%d < crs.%d), ]
    classpos.%d <- df.%d[(rs.%d >= crs.%d), ]
    classneg.%d <- classneg.%d[ complete.cases( classneg.%d), ]
    classpos.%d <- classpos.%d[ complete.cases( classpos.%d), ]

    if( nrow(classneg.%d) > 0 ) { classneg.%d$class <- 0}
    if( nrow(classpos.%d) > 0 ) { classpos.%d$class <- 1}
    class.%d <- rbind( classneg.%d, classpos.%d )
    numclass[ %d ] <- nrow( class.%d )
"
strbegin <- gsub( "%d",
    as.character( totalMarkers - numMarkers + 1), strbegin )
strrecurse <- paste( "df <- rbind(",
                     paste( paste( rep("class.", totalMarkers),
                     seq( 1, totalMarkers), sep = "" ),
```

```
                    collapse = "," ), ")" )

strmodels <- paste( paste( rep("model.", totalMarkers),
                    seq( 1, totalMarkers), sep = "" ),
                   collapse = "," )
strthresholds <- paste( rep("crs.", totalMarkers - 1),
    seq( 1, totalMarkers - 1 ), sep = "" )
strthresholds <-  c( strthresholds, strthresholds)
strthresholds <- strthresholds[ order( strthresholds ) ]
strthresholds <- paste( c( paste( strthresholds,
                    c(".1", ".2"), sep = "" ),
        paste( "crs.", totalMarkers, sep = "")), collapse = ",")

strend <- paste( "
    L <- sensspec( with(df, get( outcome )), df$class )
    cost <- cumsum( costs[perms[i,]] ) %*% numclass
    avgcost <- cost / nrow( df )
    outtmp <- paste( i,
                    L$sens, L$spec,
                    L$c[1], L$c[2], L$c[3], L$c[4],
                    avgcost", strmodels, strthresholds,
                    "sep = \",\")
    cat(outtmp, file = outputfile, append = TRUE)
    cat(\"\\n\", file = outputfile, append = TRUE)
    }", sep = ",")
} else {
 strbegin <- "
   df.%d <- get( paste( \"neutral.\", %d - 1, sep = \"\") )
   y <- with(  df.%d, get( outcome ))
   testlist <- list()
   for( testidx in seq_len( %d ) ) {
       testlist[[ testidx ]] <- with( df.%d, get(
        markers[ perms[i, testidx] ] ))
   }
```

```
logt.formula <- as.formula(paste(\"y ~ \",
  paste( \"testlist[[\", seq( 1, %d ) , \"]]\",
    collapse = \"+\" )))
if( nrow( df.%d) - 2 > %d ) {
    logt.model <- tryCatch( glm( logt.formula,
      family=binomial(link=\"logit\"), na.action=na.omit ),
       error=function(e) e, warning=function(w) w)
    if( is( logt.model, \"warning\") ) {
        rs.%d <- rep( 0, nrow( df.%d ))
        thresholdset.%d <- c(0,1)
        coefs.%d <- c(1)
    } else {
        rs.%d <- fitted.values( logt.model )
        coefs.%d <- coef( logt.model )
        thresholdset.%d <- as.numeric(c(0, quantile( rs.%d,
           probs = seq(0,1,1/divisions) ),1))
    }
} else {
    rs.%d <- rep( 0, nrow( df.%d ))
    thresholdset.%d <- c(0,1)
    coefs.%d <- c(1)
}
 revthresholdset.%d <- thresholdset.%d[ order(thresholdset.%d,
     decreasing = T)]
 model.%d <- base64encode( unname( coefs.%d ) )
 for( i.%d.1 in  seq_len(length(thresholdset.%d)/2 + .5)) {
 for( i.%d.2 in  seq_len(length(thresholdset.%d) - i.%d.1 + 1)
    ) {
     crs.%d.1 <- thresholdset.%d[i.%d.2]
     crs.%d.2 <- revthresholdset.%d[i.%d.1]

     classneg.%d <- df.%d[(rs.%d < crs.%d.1),]
     classpos.%d <- df.%d[(rs.%d >= crs.%d.2),]
```

```
            classneg.%d <- classneg.%d[ complete.cases( classneg.%d),
                ]
            classpos.%d <- classpos.%d[ complete.cases( classpos.%d),
                ]

            if( nrow(classneg.%d) > 0 ) { classneg.%d$class <- 0}
            if( nrow(classpos.%d) > 0 ) { classpos.%d$class <- 1}
            class.%d <- rbind( classneg.%d, classpos.%d )
            numclass[ %d ] <- nrow( class.%d )

            neutral.%d <- df.%d[ (rs.%d >= crs.%d.1 & rs.%d < crs.%d
                .2 ), ]
        "
        strbegin <- gsub( "%d", as.character( totalMarkers -
            numMarkers + 1 ), strbegin )
        strrecurse <- Recall( numMarkers - 1, totalMarkers )
        strend <- "}}"
    }
    paste( strbegin, strrecurse, strend )
}

f <- function( data, outcome, markers, costs, outputfile ) {
    tmpcode <- buildFunction( length(markers) )
    eval( parse( text = tmpcode))
}
})

predict.srsc <- function( row, newdata, outcome, markers, costs) {
# Description: Classify newdata with a serial risk score
##             classification rule
# Usage: predict.srsc( row, newdata, outcome, markers, costs)

# Arguments:
## row: Rule from srsc rule set
```

```
## newdata: Data frame of data to be classified
## outcome: String with name of the outcome variable in data
## markers: A string vector contain the names of the markers in data
## costs: A vector of costs corresponding to markers

##Output:
# A list 'acc' = a list of sensitivity, specificity values,
## 'avgcost' = cost of classification,
## 'data' = the data frame newdata with a
##          classification vector 'class'.
  numMarkers <- length(markers)
  perms <- permutations(numMarkers,numMarkers)
  df <- newdata
  df$class <- 0
  numclass <- rep(0, numMarkers )

  crs1vec <- rep(0, numMarkers)
  crs2vec <- rep(0, numMarkers)
  for( modelidx in seq_len( numMarkers - 1 ) ) {
      crs1vec[modelidx] <- with( row, get( paste( "crs.", modelidx, "
          .1", sep = "")))
      crs2vec[modelidx] <- with( row, get( paste( "crs.", modelidx, "
          .2", sep = "")))
  }
  crs1vec[numMarkers] <- with( row, get( paste( "crs.", numMarkers,
     sep = "")))
  crs2vec[numMarkers] <- with( row, get( paste( "crs.", numMarkers,
     sep = "")))

  coeflist <- list()
  classneg <- list()
  classpos <- list()
  class <- list()
```

```
for( modelidx in seq_len(numMarkers) ) {
  coeflist[[modelidx]] <- base64decode( with( row, get( paste( "
     model.", modelidx, sep = ""))), "double" )
  if( nrow( df ) > 0 ) {
      y <- with(  df, get( outcome ))
      modelmat <- matrix( nrow = nrow(df), ncol = modelidx + 1)
      modelmat[,1] <- rep(1, nrow(df ))
      for( testidx in seq_len( modelidx ) ) {
          tmptest <- with( df, get( markers[ perms[row$id, testidx]
              ] ))
          modelmat[,testidx+1] <- tmptest
      }
      if( length( coeflist[[modelidx]] ) > 1 ) {
          y_hat <- modelmat %*% coeflist[[modelidx]]
          rs <-  exp(y_hat)/(1 + exp(y_hat))
      } else {
          rs <- rep( 0, nrow( df ) )
      }

      classneg[[modelidx]] <- df[ (rs < crs1vec[modelidx]), ]
      classpos[[modelidx]] <- df[ (rs >= crs2vec[modelidx]), ]
      classneg[[modelidx]] <- classneg[[modelidx]][ complete.cases(
          classneg[[modelidx]]), ]
      classpos[[modelidx]] <- classpos[[modelidx]][ complete.cases(
          classpos[[modelidx]]), ]

      if( nrow(classneg[[modelidx]]) > 0 ) {classneg[[modelidx]]$
         class <- 0}
      if( nrow(classpos[[modelidx]]) > 0 ) {classpos[[modelidx]]$
         class <- 1}
      class[[modelidx]] <- rbind( classneg[[modelidx]], classpos[[
         modelidx]])
      numclass[ modelidx ] <- nrow( class[[modelidx]] )
```

```
        df <- df[ (rs >= crs1vec[modelidx] & rs < crs2vec[modelidx] )
           , ]
        df <- df[ complete.cases( df ), ]
    } else {
        class[[modelidx]] <- df
        numclass[ modelidx ] <- nrow( class[[modelidx]] )
    }
  }
  data <- do.call( "rbind", class )
  L <- sensspec( with(  data, get( outcome )), data$class )

  cost <- cumsum( costs[perms[row$id,]] ) %*% numclass
  avgcost <- cost / nrow(data)
  list(acc = L, avgcost =avgcost, data=data)
}



srsc2biomarkers <- function( data, outcome, markers, costs,
   outputfile ) {
##Description: SRSC algorithm with a fixed number of biomarkers (2)
##Usage:
#srsc2biomarkers( data, outcome, markers, costs, outputfile )

##Arguments:
# data: Data frame from which to train rules
# outcome: String with name of the outcome variable in data
# markers: A string vector contain the names of the markers in data
# costs: A vector of costs corresponding to markers
# outputfile: name of file to write set of rules

##Output:
```

```
# A set of serial risk score classification rules are written to
    outputfile
    numcols <- length(markers)
    numclass <- rep( 0, numcols )
    perms <- permutations(numcols,numcols)
    divisions <- 10
    cat("id,sens,spec,tp,tn,fp,fn,avgcost,model.1,model.2,crs.1.1,crs
        .1.2,crs.2\n", file = outputfile, append = FALSE)
    data$class <- 0
    neutral.0 <- data
    for(i in 1:nrow( perms ) ) {
        df.1 <- get( paste( "neutral.", 1 - 1, sep = "") )
        y <- with(  df.1, get( outcome ))
        testlist <- list()
        for( testidx in seq_len( 1 ) ) {
            testlist[[ testidx ]] <- with( df.1, get( markers[ perms[
                i, testidx] ] ))
        }
        logt.formula <- as.formula(paste("y ~ ",paste( "testlist[[",
            seq( 1, 1 ) , "]]", collapse = "+" )))
        if( nrow( df.1) - 2 > 1 ) {
            logt.model <- tryCatch( glm( logt.formula, family=
                binomial(link="logit"), na.action=na.omit ), error=
                function(e) e, warning=function(w) w)
            if( is( logt.model, "warning") ) {
                rs.1 <- rep( 0, nrow( df.1 ))
                thresholdset.1 <- c(0,1)
                coefs.1 <- c(1)
            } else {
                rs.1 <- fitted.values( logt.model )
                coefs.1 <- coef( logt.model )
                thresholdset.1 <- as.numeric(c(0, quantile( rs.1,
                    probs = seq(0,1,1/divisions) ),1))
            }
```

```
} else {
    rs.1 <- rep( 0, nrow( df.1 ))
    thresholdset.1 <- c(0,1)
    coefs.1 <- c(1)
}
revthresholdset.1 <- thresholdset.1[ order(thresholdset.1,
   decreasing = T)]
model.1 <- base64encode( unname( coefs.1 ) )
for( i.1.1 in  seq_len(length(thresholdset.1)/2 + .5)) {
    for( i.1.2 in  seq_len(length(thresholdset.1) - i.1.1 +
       1)) {
        crs.1.1 <- thresholdset.1[i.1.2]
        crs.1.2 <- revthresholdset.1[i.1.1]
        classneg.1 <- df.1[(rs.1 < crs.1.1),]
        classpos.1 <- df.1[(rs.1 >= crs.1.2),]
        classneg.1 <- classneg.1[ complete.cases( classneg.1)
           , ]
        classpos.1 <- classpos.1[ complete.cases( classpos.1)
           , ]
        if( nrow(classneg.1) > 0 ) { classneg.1$class <- 0 }
        if( nrow(classpos.1) > 0 ) { classpos.1$class <- 1 }
        class.1 <- rbind( classneg.1, classpos.1 )
        numclass[ 1 ] <- nrow( class.1 )
        neutral.1 <- df.1[ (rs.1 >= crs.1.1 & rs.1 < crs.1.2
           ), ]

        df.2 <- get( paste( "neutral.", 2 - 1, sep = "") )
        y <- with(  df.2, get( outcome ))
        testlist <- list()
        for( testidx in seq_len( 2 ) ) {
            testlist[[ testidx ]] <- with( df.2, get( markers
               [ perms[i, testidx] ] ))
        }
```

```
logt.formula <- as.formula(paste("y ~ ",paste( "
   testlist[[", seq( 1, 2 ) , "]]", collapse = "+" )
   ))
if( nrow( df.2) - 2 > 2 ) {
    logt.model <- tryCatch( glm( logt.formula, family
       =binomial(link="logit"), na.action=na.omit ),
        error=function(e) e, warning=function(w) w)
    if( is( logt.model, "warning") ) {
        rs.2 <- rep( 0, nrow( df.2 ))
        thresholdset.2 <- c(0,1)
        coefs.2 <- c(1)
    } else {
        rs.2 <- fitted.values( logt.model )
        coefs.2 <- coef( logt.model )
        thresholdset.2 <- as.numeric(c(0, quantile(
           rs.2, probs = seq(0,1,1/divisions) ),1))
    }
} else {
    rs.2 <- rep( 0, nrow( df.2 ))
    thresholdset.2 <- c(0,1)
    coefs.2 <- c(1)
}
model.2 <- base64encode( unname( coefs.2 ) )
for(i.2 in seq_len( length(thresholdset.2)) ) {
    crs.2 <- thresholdset.2[i.2]
    classneg.2 <- df.2[(rs.2 < crs.2), ]
    classpos.2 <- df.2[(rs.2 >= crs.2), ]
    classneg.2 <- classneg.2[ complete.cases(
       classneg.2), ]
    classpos.2 <- classpos.2[ complete.cases(
       classpos.2), ]
    if( nrow(classneg.2) > 0 ) { classneg.2$class <-
       0}
```

```
                    if( nrow(classpos.2) > 0 ) { classpos.2$class <-
                        1}
                    class.2 <- rbind( classneg.2, classpos.2 )
                    numclass[ 2 ] <- nrow( class.2 )
                    df <- rbind( class.1,class.2 )
                    L <- sensspec( with(df, get( outcome )), df$class
                        )
                    cost <- cumsum( costs[perms[i,]] ) %*% numclass
                    avgcost <- cost / nrow( df )
                    outtmp <- paste( i,
                                    L$sens, L$spec, L$c[1], L$c[2], L
                                        $c[3], L$c[4], avgcost,
                                    model.1,model.2,crs.1.1,crs.1.2,
                                        crs.2,sep = ",")
                    cat(outtmp, file = outputfile, append = TRUE)
                    cat("\n", file = outputfile, append = TRUE)
                }
            }
        }
    }
}
```