Title
Sequential Neutral Zone Classiers With Application to Longitudinal Data

Permalink
https://escholarship.org/uc/item/81v8r0w4

Author
Kim, Hyunkyoung

Publication Date
2015

Peer reviewed|Thesis/dissertation
Sequential Neutral Zone Classifiers With Application to Longitudinal Data

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Hyunkyoung Kim

August 2015

Dissertation Committee:

Dr. Daniel R. Jeske, Chairperson
Dr. Shujie Ma
Dr. James Borneman
The Dissertation of Hyunkyoung Kim is approved:

____________________________________

____________________________________

____________________________________

Committee Chairperson

University of California, Riverside
Acknowledgments

My most sincere thanks go to my advisor, Professor Daniel R. Jeske for his excellent guidance and endless encouragement during the time of my research. I appreciate all his contributions of time and effort to read my paper writing. I also want to thank him for providing me great opportunities to give talks, to receive a dissertation year program fellowship, and to work in the statistical consulting collaboratory. Without his support, I would have not completed my Ph.D. work successfully.

My grateful appreciation goes to Prof. Shujie Ma and Prof. James Borneman for serving in my doctoral committee with valuable advise. I am also very grateful to Prof. Jun Li for giving me insightful comments in my oral exam. I would like to give her special thanks for her time to write recommendation letters for me. I would like to thank Prof. Tae-Hwy Lee for offering me an opportunity to speak a talk in the department of Economics. I am also thankful to Prof. Subir Ghosh for his distinguished teaching and for providing me the opportunity to join this PhD program. I am grateful to Perla Fabelo and Paula G. Lemire for their endless support and all generous help. I would like to express my sincere gratitude to Jill Smith and Linda Penas for providing me a wonderful teaching experience. My grateful appreciation goes to all my friends in the department of Statistics. I will always remember their friendship. Especially, I wish to thank my wonderful officemates Fei He and Shangjie Xu. Lastly, I wish to thank my husband Insu Kim, my son Ujin and my parents for their unconditional love and endless support.
To my husband, my son, my parents for all the support.
ABSTRACT OF THE DISSERTATION

Sequential Neutral Zone Classifiers With Application to Longitudinal Data

by

Hyunkyoung Kim

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, August 2015
Dr. Daniel R. Jeske, Chairperson

Neutral zone classifiers include 'no-decision' as a classification outcome. This paper extends neutral zone classifiers to sequential contexts for analyzing longitudinal data. Applications could include medical diagnosis where a decision variable is repeatedly measured on each subject with the expectation of being able to ultimately identify a patient disease status. Sequential classifiers monitor the sequence of measurements and decide when to stop sampling and how to classify the subject. Bayesian sequential classification rules make classifications on the basis of the minimum expected loss. This approach is a challenge due to computational complexity associated with evaluating the expected future costs. We consider Gaussian contexts. In the homogeneous case we demonstrate the equivalence between sequential Bayesian classifier and a simpler boundary-based framework. A solution for the heterogeneous Gaussian case is presented using the boundary-based framework. A recursive algorithm is developed to efficiently determine decision boundaries that minimize the overall expected cost. Alternative decision boundaries which are competitive with the optimal decision boundaries are studied. Misclassification rates and expected sample size
are investigated and the results are compared with non-sequential classifiers.
Contents

List of Figures x

List of Tables xii

1 Introduction 1
  1.1 Classifications with Neutral Zone .................................. 1
  1.2 Motivation to Sequential Classification ................................ 2
  1.3 Sequential Decision Procedures ....................................... 3
  1.4 Structure ........................................................................ 4

2 Non-Sequential Classifiers 6
  2.1 Bayes classifier ............................................................. 6
  2.2 Non-Sequential Neutral Zone classifier .................................. 8
  2.3 Examples ....................................................................... 11

3 Sequential Neutral Zone Classifier 13
  3.1 Framework of Sequential Classification ................................. 13
  3.2 Posterior Expected Loss Based Classification ......................... 16
  3.3 Example of PEL-Based Classification ................................... 21

4 Sequential Classifications for Gaussian Variable 25
  4.1 Homogeneous Gaussian Case .............................................. 25
     4.1.1 Boundary-based Classifier ......................................... 25
     4.1.2 Transformation of Distribution .................................... 27
     4.1.3 Relationship Between Two Classification Approaches ........... 31
     4.1.4 Recursive Algorithm .................................................. 35
     4.1.5 Heterogeneous Gaussian Case ..................................... 37
  4.2 Alternative Decision Boundaries ........................................... 39
     4.2.1 Local Sequential Neutral Zone Classifier (LSNZ) ............... 39
     4.2.2 Simplified Sequential Neutral Zone Classifier (SNZ3) ............ 40
  4.3 Illustrative Comparisons .................................................... 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Application</td>
<td>51</td>
</tr>
<tr>
<td>5.1 Application to Kidney Dysfunction Data</td>
<td>51</td>
</tr>
<tr>
<td>5.2 Application to Sepsis Data</td>
<td>57</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>59</td>
</tr>
<tr>
<td>6.1 Conclusion</td>
<td>59</td>
</tr>
<tr>
<td>6.2 Future Work</td>
<td>60</td>
</tr>
<tr>
<td>Bibliography</td>
<td>62</td>
</tr>
<tr>
<td>A Proofs</td>
<td>64</td>
</tr>
<tr>
<td>A.1 Equivalence of Two Expected Costs in Chapter 3</td>
<td>64</td>
</tr>
<tr>
<td>A.2 Proof of Theorem 1 in Chapter 4</td>
<td>65</td>
</tr>
<tr>
<td>A.3 Proof of Theorem 2 in Chapter 4</td>
<td>66</td>
</tr>
<tr>
<td>A.4 Concavity of PEL of Neutral Zone</td>
<td>68</td>
</tr>
<tr>
<td>B Backward Induction Examples</td>
<td>70</td>
</tr>
<tr>
<td>B.1 Two-Stage Case</td>
<td>70</td>
</tr>
<tr>
<td>B.2 Four-Stage Case</td>
<td>73</td>
</tr>
<tr>
<td>C Supplementary Materials</td>
<td>85</td>
</tr>
<tr>
<td>C.1 Manual</td>
<td>85</td>
</tr>
<tr>
<td>C.2 Codes</td>
<td>87</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Decision boundaries for the non-sequential neutral zone classifiers . . . . . . 12

4.1 Example of decision boundaries for the posterior probability: Assumptions: $K = 5$, $\pi_0 = \pi_1 = 0.5$, $C_{01} = C_{10} = 10$, $C_{Ny} = 1$, $\mu_0^i = 0$, $\mu_1^i = 2$ and the covariance matrices $\Sigma_y = [\rho_{ij}]$ are compound symmetry with $\rho_{ij} = 0.1$ if $i \neq j$ and $\rho_{ii} = 1$ for class 0 and $\rho_{ii} = 1.8$ for class 1, for all $1 \leq i \neq j \leq K$ and $y \in \{0, 1\}$ ................................................................. 26

4.2 The PELs against the posterior probability. Assumptions: $K = 3$, $\pi_0 = \pi_1 = 0.5$, $C_{01} = C_{10} = 10$, $C_{Ny} = 1$, $\mu_0^i = 0$, $\mu_1^i = 2$ and the covariance matrices $\Sigma_y = [\rho_{ij}]$ are compound symmetry with $\rho_{ij} = 0.1$ if $i \neq j$ and $\rho_{ii} = 1$ for both classes .................................................. 32

4.3 In case where neutral zone exists, the PEL function against the posterior probability(left) and against the weighted sum(right). Assumptions in Figure 4.2 is used ................................................................. 34

4.4 In case where neutral zone do not exist, example of decision boundaries for the posterior probability. Assumptions in Figure 4.2 is used ................................................................. 34

4.5 In heterogeneous Gaussian case, the PELs against the posterior probability. From the assumptions in Figure 4.2, $\Sigma_1$ is changed to AR(1) structure with $\rho_{ij} = 0.1|t-j|$ if $i \neq j$ and $\rho_{ii} = 1$. For plot (b), the mean profiles are changed to $\mu_0 = (0.2, 0.4, 0.6)$, $\mu_1 = (2.1, 1.9, 1.7)$ and the misclassification costs are changed to $C_{01} = C_{10} = 30$ for all $i$ ................................................................. 38

4.6 Decision boundaries and performance evaluations for SNZ .......................... 41

4.7 Classification Simulation. 200 samples were generated from the distribution under the default setting ................................................................. 42

4.8 Example of decision boundaries for the posterior probability ......................... 44

4.9 For homogeneous Gaussian, comparisons of decision boundaries. (a) uses the default setting and (b) uses $\mu_0 = (1.0, 0.8, 0.6, 0.4, 0.2)$ and $\mu_1 = (1.3, 1.5, 1.7, 1.9, 2.1)$. 45

4.10 Decision boundaries under different covariance structure. Assume $\rho_{ii} = 1$ for all cases, and assume $\rho_{ij} = 0.3$ for CS and $\rho_{ij} = 0.3|t-j|$ for AR(1), and $\rho_{ij} = 0$ for IID case, $1 \leq i \neq j \leq K$ ................................................................. 46
4.11 Decision boundaries under the default setting but changing cost structures. For (a) and (d), $C_{01}^i = C_{10}^i = 10$, for (b), $C_{01}^i = C_{10}^i = 30$, for (c) $C_{01}^i = 30$ and $C_{10}^i = 10$. For (d), $C_{N0}^i = C_{N1}^i = 1.2^i$ and the others used $C_{N0}^i = C_{N1}^i = 1$.

4.12 Comparisons of recursive vs. Simultaneous optimization algorithm. Plot (a) used and $K = 10$ (right).

4.13 Verification of the expected performance measures. The expected error rate and stopping proportion are compared with the simulated error rate and stopping proportion.

4.14 For heterogeneous Gaussian, comparisons of decision boundaries and performance evaluations. From the default setting, change $\Sigma_1 = [\rho_{ij}]$ to have AR(1) structure with $\rho_{ii} = 2$ and $\rho_{ij} = 0.3^{|i-j|}$. $1 \leq i \neq j \leq K$.

5.1 Biomarker profiles of 200 patients in 'No dysfunction' group and 'dysfunction' group.

5.2 Decision boundaries and classification performance for kidney dataset.

5.3 Decision boundaries and performance evaluations for the kidney dataset.

5.4 Cross validated performance evaluation for the kidney dysfunction data.

5.5 Sepsis data set: biomarker profiles of 60 patients in each group where the blue line is for non-severe sepsis and the red line is for severe sepsis. Training data patients sampled at 0, 3, and 6 hours and connected by line segments solely for viewing purposes.

A.1 Minimum future risks against the posterior probability.
List of Tables

4.1 Comparisons of classification performance. (a) uses the default setting and (b) uses the diverging mean profiles in Figure 3. . . . . . . . . . . . . . . . . 45

5.1 Cross-validated performance evaluation for the kidney dataset. . . . . . . . . 55
Chapter 1

Introduction

1.1 Classifications with Neutral Zone

A classifier is a decision rule that maps a subject into a class based on measurements of attribute variables. Consider a binary classification where the outcome space for measurements has overlap for the two classes. It can be difficult to make a confident decision in this case because the evidence for one decision may not be considerably higher than for the other. Neutral zone classifiers enable us to handle ambiguity in decision making by assigning 'no-classification', under the belief that making no decision is better than an incorrect decision. [12] introduced the concept of a minimal cost neutral zone classifier as a modification of the classical Bayes classifier, and [3] derived the explicit form of the Bayes neutral zone classifier, which is eventually equivalent to [11]. [14], [7], and [1] studied classification rules of this type by maximizing the probability of correct classification subject to target error rates. All of these classifiers have dealt only with a single decision based on all possible data. In what follows, these classifiers are called 'non-sequential neutral zone...
1.2 Motivation to Sequential Classification

In this dissertation, we focus on classifying longitudinal profiles where individuals are repeatedly measured at different time points, but the maximum number of measurements is finite and typically not very large. In this case, the non-sequential classifier is not efficient due to the fact that it always delays a decision until all the measurements are collected, even when partial information has sufficient evidence to make a confident decision. Instead, we employ a sequential classification procedure which operates on the measurements sequentially and provides an opportunity for a faster decision.

To be more specific, we illustrate with a medical diagnosis example. If a patient is easily classifiable based on the first sample, we assign the patient accordingly and terminate the procedure, otherwise we assign the patient to a neutral zone and take another sample. The process is repeated using the accumulated data at each step, until a confident classification to one of the groups can be made, or until the maximum number of samples is collected. One can easily see that the sample size in this sequential decision scheme depends on the outcomes of each sample, but will be less than or equal to the maximum sample size. Therefore, clinicians can make faster decisions and reduce the overall cost incurred by the classification procedure, and thinking ethically, patients can receive superior treatment earlier and benefit from reduced inconvenience of over examination.
1.3 Sequential Decision Procedures

The Wald sequential probability ratio test (SPRT) controls type I and type II errors and can be used as a sequential classification technique if there is no concern about the indefiniteness of the required number of samples for a longitudinal profile. A truncated SPRT could be also used to impose a maximum number of samples. Truncated SPRTs ([17] and [2]) retain approximate type I and type II error rates provided the truncation is done at a sufficiently large sample number, which typically would be a value too large for most longitudinal data contexts. One could try to develop a finite stage sequential classification procedure that controls type I and type II error, but there is no guarantee such a procedure exists when the number of stages is small, such as in longitudinal data contexts. Moreover, if a compromise is made to only seek the minimum overall error rate, we are led to a non-sequential procedure that does not make a decision until all the data is collected.

Bayesian approaches attempt to deal with finite sequential decision problems by minimizing the overall expected cost evaluated under a loss function. The optimal Bayesian sequential decision procedure chooses the decision that minimizes the posterior expected loss at each decision stage (see, for example, [8] and [4]). The calculation of the posterior expected loss incurred by the continuation sampling is done by backward induction which is computationally complex and extensive. A simpler approach is to determine the decision boundaries for the posterior probabilities of class memberships in advance then a sequential classification is performed according to the priori-selected decision boundaries.

It turns out that our decision boundaries turn out to have similar shapes of stop-
ping boundaries of group sequential designs in clinical trials ([13] and [5]). One of differences from the group sequential designs is that classification problems are simple versus simple hypothesis tests whereas for clinical trials are usually simple versus composite hypothesis tests. In addition, classification problems take 1 sample at each stage, whereas for group sequential designs take \( m \) samples for each arm at each stage.

An unpublished related work is [15], where the decision boundary-based classification procedures are studied for two-stage sequences of cervical cancer tests. In our paper we provide explicit details for handling multistage problems. The main contributions of this dissertation are: 1) a finite stage sequential neutral zone classification methodology under a general cost structure, 2) a clearly explained utilization of the backward induction technique, 3) appealing graphical implementations of the classification methodology, and 4) an \texttt{R} function that provides an efficient computational tool for computing sequential classified decision boundaries in a Gaussian setting.

### 1.4 Structure

The rest of this dissertation is organized as follows. In chapter 2, we present non-sequential Bayes classifiers. The classic Bayes classifier and the existing neutral zone classifier are reviewed. In chapter 3, we derive the Bayes sequential classifier using backward induction and provide an example that illustrates the computational aspects of the classifier. In chapter 4, we discuss a simpler classifier that makes a decision based on decision boundaries under homogeneous and heterogeneous Gaussian distributions. In the former case, we provide a recursive algorithm for obtaining the decision boundaries. We
also introduce additional alternative classification approaches which might be considered. We also present illustrative examples to compare the different approaches. In chapter 5, we illustrate two applications of the proposed classifiers. We conclude in chapter 6 with a summary and future works.
Chapter 2

Non-Sequential Classifiers

2.1 Bayes classifier

We briefly review the Bayes classifier before explaining neutral zone classifiers. Suppose $X_K = (X_1, \cdots, X_K)$ is a vector of the longitudinal measurements belonging to group $y \in \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$ is a set of class labels, with the probability density function $f_y(X_K)$. After observing all measurements $x_K$, a Bayes classifier, denoted by $\delta_B$, allocates a subject as either groups in $A = \{0, 1\}$. Assume $\pi_y$ is the prior probability of class $y$. According to the Bayes rule, the posterior probability of class $y$ given observed measurements $x_K$ is

$$p_y(x_K) = \frac{f_y(x_K) \pi_y}{m(x_K)} \quad (2.1)$$

where $m(x_K) = \sum_{y \in \mathcal{Y}} f_y(x_K) \pi_y$.

Let $C_{dy}$ be the cost incurred by decision $d$ when the true class is $y$. The cost of a correct decision is assumed to be zero, i.e., $C_{dy} = 0$ for $d = y$. The expected cost incurred
by a classifier $\delta$ is formulated as

$$EC(\delta) = \sum_{d \in A} \sum_{y \in Y} \int_{\{x_K : \delta = d\}} [C_{dy}f_y(x_K)\pi_y] dx_K$$

(2.2)

Rewriting the above expected cost in terms of the posterior probability, we have

$$EC(\delta) = \int_{\{x_K : \delta = 1\}} [C_{10}p_0(x_K)] m(x_K) dx_K + \int_{\{x_K : \delta = 0\}} [C_{01}p_1(x_K)] m(x_K) dx_K$$

(2.3)

Denote $R(1|x_K) = C_{10}p_0(x_K)$ and $R(0|x_K) = C_{01}p_1(x_K)$ which are referred to as the posterior expected losses (PEL) incurred by decision 1 ($d = 1$) and 0 ($d = 0$), respectively.

The Bayes classifier is obtained by minimizing the expected cost. From the expression of the expected cost in (2.3), the Bayes rule chooses the decision which yields the minimum PEL:

$$\delta_B(x_K) = \arg \min_{d \in A} R(d|x_K)$$

(2.4)

This classifier can be converted into an equivalent classifier that make a decision based on decision boundaries for the posterior probability:

$$\delta_B(x_K) = \begin{cases} 
0 & p_1(x_K) \leq L \\
1 & p_1(x_K) \geq L 
\end{cases}$$

(2.5)

where

$$L = \frac{C_{10}}{C_{10} + C_{01}}$$

(2.6)

Consider 0/1 cost structure, i.e., $C_{10} = C_{01} = 1$, then the expected cost can be expressed as the marginal error rate (MER)

$$\text{MER} = \sum_{d \neq y \in Y} P(d|y)\pi_y$$

(2.7)
where

\[ P(d|y) = \int_{\{x_K: \delta = d\}} f_y(x_K) dx_K \]  \hspace{1cm} (2.8)

The Bayes classifier reduces to the maximum a posterior (MAP) classifier:

\[ \delta_B(x_K) = \arg \max_{d \in A} p_1(x_K) \]  \hspace{1cm} (2.9)

and the \( L \) becomes 0.5 under the loss function.

In case where the posterior probabilities are close to 0.5, a high possibility of being misclassified is high. In many circumstances such as a medical decision problem, a high quality of classification is emphasized since a misclassification can lead to a substantial cost. In the following section, we introduce a neutral zone classifier that can make a more cost-effective decision.

\subsection{2.2 Non-Sequential Neutral Zone classifier}

Neutral zone classifiers (\( \delta_N \)) can take one of three decisions \( d \in D \) where \( D = \{0, 1, N\} \). Let \( C_{Ny} \) be the cost incurred by assigning neutral zone, Note that the neutral zone costs are usually lower than the misclassification costs. The expected cost is formulated as

\[ EC(\delta_N) = \sum_{d \in D} \sum_{y \in Y} \int_{\{x_K: \delta = d\}} [C_{dy} f_y(x_K) \pi_y] dx_K \]  \hspace{1cm} (2.10)

which consists of mainly two parts: the expected cost incurred by a misclassification and the expected cost incurred by a neutral zone. Therefore, a neutral zone classifier is chosen by trading off between the misclassification and the neutral zone. A cost structure plays an essential role in the trade-off.
Consider $C_{10} = C_{01} = 1$ and $C_{N0} = C_{N1} \equiv \lambda < 1$, the overall expected cost can be expressed as

$$EC(\delta) = MER + \lambda NZR$$

(2.11)

where the overall neutral zone rate (NZR) is

$$NZR = \sum_{y \in \mathcal{Y}} P(N|y)\pi_y$$

(2.12)

Note that $\lambda$ can be interpreted as a penalty parameter which controls the trade-off between MER and NZR: a small value of $\lambda$ reflects high concern for misclassifications while a large value of $\lambda$ places a premium on making a decision $d \in \mathcal{A}$.

To determine the neutral zone classifier that minimizes the expected cost, we rewrite (2.10) as

$$EC(\delta) = \sum_{d \in \mathcal{D}} \int_{\mathcal{X}_K} \left\{ \sum_y C_{dy} p_y(x_K) \right\} m(x_K) dx_K$$

(2.13)

Denote the PEL incurred by the neutral zone

$$R(N|x_K) = C_{N0} p_0(x_K) + C_{N1} p_1(x_K)$$

(2.14)

From this expression of the expected cost in (2.13), it can be clearly seen that the classifier that minimizes the expected cost chooses the decision which yields the minimum PEL

$$\delta(x_K) = \arg \min_{d \in \mathcal{D}} R(d|x_K)$$

(2.15)

This classifier can be converted into an equivalent classifier that make a decision based on
decision boundaries for the posterior probability:

\[
\delta_N(x_K) = \begin{cases} 
0 & p_1(x_K) \leq L_0 \\
1 & p_1(x_K) \geq L_1 \\
N & L_0 \leq p_1(x_K) \leq L_1 
\end{cases}
\] (2.16)

If \(C_{N0}/(C_{01} - C_{N1} + C_{N0}) < C_{N1}/(C_{10} - C_{N0} + C_{N1})\), then

\[
L_0 = \frac{C_{N0}}{C_{01} - C_{N1} + C_{N0}} \quad \text{and} \quad L_1 = 1 - \frac{C_{N1}}{C_{10} - C_{N0} + C_{N1}}
\] (2.17)

If \(C_{N0}/(C_{01} - C_{N1} + C_{N0}) \geq C_{N1}/(C_{10} - C_{N0} + C_{N1})\), then the neutral zone does not exist and the neutral zone classifier reduces to the Bayes classifier in (2.5) with a single decision boundary in (2.6). Therefore, the classical Bayes classifier can be viewed as a special case of the neutral zone classifier.

[3] has derived the neutral zone classifier given the restriction \(C_{N0} = C_{N1} = C_N\).

Denoting \(\rho_{10} = C_{10}/C_N\) and \(\rho_{01} = C_{01}/C_N\), then (2.17) reduce to

\[
L_0 = \frac{1}{\rho_{01}} \quad \text{and} \quad L_1 = 1 - \frac{1}{\rho_{10}}
\] (2.18)

and the condition for the existence of neutral zone is \(1/\rho_{10} + 1/\rho_{01} \geq 1\). Similarly, [7] considered a simpler classifier with decision boundaries \(L_0 = \frac{1}{t}\) and \(L_1 = 1 - \frac{1}{t}\) where \(t\) is a value in \([0, 1]\).

We remark that both the classical Bayes and the neutral zone classifier are non-sequential approaches in the sense that a single decision is made based on the complete set of data \(x_K\).
2.3 Examples

In this section, we illustrate an example showing the non-sequential neural zone classifier. Consider a Gaussian distribution with known mean $\mu_y = (\mu_{y1}, \cdots, \mu_{yK})$ for class $y$ and common covariance $\Sigma$. From the conditions for decision 0 and 1 in (2.16)

\[
(\mu_1 - \mu_0)^T \Sigma^{-1} x_K \leq u(L_0) \\
(\mu_1 - \mu_0)^T \Sigma^{-1} x_K > u(L_1)
\]  

(2.19)

where, for any $L \in [0, 1]$,

\[
u(L) = \frac{1}{2} (\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 + \mu_0) + \log \frac{\pi_0}{\pi_1} + \log \frac{L}{1 - L}
\]  

(2.20)

Assume $\pi_0 = \pi_1 = 0.5$, $C_{01} = C_{10}^i = 10$ and $C_{N0} = C_{N1} = 1$. Consider a simple bivariate normal distribution with mean $\mu^i_0 = 0$, $\mu^i_1 = 2$ for all $1 \leq i \leq 2$ and $2 \times 2$ homogeneous covariance matrix $\Sigma = [\rho_{ij}]$ has a compound symmetry structure with $\rho_{ii} = 1$ and $\rho_{ij} = 0.1$ for all $1 \leq i \neq j \leq 2$. Given the cost structure, the lower and upper decision boundaries are $L_0 = 0.1$ and $L_1 = 0.9$. Figure 2.1 geographically illustrates the neutral zone classification with 100 samples generated from each class. The Bayes classifier has a single line between two classes and for $C_{01} = C_{10}^i = 10$ the equation of the boundary is

\[
2x_1 + 2x_2 = 3.636
\]  

(2.21)

The neutral zone classifier however has two lines separating class 0 and 1. From (2.19) and (2.20), it can be easily shown that the cut-off lines separating decisions on the space of $(x_1, x_2)$ are

\[
2x_1 + 2x_2 = 1.439
\]  

(2.22)

\[
2x_1 + 2x_2 = 5.834
\]
The area between two lines is for the neutral zone, the area above the upper line is for class 1, and the area below the lower line is for class 0. Under $C_{01} = C_{10}^i = 10$, the Bayes classifier returns $P(1|0) = P(0|1) = 0.089$ and $EC(\delta_B) = 1.887$ while the neutral zone classifier returns $MER = 0.015$, $NZR = 0.282$ and $EC(\delta_N) = 1.434$. The neutral zone classifier reduces the expected costs and the misclassification rates by classifying them as the neutral zone outcomes. If a lower value of the NZR is desired, then $\rho_{01}$ and $\rho_{10}$ should be reduced. Under $C_{01} = C_{10}^i = 5$, the neutral zone becomes narrower as shown Figure 2.1 (b) and its performances are $MER = 0.031$, $NZR = 0.171$ and $EC(\delta_N) = 1.327$. For both cost structures, the neutral zone classifier outperforms the Bayes classifier with respect to the expected cost. Therefore, the neutral zone classifier is more cost-effective approach.
Chapter 3

Sequential Neutral Zone Classifier

3.1 Framework of Sequential Classification

We now focus on the development of sequential classification procedure. In sequential classification setting, neutral zone classifiers are operated on the data sequentially to make a faster decision.

Let
\[ \delta_K = (\delta_1, \delta_2, \ldots, \delta_K) \] (3.1)
denote a finite stage sequential classification procedure where \( \delta_k \) is the classifier at stage \( k \).

Let \( \mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_K) \) be the set of the decision spaces for stages, where \( \mathcal{D}_k = \{0, 1, N\} \) for all \( k < K \) and \( \mathcal{D}_K = \{0, 1\} \). \( \mathcal{A} = \{0, 1\} \) is interpreted as the set of the terminal actions.

After a sequence of \( k \) measurements \( X_k = (X_1, \ldots, X_k) \) are observed, \( \delta_k \) assigns \( X_k \) to \( d \in \mathcal{D}_k \), stopping the classification procedure if \( d \in \mathcal{A} \), otherwise taking another sample \( X_{k+1} \). At the final stage \( K \), \( \delta_K \) only makes a terminal decision \( d \in \mathcal{D}_K \).
Let $C^k_{dy}$ denote the cost incurred by decision $\delta_k = d$ when the true class is $y$ where $C^k_{dy} = 0$ for all $d = y$. In the sequential decision setting, $C^K_{Ny}$ can be viewed as the cost of sampling for the next observation $X_{k+1}$. We note that $C^K_{Ny} = 0$ for each $y$ since no more delaying is allowed at stage $K$. We disregard the cost of taking $X_1$, since the cost does not affect the selection of decision rules. The stopping time of the decision procedure is $T = \inf \{ k : \delta_k = d \in A \}$. Given $T = k$, if a subject in group $y$ is classified as $d$, then the actual loss will be $C^k_{dy} + \sum_{i=1}^{k-1} C^i_{Ny}$, in which $C^k_{dy}$ is the decision cost and $\sum_{i=1}^{k-1} C^i_{Ny}$ is the total cost of decision delays. According to the cost structure, the overall expected cost for $\delta_K$ is defined as

$$EC(\delta_K) = \sum_{d \in A} \sum_{y \in \mathcal{Y}} \int_{\{x_1 : \delta_1 = d\}} \left[ C^1_{dy} f_y(x_1) \pi_y \right] dx_1$$

$$+ \sum_{k=2}^{K} \sum_{d \in A} \sum_{y \in \mathcal{Y}} \int_{\{x_k : \delta_k = d, \delta_{k-1} = N\}} \left[ C^k_{dy} + \sum_{i=1}^{k-1} C^i_{Ny} \right] f_y(x_k) \pi_y dx_k$$ (3.2)

Rearranging the sampling costs, we can rewrite the overall expected cost in (3.2) as the sum of the expected costs incurred at each stage:

$$EC(\delta_K) = EC(\delta_1) + \sum_{k=2}^{K} EC(\delta_k, \delta_{k-1} = N)$$ (3.3)

where

$$EC(\delta_1) = \sum_{d \in D_1} \sum_{y \in \mathcal{Y}} \int_{\{x_1 : \delta_1 = d\}} \left[ C^1_{dy} f_y(x_1) \pi_y \right] dx_1$$ (3.4)

which is the expected cost contributed at stage 1 and

$$EC(\delta_k, \delta_{k-1} = N) = \sum_{d \in D_k} \sum_{y \in \mathcal{Y}} \int_{\{x_k : \delta_k = d, \delta_{k-1} = N\}} \left[ C^k_{dy} f_y(x_k) \pi_y \right] dx_k$$ (3.5)

which is the expected cost contributed at stage $k$. See Appendix A for the proof of the equivalence between (3.2) and (3.3). In the above expression, $f_y(x_k)$ is the joint probability density function for $x_k$ given class $y \in \mathcal{Y}$. 

14
Denote
\[ P_k(d|y) = \int_{\{x_k, \delta_k = d, \delta_{k-1} = N\}} [f_y(x_k)] dx_k \] (3.6)
then the conditional error rate, misclassifying a subject from class \( y \) to \( d \neq y \in A \) is
\[ P(d|y) = \sum_{k=1}^{K} P_k(d|y) \] (3.7)
The overall average sample number (ASN) is of the form
\[ E(N) = 1 + \sum_{y \in A} \sum_{k=1}^{K} P_k(N|y) \pi_y \] (3.8)
Assuming \( C^k_{10} = C^{k}_{01} = 1 \) and \( C^k_{N0} = C^k_{N1} \equiv \lambda < 1 \) for all \( k \), the overall expected cost in (3.3) is simplified into
\[ EC(\delta_K) = MER + \lambda (E(N) - 1) \] (3.9)
It is clear from (3.9) that \( \lambda \) is a penalty parameter for the number of samples. As \( \lambda \) decreases, the sampling cost is relatively cheaper. If \( \lambda = 0 \), then the non-sequential Bayes classifier is the optimal classifier. It is obvious that the more accurate classification is available as more samples from subjects are collected. If sampling costs are all zero, then taking all samples leads to the minimum MER. On the other hand, if \( \lambda > 0 \), then the non-sequential classifier is no longer optimal classifier.

A sequential classification procedure that minimizes the overall expected cost in (3.3) is denoted by
\[ \delta^*_K = (\delta^*_1, \ldots, \delta^*_K) = \arg \min_{\delta_K} EC(\delta_K) \] (3.10)
and the rest of this chapters will focus on the determination of the optimal sequential classifier.
3.2 Posterior Expected Loss Based Classification

The optimal classification rule $\delta_K^*$ can be found by backward induction technique which begins with determining $\delta_K$, then proceeds one stage backward to determine $\delta_{K-1}$ given the selected $\delta_K$, and then continues in this fashion until $\delta_1$ is determined.

Stage $K$  
At the final stage, there are only two possible decisions $d \in \mathcal{D}_K$. One can clearly see from the overall expected cost in (3.3), $\delta_K$ contributes to only the expected cost incurred at stage $K$

$$EC(\delta_K, \delta_{K-1} = N)$$

$$= \sum_{d \in \mathcal{D}_K} \int_{\{x_K: \delta_K = d, \delta_{K-1} = N\}} \left[ \sum_y C^K_{dy} f_y(x_K) \pi_y \right] dx_K$$

$$= \sum_{d \in \mathcal{D}_K} \int_{\{x_K: \delta_K = d, \delta_{K-1} = N\}} [R_K(d|x_K) m(x_K)] dx_K$$

where

$$R_K(d|x_K) = \sum_y C^K_{dy} p_y(x_K)$$

, which is referred to the PEL incurred by decision $\delta_K = d$ based on $x_K$. To minimize (3.11) for any $\delta_{K-1} = N$, we see that $\delta_K$ should assign all $x_K$ satisfying $R(0|x_K) < R(1|x_K)$ to $d = 0$, and all $x_K$ satisfying $R(1|x_K) < R(0|x_K)$ to $d = 1$. It is summarized that the optimal classification at stage $K$ is

$$\delta_K^*(x_K) = \arg \min_{d \in \mathcal{D}_K} R_K(d|x_K)$$

(3.13)

Stage $K - 1$  
With classifier $\delta_K^*$, we next move back to stage $K - 1$ where there are three possible decisions $d_{K-1} \in \mathcal{D}_{K-1}$. Plugging in $\delta_K^*$ for $\delta_K$ in (3.11), and rewriting it in order
to focus on stage $K - 1$ gives

$$EC(\delta^*_K, \delta_{K-1} = N)$$

$$= \int_{\{x_{K-1}: \delta_{K-1} = N\}} \left[ \sum_y \left( \sum_{d \in D_K} \int_{\{x_K: \delta_K = d\}} C^K_{dy} f_y(x_K | x_{K-1}) dx_K \right) f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

$$= \int_{\{x_{K-1}: \delta_{K-1} = N\}} \left[ \sum_y \xi^{-1}_y(x_{K-1}) f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

(3.14)

where

$$\xi^{-1}_y(x_{K-1}) = \sum_{d \in D_K} \int_{\{x_K: \delta_K = d\}} C^K_{dy} f_y(x_K | x_{K-1}) dx_K$$

(3.15)

which can be interpreted as the minimum future risk (MFR) incurred after stage $K - 1$ given the class $y$ and observations $x_{K-1}$ such that $\delta_{K-1} = N$. To determine $\delta_{K-1}$, we only need to focus on minimizing the expected cost incurred at stage $K - 1$ and $K$

$$EC(\delta_{K-1}, \delta_{K-2} = N) + EC(\delta^*_K, \delta_{K-1} = N)$$

$$= \sum_{d \in D_{K-1}} \int_{\{x_{K-1}: \delta_{K-1} = d, \delta_{K-2} = N\}} \left[ \sum_y C^{-1}_{dy} f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

$$+ \int_{\{x_{K-1}: \delta_{K-1} = N\}} \left[ \sum_y \xi^{-1}_y(x_{K-1}) f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

(3.16)

Merging the integrals corresponding to $\delta_{K-1} = N$ gives

$$EC(\delta_{K-1}, \delta_{K-2} = N) + EC(\delta^*_K, \delta_{K-1} = N)$$

$$= \sum_{d \in D_{K-1}} \int_{\{x_{K-1}: \delta_{K-1} = d, \delta_{K-2} = N\}} \left[ \sum_y C^{-1}_{dy} f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

$$+ \int_{\{x_{K-1}: \delta_{K-1} = N\}} \left[ \sum_y \left( C^{-1}_{Ny} + \xi^{-1}_y(x_{K-1}) \right) f_y(x_{K-1}) \pi_y \right] dx_{K-1}$$

(3.17)

To express the above partial expected cost in terms of the PEL, we divide and multiply (3.17) by the marginal density $m(x_{K-1}) = \sum_y C^{-1}_{dy} f_y(x_{K-1}) \pi_y$. Then, the PEL of a terminal decision $d \in A$ has the simple form

$$R_{K-1}(d | x_{K-1}) = \sum_y C^{-1}_{dy} p_y(x_{K-1})$$

(3.18)
i.e., \( R_{K-1}(0|x_{K-1}) = C_{01}^{K-1} p_1(x_{K-1}) \) and \( R_{K-1}(1|x_{K-1}) = C_{10}^{K-1} p_0(x_{K-1}) \). The PEL of the neutral zone, on the other hand, has more complex form

\[
R_{K-1}(N|x_{K-1}) = \sum_y \left\{ C_{Ny}^{K-1} + \xi_y^{K-1}(x_{K-1}) \right\} p_y(x_{K-1})
\]

i.e.,

\[
R_{K-1}(N|x_{K-1}) = \left\{ C_{N0}^{K-1} + \xi_0^{K-1}(x_{K-1}) \right\} p_0(x_{K-1}) + \left\{ C_{N1}^{K-1} + \xi_1^{K-1}(x_{K-1}) \right\} p_1(x_{K-1})
\]

We see from (3.20) that the PEL of the neutral zone consists of mainly two types of costs for each \( y \): the neutral zone cost and the MFR given \( x_{K-1} \). We can now rewrite (3.17) in terms of the PELs

\[
EC(\delta_{K-1}, \delta_{K-2} = N) = \sum_{d \in D_{K-1}} \int \left\{ R_{K-1}(d|x_{K-1}) m(x_{K-1}) \right\} dx_{K-1}
\]

To minimize the expected cost in (3.21) for any \( \delta_{K-2} = N \), the optimal classifier \( \delta_{K-1} \) chooses the decision which is associated with the minimum PEL

\[
\delta_{K-1}^*(x_{K-1}) = \arg \min_{d \in D_{K-1}} R_{K-1}(d|x_{K-1})
\]

**Stage \( K - 2 \)** Plugging in \( \delta_{K-1}^* \) for \( \delta_{K-1} \) in (3.17) gives

\[
EC(\delta_{K-1}^*, \delta_{K-2} = N) = \sum_{d \in A} \int \left\{ R_{K-1}(d|x_{K-1}) m(x_{K-1}) \right\} dx_{K-1}
\]

then rewriting it in order to focus on stage \( K - 2 \) gives

\[
EC(\delta_{K-1}^*, \delta_{K-2} = N) = \sum_{y} \left[ \xi_y^{K-2}(x_{K-2}) f_y(x_{K-2}) \right] dx_{K-2}
\]
where

\[ \xi^K_{y} (x_{K-2}) = \sum_{d \in A} \int_{\{x_{K-1} : \delta^K_{y} = d \}} \left[ C^K_{d y} f_y(x_{K-1} | x_{K-2}) \right] dx_{K-1} \]

\[ + \int_{\{x_{K-1} : \delta^K_{y} = N \}} \left[ \left\{ C^K_{N y} + \xi^K_{y} (x_{K-1}) \right\} f_y(x_{K-1} | x_{K-2}) \right] dx_{K-1} \]

which is the MFR incurred after stage \( K - 2 \) given the class \( y \). To determine the decision rule of \( \delta^K_{K-2} \), we look at the expected cost incurred at stage \( K - 2 \) and beyond

\[ EC(\delta^K_{K-2}, \delta^K_{K-3} = N) + EC(\delta^K_{K-1}, \delta^K_{K-2} = N) + EC(\delta^K_{K}, \delta^K_{K-1} = N, \delta^K_{K-2} = N) \]

\[ = \sum_{d \in A} \int_{\{x_{K-2} : \delta^K_{K-2} = d, \delta^K_{K-3} = N \}} \left[ \sum_y C^K_{d y} f_y(x_{K-2} | x_{K-2}) \pi_y \right] dx_{K-2} \]

\[ + \int_{\{x_{K-2} : \delta^K_{K-2} = N \}} \left[ \sum_y \left\{ C^K_{N y} + \xi^K_{y} (x_{K-2}) \right\} f_y(x_{K-2} | x_{K-2}) \pi_y \right] dx_{K-2} \]

\[ \text{(3.26)} \]

Dividing and multiplying (3.26) by \( m(x_{K-2}) = \sum_y C^K_{d y} f_y(x_{K-2} | x_{K-2}) \pi_y \), then the PELs of each \( d \) is of the form

\[ R_{K-2}(0 | x_{K-2}) = C^K_{01} p_1(x_{K-2}) \]
\[ R_{K-2}(1 | x_{K-2}) = C^K_{10} p_0(x_{K-2}) \]
\[ R_{K-2}(N | x_{K-2}) = \left\{ C^K_{N0} + \xi^K_{0} (x_{K-2}) \right\} p_0(x_{K-2}) + \left\{ C^K_{N1} + \xi^K_{1} (x_{K-2}) \right\} p_1(x_{K-2}) \]

\[ \text{(3.27)} \]

It follows that (3.26) reduces to

\[ EC(\delta^K_{K-2}, \delta^K_{K-3} = N) + EC(\delta^K_{K-1}, \delta^K_{K-2} = N) + EC(\delta^K_{K}, \delta^K_{K-1} = N, \delta^K_{K-2} = N) \]

\[ = \sum_{d \in D_{K-2}} \int_{\{x_{K-2} : \delta^K_{K-2} = d, \delta^K_{K-3} = N \}} \left[ R_{K-2}(d | x_{K-2}) m(x_{K-2}) \right] dx_{K-2} \]

\[ \text{(3.28)} \]

To minimize the expected cost in (3.28) for any \( \delta^K_{K-3} = N \), it is clear that \( \delta^K_{K-2} \) chooses the decision associated with the minimum PEL

\[ \delta^K_{K-2}(x_{K-2}) = \arg \min_{d \in D_{K-2}} R_{K-2}(d | x_{K-2}) \]

\[ \text{(3.29)} \]
Stage $k$  Looking at an interim stage $k \in \{1, \cdots, K-1\}$ after $\delta^*_k, \cdots, \delta^*_K$ are recursively selected, the expected cost incurred after stage $k$ given $\delta^*_k+1, \cdots, \delta^*_K$ is simplified into

$$EC(\delta^*_k, \delta_k = N)$$

$$+ EC(\delta^*_{k+1}, \delta^*_k = N, \delta_k = N)$$

$$\cdots$$

$$+ EC(\delta^*_K, \delta^*_K-1 = N, \cdots, \delta^*_k+1 = N, \delta_k = N)$$

$$= \int_{\{x_k: \delta_k = N\}} \left[ \sum_y \xi^k_y(x_k) f_y(x_k) \pi_y \right] dx_k$$

where

$$\xi^k_y(x_k) = \sum_{d \in A} \int_{\{x_{k+1}: \delta^*_{k+1} = d\}} \left[ \sum_y C^k_{dy} f_y(x_{k+1}|x_k) \right] dx_{k+1}$$

$$+ \int_{\{x_{k+1}: \delta^*_{k+1} = N\}} \left[ \sum_y \left\{ C^k_{Ny} f_y(x_{k+1}|x_k) + \xi^k_{y+1} \right\} \right] dx_{k+1}$$

(3.31)

$\xi^k_y(x_k)$ is the MFR incurred after stage $k$ given the class $y$ and observations $x_k$ such that $\delta_k = N$. The expected cost incurred at stage $k$ is

$$EC(\delta_k, \delta_{k-1} = N)$$

$$= \sum_{d \in A} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} \left[ \sum_y C^k_{dy} f_y(x_k) \pi_y \right] dx_k$$

$$+ \int_{\{x_k: \delta_k = N\}} \left[ \sum_y C^k_{Ny} f_y(x_k) \pi_y \right] dx_k$$

(3.32)

$\delta_k$ contributes to the expected cost incurred at stage $k$ and beyond. Summing the expected costs at stage $k$ in (3.32) and the expected cost after stage $k$ in (3.30) gives

$$EC(\delta_k, \delta_{k-1} = N)$$

$$+ EC(\delta^*_{k+1}, \delta_k = N) + \cdots + EC(\delta^*_K, \delta^*_K-1 = N, \cdots, \delta^*_k+1 = N, \delta_k = N)$$

(3.33)
By dividing and multiplying (3.33) by \( m(x_k) \), the PEL for each \( d \) is of the form

\[
R_k(0|x_k) = C_{01}^k p_1(x_k)
\]

\[
R_k(1|x_k) = C_{10}^k p_0(x_k)
\]

\[
R_k(N|x_k) = \{ C_{N0}^k + \xi_0^k(x_k) \} p_0(x_k) + \{ C_{N1}^k + \xi_1^k(x_k) \} p_1(x_k)
\]

(3.34)

It follows that (3.33) reduces to

\[
EC(\delta, \delta_{k-1} = N) + EC(\delta_{k+1}, \delta_k = N) + \cdots + EC(\delta_{k-1}, \delta_{k-2} = N, \delta_k = N)
\]

\[
= \sum_{d \in D_k} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} [R_k(d|x_k) m(x_k)] \, dx_k
\]

(3.35)

To minimize the expected cost in (3.35) for \( \delta_{k-1} = N \), it is clear that the classifier \( \delta_k \) for each \( x_k \) chooses the decision which minimizes the PEL

\[
\delta_k^*(x_k) = \arg \min_{d \in D_k} R_k(d|x_k)
\]

(3.36)

The process for stage \( k \) is repeated recursively for selections of all \( \delta_{k-1}, \cdots, \delta_1 \). We conclude that the sequential classification procedure \( \delta_K \) that minimizes the overall expected cost is equivalent to a set of classifiers mapping the cumulative observations to the decision for which the PEL is minimum.

### 3.3 Example of PEL-Based Classification

Let us give an example that illustrates how the PEL-based classification can be done. As described earlier, a subject is sequentially classified in a forward looking approach, starting with the first observation and continuing sampling if a neutral zone is assigned.
For simplicity, we consider a classification plan with $K = 3$. After the first sample $x_1$ is observed, then we wish to classify a subject according to classification rule $\delta_1^\ast$, which requires to compute $R_1(d|x_1)$ for $d = 0, 1$ and $N$. The calculation of the PEL of two terminal decisions are easy: $R_1(0|x_1) = C_{01}^1 p_1(x_1)$ and $R_1(1|x_1) = C_{10}^1 p_0(x_1)$, whereas the PEL for the neutral zone has more the complex form:

$$R_1(N|x_1) = \{C_{N0}^1 + \xi_y^1(x_1)\} p_0(x_1) + \{C_{N1}^1 + \xi_y^1(x_1)\} p_1(x_1) \quad (3.37)$$

where

$$\xi_y^1(x_1) = \sum_{d \in A} \int_{\{x_2: \delta_2^j = d\}} \left[ C_{dy}^2 f_y(x_2|x_1) \right] dx_2 \quad (3.38)$$

$$+ \int_{\{x_2: \delta_2^N = N\}} \left[ \left\{ C_{Ny}^2 + \xi_y^2(x_2) \right\} f_y(x_2|x_1) \right] dx_2$$

and

$$\xi_y^2(x_2) = \sum_{d \in A} \int_{\{x_3: \delta_3^j = d\}} \left[ C_{dy}^3 f_y(x_3|x_2) \right] dx_3 \quad (3.39)$$

Clearly $\xi_y^1(x_1)$ needs two-stage backward induction: the calculation of $\xi_y^2(x_2)$ for all $x_2$ is followed by the calculation of $\xi_y^1(x_1)$. Computing (3.38) and (3.39) are rather difficult due to the complex integration regions. Monte Carlo integration is a feasible approach to evaluate the integrals. Let $I_d^k(\cdot)$ for $d \in D_k$ denote an indicator function defined as

$$I_d^k(\cdot) = \begin{cases} 1 & \text{if } \delta_d^k(\cdot) = d \\ 0 & \text{otherwise} \end{cases} \quad (3.40)$$

To evaluate $\xi_y^2(x_1, x_2i)$, we first generate $x_{21}, \cdots, x_{2n}$ from $f_y(x_2|x_1)$, and again generate $x_{31}, \cdots, x_{3n}$ from $f_y(x_3|x_1, x_{2i})$ for all $x_{2i}, i \in \{1, \cdots, n\}$. Next, we choose the optimal decision $d$ for each $(x_1, x_{2i}, x_{3j}), i, j \in \{1, \cdots, n\}$, according to the classifier $\delta_3^i(d|x_1, x_{2i}, x_{3j}) = \arg \min_{d \in D_3} R_3(d|x_1, x_{2i}, x_{3j})$. It follows from (3.39) that the Monte
Carlo estimate of $\xi_y^2(x_1, x_{2i})$ is

$$
\sum_{d \in A} C_{dy}^3 \frac{1}{n} \sum_{j=1}^{n} \left[ I_d^3(x_1, x_{2i}, x_{3j}; y) \right]
$$

(3.41)

Given $\xi_y^2(x_1, x_{2i})$, then we choose the optimal decision $d \in D_2$ according to the classifier $\delta_2^*(x_1, x_{2i}) = \arg \min_{d \in D_2} R_2(d|x_1, x_{2i})$. It follows from (3.38) that the Monte Carlo estimate of $\xi_y^1(x_1)$ is

$$
\sum_{d \in A} C_{dy}^2 \frac{1}{n} \sum_{i=1}^{n} \left[ I_d^2(x_1, x_{2i}; y) \right] + \frac{1}{n} \sum_{i=1}^{n} \left\{ \left( C_{N_0}^2 + \xi_y^2(x_1, x_{2i}) \right) I_N^2(x_1, x_{2i}; y) \right\}
$$

(3.42)

After $\xi_y^1(x_1)$ is computed for all $y$, then we can easily calculate $R_1(d|x_1)$ for all $d \in D_1$, then choose the decision for which the PEL is the smallest. If $x_1$ leads to the neutral zone, then we need to observe another sample $x_2$ and update the posterior probability based on the accumulated measurements $x_2$. To choose the optimal decision at stage 2, we again need to evaluate the PELs at stage 2:

$$
R_2(0|x_2) = C_{01}^2 p_1(x_2)
$$
$$
R_2(1|x_2) = C_{10}^2 p_0(x_2)
$$

(3.43)

$$
R_2(N|x_2) = \left\{ C_{N_0}^2 + \xi_y^2(x_2) \right\} p_0(x_2) + \left\{ C_{N_1}^2 + \xi_y^1(x_2) \right\} p_1(x_2)
$$

Again, $x_{31}, \cdots, x_{3n}$ are generated from $f_y(x_3|x_2)$, then it follows that the Monte Carlo estimate of $\xi_y^2(x_2)$ is

$$
\sum_{d \in A} C_{dy}^3 \frac{1}{n} \sum_{j=1}^{n} \left[ I_d^3(x_2, x_{3j}; y) \right]
$$

(3.44)

where $d$ is chosen according to $\delta_3^*(x_2, x_{3j}) = \arg \min_{d \in D_3} R_3(d|x_2, x_{3j})$.

If $x_2$ leads to the neutral zone, we observe a new sample $x_3$. We update the posterior probability given $x_3$, and compute $R_3(0|x_3) = C_{01}^3 p_1(x_3)$ and $R_3(1|x_3) = C_{10}^3 p_0(x_3)$. A final decision is made by choosing the decision for which the PEL is the smaller.
If $x_2$ leads to the neutral zone, we observe a new sample $x_3$. We update the posterior probability given $x_3$, and compute $R_3(0|x_3) = C^3_{01}p_1(x_3)$ and $R_3(1|x_3) = C^3_{10}p_0(x_3)$. A final decision is made by choosing the decision for which the PEL is the smaller.

As a simple example of the PEL-based classification, we consider a sequence of Gaussian variable with mean $\mu_{0i} = 0$ and $\mu_{1i} = 2$ for $i = 1, 2, 3$ and variance-covariance matrix $\Sigma = [\rho_{ij}]$ for both groups with a compound symmetry structure where $\rho_{ii} = 1$ and $\rho_{ij} = 0.1, 1 \leq i \neq j \leq K$. For a first sample $x_1 = 1$ at stage 1, the optimal decision is to assign the neutral zone based on the result: $R_1(0|x_1) = 5$, $R_1(1|x_1) = 5$ and $R_1(N|x_1) = 2.546$. Since $x_1$ led to the neutral zone, we continue sampling. Let another sample at stage 2 is $x_2 = 0.8$, then the cumulative samples $x_2 = (1, .8)$ led again to the neutral zone based on the results: $R_2(0|x_2) = 4.1$, $R_2(1|x_2) = 5.9$ and $R_2(N|x_2) = 2.989$. Moving to the final stage with another sample $x_3 = 0.2$, the terminal decision is that the samples belong to class 0, based on the PELs: $R_3(0|x_3) = 1.589$ and $R_3(1|x_3) = 8.841$. Using the Monte Carlo integration with random samples of $n = 1000$ at each step, the run-time for computing $R_1(N|x_1)$ were 29.07 minutes while the run-time for computing $R_2(N|x_2)$ was only 1.3 minutes. In practice, the backward induction technique gets increasingly easier as process moves toward the final stage $K$. This example indicates the difficulty of implementing the PEL-based classification. We next focus on classification problems with Gaussian distributions which enable us to deal with backward induction much easier.
Chapter 4

Sequential Classifications for
Gaussian Variable

4.1 Homogeneous Gaussian Case

4.1.1 Boundary-based Classifier

Suppose that $X_k$ in group $y \in \mathcal{A}$ has a multivariate normal distribution with known mean $\mu_{yk} = (\mu_{y1}, \cdots, \mu_{yk})$ and known covariance $\Sigma_k \equiv \Sigma_{0k} = \Sigma_{1k}$ for all $k$. From the connection between two classifiers in (2.15) and in (2.16) for the non-sequential case in section 2, we consider a possibility of substituting the PEL-based classifier with a classifier that makes a decision based on decision boundaries for the posterior probability of a class membership. The proposed decision boundary-based classifier is defined as
Figure 4.1: Example of decision boundaries for the posterior probability: Assumptions: $K = 5$, $\pi_0 = \pi_1 = 0.5$, $C_{01}^i = C_{10}^i = 10$, $C_{Ny}^i = 1$, $\mu_0^i = 0$, $\mu_1^i = 2$ and the covariance matrices $\Sigma_y = [\rho_{ij}]$ are compound symmetry with $\rho_{ij} = 0.1$ if $i \neq j$ and $\rho_{ii} = 1$ for class 0 and $\rho_{ii} = 1.8$ for class 1, for all $1 \leq i \neq j \leq K$ and $y \in \{0, 1\}$.

\[
\delta_k(x_k) = \begin{cases} 
0 & p_1(x_k) \leq L_{0k} \\
1 & p_1(x_k) \geq L_{1k} \\
N & L_{0k} < p_1(x_k) < L_{1k}
\end{cases}
\] (4.1)

where $0 \leq L_{0k} \leq L_{1k} \leq 1$ and $L_K \equiv L_{0K} = L_{1K}$. In this classification strategy, a terminal decision is made if the posterior probability of a class membership is sufficiently large, i.e., $p_1(x_k) \geq L_{1k}$ or $p_0(x_k) \geq 1 - L_{0k}$. If $p_1(x_k)$ is between $L_{0k}$ and $L_{1k}$, then sampling is continued to make a more confident decision later. Figure 4.1 provides an example of decision boundaries for the posterior of class 1. A boundary-based classification procedure is continued until $p_1(x_k)$ exceeds the above the upper curve or the lower curve.
4.1.2 Transformation of Distribution

Given the homogeneous Gaussian assumption, we specify integration regions in the overall expected cost in (3.3). For any \( L \in [0, 1], p_1(X_k) \leq L \) is if and only if

\[
(\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} X_k \leq u_k(L)
\]

(4.2)

where

\[
u_k(L) = \frac{1}{2} (\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} (\mu_{1k} + \mu_{0k}) + \log \frac{\pi_0}{\pi_1} + \log \frac{L}{1-L}
\]

(4.3)

Denote \( w_k^T = (w_{k1}, w_{k2}, \ldots, w_{kk}) = (\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} \) where \( w_{ki} \) is the weight associated with \( X_i \) at stage \( k \). Define \( S_k = w_k^T X_k = \sum_{j=1}^{k} w_{ji} X_i \), a weighted sum of measurements observed up to time \( k \). The vector of the weighted sums \( S_k = (S_1 \cdots S_k)^T \) has also the multivariate normal distribution

\[
S_k | Y = y \sim MVN \left( W_k \mu_y, W_k \Sigma_k W_k^T \right) \quad k = 1, \cdots, K
\]

(4.4)

with the probability density function \( g_y \) where \( W_k \) is the \( k \times k \) weight matrix

\[
W_k = \begin{pmatrix}
w_{11} & 0 & \cdots & 0 \\
w_{21} & w_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
w_{k1} & w_{k2} & \cdots & w_{kk}
\end{pmatrix}
\]

(4.5)

Using (4.2), the three conditions shown in (4.1) correspond, respectively, to the three sets:

\[
A^k_0 = \{s_k : s_k \leq u_k(L_{0k})\}
\]
\[
A^k_1 = \{s_k : s_k \geq u_k(L_{1k})\}
\]
\[
A^k_N = \{s_k : u_k(L_{0k}) < s_k < u_k(L_{1k})\}
\]

(4.6)
with $A^K_N$ the empty set. Under the homogeneous Gaussian assumption, $s_k$ is linearly bounded, therefore, dealing with the decision problem in the space of the weighed sum is more tractable. Let $p_y(s_k) = g_y(s_k)\pi_y/\sum_{y \in \{0,1\}} g_y(s_k)\pi_y$, $k = 1, \ldots, K$. Useful properties of distribution of $S_k$ are as follows (See, Appendix A.2 and A.3 for the proofs).

**Theorem 1** For any $x_k$ and $y \in \mathcal{Y}$,

$$p_y(s_k) = p_y(x_k)$$

**Theorem 2** For any $k = 1, \ldots, K$ and $y = 0, 1$,

$$g_y(s_k|s_{k-1}) = g_y(s_k|s_{k-1})$$

We now re-express the overall expected cost in (3.3) in terms of $s_k$

$$EC(L_0K, L_{1K}) = \sum_{d \in D_1} \sum_{y \in \mathcal{Y}} \int_{A_1^d} \left[ C_{dy}^1 g_y(s_1)\pi_y \right] ds_1$$

$$+ \sum_{k=2}^{K} \sum_{d \in D_k} \sum_{y \in \mathcal{Y}} \int_{A_1^d} \cdots \int_{A_k} \left[ C_{dy}^k g_y(s_k)\pi_y \right] ds_k \cdots ds_1$$

(4.7)

When considering an arbitrary stage $k < K$, $\delta_k$ contributes to the expected cost incurred at stage $k$ and beyond, given $\delta^*_{k+1}, \ldots, \delta^*_K$, that is

$$EC(\delta_k, \delta_{k-1} = N)$$

$$\cdots + EC(\delta_{k+1}^*, \delta_k = N) + \cdots + EC(\delta_K^*, \delta_{K-1} = N, \ldots, \delta_{k+1}^* = N, \delta_k = N)$$

$$= \sum_d \sum_y \int_{A_1^d} \cdots \int_{A_{k-1}^y} \int_{A_k^y} \left[ C_{dy}^k g_y(s_k)\pi_y \right] ds_k \cdots ds_1$$

$$+ \sum_y \int_{A_1^y} \cdots \int_{A_{k-1}^y} \int_{A_k^y} \left[ \{ C_{Ny}^k + \xi_y^k(s_k) \} g_y(s_k)\pi_y \right] ds_k \cdots ds_1$$

(4.8)

where the MFR can be recursively determined as the form

$$\xi_y^k(s_k) = \sum_d \int_{A_d^k+1} \int_{A_{k+1}^d} C_{dy}^{k+1} g_y(s_{k+1}|s_k) ds_{k+1}$$

$$+ \int_{A_{k+1}^y} \left[ C_{Ny}^{k+1} + \xi_y^{k+1}(s_{k+1}) \right] g_y(s_{k+1}|s_k) ds_{k+1}$$

(4.9)
Applying Theorem 2, the future risk in (4.9) reduces to a function of $s_k$, the MFR in (3.31) can be written in terms of $s_k$

$$
\xi^k_y(s_k) = \sum_{d \in A} \int A_{d}^{k+1} \left[ C_{dy}^{k+1} g_y(s_{k+1}|s_k) \right] ds_{k+1} + \int A_{N}^{k+1} \left[ C_{Ny}^{k+1} + \xi^k_y(s_{k+1}) \right] g_y(s_{k+1}|s_k) ds_{k+1}
$$

(4.10)

where $\xi^K_y(s_K) = 0$. Note that the above MFR can be equivalently expressed as the extensive form:

$$
\xi^k_y(s_k) = \sum_{d \in D} \int A_{d}^{k+1} C_{dy}^{k+1} g_y(s_{k+1}|s_k) ds_{k+1} + \sum_{n=k+2}^{K} \sum_{d \in D_n} \int A_{d}^{k+1} \int A_{d}^{k+2} \cdots \int A_{d}^{n} \left[ C_{dy}^{n} g_y(s_{k+1:n}|s_k) \right] ds_{k+1} \cdots ds_{n}
$$

(4.11)

which is used for the R code in Appendix C. Combining the result $\xi^k_y(s_k) = \xi^k_y(s_k)$ with Theorem 1, the PELs also can be expressed in terms of $s_k$:

$$
R_k(0|s_k) = C_{01}^k p_1(s_k)
$$

$$
R_k(1|s_k) = C_{10}^k p_0(s_k)
$$

(4.12)

$$
R_k(N|s_k) = \left[ C_{N0}^k + \xi^k_0(s_k) \right] p_0(s_k) + \left[ C_{N1}^k + \xi^k_1(s_k) \right] p_1(s_k)
$$

The partial expected cost in (3.35) become

$$
EC(\delta_k, \delta_{k-1} = N)
$$

$$
+ EC(\delta_{k+1}^*, \delta_k = N)+
$$

$$
\cdots + EC(\delta_{K}^*, \delta_{K-1} = N, \cdots, \delta_{k+1}^* = N, \delta_k = N)
$$

$$
= \int A_{N}^1 \cdots \int A_{N}^{k-1} \left[ \sum_d \int A_d^{k} R_k(d|s_k) m(s_k) ds_k \right] ds_{k-1} \cdots ds_1
$$

(4.13)

From the form of the partial expected cost in (4.13), the optimal classifier at stage $k$ for any $s_k$ is

$$
\delta^*_k(s_k) = \arg \min_{d \in D_k} R_k(d|s_k)
$$

(4.14)
Combining this result with Theorem 1, the PEL of each decision in (3.34) is expressed as a function of $s_k$:

$$R_k(0|s_k) = C_{01}^k p_1(s_k)$$

$$R_k(1|s_k) = C_{10}^k p_0(s_k)$$

$$R_k(N|s_k) = \left[ C_N^k + \xi_N^k(s_k) \right] p_0(s_k) + \left[ C_1^k + \xi_1^k(s_k) \right] p_1(s_k)$$

(4.15)

It follows that the PEL-based classifier becomes

$$\delta_k^*(s_k) = \arg\min_{d \in D_k} R_k(d|s_k)$$

(4.16)

According to Theorem 1, the boundary-based classifier in (4.1) is expressed in terms of the function of $s_k$

$$\delta_k(s_k) = \begin{cases} 
0 & p_1(s_k) \leq L_{0k} \\
1 & p_1(s_k) \geq L_{1k} \quad k = 1, \ldots, K \\
N & L_{0k} < p_1(s_k) < L_{1k}
\end{cases}$$

(4.17)

where $0 \leq L_{0k} \leq L_{1k} \leq 1$ and $L_{0K} = L_{1K}$. Under the homogeneous Gaussian distribution, the boundary-based classifier is equivalently expressed as the form making a decision based on the weighted sum:

$$\delta_k(s_k) = \begin{cases} 
0 & s_k \leq u_k(L_{0k}^*) \\
1 & s_k \geq u_k(L_{1k}^*) \\
N & u_k(L_{0k}^*) < s_k < u_k(L_{1k}^*)
\end{cases}$$

(4.18)

The boundary-based classification approach consists of two steps: a selection of decision boundaries and a classification step. An advantage of the boundary-based classification over the PEL-based classification is much less complexity in the classification step.
Given a priori-determined decision boundaries, the posterior probability is only computed for an optimal classification at each stage. Compared to the computational complexity of computing the PELs via backward induction, this is a significant advantage.

Another remaining issue of the boundary-based approach is how to determine decision boundaries in (4.17). Since we have shown the PEL classifier is optimal, it is meaningful to investigate if the equivalence between the PEL-based and the boundary-based classifiers holds in sequential classification contexts.

We initially consider a simultaneous search optimization. This approach chooses the optimal decision boundaries by jointly searching all unknowns that achieves the minimum value of the overall expected cost in (4.7) over $0 \leq L_{0k} \leq L_{1k} \leq 1$ and $0 \leq L_k \leq 1$. This search algorithm is not scalable as $K$ increases.

### 4.1.3 Relationship Between Two Classification Approaches

A more feasible approach is to leverage (4.16) to determine the decision boundaries in (4.17). Rewriting (4.16) as

\[
\delta^*_K(s_K) = \begin{cases} 
0 & R_K(0|s_K) \leq R_K(1|s_K) \\
1 & R_K(0|s_K) > R_K(1|s_K)
\end{cases}
\]  

(4.19)

for stage $K$, and

\[
\delta^*_k(s_k) = \begin{cases} 
0 & R_k(0|s_k) \leq \min \{R_k(1|s_k), R_k(N|s_k)\} \\
1 & R_k(1|s_k) \leq \min \{R_k(0|s_k), R_k(N|s_k)\} \\
N & R_k(N|s_k) < \min \{R_k(0|s_k), R_k(1|s_k)\}
\end{cases}
\]  

(4.20)
Figure 4.2: The PELs against the posterior probability. Assumptions: $K = 3$, $\pi_0 = \pi_1 = 0.5$, $C_{01}^i = C_{10}^i = 10$, $C_{Ny}^i = 1$, $\mu_0^i = 0$, $\mu_1^i = 2$ and the covariance matrices $\Sigma_y = [\rho_{ij}]$ are compound symmetry with $\rho_{ij} = 0.1$ if $i \neq j$ and $\rho_{ii} = 1$ for both classes.

for stage $k$ where $k = 1, \cdots, K - 1$. Clearly, the PEL-based classifier has the similar form to the boundary-based classifier.

For the final stage, the PEL-based and boundary-based classifiers are equivalent and the decision boundary $L_K$ is obtained when the PELs of two terminal decisions intersect.

It follows that the optimal value of $L_K$ in (4.17) is given by

$$L_K^* = \frac{C_{10}^K}{C_{10}^K + C_{01}^K}$$  (4.21)

Figure 4.2 graphically shows the equivalence of the PEL-based and the boundary-based classification approaches at the final stage ($K = 3$). Given $k = 3$, the classifier $\delta_3^*$ in (4.19) chooses $d = 0$ for $s_3$ at which $p_1(s_3) \leq L_3^*$, and chooses $d = 1$ for all values of $s_3$ at which $p_1(s_3) > L_3^*$.

Showing the equivalence between the two classifiers in (4.16) and (4.17) for an
interim stage should be more carefully done due to the complex form of the MFR. Let \( p \) denote a value of the posterior probability where \( 0 \leq p \leq 1 \), then

\[
R_k(0|p) = C_{01}^k p
\]
\[
R_k(1|p) = C_{10}^k (1 - p)
\]
\[
R_k(N|p) = \left\{ C_{N0}^k + \xi_0^k (u_k(p)) \right\} (1 - p) + \left\{ C_{N1}^k + \xi_1^k (u_k(p)) \right\} p
\]  

(4.22)

It is obvious from the above expression that \( R_k(0|p) \) and \( R_k(1|p) \) are linearly increasing and decreasing in \( p \), respectively for all \( k \). Since \( p \) reflects the probabilistic confidence in decision 1, \( \xi_1^k (u_k(p)) \) is non-increasing function of \( p \) with \( \xi_1^k (u_k(1)) = 0 \) and \( \xi_0^k (u_k(p)) \) is non-decreasing function of \( p \) with \( \xi_0^k (u_k(0)) = 0 \).

\( R_k(N|p) \) for all \( k < K \) is a concave function of \( p \) under some conditions discussed in Appendix C. The concavity of \( R_k(N|p) \) guarantees the uniqueness of lower and upper decision boundaries at each interim stage. Therefore, we can continue our discussion under the equivalence of the PEL-based and the boundary-based classifier at all stages.

Figure 4.3 provides a graphical illustration of equivalence between the two classification approaches at an interim stage \((k = 2)\) when a neutral zone exist. At the second stage, any \( s_2 \) for which \( R_2(0|s_2) \) is the lowest corresponds to the posterior probability that \( p_1(s_2) \) being less than or equal to \( L_{02} \), and any \( s_2 \) for which \( R_2(1|s_2) \) is the lowest corresponds to \( p_1(s_2) \) being greater than or equal to \( L_{12} \).

Decision boundaries are obtained when two PELs intersect. In case where a neutral zone exists, \( L_{dk}^* \) for \( d \in \{0, 1\} \) is the cutoff value of \( p \) which satisfies \( R_k(d|p) = R_k(N|p) \). If a neutral zone does not exist as shown in plot (b), then it must be that \( R_k(N|p) > \min \{ R_k(0|p), R_k(1|p) \} \) for all \( p \). Figure 4.3 is not always the case. In some case, a neutral
Figure 4.3: In case where neutral zone exists, the PEL function against the posterior probability(left) and against the weighted sum(right). Assumptions in Figure 4.2 is used.

Figure 4.4: In case where neutral zone do not exist, example of decision boundaries for the posterior probability. Assumptions in Figure 4.2 is used.
zone does not exist as shown in Figure 4.4. There is a single decision boundary in this case and is obtained when $R_k(0|p) = R_k(1|p)$. No additional sampling is required and the current stage becomes the maximum stage. The case without a neutral zone is referred to as an 'abridged design'. We next elaborate how all decision boundaries are determined recursively by considering the PEL functions.

4.1.4 Recursive Algorithm

**Stage $K - 1$** Given $L^*_K$ in (4.21), for a grid of $p \in [0, 1]$, the MFR is

$$
\begin{align*}
\xi_0^{K-1}(u_K(p)) &= \int_{u_K(L^*_K)}^{\infty} \left[ C_{10}^{K-1} g_0(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1} \\
\xi_1^{K-1}(u_K(p)) &= \int_{-\infty}^{u_K(L^*_K)} \left[ C_{01}^{K-1} g_1(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1}
\end{align*}
$$

(4.23)

where $u_{K-1}(p)$ is given by (4.3). It follows that $L^*_{d,K-1}$ for $d \in A$ is the numerical solution of the equation:

$$ R_{K-1}(d|p) = R_{K-1}(N|p) $$

(4.24)

If $L^*_{0,K-1} < L^*_{1,K-1}$, then we set $k = K - 1$ and move to the iteration step for stage $k$ below.

If $L^*_{0,K-1} \geq L^*_{1,K-1}$, then the current stage becomes the maximum stage. Thus, we replace $K$ with $k$ and run the step for stage $K - 1$ again.

**Stage $K - 2$** After $L^*_{0,K-1}$ and $L^*_{1,K-1}$ are determined, we have

$$
\begin{align*}
\xi_0^{K-2}(u_{K-1}(p)) &= \int_{u_{K-1}(L^*_{1,K-1})}^{\infty} \left[ C_{10}^{K-1} g_0(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1} \\
&+ \int_{u_{K-1}(L^*_{0,K-1})}^{u_{K-1}(L^*_{1,K-1})} \left[ \left\{ C_{N0}^{K-1} + \xi_0^{K-1}(s_{K-1}) \right\} g_0(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1} \\
\xi_1^{K-2}(u_{K-1}(p)) &= \int_{-\infty}^{u_{K-1}(L^*_{0,K-1})} \left[ C_{01}^{K-1} g_1(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1} \\
&+ \int_{u_{K-1}(L^*_{0,K-1})}^{u_{K-1}(L^*_{1,K-1})} \left[ \left\{ C_{N1}^{K-1} + \xi_1^{K-1}(s_{K-1}) \right\} g_1(s_{K-1}|u_{K-1}(p)) \right] ds_{K-1}
\end{align*}
$$

(4.25)
\( \text{If the solution of the above equations results in } L_{0,k-2}^* \geq L_{1,k-2}^*, \text{ then the current stage becomes the maximum stage. Thus, we replace } K \text{ with } k \text{ and run the step for stage } K - 1 \text{ again.} \)

**Stage** \( k \quad \text{Given } \{ L_{0,n}^k \}_{n=k+1}^K \text{ and } \{ L_{1,n}^k \}_{n=k+1}^K, \text{ for a grid of } p \in [0, 1], \text{ the MFR is} \)

\[
\begin{align*}
\xi_0^k (u_k(p)) &= \int_{L_{1,k+1}^*}^{\infty} \left[ C_{10}^{k+1} g_0(s_{k+1}|u_k(p)) \right] ds_{k+1} \\
+ &\int_{u_{k+1}(L_{0,k}^*)}^{u_{k+1}(L_{1,k}^*)} \left\{ C_{N0}^{k+1} + \xi_0^{k+1} (u_k(p)) \right\} g_0(s_{k+1}|u_k(p)) ds_{k+1} \\
\xi_1^k (u_k(p)) &= \int_{-\infty}^{L_{0,k+1}^*} \left[ C_{01}^{k+1} g_1(s_{k+1}|u_k(p)) \right] ds_{k+1} \\
+ &\int_{u_{k+1}(L_{0,k}^*)}^{u_{k+1}(L_{1,k}^*)} \left\{ C_{N1}^{k+1} + \xi_1^{k+1} (u_k(p)) \right\} g_1(s_{k+1}|u_k(p)) ds_{k+1}
\end{align*}
\]

where \( u_k(p) \) is given by (4.3). It follows that \( L_{d k}^* \) for \( d \in A \) is the numerical solution of the equation:

\( R_k(d|p) = R_k(N|p) \) (4.28)

If \( L_{0k}^* < L_{1k}^* \), then we set \( k = k - 1 \) and repeat the iteration step for stage \( k \). If \( L_{0k}^* \geq L_{1k}^* \), then the current stage becomes the maximum stage. Thus, we replace \( K \) with \( k \) and move back to the step for stage \( K - 1 \). The iteration step for stage \( k \) is continued until all boundaries are ultimately determined.

Decision boundaries at interim stages have no closed-form solutions, but it is meaningful to look at the following fixed point forms of \( L_{0k}^* \) and \( L_{1k}^* \):
\[ L_0^k = \frac{C_{N0}^k + \xi_0^k(u_k(L_{0k}^*))}{C_{N0}^k - \{C_{N1}^k + \xi_1^k(u_k(L_{0k}^*))\} + \{C_{N0}^k + \xi_0^k(u_k(L_{0k}^*))\}} \]
\[ L_{1k}^* = 1 - \frac{C_{10}^k - \{C_{N0}^k + \xi_0^k(u_k(L_{1k}^*))\} + \{C_{N1}^k + \xi_1^k(u_k(L_{1k}^*))\}}{C_{10}^k - \{C_{N0}^k + \xi_0^k(u_k(L_{0k}^*))\} + \{C_{N1}^k + \xi_1^k(u_k(L_{1k}^*))\}} \]  

(4.29)

The above expression of decision boundaries is similar to the single-stage decision boundaries in (2.17). The main difference from (2.17) is that the MFRs become zero for the non-sequential neutral zone classifier. We also indicate from (4.29) that the prior probabilities \(\pi_0\) and \(\pi_1\) have no effect on the selection of decision boundaries, but the cost structure does. If misclassification costs are relatively expensive compared to neutral zone costs, \(L_{0k}\) and \(L_{1k}^*\) become smaller and larger, respectively, resulting in the wider neutral zone at stage \(k\). Inversely, decreased neutral zone costs lead to the narrower neutral zone at stage \(k\). An important property of the boundary selection is that decision boundaries at stage \(k\) are not affected by decision boundaries at earlier stages, but depend only on the cost structure at the current stage and decision boundaries at later stages. Therefore, the selection of decision boundaries in backward manner is natural approach for the sequential classification problem.

### 4.1.5 Heterogeneous Gaussian Case

If heterogeneity in variance-covariance matrices between groups is present, then the selection of decision boundaries can not be done in the space of the weighted sum. Instead, we need Monte Carlo sample generations of \(x_k\) to evaluate the expected costs. The boundary-based classifier approach is applied in heterogeneous Gaussian case. Figure 4.5 display an example of the PEL functions under a heterogeneous Gaussian assumption. The PEL of the neutral zone is an asymmetric concave function in this example in this
Figure 4.5: In heterogeneous Gaussian case, the PELs against the posterior probability. From the assumptions in Figure 4.2, $\Sigma_1$ is changed to AR(1) structure with $\rho_{ij} = 0.1^{|i-j|}$ if $i \neq j$ and $\rho_{ii} = 1$. For plot (b), the mean profiles are changed to $\mu_0 = (0.2, 0.4, 0.6)$, $\mu_1 = (2.1, 1.9, 1.7)$ and the misclassification costs are changed to $C_{01} = C_{10} = 30$ for all $i$.

case. Decision boundaries are chosen when two posterior probabilities intercept. However, a difficulty arise in leveraging the PEL functions to seek decision boundaries.

The overall expected cost in (3.3) can be expressed as

$$EC(L_{0K}, L_{1K}) = \sum_{d \in D_1} \sum_{y \in Y} \int_{B_d^1(x_1)} \left[ C_{dy} f_y(x_1) \pi_y \right] dx_1$$

$$+ \sum_{k=2}^{K} \sum_{d \in D_k} \sum_{y \in Y} \int \cdots \int_{B_d^k(x_k)} \left[ C_{dy} f_y(x_k) \pi_y \right] dx_k$$

(4.30)

where regions of the integration are defined as

$$B_0^k(x_k) = \left\{ x_k : p_1(x_k) \leq L_{0k}, \cap_{i=1}^{k-1} (L_{0i} < p_1(x_i) < L_{1i}) \right\}$$

$$B_1^k(x_k) = \left\{ x_k : p_1(x_k) \geq L_{1k}, \cap_{i=1}^{k-1} (L_{0i} < p_1(x_i) < L_{1i}) \right\}$$

(4.31)

$$B_N^k(x_k) = \left\{ x_k : \cap_{i=1}^{k} (L_{0i} < p_1(x_i) < L_{1i}) \right\}$$

To compute the expected cost in (4.30), $K$-dimensional sample vectors $x_{1K}, \cdots, x_{nK}$ are generated from the density $f_y(x_K)$ for each $y$. The required integrals can be approximated.
by
\[
\int B_k(x) f_y(x_k) dx_k \approx \frac{1}{n} \sum_{j=1}^{n} [I^k_d(x_{jk};y)]
\]
where \(d\) is chosen by the conditions in (4.31). In a homogeneous Gaussian case, the weighted sum and the posterior probability are one-to-one mapping. In a heterogeneous case, however, there are at least one \(x_k\) corresponding to the same posterior probability. Therefore, searching the decision boundary for the posterior probability would be difficult. Instead, we suggest the simultaneous searching algorithm is applied again to find the decision boundaries that minimize the expected cost in (4.7). As earlier mentioned, this algorithm is computationally expensive for large \(K\). In the following section, we introduce some alternative boundary-based classification strategies that can substitute the global solution to the decision boundaries that minimize the expected cost.

4.2 Alternative Decision Boundaries

4.2.1 Local Sequential Neutral Zone Classifier (LSNZ)

One intuitively appealing simpler approach is to use the non-sequential neutral zone classifier at all interim stages and then the classical Bayes classifier at the final stage. This classifier was considered in [19]. Modifying the decision boundaries in (2.17) and (2.6), the lower and upper boundaries are \(L_{1k} = \frac{C^k_{N0} / (C^k_{01} - C^k_{N1} + C^k_{N0})}{C^k_{N1} / (C^k_{10} - C^k_{N0} + C^k_{N1})}\) for \(k = 1, \cdots, K-1\), respectively, and the boundary at the final stage uses (4.21). We refer to this classifier as the local sequential neutral zone classifier (LSNZ). An advantage of this classifier is that decision boundaries are easily defined on the basis of the costs with no difficulty of minimizing the overall expected cost. Of course, this
classifier is not optimal with respect to minimum expected cost. Indeed, decision boundaries at each stage are selected without consideration of the additional cost incurred by possibly continuing the procedure. It can be shown from (4.29) that this local solution ignores the MFRs as a part of delaying costs.

4.2.2 Simplified Sequential Neutral Zone Classifier (SNZ3)

Another simpler approach is to fix the lower and upper boundaries at each interim stage such that \( L_{0k} = L_0 \) and \( L_{1k} = L_1 \) for \( k = 1, \cdots K - 1 \), and to use a single boundary \( L_{0K} = L_{1K} = L \) at the final stage. This classifier is referred to as SNZ3 because there are only three unknown decision boundaries \( L_0, L_1 \) and \( L \) regardless of the maximum number of samples. The three values are numerically obtained by minimizing the overall expected cost. A compromise could be made to use (4.21) for \( L \) and search only for \( L_0 \) and \( L_1 \). This classification design can lead to fast computation and complexity reduction.

4.3 Illustrative Comparisons

The proposed classifiers are illustrated under several Gaussian examples. SNZ is compared with alternative classifiers. In addition, the simultaneous vs. recursive search algorithm for SNZ are compared. As a default setting, we assume the following homogeneous Gaussian case: \( K = 5, \pi_1 = 0.5, C_{01}^i = C_{10}^i = 10, C_{N0}^i = C_{N1}^i = 1, \mu_0^i = 0, \mu_1^i = 2 \) and \( \Sigma_0 = \Sigma_1 = \Sigma = |\rho_{ij}| \) has a compound symmetry structure with \( \rho_{ii} = 1 \) and \( \rho_{ij} = 0.1 \) for all \( 1 \leq i \neq j \leq K \). We will remark only when the assumptions change from this setting.
Figure 4.6: Decision boundaries and performance evaluations for SNZ
Figure 4.7: Classification Simulation. 200 samples were generated from the distribution under the default setting.

Plot (a) and (b) in Figure 4.6 provides the decision boundaries for SNZ in \((k, p_1(s_k))\) and \((k, s_k)\) spaces, respectively, under the default setting. In plot (a), the points on the lower curve are \(L_{0k}\) and the points on the upper curves are \(L_{1k}\) for \(k = 1, \cdots, K\). Similarly, in plot (b), the points on the lower and upper curves are \(u_k(L_{0k})\) and \(u_k(L_{1k})\), respectively. Two types of conditional error rate \(P(1|0)\) and \(P(0|1)\) are equal in this case and their cumulative values are shown in plot (c) where the total conditional error rates are equally 7.7%. The average sample number (ASN) is 1.456 and plot (d) shows that classification decisions are made for 90.4% of subjects within first two stages. To illustrate the sequential classification procedure, 200 samples are generated from the default assumption and graphically displayed in Figure 4.7. Neutral zone outcomes at stage 1 moved to the next stage. For each case in stage 2, an additional sample \(x_2\) is taken and the posterior probability is updated based on \((x_1, x_2)\). Only few cases are moved to stage 3 and 4.

For the flat mean profiles as shown in Figure 4.6 (a), the region of the neutral zone becomes gradually narrower as \(k\) increases. In case of the diverging mean profiles shown
in Figure 4.8 (b), the decision boundaries for SNZ are gradually widen before forcing a
final decision to be made at the last stage. The ASN is 2.793, which is larger than the
ASN for the flat mean profiles in above example. In contrast, for the converging mean
profiles, the neutral zone vanishes at stage 3 and the ASN is only 1.22. The curtailment
appears at stage 3 due to the fact that delaying a decision is more expensive than making an
immediate decision for any measurements at stage 3. Under a cross mean profiles, we could
have an unusual shape of decision boundaries such as plot (c.2) showing a combination of
the decreasing and increasing neutral zone regions.

Figure 4.9 compares decision boundaries for SNZ with alternative sequential clas-
sifiers and the non-sequential Bayes classifier for two cases, the flat and diverging mean pro-
files. The evaluations of performance are given in Table 4.1. Note that the non-sequential
Bayes classifier has only a single decision boundary appearing at the final stage and the
sample number is always fixed to $K$. SNZ outperforms all other classifiers with respect to
the expected cost. In comparison with LSNZ or the non-sequential Bayes, SNZ substan-
tially reduces the ASN, though the error rates do increase, where it is implicit that one is
willing to compromise on error rates in order to minimize the overall expected costs. An
interesting result is that SNZ3 is competitive to SNZ with respect to the expected cost
for all cases dealt with in this section, even though the shape of decision boundaries seem
to be very different. Both SNZ and SNZ3 lead to appreciably faster decision than other
classification rules. Therefore, if simplicity is preferred to the benefit of small improvement
in the expected cost we suggest that SNZ3 can be considered a viable alternative to SNZ.
Figure 4.8: Example of decision boundaries for the posterior probability
Figure 4.9: For homogeneous Gaussian, comparisons of decision boundaries. (a) uses the default setting and (b) uses $\mu_0 = (1.0, 0.8, 0.6, 0.4, 0.2)$ and $\mu_1 = (1.3, 1.5, 1.7, 1.9, 2.1)$.

<table>
<thead>
<tr>
<th></th>
<th>SNZ</th>
<th>SNZ3</th>
<th>LSNZ</th>
<th>Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Flat Mean Profile</td>
<td>EC</td>
<td>1.230</td>
<td>1.233</td>
<td>1.412</td>
</tr>
<tr>
<td></td>
<td>$E(N)$</td>
<td>1.456</td>
<td>1.452</td>
<td>1.950</td>
</tr>
<tr>
<td></td>
<td>$P(1</td>
<td>0)$</td>
<td>0.077</td>
<td>0.078</td>
</tr>
<tr>
<td></td>
<td>$P(0</td>
<td>1)$</td>
<td>0.077</td>
<td>0.078</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SNZ</th>
<th>SNZ3</th>
<th>LSNZ</th>
<th>Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) Diverging Mean Profile</td>
<td>EC</td>
<td>4.039</td>
<td>4.088</td>
<td>4.545</td>
</tr>
<tr>
<td></td>
<td>$E(N)$</td>
<td>2.793</td>
<td>3.152</td>
<td>4.420</td>
</tr>
<tr>
<td></td>
<td>$P(1</td>
<td>0)$</td>
<td>0.225</td>
<td>0.194</td>
</tr>
<tr>
<td></td>
<td>$P(0</td>
<td>1)$</td>
<td>0.225</td>
<td>0.194</td>
</tr>
</tbody>
</table>

Table 4.1: Comparisons of classification performance. (a) uses the default setting and (b) uses the diverging mean profiles in Figure 3.
Figure 4.10: Decision boundaries under different covariance structure. Assume \( \rho_{ii} = 1 \) for all cases, and assume \( \rho_{ij} = 0.3 \) for CS and \( \rho_{ij} = 0.3|j-i| \) for AR(1), and \( \rho_{ij} = 0 \) for IID case, \( 1 \leq i \neq j \leq K \).

Figure 4.10 shows decision boundaries of SNZ for the default setting, but changing covariance structures. Decision boundaries are narrower with higher correlations among measurements. The IID case, in which correlations between measurements are all zero, has almost horizontal boundaries. The labeled AR(1) has narrower boundaries than the IID case, and the CS covariance structure has narrowest boundaries case, all correlations between measurements are fixed to a constant, so given the assumptions in Figure 4.10, correlations for the CS is greater than or equal to correlations for AR(1). The CS covariance structure has the most narrow boundaries among these three covariance structures. The take-away message from Figure 4.10 is that increasing correlation among the repeated measures narrows the boundaries and improves the performance of the SNZ classification procedure.
Figure 4.11: Decision boundaries under the default setting but changing cost structures. For (a) and (d), $C^0_{01} = C^0_{10} = 10$, for (b), $C^0_{01} = C^0_{10} = 30$, for (c) $C^i_{01} = 30$ and $C^i_{10} = 10$. For (d), $C^i_{N0} = C^i_{N1} = 1.2^i$ and the others used $C^i_{N0} = C^i_{N1} = 1$.

The plot in Figure 4.11 shows how the decision boundaries vary with different cost structures, and the accompanying table summarizes the performance of classifiers. Note that, under different cost structures, the expected costs are not comparable, but we can see how the sample size and the error rates are affected by a selection of costs. When the misclassification costs are symmetric, the lower and upper boundaries are also symmetric about the horizontal line at $p_1(s_k) = 0.5$. In case (a), the misclassification costs are 10 times more expensive than the delaying costs at each stage. Increasing the misclassification costs to 30 in case (b) leads to the wider neutral zone meaning it requires higher confidence to make an immediate decision, and results in larger ASN and smaller $P_{10}$ and $P_{01}$ as shown in the table of Figure 4.11. An asymmetric cost structure is appropriate when two types of misclassifications are not equally important. For instance, the case where a patient suffering a disease is misclassified into the non-disease group causes more serious problem than the

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$EC$</td>
<td>1.230</td>
<td>3.096</td>
<td>1.665</td>
<td>1.356</td>
</tr>
<tr>
<td>$E(N)$</td>
<td>1.456</td>
<td>2.261</td>
<td>1.710</td>
<td>1.312</td>
</tr>
<tr>
<td>$P(1</td>
<td>0)$</td>
<td>0.077</td>
<td>0.037</td>
<td>0.109</td>
</tr>
<tr>
<td>$P(0</td>
<td>1)$</td>
<td>0.077</td>
<td>0.037</td>
<td>0.027</td>
</tr>
</tbody>
</table>
other case. If $C_{10}^k < C_{01}^k$, then the lower and upper boundaries are asymmetric and the error rate associated with higher misclassification costs is smaller than the other, as shown in the table for case (c). Sequential problems often use fixed sampling costs, but it would be possible that taking more sample gets increasingly more difficult. In case (d), the penalty for delaying a decision increases across time, and the neutral zone narrows leading to a faster decision and, in this case, an abridged design.

Comparing the computational performance of recursive and simultaneous search algorithm for SNZ, the run-times were 2.02 and 0.30 seconds, respectively, under the default setting with $K = 5$. For the case with $K = 10$, the run-time for the simultaneous search algorithm was 187.06 while the run-time for the recursive algorithm was only 1.53 seconds. Recursive algorithm facilitates very fast computations to find SNZ decision boundaries for even large $K$. We also indicated that simultaneous search algorithm often returns inaccurate
Figure 4.13: Verification of the expected performance measures. The expected error rate and stopping proportion are compared with the simulated error rate and stopping proportion.

solutions. In Figure 4.12 (b), we see decision boundaries computed by two algorithms slightly differ at advanced stages. The recursive is more accurate and stable for the case with large $K$ (greater than 6).

The accuracy of the performance measures used to evaluate the classifiers can be verified by a simulation study. From $10^5$ simulated samples drawn in the default setting, the cumulative error rate and the cumulative stopping time are compared in Figure 4.13. Since two measures almost identical at all stages, the accuracy of formulations for SNZ has been verified.

Figure 4.14 compares decision boundaries and performance results for the described classifiers under a heterogeneous Gaussian setting with $K = 3$. Since the dispersion for group 1 is wider than for group 0, SNZ leads to $P(1|0) < P(0|1)$. 20,000 samples were
Figure 4.14: For heterogeneous Gaussian, comparisons of decision boundaries and performance evaluations. From the default setting, change $\Sigma_1 = [\rho_{ij}]$ to have AR(1) structure with $\rho_{ii} = 2$ and $\rho_{ij} = 0.3^{|i-j|}$, $1 \leq i \neq j \leq K$.

generated to compute the expected cost and the run-time for SNZ were 67.49 minutes which is larger than for the homogeneous cases in the default setting. SNZ3 is often faster than SNZ while keeping the classification performance competitive to SNZ.
Chapter 5

Application

5.1 Application to Kidney Dysfunction Data

The goal is to identify whether patients develop kidney dysfunction following cardiac surgery on the basis of biomarker measurements in blood samples. We studied 282 patients, of which 36 developed kidney dysfunction \((y = 1)\) and 246 did not \((y = 0)\), and 37 missing data values are deleted from the full dataset. As a decision marker, a blood sample from each patient is repeatedly measured at time 0, 3, 6 and 12 (hours) after the surgery. Two plots on the top in Figure 5.1 show graphical profiles of patients in each group. It can be seen that there exists a substantial overlap between two groups. Mean profiles between two groups shown on the bottom in Figure 5.1 appear to differ substantially. Assuming Gaussian distributions with homogeneous covariance, the sample means of two groups are \(\hat{\mu}_0 = (-0.57, -0.40, -0.14, 0.16)\) and \(\hat{\mu}_1 = (-0.45, -0.13, 0.33, 1.05)\), and the estimated
pooled variance-covariance matrix is

$$\hat{\Sigma} = \begin{pmatrix} 0.12 & 0.16 & 0.21 & 0.27 \\ 0.16 & 0.35 & 0.42 & 0.48 \\ 0.21 & 0.42 & 0.67 & 0.79 \\ 0.27 & 0.48 & 0.79 & 1.37 \end{pmatrix}$$

To build a sequential classification model, we assume the proportional prior probabilities, $\pi_0 = 0.5$ and $\pi_1 = 0.5$ and consider the following cost structure $C_{10}^k = 30$, $C_{01}^k = 30$. 

Figure 5.1: Biomarker profiles of 200 patients in 'No dysfunction' group and 'dysfunction' group
Figure 5.2: Decision boundaries and classification performance for kidney dataset

and \( C_{N0}^k = C_{N1}^k = 1 \) where \( k = 1, 2, \ldots, 4 \) is the stage corresponding to the time that the recent measurement are observed from each subject. If there is prior knowledge on the proportion of each population, the prior probabilities can be specified. If there is no information about the prior distribution, then the proportion of each sample group is often used or the equal prior probability is can be used.

Decision boundaries in Figure 5.2 are drawn under the distributions with the plug-in estimates. The performance results shows that SNZ classifiers leads to much faster decision while keeping error rates comparable to the Bayes classifier. The performance of LSNZ was very similar to the Bayes classifier due to relatively large misclassification costs. As expected from the individual profiles, the kidney dataset results in high error rates even for the Bayes classifier that uses all data for a decision.

We next consider a modified dataset where the mean of group 1 at every stage
Figure 5.3: Decision boundaries and performance evaluations for the kidney dataset

increases by .8. Then, there were significant improvement in misclassification rates as shown in Figure 5.1. For SNZ classifier, three stages provide enough opportunity for a final decision while other classifiers allow decisions up to stage 4. SNZ3 has comparable expected cost to SNZ, and both of the classifiers lead to much faster decision. Compared to the non-sequential Bayes classifier, SNZ substantially reduces the ASN while increasing only 0.2% of error rates.

So far, we have assumed a underlying distribution is known. In practice, distributions are unknown and can be estimated from sample datasets. For the kidney data application, decision boundaries were derived under the Gaussian assumption with the plug-in estimates. The performance measures were computed from the equations (3.6) through (3.8) given in chapter 3. If the assumption is not accurate, then the expected measures would be different from the empirical performance measures. To evaluate the empirical
performance of classifiers, 10 times of the repeated 5-fold cross validations were performed.

The averaged performance measures for both of the original and modified kidney datasets are given in table and the details of the performance for the original data is displayed in Figure 5.4. In practice, the true distributions are unknown. If the assumed distributions are incorrect, the cross-validated misclassification rates would be different from the expected performance results. For our application dataset, the cross-validated misclassification rates were far away from the expected values. In the original dataset, the cross-validated $P(0|1)$ for SNZ was 0.731% for the original dataset and 0.0% for the modified dataset. Figure 5.4 provide cross-validated performance measures over time for the original dataset. This was because the kidney data does not fit Gaussian distributions. There is some evidence that not only are the variance matrices different between the groups, but the Gaussian assumption may be untenable. Several transformation methods were considered but they were not a good way to modify the distributions to be symmetrical. If a homogeneity of variance-covariance assumption is not valid, then we can only apply sequential classifier with heterogeneous variance-covariance assumption discussed in section 4.1.5. If the Gaussian assumption is not valid, then the PEL-based classifier is a feasible approach but it is

<table>
<thead>
<tr>
<th></th>
<th>SNZ</th>
<th>SNZ3</th>
<th>LSNZ</th>
<th>Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E(N)$</td>
<td>2.565</td>
<td>2.485</td>
<td>3.981</td>
<td>4</td>
</tr>
<tr>
<td>$P(1</td>
<td>0)$</td>
<td>0.240</td>
<td>0.243</td>
<td>0.201</td>
</tr>
<tr>
<td>$P(0</td>
<td>1)$</td>
<td>0.731</td>
<td>0.675</td>
<td>0.706</td>
</tr>
</tbody>
</table>

Table 5.1: Cross-validated performance evaluation for the kidney dataset.
Figure 5.4: Cross validated performance evaluation for the kidney dysfunction data
more computer intensive method.

5.2 Application to Sepsis Data

A sepsis dataset is also applied. We attempt to classify patients as non-severe \((Y = 0)\) or severe \((Y = 1)\) sepsis using the proposed sequential classification decision strategy. There are \(N_0 = 60\) patients with non-severe sepsis and \(N_1 = 509\) patients with severe sepsis. Observations from each patient are measured at time 0, 3 and 6, and there is no missing data. Figure 9 shows that two groups are substantially overlapping, indicating this is very difficult classification problem.

To apply the proposed sequential classification model in which homogeneity of variance-covariance is assumed, logarithm transformation has been used to the data. The sample means of two groups are \(\hat{\mu}_0 = (.0226, -.0254, .0909)\) and \(\hat{\mu}_1 = (1.01, 1.07, 1.05)\), and the estimated variance-covariance matrix is given by

\[
\hat{\Sigma} = \begin{pmatrix}
0.76 & 0.65 & 0.65 \\
0.65 & 0.78 & 0.65 \\
0.65 & 0.65 & 0.82
\end{pmatrix}
\]

To construct a classification procedure, we assume the equal prior probabilities, \(C_{01}^k = C_{10}^k = 20\) and \(C_N^k = 1\) for all \(k\). Figure 5.5 shows the numerical results with the different classifiers. We see again that SNZ3 has comparable expected cost to SNZ, and both of the classifiers lead to faster decisions while keeping error rates comparable to the Bayes classifier. Note that SNZ terminates by stage 2 and the neutral zone at stage 1 helps handle the toughest classification problems by moving to stage 2.
Figure 5.5: Sepsis data set: biomarker profiles of 60 patients in each group where the blue line is for non-severe sepsis and the red line is for severe sepsis. Training data patients sampled at 0, 3, and 6 hours and connected by line segments solely for viewing purposes.

<table>
<thead>
<tr>
<th></th>
<th>SNZ</th>
<th>SNZ3</th>
<th>LSNZ</th>
<th>Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$EC$</td>
<td>6.630</td>
<td>6.664</td>
<td>8.248</td>
<td>8.318</td>
</tr>
<tr>
<td>$E(N)$</td>
<td>1.162</td>
<td>1.110</td>
<td>2.931</td>
<td>3.000</td>
</tr>
<tr>
<td>$P(1</td>
<td>0)$</td>
<td>0.273</td>
<td>0.278</td>
<td>0.266</td>
</tr>
<tr>
<td>$P(0</td>
<td>1)$</td>
<td>0.273</td>
<td>0.278</td>
<td>0.266</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusions

6.1 Conclusion

We have developed a sequential neutral zone classifier as an extension of the single-decision neutral zone classifier. A PEL-based classification uses backward induction and its computational difficulty was discussed. We presented the boundary-based classifier which can facilitate very fast computations of the sequential classification procedures.

Optimal decision boundaries (SNZ) with respect to the minimum overall expected cost are found under Gaussian distributions. Their average sample number and their misclassification rates were investigated. To determined the optimal decision boundaries, we developed a recursive algorithm for homogeneous Gaussian case and simultaneous search algorithms for both heterogeneous and homogeneous Gaussian cases. Alternative decision boundaries SNZ3 and LSNZ are also proposed. Our illustrative examples for boundary-based classifiers showed that SNZ leads to lower expected cost compared to LSNZ and the non-sequential Bayes classifier, and SNZ3 is competitive with SNZ.
6.2 Future Work

We proposed a recursive algorithm only for a homogeneous Gaussian case. A recursive algorithm for a heterogeneous Gaussian case can be studied in the similar way to homogeneous Gaussian case. Further, we hope to complete the preparation of R package for evaluation of decision boundaries and performance measures of the SNZ and alternative methods. A low computational efficiency and accuracy would be issues in implementation of the recursive approach under a heterogeneous Gaussian case, since there may exist several sequences of measurements corresponding an equal posterior probability.

We have dealt with only a binary classification problems in sequential decision contexts. In practice, however, available classes are often more than two. In this case, we could build several pairwise sequential classifiers. For $M$ class levels, $M(M - 1)/2$ binary classifiers are built separately. Instead, a multi-class sequential classifier could be developed.

We can extend the neutral zone classification concept to alternative classifiers. Under a homogeneous Gaussian assumption, the Bayes classifier is equivalent to the logistic regression. We can extend the sequential neutral zone classification to logistic regression contexts. Logistic regression allows more flexible assumptions and covariates including discrete variables. Investigation of the sequential decision boundaries for logistic regression based classification is a meaningful study. Support vector machine with neural zone region could be also investigated. In this case, we may consider two types of penalty parameters in an object function: one penalty for slack variables and another penalty for a neutral zone.

Prior density functions instead of constant prior probabilities can make the classification model more general. Hinge loss functions, quadratic loss functions, or other
structures of loss functions can be applied to penalize the decisions.
Bibliography


Appendix A

Proofs

A.1 Equivalence of Two Expected Costs in Chapter 3

Let $EC_1(\delta_K)$ and $EC_2(\delta_K)$ denote the overall expected costs in (3.2) and (3.3), respectively. Our goal is to show that

$$EC_1(\delta_K) = EC_2(\delta_K)$$  \hspace{1cm} (A.1)

The expected cost in (3.4) can be partitioned into cases for terminal decisions and the neutral zone

$$\sum_{d \in A} \sum_{y \in Y} \int_{\{x_1: \delta_1 = d\}} \left[ C_{dy}^1 f_y(x_1) \pi_y \right] dx_1$$

$$\quad + \sum_{y \in Y} \int_{\{x_1: \delta_1 = N\}} \left[ C_{Ny}^1 f_y(x_1) \pi_y \right] dx_1$$  \hspace{1cm} (A.2)

and similarly, the expected cost in (3.5) can be rewritten as

$$\sum_{d \in A} \sum_{y \in Y} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} \left[ C_{dy}^k f_y(x_k) \pi_y \right] dx_k$$

$$\quad + \sum_{y \in Y} \int_{\{x_k: \delta_k = N, \delta_{k-1} = N\}} \left[ C_{Ny}^k f_y(x_k) \pi_y \right] dx_k$$  \hspace{1cm} (A.3)
Then,

\[ EC_1(\delta_K) - EC_2(\delta_K) = \]

\[ \sum_{y \in Y} \sum_{k=2}^{K} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} \left( \left\{ \sum_{i=1}^{k-1} C_{Ny}^i \right\} f_y(x_k) \pi_y \right) dx_k \]

\[ - \sum_{y \in Y} \left( \int_{\{x_1: \delta_1 = N\}} C_{Ny}^1 f_y(x_1) \pi_y \right) dx_1 + \sum_{k=2}^{K} \int_{\{x_k: \delta_k = N, \delta_{k-1} = N\}} \left( C_{Ny}^k f_y(x_k) \pi_y \right) dx_k \]

(A.4)

The first term in (A.4) can be written as

\[ C_{Ny}^1 \left[ \sum_{k=2}^{K} \sum_{d \in A} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} f_y(x_k) \pi_y dx_k \right] + C_{Ny}^2 \left[ \sum_{k=3}^{K} \sum_{d \in A} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} f_y(x_k) \pi_y dx_k \right] + \cdots + C_{Ny}^K \left[ \sum_{d \in A} \int_{\{x_k: \delta_k = d, \delta_{k-1} = N\}} f_y(x_k) \pi_y dx_k \right] \]

(A.5)

which term by term can be seen to the second term in (A.5) Hence (A.1) holds.

### A.2 Proof of Theorem 1 in Chapter 4

**Proof.** It suffices to show that the log-likelihood ratio of \( S_k \) and \( X_k \) are equal. We expand

\[ \log \left( g_1(s_k)/g_0(s_k) \right) \]

into

\[ (w_k^T \mu_{1k} - w_k^T \mu_{0k}) (w_k^T \Sigma_k w_k)^{-1} s_k - \frac{1}{2} (w_k^T \mu_{1k} - w_k^T \mu_{0k})^T (w_k^T \Sigma_k w_k)^{-1} (w_k^T \mu_{1k} + w_k^T \mu_{0k}) \]  

(A.6)

Replacing \( w_k^T \) with \( (\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} \), then \( (w_k^T \mu_{1k} - w_k^T \mu_{0k}) (w_k^T \Sigma_k w_k)^{-1} \) is written as

\[ (\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} (\mu_{1k} - \mu_{0k}) \]

which equals 1. It follows from this result that \( \log \left( g_1(s_k)/g_0(s_k) \right) \) reduces to

\[ s_k - \frac{1}{2} (\mu_{1k} - \mu_{0k})^T \Sigma_k^{-1} (\mu_{1k} + \mu_{0k}) \]  

(A.8)
which equals \( \log \left( \frac{f_1(x_k)}{f_0(x_k)} \right) \).

### A.3 Proof of Theorem 2 in Chapter 4

**Proof.** Denote \( \Phi_{y_k} = (\phi_{y1}, \cdots, \phi_{yk})^T = W_k \mu_{ck} \) and \( \Lambda_k = W_k \Sigma_k W_k^T \), then

\[
S_k | C = y \sim \text{MVN} (\Phi_{y_k}, \Lambda_k) \quad k = 1, \cdots, K
\]  

(A.9)

Partitioning the weight vector such that \( S_k = \begin{bmatrix} S_{k-1} \\ S_k \end{bmatrix} \), then the covariance matrix \( \Lambda_k \) should be partitioned into

\[
\Lambda_k = \begin{bmatrix} \Lambda_{(1:k-1)(1:k-1)} & \Lambda_{(1:k-1)k} \\ \Lambda_{(1:k-1)k} & \Lambda_{kk} \end{bmatrix}
\]  

(A.10)

where \( \Lambda_{(1:k-1)(1:k-1)} \) is \((k-1) \times (k-1)\) matrix, \( \Lambda_{kk} \) is an element in \( k \)th row and column of \( \Lambda_k \), and \( \Lambda_{(1:k-1)k} \) is the \((k-1)\) dimensional vector of the first \( k-1 \) rows and \( k \)th column.

Then the conditional density of \( S_k \) given \( s_{k-1} \) for each class \( y \) has the normal distribution with the following parameters

\[
E(S_k|s_{k-1}) = \theta_{yk} + \Lambda_{k(1:k-1)} \Lambda_{(1:k-1)(1:k-1)}^{-1} (s_{k-1} - \Phi_{y(k-1)})
\]  

(A.11)

\[
\text{Cov}(S_k|s_{k-1}) = \Lambda_{kk} - \Lambda_{k(1:k-1)} \Lambda_{(1:k-1)(1:k-1)}^{-1} \Lambda_{(1:k-1)k}
\]

Similarly, the conditional density of \( S_k \) given \( s_{k-1} \) for class \( y \) is given by

\[
E(S_k|s_{k-1}) = \theta_{yk} + \Lambda_{k(k-1)} \Lambda_{(k-1)(k-1)}^{-1} (s_{k-1} - \phi_{y(k-1)})
\]  

(A.12)

\[
\text{Cov}(S_k|s_{k-1}) = \Lambda_{kk} - \Lambda_{k(k-1)} \Lambda_{(k-1)(k-1)}^{-1} \Lambda_{(k-1)k}
\]

From Lemma 3, \( \Lambda_{k(1:k-1)} = \Lambda_{k(k-1)} \) and \( \Lambda_{(1:k-1)(1:k-1)} = \Lambda_{(k-1)(k-1)} \). Therefore, the conditional means and covariances in (A.11) and (A.12) are equal.
Lemma 3  For any $j < k \leq K$ and $1 \leq j \leq k - 1$,

\[ \Lambda_{jj} = \Lambda_{kj} \quad (A.13) \]

Proof  For clear understanding, we similarly partition the covariance matrix of $X_k$ into

\[ \Sigma_k = \begin{bmatrix} \Sigma_{(1:k-1)(1:k-1)} & \Sigma_{(1:k-1)k} \\ \Sigma_{k(1:k-1)} & \Sigma_{kk} \end{bmatrix} \quad (A.14) \]

Then we have $\Lambda_{jj} = w_j^T \Sigma_{(1:j)(1:j)} w_j$ and $\Lambda_{kj} = w_j^T \Sigma_{(1:k)(1:j)} w_j$. Replacing $w_j^T$ by $(\mu_{1j} - \mu_{0j})^T \Sigma_j^{-1}$, then $\Lambda_{jj}$ is simplified into

\[ (\mu_{1j} - \mu_{0j})^T \Sigma_j^{-1}(1:j)(1:j)(\mu_{1j} - \mu_{0j}) \quad (A.15) \]

and $\Lambda_{kj} = w_k^T \Sigma_{(1:k)(1:j)} w_j$ is rewritten as

\[ (\mu_{1k} - \mu_{0k})^T \Sigma_j^{-1}(1:k)(1:k)(\mu_{1k} - \mu_{0k}) \Sigma_j^{-1}(1:j)(1:j)(\mu_{1j} - \mu_{0j}) \quad (A.16) \]

where it can be shown that

\[ \Sigma_j^{-1}(1:k)(1:k) \Sigma_j^{-1}(1:j)(1:j) = \begin{bmatrix} I_j \\ 0_{(k-j)\times j} \end{bmatrix} \quad (A.17) \]

where $I_j$ is the $j \times j$ identity matrix and $0_{(k-j)\times j}$ is the $(k-j) \times j$ matrix with all elements being zero. It follows that the equations in (A.15) and (A.16) are equal, Therefore, $\Lambda_{jj} = \Lambda_{kj}$. 

67
A.4 Concavity of PEL of Neutral Zone

To prove that \( R_k(N|p) \) is concave, we need to show that \( \partial^2 R_k(N|p)/\partial p^2 \leq 0 \) for all \( L \)

\[
R_k(N|p) = \left\{ C_{N0}^k + \xi_0^k(u_k(p)) \right\} (1 - p) + \left\{ C_{N1}^k + \xi_1^k(u_k(p)) \right\} p \tag{A.18}
\]

Differentiating \( R_k(N|p) \) with respect to \( p \) gives

\[
\frac{\partial R_k(N|p)}{\partial p} = -C_{N0}^k - \xi_0^k(u_k(p)) + \frac{\partial \xi_0^k(u_k(p))}{\partial p} \frac{\partial u_k(p)}{\partial p} (1 - p) + C_{N1}^k + \xi_1^k(u_k(p)) + \frac{\partial \xi_1^k(u_k(p))}{\partial p} \frac{\partial u_k(p)}{\partial p} p \tag{A.19}
\]

Since \( \frac{\partial u_k(p)}{\partial p} = \frac{1}{p(1 - p)} \), it follows that

\[
C_{N1}^k - C_{N0}^k + \xi_1^k(u_k(p)) - \xi_0^k(u_k(p)) + \frac{\partial \xi_0^k(u_k(p))}{\partial p} \frac{1}{p} + \frac{\partial \xi_1^k(u_k(p))}{\partial p} \frac{1}{1 - p} \tag{A.20}
\]

The second derivative \( \partial^2 R_k(N|L)/\partial L^2 \) is

\[
\left\{ -\frac{\partial \xi_0^k(u_k(p))}{\partial p} + \frac{\partial^2 \xi_0^k(u_k(p))}{\partial p^2} \right\} \frac{1}{p^2(1 - p)} + \left\{ \frac{\partial \xi_1^k(u_k(p))}{\partial p} + \frac{\partial^2 \xi_1^k(u_k(p))}{\partial p^2} \right\} \frac{1}{p(1 - p)^2} \tag{A.21}
\]

Since \( \xi_0^k(u_k(p)) \) is a non-decreasing and \( \xi_1^k(u_k(p)) \) is a non-increasing function of \( p \) as shown in Figure A.1, we have

\[
\frac{\partial \xi_0^k(u_k(p))}{\partial p} \geq 0 \quad \text{and} \quad \frac{\partial \xi_1^k(u_k(p))}{\partial p} \leq 0 \tag{A.22}
\]

In addition, consider three possible shapes of \( \xi_0^k(u_k(p)) \) and \( \xi_1^k(u_k(p)) \). If \( \xi_0^k(u_k(p)) \) and \( \xi_1^k(u_k(p)) \) are assumed to be linearly increasing in \( p \), then

\[
\frac{\partial^2 \xi_0^k(u_k(p))}{\partial p^2} = 0 \quad \text{and} \quad \frac{\partial^2 \xi_1^k(u_k(p))}{\partial p^2} = 0 \tag{A.23}
\]
It follows that $\frac{\partial^2 R_k(N|p)}{\partial p^2} < 0$. Thus, $R_k(N|p)$ is a concave function of $p$. If $\xi^k_0(u_k(p))$ and $\xi^k_1(u_k(p))$ are assumed to be concave for all $p$, then

$$\frac{\partial^2 \xi^k_0(u_k(p))}{\partial p^2} < 0 \text{ and } \frac{\partial^2 \xi^k_1(u_k(p))}{\partial p^2} < 0 \quad (A.24)$$

It follows that $\frac{\partial^2 R_k(N|p)}{\partial p^2} < 0$. Thus, $R_k(N|p)$ is a concave function of $p$. If $\xi^k_0(u_k(p))$ and $\xi^k_1(u_k(p))$ are assumed to be convex for all $p$, then

$$\frac{\partial^2 \xi^k_0(u_k(p))}{\partial p^2} > 0 \text{ and } \frac{\partial^2 \xi^k_1(u_k(p))}{\partial p^2} > 0 \quad (A.25)$$

Assuming that

$$\left|\frac{\partial^2 \xi^k_d(u_k(p))}{\partial p^2}\right| \leq \left|\frac{\partial \xi^k_d(u_k(p))}{\partial p}\right| \quad (A.26)$$

then we have $\frac{\partial^2 R_k(N|p)}{\partial p^2} < 0$. Therefore, $R_k(N|p)$ is a concave function of $p$. 

Figure A.1: Minimum future risks against the posterior probability
Appendix B

Backward Induction Examples

Optimal decision boundaries can be also found by taking a derivative of the overall expected cost with respect to decision boundaries. Using the expected cost in (4.7), we illustrate how backward induction method is used to determine decision boundaries. For easy understand, we first look at a simplest case where the maximum stage is 2, followed by 4-stage classification case.

B.1 Two-Stage Case

From (4.7), the overall expected cost is defined as

\[ EC(\delta_2) = EC(\delta_1) + EC(\delta_2, \delta_1 = N) \]  

(B.1)
where

\[ EC(\delta_1) = \int_{u_1(L_{11})}^{\infty} C_{10}^1 g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_{11})} C_{01}^1 g_1(s_1) \pi_1 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} [C_{N1}^1 g_0(s_1) + C_{N1}^1 g_1(s_1)] ds_1 \]  \hfill (B.2)

\[ EC(\delta_2, \delta_1 = N) = \int_{u_1(L_{01})}^{u_1(L_{11})} \int_{u_2(L_2)}^{\infty} C_{10}^2 g_0(s_2) \pi_0 ds_2 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \int_{-\infty}^{u_2(L_2)} C_{01}^2 g_1(s_2) \pi_1 ds_2 \]

For any fixed \( L_{01} \) and \( L_{11} \), we only need to focus on minimizing the part involving \( L_2 \) in \( (B.1) \), that is

\[ Q(s_1, L_2) = \int_{u_2(L_2)}^{\infty} C_{10}^2 g_0(s_2) \pi_0 ds_2 + \int_{-\infty}^{u_2(L_2)} C_{01}^2 g_1(s_2) \pi_1 ds_2 \]  \hfill (B.3)

Note that \( EC(\delta_2, \delta_1 = N) = \int_{u_1(L_{01})}^{u_1(L_{11})} Q(s_1, L_2) ds_2 \). To find the minimum cost, differentiating \( (B.2) \) with respect to \( u_2(L_2) \) and setting the derivative to zero, we have

\[ C_{01}^2 g_1(s_1, u_2(L_2)) \pi_1 - C_{10}^2 g_0(s_1, u_2(L_2)) \pi_0 = 0 \]  \hfill (B.4)

\( (B.4) \) can be written as

\[ \frac{g_1(s_1, u_2(L_2))}{g_0(s_1, u_2(L_2))} = \frac{C_{10}^2 \pi_0}{C_{01}^2 \pi_1} \]  \hfill (B.5)

By Theorem 1, \( (B.5) \) reduces to

\[ \frac{g_1(u_2(L_2))}{g_0(u_2(L_2))} = \frac{C_{10}^2 \pi_0}{C_{01}^2 \pi_1} \]  \hfill (B.6)

Then, the optimal value of \( L_2 \) is the solution of the equation \( I_2(L_2) = 0 \) where

\[ I_2(L_2) = C_{01}^2 g_1(u_2(L_2)) \pi_1 - C_{10}^2 g_0(u_2(L_2)) \pi_0 \]  \hfill (B.7)

Denoting \( L_2^* \) is the solution of \( (B.7) \), the solution has the analytical form

\[ L_2^* = \frac{C_{10}^2}{C_{10}^2 + C_{01}^2} \]  \hfill (B.8)
Note that $L^*_2$ does not depend on the selection of $L_{01}$ and $L_{11}$. Therefore $L^*_2$ is the globally optimal boundary.

Next, given $L^*_2$, $L_{01}$ and $L_{11}$ are determined by minimizing the overall expected cost ($B.1$). To facilitate that we rewrite expected cost at stage 2 in ($B.1$) as

$$EC(\delta_2, \delta_1 = N) =$$

$$\int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{10}^2 g_0(s_2|s_1) ds_2 \right\} g_0(s_1) \pi_0 ds_1$$

$$+ \int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{01}^2 g_1(s_2|s_1) ds_2 \right\} g_1(s_1) \pi_1 ds_1$$

(B.9)

Denote $\xi_0^1(s_1) = \int_{u_2(L^*_2)}^{\infty} C_{10}^2 g_0(s_2|s_1) ds_2$ and $\xi_1^1(s_1) = \int_{-\infty}^{u_2(L^*_2)} C_{01}^2 g_1(s_2|s_1) ds_2$, which are the MFR given class 0 and 1, respectively. It follows that ($B.1$) can be expressed as

$$EC(\delta_2) = \int_{u_1(L_{11})}^{\infty} C_{10}^2 g_0(s_1) \pi_0 ds_1$$

$$+ \int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{10}^1 + \xi_0^1(s_1) \right\} g_0(s_1) \pi_0 ds_1$$

$$+ \int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{01}^1 + \xi_1^1(s_1) \right\} g_1(s_1) \pi_1 ds_1$$

(B.10)

Setting the derivative of ($B.10$) w.r.t. $u_1(L_{01})$ to zero gives $I_0(L_{01}; L_2) = 0$ where

$$I_0(L_{01}; L_2) = \left\{ C_{10}^1 - C_{11}^1 - \xi_0^1(u_1(L_{01})) \right\} g_1(s_1, u_1(L_{01})) \pi_1$$

$$- \left\{ C_{01}^1 + \xi_0^1(u_1(L_{01})) \right\} g_0(u_1(L_{01})) \pi_0$$

(B.11)

Similarly, setting the derivative of ($B.10$) w.r.t. $u_1(L_{11})$ to zero gives $I_1(L_{11}; L_2) = 0$ where

$$I_1(L_{11}; L_2) = \left\{ C_{10}^1 - C_{11}^1 - \xi_0^1(u_1(L_{11})) \right\} g_0(u_1(L_{11})) \pi_0$$

$$- \left\{ C_{01}^1 + \xi_1^1(u_1(L_{11})) \right\} g_1(u_1(L_{11})) \pi_1$$

(B.12)

The solutions of ($B.11$) and ($B.12$) are denoted by $L^*_0$ and $L^*_1$ respectively. These solutions are obtained numerically, but further re-expressed as follows. Let $L^*_0$ and $L^*_1$ denote solutions for the above equations. If $L^*_0 > L^*_1$, there is no possibly that an observation is assigned to $N$, so the second stage does not exist. In this case, the current stage becomes
a single-stage classification for which the expected cost is defined by

$$EC(\delta_1) = \int_{u_1(L_1)}^{\infty} C_{10}^1 g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_1)} C_{01}^1 g_1(s_1) \pi_1 ds_1$$  \hspace{1cm} (B.13)$$

Taking a derivative of the expected cost w.r.t. $u_1(L_1)$, then we have

$$I(L_1) = C_{01}^1 g_1(u_1(L_1)) \pi_1 - C_{10}^1 g_0(u_1(L_1)) \pi_0$$  \hspace{1cm} (B.14)$$

The solution is of the form:

$$L_1^* = \frac{C_{10}^1}{C_{10}^1 + C_{01}^1}$$  \hspace{1cm} (B.15)$$

### B.2 Four-Stage Case

Assume the maximum stage is 4, then the overall expected cost is given by

$$EC(\delta_4) = EC(\delta_1) + \sum_{k=2}^{4} EC(\delta_k, \delta_{k-1} = N)$$  \hspace{1cm} (B.16)$$

Using backward induction approach, the solutions of decision boundaries are found by looking at the final stage 4. To formulate $L_4$ that minimizes (B.16) for fixed other boundaries $L_{01}, L_{11}, L_{02}, L_{12}, L_{03}$, and $L_{13}$, we only need to focus on minimizing the terms involving $L_4$, denoted by $Q(s_3, L_4)$:

$$Q(s_3, L_4) = \int_{u_4(L_4)}^{\infty} C_{10}^4 g_0(s_4) \pi_0 ds_4 + \int_{-\infty}^{u_4(L_4)} C_{01}^4 g_1(s_4) \pi_1 ds_4$$  \hspace{1cm} (B.17)$$

Setting the derivative of $Q(s_3, L_4)$ w.r.t. $u_4(L_4)$ to zero, we have

$$C_{01}^4 g_1(s_3, u_4(L_4)) \pi_1 - C_{10}^4 g_0(s_3, u_4(L_4)) \pi_0 = 0$$  \hspace{1cm} (B.18)$$

Applying Theorem 1, (B.18) reduces to $I(L_4) = 0$ where

$$I(L_4) = C_{01}^4 g_1(u_4(L_4)) \pi_1 - C_{10}^4 g_0(u_4(L_4)) \pi_0$$  \hspace{1cm} (B.19)$$
It can be shown that the solution of (B.19) is \( L^*_4 = C_{10}^4 / (C_{10}^4 + C_{01}^4) \). Since \( L^*_4 \) does not depend on other decision boundaries, \( L^*_4 \) is globally optimal value of \( L_4 \).

Given \( L^*_4 \), we next focus on minimizing the part involving \( L_{03} \) and \( L_{13} \) in (B.16) for fixed \( L_{01}, L_{11}, L_{02} \) and \( L_{12} \), that is

\[
Q(s_2, L_{03}, L_{13}; L_4) = \\
\int_{u_{3}(L_{13})}^{\infty} C_{10}^3 g_0(s_3) \pi_0 ds_3 + \int_{-\infty}^{u_{4}(L_{13})} C_{01}^3 g_1(s_3) \pi_1 ds_3 \\
+ \int_{u_{4}(L_{13})}^{u_{3}(L_{03})} \left\{ C_{N0}^3 g_{0}(s_3) \pi_0 + C_{N1}^3 g_{1}(s_3) \pi_1 \right\} ds_3 \\
+ \int_{u_{3}(L_{03})}^{u_{3}(L_{13})} Q(s_3, L_4) ds_3
\]  

(B.20)

Rewriting the above equation as

\[
Q(s_2, L_{03}, L_{13}; L_4) = \\
\int_{u_{3}(L_{13})}^{\infty} C_{10}^3 g_0(s_3) \pi_0 ds_3 + \int_{-\infty}^{u_{4}(L_{13})} C_{01}^3 g_1(s_3) \pi_1 ds_3 \\
+ \int_{u_{4}(L_{13})}^{u_{3}(L_{13})} \left\{ \int_{u_{4}(L_4)}^{\infty} C_{10}^4 g_{0}(s_4|s_3) ds_4 \right\} g_{0}(s_3) \pi_0 ds_3 \\
+ \int_{u_{3}(L_{13})}^{u_{3}(L_{13})} \left\{ \int_{-\infty}^{u_{4}(L_4)} C_{01}^4 g_{1}(s_4|s_3) ds_4 \right\} g_{1}(s_3) \pi_1 ds_3
\]  

(B.21)

To express it more clearly as a function of \( L_{03} \) and \( L_{13} \), let \( \zeta_3^3(s_3) = \int_{u_{4}(L_4)}^{\infty} C_{10}^4 g_{0}(s_4|s_3) ds_4 \) and \( \zeta_1^3(s_3) = \int_{-\infty}^{u_{4}(L_4)} C_{01}^4 g_{1}(s_4|s_3) ds_4 \), then we have

\[
Q(s_2, L_{03}, L_{13}; L_4) = \\
\int_{u_{3}(L_{13})}^{\infty} C_{10}^3 g_0(s_3) \pi_0 ds_3 + \int_{-\infty}^{u_{4}(L_{13})} C_{01}^3 g_1(s_3) \pi_1 ds_3 \\
+ \int_{u_{4}(L_{13})}^{u_{3}(L_{13})} \left\{ C_{N0}^3 g_{0}(s_3) + \zeta_3^3(s_3) \right\} g_{0}(s_3) \pi_0 ds_3 \\
+ \int_{u_{3}(L_{13})}^{u_{3}(L_{13})} \left\{ C_{N1}^3 g_{1}(s_3) + \zeta_1^3(s_3) \right\} g_{1}(s_3) \pi_1 ds_3
\]  

(B.22)

Taking derivatives of \( Q(s_2, L_{03}, L_{13}; L_4) \) w.r.t. \( u_3(L_{03}) \) and setting the derivative to zero,
Similarly, we can formulate the equation of 
\[ L \] replaced by 
\[ L \] and 
\[ L \] is abridged and the current stage becomes the maximum. In this case, 
\[ L \]

Let 
\[ L \] then we have 
\[ g \]
By Theorem 2 showing 
\[ g_d(s_4,s_3) = g_0(s_4,s_3) \] , we get 
\[ \xi^3_d(s_3) = \xi^3_d(s_3), \ d \in A. \] It follows that 
\[ \{ C^3_{01} - C^3_{N1} - \xi^3_1 (u_3(L_{03})) \} g_{s_3} (s_2, u_3(L_{03})) \pi_1 \]
\[ - \{ C^3_{N0} + \xi^3_0 (u_3(L_{03})) \} g_0 (s_2, u_3(L_{03})) \pi_0 = 0 \] 
(B.23)

Applying Theorem 1, (B.24) reduces to 
\[ I_0(L_{03};L_4) = 0 \] where 
\[ I_0(L_{03};L_4) = \{ C^3_{01} - C^3_{N1} - \xi^3_1 (u_3(L_{03})) \} g_1 (u_3(L_{03})) \pi_1 \]
\[ - \{ C^3_{N0} + \xi^3_0 (u_3(L_{03})) \} g_0 (u_3(L_{03})) \pi_0 \]
(B.25)

Similarly, we can formulate the equation of 
\[ L_{13} \] as 
\[ I_1(L_{13};L_4) = 0 \] where 
\[ I_1(L_{13};L_4) = \{ C^3_{10} - C^3_{N0} - \xi^3_0 (u_3(L_{13})) \} g_1 (u_3(L_{13})) \pi_0 \]
\[ - \{ C^3_{N1} + \xi^3_1 (u_3(L_{13})) \} g_1 (u_3(L_{13})) \pi_1 \]
(B.26)

Let 
\[ L^*_{03} \] and 
\[ L^*_{13} \] denote the solutions of the two above equations. If 
\[ L^*_{03} > L^*_{13} \] , then, SNZ is abridged and the current stage becomes the maximum. In this case, 
\[ L^*_{03} \] and 
\[ L^*_{13} \] are replaced by 
\[ L^*_0 = C^3_{10} / (C^3_{10} + C^3_{01}) \].

Given 
\[ L_{3:4}^* = \{ L^*_0, L^*_{13}, L^*_1 \} \] , we next focus on minimizing the part involving 
\[ L_{02} \] and 
\[ L_{12} \] in (B.16), that is for any fixed 
\[ L_{01} \] and 
\[ L_{11} \]:

\[ Q(s_1, L_{02}, L_{12}; L_{3:4}^*) \]
\[ = \int_{u_2(L_{12})}^{\infty} C^2_{10} g_0(s_2) \pi_0 ds_2 + \int_{-\infty}^{u_2(L_{02})} C^2_{01} g_1(s_2) \pi_1 ds_2 \]
\[ + \int_{u_2(L_{02})}^{u_2(L_{12})} \{ C^2_{N0} g_0(s_2) \pi_0 + C^2_{N1} g_1(s_2) \pi_1 \} ds_2 \]
\[ + \int_{u_2(L_{02})}^{u_2(L_{12})} Q(s_2, L_{3:4}^*) ds_2 \]
(B.27)
It can be more clearly expressed as

\[
Q(s_1, L_{02}, L_{12}; L^*_{3;4}) = \int_{u_2(L_{12})}^{\infty} C^2_{10} g_0(s_2) \pi_0 ds_2 + \int_{-\infty}^{u_2(L_{02})} C^2_{01} g_1(s_2) \pi_1 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ C^2_{N0} g_0(s_2) \pi_0 + C^2_{N1} g_1(s_2) \pi_1 \right\} ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ \int_{u_3(L_{13})}^{\infty} C^3_{10} g_0(s_3|s_2) ds_3 \right\} g_0(s_2) \pi_0 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ \int_{-\infty}^{u_3(L_{03})} C^3_{01} g_1(s_3|s_2) ds_3 \right\} g_1(s_2) \pi_1 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ \int_{u_3(L_{13})}^{\infty} (C^3_{N0} + \xi_0^3(s_3)) g_0(s_3|s_2) ds_3 \right\} g_0(s_2) \pi_0 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ \int_{u_3(L_{03})}^{\infty} (C^3_{N1} + \xi_1^3(s_3)) g_1(s_3|s_2) ds_3 \right\} g_1(s_2) \pi_1 ds_2 
\]

(B.28)

Combining the integrals over \((r(L_{02}), u_2(L_{12}))\), we have

\[
Q(s_1, L_{02}, L_{12}; L^*_{3;4}) = \\
\int_{u_2(L_{12})}^{\infty} C^2_{10} g_0(s_2) \pi_0 ds_2 + \int_{-\infty}^{u_2(L_{02})} C^2_{01} g_1(s_2) \pi_1 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ C^2_{N0} + \int_{u_3(L_{13})}^{\infty} C^3_{10} g_0(s_3|s_2) ds_3 + \int_{u_3(L_{03})}^{u_3(L_{13})} (C^3_{N0} + \xi_0^3(s_3)) g_0(s_3|s_2) ds_3 \right\} g_0(s_2) \pi_0 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ C^2_{N1} + \int_{-\infty}^{u_3(L_{03})} C^3_{01} g_1(s_3|s_2) ds_3 + \int_{u_3(L_{03})}^{u_3(L_{13})} (C^3_{N1} + \xi_1^3(s_3)) g_1(s_3|s_2) ds_3 \right\} g_1(s_2) \pi_1 ds_2 
\]

(B.29)

Denote

\[
\xi_0^3(s_2) = \int_{u_3(L_{13})}^{\infty} C^3_{10} g_0(s_3|s_2) ds_3 + \int_{u_3(L_{03})}^{u_3(L_{13})} (C^3_{N0} + \xi_0^3(s_3)) g_0(s_3|s_2) ds_3 \\
\xi_1^3(s_2) = \int_{-\infty}^{u_3(L_{03})} (C^3_{N1} + \xi_1^3(s_3)) g_1(s_3|s_2) ds_3 
\]

(B.30)

then, \(Q(s_1, L_{02}, L_{12}; L^*_{3;4})\) can be simplified into

\[
Q(s_1, L_{02}, L_{12}; L^*_{3;4}) = \\
\int_{u_2(L_{12})}^{\infty} C^2_{10} g_0(s_2) \pi_0 ds_2 + \int_{-\infty}^{u_2(L_{02})} C^2_{01} g_1(s_2) \pi_1 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ C^2_{N0} + \xi_0^3(s_2) \right\} g_0(s_2) \pi_0 ds_2 \\
+ \int_{u_2(L_{02})}^{u_2(L_{12})} \left\{ C^2_{N1} + \xi_1^3(s_2) \right\} g_1(s_2) \pi_1 ds_2 
\]

(B.31)
By theorem 2, we get $\xi^2_y(s_2) = \xi^2_y(s_2)$ with

\[\xi^2_0(s_2) = \int_{u_3(L_{13})}^{\infty} C^3_{10} g_0(s_3|s_2) ds_3 + \int_{u_3(L_{03})}^{u_3(L_{13})} (C^3_{N0} + \xi^3_0(s_3)) g_0(s_3|s_2) ds_3 \]

\[\xi^2_1(s_2) = \int_{-\infty}^{u_3(L_{03})} C^3_{01} g_1(s_3|s_2) ds_3 + \int_{u_3(L_{03})}^{u_3(L_{13})} (C^3_{N1} + \xi^3_1(s_3)) g_1(s_3|s_2) ds_3 \quad (B.32)\]

Taking derivatives w.r.t. $u_2(L_{02})$ and $u_2(L_{12})$ and setting the derivatives to zero, then we have

\[\{ C^2_{01} - C^2_{N1} - \xi^2_1(u_2(L_{02})) \} g_1(s_1, u_2(L_{02})) \pi_1 \]

\[ - \{ C^2_{N0} + \xi^2_0(u_2(L_{02})) \} g_0(s_1, u_2(L_{02})) \pi_0 = 0 \quad (B.33)\]

\[\{ C^2_{10} - C^2_{N0} - \xi^2_0(u_2(L_{12})) \} g_0(s_1, u_2(L_{12})) \pi_0 \]

\[ - \{ C^2_{N1} + \xi^2_1(u_2(L_{12})) \} g_1(s_1, u_2(L_{12})) \pi_1 = 0 \]

Applying Theorem 1, then (B.33) is simplified as $I_0(L_{02}; L^*_3,4) = 0$ and $I_1(L_{12}; L^*_3,4) = 0$

where

\[I_0(L_{02}; L^*_3,4) = \{ C^2_{01} - C^2_{N1} - \xi^2_1(u_2(L_{02})) \} g_1(u_2(L_{02})) \pi_1 \]

\[ - \{ C^2_{N0} + \xi^2_0(u_2(L_{02})) \} g_0(u_2(L_{02})) \pi_0 \quad (B.34)\]

\[I_1(L_{12}; L^*_3,4) = \{ C^2_{10} - C^2_{N0} - \xi^2_0(u_2(L_{12})) \} g_0(u_2(L_{12})) \pi_0 \]

\[ - \{ C^2_{N1} + \xi^2_1(u_2(L_{12})) \} g_1(u_2(L_{12})) \pi_1 \]

The solutions of the above equations are denoted by $L^*_0$ and $L^*_1$, respectively. If $L^*_0 > L^*_1$, then SNZ is abridged and the current stage becomes the maximum. Therefore, $L^*_0$ and $L^*_1$ are replaced by

\[L^*_0 = \frac{C^2_{10}}{C^2_{10} + C^2_{01}} \quad (B.35)\]

Now given $L_{2,4} = \{L^*_0, L^*_1, L^*_3, L^*_1, L^*_2, L^*_3, L^*_1\}$, we focus on minimizing the overall expected cost in (B.16). To facilitate that, we rewrite (B.16) to express it more clearly as a function of
$L_{01}$ and $L_{11}$:

\[ Q(L_{01}, L_{11}; \mathbf{L}_{2:4}) \]

\[ = \int_{u_1(L_{11})}^{\infty} C_{10}^1 g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_{01})} C_{01}^1 g_1(s_1) \pi_1 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ C_{N0}^1 g_0(s_1) \pi_0 + C_{N1}^1 g_1(s_1) \pi_1 \} \, ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} Q(s_1, \mathbf{L}_{2:4}^*) ds_1 \] 

It can be expressed as

\[ Q(L_{01}, L_{11}; \mathbf{L}_{2:4}^*) \]

\[ = \int_{u_1(L_{11})}^{\infty} C_{10}^1 g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_{01})} C_{01}^1 g_1(s_1) \pi_1 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ C_{N0}^1 g_0(s_1) \pi_0 + C_{N1}^1 g_1(s_1) \pi_1 \} \, ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ \int_{u_2(L_{12})}^{\infty} C_{10}^2 g_0(s_2|s_1) ds_2 \} \, g_0(s_1) \pi_0 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ \int_{-\infty}^{u_2(L_{02})} C_{01}^2 g_1(s_2|s_1) ds_2 \} \, g_1(s_1) \pi_1 ds_1 \]

Combining the integrals over $(u_1(L_{01}), u_1(L_{11}))$ at the first stage, then we get

\[ Q(L_{01}, L_{11}; \mathbf{L}_{2:4}^*) \]

\[ = \int_{u_1(L_{11})}^{\infty} C_{10}^1 g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_{01})} C_{01}^1 g_1(s_1) \pi_1 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ C_{N0}^1 + \int_{u_2(L_{12})}^{\infty} C_{10}^2 g_0(s_2|s_1) ds_2 + \int_{u_2(L_{02})}^{u_2(L_{12})} (C_{N0}^2 + \xi_0^2(s_2)) \, g_0(s_2|s_1) ds_2 \} \, g_0(s_1) \pi_0 ds_1 \]

\[ + \int_{u_1(L_{01})}^{u_1(L_{11})} \{ C_{N1}^1 + \int_{-\infty}^{u_2(L_{02})} C_{01}^2 g_1(s_2|s_1) ds_2 + \int_{u_2(L_{02})}^{u_2(L_{12})} (C_{N1}^2 + \xi_1^2(s_2)) \, g_1(s_2|s_1) ds_2 \} \, g_1(s_1) \pi_1 ds_1 \]

Denoting

\[ \xi_0^1(s_1) = \int_{u_2(L_{12})}^{\infty} C_{10}^2 g_0(s_2|s_1) ds_2 + \int_{u_2(L_{02})}^{u_2(L_{12})} (C_{N0}^2 + \xi_0^2(s_2)) \, g_0(s_2|s_1) ds_2 \]

\[ \xi_1^1(s_1) = \int_{-\infty}^{u_2(L_{02})} C_{01}^2 g_1(s_2|s_1) ds_2 + \int_{u_2(L_{02})}^{u_2(L_{12})} (C_{N1}^2 + \xi_1^2(s_2)) \, g_1(s_2|s_1) ds_2 \]

(B.36)
it follows that

\[
Q(L_{01}, L_{11}; L^*_2; 4) = \int_{u_1(L_{01})}^{\infty} C_{10} g_0(s_1) \pi_0 ds_1 + \int_{-\infty}^{u_1(L_{01})} C_{01} g_{S_1}(s_1) \pi_1 ds_1
\]

\[
+ \int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{N0} + \xi_0(s_1) \right\} g_0(s_1) \pi_0 ds_1
\]

\[
+ \int_{u_1(L_{01})}^{u_1(L_{11})} \left\{ C_{N1} + \xi_1(s_1) \right\} g_{S_1}(s_1) \pi_1 ds_1
\]  

(B.40)

Taking derivatives w.r.t. \( u_1(L_{01}) \) and \( u_1(L_{11}) \) and setting the derivatives to zero, then we have \( I_0(L_{01}; L^*_2; 4) = 0 \) and \( I_1(L_{01}; L^*_2; 4) = 0 \) where

\[
I_0(L_{01}; L^*_2; 4) = \left\{ C_{10} - C_{N1} - \xi_1(u_1(L_{01})) \right\} g_1(u_1(L_{01})) \pi_1
\]

\[
- \left\{ C_{N0} + \xi_0(u_1(L_{01})) \right\} g_0(u_1(L_{01})) \pi_0
\]

(B.41)

\[
I_1(L_{11}; L^*_2; 4) = \left\{ C_{10} - C_{N0} - \xi_0(u_1(L_{11})) \right\} g_0(u_1(L_{11})) \pi_0
\]

\[
- \left\{ C_{N1} + \xi_1(u_1(L_{11})) \right\} g_1(u_1(L_{11})) \pi_1
\]

Let \( L^*_{01} \) and \( L^*_{11} \) denote the solutions of the above two equations. If \( L^*_{01} > L^*_{11} \), then SNZ is abridged and the current stage becomes the maximum. In this case, \( L^*_{01} \) and \( L^*_{11} \) are replaced by \( L^*_1 = C_{10}/(C_{10} + C_{01}) \).

The selection of all decision boundaries is completed. This process can be generalized for an arbitrary stage \( K \) case.

**Multi-Stage Case**

To formulate \( L_K \) that minimizes (4.21) for fixed other decision boundaries \( L_{01}, L_{11}, \cdots, L_{0K-1}, L_{1K-1} \), we only need to focus on minimizing the part involving \( L_K \), denoted by \( Q(s_{K-1}; L_K) \):

\[
Q(s_{K-1}; L_K) = \int_{u_K(L_K)}^{\infty} C_{10}^K g_0(s_K) \pi_0 ds_K + \int_{-\infty}^{u_K(L_K)} C_{01}^K g_1(s_K) \pi_1 ds_K
\]  

(B.42)
Taking a derivative w.r.t. $u_K(L_K^*)$ and setting the derivative to zero, then we have

$$I_K(L_K) = C_{01}^K g_1(s_{K-1}, u_K(L_K))\pi_1 - C_{10}^K g_0(s_{K-1}, u_K(L_K))\pi_0 \quad (B.43)$$

Applying Theorem 1, it follows that $I_K(L_K) = 0$ where

$$I_K(L_K) = C_{01}^K g_1(u_K(L_K))\pi_1 - C_{10}^K g_1(u_K(L_K))\pi_0 \quad (B.44)$$

The solution of $I_K(L_K) = 0$ is (4.21).

Now given $L_K^*$ and for fixed $L_{01}, L_{11}, \cdots, L_{0K-2}, L_{1K-2}$, we focus on minimizing the part involving $L_{0K-1}$ and $L_{1K-1}$ in (22):

$$Q(s_{K-2}, L_{0K-1}, L_{1K-1}; L_K^*)$$

$$= \int_{u_{K-1}(L_{1K-1})}^{\infty} C_{10}^{K-1} g_0(s_{K-1})\pi_0 ds_{K-1} + \int_{-\infty}^{u_{K-1}(L_{0K-1})} C_{10}^{K-1} g_1(s_{K-1})\pi_1 ds_{K-1} + \int_{u_{K-1}(L_{0K-1})}^{u_{K-1}(L_{1K-1})} \left\{ C_{N0}^{K-1} g_0(s_{K-1})\pi_0 + C_{N1}^{K-1} g_1(s_{K-1})\pi_1 \right\} ds_{K-1} + \int_{u_{K-1}(L_{0K-1})}^{u_{K-1}(L_{1K-1})} Q(s_{K-1}; L_K^*) ds_{K-1} \quad (B.45)$$

Equivalently, we rewrite it as

$$Q(s_{K-2}, L_{0K-1}, L_{1K-1}; L_K^*)$$

$$= \int_{u_{K-1}(L_{1K-1})}^{\infty} C_{10}^{K-1} g_0(s_{K-1})\pi_0 ds_{K-1} + \int_{-\infty}^{u_{K-1}(L_{0K-1})} C_{10}^{K-1} g_1(s_{K-1})\pi_1 ds_{K-1} + \int_{u_{K-1}(L_{0K-1})}^{u_{K-1}(L_{1K-1})} \left\{ C_{N0}^{K-1} g_0(s_{K-1})\pi_0 + C_{N1}^{K-1} g_1(s_{K-1})\pi_1 \right\} ds_{K-1} + \int_{u_{K-1}(L_{0K-1})}^{u_{K-1}(L_{1K-1})} Q(s_{K-1}; L_K^*) ds_{K-1} \quad (B.46)$$

To express it more clearly as a function of $L_{0K-1}$ and $L_{1K-1}$, we first denote $s_{00}^{K-1}(s_{K-1}) =$
\[ Q(s_{K-2}, L_{0K-1}, L_{1K-1}; \lambda_K) \]

\[ = \int_{u_{K-1}(L_{0K-1})}^{\infty} C_{01}^{K-1} g_0(s_{K-1}) \pi_0 ds_{K-1} + \int_{-\infty}^{u_{K-1}(L_{00K-1})} C_{01}^{K-1} g_1(s_{K-1}) \pi_1 ds_{K-1} \]

Taking derivatives w.r.t. \( u_{K-1}(L_{0K-1}) \) and \( u_{K-1}(L_{1K-1}) \), then we have the equation of \( L_{0K-1} \) and \( L_{1K-1} \):

\[ \left\{ C_{01}^{K-1} - C_N^{K-1} - \xi_1^{K-1} (s_{K-2}, u_{K-1}(L_{0K-1})) \right\} g_1(s_{K-2}, u_{K-1}(L_{0K-1})) \pi_1 \]

\[- - \left\{ C_{N0}^{K-1} + \xi_0^{K-1} (s_{K-2}, u_{K-1}(L_{0K-1})) \right\} g_0(s_{K-2}, u_{K-1}(L_{0K-1})) \pi_0 = 0 \]  \hspace{1cm} (B.48)

\[ \left\{ C_{10}^{K-1} - C_N^{K-1} - \xi_1^{K-1} (s_{K-2}, u_{K-1}(L_{1K-1})) \right\} g_1(s_{K-2}, u_{K-1}(L_{1K-1})) \pi_1 \]

\[- - \left\{ C_{N1}^{K-1} + \xi_1^{K-1} (s_{K-2}, u_{K-1}(L_{1K-1})) \right\} g_1(s_{K-2}, u_{K-1}(L_{1K-1})) \pi_1 = 0 \]

By Theorem 1 and Theorem 2, we have \( \xi_y^{K-1}(s_{K-1}) = \xi_y^{K-1}(s_{K-1}) \). (B.48) reduces to

\[ I_{0K-1}(L_{00K-1}; \lambda_K) \]

\[ = \left\{ C_{01}^{K-1} - C_N^{K-1} - \xi_1^{K-1} (u_{K-1}(L_{0K-1})) \right\} g_1(u_{K-1}(L_{0K-1})) \pi_1 \]

\[- - \left\{ C_{N0}^{K-1} + \xi_0^{K-1} (u_{K-1}(L_{0K-1})) \right\} g_0(u_{K-1}(L_{0K-1})) \pi_0 = 0 \]  \hspace{1cm} (B.49)

\[ I_{1K-1}(L_{10K-1}; \lambda_K) \]

\[ = \left\{ C_{10}^{K-1} - C_N^{K-1} - \xi_1^{K-1} (u_{K-1}(L_{1K-1})) \right\} g_0(u_{K-1}(L_{1K-1})) \pi_0 \]

\[- - \left\{ C_{N1}^{K-1} + \xi_1^{K-1} (u_{K-1}(L_{1K-1})) \right\} g_1(u_{K-1}(L_{1K-1})) \pi_1 = 0 \]

The solutions of the equations, denoted by \( L_{0K-1}^* \) and \( L_{1K-1}^* \), respectively, \( L_{0K-1}^* > L_{1K-1}^* \),

there is no observation assigned to \( N \), so the next stage does not exist. Consequently, we can consider the maximum stage is \( K - 1 \) with single boundary \( L_{K-1}^* = \frac{C_{10}^{K-1}}{(C_{10}^{K-1} + C_{01}^{K-1})} \).
Given $\mathbf{L}_{k+1:K}^* = \{L^*_{0,k+1}, L^*_{1,k+1}, \cdots, L^*_K\}$ and for fixed $\mathbf{L}_{k-1} = \{L_0, L_1, \cdots L_{0k-1}, L_{1k-1}\}$, we focus on minimizing the part involving $L_{0k}$ and $L_{1k}$ in (22):

$$Q(s_{k-1}, L_{0k}, L_{1k}; \mathbf{L}_{k+1:K}^*)$$

$$= \int_{u_k(L_{1k})}^{\infty} C^k_{10} g_0(s_k) \pi_0 ds_k + \int_{-\infty}^{u_k(L_{1k})} C^k_{01} g_1(s_k) \pi_1 ds_k$$

$$+ \int_{u_k(L_{0k})}^{u_k(L_{1k})} \{C^k_{N0} g_0(s_k) \pi_0 + C^k_{N1} g_1(s_k) \pi_1\} ds_k$$

$$+ \int_{u_k(L_{0k})}^{u_k(L_{1k})} Q(s_k; \mathbf{L}_{k+1:K}^*) ds_k$$

(B.50)

It can be alternatively expressed as

$$Q(s_{k-1}, L_{0k}, L_{1k}; \mathbf{L}_{k+1:K}^*)$$

$$= \int_{u_k(L_{1k})}^{\infty} C^k_{10} g_0(s_k) \pi_0 ds_k + \int_{-\infty}^{u_k(L_{1k})} C^k_{01} g_1(s_k) \pi_1 ds_k$$

$$+ \int_{u_k(L_{0k})}^{u_k(L_{1k})} \{C^k_{N0} g_0(s_k) \pi_0 + C^k_{N1} g_1(s_k) \pi_1\} ds_k$$

$$+ \int_{u_k(L_{0k})}^{u_k(L_{1k})} \left\{ \int_{u_k(L_{0,k+1})}^{\infty} C^k_{01} g_1(s_{k+1}\{s_k\}) ds_{k+1} \right\} g_0(s_k) \pi_0 ds_k$$

$$+ \int_{u_k(L_{1k})}^{u_k(L_{0,k+1})} \left\{ \int_{-\infty}^{u_k(L_{0,k+1})} C^k_{N0} g_0(s_{k+1}\{s_k\}) ds_{k+1} \right\} g_1(s_k) \pi_1 ds_k$$

$$+ \int_{u_k(L_{1k})}^{u_k(L_{0,k+1})} \left\{ \int_{u_k(L_{0,k+1})}^{u_k(L_{1,k+1})} C^k_{N0} + \xi^k_{0}(s_{k+1}) \right\} g_0(s_{k+1}\{s_k\}) ds_{k+1}$$

$$+ \int_{u_k(L_{1k})}^{u_k(L_{0,k+1})} \left\{ \int_{u_k(L_{0,k+1})}^{u_k(L_{1,k+1})} C^k_{N1} + \xi^k_{1}(s_{k+1}) \right\} g_1(s_{k+1}\{s_k\}) ds_{k+1}$$

(B.51)

To express it more clearly as a function of $L_{0k}$ and $L_{1k}$, we denote

$$\xi^k_{0}(s_{k}) = \int_{u_k(L_{0,k+1})}^{\infty} C^k_{01} g_1(s_{k+1}\{s_k\}) ds_{k+1}$$

$$+ \int_{u_k(L_{0,k+1})}^{u_k(L_{1,k+1})} \left( C^k_{N0} + \xi^k_{0}(s_{k+1}) \right) g_0(s_{k+1}\{s_k\}) ds_{k+1}$$

$$\xi^k_{1}(s_{k}) = \int_{-\infty}^{u_k(L_{1,k+1})} C^k_{01} g_1(s_{k+1}\{s_k\}) ds_{k+1}$$

$$+ \int_{u_k(L_{0,k+1})}^{u_k(L_{1,k+1})} \left( C^k_{N1} + \xi^k_{1}(s_{k+1}) \right) g_1(s_{k+1}\{s_k\}) ds_{k+1}$$

(B.52)

We combine the integrals of neutral zone at stage $k$, then $Q(s_{k-1}, L_{0k}, L_{1k}; \mathbf{L}_{k+1:K}^*)$ can be
simplified into

\[ Q(s_{k-1}, L_{0k}, L_{1k}; \mathbf{L}_{k+1:K}^*) \]

\[ = \int_{u_k}^{\infty} C_{01}^k g_0(s_k) \pi_0 ds_k + \int_{-\infty}^{u_k(L_{0k})} C_{01}^k g_1(s_k) \pi_1 ds_k \]

\[ + \int_{u_k(L_{0k})}^{u_k(L_{1k})} \{ C_{N0}^k + \xi_0^k(s_k) \} g_0(s_k) \pi_0 ds_k \]

\[ + \int_{u_k(L_{0k})}^{u_k(L_{1k})} \{ C_{N1}^k + \xi_1^k(s_k) \} g_1(s_k) \pi_1 ds_k \]  \hspace{1cm} (B.53)

Taking derivatives of \( Q(s_{k-1}, L_{0k}, L_{1k}; \mathbf{L}_{k+1:K}^*) \) w.r.t. \( L_{0k} \) and \( L_{1k} \) and setting the derivatives to zero, then we have

\[ \{ C_{01}^k - C_{N}^k - \xi_1^k(s_{k-1}, u_k(L_{0k})) \} g_1(s_{k-1}, u_k(L_{0k})) \pi_1 \]

\[ - \{ C_{N}^k + \xi_0^k(s_{k-1}, u_k(L_{0k})) \} g_0(s_{k-1}, u_k(L_{0k})) \pi_0 = 0 \]  \hspace{1cm} (B.54)

\[ \{ C_{10}^k - C_{N}^k - \xi_0^k(s_{k-1}, u_k(L_{1k})) \} g_0(s_{k-1}, u_k(L_{1k})) \pi_0 \]

\[ - \{ C_{N}^k + \xi_1^k(s_{k-1}, u_k(L_{1k})) \} g_1(s_{k-1}, u_k(L_{1k})) \pi_1 = 0 \]

By Theorem 2, we have \( \xi_0^k(s_k) = \xi_0^k(s) \) with

\[ \xi_0^k(s_k) = \int_{u_k+1(L_{1k}^*)}^{\infty} C_{01}^{k+1} g_0(s_{k+1}|s_k) ds_{k+1} \]

\[ + \int_{u_k+1(L_{0k}^*)}^{u_k+1(L_{1k}^*)} (C_{N0}^{k+1} + \xi_0^{k+1}(s_{k+1})) g_0(s_{k+1}|s_k) ds_{k+1} \]  \hspace{1cm} (B.55)

\[ \xi_1^k(s_k) = \int_{-\infty}^{u_k+1(L_{0k}^*)} C_{01}^{k+1} g_1(s_{k+1}|s_k) ds_{k+1} \]

\[ + \int_{u_k+1(L_{0k}^*)}^{u_k+1(L_{1k}^*)} (C_{N1}^{k+1} + \xi_1^{k+1}(s_{k+1})) g_1(s_{k+1}|s_k) ds_{k+1} \]

It follows that Theorem 1, (B.54) reduces to \( I_{0k}(L_{0k}; \mathbf{L}_{k+1:K}^*) = 0 \) and \( I_{1k}(L_{1k}; \mathbf{L}_{k+1:K}^*) = 0 \) where

\[ I_{0k}(L_{0k}) = \{ C_{01}^k - C_{N1}^k - \xi_1^k(u_k(L_{0k})) \} g_1(u_k(L_{0k})) \pi_1 \]

\[ - \{ C_{N0}^k + \xi_0^k(u_k(L_{0k})) \} g_0(u_k(L_{0k})) \pi_0 \]  \hspace{1cm} (B.56)

\[ I_{1k}(L_{1k}) = \{ C_{10}^k - C_{N0}^k - \xi_0^k(u_k(L_{1k})) \} g_0(u_k(L_{1k})) \pi_0 \]

\[ - \{ C_{N1}^k + \xi_1^k(u_k(L_{1k})) \} g_1(u_k(L_{1k})) \pi_1 \]

83
The solutions of the above equations are denoted by $L^*_{0k}$ and $L^*_{1k}$. If $L^*_{0k} > L^*_{1k}$ there is no observation assigned to $N$. Consequently, we can consider the maximum stage is $k$ with a boundary $L_k$ for which the optimal value is $L^*_k = C_{10}^k / (C_{10}^k + C_{01}^k)$. Next, $k$ is replaced by $k-1$ and the iteration step for stage $k$ is repeated until all decision boundaries are selected.
Appendix C

Supplementary Materials

The R code includes the recursive and simultaneous algorithms to derive decision boundaries for SNZ and provides decision boundaries for SNZ3, LSNZ, and the non-sequential Bayes classifier boundary. The code allows both heterogeneous and homogeneous Gaussian assumptions with unstructured variance-covariance matrices, but the recursive algorithm is only available for homogeneous Guassian cases. It also provides for evaluation of performance such as the overall expected cost, the conditional misclassification rates, and the expected sample size.

C.1 Manual

This is the guide to function snz() written in R to derive decision boundaries of sequential neutral zone classifiers under the multivariate normal distribution with homogeneous variance-covariance matrices.
SNZ

This function is used to derive decision boundaries for the posterior probabilities of class membership 1. The expected cost, the average sample number (stopping time), and error rates are available as measures of classification performance.

Usage

\texttt{snz(m0, m1, cov, c10, c01, cn0, cn1, pi1, method, detail, int)}

Arguments

\texttt{m0} \(K \times 1\) vector of mean profiles for class 0

\texttt{m1} \(K \times 1\) vector of mean profiles for class 1

\texttt{cov} \(K \times K\) matrix of common variance-covariance

\texttt{c10} \(K \times 1\) vector of misclassification costs given class 0

\texttt{c01} \(K \times 1\) vector of misclassification costs given class 1

\texttt{cn0} \((K - 1) \times 1\) vector of neutral-zone costs given class 0

\texttt{cn1} \((K - 1) \times 1\) vector of neutral-zone costs given class 1

\texttt{pi1} The prior probability of class 1. \texttt{pi0=1-pi1}. Default is \texttt{pi1=0.5}.

\texttt{method} The classification methods to be used: 1 (\texttt{recursive}), 2 (\texttt{simultaneous}), 3 (\texttt{snz3}), 4 (\texttt{lsnz}), and 5 (\texttt{bayes}). Default is \texttt{method=1}.
**detail** Options for performance outputs. If `detail=FALSE`, return the total performance outputs. `detail=TRUE`, return the performance outputs incurred at each stage. Default is `FALSE`.

**int** Initial values for decision boundaries to be searched. If `NULL`, default initial values are used.

**Details**

A required R package is `mnormt` to compute the multivariate normal probability.

A variance-covariance matrix can be specified. Use `cs(\sigma^2, \rho)` for the compound symmetry(CS) with variance \( \sigma^2 \) and covariance \( \rho \). Use `ar1(\sigma^2, \rho)` for AR(1) with variance \( \sigma^2 \) and covariance \( \rho^{\lfloor d \rfloor} \) at lag \( d \).

Note that means of two classes should differ at every stage.

`c10` and `c01` should be greater than `cn0` and `cn1` at every stage.

Two methods `recursive` and `simultaneous` are for classifier "SNZ". `recursive` uses backward induction and `simultaneous` uses simultaneous searching method. `recursive` is often faster and more accurate in large \( K \).

For `method=c(1,2)`, specify \( 2(K-1)+1 \) initial values, e.g., `int=c(rep(0,2,K-1),rep(0,8,K-1),0.5)`.

For `method=3`, specify three initial values, e.g., `int=c(0,2,0,8,0.5)`.

**C.2 Codes**
snz <- function (m0, m1, cov, c01, cn0, cn1, pi1, method, detail, int) {
  K <- length(m0)
  m <- K - 1
  ## default setting
  if (missing(pi1) || is.null(pi1)) {pi1 <- 0.5}
  pi0 = 1 - pi1
  if (missing(method) || is.null(method)) {method <- 1}
  if (missing(detail) || is.null(detail)) {detail <- FALSE}

  # Create Weight matrix W
  weight <- function (n) {
    w <- matrix(NA, n, n)
    for (i in 1:n) {
      for (j in 1:n) {
        if (i < j) {
          w[i, j] <- 0
        } else {
          c <- (m1 - m0)[1:i] %*% solve(cov[1:i, 1:i])
          w[i, j] <- c[j]
        }
      }
    }
    return (w)
  }

  sol.L0 <- as.vector(rep(NA, K))
  sol.L1 <- as.vector(rep(NA, K))

  W <- weight(K)
  s_cov <- W %*% cov %*% t(W)
  s_m0 <- as.vector(t(W %*% m0))
  s_m1 <- as.vector(t(W %*% m1))

  ## Selection of 'llim' and 'ulim'
  margin <- function (k, L) {

\[ s \leftarrow -t(m_1 - m_0)[1:k]\%\%solve(cov[1:k,1:k])\%\%(m_1 + m_0)[1:k]/2+ \]
\[ \log(\pi_0/\pi_1) + \log(L/(1-L)) \]
\[ \text{mar} \leftarrow \text{dnorm}(s, \text{mean} = s_m_0[k], \text{sd} = s_{cov}[k,k]^{0.5})^{\pi_0} + \]
\[ \text{dnorm}(s, \text{mean} = s_m_1[k], \text{sd} = s_{cov}[k,k]^{0.5})^{\pi_1} \]
\[ \text{return}(\text{mar}) \]
\]

llim <- as.vector(rep(c(NA), 2*m))
ulim <- as.vector(rep(c(NA), 2*m))
tol <- 1e-06

for (k in m:1) {
  if (margin(k, tol) > tol || margin(k, 1-tol) > tol) {
    llim[k] <- tol
    llim[k+m] <- tol
    ulim[k] <- 1-tol
    ulim[k+m] <- 1-tol
  } else {
    for (i in 1:999) {
      L <- i/1000
      if (margin(k, L) > 1e-310) break
    }
    llim[k] <- L
    llim[k+m] <- L
    for (i in 999:1) {
      U <- L + (1-L)*i/1000
      if (margin(k, U) > 1e-310) break
    }
    ulim[k] <- U
    ulim[k+m] <- U
    if (llim[k] == ulim[k]) {
      llim[k] <- tol
      llim[k+m] <- tol
      ulim[k] <- 1-tol
      ulim[k+m] <- 1-tol
    }
  }
}

llim[k] <- -L
llim[k+m] <- -L
for (i in 999:1) {
  U <- L + (1-L)*i/1000
  if (margin(k, U) > 1e-310) break
}
ulim[k] <- U
ulim[k+m] <- U
if (llim[k] == ulim[k]) {
  llim[k] <- tol
  llim[k+m] <- tol
  ulim[k] <- 1-tol
  ulim[k+m] <- 1-tol
}
## Calculate the EC given the vectors of L0 and L1

```
# Define domain of integration for evaluating EC
for (j in 1:K){
  rL0[j]<-t((m1-m0)[1:j])%*%solve(cov[1:j,1:j])%*%((m1+m0)[1:j]/2+log(pi0/pi1)+log(L0[j]/(1-L0[j]))

  rL1[j]<-t((m1-m0)[1:j])%*%solve(cov[1:j,1:j])%*%((m1+m0)[1:j]/2+log(pi0/pi1)+log(L1[j]/(1-L1[j]))

  lower_10[j]<-as.vector(rep(NA,j))
  lower_n[j]<-as.vector(rep(NA,j))
  lower_01[j]<-as.vector(rep(NA,j))
  upper_10[j]<-as.vector(rep(NA,j))
  upper_n[j]<-as.vector(rep(NA,j))
  upper_01[j]<-as.vector(rep(NA,j))

  for(i in 1:j){
    if(i<j){
      lower_10[i]<-rL0[i]
      lower_01[i]<-rL0[i]
      lower_n[i]<-rL0[i]
    }
  }
}
```
upper_10[i] <- rL1[i]
upper_n[i] <- rL1[i]
upper_01[i] <- rL1[i]
} else if (i != K) {
lower_10[i] <- rL1[i]
lower_01[i] <- Inf
lower_n[i] <- rL0[i]
upper_10[i] <- Inf
upper_n[i] <- rL1[i]
upper_01[i] <- rL0[i]
} else {
lower_10[i] <- rL1[i]
lower_01[i] <- Inf
lower_n[i] <- 0
upper_10[i] <- Inf
upper_n[i] <- 0
upper_01[i] <- rL0[i]
}
if (j == K) {
Cost[j] <- c10[j] * sadmvn(lower = lower_10, upper = upper_10, mean = s_m0[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi0 +
c01[j] * sadmvn(lower = lower_01, upper = upper_01, mean = s_m1[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi1
} else {
Cost[j] <- c10[j] * sadmvn(lower = lower_10, upper = upper_10, mean = s_m0[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi0 +
c01[j] * sadmvn(lower = lower_01, upper = upper_01, mean = s_m1[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi1 +
cn0[j] * sadmvn(lower = lower_n, upper = upper_n, mean = s_m0[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi0 +
cn1[j] * sadmvn(lower = lower_n, upper = upper_n, mean = s_m1[1:j],
                          varcov = s_cov[1:j, 1:j]) * pi1}
\begin{verbatim}
    varcov = s_cov[1:j, 1:j] * pi1

    nz0[j] <- sadmvn(lower=lower_n, upper=upper_n, mean=s_m0[1:j],
                      varcov=s_cov[1:j, 1:j])
    nz1[j] <- sadmvn(lower=lower_n, upper=upper_n, mean=s_m1[1:j],
                      varcov=s_cov[1:j, 1:j])

    if(j == 1) {
        stoprate0[j] <- 1 - nz0[j]
        stoprate1[j] <- 1 - nz1[j]
        stoprate[j] <- stoprate0[j] * pi0 + stoprate1[j] * pi1
        stoptime[j] <- j * stoprate0[j] * pi0 + j * stoprate1[j] * pi1
        errorrate10[j] <- sadmvn(lower=lower_10, upper=upper_10,
                                  mean=s_m0[1:j], varcov=s_cov[1:j, 1:j])
        errorrate01[j] <- sadmvn(lower=lower_01, upper=upper_01,
                                  mean=s_m1[1:j], varcov=s_cov[1:j, 1:j])
    } else {
        stoprate0[j] <- nz0[j - 1] - nz0[j]
        stoprate1[j] <- nz1[j - 1] - nz1[j]
        stoprate[j] <- stoprate[j - 1] + stoprate0[j] * pi0 + stoprate1[j] * pi1
        stoptime[j] <- j * stoprate0[j] * pi0 + j * stoprate1[j] * pi1
        errorrate10[j] <- errorrate10[j - 1] + sadmvn(lower=lower_10, upper=upper_10,
                                                  mean=s_m0[1:j], varcov=s_cov[1:j, 1:j])
        errorrate01[j] <- errorrate01[j - 1] + sadmvn(lower=lower_01, upper=upper_01,
                                                  mean=s_m1[1:j], varcov=s_cov[1:j, 1:j])
    }

    if(detail == FALSE) {
        list(error10 = errorrate10[K], error01 = errorrate01[K],
             stoptime = sum(stoptime), EC = sum(Cost))
    } else {
        list(error10 = errorrate10, error01 = errorrate01,
             stoptime = sum(stoptime), EC = sum(Cost))
    }
\end{verbatim}
stoprate = stoprate, stoptime = sum(stoptime), EC = Cost
}

## SNZ solution by recursive algorithm (method 1)
if (method == "recursive" | method == 1) {

# Solution of the final stage K
sol.L0[K] <- c10[K] / (c01[K] + c10[K])

# Define future risk for group 0 (errortype = 10) or 1 (errortype = 01)
risk <- function(k, L, errortype) {
  u <- k + 1
  n <- K - k
  L0 <- as.vector(rep(NA, n))
  L1 <- as.vector(rep(NA, n))
  rL0 <- as.vector(rep(NA, n))
  rL1 <- as.vector(rep(NA, n))
  sumr <- 0
  s <- t(m1 - m0)[1:k] %*% solve(cov[1:k, 1:k]) %*% (m1 + m0)[1:k] / 2 +
       log(pi0 / pi1) + log(L / (1 - L))

  for (j in 1:n) {
    l <- j + k
    L0[j] <- sol.L0[l]
    L1[j] <- sol.L1[l]
    rL0[j] <- t(m1 - m0)[1:l] %*% solve(cov[1:l, 1:l]) %*% (m1 + m0)[1:l] / 2
    + log(pi0 / pi1) + log(L0[j] / (1 - L0[j]))
    rL1[j] <- t(m1 - m0)[1:l] %*% solve(cov[1:l, 1:l]) %*% (m1 + m0)[1:l] / 2
    + log(pi0 / pi1) + log(L1[j] / (1 - L1[j]))
    con_m0 <- s_m0[u:1] + s_cov[u:1, k] / s_cov[k, k] * (s - s_m0[k])
    con_s0 <- s_cov[u:1, u:1] - s_cov[u:1, k] %*% solve(s_cov[k, k]) %*% s_cov[k, u:1]
  }
}
con_m1 <- s_m1[u:1] + s_cov[u:1,k]/s_cov[k,k]*(s-s_m1[k])
con_s1 <- s_cov[u:1,u:1] - s_cov[u:1,k]*%solve(s_cov[k,k])*s_cov[k,u:1]

lower_10 <- as.vector(rep(NA,j))
lower_n <- as.vector(rep(NA,j))
lower_01 <- as.vector(rep(NA,j))
upper_10 <- as.vector(rep(NA,j))
upper_n <- as.vector(rep(NA,j))
upper_01 <- as.vector(rep(NA,j))

for (i in 1:j){
  if (i<j){
    lower_10[i] <- rL0[i]
    lower_01[i] <- rL0[i]
    lower_n[i] <- rL0[i]
    upper_10[i] <- rL1[i]
    upper_n[i] <- rL1[i]
    upper_01[i] <- rL0[i]
  } else if (i!=n){
    lower_10[i] <- rL1[i]
    lower_01[i] <- -Inf
    lower_n[i] <- rL0[i]
    upper_10[i] <- Inf
    upper_n[i] <- rL1[i]
    upper_01[i] <- rL1[i]
  } else {
    lower_10[i] <- rL1[i]
    lower_01[i] <- -Inf
    lower_n[i] <- 0
    upper_10[i] <- Inf
    upper_n[i] <- 0
    upper_01[i] <- rL0[i]
  }
}
Compute Risk(L, errotype==10)

if (errotype==10){
  r<- c10[j]*sdmvn(lower=lower_10, upper=upper_10, mean=con_m0[1:j],
                 varcov=con_s0)+
  cn0[j]*sdmvn(lower=lower_n, upper=upper_n, mean=con_m0[1:j],
                 varcov=con_s0)
  sumr<-sumr+r[1]
} else{
  # Compute Risk(L, errotype==01)
  r<- c01[j]*sdmvn(lower=lower_01, upper=upper_01, mean=con_m1[1:j],
                 varcov=con_s1)+
  cn1[j]*sdmvn(lower=lower_n, upper=upper_n, mean=con_m1[1:j],
                 varcov=con_s1)
  sumr<-sumr+r[1]
}

return(sumr)

## Backward induction from stage K-1 to stage 1
for (k in m:1){
## Object function for L0k
solve_L0<--function(L){
  s<-t(m1-m0)[1:k]*solve(cov[1:k,1:k])*%(m1+m0)[1:k]/2+
      log(pi0/pi1)+log(L/(1-L))
  px0<-dnorm(s,mean=s_m0[k],sd=s_cov[k,k]^0.5)*pi0
  px1<-dnorm(s,mean=s_m1[k],sd=s_cov[k,k]^0.5)*pi1
  px<-px0+px1
  pel0<-c01[k]*px1/px
  pel1<-c10[k]*px0/px
  peln<--((cn1[k+1]+risk(k,L,01))*px1+(cn0[k+1]+risk(k,L,10))*px0)/px
  I_0<--pel0-peln
}
## Object function for L1k

```r
solve_L1 <- function (L) {
  s <- -(m1 - m0)[1:k] %*% solve(cov[1:k, 1:k]) %*% (m1 + m0)[1:k] / 2 +
  log(pi0 / pi1) + log(L / (1 - L))
  px0 <- dnorm(s, mean = s_m0[k], sd = s_cov[k, k]^0.5) * pi0
  px1 <- dnorm(s, mean = s_m1[k], sd = s_cov[k, k]^0.5) * pi1
  px <- px0 + px1
  pel0 <- c01[k] * px1 / px
  pel1 <- c10[k] * px0 / px
  peln <- ((cn1[k + 1] + risk(k, L, 01)) * px1 + (cn0[k + 1] + risk(k, L, 10)) * px0) / px
  I_1 <- pel1 - peln
  return(I_1)
}
```

## If a object function at llim and ulim have same signs, L0k=0 and L1k=1

```r
if (solve_L0(llim[k]) * solve_L0(ulim[k]) > 0) {
  root_L0 <- llim[k]
} else {
  root_L0 <- uniroot(solve_L0, lower = llim[k], upper = ulim[k])$root
}
if (solve_L1(llim[k]) * solve_L1(ulim[k]) > 0) {
  root_L1 <- ulim[k]
} else {
  root_L1 <- uniroot(solve_L1, lower = llim[k + m], upper = ulim[k + m])$root
}
```

## Argument for choosing final value of thresholds

```r
if (root_L0 > root_L1) {
  K <- k
  sol_L0 <- as.vector(rep(NA, K))
  sol_L1 <- as.vector(rep(NA, K))
  sol_L0[k] <- c10[k] / (c01[k] + c10[k])
  sol_L1[k] <- c10[k] / (c01[k] + c10[k])
} else {
  sol_L0[k] <- root_L0
}
```
297  sol.L1[k]<-root.L1
298  }
299  }
300 } else if(method=="simultaneous"|method==2){
301  joint Boundary<-function(thresholds) {
302  L0<-as.vector(rep(c(NA),m))
303  L1<-as.vector(rep(c(NA),m))
304  L<- thresholds[2*m+1]
305  detail<-FALSE
306  for(j in 1:m){
307   L0[j]<-thresholds[j]
308   L1[j]<-thresholds[j+m]
309  }
310  return(evaluation(c(L0,L),c(L1,L),detail=FALSE)$EC)
311 }
312  lim<-c10[1:m]/(c01[1:m]+c10[1:m])
313  if(missing(int)||is.null(int)){
314   int0<-cn0[1:m]/(c01[1:m]-cn1[1:m]+cn0[1:m])
315   int1<-1-cn0[1:m]/(c10[1:m]-cn0[1:m]+cn1[1:m])
316   int<-c(int0,int1,c10[K]/(c01[K]+c10[K]))
317 }
318  result<-optim(int, joint Boundary, method = "L-BFGS-B",
319                lower=c(llim,lim,llim[m]),upper=c(lim,ulim,ulim[m]))
320  a<-m+1
321  b<-2*m+1
322  sol.L0=c(result$par[1:m],result$par[b])
323  sol.L1=result$par[a:b]
324 }
325 } else if(method=="snz3"|method==3){
326  Fixed Boundary<-function(thresholds) {
327  L0<-as.vector(rep(c(NA),m))
328  L1<-as.vector(rep(c(NA),m))
L0[1:m]<-thresholds[1]
L<-thresholds[3]
return(evaluation(c(L0,L),c(L1,L),detail=FALSE)$EC)
}
if(missing(int)||is.null(int)){
    int<-c(cn0[K]/c01[K],1-cn1[K]/c10[K],c10[K]/(c01[K]+c10[K]))
}
lim<-c10[K]/(c01[K]+c10[K])
if(int[1] <= llim[m] || int[1] >= ulim[m]){ int[1]<-llim[m]+0.1*(ulim[m]-llim[m])
    lim<-0.5*(ulim[m]+llim[m])
}
if(int[2] <= llim[m] || int[2] >= ulim[m]){ int[2]<-ulim[m]-0.1*(ulim[m]-llim[m])
    lim<-0.5*(ulim[m]+llim[m])
}
result<-optim(int,Fixed_boundary,method = "L-BFGS-B",
               lower=c(llim[m],lim),upper=c(lim,ulim[m]))
sol.L0=c(rep(result$par[1],m),result$par[3])
sol.L1=c(rep(result$par[2],m),result$par[3])
else if (method=="lsemz"|method==4){
    for(j in 1:m){
        sol.L0[j]<-cn0[j]/(c01[j]-cn1[j]+cn0[j])
        sol.L1[j]<-1-cn1[j]/(c10[j]-cn0[j]+cn1[j])
    }
    sol.L0[K]<-c10[K]/(c01[K]+c10[K])
    sol.L1[K]<-c10[K]/(c01[K]+c10[K])
}
if (method=="bayesalldata"|method==5){
L <- c10[K]/(c01[K]+c10[K])

rL <- t(m1-m0)%*%solve(cov)%*%(m1+m0)/2+log(pi0/pi1)+log(L/(1-L))

sol.L0 <- c(rep(0,m),L)
sol.L1 <- c(rep(1,m),L)

Cost <- c10[K]*(1-pnorm(rL, mean=s_m0[K], sd=s_cov[K,K]^0.5))*pi0 +
    c01[K]*pnorm(rL, mean=s_m1[K], sd=s_cov[K,K]^0.5)*pi1 +
    sum(cn0[1:K-1])*pi0+sum(cn1[1:K-1])*pi1

error10 <- (1-pnorm(rL, mean=s_m0[K], sd=s_cov[K,K]^0.5))
error01 <- pnorm(rL, mean=s_m1[K], sd=s_cov[K,K]^0.5)

if(detail==FALSE){
    list(L0=sol.L0, L1=sol.L1, EC=Cost[1,1], stoptime=K,
        error10=error10[1,1], error01=error01[1,1])
} else {
    list(L0=sol.L0, L1=sol.L1,
        EC=Cost[1,1],
        stoptime=K,
        error10=c(rep(0,K-1),error10[1,1]),
        error01=c(rep(0,K-1),error01[1,1]),
        stoprate=c(rep(0,K-1),1))
}

} else {
    list(L0=sol.L0, L1=sol.L1, detail)
    if(detail==FALSE){
        list(L0=sol.L0,
            L1=sol.L1,
            EC=ev$EC,
            EN=ev$stoptime,
            error10=ev$error10,
            error01=ev$error01)
    } else {
        list(L0=sol.L0,
            L1=sol.L1,
            EC=ev$EC,

99
EN=1-ev\$stoprate,
error10=ev\$error10,
error01=ev\$error01)
}