Georgia State University
Digital Archive @ GSU

Mathematics Theses | Department of Mathematics and Statistics

11-19-2008

Semi-Parametric Inference for the Partial Area Under the ROC Curve

Fangfang Sun

Follow this and additional works at: http://digitalarchive.gsu.edu/math_theses

Recommended Citation
Sun, Fangfang, "Semi-Parametric Inference for the Partial Area Under the ROC Curve" (2008). *Mathematics Theses.* Paper 61.

This Thesis is brought to you for free and open access by the Department of Mathematics and Statistics at Digital Archive @ GSU. It has been accepted for inclusion in Mathematics Theses by an authorized administrator of Digital Archive @ GSU. For more information, please contact digitalarchive@gsu.edu.

SEMI-PARAMETRIC INFERENCE FOR THE PARTIAL AREA UNDER THE ROC CURVE

by

FANGFANG SUN

Under the Direction of Gengsheng Qin

ABSTRACT

Diagnostic tests are central in the field of modern medicine. One of the main factors for interpreting a diagnostic test is the discriminatory accuracy. For a continuous-scale diagnostic test, the area under the receiver operating characteristic (ROC) curve, AUC, is a useful one-number summary index for the diagnostic accuracy of the test. When only a particular region of the ROC curve would be of interest, the partial AUC (pAUC) is a more appropriate index for the diagnostic accuracy. In this thesis, we develop seven confidence intervals for the pAUC under the semi-parametric models for the diseased and non-diseased populations by using the normal approximation, bootstrap and empirical likelihood methods. In addition, we conduct simulation studies to compare the finite sample performance of the proposed confidence intervals for the pAUC. A real example is also used to illustrate the application of the recommended intervals.

INDEX WORDS: ROC, AUC, The partial AUC, Diagnostic test, Confidence interval

SEMI-PARAMETRIC INFERENCE FOR THE PARTIAL AREA UNDER THE ROC CURVE

by

FANGFANG SUN

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

In the College of Arts and Sciences

Georgia State University

2008

Copyright by

Fangfang Sun

2008

SEMI-PARAMETRIC INFERENCE FOR THE PARTIAL AREA

UNDER THE ROC CURVE

by

FANGFANG SUN

Major Professor: Dr. Gengsheng Qin
Committee: Dr. Yu-Sheng Hsu
Dr. Yixin Fang
Dr. Yuanhui Xiao

Electronic Version Approval:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2008

ACKNOWLEDGEMENTS

I would like to acknowledge those who have helped me in the completion of this thesis. First of all, I like to thank my advisor, Dr. Gengsheng Qin, for his patience and guidance. I would also like to acknowledge the other members of my thesis committee, Dr. Yu-Sheng Hsu, Dr. Yixin Fang and Dr. Yuanhui Xiao for taking the time to read this thesis and provide useful comments. Lastly, I want to acknowledge the support of my family and friends.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS……iv

LIST OF TABLES……..vi

CHAPTERS

1. INTRODUCTION…….1
2. NORMAL APPROXIMATION AND BOOTSTRAP BASED CONFIDENCE INTERVALS FOR THE PARTIAL AUC…….…...6
   - 2.1 Normal approximation based confidence interval ……...……..7
   - 2.2 Bootstrap based confidence intervals…….....……..……9
3. EMPIRICAL LIKELIHOOD BASED SEMI-PARAMETRIC INFERENCE FOR THE PARTIAL AUC……...11
4. SIMULATION STUDY……...16
   - 4.1 Normal distributions estimators……..……..16
   - 4.2 Exponential distributions estimators…….17
5. DUCHENNE MUSCULAR DYSTROPHY EXAMPLES…….18
6. CONCLUSION AND DISCUSSION…….….……...21

REFERENCES…….……...22

APPENDICES...…….…….24

- APPENDIX A: SIMULATION TABLES….....…….…….24
- APPENDIX B: REAL DATA ANALYSIS TABLES…….…….29
- APPENDIX C: THE S-PLUS CODE FOR SIMULATION STUDIES…….…….32
- APPENDIX D: THE S-PLUS CODE FOR REAL DATA ANALYSIS…….38

LIST OF TABLES

Table I: Normal distribution: level of 95 percent confidence interval for the AUC…...….....25

Table II: Normal distribution: level of 90 percent confidence interval for the pAUC…….…...26

Table III: Exponential distribution: level of 95 percent confidence interval for the pAUC ...….27

Table IV: Exponential distribution: level of 90 percent confidence interval for the pAUC…….28

Table V: Duchenne Muscular Dystrophy Example: Level of 95 percent confidence intervals for the normalized pAUC…….……...……..30

Table VI: Duchenne Muscular Dystrophy Example: Level of 90 percent confidence intervals for the normalized pAUC……...…….....31

CHAPTER 1

INTRODUCTION

Diagnostic tests are central by helping physicians revise disease probability from their patients in the field of modern medicine. One of the main factors for interpreting a diagnostic test is the discriminatory accuracy which is the ability of the diagnostic test to distinguish between two groups of subjects, usually non-diseased and diseased subjects (Shapiro, 1999). As a result, statistical studies for assessing discriminatory accuracy have been recently received much attention. As Shapiro mentioned in his paper, most of diagnostic tests are continuous. Therefore, the discriminatory accuracy of continuous-scale tests will only be discussed in this thesis.

The accuracy of a binary diagnostic test can be measured by its specificity and sensitivity. The sensitivity or true positive rate (TPR) of the test is the proportion that the test is positive among diseased patients. The specificity or true negative rate (TNR) of the test is the proportion that the test will be negative among non-diseased patients. Let Y and X be the results of a continuous-scale test for a diseased subject and a non-diseased subject with cumulative distribution functions G and F, respectively. For a given cut-off point c, the sensitivity and specificity of the test are defined as

$$Se = P(Y \geq c | D) = 1 - G(c), \qquad Sp = P(X \leq c | \overline{D}) = F(c),$$

respectively, where D denotes a diseased state and $\overline{D}$ denotes a non-diseased state. Equivalently, the false positive rate (FPR) and false negative rate (FNR) of the test are defined as (1-specificity) $=1-F(c)$ and (1-sensitivity) = $G(c)$, respectively. In biological, medical, and health service research, the receiver operating characteristic (ROC) curve, which is a plot of sensitivity against FPR for all possible cut-off points c, is a useful

graphical summary of the diagnosis accuracy of a test. The ROC curve can mathematically be represented by $R(p) = 1 - G(F^{-1}(1-p))$, where F^{-1} is the inverse function of F. $R(p)$ is indeed the sensitivity of the test when the specificity is fixed at level *(1-p)*. The ROC curve was derived from statistical decision theory and originally developed in the literature of electronic signal detection (Hanley, 1989). As Shapiro (1999) reviewed in his paper, many approaches have been proposed to make inference about the ROC curve, such as, parametric approaches proposed by Strike (1995), Goddard et al. (1990), Egan (1975) and England WL (1988); a semi-parametric algorithm, LABROC4, proposed by Metz et al. (1998); and non-parametric approaches proposed by Zou et al. (1997), Le (1997) and Lloyd (1998). The ROC curve best summarizes the diagnostic accuracy by plotting sensitivity versus FPR for all possible cut-off point c. However, since a one-number summary index for the diagnostic accuracy is often desired, the area under the ROC curve (AUC), defined as $\delta = \int_0^1 R(p)dp$, becomes a popular summary of the accuracy of a diagnostic test across all the possible cut-off points. The larger is the AUC, the better performance the test will have. In other word, the value of δ closer to 1 indicates the higher diagnostic accuracy of the test.

The AUC is a global summary of the diagnostic accuracy, but it has some limitations (Shapiro, 1999; Hilden 1991). For instance, when two ROC curves cross, the two tests can have similar AUC even though one test has higher sensitivity for certain specificities while the other test has better sensitivity for other specificities; the AUC includes regions of ROC space that would not be of practical interest (e.g. very high FPR, or very low TPR) (Shapiro, 1999). As a result, in case that it is critical to maintain a

particular range of specificity or sensitivities in some diagnostic testing, such as a low false positive rate or a high sensitivity, the AUC may not be an appropriate accuracy index for diagnostic tests because only the region of ROC curve corresponding to acceptable low false positive rates or high sensitivities would be of interest. For example, when the FPR must be very small to be acceptable for cancer screening (Lilienfeld, 1974), interest would only be in the lower tail of ROC curve. Alternatively, the partial AUC (pAUC) was proposed for some diagnostic tests in which only a particular region of the ROC curve would be of interest. The pAUC over the interval of false positive rates (p_0, p_1), denoted by $\delta_{p_0 p_1}$, is

$$\delta_{p_0 p_1} = \int_{p_0}^{p_1} R(p)dp \quad \text{, for } 0 \le p_0 < p_1 \le 1 .$$

It can be described as the cumulative value of sensitivity for all possible values of the false positive rates in the interval (p_0, p_1).

In recent years, many parametric and nonparametric approaches have been proposed in making inference about the partial AUC. Parametric methods based on the bi-normal model were proposed by McClish (1989), Thompson and Zucchini (1989), and Jiang, Metz, and Nishikawa (1996). However, a major concern of these parametric methods is that they are quite sensitive to model departures from the distributional assumptions (Walsh, 1997). A generalized nonparametric method was proposed by Wieand et al (1989) for the inference of both the full and the partial AUC. But the method is mathematically too complicated to be well applied in practice since it is involved in density and distribution function estimations (Qin et al. 2006). Another nonparametric method for the pAUC was proposed by Zhang et al. (2002). However, this

method is valid only for ordinal-scale data. Let $X_1, X_2, \ldots, X_m$ be the test results from a random sample of non-diseased population with distribution function F; let $Y_1, Y_2, \ldots, Y_n$ be the test results from a random sample of diseased population with distribution function G. Dodd and Pepe (2003) proposed the following consistent nonparametric estimator for the pAUC:

$$\hat{\delta}_{p_0 p_1} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I(Y_j \geq X_i) I\left(X_i \in (\hat{q}_1, \hat{q}_0)\right)$$

where $\hat{q}_l = \hat{F}^{-1}(1-p_l)$ (l=0,1) is the $(1-p_l)$-th sample quantile, and $\hat{F}$ is the empirical distribution of F. But they did not provide the asymptotic variance of the estimator. Instead, they recommended using the bootstrap to obtain the variance estimate. Qin and Zhou (2006) proposed an Empirical Likelihood (EL) based approach to derive confidence intervals for the full AUC. Qin, Jin and Zhou (2007) derived the asymptotic normal distribution for the estimator $\hat{\delta}_{p_0 p_1}$ and developed EL-based nonparametric confidence intervals for the partial AUC.

Semi-parametric methods are frequently used in statistical literature because they share the advantages of both parametric and nonparametric methods. They are more flexible and robust than the traditional parametric methods. On the other hand they may be more efficient than the pure non-parametric methods and usually have better finite sample performances than their non-parametric counterparts. Semi-parametric methods have been used for making inference of ROC curves. For instance, Li *et al.* (1999) proposed a non-parametric approach to estimate the distribution of test results in non-diseased subjects, whereas assuming a parametric model for the distribution of test results in diseased subjects. Dodd and Pepe (2003) proposed a semi-parametric regression model

for evaluating covariate effects on ROC curves. Qin and Zhang (2003) developed a semi-parametric approach by assuming a density ratio model for disease and disease-free densities. In this thesis, we make a general semi-parametric model assumption for the test results X and Y. We shall propose a semi-parametric estimator for the pAUC and derive the asymptotic normal distribution of the estimator. Seven confidence intervals for the partial AUC based on the semi-parametric models are developed by using the normal approximation, bootstrap and empirical likelihood methods.

The thesis is organized as follows: In Chapter 2, we propose a normal approximation based confidence interval and two bootstrap-based confidence intervals for the partial AUC. In Chapter 3 we propose four EL-based intervals for the partial AUC. Simulation studies are conducted in Chapter 4 to evaluate the finite sample performances of these intervals. Chapter 5 uses a real data set to illustrate the application of the proposed intervals. Finally, the conclusions are discussed in Chapter 6.

CHAPTER 2

NORMAL APPROXIMATION AND BOOTSTRAP BASED CONFIDENCE INTERVALS FOR THE PARTIAL AUC

Consider one diagnostic test which yields continuous measurements and is performed on *m* non-diseased subjects and *n* diseased cases. Let $X_1, X_2, \ldots, X_m$ be the test results of a random sample of non-diseased subjects with an unknown cumulative distribution function *F*, and $Y_1, Y_2, \ldots, Y_n$ the test results of a random sample of diseased subjects with a parametric distribution function *G(y; θ),* where $\theta \in \Theta$ is an unknown parameter for some set $\Theta \subset R^q, q \geq 1$. In this setting, we can rewrite the partial AUC as

$$\begin{aligned}\delta_{p_0 p_1} &= \int_{p_0}^{p_1} R(p)dp \\ &= P\{Y > X, X \in (q_1, q_0); \theta\} \\ &= E[P(Y > X; \theta \mid X \in (q_1, q_0))] \\ &= E[V(X; \theta)]\end{aligned}$$

for $0 \leq p_0 < p_1 \leq 1$, where $R(p) = 1 - G(F^{-1}(1-p); \theta)$ is the ROC curve of the test, and $q_k = F^{-1}(1-p_k), k = 0,1$ are the $(1-p_k)$-th quantiles of *F*, and

$$\begin{aligned}V(X;\theta) &= P(Y > X; \theta \mid X \in (q_1, q_0)) \\ &= (1 - G(X;\theta)) I(X \in (q_1, q_0))\end{aligned}$$

We propose the following estimator for the partial AUC:

$$\hat{\delta}_{p_0 p_1} = \frac{1}{m} \sum_{i=1}^{m} \hat{V}_i(X_i; \theta),$$

where $\hat{V}_i(X_i;\theta) = (1-G(X_i;\theta))I(X_i \in (q_1, q_0)), i = 1,2,\cdots m$, $\hat{\theta}$ is the maximum likelihood estimator of θ based on $Y_j's, j = 1,2,\ldots n$, and $\hat{q}_k = \hat{F}^{-1}(1-p_k)$ is the $(1-p_k)$-th sample quantiles of $X_i's, i = 1,2,\ldots m$, and $\hat{F}$ is the empirical distribution of $X_i's$.

In this section, our goal is to find the asymptotic distribution of the estimator and construct confidence intervals for $\delta_{p_0p_1}$ based on test results $X_i's$ and $Y_i's$.

2.1 Normal approximation based confidence interval

We can show that the estimator $\hat{\delta}_{p_0p_1}$ asymptotically follows a normal distribution in the following theorem.

Theorem 2.1: Assume that $0<\lim_{m,n\to\infty}\frac{m}{n} = \rho < \infty$ is a constant. Then

$$m^{1/2}(\hat{\delta}_{p_0p_1} - \delta_{p_0p_1}) \xrightarrow{L} N(0, \sigma^2_{p_0p_1}), \tag{2.1}$$

where

$$\begin{aligned}
\sigma^2_{p_0p_1} &= Var(B(X;\theta,q_0,q_1)) + \rho D'(\theta,q_0,q_1)\Sigma_\theta D(\theta,q_0,q_1),\\
D(\theta,q_0,q_1) &= E[G_\theta(X;\theta)I(X \in (q_1,q_0))],\\
B(X;\theta,q_0,q_1) &= [(1-G(X;\theta))I(X \in (q_1,q_0)) - \delta_{p_0p_1}]\\
&\quad -\sum_{k=0}^{1}[1-G(q_k;\theta)][I(X \le q_k) - (1-p_k)],
\end{aligned}$$

and Σ_θ is the asymptotic covariance matrix of $\sqrt{n}(\hat{\theta}-\theta)$, $G_\theta(X;\theta)$ denotes the derivative of $G(X;\theta)$ with respect to $\theta \in \Theta \subseteq R^q$, $D'(\theta,q_0,q_1)$ is the transpose of $D(\theta,q_0,q_1)$.

The asymptotic normality in Theorem 2.1 cannot be directly used to produce a confidence interval for the partial AUC $\delta_{p_0 p_1}$ because the asymptotic variance $\sigma^2_{p_0 p_1}$ is a function of the unknown parameter θ and the distribution function F. However, we can empirically estimate the asymptotic variance $\sigma^2_{p_0 p_1}$ by using the maximum likelihood estimator $\hat{\theta}$ and the empirical distribution $\hat{F}$. The following $\hat{\sigma}^2_{p_0 p_1}$ is a consistent estimate for $\sigma^2_{p_0 p_1}$:

$$\hat{\sigma}^2_{p_0 p_1} = \hat{Var}(B(X;\theta,q_0,q_1)) + (m/n)\hat{D}'(\theta,q_0,q_1)\hat{\Sigma}_\theta D(\theta,q_0,q_1),$$

where

$$\hat{Var}(B(X;\theta,q_0,q_1)) = \frac{1}{m-1}\sum_{i=1}^{m}(B(X_i;\hat{\theta},\hat{q}_0,\hat{q}_1) - \frac{1}{m}\sum_{i=1}^{m}B(X_i;\hat{\theta},\hat{q}_0,\hat{q}_1))^2$$

$$B(X_i;\hat{\theta},\hat{q}_0,\hat{q}_1) = [(1-G(X_i,\hat{\theta}))I(X_i \in (\hat{q}_1,\hat{q}_0)) - \hat{\delta}_{p_0 p_1}] - \sum_{k=0}^{1}[1-G(\hat{q}_i,\hat{\theta})][I(X_i \le \hat{q}_k) - (1-p_k)]$$

$$\hat{D}(\theta,q_0,q_1) = \frac{1}{m}\sum_{i=1}^{m}G'_\theta(X_i,\hat{\theta})I(X_i \in (\hat{q}_1,\hat{q}_0)),$$

and $\hat{\Sigma}_\theta$ is a consistent estimator of Σ_θ.

Therefore, a $(1-\alpha)$-th normal approximation (NA) based confidence interval for the partial AUC $\delta_{p_0 p_1}$ can be constructed as follows:

$$(\hat{\delta}_{p_0 p_1} - z_{1-\frac{\alpha}{2}}\frac{\hat{\sigma}_{p_0 p_1}}{\sqrt{m}}, \hat{\delta}_{p_0 p_1} + z_{1-\frac{\alpha}{2}}\frac{\hat{\sigma}_{p_0 p_1}}{\sqrt{m}}), \tag{2.2}$$

where $z_{1-\frac{\alpha}{2}}$ is the $(1-\alpha/2)$-th quantile of the standard normal distribution.

2.2 Bootstrap based confidence intervals

Bootstrap is a useful approach for the construction of confidence intervals of unknown parameters. It is frequently used to estimate the variance of a statistic when the variance is unknown and of a complex form. Clearly, the asymptotic variance $\sigma^2_{p_0 p_1}$ of $\hat{\delta}_{p_0 p_1}$ is complex and can be empirically estimated by using the MLE $\hat{\theta}$ and sample quantiles. However, this empirical variance estimate may not be stable when sample size is small. To produce a better confidence interval for the partial AUC, we propose the following bootstrap procedure to estimate the asymptotic variance.

Step 1. Draw a bootstrap resample $\{X_1^*, X_2^*, \ldots, X_m^*\}$ of size m from the non-diseased sample $\{X_1, X_2, \ldots, X_m\}$ and a bootstrap resample $\{Y_1^*, Y_2^*, \ldots, Y_n^*\}$ of size n from the diseased sample $\{Y_1, Y_2, \ldots, Y_n\}$.

Step 2. Calculate a bootstrap version of $\hat{\delta}_{p_0 p_1}$:

$$\delta^*_{p_0 p_1} = \frac{1}{m}\sum_{i=1}^{m} V(X_i^*, \hat{\theta}^*)$$

where

$$V(X_i^*, \hat{\theta}^*) = [1 - G(X_i^*, \theta^*)] I(X_i^* \in (q_1^*, q_0^*)),$$

$\hat{\theta}^*$ is the bootstrap version of $\hat{\theta}$, and $\hat{q}_k^* = \hat{F}^{*-1}(1-p_k)$ is the $(1-p_k)$-th sample quantile based on the bootstrap resample $\{X_1^*, X_2^*, \ldots, X_m^*\}$.

Step 3. Repeat the steps 1-2 B (B is recommended to be greater than or equal to 150) times to get B bootstrap copies of $\hat{\delta}_{p_0 p_1}$:

$$\{\delta^*_{p_0p_1,b} : \text{b=1, 2, …, } B\}.$$

Then, the bootstrap variance estimator is defined as:

$$\sigma^{*2}_{p_0p_1} = \frac{1}{B-1}\sum_{b=1}^{B}(\delta^*_{p_0p_1,b} - \bar{\delta}^*_{p_0p_1})^2,$$

where $\bar{\delta}^*_{p_0p_1} = \frac{1}{B}\sum_{b=1}^{B}\delta^*_{p_0p_1;b}$.

This leads to two $(1-\alpha)$-th bootstrap based confidence intervals for the partial AUC defined as follows:

BI interval:

$$(\delta^{\text{空}}_{p_0p_1} - z_{1-\frac{\alpha}{2}}\frac{\sigma^*_{p_0p_1}}{\sqrt{m}}, \delta_{p_0p_1} + z_{1-\frac{\alpha}{2}}\frac{\sigma^*_{p_0p_1}}{\sqrt{m}}) \tag{2.3}$$

BII interval:

$$(\bar{\delta}^*_{p_0p_1} - z_{1-\frac{\alpha}{2}}\frac{\sigma^*_{p_0p_1}}{\sqrt{m}}, \bar{\delta}^*_{p_0p_1} + z_{1-\frac{\alpha}{2}}\frac{\sigma^*_{p_0p_1}}{\sqrt{m}}) \tag{2.4}$$

CHAPTER 3

Empirical Likelihood Based Confidence Intervals for the partial AUC

It is well known that the empirical likelihood (EL) (Owen, 1990, 2001) also is a popular non-parametric method used for providing confidence intervals of unknown parameter. Qin and Zhou (2006) proposed an empirical likelihood method for the statistical inference of the full AUC. In this chapter, we develop an EL-based semi-parametric inference for the partial AUC.

By assuming that the parameter θ and quantiles q_1 and q_0 are known, Dodd and Pepe (2003) defined the restricted placement value of X as follows:

$$V(X;\theta) = (1 - G(X;\theta))I(X \in (q_1, q_0)).$$

They interpreted $V(X;\theta)$ as the restricted placement value of a given non-diseased test value, X, in the survival function of test results of diseased subjects. As shown in Chapter 2,

$$E[V(X;\theta)] = P\{Y > X, X \in (q_1, q_0); \theta\} = \delta_{p_0 p_1},$$

which implies that the expectation of $V(X;\theta)$ is equal to the partial AUC, $\delta_{p_0 p_1}$.

By using this relationship between the restricted placement value and the partial AUC, an empirical likelihood for $\delta_{p_0 p_1}$ can be defined by the following expression:

$$\tilde{L}(\delta_{p_0 p_1}) = \sup_{\mathbf{p}}\{\prod_{i=1}^{m} p_i : \sum_{i=1}^{m} p_i (V(X_i, \theta) - \delta_{p_0 p_1}) = 0\}, \tag{3.1}$$

where $\mathbf{p}=(p_1, p_2, \cdots, p_m)$ is a probability vector , i.e. $\sum_{i=1}^{m} p_i = 1$ and $p_i \geq 0$ for all i.

Obviously, $\tilde{L}(\delta_{p_0 p_1})$ cannot be calculated since it depends on the unknown parameter θ

in the diseased distribution function G and quantiles q_i's of the unknown non-diseased distribution function F. A natural solution is to replace θ by its MLE $\hat{\theta}$ and q_i's by their sample quantiles $\hat{q}_i$'s. Hence, an adjusted empirical likelihood for $\delta_{p_0 p_1}$ can be defined as follows:

$$L(\delta_{p_0 p_1}) = \sup_{\mathbf{p}}\{\prod_{i=1}^{m} p_i : \sum_{i=1}^{m} p_i(\hat{V}(X_i,\hat{\theta}) - \delta_{p_0 p_1}) = 0\},$$

where $\hat{V}(X_i,\theta) = [1 - G(X_i,\theta)]I(X_i \in (q_1, q_0)), i = 1,2,\ldots m.$

Then the corresponding empirical log-likelihood ratio (ELR) for $\delta_{p_0 p_1}$ is

$$l(\delta_{p_0 p_1}) = 2\sum_{i=1}^{m} \log\left[1 + \lambda(\hat{V}(X_i,\hat{\theta}) - \delta_{p_0 p_1})\right], \tag{3.2}$$

where λ is the solution to

$$\frac{1}{m}\sum_{i=1}^{m} \frac{\hat{V}(X_i,\hat{\theta}) - \delta_{p_0 p_1}}{1 + \lambda(\hat{V}(X_i,\hat{\theta}) - \delta_{p_0 p_1})} = 0 \ . \tag{3.3}$$

We can prove that $l(\delta_{p_0 p_1})$ follow a scaled χ^2 distribution as shown by the following theorem:

Theorem 3.1: If $\delta_{p_0 p_1}$ is the true value of the partial AUC, and $\lim_{m,n\to\infty} \frac{m}{n} = \rho$ is a constant, then the limiting distribution of $l(\delta_{p_0 p_1})$ is a scaled chi-square distribution with one degree of freedom. i.e.,

$$C(\delta_{p_0 p_1})\, l(\delta_{p_0 p_1}) \xrightarrow{L} \chi_1^2,$$

where the scale constant $C(\delta_{p_0 p_1}) = \frac{S^2_{p_0 p_1}}{\sigma^2_{p_0 p_1}}$ and $S^2_{p_0 p_1} = Var(V(X;\theta))$.

In order to construct empirical likelihood confidence intervals for the partial AUC, we need to estimate $S^2_{p_0p_1}$ and $\sigma^2_{p_0p_1}$ since the scale constant $C(\sigma_{p_0p_1})$ is unknown. Here, we will use the bootstrap variance estimate $\sigma^{*2}_{p_0p_1}$ defined in section (2.2). Then, Using Theorem 3.1, four hybrid bootstrap and empirical likelihood (HBEL) based confidence intervals for the partial AUC can be constructed as follows:

HBELI interval:

$$\left\{\delta : \hat{C}(\hat{\delta}_{p_0p_1})l(\delta) \le \chi^2_1(1-\alpha)\right\}, \tag{3.4}$$

where $\hat{C}(\hat{\delta}_{p_0p_1})$ is an estimator for $C(\delta_{p_0p_1})$:

$$\hat{C}(\hat{\delta}_{p_0p_1}) = \frac{\hat{S}^2_{p_0p_1}}{m\sigma^{*2}_{p_0p_1}}$$

$$\hat{S}^2_{p_0p_1} = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{V}(X_i;\theta) - \hat{\delta}_{p_0p_1})^2$$

HBELII interval:

$$\left\{\delta : \hat{C}^*(\hat{\delta}_{p_0p_1})l(\delta) \le \chi^2_1(1-\alpha)\right\}, \tag{3.5}$$

where $\hat{C}^*(\hat{\delta}_{p_0p_1})$ is the following estimator for $C(\delta_{p_0p_1})$:

$$\hat{C}^*(\hat{\delta}_{p_0p_1}) = \frac{\bar{\hat{S}}^{*2}_{p_0p_1}}{m\sigma^{*2}_{p_0p_1}},$$

with $\bar{\hat{S}}^{*2}_{p_0p_1}$ being the mean of B bootstrap copies of $\hat{S}^2_{p_0p_1}$.

HBELIII interval:

$$\left\{\delta : l(\delta) \le C^*\right\}, \tag{3.6}$$

where C^* is the $(1-\alpha)$-th quantile of $\{ l_1^*(\hat{\delta}_{p_0p_1}),\ l_2^*(\hat{\delta}_{p_0p_1}),\ldots,l_B^*(\hat{\delta}_{p_0p_1}) \}$ which are the B bootstrap copies of $l^*(\hat{\delta}_{p_0p_1})$ with

$$l^*(\delta_{p_0p_1}) = 2\sum_{i=1}^{m} \log[1+\lambda^*(V(X_i^*;\theta^*)-\delta_{p_0p_1})],$$

and λ^* being the solution to

$$\frac{1}{m}\sum_{i=1}^{m}\frac{V(X_i^*,\theta^*)-\delta_{p_0p_1}}{1+\lambda^*(V(X_i^*,\theta^*)-\delta_{p_0p_1})}=0\,.$$

HBELIV interval:

$$\left\{\delta : \tilde{C}^*(\hat{\delta}_{p_0p_1})l(\delta) \le \chi_1^2(1-\alpha)\right\}, \tag{3.7}$$

where $\tilde{C}^*(\hat{\delta}_{p_0p_1})$ is an estimator for $C(\delta_{p_0p_1})$:

$$\tilde{C}^*(\hat{\delta}_{p_0p_1}) = [\frac{1}{B}\sum_{b=1}^{B} l_b^*(\hat{\delta}_{p_0p_1})]^{-1}.$$

To obtain the hybrid bootstrap and empirical likelihood based confidence intervals for the partial AUC, for example, the **HBELI** interval, we need to solve the following nonlinear equations for λ and δ :

$$\begin{cases} \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\dfrac{\hat{V}(X_i;\hat{\theta})-\delta}{1+\lambda(\hat{V}(X_i;\hat{\theta})-\delta)}=0 \\ \tilde{C}(\delta_{p_0p_1})l(\delta)=\chi_1^2(1-\alpha) \end{cases}$$

There will be two solutions for δ , the smaller one and bigger one are the lower bound and the upper bound of the **HBELI** interval respectively. Use the similar procedure as above, we can find the **HBELII, HBELIII** and **HBELIV** intervals.

CHAPTER 4

SIMULATION STUDY

In this Chapter, we conduct two simulation studies to evaluate the small sample performances of the seven confidence intervals (NA, BI, BII, HBELI, HBELII, HBELIII, and HBELIV) presented in Chapter 2 and Chapter 3. Since we know the underlying parametric distributions in the simulation study, the maximum likelihood (ML) based confidence interval for the partial AUC is also included in the study as a comparison base.

In the simulation studies, binormal distributions and exponential distributions are chosen to be the underlying distributions respectively. For each study, 1000 random samples of size m from the non-diseased population F and of size n from the diseased population G have been generated respectively, where the sample sizes (m,n) = (30, 30), (50, 50), (100, 100), (50, 30), and (80, 50). The subinterval (p_0, p_1) of (0,1) for the partial AUC $\delta_{p_0 p_1}$ is chosen to be (0, 0.4), (0, 0.7), and (0.05, 0.5), respectively. For the bootstrap method, we take B=150 bootstrap re-samples from the original samples. Under these simulation settings, various confidence intervals with confidence levels at both 95% and 90% are computed for the partial AUC.

4.1 Normal distributions estimators

In the first study, the test results of non-diseased patients are generated from the standard normal distribution $N(0,1)$, and the test results of diseased patients are generated

from the normal distribution $N(1,1)$. The simulation results are shown in Table I and Table II. From these tables, we observe that the NA intervals have the longest interval lengths and always over-cover the true partial AUC. Other six confidence intervals (BI, BII, HBELI, HBELII, HBELIII and HBELIV) based on bootstrap and empirical likelihood methods have similar coverage probabilities, but the HBELII and HBELIII intervals perform slightly better in most cases considered here.

4.2 Exponential distributions estimators

The following exponential models are considered in the second simulation study: a standard exponential distribution $Exp(\lambda)$ with rate $\lambda = 1$ for the non-diseased population and an exponential distribution $Exp(\gamma)$ with rate $\gamma = 0.43$ for the diseased population. Simulation results are presented in Table III and Table IV. From these two tables, we again observe that the NA intervals are the most conservative intervals for the partial AUC. Other six confidence intervals (BI, BII, HBELI, HBELII, HBELIII and HBELIV) based on bootstrap and empirical likelihood methods have similar coverage probabilities.

In summary, our simulation studies suggest that the bootstrap and EL-based methods perform better than the normal approximation (NA) based method due to the better coverage probability and shorter interval length. The parametric ML interval has good coverage accuracy, but it can only be used when the underlying parametric assumption is true.

CHAPTER 5

DUCHENNE MUSCULAR DYSTROPHY EXAMPLE

Duchenne muscular dystrophy (DMD) is one of nine types of muscular dystrophy and is characterized by rapidly-worsening muscle weakness that starts in the legs and pelvis, and later affects the whole body. It is inherited in what is known as an X-linked recessive pattern. In other word, DMD affects children who inherit the disease through their mothers. Unfortunately, women can be carriers of DMD but usually exhibit no symptoms. As a result, early screening of females who could be potential carriers is essential because there is no cure for DMD.

Serum enzyme tests can be important aids in the diagnosis of DMD since carriers lead to a significant increase in the muscle protein levels found in the blood. In this study, we shall consider the serum pyruvate kinase (PK) level measured in known carriers and in healthy females from serum enzyme test and want to know how accurate the test can be in detecting DMD.

The study data set, which is collected during a program run at the Hospital for Sick Children of Toronto, is given in Andrews and Herzberg (1985). Totally, 127 healthy females (non-carriers) and 67 carriers are included in the study. As Adimari and Chiogna (2005) discussed in their paper, transformations are needed for the original data set to assume normal distributions for the transformed values since there are variations in normality for both models (carriers and healthy females). The transformation with the power of -0.56 is suggested by the Box-Cox method for the PK levels of carriers. In other words, we assume normal distributions for $\widetilde{Y} = Y^{-0.56}$. The same transformation is applied

to the healthy females PK levels, i.e. $\widetilde{X} = X^{-0.56}$. The normal quantile-quantile plots for $\widetilde{Y}_i$'s and $\widetilde{X}_i$'s are shown in Figure 1.

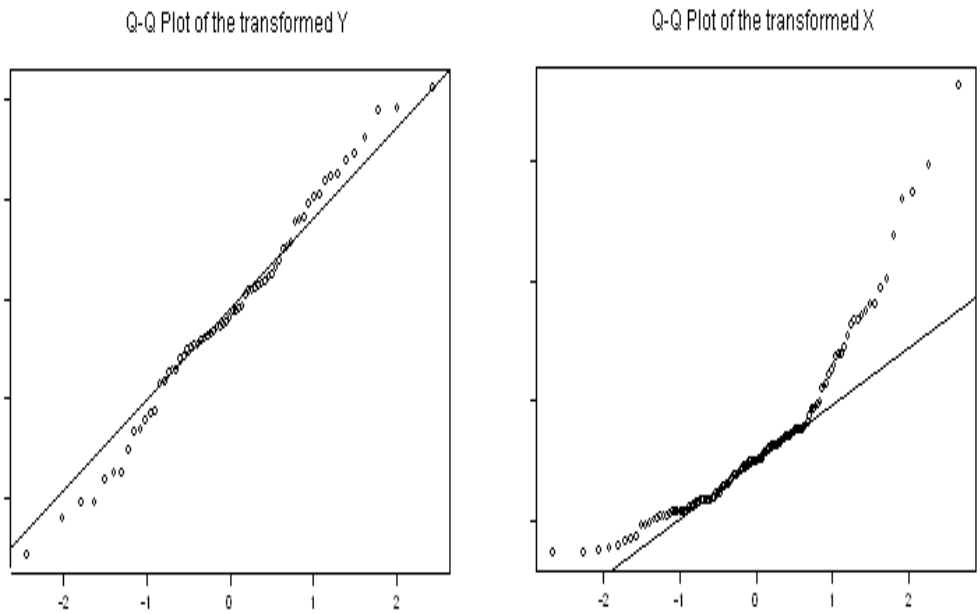

Figure 1: Normal quantile-quantile plots of the transformed PK levels.

This Q-Q plot suggests that the parametric method (ML) can not be used for the transformed data. However, we can use the newly proposed semi-parametric method to construct confidence intervals for

$$\delta_{p_0 p_1} / (p_1 - p_0) = P\{Y > X, X \in (q_1, q_0); \theta\} / (p_1 - p_0),$$

the normalized partial AUC of the test, where X and Y denote serum PK levels for a healthy and a carrier female, respectively.

Table V and Table VI show various confidence intervals for the normalized partial AUC's and full AUC. All the intervals indicate that the serum enzyme test has low to moderate accuracy in detecting DMD.

CHAPTER 6

CONCLUSION AND DISCUSSION

Diagnostic tests are central in the field of modern medicine. Accurate diagnostic tests provide critical information about the disease states of patients. The partial AUC is one of the most important summary measures for the accuracy of a diagnostic test. Finding confidence regions of the partial AUC over an interested FPR interval is an appropriate way to evaluate the diagnostic accuracy of a test. In diagnostic studies, semi-parametric approaches often are useful alternative to the parametric and nonparametric counterparts since semi-parametric methods may inherit good properties of both parametric and nonparametric methods. In this thesis, we have developed new semi-parametric methods for the inference on the partial AUC. In particular, seven confidence intervals are proposed for the partial AUC by using the normal approximation, bootstrap and empirical likelihood methods. The simulation studies indicate that the bootstrap and EL-based methods perform better than the normal approximation (NA) based method. Therefore, the use of hybrid bootstrap and EL-based confidence intervals for the partial AUC is recommended in evaluating the accuracy of diagnostic tests when the test results follow a semi-parametric model.

REFERENCES

Andrews D.F., Herzberg, A.M. (1985). Data. *Springer-Verlag, New York.*

Dodd L and Pepe MS (2003). Partial AUC Estimation and Regression. *Biometrics.* **59**: *614-623.*

Egan JP (1975) Signal detection theory and ROC analysis. *New York: Academic Press*.

England WL. (1988) An exponential model used for optimal threshold selection on ROC curves. *Medical Decision Making* **8**:120-131.

Metz CE, Herman BA, Shen J-H. (1998) Maximum likelihood estimation of receiver operating characteristic (ROC) curves from continuously-distributed data. *Statistics in Medicine* **17**: 1033-53

Adimari G, Chiogna M (2005) Partially parametric interval estimation of Pr{Y>X}. *Computational Statistics & Data Analysis* **51**: 1875-1891

Goddard MJ, Hinberg I. (1990) Receiver operating characteristic (ROC) curves and non-normal data: an empirical study. *Statistics in Medicine* **9**:325-37.

Hanley JA, McNeil BJ (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* **148**:839-43.

Hilden J (1991). The area under the ROC curve and its competitors. *Medical Decision Making* **11**: 95-101.

Jiang Y, Metz CE and Nishikawa RM (1996). A receiver operating characteristic partial area index for highly sensitive diagnostic tests. *Radiology* **143**:29-36.

Le CT. (1997) Evaluation of confounding effects in ROC studies. *Biometrics* **53**: 998-1007.

Lilienfeld AM (1974). Some limitations and problems of screening for cancer. *Cancer* **36**:1720-1724

Lloyd CJ. (1998) Using smoothed receiver operating characteristic curves to summarize and compare diagnostic systems. *Journal of the American Statistical Association* **93**:1356-64.

McClish DK (1989). Analyzing a portion of the ROC curve. *Medical Decision Making* **9**: 190–195.

Molodianovitch K, Faraggi D, Reiser B (2006). Comparing the areas under two correlated ROC curves: parametric and non-parametric approaches. *Biometrical Journal* **48**:745-57

Owen, A. (1990). Empirical likelihood ratio confidence regions. *Annals of Statistics* **18**: 90-120.

Owen, A. (2001). Empirical likelihood. Chapman & Hall/CRC, New York.

Qin, GS and Zhou XH (2006). Empirical Likelihood Inference for the Area Under the ROC Curve. *Biometrics* **62**: 613-622.

Qin, G.S., Hsu, Y.-S., and Zhou, X.H. (2006). New confidence intervals for the difference between two sensitivities at a fixed level of specificity. *Statistics in Medicine* **25**: 3487-3502.

Qin, G.S., Jin, X.P and Zhou, XH (2007). Bootstrap and Empirical Likelihood-based inference for the Partial Area Under the ROC Curve. *Manuscript.*

Shapiro DE (1999). The Interpretation of Diagnostic Tests. *Statistical Methods in Medical Research* **8**:113-134

Strike PW. (1995) Measurement in laboratory medicine: a primer on control and interpretation. *Oxford: Butterworth-Heinemann*.

Thompson ML, and Zucchini W (1989). On the statistical analysis of ROC curves. *Statistics in Medicine* **8**: 1277-1290.

Wieand S (1989). A Family of Non-Parametric Statistic for Comparing Diagnostic Markers with Paired or Unpaired Data. *Biometrika* **76**:585-592

Walsh SJ (1997). Limitations to the robustness of binormal ROC curves: effects of model misspecification and location of the decision thresholds on bias, precision, size and power. *Statistics in Medicine* **16**: 669-679

Zhang DD, Zhou XH, Freeman DH and Freeman JL(2002). A non-parametric method for the comparison of partial areas under ROC curves and its application to large health care data sets. *Statistics in Medicine* **21**(5):701-15

Zou KH, Hall W, Shapiro DE. (1997) Smooth non-parametric receiver operating characteristic (ROC) curves for continuous diagnostic tests. *Statistics in Medicine* **16**: 2143-56.

APPENDIX A: SIMULATION TABLES

Table I: Normal distribution: Level of 95 percent confidence interval for the pAUC.

Sample size (m,n)	Method	(p_0, p_1)=(0, 0.4)		(p_0, p_1)=(0, 0.7)		(p_0, p_1)=(0.05, 0.5)	
		Coverage probability	Length	Coverage probability	Length	Coverage probability	Length
(30, 30)	NA	0.984	0.231	0.983	0.301	0.982	0.266
	BI	0.965	0.181	0.960	0.243	0.967	0.195
	BII	0.954	0.181	0.937	0.243	0.943	0.195
	HBELI	0.959	0.178	0.966	0.239	0.968	0.192
	HBELII	0.947	0.167	0.936	0.230	0.964	0.179
	HBELIII	0.957	0.180	0.954	0.239	0.972	0.194
	HBELIV	0.963	0.187	0.961	0.245	0.979	0.200
	ML	0.927	0.153	0.923	0.215	0.921	0.165
(50, 50)	NA	0.984	0.177	0.990	0.233	0.986	0.204
	BI	0.954	0.134	0.952	0.182	0.959	0.144
	BII	0.931	0.134	0.935	0.182	0.932	0.144
	HBELI	0.957	0.133	0.954	0.180	0.961	0.142
	HBELII	0.960	0.127	0.953	0.176	0.963	0.137
	HBELIII	0.958	0.132	0.953	0.178	0.964	0.141
	HBELIV	0.960	0.136	0.961	0.182	0.969	0.146
	ML	0.940	0.119	0.944	0.167	0.944	0.128
(100, 100)	NA	0.993	0.125	0.995	0.164	0.992	0.144
	BI	0.961	0.091	0.954	0.124	0.953	0.099
	BII	0.949	0.091	0.948	0.124	0.940	0.099
	HBELI	0.958	0.091	0.954	0.124	0.957	0.099
	HBELII	0.961	0.088	0.959	0.122	0.959	0.097
	HBELIII	0.951	0.090	0.954	0.123	0.951	0.098
	HBELIV	0.957	0.092	0.958	0.125	0.960	0.101
	ML	0.949	0.085	0.940	0.119	0.954	0.092
(50, 30)	NA	0.980	0.187	0.980	0.251	0.987	0.217
	BI	0.950	0.150	0.945	0.204	0.957	0.162
	BII	0.927	0.150	0.939	0.204	0.936	0.162
	HBELI	0.946	0.148	0.949	0.201	0.959	0.160
	HBELII	0.933	0.140	0.934	0.197	0.951	0.154
	HBELIII	0.948	0.150	0.934	0.198	0.953	0.159
	HBELIV	0.956	0.151	0.943	0.204	0.956	0.163
	ML	0.940	0.134	0.942	0.192	0.937	0.147
(80, 50)	NA	0.984	0.147	0.990	0.197	0.991	0.170
	BI	0.951	0.114	0.949	0.157	0.951	0.125
	BII	0.943	0.114	0.942	0.157	0.938	0.125
	HBELI	0.951	0.113	0.952	0.156	0.950	0.124
	HBELII	0.944	0.110	0.942	0.153	0.947	0.121
	HBELIII	0.944	0.112	0.941	0.153	0.953	0.123
	HBELIV	0.956	0.115	0.948	0.158	0.959	0.126
	ML	0.941	0.106	0.951	0.151	0.946	0.115

Table II: Normal distribution: Level of 90 percent confidence interval for the pAUC.

Sample size (m,n)	Method	(p_0, p_1)=(0, 0.4)		(p_0, p_1)=(0, 0.7)		(p_0, p_1)=(0.05, 0.5)	
		Coverage probability	Length	Coverage probability	Length	Coverage probability	Length
(30, 30)	NA	0.970	0.194	0.966	0.253	0.976	0.223
	BI	0.915	0.154	0.910	0.204	0.941	0.162
	BII	0.886	0.154	0.868	0.204	0.907	0.162
	HBELI	0.920	0.152	0.913	0.201	0.940	0.160
	HBELII	0.893	0.140	0.891	0.195	0.925	0.152
	HBELIII	0.924	0.150	0.909	0.200	0.943	0.163
	HBELIV	0.939	0.159	0.923	0.206	0.949	0.170
	ML	0.882	0.128	0.876	0.180	0.876	0.138
(50, 50)	NA	0.963	0.148	0.974	0.195	0.971	0.171
	BI	0.913	0.113	0.923	0.152	0.916	0.121
	BII	0.895	0.113	0.896	0.152	0.888	0.121
	HBELI	0.915	0.112	0.924	0.151	0.914	0.120
	HBELII	0.900	0.106	0.907	0.148	0.918	0.115
	HBELIII	0.920	0.112	0.919	0.152	0.921	0.120
	HBELIV	0.930	0.114	0.925	0.155	0.932	0.123
	ML	0.888	0.100	0.885	0.140	0.892	0.108
(100, 100)	NA	0.975	0.104	0.975	0.138	0.984	0.121
	BI	0.912	0.076	0.891	0.105	0.931	0.083
	BII	0.903	0.076	0.903	0.105	0.905	0.083
	HBELI	0.912	0.076	0.893	0.105	0.935	0.083
	HBELII	0.905	0.075	0.897	0.104	0.884	0.081
	HBELIII	0.901	0.076	0.910	0.104	0.912	0.083
	HBELIV	0.909	0.077	0.920	0.105	0.920	0.084
	ML	0.897	0.071	0.887	0.100	0.910	0.077
(50, 30)	NA	0.961	0.157	0.953	0.211	0.973	0.182
	BI	0.912	0.126	0.898	0.172	0.905	0.135
	BII	0.893	0.126	0.881	0.172	0.882	0.135
	HBELI	0.915	0.125	0.902	0.170	0.909	0.134
	HBELII	0.894	0.118	0.902	0.165	0.904	0.130
	HBELIII	0.909	0.122	0.893	0.167	0.926	0.134
	HBELIV	0.919	0.127	0.897	0.171	0.933	0.138
	ML	0.885	0.113	0.892	0.161	0.891	0.123
(80, 50)	NA	0.964	0.124	0.959	0.165	0.976	0.142
	BI	0.926	0.096	0.919	0.133	0.904	0.105
	BII	0.912	0.096	0.912	0.133	0.883	0.105
	HBELI	0.928	0.096	0.919	0.132	0.904	0.104
	HBELII	0.886	0.092	0.902	0.129	0.887	0.101
	HBELIII	0.897	0.094	0.890	0.130	0.909	0.103
	HBELIV	0.895	0.096	0.898	0.133	0.920	0.106
	ML	0.888	0.089	0.893	0.126	0.880	0.097

Table III: Exponential distribution: Level of 95 percent confidence interval for the pAUC.

Sample size (m,n)	Method	(p_0, p_1)=(0, 0.4)		(p_0, p_1)=(0, 0.7)		(p_0, p_1)=(0.05, 0.5)	
		Coverage probability	Length	Coverage probability	Length	Coverage probability	Length
(30, 30)	NA	0.981	0.204	0.979	0.267	0.988	0.232
	BI	0.966	0.152	0.958	0.218	0.966	0.164
	BII	0.958	0.152	0.951	0.218	0.950	0.164
	HBELI	0.968	0.150	0.960	0.214	0.964	0.161
	HBELII	0.944	0.142	0.961	0.211	0.969	0.156
	HBELIII	0.967	0.152	0.965	0.218	0.979	0.168
	HBELIV	0.974	0.159	0.970	0.223	0.983	0.174
	ML	0.922	0.130	0.942	0.190	0.936	0.139
(50, 50)	NA	0.991	0.158	0.985	0.206	0.980	0.176
	BI	0.950	0.112	0.958	0.161	0.966	0.121
	BII	0.939	0.112	0.942	0.161	0.948	0.121
	HBELI	0.948	0.111	0.961	0.159	0.965	0.120
	HBELII	0.951	0.108	0.950	0.159	0.959	0.118
	HBELIII	0.956	0.111	0.949	0.161	0.972	0.122
	HBELIV	0.964	0.113	0.958	0.164	0.978	0.125
	ML	0.933	0.101	0.941	0.148	0.939	0.109
(100, 100)	NA	0.996	0.111	0.966	0.138	0.998	0.147
	BI	0.957	0.077	0.940	0.111	0.946	0.084
	BII	0.949	0.077	0.937	0.111	0.935	0.084
	HBELI	0.957	0.077	0.939	0.111	0.943	0.083
	HBELII	0.946	0.075	0.950	0.110	0.935	0.083
	HBELIII	0.951	0.076	0.948	0.109	0.959	0.083
	HBELIV	0.964	0.078	0.952	0.111	0.964	0.085
	ML	0.951	0.072	0.949	0.105	0.951	0.077
(50, 30)	NA	0.980	0.168	0.987	0.251	0.984	0.189
	BI	0.950	0.125	0.956	0.182	0.952	0.136
	BII	0.943	0.125	0.949	0.182	0.948	0.136
	HBELI	0.952	0.124	0.956	0.179	0.962	0.134
	HBELII	0.941	0.120	0.947	0.179	0.958	0.132
	HBELIII	0.956	0.126	0.949	0.181	0.962	0.138
	HBELIV	0.966	0.130	0.952	0.186	0.974	0.142
	ML	0.937	0.117	0.936	0.172	0.935	0.126
(80, 50)	NA	0.992	0.131	0.988	0.175	0.986	0.149
	BI	0.954	0.097	0.954	0.140	0.948	0.106
	BII	0.946	0.097	0.948	0.140	0.938	0.105
	HBELI	0.952	0.096	0.954	0.139	0.948	0.105
	HBELII	0.955	0.094	0.957	0.139	0.942	0.104
	HBELIII	0.950	0.097	0.946	0.139	0.955	0.105
	HBELIV	0.963	0.099	0.951	0.142	0.962	0.108
	ML	0.933	0.092	0.939	0.134	0.936	0.098

Table IV: Exponential distribution: Level of 90 percent confidence interval for the pAUC.

Sample size (m,n)	Method	(p_0, p_1)=(0, 0.4)		(p_0, p_1)=(0, 0.7)		(p_0, p_1)=(0.05, 0.5)	
		Coverage probability	Length	Coverage probability	Length	Coverage probability	Length
(30, 30)	NA	0.974	0.172	0.980	0.225	0.975	0.195
	BI	0.918	0.127	0.928	0.182	0.936	0.137
	BII	0.892	0.127	0.921	0.182	0.902	0.137
	HBELI	0.918	0.125	0.930	0.180	0.942	0.136
	HBELII	0.901	0.119	0.926	0.179	0.926	0.131
	HBELIII	0.923	0.126	0.923	0.180	0.947	0.139
	HBELIV	0.936	0.134	0.934	0.189	0.952	0.147
	ML	0.877	0.109	0.887	0.160	0.881	0.117
(50, 50)	NA	0.977	0.132	0.965	0.173	0.979	0.150
	BI	0.910	0.094	0.911	0.135	0.929	0.102
	BII	0.898	0.094	0.898	0.135	0.911	0.102
	HBELI	0.912	0.094	0.913	0.135	0.932	0.102
	HBELII	0.889	0.091	0.909	0.135	0.923	0.099
	HBELIII	0.907	0.093	0.906	0.134	0.926	0.102
	HBELIV	0.914	0.095	0.916	0.138	0.937	0.105
	ML	0.877	0.085	0.897	0.124	0.897	0.091
(100, 100)	NA	0.977	0.093	0.983	0.122	0.984	0.106
	BI	0.909	0.064	0.907	0.093	0.899	0.070
	BII	0.901	0.064	0.904	0.093	0.887	0.070
	HBELI	0.912	0.064	0.907	0.093	0.900	0.070
	HBELII	0.901	0.064	0.900	0.092	0.906	0.070
	HBELIII	0.914	0.064	0.904	0.092	0.918	0.070
	HBELIV	0.922	0.065	0.909	0.094	0.922	0.072
	ML	0.887	0.061	0.893	0.088	0.903	0.065
(50, 30)	NA	0.972	0.141	0.958	0.187	0.981	0.189
	BI	0.917	0.105	0.893	0.154	0.905	0.114
	BII	0.910	0.105	0.886	0.154	0.891	0.114
	HBELI	0.920	0.104	0.898	0.152	0.911	0.113
	HBELII	0.895	0.101	0.897	0.151	0.916	0.111
	HBELIII	0.913	0.104	0.896	0.150	0.916	0.114
	HBELIV	0.924	0.110	0.914	0.157	0.933	0.120
	ML	0.886	0.098	0.892	0.144	0.889	0.106
(80, 50)	NA	0.972	0.110	0.957	0.147	0.982	0.148
	BI	0.902	0.081	0.906	0.118	0.910	0.089
	BII	0.896	0.081	0.900	0.118	0.900	0.089
	HBELI	0.906	0.081	0.907	0.117	0.911	0.089
	HBELII	0.891	0.079	0.912	0.116	0.920	0.087
	HBELIII	0.900	0.080	0.899	0.116	0.922	0.088
	HBELIV	0.909	0.083	0.908	0.120	0.930	0.091
	ML	0.879	0.077	0.878	0.113	0.877	0.083

APPENDIX B: REAL DATA ANALYSIS TABLES

Table V: Duchenne Muscular Dystrophy Example:
Level of 95 percent confidence intervals for the normalized pAUC

Method	(p_0, p_1)= (0, 0.4)	(p_0, p_1)= (0, 0.7)	(p_0, p_1)= (0.05, 0.5)	(p_0, p_1)= (0, 1.0)
NA	(0.520,0.770)	(0.663,0.836)	(0.613,0.824)	(0.749,0.875)
BI	(0.543,0.748)	(0.673,0.824)	(0.620,0.816)	(0.747,0.876)
BII	(0.545,0.750)	(0.673,0.824)	(0.609,0.804)	(0.746,0.875)
HBELI	(0.535,0.740)	(0.667,0.819)	(0.616,0.809)	(0.740,0.870)
HBELII	(0.535,0.740)	(0.669,0.819)	(0.620,0.804)	(0.741,0.869)
HBELIII	(0.538,0.738)	(0.661,0.824)	(0.611,0.811)	(0.740,0.870)
HBELIV	(0.533,0.743)	(0.660,0.824)	(0.613,0.809)	(0.737,0.872)
Estimate of pAUC	0.6442331	0.7490747	0.7180316	0.8116641

Table VI: Duchenne Muscular Dystrophy Example:
Level of 90 percent confidence intervals for the normalized pAUC

Method	(p_0, p_1)= (0, 0.4)	(p_0, p_1)= (0, 0.7)	(p_0, p_1)= (0.05, 0.5)	(p_0, p_1)= (0, 1.0)
NA	(0.540,0.750)	(0.676,0.821)	(0.629,0.807)	(0.759,0.864)
BI	(0.555,0.733)	(0.680,0.819)	(0.636,0.800)	(0.760,0.864)
BII	(0.558,0.738)	(0.680,0.819)	(0.640,0.804)	(0.759,0.863)
HBELI	(0.550,0.730)	(0.674,0.814)	(0.631,0.796)	(0.755,0.860)
HBELII	(0.550,0.728)	(0.674,0.814)	(0.633,0.796)	(0.756,0.859)
HBELIII	(0.560,0.720)	(0.687,0.804)	(0.638,0.791)	(0.755,0.860)
HBELIV	(0.553,0.728)	(0.686,0.804)	(0.633,0.796)	(0.756,0.859)
Estimate of pAUC	0.6442331	0.7490747	0.7180316	0.8116641

APPENDIX C: THE S-PLUS CODE FOR SIMULATION STUDIES

```
#####################################################################
#               s-splus code for normal distribution:               #
#               NA, BI, BII, HBELI, HBELII, HBELIII,HBELIV          #
#####################################################################

# solve non-linear equations(for solving lambda)
solveNonlinear<-function(f,y0,x,...)
{
   g<-function(x,y0,f)sum((f(x)-y0)^2)
   g$y0<-y0
   g$f<-f
   nlmin(g,x,max.fcal=1000,max.iter=1000,...)
}

iter<-1000        # number of iteration
m<-30            # sample size of non-diseased sample: x1,...xm
n<-30            # sample size of diseased sample: y1,...yn
rho<-m/n

mux<-0           # mean of the non-diseased population
stddx<-1         # standard deviation of the non-diseased population

muy<-1           # mean of the diseased population
stddy<-1         # standard deviation of the diseased population

p0<-0.05
p1<-0.5

alpha<-0.1
z<-qnorm(1-alpha/2)

# Function R(p)

rp<-function(p,muy,stddy)
{
   y<-1-pnorm(qnorm(1-p),muy,stddy)
   y
}

# true pauc

 truepauc<-integrate(rp, muy=muy,stddy=stddy,lower=p0,upper=p1)$integral

# Function R(p)=1-pnorm((mu0-mu1+sigma0*qnorm(1-p))/sigma1) for ML estimate

rpml <- function(p, mu0, sigma0, mu1, sigma1)
{
   1-pnorm((mu0-mu1+sigma0*qnorm(1-p))/sigma1)
}

# function used to calculate the variance of ML estimate of pAUC
# when the underlying distribuotion is  the normal distribytion.


pvar1 <- function(p, mu0, sigma0, mu1, sigma1)
{
   dnorm((mu0-mu1+sigma0*qnorm(1-p))/sigma1)
}
```

```
pvar2<- function(p, mu0, sigma0, mu1, sigma1)
{
   (qnorm(1-p))*dnorm((mu0-mu1+sigma0*qnorm(1-p))/sigma1)
}



normalcoverage<-0
normallength<-0
bootcoverage1<-0
bootcoverage2<-0
bootlength1<-0
bootlength2<-0
elcoverage1<-0
elcoverage2<-0
ellength1<-0
ellength2<-0
elcoverage3<-0
elcoverage4<-0
ellength3<-0
ellength4<-0
mlcoverage<-0
mllength<-0

#bootstrap function

mybootstrap<-function(B,x,y,m,n,p0,p1){
starpauc<-0
starssquare<-0
starvm<-matrix(0,ncol=B,nrow=m)
   for (b in 1:B)
   {
      xb<-sample(x,m,replace=T)
      yb<-sample(y,n,replace=T)
      hatq0b<-0
      hatq1b<-0
      hatq0b<-quantile(xb, c(1-p0))      # hatq0b, hatq1b: sample quantiles
      hatq1b<-quantile(xb, c(1-p1))      # of F in bootstrap


      starmlemean<-mean(yb)              # MLE estimate of startheta
       starmlestdd<-(mean((yb-starmlemean)^2))^(1/2)

      starv<-0                           # compute starv
      for (i in (1:m))
      {
         starv[i]<- (1-pnorm(xb[i],starmlemean,starmlestdd))*(hatq1b
   <=xb[i])*(xb[i] <= hatq0b)
         starvm[i,b]<- (1-pnorm(xb[i],starmlemean,starmlestdd))*(hatq1b
   <=xb[i])*(xb[i] <= hatq0b)
      }
      starpauc[b]<-mean(starv)
      starssquare[b]<-var(starv)
   }
   list(starpauc,starssquare,starvm)
}
# Loop

for (loop in c(1:iter))
{
   x<-rnorm(m,mux,stddx)     # obs from the non-diseased population
   y<-rnorm(n,muy,stddy)     # obs from the diseased population
```

```
hatq0<-quantile(x, c(1-p0))   # hatq0, hatq1: sample quantiles fo F
hatq1<-quantile(x, c(1-p1))

mlemean<-mean(y)              # MLE estimate of theta
mlestdd<-(mean((y-mlemean)^2))^(1/2)

hatv<-rep(0,m)                # hat v_i=(1-hat G(x_i, hat theta))*
                              # I(x_i \in (hatq1,hatq0))
for (i in 1:m)
{
 hatv[i]<-(1-pnorm(x[i],mlemean,mlestdd))*(hatq1 <=x[i])*(x[i] <= hatq0)
}
hatpauc<- mean(hatv)          # estimate of pauc

#1111111111111111111111111111111111111111111111111111111111111
# compute the normal approxamation based C.I.

s<-matrix(0,ncol=2,nrow=2)
d<-matrix(0,ncol=1,nrow=2)

s[1,1]<-mlestdd^2
s[1,2]<-0
s[2,1]<-0
s[2,2]<-2*mlestdd^4

# compute B(x,theta,q0,q1)
b<-rep(0,m)
for (i in 1:m)
{
 b[i]<-((1-pnorm(x[i],mlemean,mlestdd))*(hatq1 <=x[i])*(x[i] <= hatq0)-
hatpauc)
       -(1-pnorm(hatq0,mlemean,mlestdd))*((x[i]<=hatq0)-(1-p0))
       -(1-pnorm(hatq1,mlemean,mlestdd))*((x[i]<=hatq1)-(1-p1))
}

# compute the deravative of G(pnorm)
dg1<-rep(0,m)
dg2<-rep(0,m)
for (i in 1:m)
{
 dg1[i]<--dnorm(x[i],mlemean,mlestdd)*(hatq1 <=x[i])*(x[i] <= hatq0)
 dg2[i]<--dnorm(x[i],mlemean,mlestdd)*(x[i]-mlemean)*(hatq1 <=x[i])*(x[i] <=
hatq0)/(2*mlestdd^2)
 }

# compute matrix D
d[1,1]<-mean(dg1)
d[2,1]<-mean(dg2)

# compute hatsigmasquare
hatsigmasquare<-var(b)+rho*t(d)%*%s%*%d

normallow<-hatpauc-z*sqrt(hatsigmasquare)/sqrt(m)
normalup<-hatpauc+z*sqrt(hatsigmasquare)/sqrt(m)

length<-normalup-normallow

normallength<-normallength+length

if ((normallow<=truepauc) & (normalup>=truepauc)) normalcoverage<-
normalcoverage+1
```

```
 #end of computing the normal approxamation based C.I.
#1111111111111111111111111111111111111111111111111111111111

#2222222222222222222222222222222222222222222222222222222222
 # compute the CI by using bootstrap method
 B<-150
 starvm<-matrix(0,m,B)
 lstar<-rep(0,B)
 bootlist<-mybootstrap(B,x,y,m,n,p0,p1)
 starpauc<-bootlist[[1]]
 starssquare<-bootlist[[2]]
 starvm<-bootlist[[3]]


 starsigmasquare<-var(starpauc)
 meanstarpauc<-mean(starpauc)

 bootlow1<-hatpauc-z*sqrt(starsigmasquare)
 bootup1<-hatpauc+z*sqrt(starsigmasquare)

 bootlow2<-meanstarpauc-z*sqrt(starsigmasquare)
 bootup2<-meanstarpauc+z*sqrt(starsigmasquare)

 length1<-bootup1-bootlow1
 length2<-bootup2-bootlow2

 bootlength1<-bootlength1+length1
 bootlength2<-bootlength2+length2

 if ((bootlow1<=truepauc) & (bootup1>=truepauc)) bootcoverage1<-
 bootcoverage1+1
 if ((bootlow2<=truepauc) & (bootup2>=truepauc)) bootcoverage2<-
 bootcoverage2+1


 # end of computing the CI by using bootstrap method
 #2222222222222222222222222222222222222222222222222222222222

 #3333333333333333333333333333333333333333333333333333333333
 # compute the EL-based semi-parametric CI:HBELI, HBELII

 starsigmasquare<-var(starpauc)
 meanstarssquare<-mean(starssquare)

 hatssquare<-mean((hatv-hatpauc)^2)
 starc<-hatssquare/(m*starsigmasquare)

 # compute lambda and el bound for first (hatc)
 # x[1]: lambda. x[2]: pauc

 # compute lambda and el bound for first (starc)
 f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
 x[2]))),starc*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
 sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
 solup<-solveNonlinear(f,c(0,0),c(0,0.95))
 ellow1<-sollow$x[2]
 elup1<-solup$x[2]


 starc<-meanstarssquare/(m*starsigmasquare)

 # compute lambda and el bound for second (starc)
```

```
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),starc*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow2<-sollow$x[2]
elup2<-solup$x[2]


length1<-elup1-ellow1
length2<-elup2-ellow2

ellength1<-ellength1+length1
ellength2<-ellength2+length2

if ((ellow1<=truepauc) & (elup1>=truepauc)) elcoverage1<-elcoverage1+1
if ((ellow2<=truepauc) & (elup2>=truepauc)) elcoverage2<-elcoverage2+1

# end of computing the EL-based semi-parametric CI
#3333333333333333333333333333333333333333333333333333333333

#4444444444444444444444444444444455555555555555555555555555555
# compute the CI by using bootstrap method

for (b in 1:B)
{
wjhatb<-rep(0,m)
  for (i in 1:m)
{
  wjhatb[i]<-starvm[i,b]-hatpauc
}
   f<-function(lam)mean(wjhatb/(1+lam*wjhatb))
  lambda<-solveNonlinear(f,c(0),c(0.01))$x
  lstar[b]<-2*sum(log(abs(1+lambda*wjhatb)))
}

ctail<-1/mean(lstar)
cstar<-quantile(lstar, c(1-alpha))


# compute lambda and el bound for first (hatc)
# x[1]: lambda. x[2]: pauc


# compute lambda and el bound for third
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),2*sum(log(abs(1+x[1]*(hatv-x[2]))))-cstar)
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow3<-sollow$x[2]
elup3<-solup$x[2]


# compute lambda and el bound for fourth
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),ctail*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow4<-sollow$x[2]
elup4<-solup$x[2]


length3<-elup3-ellow3
length4<-elup4-ellow4
```

```
   ellength3<-ellength3+length3
   ellength4<-ellength4+length4

   if ((ellow3<=truepauc) & (elup3>=truepauc)) elcoverage3<-elcoverage3+1
   if ((ellow4<=truepauc) & (elup4>=truepauc)) elcoverage4<-elcoverage4+1

   # end of computing the EL-based semi-parametric CI

   #44444444444444444444444444444444555555555555555555555555555555
   #66666666666666666666666666666666666666666666666666666666666666
   #computing the ML CI

    mu0<-mean(x)
   mu1<-mean(y)
   sigma0<-sqrt(var(x))
   sigma1<-sqrt(var(y))

   hatpauc<-integrate(rpml, mu0=mu0, sigma0=sigma0, mu1=mu1, sigma1=sigma1,
   lower = p0, upper = p1)$integral
   # ML estimate of pauc

   k1<-integrate(pvar1, mu0=mu0, sigma0=sigma0, mu1=mu1, sigma1=sigma1, lower =
   p0, upper = p1)$integral
   k2<-integrate(pvar2, mu0=mu0, sigma0=sigma0, mu1=mu1, sigma1=sigma1, lower =
   p0, upper = p1)$integral

   varpauc<-
   k1^2*(sigma0^2/m+sigma1^2/n)/sigma1^2+k2^2*sigma0^2/(2*m*sigma1^2)+((mu0-
   mu1)*k1+sigma0*k2)^2/(2*n*sigma1^2)
   # variance of ML estimate of pauc


   ##  compute the ML-based interval.

      mllow<-hatpauc-z*sqrt(varpauc)
      mlup<-hatpauc+z*sqrt(varpauc)

      if ((mllow <= truepauc) & (mlup >= truepauc))mlcoverage<-mlcoverage+1
      mllength<-mllength+2*z*sqrt(varpauc)           # The length of CI


   # end of ML CI
   #66666666666666666666666666666666666666666666666666666666666666

} # end of loop

coverage<-c(normalcoverage/iter,bootcoverage1/iter,
   bootcoverage2/iter,elcoverage1/iter,elcoverage2/iter,elcoverage3/iter,elcove
   rage4/iter,mlcoverage/iter)
length<-
   c(normallength/iter,bootlength1/iter,bootlength2/iter,ellength1/iter,ellengt
   h2/iter,ellength3/iter,ellength4/iter,mllength/iter)

cat("m=", m, "n=", n, "iter=", iter, "\n")
cat("meanx=", mux, "meany=", muy, "STDx=", stddx, "STDy=", stddy, "p0=", p0,
   "p1=", p1, "AUC(p0,p1)=", truepauc, "\n")
cat("Coverage of (NA, BOOT1, BOOT2, EL1, EL2):", coverage, "\n")
cat("Average length of (NA, BOOT1, BOOT2, EL1, EL2):", length, "\n")
```

APPENDIX D: THE S-PLUS CODE FOR REAL DATA ANALYSIS

```
# solve non-linear equations(for solving lambda)
solveNonlinear<-function(f,y0,x,...)
{
   g<-function(x,y0,f)sum((f(x)-y0)^2)
   g$y0<-y0
   g$f<-f
   nlmin(g,x,max.fcal=1000,max.iter=1000,...)
}

m<-127            # sample size of non-diseased sample: x1,...xm
n<-67            # sample size of diseased sample: y1,...yn

p0<-0
p1<-0.7

alpha<-0.05
z<-qnorm(1-alpha/2)

mybootstrap<-function(B,x,y,m,n,p0,p1){
starpauc<-0
starssquare<-0
starvm<-matrix(0,ncol=B,nrow=m)
   for (b in 1:B)
   {
      xxb<-sample(x,m,replace=T)
      yyb<-sample(y,n,replace=T)
      hatq0b<-0
      hatq1b<-0
      xb<-xxb**(-0.56)
       yb<-yyb**(-0.56)

      hatq1b<-quantile(xxb, c(1-p0))**(-0.56)   # hatq0b, hatq1b: sample
   quantiles
      hatq0b<-quantile(xxb, c(1-p1))**(-0.56)   # of F in bootstrap

      starmlemean<-mean(yb)              # MLE estimate of startheta
       starmlestdd<-(mean((yb-starmlemean)^2))^(1/2)

      starv<-0                           # compute starv
      for (i in (1:m))
      {
         starv[i]<- (1-pnorm(xb[i],starmlemean,starmlestdd))*(hatq1b
   <=xb[i])*(xb[i] <= hatq0b)
         starvm[i,b]<- (1-pnorm(xb[i],starmlemean,starmlestdd))*(hatq1b
   <=xb[i])*(xb[i] <= hatq0b)
      }
      starpauc[b]<-mean(starv)
      starssquare[b]<-var(starv)
   }
   list(starpauc,starssquare,starvm)
}
#main code

   xx<-
   c(10.9,11,13.2,22.6,15.2,9.6,13.5,17.5,13.3,17.1,22.7,6.9,14.6,18.2,5.6,7.9,
   12.6,16.1,9.9,10.1,

   16.4,15.3,8.1,3.5,20.7,19.9,11.8,13,7.4,6,8.8,9.9,10.2,16.8,6.4,10.9,8.6,16.
   4,10.3,2.8,
```

```
17.1,10.9,8,15.6,11.8,5.1,12,16.6,15.3,4.4,9.3,15.1,16.5,21.8,15.8,10.3,12,1
0.5,6.7,11.3,

15.3,13.7,12.2,17.9,15.4,22.3,8.7,5.3,16.1,9.8,12.9,3.9,14.2,16.2,9.7,10.3,5
.8,10.6,11.9,14.5,

14,8.9,17.1,10.3,10,12.3,10,5.9,9.9,13.7,12.7,11.3,6.9,15.1,6.1,12.2,7.3,10.
7,7,11.9,

16.6,7.6,18.1,21.6,12.5,5.6,8.9,3.8,11.1,10.8,10.7,17.4,14.5,15.3,15.3,11.9,
12,16.4,12.1,12.9,
     11.7,14,10.9,6,15.3,11.4,20.3)

yy<-
c(25.6,26.8,8.8,17.4,23.8,20.2,11,18.3,16.7,21.6,16.1,36.4,49.1,32.2,14,16.9
,12.7,9.7,110,63.7,

73,23.3,31.9,41.6,18.8,17,22,10.9,19.9,18.8,12.9,15.5,20.6,16,19.8,16.4,10.4
,17.1,25.3,62.9,

51.6,33.9,22.2,20.9,36,12.8,11.7,18.6,19.4,11.2,21,20.1,8.3,25.2,16.6,22.7,2
1.3,10.2,12.1,22,
     14.4,8.9,27.1,49.1,11.8,15.1,14.2)
x<-xx^(-0.56)
y<-yy^(-0.56)

hatq1<-quantile(xx, c(1-p0))^(-0.56)   # hatq0, hatq1: sample quantiles fo F
hatq0<-quantile(xx, c(1-p1))^(-0.56)


mlemean<-mean(y)               # MLE estimate of theta
mlestdd<-(mean((y-mlemean)^2))^(1/2)

hatv<-rep(0,m)                 # hat v_i=(1-hat G(x_i, hat theta))*
t<-rep(0,m)                              # I(x_i \in (hatq1,hatq0))
for (i in 1:m)
{
 hatv[i]<-(1-pnorm(x[i],mlemean,mlestdd))*(hatq1 <=x[i])*(x[i] <= hatq0)
 t[i]<-(hatq1 <=x[i])*(x[i] <= hatq0)
}
hatpauc<-mean(hatv)           # estimate of pauc
p<-mean(t)


#1111111111111111111111111111111111111111111111111111111111111
# compute the normal approxamation based C.I.

s<-matrix(0,ncol=2,nrow=2)
d<-matrix(0,ncol=1,nrow=2)

s[1,1]<-mlestdd^2
s[1,2]<-0
s[2,1]<-0
s[2,2]<-2*mlestdd^4

# compute B(x,theta,q0,q1)
b<-rep(0,m)
for (i in 1:m)
{
 b[i]<-((1-pnorm(x[i],mlemean,mlestdd))*(hatq1 <=x[i])*(x[i] <= hatq0)-
hatpauc)
       -(1-pnorm(hatq0,mlemean,mlestdd))*((x[i]<=hatq0)-(1-p0))
```

```
        -(1-pnorm(hatq1,mlemean,mlestdd))*((x[i]<=hatq1)-(1-p1))
 }

 # compute the deravative of G(pnorm)
 dg1<-rep(0,m)
 dg2<-rep(0,m)
 for (i in 1:m)
 {
  dg1[i]<--dnorm(x[i],mlemean,mlestdd)*(hatq1 <=x[i])*(x[i] <= hatq0)
  dg2[i]<--dnorm(x[i],mlemean,mlestdd)*(x[i]-mlemean)*(hatq1 <=x[i])*(x[i] <=
 hatq0)/(2*mlestdd^2)
  }

 # compute matrix D
 d[1,1]<-mean(dg1)
 d[2,1]<-mean(dg2)

 # compute hatsigmasquare
 hatsigmasquare<-var(b)+rho*t(d)%*%s%*%d

 normallow<-p-hatpauc-z*sqrt(hatsigmasquare)/sqrt(m)
 normalup<-p-hatpauc+z*sqrt(hatsigmasquare)/sqrt(m)

#end of computing the normal approxamation based C.I.
#1111111111111111111111111111111111111111111111111111111

#2222222222222222222222222222222222222222222222222222222
 # compute the CI by using bootstrap method
 B<-150
 starvm<-matrix(0,m,B)
 lstar<-rep(0,B)
 bootlist<-mybootstrap(B,xx,yy,m,n,p0,p1)
 starpauc<-bootlist[[1]]
 starssquare<-bootlist[[2]]
 starvm<-bootlist[[3]]


 starsigmasquare<-var(starpauc)
 meanstarpauc<-mean(starpauc)

 bootlow1<-p-hatpauc-z*sqrt(starsigmasquare)
 bootup1<-p-hatpauc+z*sqrt(starsigmasquare)

 bootlow2<-p-meanstarpauc-z*sqrt(starsigmasquare)
 bootup2<-p-meanstarpauc+z*sqrt(starsigmasquare)


 # end of computing the CI by using bootstrap method
 #2222222222222222222222222222222222222222222222222222222

 #3333333333333333333333333333333333333333333333333333333
 # compute the EL-based semi-parametric CI:HBELI, HBELII

 starsigmasquare<-var(starpauc)
 meanstarssquare<-mean(starssquare)

 hatssquare<-mean((hatv-hatpauc)^2)
 starc<-hatssquare/(m*starsigmasquare)

 # compute lambda and el bound for first (hatc)
 # x[1]: lambda. x[2]: pauc

 # compute lambda and el bound for first (starc)
```

```
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),starc*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow1<-p-solup$x[2]
elup1<-p-sollow$x[2]


starc<-meanstarssquare/(m*starsigmasquare)

# compute lambda and el bound for second (starc)
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),starc*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow2<-p-solup$x[2]
elup2<-p-sollow$x[2]


# end of computing the EL-based semi-parametric CI
#3333333333333333333333333333333333333333333333333333333333

#4444444444444444444444444444444455555555555555555555555555555
# compute the CI by using bootstrap method

for (b in 1:B)
{
wjhatb<-rep(0,m)
  for (i in 1:m)
{
  wjhatb[i]<-starvm[i,b]-hatpauc
}
   f<-function(lam)mean(wjhatb/(1+lam*wjhatb))
  lambda<-solveNonlinear(f,c(0),c(0.01))$x
  lstar[b]<-2*sum(log(abs(1+lambda*wjhatb)))
}

ctail<-1/mean(lstar)
cstar<-quantile(lstar, c(1-alpha))


# compute lambda and el bound for first (hatc)
# x[1]: lambda. x[2]: pauc


# compute lambda and el bound for third
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),2*sum(log(abs(1+x[1]*(hatv-x[2]))))-cstar)
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow3<-p-solup$x[2]
elup3<-p-sollow$x[2]


# compute lambda and el bound for fourth
f<-function(x) c(mean((hatv-x[2])/(1+x[1]*(hatv-
x[2]))),ctail*2*sum(log(abs(1+x[1]*(hatv-x[2]))))-qchisq(1-alpha,1))
sollow<-solveNonlinear(f,c(0,0),c(0,0.01))
solup<-solveNonlinear(f,c(0,0),c(0,0.95))
ellow4<-p-solup$x[2]
elup4<-p-sollow$x[2]

# end of computing the EL-based semi-parametric CI
```

```
   #444444444444444444444444444444444555555555555555555555555555555

#output
cat(normallow/(p1-p0),normalup/(p1-p0),bootlow1/(p1-p0),bootup1/(p1-
   p0),bootlow2/(p1-p0),bootup2/(p1-p0),ellow1/(p1-p0),elup1/(p1-
   p0),ellow2/(p1-p0),elup2/(p1-p0),ellow3/(p1-p0),elup3/(p1-p0),ellow4/(p1-
   p0),elup4/(p1-p0))
```