7-17-2008

Bootstrap and Empirical Likelihood-based Semi-parametric Inference for the Difference between Two Partial AUCs

Xin Huang

Bootstrap and Empirical Likelihood-based Semi-parametric Inference For the Difference between Two Partial AUCs

by

Xin Huang

Under the Direction of Dr. Gengsheng Qin

ABSTRACT

With new tests being developed and marketed, the comparison of the diagnostic accuracy of two continuous-scale diagnostic tests are of great importance. Comparing the partial areas under the receiver operating characteristic curves (pAUC) is an effective method to evaluate the accuracy of two diagnostic tests. In this thesis, we study the semi-parametric inference for the difference between two pAUCs. A normal approximation for the distribution of the difference between two pAUCs has been derived. The empirical likelihood ratio for the difference between two pAUCs is defined and its asymptotic distribution is shown to be a scaled chi-quare distribution. Bootstrap and empirical likelihood based inferential methods for the difference are proposed. We construct five confidence intervals for the difference between two pAUCs. Simulation studies are conducted to compare the finite sample performance of these intervals. We also use a real example as an application of our recommended intervals.

INDEX WORDS: ROC, AUC, Partial AUC, Semi-parametric, Bootstrap, Empirical likelihood, Confidence interval

Bootstrap and Empirical Likelihood-based Semi-parametric Inference For the Difference between Two Partial AUCs

by

Xin Huang

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

Bootstrap and Empirical Likelihood-based Semi-parametric Inference For the Difference between Two Partial AUCs

by

Xin Huang

Chair:	Dr. Gengsheng Qin
Committee:	Dr. Yu-Sheng Hsu
	Dr. Yixin Fang

Electronic Version Approved:

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

This thesis is dedicated to Gengsheng Qin,

My parents,

and all my best friends.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Gengsheng Qin, for all of his guidance and support.

I would also like to thank Dr. Yu-Sheng Hsu and Dr. Yixin Fang.

Finally, I would like to thank my family and friends for all of their support.

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1

INTRODUCTION

The laboratory diagnostic test is a vital component of modern medical practice in discriminating diseased subjects from healthy individuals. As far as a continuous-scale test is concerned, a person is assessed as diseased (positive) if the test value is greater than a given threshold, otherwise the subject is diagnosed as healthy (negative). The accuracy of the test can be measured by the true positive rate (sensitivity) and the true negative rate (specificity). Let X and Y be the results of a continuous-scale test for a non-diseased and a diseased subject, respectively. The sensitivity and specificity of the test for a given threshold c are defined as follows:

$$Se = P(Y \geq c) = 1 - G(c);\ Sp = P(X \leq c) = F(c), \tag{1.1}$$

respectively, where F and G are the distribution functions of X and Y, respectively. From (1.1), there is a trade-off between sensitivity and specificity as the threshold varies, which affects the accuracy of the test.

The receiver operating characteristic (ROC) curve, which is a plot of the sensitivity against 1- specificity over all possible thresholds, is a useful graphical tool for assessing the discriminatory accuracy of a diagnostic test. The ROC curve can mathematically be represented by $R(p) = 1 - G(F^{-1}(1-p))$, where $F^{-1}(\cdot)$ is the inverse function of $F(\cdot)$. The area under the ROC curve (AUC), defined as $\delta = \int_0^1 R(p)dp$,

is the most popular one-number summary index of the discriminatory accuracy. The larger the AUC is, the better the diagnostic test will be. Under some circumstance, such as a particular range of specificity or sensitivity values is relevant, Shapiro (1999) recommended a more appropriate index, partial AUC (pAUC), as an alternative measurement for the diagnostic accuracy. The pAUC over the interval (p_0, p_1) of FPR is defined as

$$\delta_{p_0,p_1} = \int_{p_0}^{p_1} R(p)dp, \text{ for } 0 \leq p_0 \leq p_1 \leq 1. \tag{1.2}$$

With new tests being developed and marketed, the comparison of diagnostic tests are of greatly importance (Delong *et al*, 1988). A common method for such comparison is based on the full or partial AUC. Metz *et al.* (1984) proposed a parametric approach to compare two full AUCs under a binormal model assumption. McClish (1989), Thompson *et al.* (1989), and Jiang *et al.* (1996) provided parametric normal-theory methods for comparing diagnostic tests with respect to partial AUC. However, the parametric methods, which assume parametric models for both X and Y, may be sensitive to departures from the distributional assumptions and can only provided a limited range of distributional forms for the diseased and non-diseased populations.

Delong *et al.* (1988) suggested a non-parametric method for testing the equality of two full AUCs by using the theory on generalized U-statistic. Wieand *et al.* (1989) gave a class of nonparametric statistics for comparing two partial AUCs based on weighted average of sensitivities, with the asymptotic variance for the difference involved in estimation of distribution and quantile functions. Wieand *et al.*'s method is mathematically complicated and thus hard to apply in practice. Mossman (1995) suggested using resampling methods such as the bootstrap and jackknife to make inference about the difference of two partial AUCs. Qin *et al.* (2008) recently proposed

nonparametric inferential methods for the difference of two pAUCs based on bootstrap and empirical likelihood. Although the nonparametric methods are distribution-free, i.e., assuming no parametric forms for the distributions of X and Y, they may be less efficient than its parametric counterpart in making inference for the full or partial AUC.

Semi-parametric models are statistical models between pure parametric and non-parametric models. Semi-parametric methods may inherit the good properties of both parametric and nonparametric methods. They 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 diagnostic testing, it may not be reasonable to adopt two parametric models for the test results from the diseased and non-diseased subjects. A semi-parametric model assumption for the test results X and Y seems to be more desirable than a parametric one and the semi-parametric approach may be more efficient than the fully nonparametric one. In this thesis, we will develop new semi-parametric methods for the difference between two partial AUCs. The normal approximation, bootstrap and empirical likelihood will be used for making inference about the difference between two partial AUCs.

The thesis is organized as follows. In Chapter 2, we establish the normal approximation theory and construct normal approximation and bootstrap based confidence intervals for the difference between two partial AUCs under a semi-parametric model for the test results. In Chapter 3, we propose hybrid bootstrap and empirical likelihood based confidence intervals for the difference between two partial AUCs. In

Chapter 4, we conduct a series of simulation studies to evaluate the performance of the proposed intervals. In Chapter 5, we use a pancreatic cancer serum biomarkers example to illustrate the recommended intervals. At last, we discuss the conclusions in Chapter 6.

Chapter 2

NORMAL APPROXIMATION AND BOOTSTRAP BASED CONFIDENCE INTERVALS

Consider two diagnostic tests T_1 and T_2 that yield continuous measurements. Assume that both tests are performed on the same m controls (non-diseased) and n cases (diseased). Let (X_{1i}, X_{2i}), $i = 1, 2, \cdots, m$, be i.i.d. bivariate outcomes from the population (X_1, X_2) that have an unknown joint distribution $F(x_1, x_2)$, and (Y_{1j}, Y_{2j}), $j = 1, 2, \cdots, n$, be i.i.d. bivariate outcomes from the population (Y_1, Y_2) that have a joint parametric distribution $G(y_1, y_2; \theta)$, where the parameter θ is unknown and belongs to some set $\Theta \subset \mathbb{R}^r$, $r \geq 1$. Assume also (X_{1i}, X_{2i})'s and (Y_{1j}, Y_{2j})'s are mutually independent. Denote the marginal distribution functions of $F(x_1, x_2)$ and $G(y_1, y_2; \theta)$ by $F_k(x_k)$ and $G_k(y_k; \theta)$, respectively, $k = 1, 2$. The partial AUC of test T_k (k = 1, 2) over the interval (p_0, p_1), denoted by $\delta^{(k)}_{p_0,p_1}$, is

$$\delta^{(k)}_{p_0,p_1} = \int_{p_0}^{p_1} R_k(p)dp, \quad \text{for} \quad 0 \leq p_0 < p_1 \leq 1,$$

where $R_k(p) = 1 - G_k(F_k^{-1}(1-p); \theta)$ is the ROC curve of test T_k (k=1, 2). The difference between two partial AUCs is $\Delta_{p_0,p_1} = \delta^{(2)}_{p_0,p_1} - \delta^{(1)}_{p_0,p_1}$. We wish to make inference about Δ_{p_0,p_1} based on test results (X_{1i}, X_{2i})'s and (Y_{1j}, Y_{2j})'s.

2.1 Normal Approximation Based Confidence Interval

One of the most popular methods to construct a confidence interval for an unknown parameter is normal approximation. To construct a normal approximation based confidence interval for $\Delta_{p_0p_1}$, first, we need to obtain an appropriate estimator for $\Delta_{p_0p_1}$; and then, we have to derive the asymptotic normal distribution of this estimator.

Bamber (1975) showed that the full AUC $\delta^{(k)} = AUC_k(0,1) = P(Y_k \geq X_k)$, which can be interpreted as the probability that in a randomly selected pair of diseased and non-diseased subjects, the $k-th$ test value of the diseased subject is higher than or equal to that of the non-diseased subject. The partial AUC where p falls in (p_0, p_1) can be expressed as follows:

$$\begin{aligned}
\delta^{(k)}_{p_0p_1} &= AUC_k(p_0,p_1) \\
&= \int_{p_0}^{p_1} R_k(p)dp \\
&= P(Y_k \geq X_k,\ X_k \in (q_{k1}, q_{k0});\ \theta) \\
&= E[P(Y_k \geq X_k;\theta)|X_k \in (q_{k1}, q_{k0})] \\
&= E[V_k(X_k;\theta)], \qquad (2.1)
\end{aligned}$$

where $k = 1, 2$, $q_{kl} = F_k^{-1}(1-p_l)$, $l = 0, 1$, F_k^{-1} is the inverse function of F_k, and

$$\begin{aligned}
V_k(X_k;\theta) &= P(Y_k \geq X_k;\theta|X_k \in (q_{k1}, q_{k0})) \\
&= [1 - G_k(X_k;\theta)]I(X_k \in (q_{k1}, q_{k0})). \qquad (2.2)
\end{aligned}$$

We can use maximum likelihood method to estimate the unknown parameter θ based on test results (Y_{1j}, Y_{2j})'s from the population $G(y_1, y_2;\theta)$, and then obtain estimators

for $\delta^{(k)}_{p_0p_1}$ and $\Delta_{p_0p_1}$:

$$\hat{\delta}^{(k)}_{p_0p_1} = \frac{1}{m}\sum_{i=1}^{m} V_k(X_{ki};\hat{\theta}),$$

$$\hat{\Delta}_{p_0p_1} = \hat{\delta}^{(2)}_{p_0p_1} - \hat{\delta}^{(1)}_{p_0p_1}, \tag{2.3}$$

where $\hat{\theta}$ is the MLE of θ based on $(Y_{1j}, Y_{2j})'s,\ j = 1,\ldots,n,\ V_k(X_{ki};\hat{\theta}) = [1 - G_k(X_{ki};\hat{\theta})]I(X_{ki} \in (\hat{q}_{k1}, \hat{q}_{k0})),\ \hat{q}_{kl} = \hat{F}_k^{-1}(1-p_l), l = 0, 1$, and $\hat{F}_k$ is the empirical distributions of F_k (k = 1, 2).

Using Central Limit Theorem (CLT) for multivariate variables, we derive the asymptotic distribution of $\hat{\Delta}_{p_0p_1}$ in the following theorem.

Theorem 2.1. Assume that $0 < \lim_{m,n\to\infty} m/n \equiv \nu < \infty$ is a constant. Then

$$\sqrt{m}(\hat{\Delta}_{p_0p_1} - \Delta_{p_0p_1}) \longrightarrow N(0, \Sigma^2_{p_0p_1}),$$

where

$$\begin{aligned}
\Sigma^2_{p_0p_1} &= Var[\sum_{k=1}^{2} B_k(X_k;\theta,q_{k0},q_{k1})] + \\
&\quad \nu[\sum_{k=1}^{2} D_k(\theta,q_{k0},q_{k1})]^\tau \Sigma_\theta [\sum_{k=1}^{2} D_k(\theta,q_{k0},q_{k1})], \\
B_k(X_k;\theta,q_{k0},q_{k1}) &\equiv [(1-G_k(X_k,\theta))I(X_k \in (q_{k1},q_{k0})) - \delta^{(k)}_{p_0p_1}] - \\
&\quad \sum_{l=0}^{1}[1-G_k(q_{kl};\theta)][I(X_k \le q_{kl}) - (1-p_l)], \\
D_k(\theta,q_{k0},q_{k1}) &\equiv E[g_k(X_k;\theta)I(X_k \in (q_{k1},q_{k0}))],
\end{aligned}$$

$g_k(X_k;\theta)$ is the derivative of $G_k(X_k,\theta)$ with respect to $\theta \in \Theta \subset \mathbb{R}^r$, and Σ_θ is the asymptotic covariance matrix of $\sqrt{n}(\hat{\theta}-\theta)$. The asymptotic variance $\Sigma^2_{p_0,p_1}$ of $\hat{\Delta}_{p_0,p_1}$ in Theorem 2.1 is a function of unknown parameter θ and quantiles q_{kl}'s. To construct

a confidence interval for Δ_{p_0,p_1}, we have to estimate $\Sigma^2_{p_0,p_1}$. Replacing θ and q_{kl}'s by the MLE $\hat{\theta}$ and sample quantiles $\hat{q}_{kl}$'s respectively, we obtain the following estimator for $\Sigma^2_{p_0,p_1}$:

$$\begin{aligned}\hat{\Sigma}^2_{p_0,p_1} &= (m-1)^{-1}\sum_{i=1}^{m}((\hat{B}_{1i}+\hat{B}_{2i})-(\bar{\hat{B}}_1+\bar{\hat{B}}_2))^2 + \\ &\quad (m/n)[\sum_{k=1}^{2}\hat{D}_k(\hat{\theta},\hat{q}_{k0},\hat{q}_{k1})]^{\tau}\hat{\Sigma}_{\theta}[\sum_{k=1}^{2}\hat{D}_k(\hat{\theta},\hat{q}_{k0},\hat{q}_{k1})],\end{aligned}$$

where

$$\begin{aligned}\hat{B}_{ki} &= [(1-G_k(X_{ki},\hat{\theta}))I(X_{ki}\in(\hat{q}_{k1},\hat{q}_{k0}))-\hat{\delta}^{(k)}_{p_0p_1}] - \\ &\quad \sum_{l=0}^{1}[1-G_k(\hat{q}_{kl};\hat{\theta})][I(X_{ki}\le\hat{q}_{kl})-(1-p_l)], \\ \bar{\hat{B}}_k &= m^{-1}\sum_{i=1}^{m}\hat{B}_{ki}, \\ \hat{D}_k(\hat{\theta},\hat{q}_{k0},\hat{q}_{k1}) &= m^{-1}\sum_{i=1}^{m}g_k(X_{ki};\hat{\theta})I(X_{ki}\in(\hat{q}_{k1},\hat{q}_{k0}))],\end{aligned}$$

and $\hat{\Sigma}_{\theta}$ is a consistent estimate for Σ_{θ}.

Therefore, a $(1-\alpha)$-th normal approximation (NA) based confidence interval for $\Delta_{p_0p_1}$ can be constructed as follows:

$$\hat{\Delta}_{p_0p_1} \pm z_{1-\frac{\alpha}{2}}\cdot\frac{\hat{\Sigma}_{p_0p_1}}{\sqrt{m}},$$

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

The estimate for the asymptotic variance $\Sigma^2_{p_0p_1}$ is a plug-in estimate by using the MLE $\hat{\theta}$ and sample quantiles. It may be an unstable estimate for the asymptotic variance. In fact, our simulation studies in this thesis indicate that the NA-based intervals have longer interval length and are too conservative than its competitors

such as the bootstrap and EL-based intervals introduced in the following chapters.

2.2 Bootstrap Based Confidence Interval

When the asymptotic variance of an estimator is unknown and of a complex form, bootstrap method is usually used to estimate the asymptotic variance. In this section, we apply bootstrap method to estimate the asymptotic variance of $\hat{\Delta}_{p_0p_1}$ and then construct confidence intervals for the difference between two pAUCs.

We draw a bootstrap resample $\{X^*_{k1}, \cdots, X^*_{km}\}$ with replacement from $\{X_{k1}, \cdots, X_{km}\}$, and another bootstrap resample $\{Y^*_{k1}, \cdots, Y^*_{kn}\}$ with replacement from $\{Y_{k1}, \cdots, Y_{kn}\}$. Then calculate a bootstrap copy of $\hat{\delta}^{(k)}_{p_0p_1}$:

$$\delta^{(k)*}_{p_0p_1} = \frac{1}{m}\sum_{i=1}^{m} V_k(X^*_{ki}; \theta^*),$$

where

$$V_k(X^*_{ki}; \theta^*) = [1 - G_k(X^*_{ki}; \theta^*)]I(X^*_{ki} \in (q^*_{k1}, q^*_{k0})),$$

θ^* is the bootstrap version of $\hat{\theta}$, and $q^*_{kl} = F^{*-1}_k(1 - p_l)$ is the $(1 - p_l)$-th sample quantile of $\{X^*_{k1}, \cdots, X^*_{km}\}$, $k = 1, 2,\ l = 0, 1$. Then the bootstrap estimate for the difference between two pAUCs can be calculated as:

$$\Delta^*_{p_0p_1} = \delta^{(2)*}_{p_0p_1} - \delta^{(1)*}_{p_0p_1}.$$

.

After B repetitions of above processes, B bootstrap copies of $\hat{\delta}^{(k)}_{p_0p_1}$ and $\hat{\Delta}_{p_0p_1}$ are obtained:

$$\{\delta^{(k)*}_{p_0p_1,b} : b = 1, 2, \ldots, B\},\ k = 1, 2\ ,$$

$$\{\Delta^*_{p_0p_1,b} = \delta^{(2)*}_{p_0p_1,b} - \delta^{(1)*}_{p_0p_1,b} :\ b = 1, 2, \ldots, B\}.$$

The bootstrap estimate for the variance of $\hat{\Delta}_{p_0p_1}$ is:

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

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

Two bootstrap based normal approximation confidence intervals are defined as follows:

1. BI interval:

$$\hat{\Delta}_{p_0p_1} \pm z_{1-\frac{\alpha}{2}} \cdot \Sigma^*_{p_0p_1};$$

2. BII interval:

$$\bar{\Delta}^*_{p_0p_1} \pm z_{1-\frac{\alpha}{2}} \cdot \Sigma^*_{p_0p_1}.$$

Chapter 3

EMPIRICAL LIKELIHOOD BASED CONFIDENCE INTERVALS

Another popular nonparametric method to obtain confidence intervals for the mean is the empirical likelihood (EL) method introduced by Owen (1990). The EL method has several advantages over the other nonparametric methods (Owen, 2001). For example, it has better small performance than normal approximation based approaches; there is no need for a pivot due to its internal studentization. Qin *et al* (2006) developed an EL approach for the inference on the full AUC. In this chapter, we extend the EL approach to the semi-parametric models for the inference about the difference between two partial AUCs.

From (2.2) and (2.3), we can have:

$$\Delta_{p_0p_1} = \delta^{(2)}_{p_0p_1} - \delta^{(1)}_{p_0p_1} = E[V_2(X_2;\theta) - V_1(X_1;\theta)],$$

where $V_k(X_k;\theta) = [1 - G_k(X_k;\theta)]I(X_k \in (q_{k1}, q_{k0}))$, $k = 1, 2$, and $q_{kl} = F_k^{-1}(1 - p_l)$, $l = 0, 1$.

Based on the relationship of the difference between two pAUCs and the restrict placement values $V_2(X_2;\theta)$ and $V_1(X_1;\theta)$, we can define the profile EL for $\Delta_{p_0p_1}$ as

follows:

$$\begin{aligned} L(\Delta_{p_0p_1}) &= \sup\{\prod_{k=1,2}\prod_{i=1}^{m} P_{ki} : \sum_{i=1}^{m} P_{ki} = 1, P_{ki} \geq 0, i = 1, \cdots, m, \\ &\quad \sum_{i=1}^{m} P_{ki}(V_k(X_{ki};\hat{\theta}) - \delta_{p_0p_1}^{(k)}) = 0,\ k = 1, 2, \\ &\quad \sum_{i=1}^{m} P_{2i}V_2(X_{2i};\hat{\theta}) - \sum_{i=1}^{m} P_{1i}V_1(X_{1i};\hat{\theta}) = \Delta_{p_0p_1}\}. \end{aligned}$$

Then, by the Lagrange multiplier, we obtain the following empirical log-likelihood ratio (ELR) for $\Delta_{p_0p_1}$:

$$\begin{aligned} l(\Delta_{p_0p_1}) &= 2[\sum_{i=1}^{m} \log(1 - 2\lambda(V_1(X_{1i};\hat{\theta}) - \delta_{p_0p_1}^{(1)})) + \\ &\quad \sum_{i=1}^{m} \log(1 + 2\lambda(V_2(X_{2i};\hat{\theta}) - \delta_{p_0p_1}^{(2)}))], \end{aligned} \tag{3.1}$$

where λ, $\delta_{p_0p_1}^{(k)}$ satisfy the following equations:

$$\frac{1}{m}\sum_{i=1}^{m} \frac{V_1(X_{1i};\hat{\theta}) - \delta_{p_0p_1}^{(1)}}{1 - 2\lambda(V_1(X_{1i};\hat{\theta}) - \delta_{p_0p_1}^{(1)})} = 0, \tag{3.2}$$

$$\frac{1}{m}\sum_{i=1}^{m} \frac{V_2(X_{2i};\hat{\theta}) - \delta_{p_0p_1}^{(2)}}{1 + 2\lambda(V_2(X_{2i};\hat{\theta}) - \delta_{p_0p_1}^{(2)})} = 0, \tag{3.3}$$

$$\frac{1}{m}\sum_{i=1}^{m} \frac{V_2(X_{2i};\hat{\theta})}{1 + 2\lambda(V_2(X_{2i};\hat{\theta}) - \delta_{p_0p_1}^{(2)})} - \frac{1}{m}\sum_{i=1}^{m} \frac{V_1(X_{1i};\hat{\theta})}{1 - 2\lambda(V_1(X_{1i};\hat{\theta}) - \delta_{p_0p_1}^{(1)})} = \Delta_{p_0p_1}. \tag{3.4}$$

The empirical log-likelihood ratio $l(\Delta_{p_0p_1})$ for the difference between two pAUCs is a sum of dependent variables. Hence, we can not directly apply the standard EL theory to derive its asymptotic distribution. However, in the following theorem, we show that $l(\Delta_{p_0p_1})$ follows a scaled χ^2 distribution.

Theorem 3.1. If $\Delta_{p_0p_1}$ is the true value of the difference between two partial AUC's, $\lim \frac{m}{n} = \nu > 0$, then

$$r(\Delta_{p_0p_1})l(\Delta_{p_0p_1}) \longrightarrow \chi_1^2,$$

where

$$\begin{aligned} r(\Delta_{p_0p_1}) &= \frac{(S_{p_0p_1}^{(1)2} + S_{p_0p_1}^{(2)2})}{\sum_{p_0p_1}^2}, \\ S_{p_0p_1}^{(k)2} &= Var[(1 - G_k(X_k;\theta))I(X_k \in (q_{k1}, q_{k0}))],\ k = 1, 2. \end{aligned}$$

Notice that the scale constant $r(\Delta_{p_0p_1})$ is a function of unknown parameter θ and quantiles q_{kl}'s. In order to obtain a good estimate for $r(\Delta_{p_0p_1})$ and avoid possibly poor empirical variance estimate, we can consider using the bootstrap method defined in Chapter 2 to estimate the variances. We propose two hybrid bootstrap and empirical likelihood (HBEL) confidence intervals for the difference between two pAUCs as follows:

1. The first $(1 - \alpha)$ level semi-parametric HBEL (HBELI) confidence interval for $\Delta_{p_0p_1}$ is defined by

$$\{\Delta : \hat{r}(\Delta_{p_0p_1})l(\Delta) \leq \chi_1^2(1 - \alpha)\},$$

where

$$\begin{aligned} \hat{r}(\Delta_{p_0p_1}) &= \frac{\hat{S}_{p_0p_1}^{(1)2} + \hat{S}_{p_0p_1}^{(2)2}}{\sum_{p_0p_1}^{*2}}, \\ \hat{S}_{p_0p_1}^{(k)2} &= \frac{1}{m}\sum_{i=1}^{m}[V_k(X_{ki};\hat{\theta}) - \hat{\delta}_{p_0p_1}^{(k)}]^2,\ k = 1, 2, \end{aligned}$$

and $\chi_1^2(1 - \alpha)$ is the $(1 - \alpha)$-quantile of χ_1^2.

This interval can be found by solving equations (3.2), (3,3), (3.4) and

$$\hat{r}(\Delta_{p_0p_1})l(\Delta) - \chi_1^2(1-\alpha) = 0 \tag{3.5}$$

for the unknown λ, $\delta_{p_0p_1}^{(k)}$ (k=1,2) and $\Delta_{p_0p_1}$. There will be two solutions for $\Delta_{p_0p_1}$: the smaller one is lower bound of the HBELI interval while the bigger one is upper bound of the interval.

2. The second $(1-\alpha)$ level semi-parametric HBEL (HBELII) confidence interval for $\Delta_{p_0p_1}$ is given by

$$\{\Delta : r^*(\Delta_{p_0p_1})l(\Delta) \leq \chi_1^2(1-\alpha)\}$$

where

$$\begin{aligned} r^*(\Delta_{p_0p_1}) &= \frac{\bar{S}_{p_0p_1}^{(1)*} + \bar{S}_{p_0p_1}^{(2)*}}{\sum_{p_0p_1}^{*2}}, \\ \bar{S}_{p_0p_1}^{(k)*} &= \frac{1}{B}\sum_{i=1}^{B} S_{p_0p_1,b}^{*(k)2}, \quad k = 1, 2, \\ S_{p_0p_1,b}^{*(k)2} &= \frac{1}{m}\sum_{i=1}^{m}[V_k(X_{ki,b}^*; \theta_b^*) - \delta_{p_0p_1,b}^{*(k)}]^2, b = 1, \cdots, B, \end{aligned}$$

θ_b^* is the $\hat{\theta}$ and $\delta_{p_0p_1,b}^{*(k)}$ is the $\hat{\delta}_{p_0p_1}^{(k)}$ based on the b-th bootstrap resample from $\{X_1, \ldots, X_m\}$.

Both HBELI and HBELII are approximate confidence intervals for the difference between two pAUCs with coverage probability $1-\alpha$.

Chapter 4

SIMULATION STUDIES

In this chapter, we report series of simulation results for evaluating coverage accuracy and interval length of the NA, BI, BII, HBELI, HBELII intervals proposed in chapter 2 and chapter 3. For each study, we generate 1000 random samples of size m from F for test responses of non-diseased subjects and another 1000 independent random samples of size n from G for test responses of diseased subjects. In theses studies, three interested intervals of FPR under the ROC curves are chosen: $(p_0, p_1) = (0, 0.4), (0, 0.7), (0.05, 0.5)$; meanwhile the difference between two pAUCs, $\Delta_{p_0p_1}$, is chosen to be 0 and 0.2, respectively. For computational simplicity, we randomly pre-selected specific pAUC for the corresponding ROC curve. Under this simulation setting, the parameters in F and G can be obtained by solving the following equations:

$$\delta^{(k)}_{p_0,p_1} = \int_{p_0}^{p_1} R_k(p)dp \text{ with } R_k(p) = 1 - G_k(F_k^{-1}(1-p);\theta), k = 1, 2. \tag{4.1}$$

In the first simulation study, F is chosen to be a two-dimensional normal distribution with mean $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and covariance $\begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$, where r is a pre-selected correlation coefficient. G is chosen to be another two-dimensional normal distribution with mean $\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, and covariance $\begin{bmatrix} 2^2 & r \\ r & 2^2 \end{bmatrix}$, where μ_1 and μ_2 are the solutions to (4.1).

When constructing the normal approximation confidence interval for $\Delta_{p_0p_1}$, ac-

cording to Theorem 2.1, we have to derive the asymptotic covariance matrix of $\sqrt{n}(\hat{\theta} - \theta)$. Shao (2003) showed that, for normal distribution,

$$\sqrt{n}(\hat{\theta} - \theta) \overset{d}{\sim} N_4(0, \Sigma_\theta^2),$$

where Σ_θ^2 can be derived from the Fisher information matrix.

In the second simulation study, the distributions $F(x_1, x_2)$ and $G(y_1, y_2; \theta)$ are chosen to be different bivariate exponential distributions that have exponential distributions as their marginal distributions. Marshall *et al.* (1967) proposed a method to generate bivariate exponential distributions. Depending on the possible correlation between the test results from two diagnostic tests, we use two different procedures to generate the random samples of test responses. First, we choose the correlation as zero. We generate two independent samples from standard exponential distributions, i.e., $X_1 \sim exp(1)$, $X_2 \sim exp(1)$; and two independent samples from exponential distributions with rates λ_1 and λ_2 for $G(y_1, y_2; \theta)$ with $\theta = (\lambda_1, \lambda_2)$, i.e., $Y_1 \sim exp(\lambda_1)$, $Y_2 \sim exp(\lambda_2)$, where the rates λ_1 and λ_2 are solutions to equation (4.1). Second, to generate bivariate exponential random samples with positive correlation, we first generate random samples $U_{ti} \sim exp(0.5), i = 1, \cdots m$, for $t = 1, 2, 3$, $W_{kj} \sim exp(l_k), j = 1, \cdots n$, $k = 1, 2$, where the rates $l_k = \lambda_k - 0.02$, and λ_k's are solutions to (4.1); and $W_{3j} \sim exp(0.02), j = 1, \cdots n$. Thus the simulated test responses for the non-diseased subjects are $X_{ki} = min(U_{ki}, U_{3i})$, $k = 1, 2, i = 1, 2, \ldots, m$, which are random samples from two standard exponential distributions with positive correlation; and those for diseased subjects are $Y_{kj} = min(W_{kj}, W_{3j})$, $k = 1, 2, j = 1, 2, \ldots, n$, which are random samples from two exponential distributions with positive correlation and rates $l_k + 0.02 = \lambda_k$, respectively.

In the bootstrap step of these simulation studies, we draw $B = 150$ bootstrap

resamples with replacement from our generated samples. Various 95% confidence intervals for Δ_{p_0,p_1} are constructed for different combinations of sample size (m, n). The coverage probability and average length of these confidence intervals for Δ_{p_0,p_1} are shown in Tables 1-8.

From Tables 1-8, we can observe that the most of NA-based intervals are the most conservative intervals and have the longest interval length. The HBELI, HBELII, BI and BII intervals have similar coverage probabilities, but the HBELI and HBELII have slightly shorter interval length. When the sample size is small, all intervals overcover Δ_{p_0,p_1}, but the HBELI and HBELII intervals perform slightly better than the others.

Chapter 5

PANCREATIC CANCER SERUM BIOMARKERS EXAMPLE

There are two continuous positive scale serum biomarkers used to diagnose a patient who has pancreatic cancer (Wieand *et al.*, 1989): CA-125, a cancer antigen, and CA-19-9, a carbohydrate antigen. The dataset comes from a case-control study at Mayo Clinic which include 90 patients with pancreatic cancer and 51 subjects with pancreatitis. This dataset have been used by various statisticians for the purpose of diagnostic tests. It has been first used by Wieand *et al.* (1989) to illustrate the non-parametric method for comparing the accuracy of two diagnostic tests. Molodianovitch *et al.* (2006) examined the normality of this data set. They pointed out that for diseased and non-diseased subjects, both biomarkers are not normally distributed. They suggested to apply a Box-Cox type power transformation to the data to reduce the skewness. They shown that, applying the Box-Cox transformation, the CA-125 values be taken to the power -0.424 while CA-19-9 values to the power -0.015, the data is more normal like.

After the Box-Cox transformation, we apply our bootstrap and empirical likelihood-based semi-parametric inference method to test the difference between the pAUCs for biomarkers CA-125 and CA-19-9. Wieand *et al.* (1989) plotted the sensitivity against specificity, and demonstrated that when the specificity falls in (0.8,

1), the difference of the curves is obvious. Since the false positive rate (FPR) equals 1-specificity, we choose to compare the partial areas under the ROC curves over the interval $(0, 0.2)$ of FPR.

Table 9 shows the 95% confidence intervals for the difference between pAUCs on (0, 0.2) for biomarkers CA-125 and CA-19-9. All the four confidence intervals, BI, BII, HBELI and HBELII, demonstrate that CA-19-9 have larger pAUC than CA-125 over $(0, 0.2)$. If we want the specificity of the biomakers to be at least 80%, CA-19-9 will have better diagnostic accuracy than CA-125 in detecting pancreatic cancer. Wieand *et al.* (1989) also reached the same conclusion based on their non-parametric method.

Chapter 6

DISCUSSION

With new tests being developed and marketed, comparing the accuracy of two continuous-scale diagnostic tests are of great importance. Comparing the partial areas under the ROC curves is an effective way to evaluate the accuracy of two diagnostic tests. In medical diagnostic studies, it may not be reasonable to adopt two parametric models for the test results from the diseased and non-diseased subjects. The semi-parametric models may be an useful alternative. In this thesis, we have studied the semi-parametric inferences for the difference between two partial AUCs. We have derived a normal approximation (NA) based confidence interval, two bootstrap (BI and BII) based confidence intervals and two hybrid bootstrap and empirical likelihood (HBELI and HBELII) based confidence intervals. The simulation studies shows that the bootstrap and empirical likelihood based intervals are superior to the NA-based intervals. The Pancreatic Cancer Serum Biomarkers example also suggests that the hybrid bootstrap and empirical likelihood based confidence intervals have better interval estimates.

REFERENCES

[1] Shapiro DE (1999), The interpretation of diagnostic tests, *Statistical Methods in Medical Research*, Vol. 8, pp. 113-134.

[2] DeLong ER, DeLong DM and Clarke-Pearson DL. (1984), Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach., *Biometrics*, The Hague: Martinus Nijhoff, pp. 432-445.

[3] Metz CE, Wang P-L and Kronman HB. (1984), A new approach for testing the significance of differences between ROC curves measured from correlated data., *Information processing in Medical Imaging: Proceedings of the Eighth Conference*, Vol. 1, pp. 5-9.

[4] McClish DK (1989), Analyzing a portion of the ROC curve, *Medical Decision Making*, Vol. 9, pp. 190-195.

[5] Thompson ML, Zuccini W (1989), On the statistical analysis of ROC curves, *Statistics in Medicine*, Vol. 9, pp. 1277-1290.

[6] Jiang Y, Metz CE, Nishikawa RM (1996), A receiver operating characteristic partial area index for highly sensitive diagnostic tests, *Radiology*, Vol. 201, pp. 745-750.

[7] Wieand S, Gail MH, James BR and James KL (1989), A family of nonparametric statistics for comparing diagnostic markers with paired or unpaired data, *Biometrika*, Vol. 76, pp. 585-592.

[8] Mossman D (1995), Resampling techniques in the analysis of non-binormal ROC data, *Medical Decision Making*, Vol. 15, pp. 358-366.

[9] Qin GS, Yuan Y, Zhou XH (2008), Bootstrap and empirical likelihood-based nonparametric inference for the difference between two partial AUCs, *Manuscript*,

[10] Li G, Tiwari RC and Wells MT (1999), Semiparametric inference for a quantile comparison function with applications to receiver operating characteristic curves, *Biometrika*, Vol. 86, pp. 487-502.

[11] Dodd LE and Pepe MS (2003), Partial AUC estimation and regression, *Biometrics*, Vol. 59, pp. 614-623.

[12] Qin J and Zhang B (2003), Using logistic regression procedures for estimating receiver operating characteristic curves, *Biometrika*, Vol. 90, pp. 586-596.

[13] Bamber DC (1975), The area above the ordinal dominance graph and the area below the receiver operating characteristic curve graph, *Journal of Mathematical Psychology*, Vol. 12, pp. 387-415.

[14] Owen A (1990), Empirical likelihood ratio confidence regions, *Annals of Statistics*, Vol. 18, pp. 90-120.

[15] Owen A (2001), *Empirical likelihood*, New York: Chapman and Hall/CRC.

[16] Qin GS and Zhou XH (2006), Empirical likelihood inference for the area under the ROC curve, *Biometrics*, Vol. 62, pp. 613-622.

[17] Shao J (2003), *Mathematical Statistics, second edition*, New York: Springer-Verlag.

[18] Marshall AW and Olkin I (1967), A multivariate exponential distribution, *Journal of the American Statistical Associacian*, Vol. 62, pp. 30-44.

[19] Molodianovitch K, Faraggi D and Reiser B (2006), Comparing the area under two correlated ROC curves: parametric and non-parametric approaches, *Biometrical Journal*, Vol. 48, pp. 745-757.

APPENDIX A: SIMULATION TABLES

Table I: Level of 95 per cent confidence interval for $\Delta_{p_0 p_1} = 0$. Bivariate normal distribution with $\rho = 0$.

True $\Delta_{p_0 p_1}$	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
0	(20, 20)	HBELI	0.956	0.260	0.973	0.387	0.962	0.297
		HBELII	0.964	0.273	0.973	0.387	0.967	0.304
		BI	0.97	0.277	0.968	0.403	0.972	0.309
		BII	0.958	0.277	0.958	0.403	0.953	0.309
		NA	0.982	0.265	0.976	0.453	0.983	0.310
	(50, 50)	HBELI	0.952	0.152	0.962	0.236	0.964	0.169
		HBELII	0.957	0.156	0.962	0.237	0.968	0.171
		BI	0.961	0.157	0.958	0.238	0.968	0.172
		BII	0.955	0.156	0.954	0.238	0.957	0.172
		NA	0.954	0.141	0.97	0.266	0.957	0.165
	(80, 80)	HBELI	0.95	0.117	0.959	0.183	0.961	0.131
		HBELII	0.951	0.119	0.959	0.184	0.962	0.132
		BI	0.955	0.119	0.958	0.184	0.962	0.133
		BII	0.946	0.119	0.956	0.184	0.953	0.133
		NA	0.948	0.106	0.964	0.204	0.951	0.126
	(50, 20)	HBELI	0.94	0.209	0.951	0.324	0.951	0.234
		HBELII	0.95	0.216	0.952	0.327	0.953	0.240
		BI	0.953	0.217	0.95	0.334	0.956	0.242
		BII	0.946	0.217	0.938	0.334	0.949	0.242
		NA	0.97	0.216	0.971	0.407	0.972	0.252
	(80, 50)	HBEL	0.948	0.138	0.95	0.218	0.947	0.156
		EL	0.949	0.141	0.95	0.219	0.947	0.157
		BT	0.952	0.141	0.949	0.220	0.95	0.158
		BS	0.952	0.141	0.947	0.220	0.948	0.158
		NA	0.953	0.131	0.968	0.258	0.965	0.157

Table 2: Level of 95 per cent confidence interval for $\Delta_{p_0p_1} = 0.2$. Bivariate normal distribution with $\rho = 0$.

True $\Delta_{p_0p_1}$	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
0.2	(20, 20)	HBELI	0.979	0.248	0.966	0.365	0.978	0.280
		HBELII	0.977	0.254	0.966	0.362	0.978	0.281
		BI	0.993	0.255	0.973	0.367	0.981	0.282
		BII	0.994	0.255	0.969	0.367	0.965	0.282
		NA	1	0.333	0.998	0.461	0.998	0.344
	(50, 50)	HBELI	0.982	0.132	0.954	0.210	0.976	0.152
		HBELII	0.983	0.134	0.954	0.209	0.976	0.152
		BI	0.983	0.134	0.955	0.210	0.976	0.153
		BII	0.984	0.134	0.956	0.210	0.969	0.153
		NA	0.998	0.167	0.977	0.250	0.989	0.168
	(80, 80)	HBELI	0.973	0.098	0.950	0.161	0.965	0.116
		HBELII	0.975	0.099	0.951	0.161	0.966	0.116
		BI	0.974	0.099	0.957	0.161	0.966	0.116
		BII	0.975	0.099	0.959	0.161	0.963	0.116
		NA	0.986	0.120	0.969	0.188	0.98	0.128
	(50, 20)	HBELI	0.951	0.178	0.944	0.292	0.959	0.207
		HBELII	0.955	0.180	0.944	0.293	0.961	0.208
		BI	0.956	0.180	0.950	0.295	0.961	0.209
		BII	0.954	0.180	0.952	0.295	0.958	0.209
		NA	0.999	0.265	0.992	0.409	0.989	0.266
	(80, 50)	HBELI	0.961	0.115	0.937	0.192	0.954	0.135
		HBELII	0.962	0.116	0.935	0.192	0.957	0.137
		BI	0.960	0.116	0.942	0.193	0.957	0.137
		BII	0.969	0.116	0.94	0.193	0.945	0.137
		NA	0.99	0.150	0.983	0.238	0.979	0.160

Table 3: Level of 95 per cent confidence interval for $\Delta_{p_0p_1} = 0$. Bivariate normal distribution with $\rho = 0.3$.

True $\Delta_{p_0p_1}$	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
0	(50, 50)	HBELI	0.976	0.133	0.967	0.204	0.975	0.148
		HBELII	0.977	0.136	0.967	0.204	0.977	0.150
		BI	0.979	0.137	0.965	0.205	0.978	0.151
		BII	0.976	0.137	0.959	0.205	0.972	0.151
		NA	0.992	0.157	0.994	0.274	0.988	0.179
	(80, 80)	HBELI	0.961	0.102	0.953	0.157	0.966	0.114
		HBELII	0.966	0.103	0.955	0.158	0.967	0.115
		BI	0.968	0.104	0.953	0.158	0.967	0.115
		BII	0.963	0.104	0.948	0.158	0.964	0.115
		NA	0.985	0.118	0.983	0.212	0.992	0.137
	(150, 150)	HBELI	0.958	0.072	0.955	0.112	0.958	0.080
		HBELII	0.960	0.073	0.955	0.112	0.960	0.080
		BI	0.960	0.073	0.955	0.113	0.960	0.080
		BII	0.960	0.073	0.950	0.113	0.957	0.080
		NA	0.978	0.083	0.996	0.152	0.988	0.097
	(80, 50)	HBELI	0.947	0.119	0.957	0.188	0.952	0.134
		HBELII	0.949	0.121	0.958	0.188	0.953	0.136
		BI	0.950	0.121	0.957	0.189	0.954	0.136
		BII	0.945	0.121	0.952	0.189	0.947	0.136
		NA	0.992	0.146	0.987	0.265	0.986	0.171
	(150, 80)	HBELI	0.937	0.090	0.954	0.143	0.944	0.101
		HBELII	0.939	0.091	0.955	0.144	0.945	0.101
		BI	0.940	0.091	0.954	0.144	0.946	0.102
		BII	0.937	0.091	0.948	0.144	0.943	0.102
		NA	0.986	0.112	0.994	0.207	0.989	0.130

Table 4: Level of 95 per cent confidence interval for $\Delta_{p_0p_1} = 0.2$. Bivariate normal distribution with $\rho = 0.3$.

True $\Delta_{p_0p_1}$	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
0.2	(50, 50)	HBELI	0.984	0.124	0.944	0.187	0.975	0.142
		HBELII	0.984	0.125	0.945	0.187	0.975	0.142
		BI	0.985	0.125	0.950	0.187	0.975	0.142
		BII	0.987	0.125	0.947	0.187	0.975	0.142
		NA	0.999	0.160	0.987	0.253	0.992	0.168
	(80, 80)	HBELI	0.970	0.091	0.948	0.144	0.965	0.106
		HBELII	0.972	0.092	0.947	0.143	0.967	0.106
		BI	0.971	0.092	0.951	0.143	0.967	0.107
		BII	0.973	0.092	0.951	0.143	0.954	0.107
		NA	0.998	0.116	0.984	0.192	0.996	0.128
	(150, 150)	HBELI	0.958	0.063	0.946	0.102	0.958	0.073
		HBELII	0.958	0.063	0.946	0.102	0.959	0.073
		BI	0.958	0.063	0.947	0.102	0.959	0.073
		BII	0.966	0.063	0.949	0.102	0.957	0.073
		NA	0.989	0.078	0.982	0.135	0.981	0.086
	(80, 50)	HBELI	0.962	0.106	0.949	0.169	0.962	0.124
		HBELII	0.963	0.107	0.949	0.169	0.965	0.124
		BI	0.961	0.107	0.951	0.170	0.965	0.124
		BII	0.965	0.107	0.951	0.170	0.953	0.124
		NA	0.997	0.141	0.994	0.243	0.998	0.157
	(150, 80)	HBELI	0.955	0.078	0.934	0.129	0.951	0.091
		HBELII	0.955	0.079	0.934	0.129	0.951	0.091
		BI	0.955	0.079	0.940	0.130	0.953	0.092
		BII	0.956	0.079	0.944	0.130	0.950	0.092
		NA	0.993	0.104	0.988	0.185	0.987	0.117

Table 5: Level of 95 per cent confidence interval for $\Delta_{p_0 p_1} = 0$. Bivariate exponential distribution with $\rho = 0$.

True $\Delta_{p_0 p_1}$	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
0	(20, 20)	HBELI	0.971	0.263	0.977	0.376	0.968	0.292
		HBELII	0.974	0.272	0.974	0.370	0.971	0.297
		BI	0.981	0.274	0.973	0.379	0.973	0.300
		BII	0.977	0.274	0.972	0.379	0.964	0.300
		NA	0.989	0.302	0.998	0.590	0.993	0.370
	(50, 50)	HBELI	0.963	0.154	0.965	0.219	0.961	0.168
		HBELII	0.967	0.157	0.964	0.218	0.964	0.169
		BI	0.968	0.157	0.963	0.218	0.966	0.169
		BII	0.967	0.157	0.958	0.218	0.963	0.169
		NA	0.975	0.171	1	0.361	0.995	0.214
	(80, 80)	HBELI	0.962	0.119	0.958	0.167	0.950	0.130
		HBELII	0.964	0.121	0.956	0.167	0.951	0.131
		BI	0.964	0.121	0.956	0.167	0.952	0.131
		BII	0.962	0.121	0.954	0.167	0.951	0.131
		NA	0.965	0.131	1	0.282	0.992	0.168
	(50, 20)	HBELI	0.931	0.191	0.944	0.275	0.951	0.209
		HBELII	0.935	0.194	0.941	0.272	0.952	0.209
		BI	0.938	0.194	0.939	0.276	0.953	0.210
		BII	0.939	0.194	0.943	0.276	0.947	0.210
		NA	0.984	0.256	1	0.559	0.997	0.322
	(80, 50)	HBELI	0.940	0.134	0.954	0.188	0.949	0.146
		HBELII	0.940	0.135	0.953	0.188	0.949	0.147
		BI	0.941	0.136	0.950	0.188	0.949	0.147
		BII	0.943	0.136	0.949	0.188	0.949	0.147
		NA	0.977	0.163	1	0.355	0.993	0.208

Table 6: Level of 95 per cent confidence interval for $\Delta_{p_0p_1}=0.2$. Bivariate exponential distribution with $\rho=0$.

True $\Delta_{p_0p_1}$	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
0.2	(20, 20)	HBELI	0.996	0.240	0.928	0.324	0.991	0.273
		HBELII	0.997	0.243	0.926	0.317	0.991	0.274
		BI	0.997	0.245	0.989	0.329	0.991	0.277
		BII	0.994	0.245	0.988	0.329	0.994	0.277
		NA	1	0.417	1	0.676	1	0.433
	(50, 50)	HBELI	0.978	0.128	0.926	0.187	0.961	0.150
		HBELII	0.978	0.129	0.923	0.185	0.962	0.151
		BI	0.978	0.130	0.961	0.183	0.961	0.151
		BII	0.979	0.130	0.961	0.183	0.973	0.151
		NA	1	0.242	1	0.416	1	0.250
	(80, 80)	HBELI	0.968	0.096	0.942	0.143	0.964	0.115
		HBELII	0.969	0.097	0.942	0.143	0.964	0.115
		BI	0.968	0.097	0.964	0.140	0.964	0.115
		BII	0.972	0.097	0.961	0.140	0.974	0.115
		NA	1	0.186	1	0.326	1	0.200
	(50, 20)	HBELI	0.970	0.154	0.905	0.229	0.956	0.184
		HBELII	0.971	0.155	0.903	0.226	0.954	0.184
		BI	0.970	0.155	0.959	0.227	0.954	0.185
		BII	0.973	0.155	0.955	0.227	0.965	0.185
		NA	1	0.366	1	0.646	1	0.381
	(80, 50)	HBELI	0.961	0.106	0.930	0.158	0.963	0.127
		HBELII	0.961	0.106	0.929	0.157	0.962	0.127
		BI	0.961	0.107	0.948	0.155	0.962	0.127
		BII	0.962	0.107	0.949	0.155	0.973	0.127
		NA	1	0.232	1	0.410	0.999	0.245

Table 7: Level of 95 per cent confidence interval for $\Delta_{p_0p_1}=0$. Bivariate exponential distribution with $\rho>0$.

True $\Delta_{p_0p_1}$	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
0	(50, 50)	HBELI	0.947	0.141	0.970	0.195	0.972	0.152
		HBELII	0.950	0.144	0.970	0.194	0.974	0.153
		BI	0.955	0.144	0.970	0.195	0.974	0.153
		BII	0.952	0.144	0.971	0.195	0.968	0.153
		NA	0.992	0.177	1	0.369	0.998	0.221
	(80, 80)	HBELI	0.971	0.110	0.950	0.150	0.963	0.118
		HBELII	0.974	0.111	0.950	0.150	0.966	0.119
		BI	0.974	0.111	0.947	0.150	0.966	0.119
		BII	0.974	0.111	0.950	0.150	0.962	0.119
		NA	0.984	0.135	1	0.288	0.998	0.171
	(150, 150)	HBELI	0.950	0.078	0.950	0.107	0.955	0.083
		HBELII	0.952	0.079	0.950	0.107	0.955	0.083
		BI	0.953	0.079	0.949	0.107	0.955	0.083
		BII	0.947	0.079	0.944	0.107	0.950	0.084
		NA	0.987	0.095	1	0.208	0.995	0.122
	(80, 50)	HBELI	0.957	0.124	0.946	0.149	0.966	0.135
		HBELII	0.960	0.125	0.943	0.149	0.966	0.135
		BI	0.961	0.125	0.941	0.150	0.967	0.135
		BII	0.960	0.125	0.946	0.150	0.962	0.135
		NA	0.990	0.166	1	0.359	0.995	0.212
	(150, 80)	HBELI	0.944	0.094	0.944	0.111	0.955	0.100
		HBELII	0.944	0.094	0.944	0.111	0.956	0.101
		BI	0.946	0.095	0.944	0.111	0.957	0.101
		BII	0.941	0.095	0.943	0.112	0.952	0.101
		NA	0.988	0.12	1	0.283	1	0.165

Table 8: Level of 95 per cent confidence interval for $\Delta_{p_0p_1} = 0.2$. Bivariate exponential distribution with $\rho > 0$.

True $\Delta_{p_0p_1}$	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
0.2	(50, 50)	HBELI	0.964	0.124	0.937	0.172	0.972	0.152
		HBELII	0.964	0.125	0.937	0.171	0.974	0.153
		BI	0.964	0.125	0.959	0.169	0.974	0.153
		BII	0.974	0.125	0.961	0.169	0.968	0.153
		NA	1	0.249	1	0.425	1	0.257
	(80, 80)	HBELI	0.970	0.093	0.951	0.132	0.963	0.118
		HBELII	0.971	0.094	0.949	0.131	0.966	0.119
		BI	0.970	0.094	0.964	0.129	0.966	0.119
		BII	0.975	0.094	0.967	0.129	0.962	0.119
		NA	1	0.191	1	0.333	1	0.201
	(150, 150)	HBELI	0.945	0.065	0.939	0.093	0.955	0.083
		HBELII	0.946	0.065	0.939	0.093	0.955	0.083
		BI	0.944	0.065	0.946	0.092	0.955	0.084
		BII	0.948	0.065	0.945	0.092	0.950	0.084
		NA	1	0.136	1	0.242	1	0.143
	(80, 50)	HBELI	0.965	0.103	0.934	0.147	0.966	0.135
		HBELII	0.965	0.103	0.933	0.147	0.966	0.135
		BI	0.965	0.103	0.952	0.144	0.967	0.135
		BII	0.969	0.103	0.955	0.144	0.962	0.135
		NA	1	0.237	1	0.419	1	0.249
	(150, 80)	HBELI	0.953	0.076	0.968	0.159	0.955	0.100
		HBELII	0.953	0.076	0.968	0.159	0.956	0.101
		BI	0.953	0.076	0.967	0.158	0.957	0.101
		BII	0.957	0.076	0.965	0.158	0.952	0.101
		NA	1	0.183	1	0.329	1	0.193

Table 9: Pancreatic Cancer Serum Biomarkers Example
Level of 95 percent confidence interval for $\Delta_{(0,0.2)}$

Method	Lower-Limit	Upper-Limit	Length
HBELI	-0.079	-0.004	0.075
HBELII	-0.080	-0.003	0.077
BI	-0.079	-0.002	0.077
BII	-0.084	-0.008	0.076

APPENDIX B: THE S-PLIS CODE FOR SIMULATION

```
###Code for normal simulation studies ###
#######################part 1: Functions#####################
## Function R(p)##
  Rp<-function(p, muy, stdd) 1-pnorm(qnorm(1-p),muy, stdd)
## solveNonlinear##
##nlmin can be used to solve a system of nonlinear equations:
  solveNonlinear <- function(f, y0, x, ...){
  # solve f(x) = y0
  # x is vector of initial guesses, same length as y0
  # ... are additional arguments to nlmin (not to f)
  g <- function(x, y0, f) sum((f(x) - y0)^2)
  g$y0 <- y0   # set g's default value for y0
  g$f <- f     # set g's default value for f
  nlmin(g, x, max.fcal = 10000, max.iter = 10000, ...)
  }
##calculate x[1]=y1.mean x[2]=y2.mean##
mu <- function(x){
c( integrate(Rp, muy=x[1], stdd=y1.sd, lower=p0, upper = p1)$integral,
  integrate(Rp, muy=x[2], stdd=y2.sd, lower=p0, upper = p1)$integral )
}

##function for S_{p_0p_1}^2##

my.mean <- function(vv) mean((vv-mean(vv))^2) ;
##x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta using r.deltap0p1.hat
g2 <- function(x) c( mean((V.hat[,1]-x[1])/(1-2*x[3]*(V.hat[,1]-x[1]))),
                     mean((V.hat[,2]-x[2])/(1+2*x[3]*(V.hat[,2]-x[2]))),
    mean(V.hat[,2]/(1+2*x[3]*(V.hat[,2]-x[2])))-mean(V.hat[,1]/(1-
    2*x[3]*(V.hat[,1]-x[1])))-x[4],
##why we need the absolute value for the LOG function?##
    r.deltap0p1.hat*(2*(sum( log(abs(1-2*x[3]*(V.hat[,1]-x[1]))))+sum(
    log(abs(1+2*x[3]*(V.hat[,2]-x[2])))))-CritVal)
##x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta using r.deltap0p1
g1 <- function(x) c( mean((V.hat[,1]-x[1])/(1-2*x[3]*(V.hat[,1]-x[1]))),
                     mean((V.hat[,2]-x[2])/(1+2*x[3]*(V.hat[,2]-x[2]))),
    mean(V.hat[,2]/(1+2*x[3]*(V.hat[,2]-x[2])))-mean(V.hat[,1]/(1-
    2*x[3]*(V.hat[,1]-x[1])))-x[4],
    r.deltap0p1*(2*(sum( log(abs(1-2*x[3]*(V.hat[,1]-x[1]))))+sum(
    log(abs(1+2*x[3]*(V.hat[,2]-x[2])))))-CritVal)

##function for deltapAUC.hat##
deltapAUC <- function(X1X2, Y1Y2, p0, p1, m){
  # Caculate X Quantile of 1-pi (i=0,1) for q.hat
  q0.1.hat<-quantile(X1X2[,1],1-p0);
  q0.2.hat<-quantile(X1X2[,2],1-p0);
  q1.1.hat<-quantile(X1X2[,1],1-p1);
  q1.2.hat<-quantile(X1X2[,2],1-p1);
   # Caculate V(ki).hat & delta.pAUC.hat
   V.hat<-matrix(, m, 2)
    Y1mean <- mean(Y1Y2[,1])
    Y1sd <- stdev(Y1Y2[,1])
    Y2mean <- mean(Y1Y2[,2])
    Y2sd <- stdev(Y1Y2[,2])

  for (i in 1 : m){
    V.hat[i,1]<-(1-pnorm(X1X2[i,1], mean=Y1mean, sd=Y1sd))*(q1.1.hat <=
    X1X2[i,1])*(X1X2[i,1]<=q0.1.hat)
```

```
   V.hat[i,2]<-(1-pnorm(X1X2[i,2], mean=Y2mean, sd=Y2sd))*(q1.2.hat <=
   X1X2[i,2])*(X1X2[i,2]<=q0.2.hat)
     }
   delta.pAUC.hat<-mean(V.hat[,2])-mean(V.hat[,1])
   r.deltap0p1.hat<-(my.mean(V.hat[,1])+my.mean(V.hat[,2]))/(m*Vstar)
   list(delta.pAUC.hat, r.deltap0p1.hat, V.hat)
 }

##bootstrap function##
booth.trap <- function(B, X1X2, Y1Y2, m, n, p0, p1){
   delta.pAUC<-0;
  sigma <- matrix(,B, 2)
  for (b in 1:B) {
   sampleX.index <- sample(size = m, replace = T, prob = NULL, n = m )
        X1B <- X1X2[sampleX.index,1]
        X2B <- X1X2[sampleX.index,2]

   sampleY.index <- sample(size = n, replace = T, prob = NULL, n = n )
        Y1B <- Y1Y2[sampleY.index,1]
        Y2B <- Y1Y2[sampleY.index,2]
   Y1Bmean <- mean(Y1B)
   Y1Bsd <- stdev(Y1B)
   Y2Bmean <- mean(Y2B)
   Y2Bsd <- stdev(Y2B)

        q0B.1.hat<-quantile(X1B, c(1-p0)) # hatq0, hatq1: sample quantiles of F
        q0B.2.hat<-quantile(X2B, c(1-p0)) # hatq0, hatq1: sample quantiles of F
        q1B.1.hat<-quantile(X1B, c(1-p1))
        q1B.2.hat<-quantile(X2B, c(1-p1))
        VB    <- matrix(,m, 2)

for (i in 1:m)
    {
VB[i,1]<- (1-pnorm(X1B[i], mean=Y1Bmean, sd=Y1Bsd)) *(q1B.1.hat <=
   X1B[i])*(X1B[i] <= q0B.1.hat)
VB[i,2]<- (1-pnorm(X2B[i], mean=Y2Bmean, sd=Y2Bsd)) *(q1B.2.hat <=
   X2B[i])*(X2B[i] <= q0B.2.hat)
    }
       sigma[b,1]<-my.mean(VB[,1])
       sigma[b,2]<-my.mean(VB[,2])
       delta.pAUC[b]<-mean(VB[,2])-mean(VB[,1])
       }
  list(delta.pAUC, sigma)
  }
############################## End function part ####################

############################## Part2: initial value#################
iter<-100
B<-150
rho<-0
#rho<-0.3
m<-50; n<-20;
y1.sd<-2; y2.sd<-2;
i12<-1;
levelc<-0.95
#levelc<-0.90
CritVal<-qchisq(levelc,1)
Z<-qnorm(1-(1-levelc)/2)
y1.mean<-y2.mean<-0
p0<-0 ; p1<-0.4

 pAUC1 <- 0.17
 pAUC2 <- 0.37
```

```
 deltapAUC.true<- pAUC2-pAUC1
 S<-solveNonlinear(mu, c(  pAUC1, pAUC2), c(0.1, 0.1))
  y1.mean<-S$x[1]
  y2.mean<-S$x[2]
#####################  End part2  ########################

#################### Part3:  Loop    ###################
CovCount<-c(0,0,0,0)
CIL<-c(0,0,0)

while ( i12 <= iter ){

# generate non-diseased population F(X1, X2)
# the sample from 2-dimensinal multinormal distribution with mean 0 and std=1
   X1X2<-rmvnorm(m, mean=c(0,0), cov=matrix(c(1,rho,rho,1),2))

# generate  diseased population G(Y1,Y2)
# the sample from 2-dimensinal multinormal distribution with mean
#(y1.mean,y2.mean) and std=(y1.sd,y2.sd)
  Y1Y2<-rmvnorm(n, mean=c(y1.mean,y2.mean),
    cov=matrix(c(y1.sd^2,rho*y1.sd*y2.sd, rho*y1.sd*y2.sd, y2.sd^2),2))

  ##### 1. bootstrap ######
  boot.list<- booth.trap(B, X1X2, Y1Y2, m, n, p0, p1)

  delta.pAUC <- boot.list[[1]]
  sigma <- boot.list[[2]]
  delta.pAUCbar.B<-mean(delta.pAUC); delta.pAUCbar.B  # Estimate mean
    difference of two pAUCs by bootstrap
  Vstar<-var(delta.pAUC);        #Variance of delta.pAUC by bootstrap
  r.deltap0p1<-(mean(sigma[,1])+mean(sigma[,2]))/(m*Vstar);

  #####END OF BOOTSTRAP#######

  ###### 2. Caculate delta.pAUC.hat######
  delta.pAUC.hat.list <- deltapAUC(X1X2, Y1Y2, p0, p1, m)
  delta.pAUC.hat <- delta.pAUC.hat.list[[1]]
  r.deltap0p1.hat <- delta.pAUC.hat.list[[2]]
  V.hat <- delta.pAUC.hat.list[[3]]
  ########END OF 2.  #########


###### 3. Caculate C.I and coverage######
## compute the HBEL interval(Vel from bootstrap)##

     #x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta

       bd<-solveNonlinear(g1, c( 0,0,0,0), c(0.3, 0.1, 0.001, 0.08))
       b<-solveNonlinear(g1, c( 0,0,0,0), c(0.1, 0.3, 0.001, 0.28))
if (bd$x[4] < b$x[4]) {low.HBEL<-bd$x[4];up.HBEL<-b$x[4]} else {low.HBEL<-
   b$x[4];up.HBEL<-bd$x[4]};
if (abs(up.HBEL-low.HBEL)<0.01) next;
## compute the EL interval(Vel.hat)##
 #x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta

     lw<-solveNonlinear(g2, c( 0,0,0,0), c(0.3, 0.1, 0.001, 0.08))
     upb<-solveNonlinear(g2, c( 0,0,0,0), c(0.1, 0.3, 0.001, 0.28))
if (lw$x[4] < upb$x[4]) {low.EL<-lw$x[4];up.EL<-upb$x[4]} else {low.EL<-
   upb$x[4];up.EL<-lw$x[4]};
if (abs(up.EL-low.EL)<0.01) next;

###compute the CI and coverage for HBEL and EL###
```

```
      CIL[1]<- CIL[1] + (up.HBEL- low.HBEL)
      CIL[2]<- CIL[2] + (up.EL - low.EL);
  if ((low.HBEL<= deltapAUC.true) & (up.HBEL >= deltapAUC.true)) CovCount[1]<-
   CovCount[1]+1;
record[i12,1]<-low.HBEL;
record[i12,2]<-up.HBEL;
  if ((low.EL <= deltapAUC.true) & (up.EL >= deltapAUC.true)) CovCount[2]<-
   CovCount[2]+1;
## compute the BTI interval.
   hwidth<-Z*sqrt(Vstar)
      #tlow<- delta.pAUC.hat-hwidth           # lower limit of the CI
      #tup<- delta.pAUC.hat+hwidth            # upper limit of the CI
     if (((delta.pAUC.hat-hwidth)<= deltapAUC.true) & ((delta.pAUC.hat+hwidth)
   >= deltapAUC.true)) CovCount[3]<-CovCount[3]+1
      #low and up band
       #LP[3]<-LP[3]+(delta.pAUC.hat-hwidth)
      #UP[3]<-UP[3]+(delta.pAUC.hat+hwidth)
       CIL[3]<- CIL[3]+2*hwidth               # The length of BT and BS CI

## compute the bootstrap(BS) interval
      #bslow<- delta.pAUCbar.B-hwidth         # lower limit of the CI
      #bsup<- delta.pAUCbar.B+hwidth           # upper limit of the CI
       if (((delta.pAUCbar.B-hwidth) <= deltapAUC.true) &
   ((delta.pAUCbar.B+hwidth)>= deltapAUC.true)) CovCount[4]<-CovCount[4]+1
      #low and up band
      #LP[4]<-LP[4]+(delta.pAUCbar.B-hwidth)
      #UP[4]<-UP[4]+(delta.pAUCbar.B+hwidth)
i12 <- i12+1;
}
#End of LOOP##

 cov<-CovCount/iter; cov
  wid<-CIL/iter;wid
#Result Output
sink("C:\\Temp\\semipAUC.txt",append = T)

cat("iter=", iter,"At level=", levelc, "m=", m, "n=",
   n,"rho=",rho,"Delta=",deltapAUC.true, "p0=", p0, "p1=", p1, "\n")
cat("mean1=",y1.mean,"mean2=", y2.mean,"y1std=", y1.sd, "y2std=", y2.sd, "B=",
   B, "\n")

cat("Coverage of the (HBEL, EL, BT, BS) CI's for delta :", cov, "\n")
cat("Average length of (HBEL,EL,BTI&BS):", wid, "\n")

cat("-------------------------------------------------------------------------
   ------------------------------","\n")

sink();
```

```
##### Code for Normal approximation CI for Normal simulation #####
##### One more function is added to calculate the NA variance: ############
##function for normal approximation variance ##

normalApr <- function(X1X2, Y1Y2, p0, p1, m, n, V.hat, rho) {
  # Caculate X Quantile of 1-pi (i=0,1) for q.hat
  q0.1.hat<-quantile(X1X2[,1],1-p0);
  q0.2.hat<-quantile(X1X2[,2],1-p0);
  q1.1.hat<-quantile(X1X2[,1],1-p1);
  q1.2.hat<-quantile(X1X2[,2],1-p1);
   # Caculate V(ki).hat & delta.pAUC.hat
   Bv1.hat<-matrix(,m,1)
   Bv2.hat<-matrix(,m,1)
   Dv.hat<-matrix(,m,4)
   Y1mean <- mean(Y1Y2[,1])
   Y1sd <- stdev(Y1Y2[,1])
   Y2mean <- mean(Y1Y2[,2])
   Y2sd <- stdev(Y1Y2[,2])
   meanV1.hat <- mean(V.hat[,1])
   meanV2.hat <- mean(V.hat[,2])
   for (i in 1:m){
      Bv1.hat[i] <-  (1-pnorm(X1X2[i,1], mean=Y1mean, sd=Y1sd))*(q1.1.hat <=
   X1X2[i,1])*(X1X2[i,1]<=q0.1.hat) - meanV1.hat - (1-pnorm(q0.1.hat,
   mean=Y1mean, sd=Y1sd))*((X1X2[i,1]<=q0.1.hat)-(1-p0)) - (1-pnorm(q1.1.hat,
   mean=Y1mean, sd=Y1sd))*((X1X2[i,1]<=q1.1.hat)-(1-p1))
      Bv2.hat[i] <-  (1-pnorm(X1X2[i,2], mean=Y2mean, sd=Y2sd))*(q1.2.hat <=
   X1X2[i,2])*(X1X2[i,2]<=q0.2.hat) - meanV2.hat - (1-pnorm(q0.2.hat,
   mean=Y2mean, sd=Y2sd))*((X1X2[i,2]<=q0.2.hat)-(1-p0)) - (1-pnorm(q1.2.hat,
   mean=Y2mean, sd=Y2sd))*((X1X2[i,2]<=q1.2.hat)-(1-p1))
      Dv.hat[i,1] <- -dnorm(X1X2[i,1], mean=Y1mean, sd=Y1sd)*(q1.1.hat <=
   X1X2[i,1])*(X1X2[i,1]<=q0.1.hat)
      Dv.hat[i,2] <- ((X1X2[i,1]-Y1mean)/Y1sd)*Dv.hat[i,1]
      Dv.hat[i,3] <- -dnorm(X1X2[i,2], mean=Y2mean, sd=Y2sd)*(q1.2.hat <=
   X1X2[i,2])*(X1X2[i,2]<=q0.2.hat)
      Dv.hat[i,4] <- ((X1X2[i,2]-Y2mean)/Y2sd)*Dv.hat[i,2]
   }
   B.hat <- Bv1.hat + Bv2.hat
   VarB <- var(B.hat)
   D.hat <-
   matrix(c(mean(Dv.hat[,1]),mean(Dv.hat[,2]),mean(Dv.hat[,3]),mean(Dv.hat[,4])
   ),1)
   sigmaY1=matrix(c(Y1sd,0,0,sqrt(2)*Y1sd^2),2)
   sigmaY2=matrix(c(Y2sd,0,0,sqrt(2)*Y2sd^2),2)
   i1 <- matrix(c(1,0,0,0),2)

   i2 <- matrix(c(0,1,1,0),2)
   i3 <- matrix(c(0,0,0,1),2)
   l1 <- kronecker(i1, sigmaY1^2)
   l2 <- kronecker(i2, rho*sigmaY1%*%sigmaY2)
   l3 <- kronecker(i3, sigmaY2^2)
   sigma.theta <- l1+l2+l3
   sigma.p0p1 <- VarB + m/n*D.hat%*%sigma.theta%*%t(D.hat)
   list(sigma.p0p1)
}


############################ Part2: initial value##################
iter<-100
B=150
rho=0
#rho=0.3
m<-50; n<-20;
```

```
y1.sd<-2; y2.sd<-2;

levelc<-0.95
#levelc<-0.90
Z<-qnorm(1-(1-levelc)/2)
y1.mean<-y2.mean<-0
p0<-0 ; p1<-0.4
pAUC1 <- 0.2
pAUC2 <- 0.2
 deltapAUC.true<- pAUC2-pAUC1
S<-solveNonlinear(mu, c(  pAUC1, pAUC2), c(0.1, 0.1))
  y1.mean<-S$x[1]
  y2.mean<-S$x[2]
####################  End part2  ######################
#################### Part3:  Loop    ###################

CovCount<-0
CIL<-0
for ( i12 in c(1:iter)){
# generate non-diseased population F(X1, X2)
# the sample from 2-dimensinal multinormal distribution with mean 0 and std=1

   X1X2<-rmvnorm(m, mean=c(0,0), cov=matrix(c(1,rho,rho,1),2))
# generate  diseased population G(Y1,Y2)
# the sample from 2-dimensinal multinormal distribution with mean
#(y1.mean,y2.mean) and std=(y1.sd,y2.sd)

  Y1Y2<-rmvnorm(n, mean=c(y1.mean,y2.mean),
   cov=matrix(c(y1.sd^2,rho*y1.sd*y2.sd, rho*y1.sd*y2.sd, y2.sd^2),2))
###### 2. Caculate delta.pAUC.hat######
delta.pAUC.hat.list <- deltapAUC(X1X2, Y1Y2, p0, p1, m)
delta.pAUC.hat <- delta.pAUC.hat.list[[1]]
  V.hat <- delta.pAUC.hat.list[[2]]
########END OF 2.  #########
####### 2.5 calculate sigma for delta.pAUC.hat #####

#  sqrt.rho.y1y2 <- sqrt(rho*y1.sd*y2.sd)
#  Ycov <- matrix(c(y1.sd,sqrt.rho.y1y2, sqrt.rho.y1y2, y2.sd),2)
   sigma.delta.pAUC.hat.list <- normalApr(X1X2, Y1Y2, p0, p1, m, n, V.hat, rho)
   sigma.normalApr <- sigma.delta.pAUC.hat.list[[1]]
########END OF 2.5 #######

###### 4. Caculate C.I and coverage######
   aprwidth <- Z*sqrt(sigma.normalApr)/sqrt(m)
   if (((delta.pAUC.hat-aprwidth) <= deltapAUC.true) &&
   ((delta.pAUC.hat+aprwidth) >= deltapAUC.true)) CovCount<-CovCount+1
   CIL<- CIL+2*aprwidth
}

#End of LOOP##
cov<-CovCount/iter
  wid<-CIL/iter
#Result Output
sink("C:\\Temp\\semipAUC.txt",append = T)
cat("iter=", iter,"At level=", levelc, "m=", m, "n=",
   n,"rho=",rho,"Delta=",deltapAUC.true, "p0=", p0, "p1=", p1, "\n")
cat("mean1=",y1.mean,"mean2=", y2.mean,"y1std=", y1.sd, "y2std=", y2.sd, "B=",
   B, "\n")
cat("Coverage of the normal approximation CI's for delta :", cov, "\n")
cat("Average length of normal approximation:", wid, "\n")
cat("-------------------------------------------------------------------------
   ------------------------------","\n")
sink();
```

```
####Code for exponential simulation study #####
#### The function part is the same as normal case, except all the normal
   related function have been changed into exponential distribution #####
#### The generating part for no covariance case is the same as normal case,
   except change normal function into exponential function ###
#### Here just show how to generate the bivariate exponential function with
   covariance > 0 ####

# generate non-diseased population F(X1, X2)
# the sample from bivariate exponential distribution with rate=1
   u1<-rexp(m, rate=explambda)
   u2<-rexp(m, rate=explambda)
   u3<-rexp(m, rate=explambda)
   for (k in 1:m){
      X1X2[k,1]<-min(u1[k], u3[k])   #Exp(1): the sample from the first non-
   disease pop.
      X1X2[k,2]<-min(u2[k], u3[k])   #Exp(1): the sample from the second non-
   disease pop.
   }
# generate  diseased population G(Y1,Y2)
# the sample from bivariate exponential distribution with rate
#(y1.mean,y2.mean)
   v1<-rexp(n, rate=y1.mean)
   v2<-rexp(n, rate=y2.mean)
   v3<-rexp(n, rate=expcov)
```

```
### The normal approximation function for exponential case ###
normalApr <- function(X1X2, Y1Y2, p0, p1, m, n, V.hat, rho) {

  # Caculate X Quantile of 1-pi (i=0,1) for q.hat
  q0.1.hat<-quantile(X1X2[,1],1-p0);
  q0.2.hat<-quantile(X1X2[,2],1-p0);
  q1.1.hat<-quantile(X1X2[,1],1-p1);
  q1.2.hat<-quantile(X1X2[,2],1-p1);

# Caculate V(ki).hat & delta.pAUC.hat
   Bv1.hat<-matrix(,m,1)
   Bv2.hat<-matrix(,m,1)
   Dv.hat<-matrix(,m,2)
   sigma.theta <- matrix(,2,2)
   Y1mean <- 1/mean(Y1Y2[,1])
   Y2mean <- 1/mean(Y1Y2[,2])
   meanV1.hat <- mean(V.hat[,1])
   meanV2.hat <- mean(V.hat[,2])
   for (i in 1:m){
      Bv1.hat[i] <-  (1-pexp(X1X2[i,1], rate=Y1mean))*(q1.1.hat <=
X1X2[i,1])*(X1X2[i,1]<=q0.1.hat) - meanV1.hat - (1-pexp(q0.1.hat,
rate=Y1mean))*((X1X2[i,1]<=q0.1.hat)-(1-p0)) - (1-pexp(q1.1.hat,
rate=Y1mean))*((X1X2[i,1]<=q1.1.hat)-(1-p1))
      Bv2.hat[i] <-  (1-pexp(X1X2[i,2], rate=Y2mean))*(q1.2.hat <=
X1X2[i,2])*(X1X2[i,2]<=q0.2.hat) - meanV2.hat - (1-pexp(q0.2.hat,
rate=Y2mean))*((X1X2[i,2]<=q0.2.hat)-(1-p0)) - (1-pexp(q1.2.hat,
rate=Y2mean))*((X1X2[i,2]<=q1.2.hat)-(1-p1))
      Dv.hat[i,1] <- dexp(X1X2[i,1], rate=Y1mean)*(q1.1.hat <=
X1X2[i,1])*(X1X2[i,1]<=q0.1.hat)
      Dv.hat[i,2] <- dexp(X1X2[i,2], rate=Y2mean)*(q1.2.hat <=
X1X2[i,2])*(X1X2[i,2]<=q0.2.hat)
   }

   B.hat <- Bv1.hat + Bv2.ha
   VarB <- var(B.hat)
   D.hat <- matrix(c(mean(Dv.hat[,1]),mean(Dv.hat[,2])),1)
   sigma.theta[1,1] <- (1/Y1mean)^2
        sigma.theta[2,2] <- (1/Y2mean)^2
   sigma.theta[1,2] <- sigma.theta[2,1] <- rho*(1/Y1mean)*(1/Y2mean)
   sigma.p0p1 <- VarB + m/n*D.hat%*%sigma.theta%*%t(D.hat)
   list(sigma.p0p1)
}
```

```
######The Splus code for real data analysis#####

############################ Part2: data input ##################
coln<-c("T1", "T2","D")
realdata<-read.table("C:\\Temp\\wiedat2b.txt", sep=',', col.names=coln,
   header=T)
realdata
m=51;
n=90;
X1X2 <- matrix(nrow=51, ncol=2);
Y1Y2 <- matrix(nrow=90, ncol=2);

########################################
######  Box-Cox transformation  ########
########################################

X1X2[,1]<-(realdata$T1[realdata$D==0]^(-0.015)-1)/-0.015;

X1X2[,2]<-(realdata$T2[realdata$D==0]^(-0.424)-1)/-0.424;

Y1Y2[,1]<-(realdata$T1[realdata$D==1]^(-0.015)-1)/-0.015;

Y1Y2[,2]<-(realdata$T2[realdata$D==1]^(-0.424)-1)/-0.424;



B=500
levelc<-0.9;
CritVal<-qchisq(levelc,1)
Z<-qnorm(1-(1-levelc)/2)
p0<-0 ; p1<-0.2

####################  End part2  ######################


#################### Part3:  real example    ###################

  ##### 1. bootstrap ######
  boot.list<- booth.trap(B, X1X2, Y1Y2, m, n, p0, p1)
  delta.pAUC <- boot.list[[1]]
  sigma <- boot.list[[2]]
  delta.pAUCbar.B<-mean(delta.pAUC); delta.pAUCbar.B  # Estimate mean
   difference of two pAUCs by bootstrap
  Vstar<-var(delta.pAUC);        #Variance of delta.pAUC by bootstrap
  r.deltap0p1<-(mean(sigma[,1])+mean(sigma[,2]))/(m*Vstar);
  #####END OF BOOTSTRAP#######
  ###### 2. Caculate delta.pAUC.hat######
  delta.pAUC.hat.list <- deltapAUC(X1X2, Y1Y2, p0, p1, m)
  delta.pAUC.hat <- delta.pAUC.hat.list[[1]]
  r.deltap0p1.hat <- delta.pAUC.hat.list[[2]]
  V.hat <- delta.pAUC.hat.list[[3]]
  ########END OF 2.  #########

###### 4. Caculate C.I and coverage######

## compute the HBEL interval(Vel from bootstrap)##

     #x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta

       bd<-solveNonlinear(g1, c( 0,0,0,0), c(0.3, 0.1, 0.001, -0.999))
      low.HBEL<-bd$x[4]  # lower limit of the CI
      b<-solveNonlinear(g1, c( 0,0,0,0), c(0.1, 0.3, 0.001, 0.999))
      up.HBEL<-b$x[4]    # upper limit of the CI
```

```
## compute the EL interval(Vel.hat)##
      #x[1]: p0p1.1  x[2]: p0p1.2  x[3]: lamda x[4]: delta

     lw<-solveNonlinear(g2, c( 0,0,0,0), c(0.3, 0.1, 0.001, -0.999))
     low.EL<-lw$x[4]      # lower limit of the CI
     upb<-solveNonlinear(g2, c( 0,0,0,0), c(0.1, 0.3, 0.001, 0.999))
     up.EL<-upb$x[4]        # upper limit of the CI



## compute the BTI interval.
   hwidth<-Z*sqrt(Vstar)
      tlow<- delta.pAUC.hat-hwidth            # lower limit of the CI
      tup<- delta.pAUC.hat+hwidth          # upper limit of the CI

## compute the bootstrap(BS) interval
      bslow<- delta.pAUCbar.B-hwidth       # lower limit of the CI
      bsup<- delta.pAUCbar.B+hwidth            # upper limit of the CI

#Result Output
sink("C:\\Temp\\real.txt",append = T)
cat("B=", B,"At level=", levelc, "m=", m, "n=", n, "p0=", p0, "p1=", p1, "\n")
cat("The difference between two pAUCs are:", delta.pAUC.hat,"\n")
cat("Confidence Interval of the HBEL for delta is :",low.HBEL, "to",
   up.HBEL,"\n")
cat("Confidence Interval of the EL for delta is :",low.EL, "to", up.EL,"\n")
cat("Confidence Interval of the BTI for delta is :",tlow, "to", tup,"\n")
cat("Confidence Interval of the BS for delta is :",bslow, "to", bsup,"\n")
cat("-------------------------------------------------------------------------
   ------------------------------","\n")
sink();
```