A New Class of Depth-based Statistics with Same Attractor

by

Yiting Chen

B.Sc., University of California, Santa Barbara, 2021

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2024

Declaration of Committee

Name:	**Yiting Chen**
Degree:	**Master of Science**
Thesis title:	**A New Class of Depth-based Statistics with Same Attractor**
Committee:	**Chair:** Haolun Shi Assistant Professor, Statistics and Actuarial Science
	Xiaoping Shi Co-Supervisor Assistant Professor, Department of Computer Science, Mathematics, Physics and Statistics University of British Columbia
	Wei (Becky) Lin Co-Supervisor Lecturer, Statistics and Actuarial Science
	Liangliang Wang Committee Member Associate Professor, Statistics and Actuarial Science
	Richard Lockhart Examiner Professor, Statistics and Actuarial Science

Abstract

Data depth has emerged as an invaluable nonparametric measure for the ranking of multivariate samples. The main contribution of depth-based two-sample comparisons is the introduction of the Q statistic [31], a quality index. Unlike traditional methods, data depth does not require the assumption of normality distributions and adheres to four fundamental properties: affine invariance, maximality at the center, monotonicity relative to the deepest point, and vanishing at infinity [40, 31]. Many existing two-sample homogeneity tests, which assess mean and/or scale changes in distributions often suffer from low statistical power or indeterminate asymptotic distributions. To overcome these challenges, we have introduced three innovative depth-based test statistics. Notably, two of these statistics share a 'common attractor' and are applicable across all depth functions. Our approach extends the concept of same attractive depth functions, rooted in Q statistics, to include both sum and product statistics. We further proved the asymptotic distribution of these statistics for one-dimensional cases under Euclidean depth, in addition to the minimum statistics valid for all depths. Our proof has been extended to the multidimensional case for all depths. These proposed statistics use three depth functions: Mahalanobis depth [31], Spatial depth [7, 20], and Projection depth [29]. all of which are implemented in the R package *ddalpha*. Through two-sample simulations, we have demonstrated that our sum and product statistics exhibit superior power performance, utilizing a strategized permutation algorithm and standing up to comparison with popular methods in literature. Our tests are further validated through analysis on spectrum data, highlighting the effectiveness of the proposed tests.

Keywords: Non-parametric tests; data depth; two-sample test; hypothesis test

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Becky Wei Lin and Dr. Xiaoping Shi, for their constant support, encouragement, and guidance during the time of study. Besides, I extend my appreciation to my committee members, Dr. Richard Lockhart, Dr. Liangliang Wang, and Dr. Haolun Shi, especially for the insightful suggestions generously provided by Dr. Richard Lockhart for my thesis.

I am deeply grateful for Dr. Xiaoping Shi's insightful feedback and invaluable guidance throughout my research. His constant encouragement is always a source of inspiration and motivation. I want to express my sincere gratitude for the tremendous effort he invested in my research and providing abundant resources to explore many topics. I will never forget those days and nights we spent on discussing the challenging problem. It would be impossible to complete the thesis successfully and smoothly without him.

I am also thankful to Dr. Becky Wei Lin for the invaluable support she provided in various aspects, in both academics and life. She invested tremendous efforts and time in reviewing and revising my thesis, coupled with providing valuable feedback. Beyond her help in research, I am particularly grateful for her belief in my capabilities and consistent encouragement for many times. I was deeply impressed and inspired by her passion for statistics and positive attitude towards life.

It is a journey of growth. Life was changed by several decisions, though it was perceived as ordinary at that moment.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction to Data Depth

1.1 Study Background

In recent years, multivariate statistical analysis has been widely applied in many fields. One significant application is the discrimination between two groups of spectra with potential tumor samples, a topic extensively covered in Chapter 4. As we encounter diverse types of data, these analytical challenges are often formulated as homogeneity tests for two multivariate samples with distributions F and G, i.e. testing

$$H_0 : F = G \text{ vs } H_1 : F \neq G. \tag{1.1}$$

The homogeneity tests determine whether the two samples or distributions are statistically the same with some essential parameters. Among the well-known homogeneity tests is Multivariate Analysis of Variance (MANOVA) [17], a parametric test that extends the univariate ANOVA for multivariate sample comparison. Initially introduced by Fisher, MANOVA requires prior assumptions of normality, which often do not hold in real-world data scenarios where distributions may not be well-defined. This limitation underscores the importance of non-parametric tests, such as the Cramér test [1], Wilcoxon Rank-Sum test [47], and Energy Distance test [43], which are not constrained by the assumption of normality.

The detailed formulations of homogeneity test including MANOVA will be elaborated upon subsequently.

Multivariate analysis of variance (MANOVA)

Consider a MANOVA model in a two-sample case structured as follows:

$$X_{i,j} = \mu_i + e_{i,j}, i = 1, 2, \text{ and } j = 1, 2, \ldots, n_i,$$

where μ_i is the mean of i-th population, and $e_{i,j}$ are independent p-dimensional normal variables with mean $\mathbf{0}$ and covariance matrix denoted as Σ, i.e. $e_{i,j} \sim N_p(\mathbf{0}, \Sigma)$.

In MANOVA, the total sum of squares (S_T) is decomposed into the between-group sum squares (S_B) and within-group sum of squares (S_W) such that $S_T = S_B + S_W$. Here S_T is total variability around the overall mean, calculated as $S_T = \sum_{i=1}^{2}\sum_{j=1}^{n_i}(X_{i,j} - \bar{X})(X_{i,j} - \bar{X})^T$, where $\bar{X}$ is the overall sample mean. The between-group sum of squares, S_B measures the variability of the group means around the overall mean and is given by $S_B = \sum_{i=1}^{2}\sum_{j=1}^{n_i}(\bar{X}_i - \bar{X})(\bar{X}_i - \bar{X})^T$, with $\bar{X}_i$ denoting mean of the i-th sample. The within-group sum of squares, S_W, quantifies the variability within each group and is defined as $S_W = \sum_{i=1}^{2}\sum_{j=1}^{n_i}(X_{i,j} - \bar{X}_i)(X_{i,j} - \bar{X}_i)^T$.

Upon calculating the eigenvalues of $S_B^{-1}S_W$, denoted as $\lambda_1, \ldots, \lambda_p$, and with a common covariance matrix, several test statistics can be derived for the MANOVA homogeneity test, i.e.,

$$H_0 : \mu_1 = \mu_2$$

- Hotelling's test statistic [23, 39], denoted as H, is the sum of these eigenvalues, $H = \sum_{i=1}^{p}\lambda_i$. The scaled version of H follows an F-distribution:

$$\left(\frac{n_1 + n_2 - p - 1}{p}\right) H \sim F_{p,n_1+n_2-p-1}.$$

- Wilks' test statistic [48, 39], W, is the product of the reciprocal of each eigenvalue plus one, $W = \prod_{i=1}^{p} 1/(1+\lambda_i)$. A lower value of W suggests disparity between the groups. The statistic follows an F-distribution:

$$\frac{1-W}{W}\left(\frac{n_1 + n_2 - p - 1}{p}\right) \sim F_{p,n_1+n_2-p-1}.$$

- Pillai's trace [38, 39], T, is the sum of the ratios of each eigenvalue to one plus that eigenvalue, $T = \sum_{i=1}^{n}\lambda_i/(1+\lambda_i)$, which also adheres to an F distribution:

$$\frac{T}{1-T}\left(\frac{n_1 + n_2 - p - 1}{p}\right) \sim F_{p,n_1+n_2-p-1}.$$

Cramér test

First introduced by [1], the Cramér test is a non-parametric method for assessing the disparity between two continuous distributions, F and G, without reliance on any presumptive distributional forms. This test is grounded in the comparison of theoretical and empirical distributions, specifically evaluating the integral $\int_{-\infty}^{\infty}[F_n(x) - F(x)]^2\, dF(x)$ for a sample of size n with observations x_i from distribution F, where $i = 1, \ldots, n$. here, $F_n(x)$ denotes the empirical distribution, and $F(x)$ is the corresponding theoretical distribution.

Consider two univariate independent samples $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$, with X containing m independent and identically distributed (i.i.d.) random variables from

distribution F, and Y comprising n i.i.d. variables from distribution G, we define their empirical distributions as $F_m(t)$ and $G_n(t)$, respectively.

To test whether these two samples are drawn from the same distribution, i.e. $H_0 : F = G$, the Cramér test statistic, T, is formulated as

$$T = \frac{mn}{m+n} \int_{-\infty}^{\infty} (F_m(t) - G_n(t))^2 \, d\hat{H}(t),$$

where $\hat{H}(t) = (mF_m(t)) + nG_n(t))/(m+n)$.

With a significance level α, the decision criterion is based on the upper α quantile of T, denoted by T_α. If the calculated T statistic is greater than or equal to T_α, i.e. $T \geq T_\alpha$, the null hypothesis is rejected, suggesting a significant difference between the two distributions.

Wilcoxon Rank-Sum test

The Wilcoxon Rank-Sum test, also known as the Mann-Whitney U test, was originally proposed by Wilcoxon [47]. It is a popular non-parametric test for univariate data. The test evaluates the difference between two samples by collectively ranking all data points and then comparing these ranks between the groups.

Given two univariate independent distributions F and G, from which the random samples $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ are drawn, the Wilcoxon Rank-Sum test can be expressed as the sum of indicator functions:

$$U = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I(x_i < y_j), \tag{1.2}$$

where $I(\cdot)$ is an indicator function that takes 1 if the condition within is true, and 0 otherwise. Under null hypothesis that two samples are from the same distribution, the expected value of U is $\frac{1}{2}$.

A comparative analysis of statistical power between our proposed test statistics and the univariate Wilcoxon Rank-Sum test is presented in Chapter 3. Additionally, a recent extension of Wilcoxon Rank-Sum test to multivariate data is introduced by [28].

Energy Distance test

The energy distance test [43, 42] quantifies the statistical distances between two distributions. Consider two independent random variables $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$, with cumulative distribution functions F and G respectively, and let X' and Y' be independent and identically distributed (i.i.d.) copies of X and Y. Within a Euclidean space, the energy distance [25, 43] can be written as the mean of the pairwise distances between two samples, i.e.,

$$D^2 = 2E||X - Y|| - E||X - X'|| - E||Y - Y'||.$$

The energy-statistic is defined as

$$E_{n,m}(X,Y) = \frac{mn}{m+n}\left\{2\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}||x_i - y_j|| - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}||x_i - x_j|| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}||y_i - y_j||\right\}.$$

The generalized form of the energy distance for any metric space is expressed as:

$$D^2 = 2E[d(X,Y)] - E[d(X,X')] - E[d(Y,Y')],$$

where $d(X,Y)$ represents the distance in a metric space.

When assessing whether two random variables X and Y are from the same distribution, the hypothesis test is framed as $H_0 : F = G$ versus $H_1 : F \neq G$. Under H_0, the $\frac{mn}{m+n}D^2$ will converge to zero, while under H_1, it tends towards infinity. Thus, the coefficient H is formulated as

$$H = \frac{2E||X-Y|| - E||X-X'|| - E||Y-Y'||}{2E||X-Y||},$$

with $0 \leq H \leq 1$.

With $H = 0$ when F and G are identical, the test is performed by determining the upper α quantile c_α as the critical value [42], where $P(E_{n,m}(X,Y) < c_\alpha) = 1 - \alpha$.

1.2 Types of Data Depth

Traditional non-parametric methods are mostly suitable for univariate data, posing challenges when extended to multivariate contexts. Data depth is a method that addresses this by determining differences between two samples without prior normality assumptions of distributions and by providing data rankings. It adheres to four fundamental properties outlined by Serfling [40]: affine invariance, maximality at the center, monotonicity relative to the deepest point, and vanishing at infinity. Data depth $D(x;F)$ measures the centrality of the x point in the distribution $F(x)$ in d-dimensional space, mapping x from R^d to the interval $[0,1]$. These four properties in detail are:

1. Affine invariance: Depth is invariant to coordinate system or scale transformations.

2. Maximality at the center: The center of a distribution has the highest depth value, denoted as $D(\mu,F)$, where μ is the center.

3. Monotonicity relative to the deepest point: For any $0 < \alpha < 1$, the depth decreases as a point x moves away from the center μ, i.e. $D(x,F) < D(\mu + \alpha(x-\mu),F)$.

4. Vanishing at infinity: The depth function approaches zero as the $||x||$ goes to infinity.

Various data depth functions can be used, including Euclidean depth [31, 26], Mahalanobis depth [31], L_p depth [40], Spatial depth [7, 20], Projection depth [29], and Tukey depth [44].

Euclidean Depth

The most simplest depth function is the Euclidean depth for a point x in a one-dimensional distribution F, defined as

$$D(x; \hat{F}) = \frac{1}{1 + (x - \mu)^2}, \tag{1.3}$$

where μ is mean of distribution F [31, 26]. This computation is implemented in R package *DepthProc* using `depthEuclid()`.

Mahalanobis Depth

For any point x in R-dimensional distribution F, Mahalanobis depth [31] is :

$$D(x; \hat{F}) = \frac{1}{1 + (x - \mu)^T \Sigma^{-1} (x - \mu)},$$

with μ as the mean and Σ as the covariance matrix. Note that $(x - \mu)^T \Sigma^{-1} (x - \mu)$ is Mahalanobis distance. The computation of Mahalanobis depth [31] is available in the R package *ddalpha* with `depth.Mahalanobis()`.

L_p Depth

Consider a random variable X from distribution F, the L_p depth [40] for a point x with respect to distribution F is:

$$D(x; \hat{F}) = \frac{1}{1 + E||x - X||_p},$$

where $||x - X||_p$ is the L_p norm, encompassing cases like the Manhattan distance with $p = 1$ and Euclidean distance with $p = 2$.

Spatial Depth

Consider a random variable X from distribution F, the Spatial depth , also known as L_1 depth, for a point x with respect to distribution F is:

$$D(x; \hat{F}) = \frac{1}{1 + E(||x - X||)}.$$

The computation of spatial depth is available in R package *ddalpha* with `depth.spatial()`.

Projection Depth

For a random variable X from distribution F, the projection depth is defined as

$$D(x;F) = \frac{1}{1+O(x;F)},$$

where $O(x;\hat{F}) = \sup_{||y||=1} \frac{|<y,x>-med(<y,X>)|}{med|<y,x>-med(<y,X>)|}$, with $< y, x >$ denoting the dot product of x and y, and med the median. The projection depth calculation is available in R package *ddalpha* with `depth.projection()`.

Tukey Depth

For any points x in a R-dimensional distribution F, the Tukey depth [44], also known as halfspace depth, measures the minimum fraction of points in a halfspace,

$$D(x;\hat{F}) = \inf\{P(H) : H \text{ is a closed halfspace}, x \in H\},$$

where $P(H)$ is probability measure in a closed halfspace H. For a given point x, $P(H)$ usually represents the number of observations in H that contains point x.

In one-dimension, the Tukey depth simplifies significantly. For any point x, it is given by $D(x;F) = \min\{F(x), 1-F(x)\}$. The calculation of Tukey depth is implemented in the R package *ddalpha* with `depth.halfspace()` function.

Considering the above depth functions, we can now visualize the depth values of each depth function through contour plots. A unique type of data is banana shape data, which has a crescent-like shape; see the scatter plot of 100 random simulations in Figure 1.1. I simulated banana shape data for 100 generations.

The contour plots using Tukey depth, Mahalanobis depth, projection depth, and spatial depth are presented in Figure 1.2a, 1.2b, 1.2c, and 1.2d, respectively. The shape of contour plots is similar to that of the data simulated with a bivariate normal distribution.

In Chapter 2, we study a case using univariate Euclidean depth for our proposed test statistics with its asymptotic null distribution. The test statistics proposed could also be applied to multivariate data for various depth functions. In the simulation study to compare the statistical power, we consider Euclidean depth for univariate data and apply Mahalanobis depth, Spatial depth, and Projection depth for multivariate data; for more details, refer to Chapter 3.

1.3 Depth-based tests

Depth-based tests have been increasingly developed in recent years. A popular method is the Depth-based Rank Statistic (DbR), proposed by [10], which generalizes the Kruskal-Wallis test. More recently, [3] have enhanced the DbR test to improve its statistical power, introducing the Modified Depth-based Rank Test (BDbR).

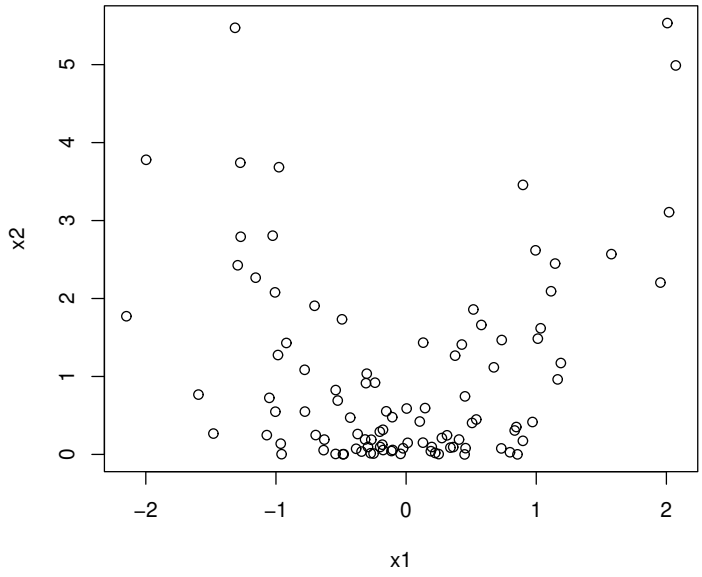


Figure 1.1: Scatter plot of banana shape data

Depth-based Rank Statistic (DbR)

The DbR test, as outlined by Small [10], arranges data into ranks. For a univariate sample $X = \{x_1, \ldots, x_n\}$ from distribution F, the rank of each point x_i is determined by its order in the data sequence, defined as $R(x_i) = \#\{x_j : x_j \geq x_i\}$, where $\#$ denotes the set cardinality. In the case of multivariate data, ranking is based on the data depth $D(x, F)$ of each data point x_i with respect to distribution F, defined as $R(x_i) = \#\{j : D(x_j, F) \geq D(x_i, F)\}$.

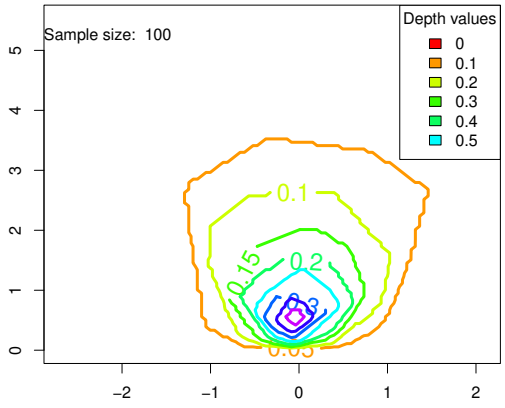
Consider two independent samples X_1 and X_2 from empirical distributions $\hat{F}_1$ and $\hat{F}_2$ with sample size n_1 and n_2, respectively. The rank $R_{i,j}(k)$ of point $X_{i,j}$ with respect to distribution $\hat{F}_k$ (where $k = 1, 2$) is calculated. The null hypothesis is $H_0 : F_1 = F_2$ and alternative hypothesis is $H_\alpha : F_1 \neq F_2$. The test statistics H is defined as

$$H = \frac{12}{n(n+1)t} \sum_{k=1}^{2} \sum_{j=1}^{2} \frac{R_{\cdot j}^2(k)}{n_j} - 3(n+1),$$

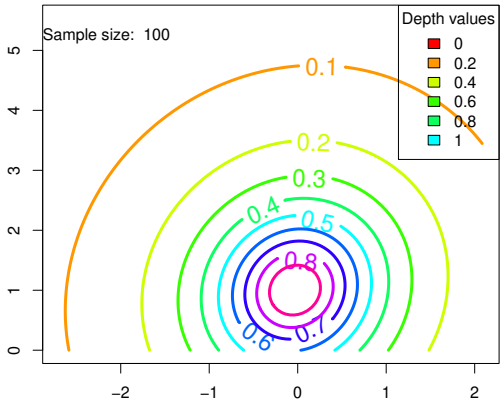
where $t = 2$ (the number of distributions), $n = n_1 + n_2$ with $j = 1, 2$ (the number of samples X_j), and $R_{\cdot j}(k) = \sum_{i=1}^{n_j} R_{i,j}$.

The test statistic rejects at significance level α when $H > T$, where T represents the upper α quantile of the distribution for H.

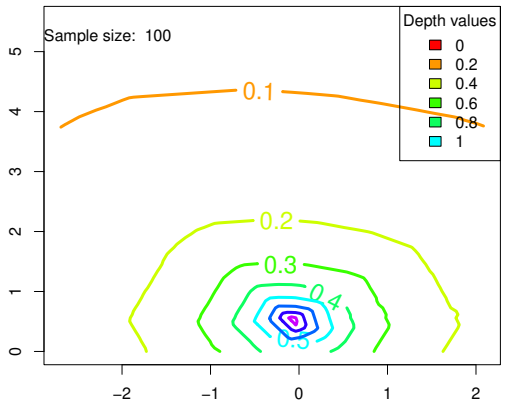
Modified Depth-based Rank Test (BDbR)



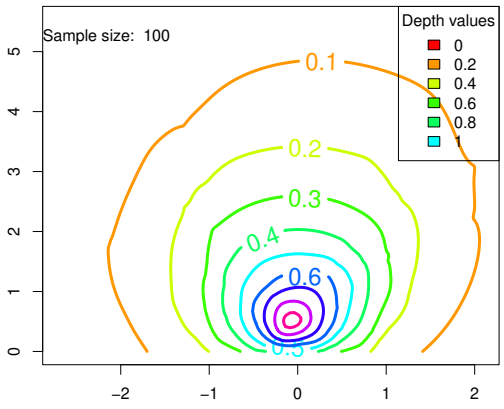
(a) Contour plot using Tukey depth



(b) Contour plot using Mahalanobis depth



(c) Contour plot using projection depth



(d) Contour plot using spatial depth

Figure 1.2: Contour plots of Banana shape data

Barale and Shirke [3] modified the two-sample homogeneity depth-based tests for scale-location problems. Consider two independent samples $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_m\}$ from distributions F and G, respectively, the goal is to test whether these two samples are from the same distribution, i.e. $H_0 : F = G$ vs. $H_1 : F \neq G$. The Baumgartner statistic, as proposed by Baumgartner [4], is utilized for univariate data ranking. This combines samples X and Y into a single set of size $N = n+m$, with ranks $R_{(1)} < ... < R_{(n)}$ and $Q_{(1)} < ... < Q_{(m)}$ corresponding to each sample. The test statistics is computed by taking average of individual test statistic values for each sample, i.e., $B = \frac{1}{2}(B_1 + B_2)$, where

$$B_1 = \frac{1}{n}\sum_{i=1}^{n} \frac{(R_{(i)} - \frac{N}{n}i)^2}{\frac{i}{n+1}(1-\frac{i}{n+1})\frac{mN}{n}},$$

and

$$B_2 = \frac{1}{m}\sum_{j=1}^{m} \frac{(Q_{(j)} - \frac{N}{m}j)^2}{\frac{j}{m+1}(1-\frac{j}{m+1})\frac{nN}{m}}.$$

Denote the upper α quantile as T, the test can be performed by $P(B < T) = 1 - \alpha$. That is, a larger value of B will result in the rejection of the null hypothesis.

However, the Baumgartner statistic is not invariant when transforming all data into negative values in the case of unequal sample sizes. A modified test is proposed by Murakami [37], which considers taking average of the squared standardized linear ranks. The modified test statistics B^* is:

$$B^* = \frac{1}{2}(B_1^* + B_2^*),$$

where

$$B_1^* = \frac{1}{n}\sum_{i=1}^{n} \frac{(R_{(i)} - E(R_{(i)}))^2}{Var(R_{(i)})},$$

$$B_2^* = \frac{1}{m}\sum_{j=1}^{m} \frac{(Q_{(j)} - E(Q_{(j)}))^2}{Var(Q_{(j)})},$$

with $E(R_{(i)}) = \frac{N+1}{n+1}i$, $E(Q_{(j)}) = \frac{N+1}{m+1}j$, $Var(R_{(i)}) = \frac{i}{n+1}\Big(1-\frac{i}{n+1}\Big)\frac{m(N+1)}{n+2}$, and $Var(Q_{(j)}) = \frac{j}{m+1}\Big(1-\frac{j}{m+1}\Big)\frac{n(N+1)}{m+2}$. Similarly, we reject the test statistic with $B^* > T$ at significance level α, where T denotes the upper α quantile for this distribution, and large test statistic values B^* will result in the rejection of the null hypothesis.

In the multivariate case [3], for $X_{ij} \in \mathbb{R}^p$, let $X_1 = \{X_{11}, ..., X_{1n_1}\}$ and $X_2 = \{X_{21}, ..., X_{2n_2}\}$ be two independent samples from distribution F_1 and F_2, respectively. Let $\hat{F_1}$ and $\hat{F_2}$ be the corresponding empirical distributions. To test the homogeneity of two distributions for both location vector μ and scale matrix Σ, we have $H_0 : F_1 = F_2$ vs. $H_1 : F_1 \neq F_2$.

The proposed procedure for this test statistic is summarized below:

(1) Combine the two samples X_1 and X_2 with total size N, i.e. $N = n_1 + n_2$, denote the combined samples as $Z = X_1 \bigcup X_2$, and Z_t stands for an observation in Z for $t = 1, \ldots, N$.

(2) Compute the depth of all Z_t with respect to $\hat{F}_1$ and $\hat{F}_2$, and denote them as $D(Z_t, \hat{F}_1)$ and $D(Z_t, \hat{F}_2)$ respectively.

(3) Rank all observations based on depth values $D(Z_t, \hat{F}_1)$ and record them as $R_t^{\hat{F}_1}$. Similarly, rank all observations based on $D(Z_t, \hat{F}_2)$ and record them as $R_t^{\hat{F}_2}$.

(4) Rearrange these ranks in increasing order, select those ranks corresponding to sample X_2 and record as $R_{(1)}^{\hat{F}_1} < \ldots < R_{(n_2)}^{\hat{F}_1}$, select those ranks corresponding to sample X_1 and record as $R_{(1)}^{\hat{F}_2} < \ldots < R_{(n_1)}^{\hat{F}_2}$.

(5) The test statistics is defined as $B = \max(B^{\hat{F}_1}, B^{\hat{F}_2})$, with

$$B^{\hat{F}_1} = \frac{1}{n_2} \sum_{j=1}^{n_2} \frac{\left(R_{(j)}^{\hat{F}_1} - E(R_{(j)}^{\hat{F}_1})\right)^2}{Var(R_{(j)}^{\hat{F}_1})},$$

$$B^{\hat{F}_2} = \frac{1}{n_1} \sum_{j=1}^{n_1} \frac{\left(R_{(j)}^{\hat{F}_2} - E(R_{(j)}^{\hat{F}_2})\right)^2}{Var(R_{(j)}^{\hat{F}_2})}.$$

where

$$E\left(R_{(j)}^{\hat{F}_1}\right) = \frac{N+1}{n_2+1} j, \quad Var\left(R_{(j)}^{\hat{F}_1}\right) = \frac{j}{n_2+1}\left(1 - \frac{j}{n_2+1}\right)\frac{n_1(N+1)}{n_2+2},$$

and

$$E\left(R_{(j)}^{\hat{F}_2}\right) = \frac{N+1}{n_1+1} j, \quad Var\left(R_{(j)}^{\hat{F}_2}\right) = \frac{j}{n_1+1}\left(1 - \frac{j}{n_1+1}\right)\frac{n_2(N+1)}{n_1+2}.$$

We reject this test when $B > T$ at significance level α, where T denotes the upper α quantile for this distribution.

For the statistical power comparison study of DbR and BDbR with our proposed test statistics, we elaborate it with great details in Chapter 3.

1.4 Q Statistics

The previous methods, such as Cramér test [1], Wilcoxon Rank-Sum test [47], and Energy Distance test [43], are regarded as point-to-point (PtP) distance by measuring the relative distance between pairs of data points. These point-to-point (PtP) distances may encounter the problem of large variance, resulting in a less accurate comparison between two

distributions and potentially reducing the test's power, especially in scenarios with finite dimensions. Tests based on point-to-sample (PtS) central-outward ranking could perform better than PtP distances with less variance, such as Depth-based Rank Statistic (DbR) [10] and Modified Depth-based Rank Test (BDbR) [3], with a comparison of a point to a sample distribution. Despite this advancement, the Depth-based Rank Statistic (DbR) [10], which is based on the PtS central-outward ranking, still incorporates further PtP comparisons through univariate Kruskal-Wallis type tests. This addition can increase variance and subsequently decrease the test's power, as evidenced in simulations discussed in Chapter 3.

In this regard, we explore the depth-based quality index $Q(F, G)$, measuring the relative "outlyingness" of distribution F in comparison to distribution G, which is defined as

$$Q(F, G) = P\{D(X; F) \leq D(Y; F) | X \sim F, Y \sim G\},$$

where F is the reference distribution. When the two distributions, F and G, are unknown, the empirical distributions of F_m and G_n can be employed, assuming sample sizes of X and Y are m and n, respectively. The Q can be estimated by the statistic

$$Q(F_m, G_n) = \frac{1}{n} \sum_{j=1}^{n} R(y_j; F_m),$$

where the sample proportion $R(y_j; F_m) = \frac{1}{m} \sum_{i=1}^{m} I(D(x_i, F_m) \leq D(y_j, F_m))$ and $I(\cdot)$ is an indicator function.

The $Q(F_m, G_n)$ statistic serves as a sample-to-sample (StS) central-outward ranking, averaging the depth-based PtS central-outward ranking $R(y_i; F_m)$. The method provides a more accurate comparison than the PtP distance, as it encompasses broader information about the distribution rather than focusing on single points. In addition to its enhanced power, the depth-based StS central-outward ranking benefits from the free distribution assumption and adheres to four fundamental properties of data depth: affine invariance, centroid maximality, monotonicity about the deepest point, and vanishing at infinity [40].

With different reference distributions F_m and G_n, usually $Q(F_m, G_n) \neq Q(G_n, F_m)$. Under null hypothesis, i.e., $H_0 : F = G$, the $Q(F, G) = \frac{1}{2}$. Under alternative hypothesis, a large difference from $Q(F_m, G_n)$, or $Q(G_n, F_m)$, to $\frac{1}{2}$ indicates a significant difference between these two distributions F_m and G_n.

While the depth-based StS central-outward ranking effectively handles multivariate data comparisons, it may lose certain information, such as data direction, by the one-dimensional projection of the data depth. This loss can be critical in achieving higher statistical power. Addressing this, a recent advancement by [41] suggests preserving power by considering the maximum of two Q statistics $Q(F_m, G_n)$ and $Q(G_n, F_m)$. This insight has led us to explore a new approach to pairwise StS central-outward ranking, derived from Q statistics. Our method maintains the direction of the StS central-outward ranking by analyzing the sign

of the partial derivatives of the Q statistics. Q statistics sharing the same sign indicate a unified direction of change under the alternative hypothesis. By considering making a combination of two Q Statistics with the property of "same-attraction", where the two Q Statistics have the same limit under the null hypothesis and approach to the same value under the alternative hypothesis, we can enhance the power of a test based on the derived Q statistic. In physics, an attractor refers to a set of numerical values toward which a system tends to evolve, regardless of its starting conditions. This can present the long-term behavior of the system. For example, consider all objects near a black hole; they are attracted in the same direction. In this thesis, we apply this concept of attractors, with the "same attractor" referring to the convergence of distributions over time toward a specific limit. Specifically, the statistic $Q(F_m, G_n)$ is attracted to $\frac{1}{2}$ under H_0, while under H_α, it is attracted to the limit of 0 or 1. Here, the term "attraction" denotes the direction of movement. While considering the maximum of $Q(F_m, G_n)$ and $Q(G_n, F_m)$ is one such combination, it is not the most efficient one because their partial derivatives with respect to each Q statistic are not strictly positive or negative, and may be zero. Our two new combinations, $Q(F_m, G_n) + Q(G_n, F_m)$ and $Q(F_m, G_n) \times Q(G_n, F_m)$, promise a greater improvement in power since their partial derivatives have the same sign and are almost never zero; for more details, see Definition 1 in Chapter 2.1. We have named our proposed technique DEEPEAST, short for depth-explored same-attraction StS central-outward ranking.

1.5 U-Statistics

1.5.1 U-Statistics

Wassily Hoeffding first introduced U-Statistics, dates backs to 1948 [22, 21]. More generalizations and properties of U-Statistics can be found in [27] by A. J. Lee.

Consider a function $f(x_1, x_2, \ldots, x_k)$ with k variables, we can take average of all f values over the set of all $k!$ permutations, denoted as $\pi = (\pi_{(1)}, \ldots, \pi_{(k)})$, where $\pi_{(i)} \in \{1, \ldots, k\}$ without replacements, that is

$$f(x_1, x_2, \ldots, x_k) = \frac{1}{k!} \sum_{\text{all } \pi} f(x_{\pi_{(1)}}, x_{\pi_{(2)}}, \ldots, x_{\pi_{(k)}}).$$

In this way, the function $f(x_1, x_2, \ldots, x_k)$ is symmetric as it takes average of all possible permutations.

Then U-Statistics [22], with "U" stands for "unbiased", is defined as

$$U_n = \frac{1}{\binom{n}{k}} \sum_{(n,k)} f(x_1, x_2, \ldots, x_k),$$

where f is called symmetric kernel and (n, k) is the set of all combinations $\binom{n}{k}$. The number $k \leq n$ is the degree of U-Statistics.

Example 1.5.1. A simple example is sample mean. Suppose $f(x) = x$, then the corresponding U-Statistics is

$$U_n = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

The degree of this U-Statistics is 1.

Example 1.5.2. Consider a more complex situation with degree 2. Let $f(x_1, x_2) = \frac{1}{2}(x_1 - x_2)^2$. Then the corresponding U-Statistics for this symmetric kernel is

$$\begin{aligned}
U_n &= \frac{1}{\binom{n}{2}} \sum_{i<j} f(x_i, x_j) \\
&= \frac{2}{n(n-1)} \sum_{i<j} \frac{1}{2}(x_i - x_j)^2 \\
&= \frac{1}{n(n-1)} \sum_{i\neq j} \frac{1}{2}(x_i^2 + x_j^2 - 2x_i x_j) \\
&= \frac{1}{n(n-1)} [(n\sum_{i=1}^{n} x_i^2) - (n\bar{x})^2)] \\
&= \frac{1}{n-1} [(\sum_{i=1}^{n} x_i^2) - n\bar{x}^2] \\
&= \frac{1}{n-1} [(\sum_{i=1}^{n} x_i^2) - 2\bar{x}(n\bar{x}) + n\bar{x}^2] \\
&= \frac{1}{n-1} \sum_{i=1}^{n} (x_i^2 - \bar{x})^2
\end{aligned}$$

Note that this is sample variance.

For two-sample cases, the idea is similar to one-sample case [16]. Consider two distributions F and G, independent samples $X_1, X_2, \ldots, X_{n_1}$ and $Y_1, Y_2, \ldots, Y_{n_2}$ are from F and G respectively. Then denote $f(x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2})$ as a symmetric kernel with degree (k_1, k_2), contains elements from both samples, and $k_1 \leq n_1, k_2 \leq n_2$. The U-Statistics for two-sample problem is

$$U_{n_1,n_2} = \frac{1}{\binom{n_1}{k_1}\binom{n_2}{k_2}} \sum_{(n_1,k_1)} \sum_{(n_2,k_2)} f(x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2}),$$

where (n_1, k_1) is the set of all combinations $\binom{n_1}{k_1}$, same to (n_2, k_2).

Example 1.5.3. A popular example of two-sample U-Statistics is Wilcoxon Rank-Sum test with Equation (1.2). In this case, the kernel is $f(x, y) = I(x < y)$.

1.5.2 V-Statistics

V-Statistics is closely related to U-Statistics, which is first proposed by von Mises in [45]. For a symmetric kernel $f(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, the V-Statistic with degree k is defined as [27]

$$V_n = n^{-k} \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} f(x_{i_1}, x_{i_2}, \ldots, x_{i_k}).$$

Example 1.5.4. With the same kernel $f(x_1, x_2) = \frac{1}{2}(x_1 - x_2)^2$ in Example 1.5.2, the V-Statistic with degree 2 can be written as

$$\begin{aligned}
V_n &= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2}(x_i - x_j)^2 \\
&= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2}(x_i^2 + x_j^2 - 2x_i x_j) \\
&= n^{-2}[(n \sum_{i=1}^{n} x_i^2) - (n\bar{x})^2)] \\
&= \frac{1}{n}[(\sum_{i=1}^{n} x_i^2) - n\bar{x}^2] \\
&= \frac{1}{n}[(\sum_{i=1}^{n} x_i^2) - 2\bar{x}(n\bar{x}) + n\bar{x}^2] \\
&= \frac{1}{n} \sum_{i=1}^{n} (x_i^2 - \bar{x})^2
\end{aligned}$$

There are some difference between V-Statistics and U-Statistics, as the U-Statistic with the same kernel is $\frac{1}{n-1} \sum_{i=1}^{n} (x_i^2 - \bar{x})^2$.

V-Statistics can also be written in the form of a combination of U-Statistics [27], with

$$V_n = n^{-k} \sum_{j=1}^{k} j! S_k^{(j)} \binom{n}{j} U_n^{(j)},$$

where $U_n^{(j)}$ are U-Statistics with degree j and $S_k^{(j)}$ represents the Stirling numbers of the second kind. The kernel $h_{(j)}(x_1, \ldots, x_j)$ of each $U_n^{(j)}$ will be determined by

$$h_{(j)}(x_1, \ldots, x_j) = (j! S_k^{(j)})^{-1} \sum_{(j)} f(x_{i_1}, x_{i_2}, \ldots, x_{i_k}),$$

with $\sum_{(j)}$ denoting taking the summation of all possible permutations for the set $(i_1, \ldots, i_k)$. The Stirling numbers of second kind $S_k^{(j)}$ can be determined by [27]

$$x^k = \sum_{j=1}^{k} S_k^{(j)} x(x-1)(x-2)\ldots(x-j+1).$$

The proof of transforming V-Statistics into U-Statistics can be found in Section 4.2 in [27].

Example 1.5.5. Consider a simple case $k = 3$, define V-Statistic as

$$V_n = n^{-3} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \sum_{i_3=1}^{n} f(x_{i_1}, x_{i_2}, x_{i_3}).$$

Then we have $x^3 = S_k^{(1)}x + S_k^{(2)}x(x-1) + S_k^{(3)}x(x-1)(x-2) = S_k^{(1)}x + S_k^{(2)}(x^2-x) + S_k^{(3)}(x^3 - 3x^2+2x) = S_k^{(3)}x^3 + (S_k^{(2)} - 3S_k^{(3)})x^2 + (S_k^{(1)} - S_k^{(2)} + 2S_k^{(3)})x$, then $S_k^{(3)} = 1, S_k^{(2)} = 3, S_k^{(1)} = 1$. The V-Statistics can be written as

$$\begin{aligned} n^{-3}V_n &= \sum_{j=1}^{k} j! S_k^{(j)} \binom{n}{j} U_n^{(j)} \\ &= 1! S_k^{(1)} \binom{n}{1} U_n^{(1)} + 2! S_k^{(2)} \binom{n}{2} U_n^{(2)} + 3! S_k^{(3)} \binom{n}{3} U_n^{(3)} \\ &= \binom{n}{1} U_n^{(1)} + 6\binom{n}{2} U_n^{(2)} + 6\binom{n}{3} U_n^{(3)} \end{aligned}$$

The kernels of U-Statistics will be as follows:

$$\begin{aligned} h_{(1)}(x_1) =& (1! S_k^{(1)})^{-1} \sum_{(i)} f(x_{i_1}, x_{i_2}, x_{i_3}) = f(x_1, x_1, x_1) \\ h_{(2)}(x_1, x_2) =& (2! S_k^{(2)})^{-1} \sum_{(i)} f(x_{i_1}, x_{i_2}, x_{i_3}) \\ =& \frac{1}{6}[f(x_1, x_1, x_2) + f(x_1, x_2, x_1) + f(x_2, x_1, x_1) \\ & + f(x_2, x_2, x_1) + f(x_2, x_1, x_2) + f(x_1, x_2, x_2)] \\ =& \frac{1}{2}(f(x_1, x_1, x_2) + f(x_1, x_2, x_2)) \\ h_{(3)}(x_1, x_2, x_3) =& (3! S_k^{(3)})^{-1} \sum_{(i)} f(x_{i_1}, x_{i_2}, x_{i_3}) \\ =& \frac{1}{6}[f(x_1, x_2, x_3) + f(x_1, x_3, x_2) + f(x_2, x_1, x_3) \\ & + f(x_2, x_3, x_1) + f(x_3, x_1, x_2) + f(x_3, x_2, x_1)] \\ =& f(x_1, x_2, x_3), \end{aligned}$$

where $h_{(1)}(x_1), h_{(2)}(x_1, x_2),$ and $h_{(3)}(x_1, x_2, x_3)$ are kernels of $U_n^{(1)}, U_n^{(2)},$ and $U_n^{(3)}$ respectively.

1.5.3 Hoeffding decomposition

Hoeffding decomposition [22] decomposes unbiased U-Statistics of degree k into a range of uncorrelated U-Statistics with degree $1, 2, \ldots, k$.

For kernel $f(x_1, x_2, \ldots, x_k)$, denote $\theta = E[f(x_1, x_2, \ldots, x_k)]$, the U-Statistics can be written in the form

$$\begin{aligned} U_n &= \theta + \sum_{j=1}^{k} \binom{k}{j} H_n^{(j)} \\ &= \theta + \sum_{j=1}^{k} \binom{k}{j} \binom{n}{j}^{-1} \sum_{(n,k)} h^{(j)}(x_{v_1}, \ldots, x_{v_j}), \end{aligned}$$

where $H_n^{(j)}$ is the degenerated U-Statistics with degree j and kernel $h^{(j)}$ [27].

Example 1.5.6. Continue the example above 1.5.3 on Wilcoxon Rank-Sum test, we can apply Hoeffding decomposition to derive the asymptotic null distribution. Let the test statistics be

$$U_n = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I(x_i < y_j) - \frac{1}{2},$$

the kernel is $f(x, y) = I(x_i < y_j) - \frac{1}{2}$. By Hoeffding decomposition, $f(x, y) = \theta + f_1(x) + f_2(y) + \phi(x, y)$, where $\theta = E[f(x, y)]$, $f_1(x) = E[f(x, Y)] - \theta$, $f_2(y) = E[f(X, y)] - \theta$, and $\phi(x, y) = f(x, y) - f_1(x) - f_2(y) - \theta$.

Suppose $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ are from normal distributions F and G respectively, and under null hypothesis $F = G$, we have

$$\begin{aligned} \theta &= E[f(x, y)] \\ &= \int_{-\infty}^{y} \int_{-\infty}^{\infty} f(x) f(y)\, dx\, dy - \frac{1}{2}, \text{where } f \text{ is PDF of normal distribution} \\ &= \int_{-\infty}^{\infty} F(y) f(y)\, dy - \frac{1}{2}, \text{where } F \text{ is CDF of normal distribution} \\ &= \frac{1}{2} - \frac{1}{2} \\ &= 0 \\ f_1(x) &= E[f(x, Y)] - \theta \\ &= E_y[I(x_i < y_j) - \frac{1}{2}] - 0 \\ &= 1 - F(x_i) - \frac{1}{2} \\ &= \frac{1}{2} - F(x_i) \end{aligned}$$

$$
\begin{aligned}
f_2(y) &= E[f(X,y)] - \theta \\
&= E_x[I(x_i < y_j) - \frac{1}{2}] - 0 \\
&= F(y_j) - \frac{1}{2}
\end{aligned}
$$

Hence, it can be written as

$$
\begin{aligned}
U_n &= \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(x_i < y_j) - \frac{1}{2} \\
&= 0 + \frac{1}{m}\sum_{i=1}^{m}[\frac{1}{2} - F(x_i)] + \frac{1}{n}\sum_{j=1}^{n}[F(y_j) - \frac{1}{2}] + \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\phi(x,y) \\
&= \frac{1}{m}\sum_{i=1}^{m}[\frac{1}{2} - F(x_i)] + \frac{1}{n}\sum_{j=1}^{n}[F(y_j) - \frac{1}{2}] + O_p(\frac{1}{mn})
\end{aligned}
$$

Then we can have $F(x_i)$ and $F(y_j)$ follows uniform distribution $\mathcal{U}(0,1)$ as $P(F(x_i) < x) = P(x_i < F^{-1}(x)) = F(F^{-1}(x)) = x$. Then $E[F(x_i)] = E[F(y_j)] = \frac{1}{2}$ and $Var[F(x_i)] = Var[F(y_j)] = \frac{1}{12}$. Hence, $\frac{1}{2} - F(x_i) \sim \mathcal{N}(0, \frac{1}{12})$ and $F(y_j) - \frac{1}{2} \sim \mathcal{N}(0, \frac{1}{12})$.

Then the normalized asymptotic distribution of U_n is

$$
\sqrt{\frac{mn}{m+n}}U_n \xrightarrow{d} \mathcal{N}(0, \frac{1}{12}).
$$

While U_n is recognized as a V-statistic with an asymptotic normal distribution, the same cannot be straightforwardly inferred for our proposed Sum Statistic $S_{m,n}$, which takes the summation of two Q Statistics, $Q(F_m, G_n) + Q(G_n, F_m)$, and Product Statistic $P_{m,n}$, which takes the product of two Q Statistics, $Q(F_m, G_n) \times Q(G_n, F_m)$. Through a higher-order approximation based on the Hoeffding decomposition, we can approach a more accurate determination of their asymptotic distributions under univariate Euclidean depth and extended to multidimensional case for all depths, leading to the formulation of relevant theorems in Chapter 2.3.

Chapter 2

DEEPEAST Technique

2.1 Same-attraction Function

The use of Q Statistics as StS central-outward ranking leads to a natural question: How can we ensure functional consistency across all Q Statistics to enhance power? This section is dedicated to address this question. Let us consider the scenario where we aim to combine L Q Statistics, denoted as $Q_1, \ldots, Q_L$. We present the combined function as $\mathcal{G}(Q_1, \ldots, Q_L)$. To optimally gauge similarity within the same distribution and dissimilarity across different distributions, the combined function $\mathcal{G}$ should ideally satisfy two properties: selfsame and coordinate, which are crucial for ensuring both the efficacy and reliability of the function in different statistical context. We have formalized and detailed these properties in Definition 1, providing a framework for evaluating and applying the combined Q Statistics function in practical scenarios.

Definition 1 (Optimal same-attraction function)**.** Assume the following properties for $Q_1, \ldots, Q_L$:

(i) P1. *Selfsame:* $Q_1, \ldots, Q_L$ share the asymptotic "same" null distribution.

(ii) P2. *Coordinate:* The partial derivative $\frac{\partial \mathcal{G}(Q_1,\ldots,Q_L)}{\partial Q_\ell}$ is non-negative (≥ 0) or non-positive (≤ 0) almost surely for all $\ell = 1, \ldots, L$ under the alternative hypothesis.

(iii) P3. *Optimum:* Consider G^* a set of all possible combinations of function $\mathcal{G}(Q_1, \ldots, Q_L)$, the most powerful test statistics G^0 can be selected according to taking the maximum of equations below:

$$\underset{G \in G^*}{\operatorname{argmax}} \frac{G(Q_1, \ldots, Q_L)}{c_{\alpha,G}} = G^0,$$

where the $c_{\alpha,G}$ are defined as $P_{H_0}[G > c_{\alpha,G}] = \alpha$ with type I error probability α under null hypothesis.

Any function $\mathcal{G}(Q_1, \ldots, Q_L)$ that satisfy P1 and P2 are same-attraction function, and among a family of same-attraction functions, the optimal same-attraction function can be found through P3.

Note that a same-attraction function $\mathcal{G}(Q_1, \ldots, Q_L)$ is strictly same-attraction if the inequalities in P2 are strict almost surely, meaning:

$$\frac{\partial \mathcal{G}(Q_1, \ldots, Q_L)}{\partial Q_\ell} > 0 \quad \text{or} \quad < 0.$$

It can be shown that a collection of same-attraction functions $\mathcal{G}_s(Q_1, \ldots, Q_L)$, $1 \leq s \leq S$ is closed under countable additions. This means that the sum of a number of same-attraction functions remains to be a same-attraction function. However, it is important to note that this closure may not apply to subtraction.

There will be some examples of same-attraction functions.

Example 2.1.1 (Maximum statistic [41])**.** Consider the maximum statistic

$$M_{m,n} = \max(Q_1, Q_2), \tag{2.1}$$

where

$$Q_1 = \left[\frac{1}{12}(\frac{1}{m} + \frac{1}{n})\right]^{-1} (Q(F_m, G_n) - \frac{1}{2})^2$$

and

$$Q_2 = \left[\frac{1}{12}(\frac{1}{m} + \frac{1}{n})\right]^{-1} (Q(G_n, F_m) - \frac{1}{2})^2.$$

Both Q Statistics Q_1 and Q_2 are selfsame (P1), as they follow the same asymptotic null chi-squared distribution [50]. The coordinate (P2) is also met under H_1 since for $r = 1, 2$,

$$\frac{\partial M_{m,n}}{\partial Q_r} = \begin{cases} 1 & \text{if } M_{m,n} = Q_r \\ 0 & \text{otherwise .} \end{cases}$$

It is worth noting that $M_{m,n}$ is non-differentiable at $Q_1 = Q_2$ with zero probability almost surely. Thus, by Definition 1, $M_{m,n}$ qualifies as a same-attraction function.

Example 2.1.2 (Weighted average statistic [41])**.** The weighted average statistic, $W_{m,n}(w_1, w_2)$, is defined as

$$W_{m,n}(w_1, w_2) = w_1 Q_1 + w_2 Q_2, \tag{2.2}$$

where $w_1, w_2 > 0$, $w_1 + w_2 = 1$, and Q_1, Q_2 are defined in (2.1).

Contrasting with the maximum statistic in Example 2.1.1, for $r = 1, 2$ we observe:

$$\frac{\partial W_{m,n}(w_1, w_2)}{\partial Q_r} = w_r > 0.$$

Thus, $W_{m,n}(w_1, w_2)$ qualifies as a strictly same-attraction function.

Example 2.1.3 (Minimum statistic [8])**.** The minimum statistic $M^*_{m,n}$ is defined as

$$M^*_{m,n} = -\min(Q_1, Q_2), \tag{2.3}$$

where

$$Q_1 = \left[\frac{1}{12}(\frac{1}{m} + \frac{1}{n})\right]^{-1/2}(Q(F_m, G_n) - \frac{1}{2})$$

and

$$Q_2 = \left[\frac{1}{12}(\frac{1}{m} + \frac{1}{n})\right]^{-1/2}(Q(G_n, F_m) - \frac{1}{2}).$$

Both Q-statistics Q_1 and Q_2 fulfill the selfsame property (P1). The coordinate (P2) is also met under the alternative hypothesis H_1, as for $r = 1, 2$,

$$\frac{\partial M^*_{m,n}}{\partial Q_r} = \begin{cases} -1 & \text{if } \min(Q_1, Q_2) = Q_1 \\ 0 & \text{otherwise .} \end{cases}$$

Thus, $M^*_{m,n}$ is classified as a same-attraction.

Example 2.1.4 (Sum statistic [8, 19])**.** The sum statistic $S_{m,n}$ was firstly proposed by [8] and later studied by [19] and is defined as

$$S_{m,n} = -\frac{mn}{m+n}(Q(F_m, G_n) + Q(G_n, F_m) - 1) \tag{2.4}$$

The $Q(F_m, G_n)$ and $Q(G_n, F_m)$ are selfsame and $\frac{\partial S_{m,n}}{\partial Q(F_m, G_n)} = \frac{\partial S_{m,n}}{\partial Q(G_n, F_m)} = -\frac{mn}{m+n} < 0$. Hence, $S_{m,n}$ qualifies as a strictly same-attraction function.

Example 2.1.5 (Product statistic [8])**.** The product statistic $P_{m,n}$ is defined as

$$P_{m,n} = -\frac{mn}{m+n}\Big(Q(F_m, G_n)Q(G_n, F_m) - \frac{1}{4}\Big) \tag{2.5}$$

Consider the partial derivatives of $P_{m,n}$, we have

$$\frac{\partial P_{m,n}}{\partial Q(F_m, G_n)} = -\frac{mn}{m+n}Q(G_n, F_m) < 0$$

and

$$\frac{\partial P_{m,n}}{\partial Q(G_n, F_m)} = -\frac{mn}{m+n}Q(F_m, G_n) < 0$$

almost surely as $Q(F_m, G_n)$ and $Q(G_n, F_m)$ are almost surely positive. Hence, $P_{m,n}$ is strictly same-attraction.

2.2 Power Analysis

As indicated above, the definition of a strictly same-attraction function imposes a constraint on the sign of the partial derivatives. They must be consistently positive or negative and cannot equal zero. This characteristic potentially makes a strictly same-attraction function more effective than a non-strict same-attraction function in most cases, as zeros do not maintain the directionality of the StS central-outward ranking. Nevertheless, there are exceptions to this generalization, as illustrated in Example 2.1.2, which is less powerful than $M_{m,n}$ in practice; see simulation results in [41]. Therefore, a more strong criterion is needed to determine the benefit in power of one same-attraction function to another. This criterion is elaborated in Proposition 1.

Proposition 1. Suppose there are two same-attraction functions, $\mathcal{G}_1(Q_1, \ldots, Q_L)$ and $\mathcal{G}_2(Q_1, \ldots, Q_L)$. Given a type I error probability α, there are two decision rules: $\mathcal{G}_1(Q_1, \ldots, Q_L) > c_{\alpha,1}$ and $\mathcal{G}_2(Q_1, \ldots, Q_L) > c_{\alpha,2}$ so that $P_{H_0}[\mathcal{G}_r(Q_1, \ldots, Q_L) > c_{\alpha,r}] = \alpha$ for $r = 1, 2$. If $\frac{\mathcal{G}_1(Q_1,\ldots,Q_L)}{\mathcal{G}_2(Q_1,\ldots,Q_L)} \geq \frac{c_{\alpha,1}}{c_{\alpha,2}}$ under H_1, then $\mathcal{G}_1(Q_1, \ldots, Q_L)$ is more powerful than $\mathcal{G}_2(Q_1, \ldots, Q_L)$.

The proof of Proposition 1 hinges on the fact that $P_{H_1}[\mathcal{G}_1(Q_1, \ldots, Q_L) > c_{\alpha,1}] \geq P_{H_1}[\mathcal{G}_2(Q_1, \ldots, Q_L) > c_{\alpha,2}]$.

This criterion provides a framework for comparing the efficacy of various same-attraction functions. The following examples demonstrate its application in evaluating the power of different types of same-attraction functions.

The following examples will apply the criterion to compare the power of many different types of same-attraction functions.

Example 2.2.1 (Maximum statistic [41])**.** Continued the Example 2.1.1, the maximum statistic $M_{m,n}$ is more powerful than either Q_1 or Q_2, which can be further verified by the Proposition 1. Let $\mathcal{G}_1 = M_{m,n}$, $\mathcal{G}_2 = Q_1$ or Q_2. We observed that $\mathcal{G}_1 \geq \mathcal{G}_2$. Moreover, both $M_{m,n}$ and $\mathcal{G}_2$ converge in distribution to χ^2_1 under H_0, leading $c_{\alpha,1} = c_{\alpha,2}$, where $P(\chi^2_1 > c_{\alpha,1}) = \alpha$. Therefore, $\frac{\mathcal{G}_1}{\mathcal{G}_2} \geq \frac{c_{\alpha,1}}{c_{\alpha,2}} = 1$, indicating that $\mathcal{G}_1$ is asymptotically more powerful than $\mathcal{G}_2$.

Example 2.2.2 (Weighted average statistic [41])**.** Although $W_{m,n}(w_1, w_2)$ is strictly same-attraction, it is less powerful than $M_{m,n}$. From definition, $M_{m,n} \geq W_{m,n}(w_1, w_2)$. Moreover, both $M_{m,n}, W_{m,n}(w_1, w_2) \xrightarrow{d} \chi^2_1$ [41]. Therefore, $c_{\alpha,1} = c_{\alpha,2}$, where $P(\chi^2_1 > c_{\alpha,1}) = \alpha$. Therefore, $\frac{M_{m,n}}{W_{m,n}(w_1,w_2)} \geq 1 = \frac{c_{\alpha,1}}{c_{\alpha,2}}$, indicating that $M_{m,n}$ is asymptotically more powerful than $W_{m,n}(w_1, w_2)$.

Example 2.2.3 (Minimum statistic [8])**.** $M^*_{m,n}$ have the same asymptotic power with $M_{m,n}$. This equivalence is demonstrated by $M^*_{m,n} \xrightarrow{d} |\mathcal{N}(0,1)|$ and $(M^*_{m,n})^2 \xrightarrow{d} \chi^2_1$. Detailed explanations and proofs can be found in Theorem 3 in Chapter 2.3.3.

Example 2.2.4 (Sum statistic [8, 19])**.** We note that the rate of convergence of $S_{m,n}$ is very different from that of Q_1 or Q_2. Determining the asymptotic distribution of $S_{m,n}$ poses a challenge. [19] derived an asymptotic approximation of χ^2_1 for the Tukey depth in one-dimensional Euclidean space. Using an alternative method based on the second-order approximation of the V-statistic and Hoeffding decomposition [27], we derive an asymptotic Craig distribution for univariate Euclidean depths; details are in Chapter 2.3.1 and 2.3.4. Consequently, we set $c_{\alpha,1} = 1.6566$, where $P(S_{m,n} > c_{\alpha,1}) = 0.05$.

This leads to the conclusion that the sum statistic $S_{m,n}$ is asymptotically more powerful than the maximum statistic $M_{m,n}$ in certain scenarios. For instance, consider the condition:

$$Q(F_m, G_n) + Q(G_n, F_m) \leq q^+, \quad (0 \leq q^+ < 1). \tag{2.6}$$

This condition often applies in cases of mean shifts (see Chapter 3.1). Then,

$$\begin{aligned}
\frac{S_{m,n}}{\sqrt{M_{m,n}}} &= \frac{-(\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}}(Q(F_m,G_n)+Q(G_n,F_m)-1)}{\sqrt{12}\max(|Q(F_m,G_n)-\frac{1}{2}|,|Q(G_n,F_m)-\frac{1}{2}|)} \\
&\geq \frac{2(1-q^+)(\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}}}{\sqrt{12}}, \\
&\geq \frac{1.6566}{\sqrt{3.84}}, \text{ as } (\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}} \to \infty,
\end{aligned}$$

suggesting that $S_{m,n}$ is asymptotically more powerful than $M_{m,n}$ under the specified condition in (2.6).

Example 2.2.5 (Product statistic [8])**.** The asymptotic distribution of $P_{m,n}$ for univariate Euclidean depth can be obtained in a manner similar to that of $S_{m,n}$; detailed explanations are provided in Chapter 2.3.2. Setting $\alpha = 0.05$ and $c_{\alpha,1} = 0.9384$, we find $P(P_{m,n} > c_{\alpha,1}) = 0.05$; see Chapter 2.3.4. Additionally, under the condition specified in (2.6), the inequality $Q(F_m, G_n) \times Q(G_n, F_m) \leq (q^+)^2/4 < 1/4$ holds, which indicates that $P_{m,n}$ can capture more mean change than $S_{m,n}$. Consequently, we derive

$$\begin{aligned}
\frac{P_{m,n}}{\sqrt{M_{m,n}}} &= \frac{-(Q(F_m,G_n)Q(G_n,F_m)-\frac{1}{4})(\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}}}{\sqrt{12}\max(|Q(F_m,G_n)-\frac{1}{2}|,|Q(G_n,F_m)-\frac{1}{2}|)} \\
&> \frac{2(1/4-(q^+)^2/4)(\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}}}{\sqrt{12}} \\
&\geq \frac{0.9384}{\sqrt{3.84}}, \text{ as } (\frac{1}{m}+\frac{1}{n})^{-\frac{1}{2}} \to \infty
\end{aligned}$$

indicating that $P_{m,n}$ is asymptotically more powerful than $M_{m,n}$ under condition in (2.6). It is also noteworthy that $S_{m,n}$ and $P_{m,n}$ are comparable, as they have similar asymptotic distributions.

Moreover, we could visualize the rejection region through figures.

The Figure 2.1 below illustrates the rejection region of $S_{m,n}$, $P_{m,n}$, $M_{m,n}$, $Q(F_m, G_n)$, and $Q(G_n, F_m)$ under univariate Euclidean depth. The simulations are conducted for 1000 simulations with $m = n = 100$. Blue triangles are the Q Statistic values under the alternative hypothesis with $F = \mathcal{N}(0, 1), G = \mathcal{N}(0.8, 1.2)$; and the purple dots are the Q Statistic values under null hypothesis with $F = \mathcal{N}(0, 1), G = \mathcal{N}(0, 1)$. The rejection regions are shaded with colored lines.

As shown from the figure, almost all the Q Statistic values are covered by rejection region of $S_{m,n}$, $P_{m,n}$; the majority of the Q Statistic values are covered by rejection region of $M_{m,n}$. The rejection region of $M_{m,n}$ is the union of the rejection region of $Q(F_m, G_n)$ and $Q(G_n, F_m)$, which means $M_{m,n}$ is a more powerful test than considering only one of $Q(F_m, G_n)$ or $Q(G_n, F_m)$.

In the subsequent section, we will show the asymptotic distributions of $S_{m,n}$ and $P_{m,n}$ for univariate Euclidean depths and provide an extension to the multidimensional case for all depths.

2.3 Asymptotic null distribution

2.3.1 Sum Statistic

The asymptotic distribution of $S_{m,n}$ when common distributions are normal is shown in following theorem, the proof is based on Hoeffding decomposition and V-Statistics.

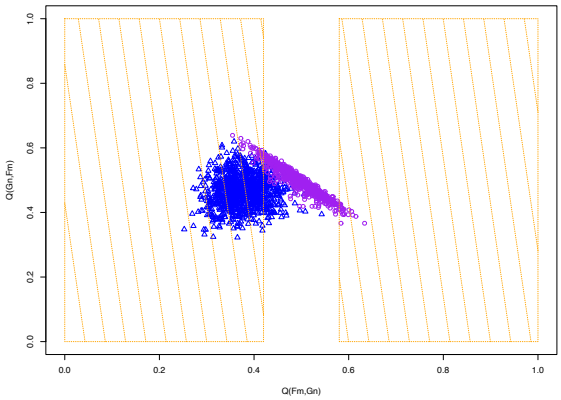
Theorem 1. Consider two independent and identical (iid) samples of the random variables $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ from the normal distributions F and G, and denote the corresponding empirical distributions as F_m and G_n, respectively. Under the null hypothesis $F = G$, the asymptotic distribution of $S_{m,n}$ follows the related Craig distribution [14] in one-dimensional Euclidean depth (1.3):

$$S_{m,n} \xrightarrow{d} -Z_1 Z_2, \tag{2.7}$$

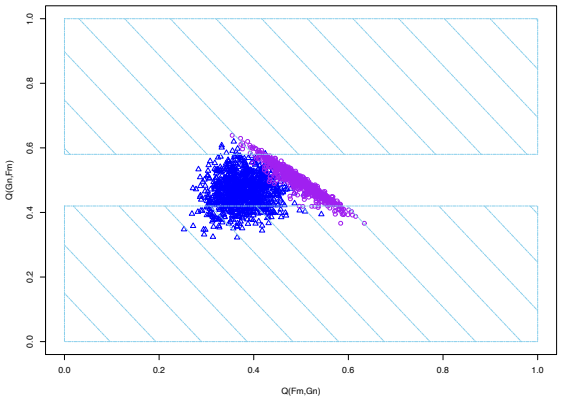
where $Z_1 \sim \mathcal{N}(0, 1)$, $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, $Cov(Z_1, Z_2) = -\frac{1}{\pi}$.

Proof. Under one-dimensional Euclidean depth,

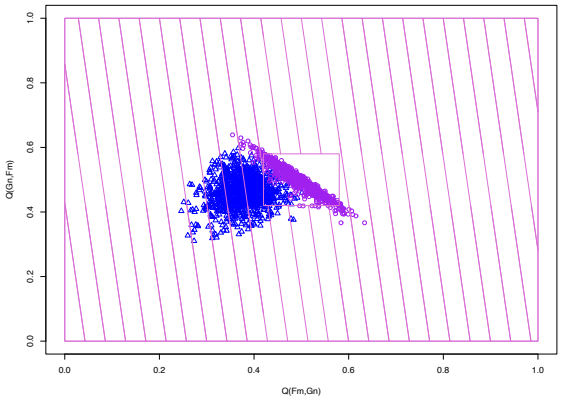
$$\begin{aligned} &Q(F_m, G_n) + Q(G_n, F_m) - 1 \\ =&\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I\left(\frac{1}{1 + (x_i - \bar{x})^2} \leq \frac{1}{1 + (y_j - \bar{x})^2}\right) \end{aligned}$$



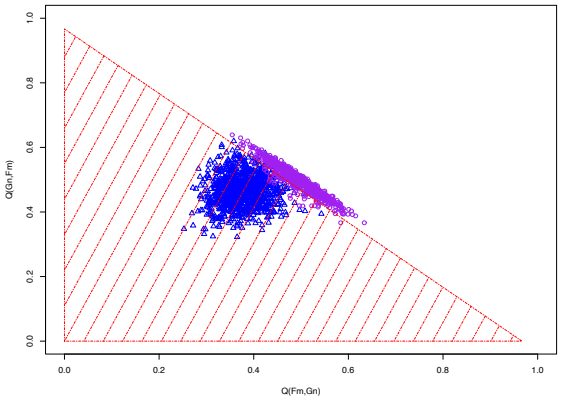
(a) Rejection region of $Q(F_m, G_n)$



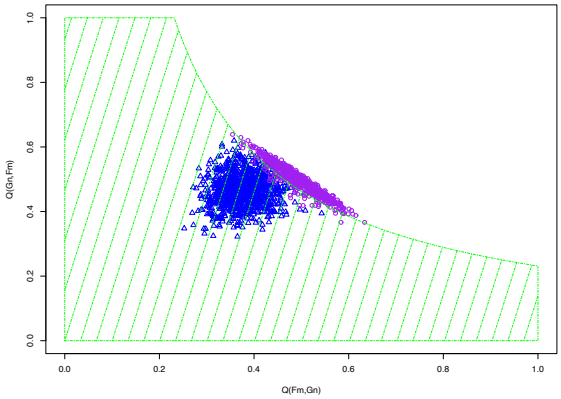
(b) Rejection region of $Q(F_m, G_n)$



(c) Rejection region of $M_{m,n}$



(d) Rejection region of $S_{m,n}$



(e) Rejection region of $P_{m,n}$

Figure 2.1: Rejection region of $S_{m,n}$, $P_{m,n}$, $M_{m,n}$, $Q(F_m, G_n)$, and $Q(G_n, F_m)$ under univariate Euclidean depth. Blue triangles ($F = \mathcal{N}(0,1), G = \mathcal{N}(0.8, 1.2)$); purple dots ($F = \mathcal{N}(0,1), G = \mathcal{N}(0,1)$) .

$$
\begin{aligned}
&+\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(\frac{1}{1+(y_j-\bar{y})^2} \leq \frac{1}{1+(x_i-\bar{y})^2}) - 1 \\
=&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i-\bar{x})^2 \geq (y_j-\bar{x})^2) + \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((y_j-\bar{y})^2 \geq (x_i-\bar{y})^2) - 1 \\
=&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i-\bar{x})^2 \geq (y_j-\bar{x})^2) - I((x_i-\bar{y})^2 \geq (y_j-\bar{y})^2)
\end{aligned}
$$

Denote $V(\lambda) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i-\lambda)^2 \geq (y_j-\lambda)^2) - \frac{1}{2}$, and $V(\lambda_1,\lambda_2) = V(\lambda_1) - V(\lambda_2) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i-\lambda_1)^2 \geq (y_j-\lambda_1)^2) - I((x_i-\lambda_2)^2 \geq (y_j-\lambda_2)^2)$.

By Hoeffding decomposition, $V(\lambda)$ can be decomposed to

$$
\begin{aligned}
V(\lambda) &= \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i-\lambda)^2 \geq (y_j-\lambda)^2) - \frac{1}{2} \\
&= \frac{1}{m}\sum_{i=1}^{m} E_y I((x_i-\lambda)^2 \geq (y_j-\lambda)^2) - \frac{1}{2} \\
&+ \frac{1}{n}\sum_{j=1}^{n} E_x I((x_i-\lambda)^2 \geq (y_j-\lambda)^2) - \frac{1}{2} + Op(\frac{1}{mn})
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
V(\lambda_1,\lambda_2) = &\frac{1}{m}\sum_{i=1}^{m} E_y I((x_i-\lambda_1)^2 \geq (y_j-\lambda_1)^2) - I((x_i-\lambda_2)^2 \geq (y_j-\lambda_2)^2) \\
&+ \frac{1}{n}\sum_{j=1}^{n} E_x I((x_i-\lambda_1)^2 \geq (y_j-\lambda_1)^2) - I((x_i-\lambda_2)^2 \geq (y_j-\lambda_2)^2) + op(\frac{1}{mn})
\end{aligned}
$$

By Taylor expansion and approximations to non-central chi-squared distribution [13],

$$
\begin{aligned}
&E_y I((x_i-\lambda_1)^2 \geq (y_j-\lambda_1)^2) - I((x_i-\lambda_2)^2 \geq (y_j-\lambda_2)^2) \\
=&F(\frac{(x_i-\lambda_1)^2}{1+\lambda_1^2}) - F(\frac{(x_i-\lambda_2)^2}{1+\lambda_2^2}) + O_p(\lambda_1^{-3}) + O_p(\lambda_2^{-3}), \text{where } F \text{ is CDF of } \chi_1^2, \\
&\text{by approximations to non-central chi-squared distribution [13]} \\
=&F((x_i-\lambda_1)^2) - F((x_i-\lambda_2)^2) + O_p(\lambda_1^{-3}) + O_p(\lambda_2^{-3}) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}) \\
=&F(x_i^2) - 2\lambda_1 x_i f(x_i^2) - F(x_i^2) + 2\lambda_2 x_i f(x_i^2) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}), \\
&\text{by Taylor expansion} \\
=&-2\lambda_1 x_i f(x_i^2) + 2\lambda_2 x_i f(x_i^2) + O_p(\lambda_1^{-2}) \\
=&-2x_i f(x_i^2)(\lambda_1-\lambda_2) + O_p(\lambda_1^{-2})
\end{aligned}
$$

and

$$
\begin{aligned}
&E_x I((x_i-\lambda_1)^2 \geq (y_j-\lambda_1)^2) - I((x_i-\lambda_2)^2 \geq (y_j-\lambda_2)^2) \\
=&1 - F(\frac{(y_j-\lambda_1)^2}{1+\lambda_1^2}) - 1 + F(\frac{(y_j-\lambda_2)^2}{1+\lambda_2^2}) + O_p(\lambda_1^{-3}) + O_p(\lambda_2^{-3}) \\
=&- F((y_j-\lambda_1)^2) + F((y_j-\lambda_2)^2) + O_p(\lambda_1^{-3}) + O_p(\lambda_2^{-3}) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}) \\
=&- F(y_j^2) + 2\lambda_1 y_j f(y_j^2) + F(y_j^2) - 2\lambda_2 y_j f(y_j^2) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}) + O_p(\lambda_1^{-2}) + O_p(\lambda_2^{-2}) \\
=&2\lambda_1 y_j f(y_j^2) - 2\lambda_2 y_j f(y_j^2) + O_p(\lambda_1^{-2}) \\
=&2y_j f(y_j^2)(\lambda_1-\lambda_2) + O_p(\lambda_1^{-2})
\end{aligned}
$$

As $\lambda_1, \lambda_2 \to 0$, $V(\lambda_1,\lambda_2) = \frac{1}{m}\sum_{i=1}^m -2x_i f(x_i^2)(\lambda_1-\lambda_2) + \frac{1}{n}\sum_{j=1}^n 2y_j f(y_j^2)(\lambda_1-\lambda_2) + [Op(\frac{1}{m}) + Op(\frac{1}{n})]|\lambda_1-\lambda_2|$.

Therefore, $V(\bar{x},\bar{y}) = \frac{1}{m}\sum_{i=1}^m -2x_i f(x_i^2)(\bar{x}-\bar{y}) + \frac{1}{n}\sum_{j=1}^n 2y_j f(y_j^2)(\bar{x}-\bar{y}) + [Op(\frac{1}{m}) + Op(\frac{1}{n})]|\bar{x}-\bar{y}|$.

In order to find the distribution of $V(\bar{x},\bar{y})$, we need to find the expectation and variance of $-2x_i f(x_i^2)$ and $2y_j f(y_j^2)$, with details:

$$
\begin{aligned}
&E[-2x_i f(x_i^2)] \\
=&\int_{-\infty}^{\infty} -2x_i f(x_i^2) g(x_i)\, dx_i, \text{where } f(x_i^2) = \frac{1}{\sqrt{2\pi}}(x_i^2)^{-\frac{1}{2}} e^{-\frac{x_i^2}{2}}, g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}} \\
=&\int_{-\infty}^{\infty} -2x_i \frac{1}{2\pi} |x_i^{-1}| e^{-x_i^2}\, dx_i \\
=&0
\end{aligned}
$$

and

$$
\begin{aligned}
&E[(-2x_i f(x_i^2))^2] \\
=&E[4x_i^2 f^2(x_i^2)] \\
=&\int_0^{\infty} 4z f^2(z) g(z)\, dz, \text{where } z = x_i^2, f(z) = g(z) = \frac{1}{\sqrt{2\pi}} z^{-\frac{1}{2}} e^{-\frac{z}{2}} \\
=&\int_0^{\infty} 4z (\frac{1}{\sqrt{2\pi}})^3 z^{-\frac{3}{2}} e^{-\frac{3z}{2}}\, dz \\
=&\frac{2}{\sqrt{3}\pi}
\end{aligned}
$$

Hence, $Var[-2x_i f(x_i^2)] = E[(-2x_i f(x_i^2))^2] - E[-2x_i f(x_i^2)]^2 = \frac{2}{\sqrt{3}\pi}$.
In a similar way, $E[2y_j f(y_j^2)] = 0$ and $Var[2y_j f(y_j^2)] = \frac{2}{\sqrt{3}\pi}$.
Therefore,

$$\sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^m -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^n 2y_j f(y_j^2)) \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$$

Since $(\bar{x}-\bar{y}) \sim \mathcal{N}(0, \frac{1}{m}+\frac{1}{n})$, then $\sqrt{\frac{mn}{m+n}}(\bar{x}-\bar{y}) \sim \mathcal{N}(0,1)$.

We can get the normalized $S_{m,n}$, specifically

$$S_{m,n} = -\frac{mn}{m+n} V(\bar{x}, \bar{y}) \xrightarrow{d} -Z_1 Z_2,$$

where $Z_1 \sim \mathcal{N}(0,1)$, and $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, with $Cov(Z_1, Z_2) = -\frac{1}{\pi}$.

The covariance between Z_1 and Z_2 can be computed as:

First, we have

$$\begin{aligned}
&E[-2x_i^2 f(x_i^2)] \\
&= \int_{-\infty}^{\infty} -2x_i^2 f(x_i^2) g(x_i^2)\, dx_i, \text{where } f(x_i^2) = \frac{1}{\sqrt{2\pi}}(x_i^2)^{-\frac{1}{2}} e^{-\frac{x_i^2}{2}}, g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}} \\
&= \int_0^{\infty} -2z^2 \frac{1}{2\pi} z^{-1} e^{-\frac{z}{2}} e^{-\frac{z}{2}}\, dx_i, \text{with } z = x_i^2 \\
&= -\frac{1}{\pi}
\end{aligned}$$

Then, $Cov(x_i, -2x_i f(x_i^2)) = E[-2x_i^2 f(x_i^2)] - E[x_i]E[-2x_i f(x_i^2)] = -\frac{1}{\pi} - 0 = -\frac{1}{\pi}$.

Similarly, $Cov(y_j, 2y_j f(y_j^2)) = \frac{1}{\pi}$.

Therefore,

$$\begin{aligned}
Cov(Z_1, Z_2) &= Cov(\sqrt{\frac{mn}{m+n}}(\bar{x}-\bar{y}), \sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2))) \\
&= \frac{mn}{m+n} Cov(\bar{x}-\bar{y}, \frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2)) \\
&= \frac{mn}{m+n}[Cov(\bar{x}, \frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2)) - Cov(\bar{y}, \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2))] \\
&= \frac{mn}{m+n}[\frac{1}{m} Cov(x_i, -2x_i f(x_i^2)) - \frac{1}{n} Cov(y_j, 2y_j f(y_j^2))] \\
&= \frac{mn}{m+n}[\frac{1}{m}(-\frac{1}{\pi}) - \frac{1}{n}(\frac{1}{\pi})] \\
&= -\frac{1}{\pi}
\end{aligned}$$

□

Remark 1. The convergence rate $\frac{mn}{m+n}$ in $S_{m,n}$ apply to other multivariate distributions F and G and to all depths, which provides a further theoretical support for broader applications. Under regular conditions, we can obtain a form similar to that shown in Theorem 1.

$$S_{m,n} \approx -\frac{mn}{m+n}\left\{\frac{1}{m}\sum_{i=1}^{m} -f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;G_n)] \right.$$
$$\left. +\frac{1}{n}\sum_{j=1}^{n} f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;G_n)]\right\},$$

where $F_{D(x;F)}(\cdot)$ and $f_{D(x;F)}(\cdot)$ are the distribution function and density of $D(x;F)$ respectively. For one-dimensional Euclidean depth (1.3), the extension is consistent with the result of the theorem.

Proof.

$$\begin{aligned}
S_{m,n} &= -\frac{mn}{m+n}(Q(F_m,G_n) + Q(G_n,F_m) - 1) \\
&= -\frac{mn}{m+n}[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i;F_m) \le D(y_j;F_m)) \\
&+ \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j;G_n) \le D(x_i;G_n)) - 1] \\
&= -\frac{mn}{m+n}\{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(D(x_i;F_m) \le D(y_j;F_m)) - I(D(x_i;G_n) \le D(y_j;G_n))]\}
\end{aligned}$$

By Hoeffding decomposition,

$$S_{m,n} = -\frac{mn}{m+n}\{\frac{1}{m}\sum_{i=1}^{m} E_y I(x_i,y_j;F_m,G_n) + \frac{1}{n}\sum_{j=1}^{n} E_x I(x_i,y_j;F_m,G_n) + O_p(\frac{1}{mn})\},$$

where $I(x_i,y_j;F_m,G_n) = I(D(x_i;F_m) \le D(y_j;F_m)) - I(D(x_i;G_n) \le D(y_j;G_n))$.

By the following two assumptions,

(1) Under null hypothesis H_0, $F = G$

(2) The limit of empirical distribution F_m is F.

According to above two assumptions and using Taylor expansions, we have

$$\begin{aligned}
&E_y I(D(x_i;F_m) \le D(y_j;F_m)) \\
\approx &E_y I(D(x_i;F_m) \le D(y_j;F)) \\
= &1 - F_{D(y;F)}(D(x_i;F_m)), \text{ where } F_{D(y;F)} \text{ denotes distribution of } D(y;F) \\
= &1 - F_{D(y;F)}(D(x_i;F_m) - D(x_i;F) + D(x_i;F)) \\
\approx &1 - F_{D(y;F)}(D(x_i;F)) - f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;F)],
\end{aligned}$$

and

$$\begin{aligned}
&E_y I(D(x_i;G_n) \le D(y_j;G_n)) \\
&E_y I(D(x_i;G_n) \le D(y_j;G)) \\
\approx &E_y I(D(x_i;G_n) \le D(y_j;F)) \\
= &1 - F_{D(y;F)}(D(x_i;G_n)), \text{ where } F_{D(y;F)} \text{ denotes distribution of } D(y;F) \\
= &1 - F_{D(y;F)}(D(x_i;G_n) - D(x_i;F) + D(x_i;F)) \\
\approx &1 - F_{D(y;F)}(D(x_i;F)) - f_{D(y;F)}(D(x_i;F))[D(x_i;G_n) - D(x_i;F)].
\end{aligned}$$

Therefore, we have $\frac{1}{m}\sum_{i=1}^{m} E_y I(x_i, y_j; F_m, G_n) = \frac{1}{m}\sum_{i=1}^{m} -f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;G_n)]$.

Similarly,

$$\begin{aligned}
&E_x I(D(x_i;F_m) \le D(y_j;F_m)) \\
\approx &E_x I(D(x_i;F) \le D(y_j;F_m)) \\
= &F_{D(x,F)}(D(y_j;F_m)) \\
\approx &F_{D(x,F)}(D(y_j;F)) + f_{D(x,F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;F)],
\end{aligned}$$

and

$$\begin{aligned}
&E_x I(D(x_i;G_n) \le D(y_j;G_n)) \\
\approx &E_x I(D(x_i;G) \le D(y_j;G_n)) \\
= &F_{D(x,F)}(D(y_j;G_n)) \\
\approx &F_{D(x,F)}(D(y_j;F)) + f_{D(x,F)}(D(y_j;F))[D(y_j;G_n) - D(y_j;F)],
\end{aligned}$$

Therefore, $\frac{1}{n}\sum_{j=1}^{n} E_x I(x_i, y_j; F_m, G_n) = \frac{1}{n}\sum_{j=1}^{n} f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;G_n)]$.

Then, $S_{m,n}$ can be written as

$$\begin{aligned}
S_{m,n} \approx -\frac{mn}{m+n}\{&\frac{1}{m}\sum_{i=1}^{m} -f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;G_n)] \\
&+ \frac{1}{n}\sum_{j=1}^{n} f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;G_n)]\}.
\end{aligned}$$

For $S_{m,n}$ under one-dimensional Euclidean depth, we can use Euclidean distance to replace the depth functions. Therefore,

$$S_{m,n} \approx -\frac{mn}{m+n}\{\frac{1}{m}\sum_{i=1}^{m} f_{d(y;F)}(d(x_i;F))[d(x_i;F_m) - d(x_i;G_n)] + \frac{1}{n}\sum_{j=1}^{n} -f_{d(x;F)}(d(y_j;F))[d(y_j;F_m) - d(y_j;G_n)]\},$$

where $d()$ represents Euclidean distance, and $F_{d(x;F)}$ and $f_{d(x;F)}$ are CDF and PDF of distribution $d(x;F)$.

Then we have $d(x_i;F_m) = (x_i - \bar{x})^2, d(y_j;F_m) = (y_j - \bar{x})^2, d(x_i;G_n) = (x_i - \bar{y})^2, d(y_j;G_n) = (y_j - \bar{y})^2, d(x_i;F) = x_i^2, d(y_j;F) = y_j^2$. Then, $f_{d(y;F)}(d(x_i;F)) = f(x_i^2)$ and $f_{d(x;F)}(d(y_j;F)) = f(y_j^2)$, where f is density of χ_1^2.

$$\begin{aligned}
S_{m,n} \approx & -\frac{mn}{m+n}\{\frac{1}{m}\sum_{i=1}^{m} f_{d(y;F)}(d(x_i;F))[(x_i-\bar{x})^2 - (x_i-\bar{y})^2] \\
& + \frac{1}{n}\sum_{j=1}^{n} -f_{d(x;F)}(d(y_j;F))[(y_j-\bar{x})^2 - (y_j-\bar{y})^2]\} \\
= & -\frac{mn}{m+n}\{\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)[(\bar{y}-\bar{x})(2x_i-\bar{x}-\bar{y})] + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)[(\bar{y}-\bar{x})(2y_j-\bar{x}-\bar{y})]\} \\
\approx & -\frac{mn}{m+n}\{\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)[(\bar{y}-\bar{x})2x_i] + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)[(\bar{y}-\bar{x})2y_j]\} \\
= & -\sqrt{\frac{mn}{m+n}}(\bar{y}-\bar{x})\sqrt{\frac{mn}{m+n}}[\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)2x_i + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)2y_j]
\end{aligned}$$

Since $\sqrt{\frac{mn}{m+n}}(\bar{y}-\bar{x}) \sim \mathcal{N}(0,1)$ and $\sqrt{\frac{mn}{m+n}}[\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)2x_i + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)2y_j] \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, $S_{m,n} \xrightarrow{d} -Z_1Z_2$, where $Z_1 \sim \mathcal{N}(0,1)$ and $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, with $Cov(Z_1, Z_2) = -\frac{1}{\pi}$.

□

Properties of $S_{m,n}$

PDF and CDF

The probability density functions $S_{m,n}$ can be written as $f_S(x)$:

$$f_S(x) = \frac{1}{\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}} e^{\frac{\sqrt{3}x}{2-\frac{\sqrt{3}}{\pi}}} \int_0^\infty \frac{1}{z_1} e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2 + \frac{\sqrt{3}\pi x^2}{2z_1^2})}\, dz_1.$$

$f_S(x)$ is derived in the following way:

First, the correlation coefficient $\rho = \frac{Cov(Z_1,Z_2)}{\sigma_{z_1}\sigma_{z_2}} = -\sqrt{\frac{\sqrt{3}}{2\pi}}$, where $\sigma_{z_1}, \sigma_{z_2}$ are standard deviation of Z_1, Z_2 respectively. The covariance matrix is: $\Sigma = \begin{pmatrix} 1 & -\frac{1}{\pi} \\ -\frac{1}{\pi} & \frac{2}{\sqrt{3}\pi} \end{pmatrix}$.

The density of $S_{m,n}$ is:

$$f(z_1, z_2) = \frac{1}{2\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}} e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}[z_1^2+\sqrt{3}z_1z_2+\frac{\sqrt{3}\pi}{2}z_2^2]}$$

.

To find $P(-Z_1Z_2 < x)$, there are two cases: $x > 0$ and $x < 0$.

- $x > 0$

  1. $z_1 > 0$, then $z_2 > -\frac{x}{z_1}$,

  $$P(-Z_1Z_2 < x) = P(Z_1Z_2 > -x) = \int_0^\infty dz_1 \int_{-\frac{x}{z_1}}^\infty f(z_1, z_2)\, dz_2.$$

  2. $z_1 < 0$, then $z_2 < -\frac{x}{z_1}$,

  $$P(-Z_1Z_2 < x) = P(Z_1Z_2 > -x) = \int_{-\infty}^0 dz_1 \int_{-\infty}^{-\frac{x}{z_1}} f(z_1, z_2)\, dz_2.$$

- $x < 0$

  1. $z_1 > 0$, then $z_2 > -\frac{x}{z_1}$,

  $$P(-Z_1Z_2 < x) = P(Z_1Z_2 > -x) = \int_0^\infty dz_1 \int_{-\frac{x}{z_1}}^\infty f(z_1, z_2)\, dz_2.$$

  2. $z_1 < 0$, then $z_2 < -\frac{x}{z_1}$,

  $$P(-Z_1Z_2 < x) = P(Z_1Z_2 > -x) = \int_{-\infty}^0 dz_1 \int_{-\infty}^{-\frac{x}{z_1}} f(z_1, z_2)\, dz_2.$$

Both $x > 0$ and $x < 0$ will have

$$\begin{aligned} f_S(x) &= \int_0^\infty dz_1 \int_{-\frac{x}{z_1}}^\infty f(z_1, z_2)\, dz_2 + \int_{-\infty}^0 dz_1 \int_{-\infty}^{-\frac{x}{z_1}} f(z_1, z_2)\, dz_2 \\ &= \int_0^\infty -f(z_1, -\frac{x}{z_1})(-\frac{1}{z_1})\, dz_1 + \int_{-\infty}^0 f(z_1, -\frac{x}{z_1})(-\frac{1}{z_1})\, dz_1 \\ &= 2\int_0^\infty f(z_1, -\frac{x}{z_1})\frac{1}{z_1}\, dz_1 \end{aligned}$$

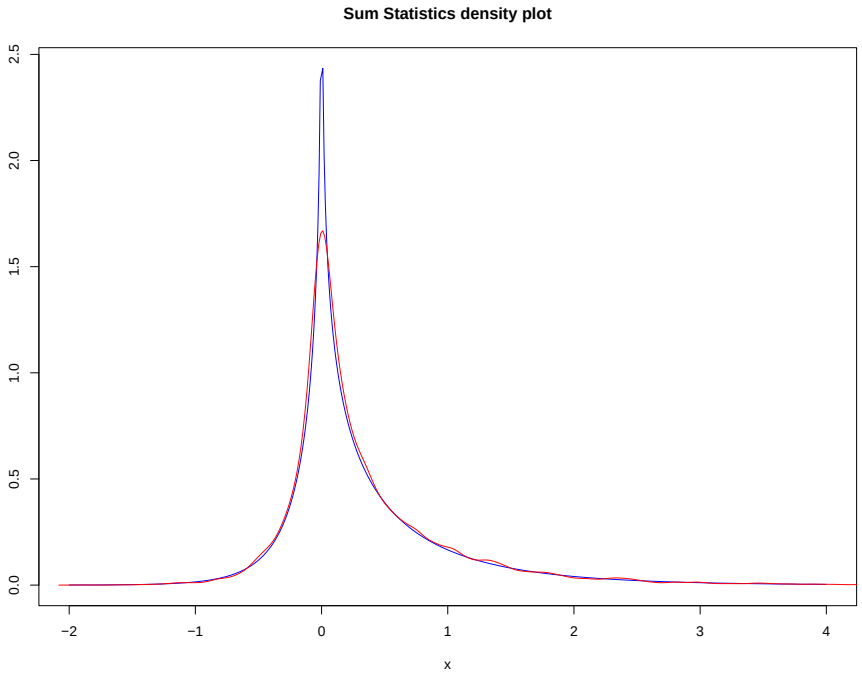


Figure 2.2: The true (blue) probability density functions vs. the estimated (red) density function for $f_S(x)$

$$= \frac{1}{\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}} e^{\frac{\sqrt{3}x}{2-\frac{\sqrt{3}}{\pi}}} \int_0^\infty \frac{1}{z_1} e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2+\frac{\sqrt{3}\pi x^2}{2z_1^2})}\, dz_1$$

We plot $f_S(x)$ in Fig. 2.2, compared with simulated density with $m = n = 1000$ and 10,000 repetitions. It can be seen that the densities are skewed to the right and the peaks are sharp.

The cumulative distribution function of $S_{m,n}$ is:

$$F_S(x) = \iint_{-z_1 z_2 < x} \frac{1}{2\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}} e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2+\sqrt{3}z_1z_2+\frac{\sqrt{3}\pi}{2}z_2^2)}\, dz_1\, dz_2.$$

Expectation and Variance

Since $S_{m,n} \xrightarrow{d} -Z_1Z_2$, and $Cov(Z_1, Z_2) = -\frac{1}{\pi}$, we can write Z_2 as a variable related to Z_1, which is $Z_2 = -\frac{1}{\pi}Z_1 + \sqrt{\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2}}Z_0$, where $Z_0 \sim \mathcal{N}(0,1)$.

Therefore, the expectation is $E[-Z_1Z_2] = -(-\frac{1}{\pi}) = \frac{1}{\pi}$.

The variance is calculated as

$$E[(-Z_1Z_2)^2] = E[Z_1^2(\frac{1}{\pi^2}Z_1^2 + (\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2})Z_0^2 - 2\frac{1}{\pi}\sqrt{\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2}}Z_1Z_0)]$$

$$
\begin{aligned}
&= E[\frac{1}{\pi^2}Z_1^4 + (\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2})Z_1^2 Z_0^2 - 2\frac{1}{\pi}\sqrt{\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2}}Z_1^3 Z_0] \\
&= \frac{1}{\pi^2}E[Z_1^4] + (\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2})E[Z_1^2 Z_0^2] - 2\frac{1}{\pi}\sqrt{\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2}}E[Z_1^3 Z_0] \\
&= 3\frac{1}{\pi^2} + (\frac{2}{\sqrt{3}\pi} - \frac{1}{\pi^2}) - 0 \\
&= \frac{2}{\sqrt{3}\pi} + \frac{2}{\pi^2} \\
Var(-Z_1 Z_2) &= E[(-Z_1 Z_2)^2] - E[-Z_1 Z_2]^2 \\
&= \frac{2}{\sqrt{3}\pi} + \frac{2}{\pi^2} - (\frac{1}{\pi})^2 \\
&= \frac{2}{\sqrt{3}\pi} + \frac{1}{\pi^2}
\end{aligned}
$$

2.3.2 Product Statistic

The asymptotic distribution of $P_{m,n}$ with common normal distributions is similar to $S_{m,n}$, shown in following theorem.

Theorem 2. Consider two independent and identical (iid) samples of the random variables $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ from the normal distributions F and G, and denote the corresponding empirical distributions as F_m and G_n, respectively. Under the null hypothesis $F = G$, the asymptotic distribution of $P_{m,n}$ follows the related Craig distribution [14] in one-dimensional Euclidean depth (1.3):

$$
P_{m,n} \xrightarrow{d} Z_3^2 - \frac{1}{2}Z_1 Z_2, \tag{2.8}
$$

where $Z_1 \sim \mathcal{N}(0,1)$, $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, $Z_3 \sim \mathcal{N}(0, \frac{1}{12})$, $Cov(Z_1, Z_2) = -\frac{1}{\pi}$, and Z_3 is independent of Z_1 and Z_2.

Proof. For $P_{m,n}$, under one-dimensional Euclidean depth,

$$
\begin{aligned}
&Q(F_m, G_n) \times Q(G_n, F_m) - \frac{1}{4} \\
=&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(\frac{1}{1+(x_i - \bar{x})^2} \leq \frac{1}{1+(y_j - \bar{x})^2}) \\
&\times \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(\frac{1}{1+(y_j - \bar{y})^2} \leq \frac{1}{1+(x_i - \bar{y})^2}) - \frac{1}{4} \\
=&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i - \bar{x})^2 \geq (y_j - \bar{x})^2) \times \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((y_j - \bar{y})^2 \geq (x_i - \bar{y})^2) - \frac{1}{4} \\
=&[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i - \bar{x})^2 \geq (y_j - \bar{x})^2) - \frac{1}{2}] \times [\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((y_j - \bar{y})^2 \geq (x_i - \bar{y})^2) - \frac{1}{2}]
\end{aligned}
$$

$$+\frac{1}{2}[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\bar{x})^2\geq(y_j-\bar{x})^2)-I((x_i-\bar{y})^2\geq(y_j-\bar{y})^2)]$$
$$=[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\bar{x})^2\geq(y_j-\bar{x})^2)-\frac{1}{2}]\times[\frac{1}{2}-\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\bar{y})^2\geq(y_j-\bar{y})^2)]$$
$$+\frac{1}{2}[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\bar{x})^2\geq(y_j-\bar{x})^2)-I((x_i-\bar{y})^2\geq(y_j-\bar{y})^2)]$$

By Hoeffding decomposition, we have

$$\begin{aligned}V_1(\lambda)&=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\lambda)^2\geq(y_j-\lambda)^2)-\frac{1}{2}\\&=\frac{1}{m}\sum_{i=1}^{m}E_yI((x_i-\lambda)^2\geq(y_j-\lambda)^2)-\frac{1}{2}\\&+\frac{1}{n}\sum_{j=1}^{n}E_xI((x_i-\lambda)^2\geq(y_j-\lambda)^2)-\frac{1}{2}+Op(\frac{1}{mn}),\end{aligned}$$

and

$$\begin{aligned}V_2(\lambda)&=\frac{1}{2}-\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}I((x_i-\lambda)^2\geq(y_j-\lambda)^2)\\&=\frac{1}{2}-\frac{1}{m}\sum_{i=1}^{m}E_yI((x_i-\lambda)^2\geq(y_j-\lambda)^2)\\&+\frac{1}{2}-\frac{1}{n}\sum_{j=1}^{n}E_xI((x_i-\lambda)^2\geq(y_j-\lambda)^2)+Op(\frac{1}{mn}).\end{aligned}$$

By Taylor expansion and approximations to non-central chi-squared distribution [13],

$$\begin{aligned}&E_yI((x_i-\lambda)^2\geq(y_j-\lambda)^2)\\=&F(\frac{(x_i-\lambda)^2}{1+\lambda^2})+O_p(\lambda^{-3}),\text{ by [13]}\\=&F((x_i-\lambda)^2)+O_p(\lambda^{-3})+O_p(\lambda^{-2})\\=&F(x_i^2)-2x_i\lambda f(x_i^2)+O_p(\lambda^{-3})+O_p(\lambda^{-2})+O_p(\lambda^{-2}),\text{ Taylor expansion}\end{aligned}$$

and

$$\begin{aligned}&E_xI((x_i-\lambda)^2\geq(y_j-\lambda)^2)\\=&1-F(\frac{(y_j-\lambda)^2}{1+\lambda^2})+O_p(\lambda^{-3}),\text{ by [13]}\\=&1-F((y_j-\lambda)^2)+O_p(\lambda^{-3})+O_p(\lambda^{-2})\\=&1-F(y_j^2)+2\lambda y_jf(y_j^2)+O_p(\lambda^{-3})+O_p(\lambda^{-2})+O_p(\lambda^{-2}),\text{ Taylor expansion}\end{aligned}$$

Therefore,

$$V_1(\lambda) = \frac{1}{m}\sum_{i=1}^{m} F(x_i^2) - 2x_i\lambda f(x_i^2) - \frac{1}{2} + \frac{1}{n}\sum_{j=1}^{n} -F(y_j^2) + 2\lambda y_j f(y_j^2) + \frac{1}{2} + Op(\frac{1}{mn})$$

$$V_2(\lambda) = -\frac{1}{m}\sum_{i=1}^{m} F(x_i^2) - 2x_i\lambda f(x_i^2) - \frac{1}{2} - \frac{1}{n}\sum_{j=1}^{n} -F(y_j^2) + 2\lambda y_j f(y_j^2) + \frac{1}{2} + Op(\frac{1}{mn})$$

As $\bar{x} \to 0$, $V_1(\bar{x}) = \frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{x}f(x_i^2) - \frac{1}{2}] + \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{x}y_j f(y_j^2) + \frac{1}{2}]$.
As $\bar{y} \to 0$, $V_2(\bar{y}) = -\frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{y}f(x_i^2) - \frac{1}{2}] - \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{y}y_j f(y_j^2) + \frac{1}{2}]$.

$$\begin{aligned}
&V_1(\bar{x})V_2(\bar{y}) \\
=&[\frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{x}f(x_i^2) - \frac{1}{2}] + \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{x}y_j f(y_j^2) + \frac{1}{2}]] \\
\times&[-\frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{y}f(x_i^2) - \frac{1}{2}] - \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{y}y_j f(y_j^2) + \frac{1}{2}]] \\
=&-[\frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{x}f(x_i^2) - \frac{1}{2}] + \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{x}y_j f(y_j^2) + \frac{1}{2}]] \\
\times&[\frac{1}{m}\sum_{i=1}^{m}[F(x_i^2) - 2x_i\bar{y}f(x_i^2) - \frac{1}{2}] + \frac{1}{n}\sum_{j=1}^{n}[-F(y_j^2) + 2\bar{y}y_j f(y_j^2) + \frac{1}{2}]] \\
=&-[\bar{x}(\frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2)) + \frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2})] \\
\times&[\bar{y}(\frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2)) + \frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2})]
\end{aligned}$$

From the proof of Theorem 1, we have the following distributions,

$$Z_1 = \sqrt{\frac{mn}{m+n}}(\bar{x} - \bar{y}) \sim \mathcal{N}(0, 1),$$

and

$$Z_2 = \sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2)) \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi}).$$

The distribution of $\sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2}))$ is as follows:

Both $F(x_i^2)$ and $F(y_j^2)$ follows uniform distribution $\mathcal{U}(0,1)$ as $P(F(x_i^2) < x) = P(x_i^2 < F^{-1}(x)) = F(F^{-1}(x)) = x$. Then $E[F(x_i^2)] = E[F(y_j^2)] = \frac{1}{2}$ and $Var[F(x_i^2)] = Var[F(y_j^2)] = \frac{1}{12}$. Therefore, $E[F(x_i^2) - \frac{1}{2}] = E[-F(y_j^2) + \frac{1}{2}] = \frac{1}{2} - \frac{1}{2} = 0$, and $Var(F(x_i^2) - \frac{1}{2}) = Var(-F(y_j^2) + \frac{1}{2}) = \frac{1}{12}$.

Hence, $F(x_i^2) - \frac{1}{2} \sim \mathcal{N}(0, \frac{1}{12})$ and $-F(y_j^2) + \frac{1}{2} \sim \mathcal{N}(0, \frac{1}{12})$.

Therefore,

$$Z_3 = \sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2})) \sim \mathcal{N}(0, \frac{1}{12}).$$

Then, the normalized form of $V_1(\bar{x})V_2(\bar{y})$ can be written as

$$\begin{aligned} -\frac{mn}{m+n}V_1(\bar{x})V_2(\bar{y}) &= (\bar{x}Z_2 + Z_3)(\bar{y}Z_2 + Z_3) \\ &= \bar{x}\bar{y}Z_2^2 + (\bar{x} + \bar{y})Z_2Z_3 + Z_3^2 \\ &= Z_3^2 + O_p(\frac{1}{\sqrt{mn}}) + O_p(\frac{1}{\sqrt{m}}) + O_p(\frac{1}{\sqrt{n}}), \end{aligned}$$

From $S_{m,n}$, $-\frac{mn}{m+n}V(\bar{x},\bar{y}) \xrightarrow{d} -Z_1Z_2$, where $V(\bar{x},\bar{y}) = \frac{1}{2}[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I((x_i - \bar{x})^2 \geq (y_j - \bar{x})^2) - I((x_i - \bar{y})^2 \geq (y_j - \bar{y})^2)]$, $Z_1 = \sqrt{\frac{mn}{m+n}}(\bar{x} - \bar{y})$, and $Z_2 = \sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2))$.

Therefore,

$$P_{m,n} = -\frac{mn}{m+n}V_1(\bar{x})V_2(\bar{y}) - \frac{1}{2}\frac{mn}{m+n}V(\bar{x},\bar{y}) \xrightarrow{d} Z_3^2 - \frac{1}{2}Z_1Z_2,$$

where $Z_1 \sim \mathcal{N}(0,1)$, $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, and $Z_3 \sim \mathcal{N}(0, \frac{1}{12})$, with $Cov(Z_1, Z_2) = -\frac{1}{\pi}$. Z_3 is independent of Z_1 and Z_2.

The following calculation will show independence of Z_3 with Z_1 and Z_2.

First, we have

$$\begin{aligned} &E[x_i(F(x_i^2) - \frac{1}{2})] \\ =&\int_{-\infty}^{\infty} x_i(F(x_i^2) - \frac{1}{2})g(x_i), \text{ where } g(x_i) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x_i^2}{2}} \\ =&0 \end{aligned}$$

Then, $Cov(x_i, F(x_i^2) - \frac{1}{2}) = E[x_i(F(x_i^2) - \frac{1}{2})] - E[x_i]E[F(x_i^2) - \frac{1}{2}] = 0$. Similarly, $Cov(-F(y_j^2) + \frac{1}{2}, y_j) = 0$.

In the same way,

$$E[-2x_i f(x_i^2)(F(x_i^2) - \frac{1}{2})]$$

$$= \int_{-\infty}^{\infty} -2x_i f(x_i^2)(F(x_i^2) - \frac{1}{2})g(x_i), \text{ where } g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}}$$
$$=0,$$

Then, $Cov(F(x_i^2)-\frac{1}{2}, -2x_i f(x_i^2)) = E[-2x_i f(x_i^2)(F(x_i^2)-\frac{1}{2})]-E[F(x_i^2)-\frac{1}{2}]E[-2x_i f(x_i^2)] = 0$. Similarly, $Cov(-F(y_j^2)+\frac{1}{2}, 2y_j f(y_j^2)) = 0$.

$$\begin{aligned}
&Cov(Z_3, Z_1) \\
=&Cov(\sqrt{\frac{mn}{m+n}}(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2})), \sqrt{\frac{mn}{m+n}}(\bar{x} - \bar{y})) \\
=&\frac{mn}{m+n}Cov(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2}), \bar{x} - \bar{y}) \\
=&\frac{mn}{m+n}[Cov(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}), \bar{x}) - Cov(\frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2}), \bar{y})] \\
=&\frac{mn}{m+n}[\frac{1}{m}Cov(F(x_i^2) - \frac{1}{2}, x_i) - \frac{1}{n}Cov(-F(y_j^2) + \frac{1}{2}, y_j)] \\
=&0
\end{aligned}$$

and

$$\begin{aligned}
&Cov(Z_3, Z_2) \\
=&\frac{mn}{m+n}Cov(\frac{1}{m}\sum_{i=1}^{m}(F(x_i^2) - \frac{1}{2}) + \frac{1}{n}\sum_{j=1}^{n}(-F(y_j^2) + \frac{1}{2}), \frac{1}{m}\sum_{i=1}^{m} -2x_i f(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} 2y_j f(y_j^2)) \\
=&\frac{mn}{m+n}[\frac{1}{m}Cov(F(x_i^2) - \frac{1}{2}, -2x_i f(x_i^2)) + \frac{1}{n}Cov(-F(y_j^2) + \frac{1}{2}, 2y_j f(y_j^2))] \\
=&0.
\end{aligned}$$

Since all Z_1, Z_2, and Z_3 are normal distribution and $Cov(Z_3, Z_1) = 0$ and $Cov(Z_3, Z_2) = 0$, Z_3 is independent of Z_1 and Z_2.

□

Remark 2. The convergence rate $\frac{mn}{m+n}$ in $P_{m,n}$ apply to other multivariate distributions F and G and to all depths, which provides a further theoretical support for broader applications. Under regular conditions, we can obtain a form similar to that shown in Theorem 2.

$$P_{m,n} \approx \frac{mn}{m+n}\left\{\frac{1}{m}\sum_{i=1}^{m}[\frac{1}{2} - F_{D(y;F)}(D(x_i;F))] + \frac{1}{n}\sum_{j=1}^{n}[F_{D(x;F)}(D(y_j;F)) - \frac{1}{2}]\right\}^2 + \frac{1}{2}S_{m,n},$$

where $F_{D(x;F)}(\cdot)$ is the distribution function of $D(x;F)$ and $S_{m,n}$ is Sum Statistic.

For one-dimensional Euclidean depth (1.3), the extension is consistent with the result of the theorem.

Proof.

$$\begin{aligned}
P_{m,n} =& -\frac{mn}{m+n}[Q(F_m, G_n) + Q(G_n, F_m) - \frac{1}{4}] \\
=& -\frac{mn}{m+n}[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i; F_m) \le D(y_j; F_m)) \\
& \times \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j; G_n) \le D(x_i; G_n)) - \frac{1}{4}] \\
=& -\frac{mn}{m+n}\{[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i; F_m) \le D(y_j; F_m)) - \frac{1}{2}] \\
& \times [\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j; G_n) \le D(x_i; G_n)) - \frac{1}{2}] \\
& + \frac{1}{2}\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(D(x_i; F_m) \le D(y_j; F_m)) - I(D(x_i; G_n) \le D(y_j; G_n))]\}
\end{aligned}$$

By Hoeffding decomposition, we have

$$\begin{aligned}
& \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i; F_m) \le D(y_j; F_m)) - \frac{1}{2} \\
=& \frac{1}{m}\sum_{i=1}^{m} E_y I(D(x_i; F_m) \le D(y_j; F_m)) - \frac{1}{2} \\
+& \frac{1}{n}\sum_{j=1}^{n} E_x I(D(x_i; F_m) \le D(y_j; F_m)) - \frac{1}{2} + O_p(\frac{1}{mn}),
\end{aligned}$$

and

$$\begin{aligned}
& \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j; G_n) \le D(x_i; G_n)) - \frac{1}{2} \\
=& \frac{1}{m}\sum_{i=1}^{m} E_y I(D(y_j; G_n) \le D(x_i; G_n)) - \frac{1}{2} \\
+& \frac{1}{n}\sum_{j=1}^{n} E_x I(D(y_j; G_n) \le D(x_i; G_n)) - \frac{1}{2} + O_p(\frac{1}{mn}).
\end{aligned}$$

We have same two assumptions,

(1) Under H_0, $F = G$

(2) The limit of empirical distribution F_m is F.

According to above two assumptions and using Taylor expansions, we have

$$\begin{aligned}
&E_y I(D(x_i;F_m) \le D(y_j;F_m)) \\
\approx &1 - F_{D(y;F)}(D(x_i;F)) - f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;F)] \\
&E_y I(D(y_j;G_n) \le D(x_i;G_n)) \\
\approx &F_{D(y;F)}(D(x_i;F)) + f_{D(y;F)}(D(x_i;F))[D(x_i;G_n) - D(x_i;F)] \\
&E_x I(D(x_i;F_m) \le D(y_j;F_m)) \\
\approx &F_{D(x;F)}(D(y_j;F)) + f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;F)] \\
&E_x I(D(y_j;G_n) \le D(x_i;G_n)) \\
\approx &1 - F_{D(x;F)}(D(y_j;F)) - f_{D(x;F)}(D(y_j;F))[D(y_j;G_n) - D(y_j;F)]
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i;F_m) \le D(y_j;F_m)) - \frac{1}{2} \\
=&\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2} - F_{D(y;F)}(D(x_i;F)) - f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;F)] \\
+&\frac{1}{n}\sum_{j=1}^{n} F_{D(x;F)}(D(y_j;F)) + f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;F)] - \frac{1}{2} + O_p(\frac{1}{mn})
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j;G_n) \le D(x_i;G_n)) - \frac{1}{2} \\
=&\frac{1}{m}\sum_{i=1}^{m} F_{D(y;F)}(D(x_i;F)) + f_{D(y;F)}(D(x_i;F))[D(x_i;G_n) - D(x_i;F)] - \frac{1}{2} \\
+&\frac{1}{n}\sum_{j=1}^{n}\frac{1}{2} - F_{D(x;F)}(D(y_j;F)) - f_{D(x;F)}(D(y_j;F))[D(y_j;G_n) - D(y_j;F)] + O_p(\frac{1}{mn})
\end{aligned}$$

By Assumption (2) the limiting distribution of empirical distribution F_m is F, we have the following approximation

$$\begin{aligned}
&[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(x_i;F_m) \le D(y_j;F_m)) - \frac{1}{2}][\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I(D(y_j;G_n) \le D(x_i;G_n)) - \frac{1}{2}] \\
\approx&[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2} - F_{D(y;F)}(D(x_i;F)) - f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;F)] \\
&+ \frac{1}{n}\sum_{j=1}^{n} F_{D(x;F)}(D(y_j;F)) + f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;F)] - \frac{1}{2}]
\end{aligned}$$

$$
\begin{aligned}
&[\frac{1}{m}\sum_{i=1}^{m} F_{D(y;F)}(D(x_i;F)) + f_{D(y;F)}(D(x_i;F))[D(x_i;G_n) - D(x_i;F)] - \frac{1}{2} \\
&+ \frac{1}{n}\sum_{j=1}^{n} \frac{1}{2} - F_{D(x;F)}(D(y_j;F)) - f_{D(x;F)}(D(y_j;F))[D(y_j;G_n) - D(y_j;F)]] \\
\approx &- [\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2} - F_{D(y;F)}(D(x_i;F)) + \frac{1}{n}\sum_{j=1}^{n} F_{D(x;F)}(D(y_j;F)) - \frac{1}{2}]^2
\end{aligned}
$$

Therefore, $P_{m,n}$ can be written in the form

$$
\begin{aligned}
P_{m,n} \approx &- \frac{mn}{m+n}\Big\{ - [\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2} - F_{D(y;F)}(D(x_i;F)) + \frac{1}{n}\sum_{j=1}^{n} F_{D(x;F)}(D(y_j;F)) - \frac{1}{2}]^2 \\
&+ \frac{1}{2}\Big[\frac{1}{m}\sum_{i=1}^{m} -f_{D(y;F)}(D(x_i;F))[D(x_i;F_m) - D(x_i;G_n)] \\
&+ \frac{1}{n}\sum_{j=1}^{n} f_{D(x;F)}(D(y_j;F))[D(y_j;F_m) - D(y_j;G_n)]\Big]\Big\} \\
= &\frac{mn}{m+n}\left\{\frac{1}{m}\sum_{i=1}^{m}[\frac{1}{2} - F_{D(y;F)}(D(x_i;F))] + \frac{1}{n}\sum_{j=1}^{n}[F_{D(x;F)}(D(y_j;F)) - \frac{1}{2}]\right\}^2 + \frac{1}{2}S_{m,n}
\end{aligned}
$$

For $P_{m,n}$ under one-dimensional Euclidean depth, we can use Euclidean distance to replace the depth functions. Therefore,

$$
\begin{aligned}
P_{m,n} \approx &- \frac{mn}{m+n}\Big\{ - [\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2} - F_{d(y;F)}(d(x_i;F)) + \frac{1}{n}\sum_{j=1}^{n} F_{d(x;F)}(d(y_j;F)) - \frac{1}{2}]^2 \\
&+ \frac{1}{2}\Big[\frac{1}{m}\sum_{i=1}^{m} f_{d(y;F)}(d(x_i;F))[d(x_i;F_m) - d(x_i;G_n)] \\
&+ \frac{1}{n}\sum_{j=1}^{n} -f_{d(x;F)}(d(y_j;F))[d(y_j;F_m) - d(y_j;G_n)]\Big]\Big\},
\end{aligned}
$$

where $d()$ represents Euclidean distance, and $F_{d(x;F)}$ and $f_{d(x;F)}$ are CDF and PDF of distribution $d(x;F)$.

Then we have $d(x_i;F_m) = (x_i - \bar{x})^2, d(y_j;F_m) = (y_j - \bar{x})^2, d(x_i;G_n) = (x_i - \bar{y})^2, d(y_j;G_n) = (y_j - \bar{y})^2, d(x_i;F) = x_i^2, d(y_j;F) = y_j^2$. Then, $F_{d(y;F)}(d(x_i;F)) = F(x_i^2)$, $F_{d(x;F)}(d(y_j;F)) = F(y_j^2)$, $f_{d(y;F)}(d(x_i;F)) = f(x_i^2)$, and $f_{d(x;F)}(d(y_j;F)) = f(y_j^2)$, where F is CDF of χ_1^2 and f is density of χ_1^2.

$$
\begin{aligned}
P_{m,n} \approx &\frac{mn}{m+n}[\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2} - F(x_i^2) + \frac{1}{n}\sum_{j=1}^{n} F(y_j^2) - \frac{1}{2}]^2 \\
&- \frac{1}{2}\frac{mn}{m+n}\Big[\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)[(x_i - \bar{x})^2 - (x_i - \bar{y})^2] + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)[(y_j - \bar{x})^2 - (y_j - \bar{y})^2]\Big]
\end{aligned}
$$

Since we have $\sqrt{\frac{mn}{m+n}}[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2} - F(x_i^2) + \frac{1}{n}\sum_{j=1}^{n}F(y_j^2) - \frac{1}{2}] \sim \mathcal{N}(0, \frac{1}{12})$, and $S_{m,n} \approx -\frac{mn}{m+n}\Big[\frac{1}{m}\sum_{i=1}^{m} f(x_i^2)[(x_i-\bar{x})^2 - (x_i-\bar{y})^2] + \frac{1}{n}\sum_{j=1}^{n} -f(y_j^2)[(y_j-\bar{x})^2 - (y_j-\bar{y})^2]\Big]$, then $P_{m,n} \xrightarrow{d} Z_3^2 - \frac{1}{2}Z_1Z_2$, where $Z_1 \sim \mathcal{N}(0,1)$, $Z_2 \sim \mathcal{N}(0, \frac{2}{\sqrt{3}\pi})$, and $Z_3 \sim \mathcal{N}(0, \frac{1}{12})$, with $Cov(Z_1, Z_2) = -\frac{1}{\pi}$. Z_3 is independent of Z_1 and Z_2.

□

Properties of $P_{m,n}$

PDF and CDF

The cumulative distribution function of $P_{m,n}$ is:

$$F_P(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[\int_0^{12x+6z_1z_2}\frac{1}{\sqrt{2\pi}}t^{-\frac{1}{2}}e^{-\frac{t}{2}}\,dt]\frac{1}{2\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}}e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2+\sqrt{3}z_1z_2+\frac{\sqrt{3}\pi}{2}z_2^2)}\,dz_1\,dz_2.$$

It is derived as:

$$\begin{aligned}
F_P(x) &= P(z_3^2 - \frac{1}{2}z_1z_2 < x) \\
&= \iiint_{z_3^2-\frac{1}{2}z_1z_2<x}\frac{1}{\sqrt{2\pi}\sqrt{\frac{1}{12}}}e^{-\frac{1}{2}(12z_3^2)}f(z_1,z_2)\,dz_1\,dz_2\,dz_3 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(12x+6z_1z_2)f(z_1,z_2)\,dz_1\,dz_2, \text{ where } F \text{ is } \chi_1^2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[\int_0^{12x+6z_1z_2}\frac{1}{\sqrt{2\pi}}t^{-\frac{1}{2}}e^{-\frac{t}{2}}\,dt]\frac{1}{2\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}}e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2+\sqrt{3}z_1z_2+\frac{\sqrt{3}\pi}{2}z_2^2)}\,dz_1\,dz_2
\end{aligned}$$

The $F(12x+6z_1z_2)$ is obtained by $E[I(z_3^2 - \frac{1}{2}z_1z_2 < x) = E[I(z_3^2 < \frac{1}{2}z_1z_2 + x) = F(12x+6z_1z_2)$, as $Z_3 \sim \mathcal{N}(0, \frac{1}{12})$.

The probability density function of $P_{m,n}$ can be written as $f_P(x)$, which is the derivative of $F_P(x)$ with respect to x:

$$f_P(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}12f_{\chi^2}(12x+6z_1z_2)\frac{1}{2\pi\sqrt{\frac{2\pi-\sqrt{3}}{\sqrt{3}\pi^2}}}e^{-\frac{1}{2-\frac{\sqrt{3}}{\pi}}(z_1^2+\sqrt{3}z_1z_2+\frac{\sqrt{3}\pi}{2}z_2^2)}\,dz_1\,dz_2,$$

where $f_{\chi^2}(12x+6z_1z_2)$ is probability density of χ_1^2 at $12x+6z_1z_2$.

We plot $f_P(x)$ in Fig. 2.3, compared with simulated density with $m = n = 1000$ and 10,000 repetitions. It can be seen that the densities are skewed to the right and the peaks are sharp.

Expectation and Variance

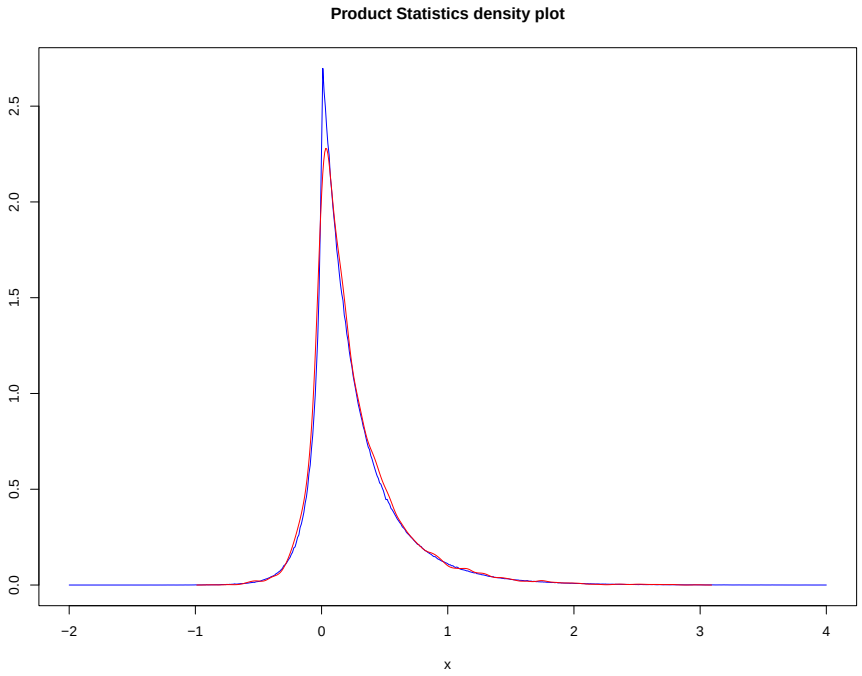


Figure 2.3: The true (blue) probability density functions vs. the estimated (red) density function for $f_P(x)$.

Since $P_{m,n} \xrightarrow{d} Z_3^2 - \frac{1}{2}Z_1Z_2$, the expectation is $E[Z_3^2 - \frac{1}{2}Z_1Z_2] = E[Z_3^2] - E[\frac{1}{2}Z_1Z_2] = \frac{1}{12} - \frac{1}{2}(-\frac{1}{\pi}) = \frac{1}{12} + \frac{1}{2\pi}$.

Since $E[Z_3^2] = E[Z_3^4] - E[Z_3^2]^2 = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\frac{1}{12}}} e^{-\frac{x^2}{2\frac{1}{12}}} x^4\, dx - (\frac{1}{12})^2 = \frac{1}{48} - (\frac{1}{12})^2 = \frac{1}{72}$, the variance can be computed as $Var(Z_3^2 - \frac{1}{2}Z_1Z_2) = Var(Z_3^2) - \frac{1}{4}Var(Z_1Z_2) = \frac{1}{72} + \frac{1}{4}(\frac{2}{\sqrt{3}\pi} + \frac{1}{\pi^2})$.

Note that we show $S_{m,n}$ and $P_{m,n}$ are related to each other, we can also calculate their covariance, which is

$$\begin{aligned} Cov(P_{m,n}, S_{m,n}) &= Cov(Z_3^2 - \frac{1}{2}Z_1Z_2, -Z_1Z_2) \\ &= \frac{1}{4}Cov(Z_1Z_2, Z_1Z_2) \\ &= \frac{1}{4}Var(Z_1Z_2) \\ &= \frac{1}{4}(\frac{2}{\sqrt{3}\pi} + \frac{1}{\pi^2}). \end{aligned}$$

2.3.3 Minimum Statistic

Theorem 3. Consider two independent and identical (iid) samples of the random variables $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ from the normal distributions F and G, and denote the corresponding empirical distributions as F_m and G_n, respectively. Under the

null hypothesis $F = G$, the asymptotic distribution of $M^*_{m,n} = \left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-\frac{1}{2}}(\frac{1}{2} - \min(Q(F_m,G_n),Q(G_n,F_m)))$ is as follows, applicable for all depth functions:

$$M^*_{m,n} \xrightarrow{d} |\mathcal{N}(0,1)|. \tag{2.9}$$

Proof. By [31, 50, 41], we have the property of Q statistics:

$$Q(G_n,F_m) - 1/2 = 1/2 - Q(F_m,G_n) + o_p(n^{-1/2}) + o_p(m^{-1/2}),$$

and [50] showed that under null hypothesis

$$\left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-\frac{1}{2}}(Q(F_m,G_n)-\frac{1}{2}) \xrightarrow{d} \mathcal{N}(0,1).$$

We have

$$\begin{aligned}
&\min(Q(F_m,G_n),Q(G_n,F_m)) \\
=&\min(Q(F_m,G_n)-\frac{1}{2},Q(G_n,F_m)-\frac{1}{2})+\frac{1}{2} \\
=&-\left|Q(F_m,G_n)-\frac{1}{2}\right|+\frac{1}{2}, \text{under } H_0
\end{aligned}$$

Hence,

$$\begin{aligned}
M^*_{m,n} &= \left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-\frac{1}{2}}(\frac{1}{2}-\left|Q(F_m,G_n)-\frac{1}{2}\right|-\frac{1}{2}) \\
&= \left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-\frac{1}{2}}\left|Q(F_m,G_n)-\frac{1}{2}\right| \xrightarrow{d} |\mathcal{N}(0,1)|
\end{aligned}$$

We could also show that $(M^*_{m,n})^2 \xrightarrow{d} \chi^2_1$.
From previous part, we have $(M^*_{m,n})^2 = \left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-1}\left(Q(F_m,G_n)-\frac{1}{2}\right)^2$.
By [50], we have

$$\left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-\frac{1}{2}}(Q(F_m,G_n)-\frac{1}{2}) \xrightarrow{d} \mathcal{N}(0,1),$$

which means

$$\left[\frac{1}{12}(\frac{1}{m}+\frac{1}{n})\right]^{-1}((Q(F_m,G_n)-\frac{1}{2}))^2 \xrightarrow{d} \chi^2_1.$$

Therefore, $(M^*_{m,n})^2 \xrightarrow{d} \chi^2_1$. □

α	100	200	300	400	500	600	700	800	900	1000	∞
0.2	0.6500	0.6525	0.6567	0.6200	0.6352	0.6425	0.6521	0.6650	0.6261	0.6561	0.6417
0.1	1.1150	1.1128	1.1483	1.1051	1.1021	1.1393	1.1301	1.1600	1.1253	1.1390	1.1312
0.05	1.6150	1.6100	1.6601	1.6350	1.6171	1.6959	1.6237	1.7025	1.6250	1.6523	1.6566
0.01	2.8052	2.7803	2.8767	2.9901	2.8530	2.9526	2.8629	2.9463	2.9372	2.9671	2.9608

Table 2.1: Table of empirical quantiles vs. theoretical quantiles of $S_{m,n}$ for different α with $m = n = 100, \ldots, 1000$

α	100	200	300	400	500	600	700	800	900	1000	∞
0.2	0.6533	0.6433	0.6467	0.6517	0.6547	0.6289	0.6514	0.6392	0.6639	0.6453	0.6417
0.1	1.1007	1.1467	1.1378	1.1383	1.1293	1.0812	1.1391	1.1283	1.1661	1.1367	1.1312
0.05	1.6203	1.6533	1.6313	1.6650	1.5947	1.5867	1.6286	1.6578	1.6853	1.6093	1.6566
0.01	2.7733	2.8501	2.8089	2.9218	2.9653	2.8023	2.8420	2.8934	2.9459	2.9187	2.9608

Table 2.2: Table of empirical quantiles vs. theoretical quantiles of $S_{m,n}$ for different α with $m = 2n = 100, \ldots, 1000$

2.3.4 Convergence rate

Sum Statistic

To further demonstrate the rate of convergence for distributions of $S_{m,n}$, we conduct a simulation study. Assume there are two equal distributions $F = G = \mathcal{N}(0, 1)$ with mean 0 and variance 1, with sample size m and n respectively. Consider sample size m varying from 100 to 1000, i.e., $m = 100, 200, \ldots, 1000$, and two cases of sample size n with either $n = m$ or $n = m/2$. As we proved the asymptotic distribution of $S_{m,n}$ under Euclidean depth, we repeat the generation 10,000 times and compute empirical α quantiles, with $\alpha = 0.2, 0.1, 0.05, 0.01$. The empirical quantiles are then compared with the theoretical quantiles by assuming $m, n \to \infty$, based on the asymptotic distribution of $S_{m,n}$ specified in (2.7). This comparison is intended to quantitatively assess how well the finite sample distributions of $S_{m,n}$ match their respective asymptotic behaviors, concluded in Table 2.1, 2.2 and Figure 2.4.

The plot of the comparison of empirical quantiles of $S_{m,n}$ for different $1 - \alpha$ quantiles: 80% (Row 1), 90% (Row 2), 95% (Row 3), 99% (Row 4) are shown in Figure 2.4, comparing with theoretical quantiles (in red). The detailed values of empirical quantiles vs. theoretical quantiles are also summarized in Table 2.1 and 2.2.

As shown in Tables 2.1 and 2.2, there is significant agreement between empirical and theoretical quantiles at different α levels. For all evaluated values, the empirical quantiles are very close to the theoretical quantiles, except for $\alpha = 0.01$ which requires a larger sample size. This observation holds true even with relatively small sample sizes, thus demonstrating the fast convergence rate of the asymptotic distribution of $S_{m,n}$ and in approximating the behavior of their finite sample counterparts.

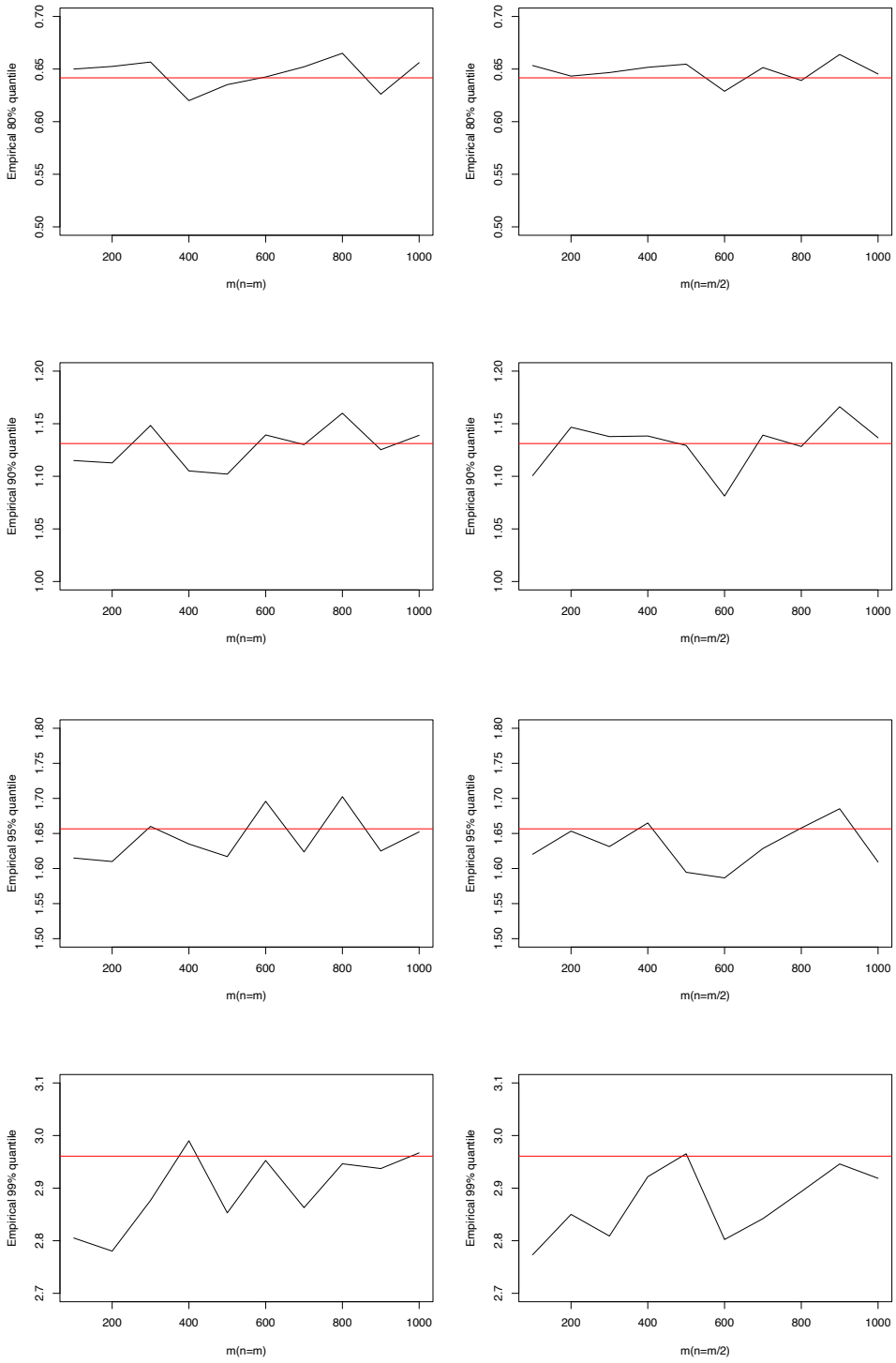


Figure 2.4: Comparison of empirical quantiles of sum statistic under one dimensional Euclidean depth for $m = 100, \ldots, 1000$ with $n = m$(1st column) or $n = m/2$(2nd column) for different $1 - \alpha$ quantiles: 80% (Row 1), 90% (Row 2), 95% (Row 3), 99% (Row 4). The red line denotes the theoretical quantile.

α	100	200	300	400	500	600	700	800	900	1000	∞
0.2	0.4342	0.4372	0.4471	0.4331	0.4324	0.4400	0.4470	0.4528	0.4234	0.4457	0.4379
0.1	0.6656	0.6729	0.6869	0.6714	0.6701	0.6775	0.6763	0.7014	0.6794	0.6840	0.6818
0.05	0.9008	0.9187	0.9402	0.9232	0.9337	0.9532	0.9204	0.9606	0.9357	0.9452	0.9384
0.01	1.4527	1.5048	1.5473	1.5930	1.5522	1.5883	1.5222	1.5673	1.5816	1.5739	1.5706

Table 2.3: Table of empirical quantiles vs. theoretical quantiles of Product Statistics for different α with $m = n = 100, \ldots, 1000$

α	100	200	300	400	500	600	700	800	900	1000	∞
0.2	0.4398	0.4308	0.4234	0.4540	0.4289	0.4432	0.4358	0.4432	0.4373	0.4360	0.4379
0.1	0.6568	0.6651	0.6746	0.6992	0.6703	0.6988	0.6795	0.6828	0.6827	0.6821	0.6818
0.05	0.8674	0.9008	0.9156	0.9493	0.9115	0.9355	0.9195	0.9301	0.9370	0.9401	0.9384
0.01	1.391	1.4810	1.5149	1.5050	1.5231	1.5659	1.5666	1.5227	1.6081	1.6014	1.5706

Table 2.4: Table of empirical quantiles vs. theoretical quantiles of Product Statistics for different α with $m = 2n = 100, \ldots, 1000$

Product Statistic

Similar to $S_{m,n}$, the same procedure is applied on $P_{m,n}$ to show its convergence rate. With the same condition of distributions and sample sizes, we generated 10,000 repetitions to compute the empirical quantiles of Product Statistic for different α values ($\alpha = 0.2, 0.1, 0.05, 0.01$) and different sample sizes ($n = m$ and $n = m/2$), comparing with theoretical quantiles (when assuming $m, n \to \infty$). The plot of empirical quantiles is presented in Figure 2.5, comparing with theoretical quantile (in red line). The detailed values are also summarized in Tables 2.3 and 2.4.

For all evaluated values, the empirical quantiles of $P_{m,n}$ are close to the theoretical quantiles even when sample size is small for all α values, except for $\alpha = 0.01$, which requires a larger sample size. The conclusion is same as $S_{m,n}$ that the convergence rate of asymptotic distribution of $P_{m,n}$ is fast.

Minimum Statistic

Since we proved the asymptotic distribution of $M^*_{m,n}$, which is applicable for all depth functions, we make comparisons of convergence rate of $M^*_{m,n}$ for multivariate cases with these three depth functions: Mahalanobis depth [31], Spatial depth [7, 20], and Projection depth [29].

We considered random samples $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ from distributions F and G, respectively, with sample sizes m and n. Assume we have two equal distributions $F = G = N(\mathbf{0}, I_{2\times 2})$, where $N(\mathbf{0}, I_{2\times 2})$ represents the bivariate normal distribution with mean vector $\mathbf{0}$ and two-by-two identity covariance matrix. We varied the sample size m from 100 to 1000, with n set as either m or $m/2$. Since we proved that the minimum statistic $M^*_{m,n}$ follows a half-normal asymptotic null distribution, we used the upper 95%

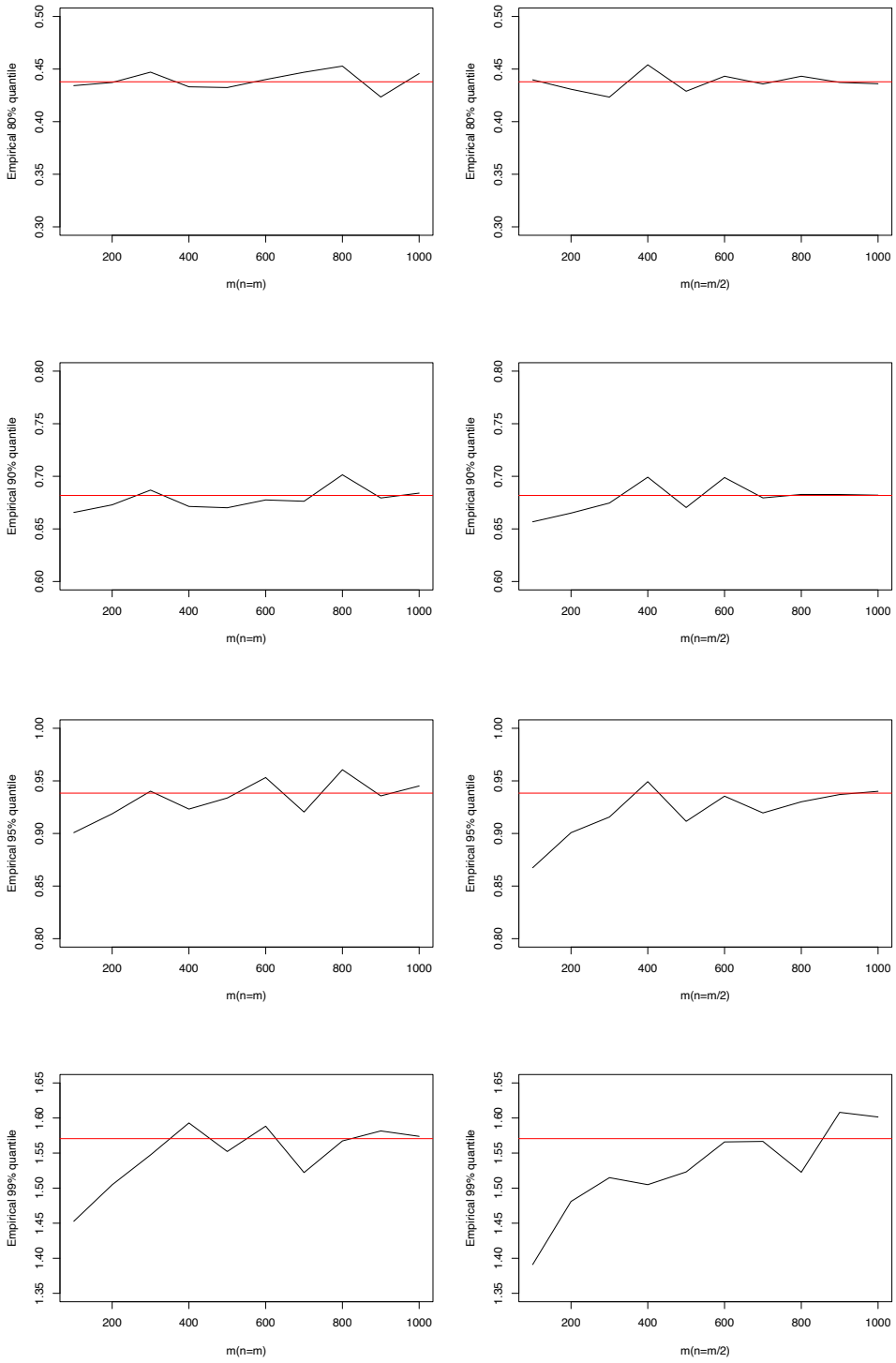


Figure 2.5: Comparison of empirical quantiles of product statistic under one dimensional Euclidean depth for $m = 100, \ldots, 1000$ with $n = m$(1st column) or $n = m/2$(2nd column) for different $1 - \alpha$ quantiles: 80% (Row 1), 90% (Row 2), 95% (Row 3), 99% (Row 4). The red line denotes the theoretical quantile

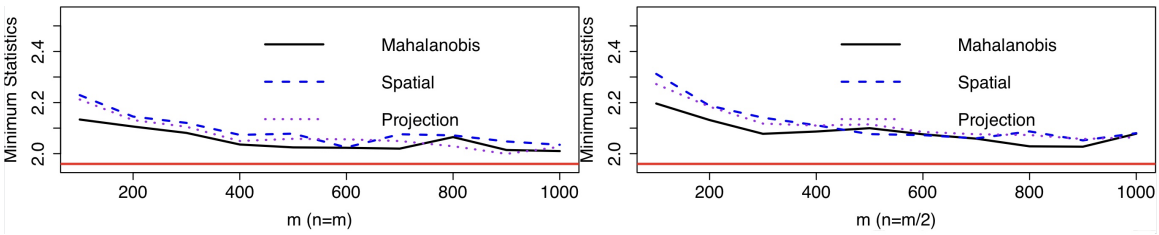


Figure 2.6: Comparison of empirical 95% quantiles of minimum statistic M_n for $m = 100, 200, \ldots, 1000$ and $n = m$ (1st column) or $n = m/2$ (2nd column).

quantile (1.96) as the theoretical quantile in this case. We conducted simulations with 10,000 repetitions and plotted the empirical quantiles of the Minimum Statistic in Figure 2.6. The empirical quantiles based on different values of m, n, and different depth functions were compared with theoretical quantiles. From the results, we observed that the Mahalanobis depth function had the fastest convergence rate compared to the Spatial and Projection depth functions.

2.4 Permutation Algorithm

The previous subsection introduces the asymptotic distributions $S_{m,n}$ and $P_{m,n}$. This subsection extends the discussion to the power of these statistics under permutation testing approach, which does not require the estimation of parameters in the asymptotic distribution and can be applied to other depth measures such as Mahalanobis depth [31], Spatial depth [7, 20], and Projection depth [29], with further simulation studies detailed in Chapter 3.

Acknowledging the permutation tests are inherently time-intensive, we employ the Strategic Block Permutation Algorithm [46, 11]. Initially, the raw test statistic T such as $S_{m,n}$ and $P_{m,n}$ is calculated from the empirical distributions F_m and G_n. We divide all samples in F_m into b_1 blocks of size s, i.e., $b_1 = \frac{m}{s}$, and all samples in G_n into b_2 blocks, i.e., $b_2 = \frac{n}{s}$. Assume that the total number of blocks for all samples $x_1, \ldots, x_m, y_1, \ldots, y_n$ is $N = b_1 + b_2$. Combining all N sample blocks together denotes all sample blocks as $Z = (Z_1, \ldots, Z_{b_1}, Z_{b_1+1}, \ldots, Z_{b_1+b_2})$, where the first b_1 sample blocks come from F_m and the second b_2 sample blocks come from G_n. Then, by randomizing all the blocks, there is a total of $N!$ permutations, denoting the set of all permutations as $(\pi(1), \ldots, \pi(N))$. After randomizing all blocks, we have $\underset{\sim}{Z} = (Z_{\pi(1)}, \ldots, Z_{\pi(N)})$. Considering the first b_1 blocks as $\tilde{F}_m$ and the next b_2 blocks as the $\tilde{G}_n$, we derive the new test statistic T^* from the $\tilde{F}_m$ and $\tilde{G}_n$. By calculating all T^* values, in a one-sided test, the p-value is calculated based on the proportion of all T^* for which $T^* > T$.

The pseudo-code for the Strategic Block Permutation Algorithm to compute the p-value is as follows:

Algorithm 1 Pseudo-code for the Strategic Block Permutation Algorithm

Determine $S^0_{m,n}$ and $P^0_{m,n}$ test statistic values from F_m and G_n.
Set block size s and number of repetitions $\mathcal{C}$.
Calculate the total number of blocks $B = (m+n)/s$.
for $i = 1$ to $\mathcal{C}$ **do**
  Permute all B blocks,
  Produce $\tilde{F}_m$ from the first b_1 permuted blocks,
  Produce $\tilde{G}_n$ from remaining blocks,
  Calculate new $P^*_{m,n}$, $S^*_{m,n}$ values from $\tilde{F}_m$ and $\tilde{G}_n$.
end for
Let p-value$_S$=$\sum[I(S^*_{m,n} > S^0_{m,n})]/\mathcal{C}$.
Let p-value$_P$=$\sum[I(P^*_{m,n} > P^0_{m,n})]/\mathcal{C}$.
Output p-value$_S$ and p-value$_P$.

At a predetermined significance level of α, for the Sum statistic, we reject the null hypothesis if p-value$_S < \alpha$. As with the Product statistic, we reject the null hypothesis if p-value$_P < \alpha$.

To discuss the power of permutation test, we make the following mild condition.

Assumption A. Under alternative hypothesis, $F \neq G$, such that $||\boldsymbol{\theta}(F) - \boldsymbol{\theta}(G)|| \neq 0$, with $\boldsymbol{\theta}(F)$ and $\boldsymbol{\theta}(G)$ are the parameters of the distributions F and G, respectively, we further suppose $-[Q(F_m, G_n) + Q(G_n, F_m) - 1]$ and $-[Q(F_m, G_n)Q(G_n, F_m) - 1/4]$ can be approximated by $q(||\boldsymbol{\theta}(F) - \boldsymbol{\theta}(G)||) + o_p(1)$, where $q(x)$ is a monotonically increasing function of x.

To justify this assumption, consider Assumption A, where $\boldsymbol{\theta}(F) = E(X) = \mu_1 \neq \boldsymbol{\theta}(G) = E(Y) = \mu_2$, with X and Y adhering to normal distributions F and G, respectively, each with a variance of 1. By Taylor expansions of X and Y around their expectation $E(X) = \mu_1$ and $E(Y) = \mu_2$ respectively, we have

$$\begin{aligned}
\frac{m+n}{mn} S_{m,n} &= -\left[E\left\{1 - F_{\chi^2}[(Y-\mu_1)^2]\right\} - E\left\{[F_{\chi^2}[(X-\mu_2)^2]\right\}\right] + o_p(1), \\
&= -1 + 2F_{\chi^2}[(\mu_2-\mu_1)^2] + o_p(1), \\
\frac{m+n}{mn} P_{m,n} &= -\left[E\left\{1 - F_{\chi^2}[(Y-\mu_1)^2]\right\} E\left\{1 - F[(X-\mu_2)^2]\right\} - 1/4\right] \\
&= -\left\{1 - F_{\chi^2}[(\mu_2-\mu_1)^2]\right\}^2 + 1/4 + o_p(1),
\end{aligned}$$

where F_{χ^2} denotes the distribution function of χ^2_1. It is evident that both $-1 + 2F_{\chi^2}[(\mu_2 - \mu_1)^2]$ and $-\left\{1 - F_{\chi^2}[(\mu_2-\mu_1)^2]\right\}^2 + 1/4$ are monotonically increasing functions of $(\mu_2 - \mu_1)^2$.

For multivariate distributions, we draw upon Theorem 6.1 in [31] to extend our analysis. The statistics $-[Q(F_m, G_n) + Q(G_n, F_m) - 1]$ and $-[Q(F_m, G_n)Q(G_n, F_m) - 1/4]$ can be represented as $1 - Q(F, G) - Q(G, F) + o_p(1)$ and $1/4 - Q(F, G)Q(G, F) + o_p(1)$, respectively.

The focus then shifts to demonstrating that

$$Q[(1-\beta)F+\beta G,(1-\beta)G+\beta F]+Q[(1-\beta)G+\beta F,(1-\beta)F+\beta G]>Q(F,G)+Q(G,F)$$

and

$$Q[(1-\beta)F+\beta G,(1-\beta)G+\beta F]Q[(1-\beta)G+\beta F,(1-\beta)F+\beta G]>Q(F,G)Q(G,F)$$

for $0<\beta<1$. As argued by [31], Q decreases if there is a location shift or a scale increase, or both in terms of contamination:

$$\begin{aligned}Q(F,G)&<Q[F,(1-\beta)G+\beta F]\\&<Q[(1-\gamma)F+\gamma\{(1-\beta)G+\beta F\},(1-\beta)G+\beta F]\\&=Q[(1-\beta)F+\beta G,(1-\beta)G+\beta F],\end{aligned}$$

where $\gamma=\beta/(1-\beta)$. A similar argument holds for $G(G,F)$, leading to

$$Q(G,F)<Q[G,(1-\beta)F+\beta G]<Q[(1-\beta)G+\beta F,(1-\beta)F+\beta G].$$

Therefore, the following Theorem 4 follows too.

Theorem 4. Assume Assumption A holds, under the alternative hypothesis, the power of the permuted Sum or Product tests at the significance level α approaches 1 as both the block size s and the number of repetitions $\mathcal{C}$ in the Strategic Block Permutation Algorithm go to infinity.

Proof. As each block from the same distribution are the same, we can express $||\boldsymbol{\theta}(F)-\boldsymbol{\theta}(G)||$ as $||\sum_{i=1}^{b_1}\boldsymbol{\theta}(F_i)/b_1-\sum_{j=1}^{b_2}\boldsymbol{\theta}(G_{b_1+j})/b_2||$, where F_i and G_{b_1+j} are distributions of $i-$th block and b_1+j-th block in combined samples $x_1,\ldots,x_m,y_1,\ldots,y_n$, respectively. For the permutation $(\pi(1),\ldots,\pi(N))$, we have $||\sum_{i=1}^{b_1}\boldsymbol{\theta}(F_{\pi(i)})/b_1-\sum_{j=1}^{b_2}\boldsymbol{\theta}(G_{\pi(b_1+j)})/b_2||$, which can be expressed as

$$||\frac{(b_1-c)\boldsymbol{\theta}(F)+c\boldsymbol{\theta}(G)}{b_1}-\frac{(b_2-c)\boldsymbol{\theta}(G)+c\boldsymbol{\theta}(F)}{b_2}||=||\boldsymbol{\theta}(F)-\boldsymbol{\theta}(G)||\,|1-\frac{c}{b_1}-\frac{c}{b_2}|,$$

where $0\le c\le\min(b_1,b_2)$.

We note that

$$||\sum_{i=1}^{b_1}\boldsymbol{\theta}(F_i)/b_1-\sum_{j=1}^{b_2}\boldsymbol{\theta}(G_{b_1+j})/b_2||=\sum_{i=1}^{b_1}\boldsymbol{\theta}(F_{\pi(i)})/b_1-\sum_{j=1}^{b_2}\boldsymbol{\theta}(G_{\pi(b_1+j)})/b_2||$$

with probability less than $\frac{2m!n!}{(m+n)!}$, i.e., the permutations occur only within each sample with probability $\frac{m!n!}{(m+n)!}$ for $c=0$ or two samples are exchanged when $m=n$ with probability

$\frac{m!m!}{(2m)!}$ for $c = b_1 = b_2$. Except these case, for $0 < c < \min(b_1, b_2)$. we have

$$||\sum_{i=1}^{b_1} \boldsymbol{\theta}(F_i)/b_1 - \sum_{j=1}^{b_2} \boldsymbol{\theta}(G_{b_1+j})/b_2|| > ||\sum_{i=1}^{b_1} \boldsymbol{\theta}(F_{\pi(i)})/b_1 - \sum_{j=1}^{b_2} \boldsymbol{\theta}(G_{\pi(b_1+j)})/b_2||.$$

That's because $-1 < 1 - \frac{\min(b_1,b_2)-1}{b_1} - \frac{\min(b_1,b_2)-1}{b_2} \leq 1 - \frac{c}{b_1} - \frac{c}{b_2} \leq 1 - \frac{1}{b_1} - \frac{1}{b_2} < 1$.

As $s \to \infty$, under Assumption A1, the permuted new $S_{m,n}$ and $P_{m,n}$ are less than their original ones in probability. In addition, the p-value$_S$ and p-value$_P$ converge to the probabilities of permuted new $S_{m,n}$ and $P_{m,n}$ greater than their original ones as $\mathcal{C} \to \infty$, which is zero and less than α. So, Theorem 4 follows.

□

Chapter 3

Power Comparisons

In this Chapter, we compare the power of our proposed test statistics, Sum statistic $S_{m,n}$ (2.5) and the Product statistic $P_{m,n}$ (2.4), with other existing methods to show their power performance for two cases: univariate distribution, and multivariate distribution. For univariate cases, we also compare with a popular test statistics, Wilcoxon rank sum test (Wilcoxon) [47], which we introduced its property and asymptotic distribution on Chapter 1.

3.1 Univariate distribution

We conducted power comparisons between the Sum statistic $S_{m,n}$ (2.5) and the Product statistic $P_{m,n}$ (2.4) alongside of other existing methods: $M_{m,n}$ (2.1), $M^*_{m,n}$ (2.3), Depth-Based Rank Statistics (DbR) [10], Modified Depth-Based Rank Statistics (BDbR) [3], and the Wilcoxon rank sum test (Wilcoxon) [47]. These comparisons were made within the framework of one-dimensional Euclidean depth.

We set the significance level at $\alpha = 0.05$. For $M_{m,n}$, $M^*_{m,n}$, DbR and BDbR, critical values were determined as the upper 95% quantiles from 1000 replications under the null hypothesis ($F = G = \mathcal{N}(0, 1)$). Power was then calculated as the proportion of instances across 1000 repetitions where the test statistic exceeded these critical values. For permutation tests based on $P_{m,n}$ and $S_{m,n}$ using the Strategic Block Permutation algorithm, we set the threshold for the p-value to be the lower 5% quantile of the simulated 1000 p-values under the null hypotheses, with the number of repetitions $\mathcal{C} = 200$ and block size $s = 25$. The power of the permutation test is then calculated using the proportion of times in 1000 repetitions that the statistic is less than the lower 5% quantile.

We consider three distinct scenarios: change in scale, change in mean, and change in both scale and mean. The results of these power comparisons are depicted in Figure 3.1, where rows in a column correspond to the three scenarios, while columns in the same row representing varying sample sizes, specifically, $m = n = 50, \cdots, 500$ (left) and $n = m/2$ (right).

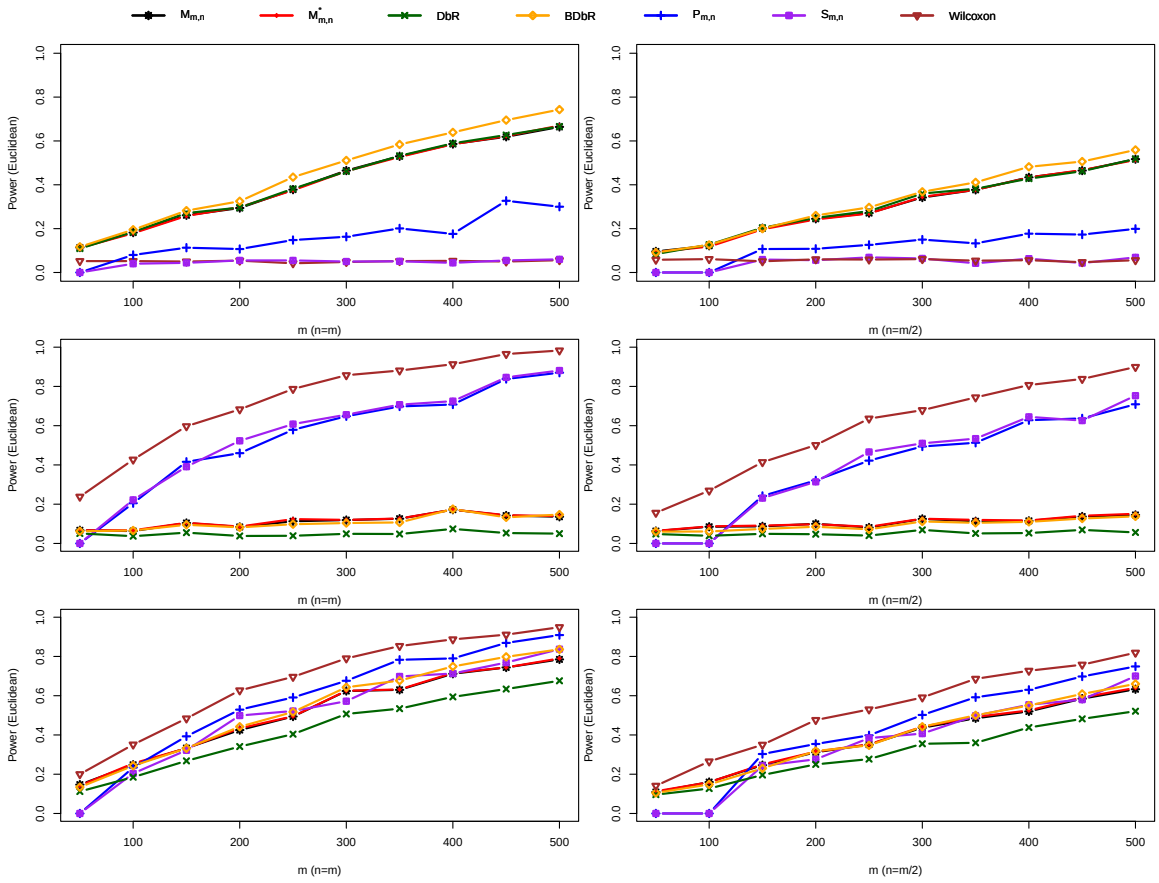


Figure 3.1: Power Comparison of seven test statistics under one dimension for $m = 50, \ldots, 500$ with $n = m$ (1st column) or $n = m/2$ (2nd column) for change in scale (1st row), change in mean (2nd row), and change in both mean and scale (3rd row).

For change in scale, we consider distributions $F = \mathcal{N}(0.25, 1)$ and $G = \mathcal{N}(0.25, 1.15)$. The results in first row indicate that the product statistic $P_{m,n}$ is more sensitive to variations in variance compared to the sum statistic $S_{m,n}$, which shows comparable sensitivity to the Wilcoxon statistic. The maximum statistic $M_{m,n}$, minimum statistic $M^*_{m,n}$, DbR, and BDbR exhibit similar performance levels but outperform $P_{m,n}$.

For change in mean, with $F = \mathcal{N}(0, 1)$ and $G = \mathcal{N}(0.25, 1)$. The Wilcoxon statistic is the best. The $P_{m,n}$ and $S_{m,n}$ statistics show comparable effectiveness and outperform the other considered statistics.

Upon introducing both mean and scale changes, by setting $F = \mathcal{N}(0.25, 1)$ and $G = \mathcal{N}(0, 1.15)$, all tested statistics demonstrate improved performance over the first two scenarios. The Wilcoxon, $P_{m,n}$ and $S_{m,n}$ statistics notably exhibit the highest levels of effectiveness.

In summary, within a one-dimensional Euclidean depth framework, the $P_{m,n}$ can capture variations in either mean or scale. The $S_{m,n}$ and Wilcoxon can only capture mean variations, whereas the $M_{m,n}$, $M^*_{m,n}$, DbR and BDbR statistics are more suited for identifying scale variations.

3.2 Multivariate distribution

In this subsection, we consider power comparisons within the context of multivariate data, focusing on multivariate distributions and employing various depth functions. Under the null hypothesis, we assume $F = G = N(\mathbf{0}, I_{2\times 2})$, where $N(\mathbf{0}, I_{2\times 2})$ denotes the bivariate normal distribution with a mean vector $\mathbf{0}$ and a two-by-two identity covariance matrix $I_{2\times 2}$. Our objective is to assess the power of different test statistics under three distinct scenarios: changes in scale, mean, and both scale and mean.

Similarly, we set the significance level at $\alpha = 0.05$. Critical values for $M_{m,n}$, $M^*_{m,n}$, DbR and BDbR are upper 95% quantiles based on the null hypothesis ($F = G$) of 1000 replications. The power is calculated by the proportion of times in 1000 repetitions that the statistic is greater than the upper 95% quantile. For the permutations test based on $P_{m,n}$ and $S_{m,n}$ in the strategic block permutation algorithm, we set the threshold for the p-value to be the lower 5% quantile of the simulated 1000 p-values under the null hypotheses of number of repetitions $\mathcal{C} = 200$ and block size $s = 25$. The power of the permutation test is then calculated using the proportion of times in 1000 repetitions that the statistic is less than the lower 5% quantile.

(1) Change in scale

For the case of scale change, we consider the distributions $F = N(\mathbf{0}, I_{2\times 2})$ and $G = N(\mathbf{0}, I_{2\times 2} + 0.5\tilde{I}_{2\times 2})$, where $\tilde{I}_{2\times 2} = ((0, 1)^\top, (1, 0)^\top)$. Power comparisons, depicted in Figure 3.2, are based on Mahalanobis depth, spatial depth and projection depth, for each scenario.

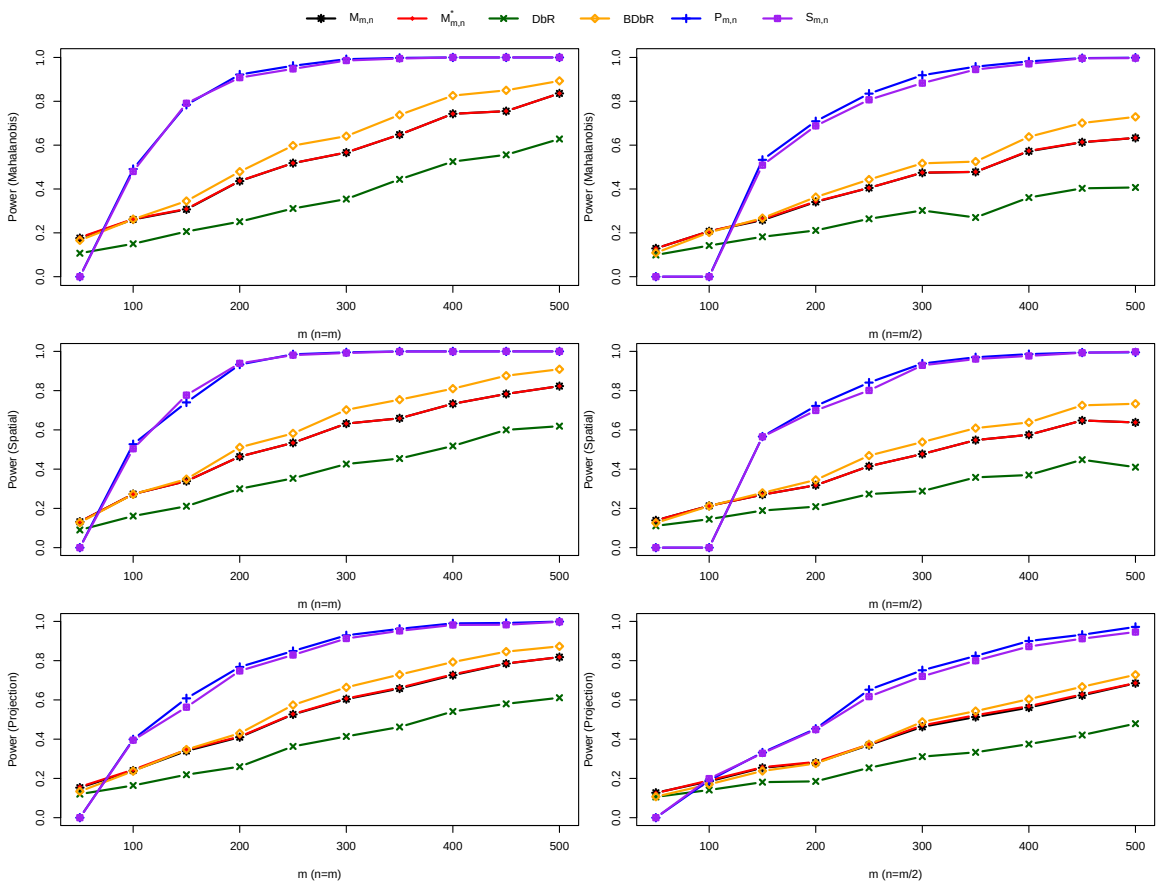


Figure 3.2: Power comparison under alternative hypothesis $F = N(\mathbf{0}, I_{2\times 2})$ against $G = N(\mathbf{0}, I_{2\times 2} + 0.5\tilde{I}_{2\times 2})$ for m=50, 100, ..., 500 and $n = m$ (1st column) or $n = m/2$ (2nd column) for Mahalanobis depth (Row 1), Spatial depth (Row 2), and Projection depth (Row 3).

It is evident from the Figure 3.2 that all three depth functions exhibit similar trends in the power of the various test statistics under examination. Notably, $P_{m,n}$ and $S_{m,n}$ are comparable and outperform all other tested statistics across all depth functions, attributed to their efficacy in detecting scale changes. Additionally, $M_{m,n}$, $M^*_{m,n}$, and BDbR test statistics show similar levels of performance.

(2) Change in mean

For change in mean, we consider $F = N(\mathbf{0}, I_{2\times 2})$ and $G = N((0.3, 0.3)^\top, I_{2\times 2})$. Figure 3.3 demonstrates the power comparison, where the $P_{m,n}$ and $S_{m,n}$ not only demonstrate comparable performance to each other but also consistently outperform the other tested statistics. In this case, other statistics show relatively low power, and the Maximum Statistic, Minimum Statistic, and BDbR follow a similar trend and visually fall on the same line in terms of power.

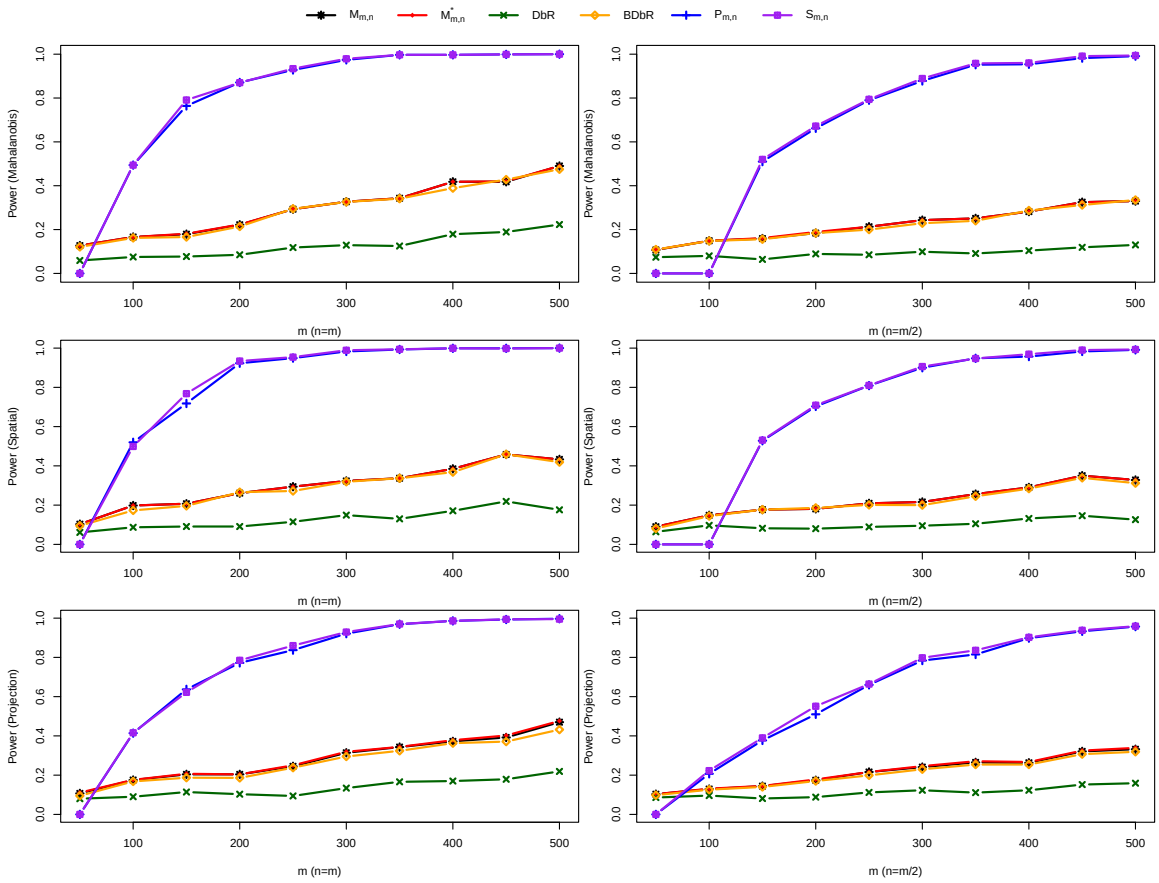


Figure 3.3: Power comparison under alternative hypothesis $F = N(\mathbf{0}, I_{2\times 2})$ against $G = N((0.3, 0.3)^\top, I_{2\times 2})$ for m=50, 100, …, 500 and $n = m$ (1st column) or $n = m/2$ (2nd column) for Mahalanobis depth (Row 1), Spatial depth (Row 2), and Projection depth (Row 3).

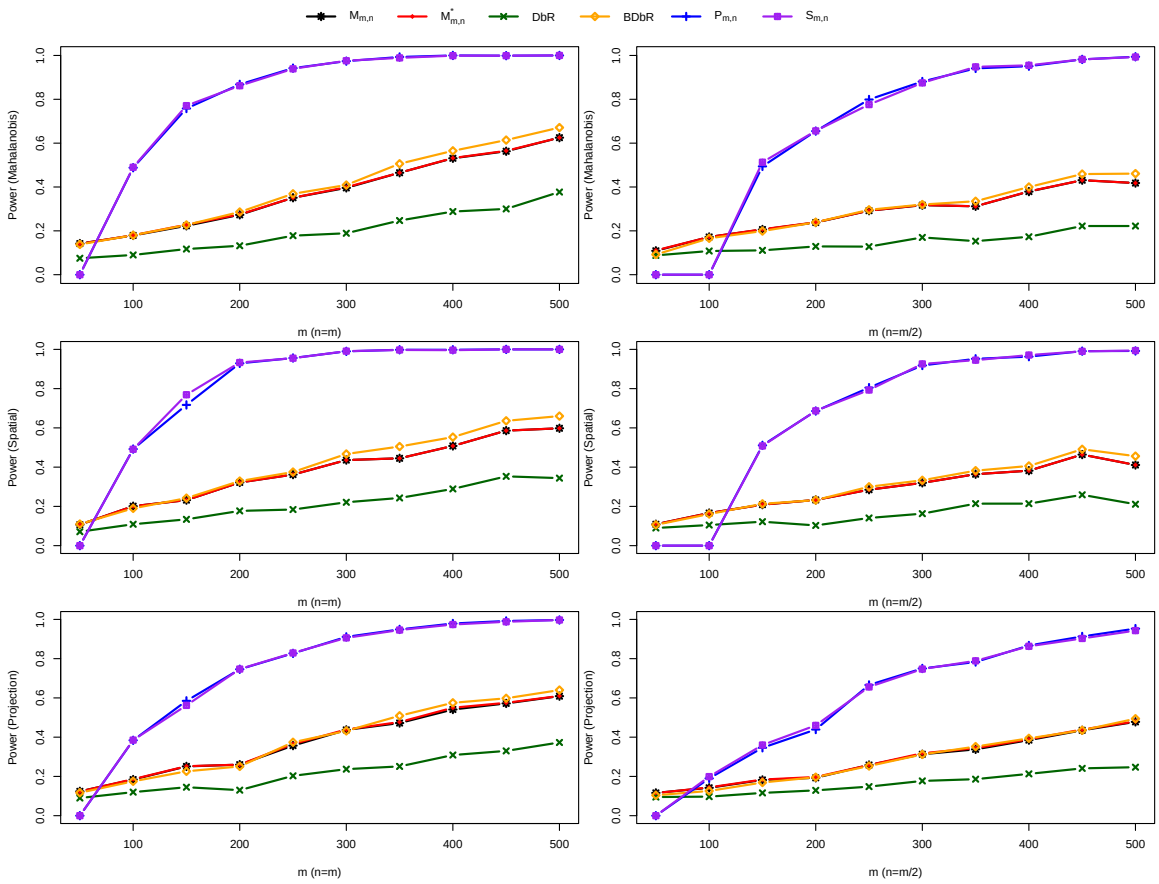


Figure 3.4: Power comparison under alternative hypothesis $F = N(\mathbf{0}, I_{2\times 2})$ against $G = N((0.2, 0.2)^\top, I_{2\times 2} + 0.4\tilde{I}_{2\times 2})$ for $m = 50, 100, \ldots, 500$ and $n = m$ (1st column) or $n = m/2$ (2nd column) for Mahalanobis depth (Row 1), Spatial depth (Row 2), and Projection depth (Row 3).

(3) Change in both mean and scale

We investigated the scenario where both scale and mean change occur. We consider the distributions $F = N(\mathbf{0}, I_{2\times 2})$, and the other sample was generated from $G = N((0.2, 0.2)^\top, I_{2\times 2} + 0.4\tilde{I}_{2\times 2})$. The power comparisons, as shown in Figure 3.4, further validate the superior performance of the $P_{m,n}$ and $S_{m,n}$ statistics, while the Minimum Statistic shows comparable performance to the BDbR.

In conclusion, both $P_{m,n}$ and $S_{m,n}$ demonstrate comparable and notably high efficacy across various multivariate depth functions. These statistics have proven to be promising tools, particularly in scenarios involving changes in mean, scale, or both, within multivariate distributions. Their consistent performance across different scenarios highlights their potential as versatile and robust choices for statistical testing in multivariate analyses.

Chapter 4

Real Data Analysis

Our proposed test statistics $P_{m,n}$ and $S_{m,n}$ can be applied in real data sets. A significant contribution is the detection of different types of peaks in spectra data. The same method is also applied to two data sets: Sloan digital sky survey data and skull data to show its performance, presenting both estimated p-values and asymptotic p-values, and verified through scale curves [30].

4.1 Raman Spectrum

Prostate cancer is one of the most common diseases in Canada and the second most common in the world. Its potential to be fatal underscores the critical importances of early diagnosis [5]. Additionally, a significant challenge in optimizing treatment protocols is the lack of consideration for individual patient radiosensitivity when prescribing radiation doses. Consequently, there is a pressing need to develop methods for monitoring radiation response in individuals undergoing radiation therapy. Various techniques have been explored for this purpose [24, 15]. In recent years, Raman spectroscopy (RS) has been investigated as a potential augmentative tool for biochemical analysis of tumour response [15, 2, 12]. RS provides detailed 'fingerprint' biochemical information on various biomolecules (e.g., protein, lipid, DNA, etc.) through a vibrational inelastic light scattering process [36]. Recent studies have indicated that RS can offer predictive capabilities regarding tumour proliferation status [35, 18]. Moreover, when RS is combined with group and basis-restricted non-negative matrix factorization along with random forest strategies, this enhanced technique can yield valuable ranked information about the biochemical dynamics within irradiated tumours [34]. Raman spectroscopy (RS) works by shining a monochromatic laser beam onto a sample and directing the incident light to collide with molecules, resulting in a change in frequency between the scattered and incident photons. Raman spectroscopy plots this change and measures the frequency shift in units of the reciprocal of the wavelength (cm^{-1}). The change in frequency caused by the vibration of chemical bonds produces peaks in the spectrum that are specific to different molecules [36]. This characteristic of peaks with different

frequency shifts can be used to localize specific biological tissues. An important finding was that the spectral peak appearing at 1523.71 cm^{-1} was associated with the presence of carotenoids, which are absent from normal tissues but present in the spectra of neuromas [33]. This has inspired further experimental studies of Raman spectroscopy. Recently, this Raman spectroscopy technique was applied to patient samples collected at the Kelowna Cancer Center in British Columbia.

Despite the potential of Raman Spectroscopy (RS) in cancer diagnosis, several systematic issues in data processing need to be addressed. These include data interference and subjective determination errors [6]. Challenges such as baseline variability between sample acquisitions are prevalent. Notably, approximately 10% of Raman spectra suffer interference from cosmic rays, leading to spikes and potential false peaks in the spectra. Furthermore, the analysis of most Raman spectra relies on manual evaluation, resulting in subjective determination errors due to the lack of a uniform and efficient automated method. Our goal is to identify statistically whether there is a significant abnormal peak at 1523.71 cm^{-1}, which may become a new diagnostic criterion for spectroscopically assisted diagnosis and prognosis of prostate cancer.

With a range of spectra data of patient samples, the classification of different shapes in a range of spectra is crucial in the context of prostate cancer. Our dataset comprises 48 spectra, each containing 1019 wavenumbers, totaling 48912 wavenumbers. Spectra 17 and 18 were not included in the analysis because of the predominance of zero observations due to detector saturation, so data from 46 spectra were considered. The dataset includes a column of wavenumber values, ranging from 147 cm^{-1} to 1870 cm^{-1}, alongside a corresponding column of intensity values measured in counts. Our primary focus is on the spectral shapes that exhibit a peak at 1523.71 cm^{-1}, a wavenumber indicative of carotenoids presence – a crucial biomarker in the diagnosis and prognosis of prostate cancer [36].

For the analysis, we concentrated on the wavenumber range from 1503.539 cm^{-1} to 1543.807 cm^{-1}, centered at 1523.71 cm^{-1}. This range encompasses 27 wavenumber values, referred to as the data dimension d. Our approach involved a two-stage classification process. In the first stage, we employed a linear regression model to fit the quadratic values to the spectral intensities, using the R-squared value as a measure of fit. A spectrum with a peak should resemble a quadratic curve, as illustrated in Figure 4.1. Spectra were then categorized into two groups based on an R-squared threshold of 0.5. Specifically, the quadratic function used was $(x - x_0)^2/27^2$, where x ranges from 1 to 27 and x_0 is the central point at 14. The R-squared value determines the fit of this quadratic model to the index values of x, with a threshold of 0.5 employed to distinguish between two possible peak types. Spectra with R-squared values below 0.5 were classified into Group 1, while those with higher values were categorized into Group 2. In total, 35 spectra were assigned to Group 1 and 11 spectra to Group 2.

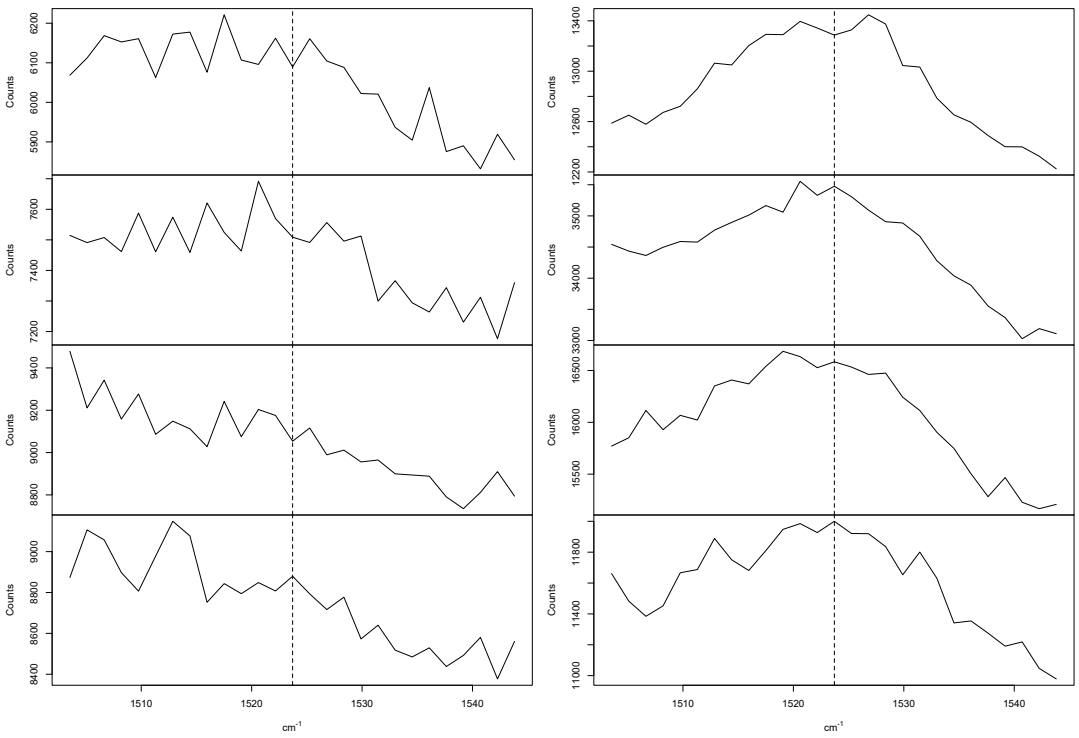


Figure 4.1: Initial screening plots of some of the spectra in Group 1 (left) and Group 2 (right) with a possible peak at 1523.71 cm^{-1}.

In the second stage of our analysis, aimed at more accurately distinguishing between the two spectral shapes, we conducted a two-sample test considering various dimensions. Initially, based on the classifications from the first stage, we focused on the central 27 points around 1523.71 cm^{-1}, denoted as $27M$. To assess the impact of smaller dimensions on test power, we also considered 15 points in the middle ($15M$), 5 points to the left of the center ($5L$), and 5 points to the right of the center ($5R$).

For each dimension, we computed the p-values for $S_{m,n}$ and $P_{m,n}$ with a block size $s = 2$ and repetition number $\mathcal{C} = 1000$, and compared these with $M_{m,n}$, $M^*_{m,n}$, DbR, and BDbR as shown in the Table 4.1. It is important to note that in larger dimensions such $27M$ and $15M$, Mahalanobis and spatial depths were not applicable due to the non-invertibility of the sample covariance matrix, and thus are omitted from these comparisons.

Additionally, we investigated the effects of logarithmic transformation on the depth measures. To distinguish between the original and log-transformed depths, we added an "O" suffix for the original depths and an "L" suffix for log-transformed depths in our notation. For example, "MO" signifies Mahalanobis depth applied to the original counts, while "ML" refers to Mahalanobis depth on log-transformed counts. Using significance level $\alpha = 0.1$ and comparing the resulting p-values, we observed that $P_{m,n}$, $S_{m,n}$, and BDbR consistently ranked among the top three, with projected depths showing greater dominance in higher dimensions. Conversely, Mahalanobis depths were more influential in lower dimensions.

Table 4.1: p-values of $P_{m,n}$, $S_{m,n}$, $M_{m,n}$, $M^*_{m,n}$, DbR, BDbR under different depth, d, and data transformation of counts.

d	27M		15M		5L						5R					
Depth	PO	PL	PO	PL	MO	ML	SO	SL	PO	PL	MO	ML	SO	SL	PO	PL
$P_{m,n}$	0.006	0	0.004	0.010	0.079	0.087	0.086	**0.129**	0.054	**0.293**	0.051	0.051	0.037	0.057	0.045	**0.221**
$S_{m,n}$	0.006	0	0.004	0.006	0.018	0.086	0.023	**0.109**	0.028	**0.240**	0.051	**0.246**	0.070	**0.345**	0.027	**0.224**
$M_{m,n}$	0.008	0.045	0.021	0.031	**0.569**	**0.145**	**0.558**	**0.149**	**0.298**	**0.767**	**0.199**	0.029	**0.101**	0.034	**0.417**	0.071
$M^*_{m,n}$	0.008	0.045	0.02	0.031	**0.569**	**0.145**	**0.558**	**0.149**	**0.294**	**0.759**	**0.199**	0.029	**0.101**	0.034	**0.411**	0.07
DbR	0.006	0.004	0.036	**0.185**	**0.384**	**0.148**	**0.325**	**0.133**	**0.263**	**0.68**	**0.244**	0.044	**0.145**	0.067	**0.256**	0.096
BDbR	0.004	0.013	0.027	0.04	**0.613**	**0.149**	**0.595**	**0.137**	**0.23**	**0.541**	**0.163**	0.03	0.077	0.038	**0.145**	0.015

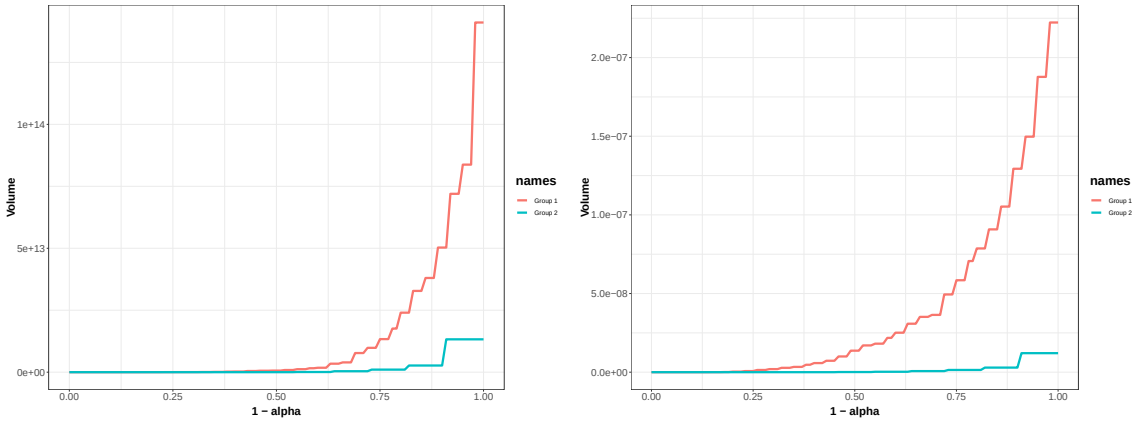


Figure 4.2: Scale curves for 5L derived from Mahalanobis depth to the original intensity values (left) and log-transformed intensity values (right).

We also noted that logarithmic transformations tended to slightly improve consistency in inference.

In addition to the methods previously discussed, we employed the concept of a scale curve, introduced by [30], to compare the dispersion or scale of two samples. The scale curve quantifies the volume of the α-trimmed region of distribution F, denoted as $D_\alpha(F)$, which is defined as

$$D_\alpha(F) = \left\{ x \in \mathbb{R}^d : D(x; F) \geq \alpha \right\}.$$

Consequently, we plotted the volume of this convex region $V(\alpha; F_m)$ against the $1-\alpha$ scale. Figure 4.2 displays the scale curves derived from the Mahalanobis depth, illustrating both the raw (left) and log-transformed (right) intensity values for the $5L$ spectral dimension. This analysis further validates the differences between the two samples.

Additionally, adjusting the R-squared threshold in the initial spectral classification step results in varying compositions of spectra in each group, as detailed in Table 4.2. It is noteworthy that the spectra in Group 2 exhibit more complexity compared to those in Group 1, as depicted in Figure 4.1. For instance, setting the R-squared threshold at 0.4 yields small p-values for all test statistics (as seen in Table 4.3), suggesting a significant

R^2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0.3	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	1	1	1	1	1	2
0.4	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	1	1	1	1	1	2
0.5	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
0.6	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
0.7	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1

R^2	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
0.3	2	2	2	1	1	1	1	2	2	2	1	2	2	2	1	2	2	1	2	2	2	2	2
0.4	2	2	2	1	1	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	2	2
0.5	2	2	2	1	1	1	1	2	2	1	1	1	1	1	1	2	2	1	1	1	2	2	2
0.6	1	2	2	1	1	1	1	2	1	1	1	1	1	1	1	2	2	1	1	1	2	1	2
0.7	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2

Table 4.2: Classification of 46 spectra into 2 groups with different R-squared threshold values (1: Group 1; and 2: Group 2)

Table 4.3: p-values of $P_{m,n}$, $S_{m,n}$, $M_{m,n}$, $M^*_{m,n}$, DbR, BDbR under different depth, d, and data transformation of counts with R-squared threshold 0.4.

d	27M		15M		5L						5R					
Depth	PO	PL	PO	PL	MO	ML	SO	SL	PO	PL	MO	ML	SO	SL	PO	PL
$P_{m,n}$	0	0	0	0.001	0.01	0.001	0.009	0.007	0.001	0.004	0.008	0.002	0.01	0	0.003	0.006
$S_{m,n}$	0	0	0	0.001	0.012	0	0.005	0.005	0	0.002	0.003	0	0.004	0.002	0.004	0.003
$M_{m,n}$	0	0	0.001	0	0.052	0.022	0.064	0.037	0	0	**0.106**	0	**0.19**	0	0.005	0.001
$M^*_{m,n}$	0	0	0.001	0	0.052	0.022	0.064	0.037	0	0	**0.106**	0	**0.19**	0	0.005	0.001
DbR	0.003	0.001	0.002	0.001	0.084	0.004	0.082	0.002	0.002	0	0.098	0	0.089	0	0.012	0.002
BDbR	0	0	0	0.001	0.053	0.025	0.066	0.034	0	0.001	**0.116**	0	**0.199**	0.001	0.009	0.006

difference between the two groups. This significance is attributed to the reclassification of some spectra from Group 1 into Group 2, leading to a smaller yet still discernible difference. Conversely, when the threshold is increased to 0.6, most p-values become larger (refer to Table 4.4), indicating that the differences between the groups are not statistically significant. This shift results from some spectra originally in Group 2 being categorized into Group 1, further narrowing the differences.

The comprehensive analysis conducted on various spectral shapes firmly establishes the efficacy of our proposed statistical method. This method not only effectively differentiates

Table 4.4: p-values of $P_{m,n}$, $S_{m,n}$, $M_{m,n}$, $M^*_{m,n}$, DbR, BDbR under different depth, d, and data transformation of counts with R-squared threshold 0.6.

d	27M		15M		5L						5R					
Depth	PO	PL	PO	PL	MO	ML	SO	SL	PO	PL	MO	ML	SO	SL	PO	PL
$P_{m,n}$	0.009	0.001	0.017	**0.106**	**0.533**	**0.286**	**0.469**	**0.218**	0.060	**0.525**	0.050	**0.304**	0.048	**0.372**	**0.292**	**0.193**
$S_{m,n}$	0.005	0	0.003	**0.124**	**0.311**	**0.171**	**0.246**	**0.136**	0.035	**0.421**	**0.211**	**0.732**	**0.322**	**0.814**	**0.351**	**0.273**
$M_{m,n}$	0.055	**0.11**	**0.167**	0.054	**0.783**	**0.462**	**0.842**	**0.456**	**0.754**	**0.397**	0.081	**0.217**	**0.104**	**0.29**	**0.429**	0.009
$M^*_{m,n}$	0.055	**0.11**	**0.167**	0.054	**0.783**	**0.462**	**0.842**	**0.456**	**0.754**	**0.397**	0.081	**0.217**	**0.104**	**0.29**	**0.427**	0.009
DbR	0.014	0.004	0.094	**0.332**	**0.809**	**0.456**	**0.859**	**0.433**	**0.292**	0.071	**0.137**	**0.25**	**0.208**	**0.352**	**0.606**	0.014
BDbR	0.013	0.002	0.013	0.061	**0.77**	**0.477**	**0.847**	**0.459**	**0.28**	**0.69**	0.082	**0.207**	**0.104**	**0.277**	**0.351**	0.008

between distinct types of tumor samples but also demonstrates superior performance compared to other competing methodologies. The ability to discern subtle variations in spectral data is crucial in the context of tumor sample classification, and our approach has proven to be a robust tool in this regard.

Through the use of scale curves, depth functions, and strategic permutation testing, our method offers a reasonable and precise means of detecting and categorizing spectral differences. This is particularly vital in the diagnosis and prognosis of conditions like prostate cancer, where accurate identification of biomarkers such as carotenoids is essential. The success of this method in outperforming other statistical techniques underscores its potential as a valuable asset in medical spectral analysis and related fields.

4.2 Sloan Digital Sky Survey Data

The Sloan Digital Sky Survey (SDSS) data dataset, available in the `astrodatR` package in *R*, consisting of three classes of point source with measurements on four color indices (u-g, g-r, r-i, i-z). The three classes are categorized as quasars (Class 1), main sequence and giant stars (Class 2), and giant stars (Class 3), with sample sizes 2000, 5000, and 2000 respectively. We conducted two-sample tests and three-sample tests on this data set to investigate any correlation between the distribution of four color indices among these classes.

To compare the dispersion or scale of multiple distributions, we used scale curves. For two-sample tests, we consider the pairwise comparisons between samples, i.e., Class 1 vs. Class 2, Class 1 vs. Class 3, and Class 2 vs. Class 3. We plotted the scale curve in a logarithmic scale in order to enhance the visualization of dispersion, the scale curves of three cases under Mahalanobis depth are shown in Figure 4.3. As observed from the figure, there is a large dispersion between Class 1 and Class 2, and Class 2 and Class 3, while Class 1 and Class 3 have some overlap in the scale curve. The non-overlapping curves represent the potential differences between the two classes, while some overlap may indicate some similarity between the two samples.

To quantify the result from scale curves, we calculated estimated p-values and asymptotic p-values; see equations (4.1) and (4.2).

The asymptotic p-value for Maximum Statistic M_{n_1,n_2}, with sample size n_1, n_2 respectively, in two-sample cases can be written in this form [41]:

$$P(M_{n_1,n_2} \leq x) \to P\{(c_{1,2}Z_1 + \tilde{c}_{1,2}Z_2)^2 \leq x\}, \tag{4.1}$$

where Z_1, Z_2 are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$ with $c_{i,j}^2 + \tilde{c}_{i,j}^2 = 1$.

Similarly, the asymptotic p-value for Minimum Statistic $M^*_{n_1,n_2}$, with sample size n_1, n_2 respectively, in two-sample cases can be written in this form, based on the proof of its asymptotic distribution:

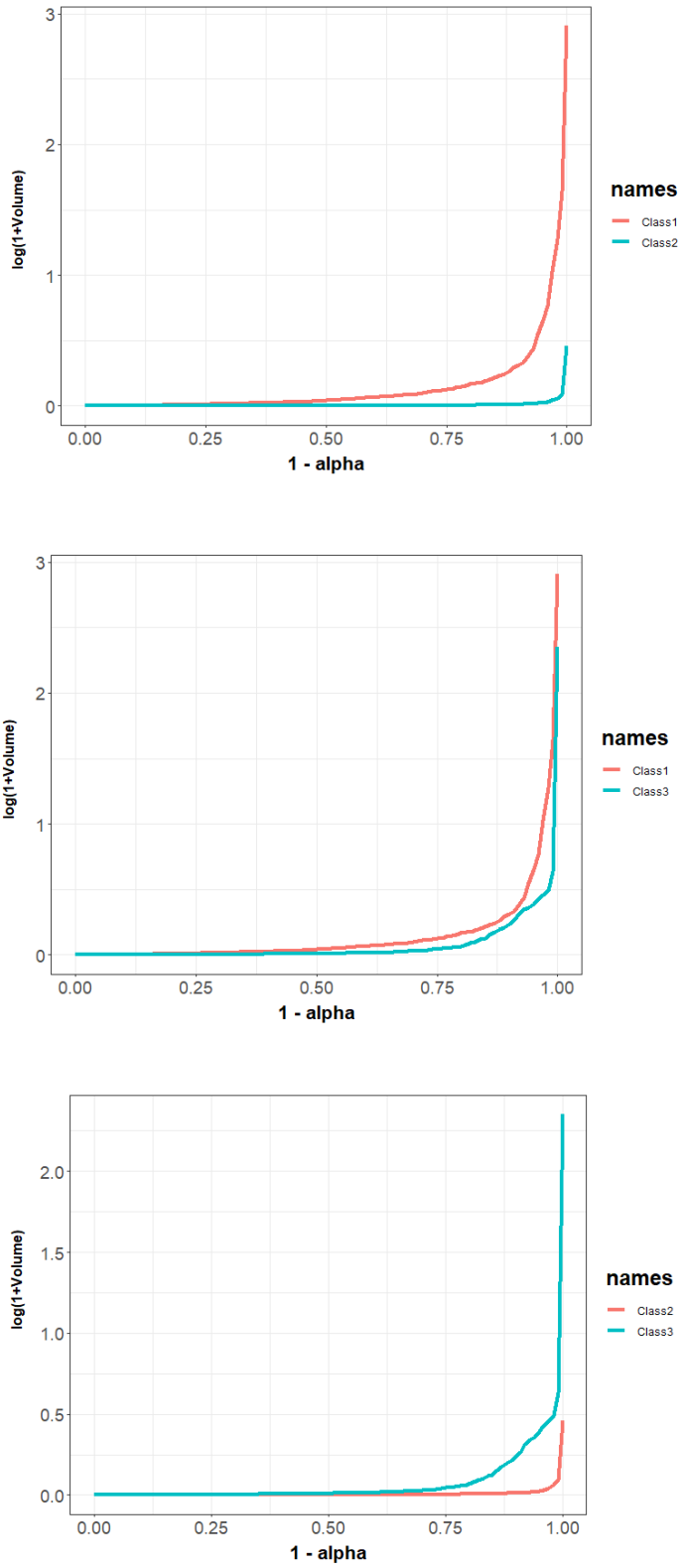


Figure 4.3: Scale curves for three classes under Mahalanobis depth in log scale in Sloan Digital Sky Survey data: Class 1 vs. Class 2 (first row), Class 1 vs. Class 3 (second row), and Class 2 vs. Class 3 (third row)

$$P(M^*_{n_1,n_2} \leq x) \to P\{-x \leq (c_{1,2}Z_1 + \tilde{c}_{1,2}Z_2) \leq x\}, \tag{4.2}$$

where Z_1, Z_2 are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$ with $c_{i,j}^2 + \tilde{c}_{i,j}^2 = 1$.

1. Class 1 vs. Class 2

   To obtain the estimated p-values for $M_{m,n}$, $M^*_{m,n}$, DbR, and BDbR, we simulated the data set with 1000 repetitions for each depth function with sample sizes: $m = 2000$, $n = 5000$. The p-values for $P_{m,n}$ and $S_{m,n}$ are calculated through block permutation with block size $s = 100$ and repetition number $C = 200$. We compare all these p-values for different test statistics. In addition, we also calculated the asymptotic p-value for $M_{m,n}$ and $M^*_{m,n}$ to further make comparisons.

   The results show that all estimated p-values with different test statistics are zero for all depth functions, which is the same as the asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$. These findings suggest a significant relationship between the combined four color indices between Class 1 (quasars) and Class 2 (main sequence and giant stars) in the SDSS data.

2. Class 1 vs. Class 3

   Since the scale curves show a possible similarity between Class 1 and Class 3, we further calculated estimated p-values and asymptotic p-values to verify the result.

   Similarly, we simulated the data set with 1000 repetitions for each depth function with sample sizes ($m = 2000$, $n = 2000$) to obtain estimated p-values for $M_{m,n}$, $M^*_{m,n}$, DbR, and BDbR. The p-values for $P_{m,n}$ and $S_{m,n}$ are calculated through block permutation with block size $s = 100$ and repetition number $C = 200$.

   The estimated p-values and asymptotic p-values are all zero, which is the same as the result when comparing Class 1 and Class 2. This represents that there is a significant difference between Class 1 and Class 3, although the scale curve shows some similarity.

3. Class 2 vs. Class 3

   Lastly, we make comparisons between Class 2 vs. Class 3. Under the same method and settings, we computed estimated p-values and asymptotic p-values, which are all zeros for all depth functions. This result indicates significant differences in color indices between Class 2 and Class 3, which validates the dispersion of curves of two classes in scale curve in Figure 4.3.

All two-sample tests revealed a strong correlation between the four color indices and the three classes of point sources. We could further conduct three-sample tests on all three

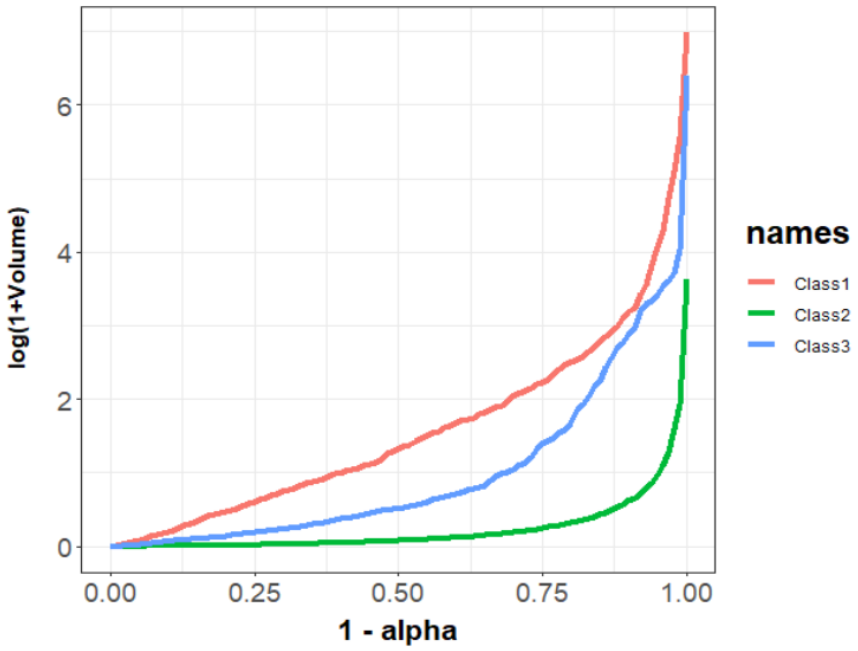


Figure 4.4: Scale curves for three classes under Mahalanobis depth in log scale

classes and find p-values for $M_{m,n}$, $M^*_{m,n}$, and DbR tests. The detailed three-sample test statistics for $M_{m,n}$ and $M^*_{m,n}$ and their asymptotic p-values are presented in Chapter 5.1.

Similarly, the scale curve for three samples in Figure 4.4 presented non-overlapping curves, indicating that these three classes are significantly different.

Furthermore, the estimated p-values are all zeros for $M_{m,n}$, $M^*_{m,n}$, and DbR for all Mahalanobis depth, spatial depth, and projection depth; same for asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for all depth functions. These values align with the results from scale curve in Figure 4.4.

In summary, the analysis of the SDSS data using both our proposed test statistics and existing methods revealed a strong correlation between the four color indices and the three classes of point sources for all pairwise comparison and three-sample comparison.

4.3 Skull Data

In addition, we can apply our proposed test statistic to Egyptian skull data to examine changes in skull size over time. The Egyptian skulls were obtained from the R package `HSAUR`, consisting of four measurements of skulls (maximum breaths, basibregmatic heights, basialiveolar length, and nasal heights of the skull) ranging from five epochs (4000 B.C., 3300 B.C., 1850 B.C., 200 B.C., and 150 A.D.). Each epochs contain 30 samples. We could

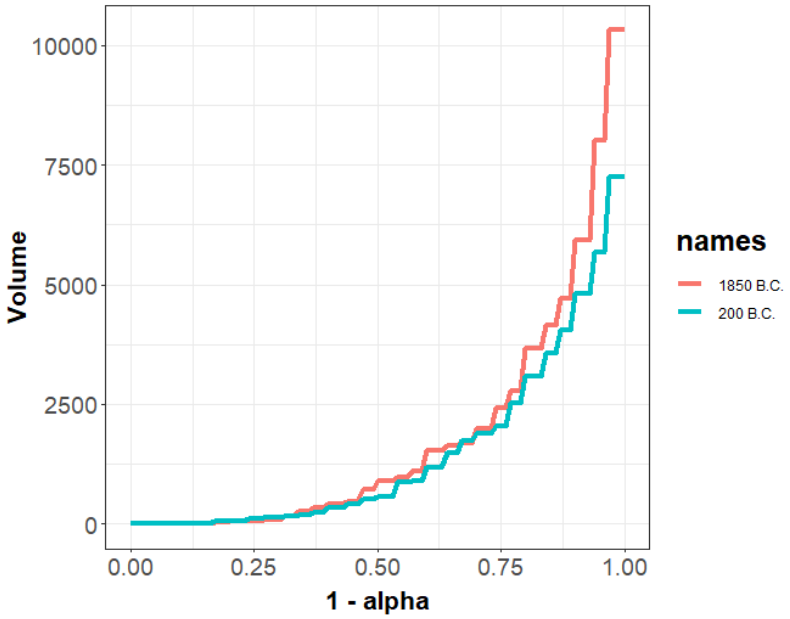


Figure 4.5: Scale curves of skull data for epochs: 1850 B.C. and 200 B.C. under Mahalanobis depth

perform two-sample and three-sample tests to investigate whether skull size changes over time during interbreeding with immigrants.

The analysis of this skull data set contains five comparisons of epochs. For two-sample tests, we mainly focus on three pairwise comparisons between (1) 1850 B.C. vs. 200 B.C., (2) 3300 B.C. vs. 150 A.D., (3) 200 B.C. vs. 150 A.D. For three-sample tests, we could make comparisons on (4) 1850 B.C., 200 B.C., and 150 A.D., and (5) 3300 B.C., 200 B.C., and 150 A.D.

1. 1850 B.C. and 200 B.C.

   First, we made the scale curve to visualize the differences in skull sizes between 1850 B.C. and 200 B.C., shown in Figure 4.5. As observed from the scale curve, these two curves illustrate a small difference between 1850 B.C. and 200 B.C., suggesting no significant changes in skull sizes during these two periods.

   Therefore, we present the estimated p-values for $M_{m,n}$, $M^*_{m,n}$, $P_{m,n}$, $S_{m,n}$, DbR, and BDdR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ in Table 4.5. For $M_{m,n}$, $M^*_{m,n}$, DbR, and BDdR, the estimated p-values are conducted through 1000 iterations; and the estimated p-values for $P_{m,n}$ and $S_{m,n}$ are computed through block permutation algorithm with block size $s = 5$ and repetition number $\mathcal{C} = 1000$.

   The results in the table show that all the estimated p-values are greater than the significance level of 0.05, indicating that there is no strong correlation between skull sizes

	$M_{m,n}$	$M^*_{m,n}$	$P_{m,n}$	$S_{m,n}$	DbR	BDbR	Asy $M_{m,n}$	Asy $M^*_{m,n}$
Mahalanobis	0.312	0.312	0.177	0.127	0.351	0.379	0.051	0.051
Spatial	0.396	0.396	0.249	0.180	0.411	0.447	0.038	0.038
Projection	0.369	0.365	0.163	0.137	0.145	0.212	0.124	0.124

Table 4.5: Estimated p-values for $M_{m,n}$, $M^*_{m,n}$, $P_{m,n}$, $S_{m,n}$, DbR, and BDbR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for 1850 B.C. vs. 200 B.C. under Mahalanobis depth, Spatial depth, and Projection depth (Asy: asymptotic)

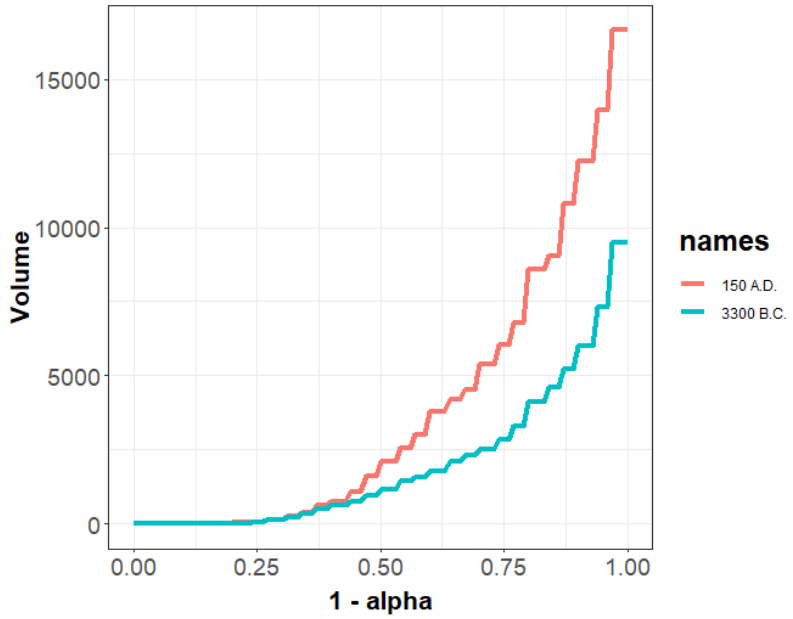


Figure 4.6: Scale curves of skull data for epochs: 3300 B.C. and 150 A.D. under Mahalanobis depth

and interbreeding with immigrations for these two epochs. Moreover, the asymptotic p-values under Mahalanobis depth, Spatial depth, and Projection depth are smaller than all estimated p-values; this may be due to the effect of a small sample size of 30 for each epoch.

2. 3300 B.C. and 150 A.D.

   We focus on these two epochs, 3300 B.C. and 150 A.D., as these two epochs are far apart in time. To examine if there are differences in skull size distributions between the two time epochs across 3300 B.C. and 150 A.D., the scale curve is shown in Figure 4.6. The two curves are noticeably separated, suggesting a significant difference in skull sizes as time changes.

   To further support this observation, we calculated the estimated p-values for $M_{m,n}$, $M^*_{m,n}$, $P_{m,n}$, $S_{m,n}$, DbR, and BDdR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$,

	$M_{m,n}$	$M^*_{m,n}$	$P_{m,n}$	$S_{m,n}$	DbR	BDbR	Asy $M_{m,n}$	Asy $M^*_{m,n}$
Mahalanobis	0.014	0.014	0.007	0.014	0.017	0.015	0.000	0.000
Spatial	0.014	0.014	0.010	0.018	0.011	0.013	0.000	0.000
Projection	0.006	0.006	0.024	0.048	0.013	0.010	0.001	0.001

Table 4.6: Estimated p-values for $M_{m,n}$, $M^*_{m,n}$, $P_{m,n}$, $S_{m,n}$, DbR, and BDbR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for 3300 B.C. vs. 150 A.D. under Mahalanobis depth, Spatial depth, and Projection depth (Asy: asymptotic)

	$M_{m,n}$	$M^*_{m,n}$	$P_{m,n}$	$S_{m,n}$	DbR	BDbR	Asy $M_{m,n}$	Asy $M^*_{m,n}$
Mahalanobis	0.108	0.108	0.424	0.694	0.094	0.109	0.010	0.010
Spatial	0.128	0.128	0.362	0.600	0.136	0.143	0.005	0.005
Projection	0.068	0.065	0.200	0.293	0.097	0.128	0.013	0.013

Table 4.7: Estimated p-values for $M_{m,n}$, $M^*_{m,n}$, $P_{m,n}$, $S_{m,n}$, DbR, and BDbR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for 200 B.C. vs. 150 A.D. under Mahalanobis depth, Spatial depth, and Projection depth (Asy: asymptotic)

summarized them in Table 4.6. Similarly, we performed simulations for 1000 repetitions to calculate estimated p-values for $M_{m,n}$, $M^*_{m,n}$, DbR, and BDdR; and using block permutation algorithm with block size $s = 5$ and repetition number $\mathcal{C} = 1000$ for $P_{m,n}$ and $S_{m,n}$.

Notably, all these estimated p-values are smaller than the significance level $\alpha = 0.05$ and are close to zero. This leads us to the conclusion that there is a significant difference in skulls between these two epochs. The small asymptotic p-values for all depth functions also align with the earlier conclusion, supporting the presence of substantial differences in skull sizes across 3300 B.C. and 150 A.D.

3. 200 B.C. and 150 A.D.

   In this part, we compare the difference in skull size between 200 B.C. and 150 A.D., which are relatively close in time span. Figure 4.7 shows the scale curves revealing the difference between these two epochs. These two curves are still relatively far apart, while their dispersion is observed to be larger than the case of 3300 B.C. vs. 150 A.D.

   Therefore, we need to calculate the estimated p-values and asymptotic p-values to verify the result in the scale curve. The estimated p-values are calculated with similar settings as before. As shown in Table 4.7, all estimated p-values are larger than significance level $\alpha = 0.05$, indicating no strong correlation between skull sizes and interbreeding with immigrations for these two epochs. Notice that the estimated p-values for $P_{m,n}$ and $S_{m,n}$ are larger than all other test statistics, revealing that $P_{m,n}$ and $S_{m,n}$ have more potential to capture the difference between samples. Moreover, the asymptotic p-values are small due to small sample sizes, which is not very informative in our data analysis on skull data.

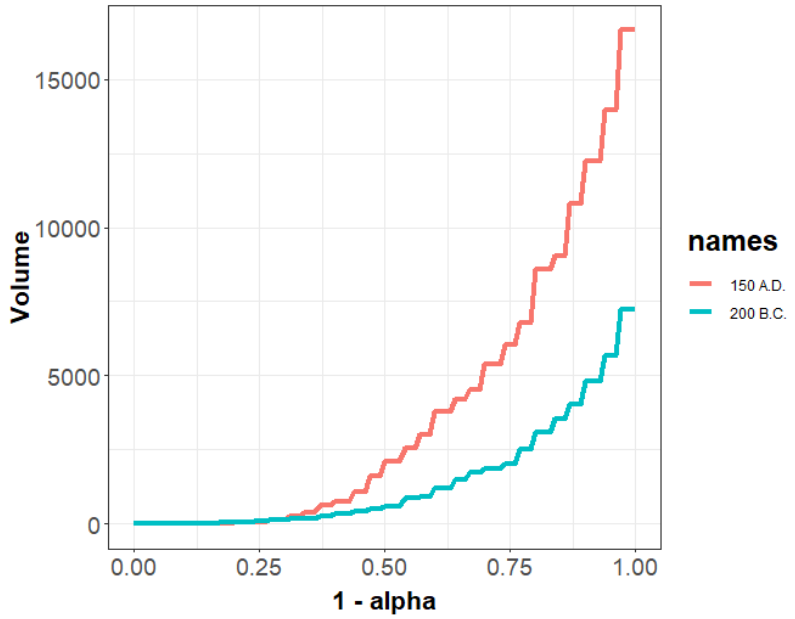


Figure 4.7: Scale curves of skull data for epochs: 200 B.C. and 150 A.D. under Mahalanobis depth

	$M_{m,n}$	$M^*_{m,n}$	DbR	Asy $M_{m,n}$	Asy $M^*_{m,n}$
Mahalanobis	0.2612	0.2612	0.1422	0.0269	0.0269
Spatial	0.2740	0.2740	0.1428	0.0152	0.0152
Projection	0.2468	0.2446	0.1136	0.0314	0.0314

Table 4.8: Estimated p-values for $M_{m,n}$, $M^*_{m,n}$, and DbR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for 1850 B.C., 200 B.C. and 150 A.D. under Mahalanobis depth, Spatial depth, and Projection depth (Asy: asymptotic)

4. 1850 B.C., 200 B.C., and 150 A.D.

   For three-sample tests with epochs: 1850 B.C., 200 B.C., and 150 A.D., the scale curves are shown in Figure 4.8, illustrating a small difference between the epochs 1850 B.C. and 200 B.C., and large difference between 1850 B.C. and other two epochs.

   Therefore, we present the estimated p-values for $M_{m,n}$, $M^*_{m,n}$, and DbR in Table 4.8, based on 5000 iterations; and asymptotic p-values for $M_{m,n}$, $M^*_{m,n}$.

   All the estimated p-values are greater than the significance level of 0.05, indicating that there is no strong correlation between skull sizes and interbreeding with immigrations for these three epochs. However, the asymptotic p-values are all very small because of a small sample size.

5. 3300 B.C., 200 B.C., and 150 A.D.

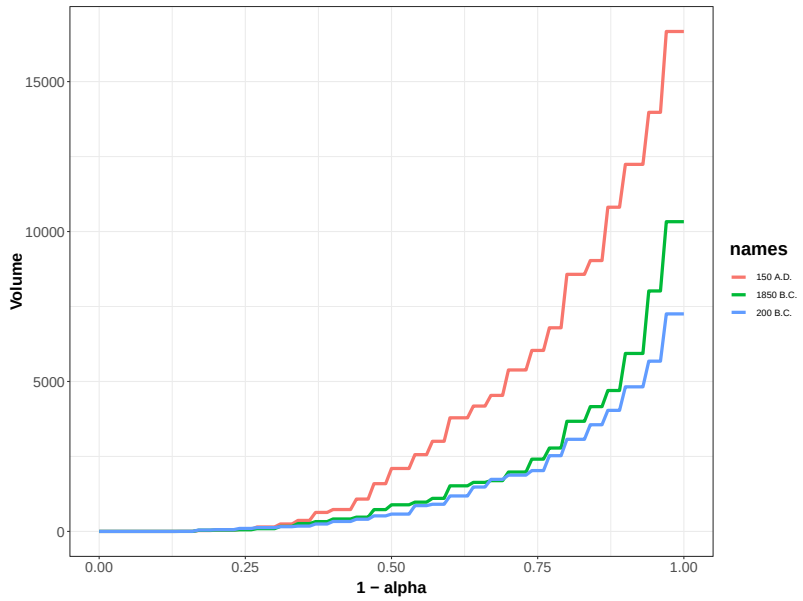


Figure 4.8: Scale curves of skull data for epochs: 1850 B.C., 200 B.C., and 150 A.D. under Mahalanobis depth

	$M_{m,n}$	$M^*_{m,n}$	DbR	Asy $M_{m,n}$	Asy $M^*_{m,n}$
Mahalanobis	0.0378	0.0378	0.0120	0.0012	0.0012
Spatial	0.0328	0.0328	0.0094	0.0005	0.0005
Projection	0.0264	0.0260	0.0100	0.0043	0.0043

Table 4.9: Estimated p-values for $M_{m,n}$, $M^*_{m,n}$, and DbR, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$ for 3300 B.C., 200 B.C. and 150 A.D. under Mahalanobis depth, Spatial depth, and Projection depth (Asy: asymptotic)

In a similar way, the scale curves for 3300 B.C., 200 B.C., and 150 A.D. are shown in Figure 4.9, the three curves are noticeably separated, suggesting a significant difference in skull sizes as time changes. To further support this observation, we calculated the estimated p-values for $M_{m,n}$, $M^*_{m,n}$, and DbR based on 5000 iterations, and asymptotic p-values for $M_{m,n}$ and $M^*_{m,n}$, summarized them in Table 4.9.

Notably, all these p-values are smaller than the significance level of 0.05 and are close to zero. This leads us to the conclusion that there is a significant difference in skulls between these three epochs. The computed asymptotic p-values align with the earlier conclusion, supporting the presence of substantial differences in skull sizes across the specified epochs.

By using scale curves and computing estimated p-values to compare the change of skull sizes on different epochs in skull data, we could make summaries on whether there exist significant changes in skull sizes statistically. We can visually illustrate the differences be-

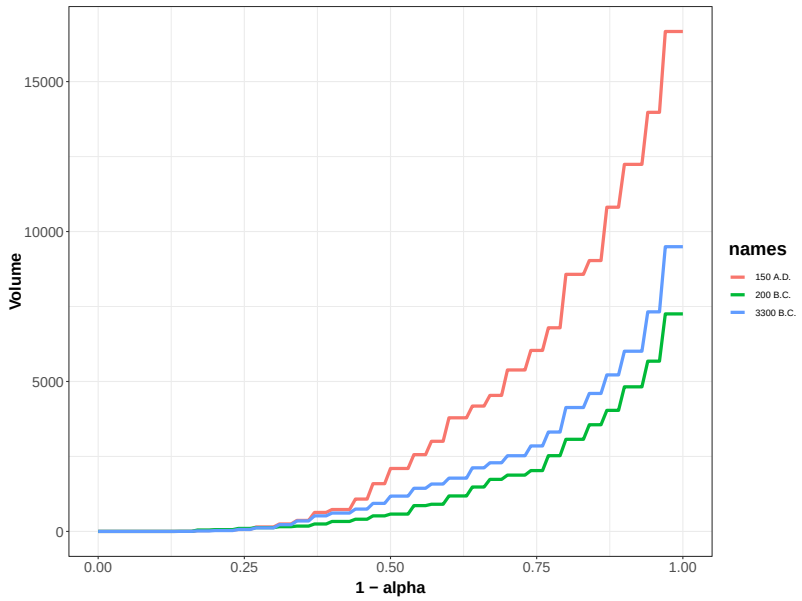


Figure 4.9: Scale curves of skull data for epochs: 3300 B.C., 200 B.C., and 150 A.D. under Mahalanobis depth

tween groups with scale curves and compute estimated p-values with different test statistics to further support the findings. Moreover, this method can be applied to more data to solve practical problems.

Chapter 5

Discussion and Conclusions

5.1 Multi-sample Test

The two-sample test statistics can be extended to perform homogeneity tests on multi-samples. This section will specifically discuss extended maximum statistic $M_{m,n}$ and minimum statistic $M^*_{m,n}$ in multi-sample tests. Furthermore, similar technique could apply on sum statistic $S_{m,n}$ and product statistic $P_{m,n}$.

Consider there are k samples, each drawn from distribution $F^{(j)}$ with sample size n_j, with $j = 1, 2, \ldots, k$. The corresponding empirical distribution is $F^{(j)}_{n_i}$.

Maximum statistic

The generalized maximum statistic for k sample [41] is:

$$M_{n_1,\ldots,n_k} = \max_{1\le i,j\le k, i\neq j} \left[\frac{1}{12}(\frac{1}{n_i}+\frac{1}{n_j})\right]^{-1} \left[Q(F^{(i)}_{n_i}, F^{(j)}_{n_j}) - \frac{1}{2}\right]^2. \tag{5.1}$$

Based on the asymptotic null distribution of maximum statistic [41], we can write

$$P(M_{n_1,\ldots,n_k} \le x) \to P\left\{\max_{1\le i<j\le k} (c_{i,j}Z_i + \tilde{c}_{i,j}Z_j)^2 \le x\right\}, \tag{5.2}$$

where $Z_1, Z_2, \ldots, Z_k$ are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1}+n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1}+n_j^{-1})^{-1/2}$ with $c^2_{i,j} + \tilde{c}^2_{i,j} = 1$.

A simple example is three-sample maximum statistic,

$$\begin{aligned} M_{n_1,n_2,n_3} = \max\{&[\frac{1}{12}(\frac{1}{n_1}+\frac{1}{n_2})]^{-1}[Q(F^{(1)}_{n_1}, F^{(2)}_{n_2}) - \frac{1}{2}]^2, [(\frac{1}{12}(\frac{1}{n_1}+\frac{1}{n_3})]^{-1}[Q(F^{(1)}_{n_1}, F^{(3)}_{n_3}) - \frac{1}{2}]^2, \\ &[(\frac{1}{12}(\frac{1}{n_2}+\frac{1}{n_1})]^{-1}[Q(F^{(2)}_{n_2}, F^{(1)}_{n_1}) - \frac{1}{2}]^2, [(\frac{1}{12}(\frac{1}{n_2}+\frac{1}{n_3})]^{-1}[Q(F^{(2)}_{n_2}, F^{(3)}_{n_3}) - \frac{1}{2}]^2, \\ &[(\frac{1}{12}(\frac{1}{n_3}+\frac{1}{n_2})]^{-1}[Q(F^{(3)}_{n_3}, F^{(2)}_{n_2}) - \frac{1}{2}]^2, [(\frac{1}{12}(\frac{1}{n_3}+\frac{1}{n_1})]^{-1}[Q(F^{(3)}_{n_3}, F^{(1)}_{n_1}) - \frac{1}{2}]^2\}. \end{aligned}$$

and

$$P(M_{n_1,n_2,n_3} \le x) \to P\{(c_{1,2}Z_1 + \tilde{c}_{1,2}Z_2)^2 \le x, (c_{1,3}Z_1 + \tilde{c}_{1,3}Z_3)^2 \le x, \tag{5.3}$$

$$(c_{2,3}Z_2 + \tilde{c}_{2,3}Z_3)^2 \le x\}, \tag{5.4}$$

where Z_1, Z_2, Z_3 are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$ with $c_{i,j}^2 + \tilde{c}_{i,j}^2 = 1$.

Minimum statistic

Similarly, the minimum statistic for k samples can be written in the following form:

$$M^*_{n_1,\ldots,n_k} = \max_{1\le i,j\le k, i\ne j} \left[\frac{1}{12}(\frac{1}{n_i} + \frac{1}{n_j})\right]^{-\frac{1}{2}} (\frac{1}{2} - Q(F^{(i)}_{n_i}, F^{(j)}_{n_j})) \tag{5.5}$$

Based on the proof of asymptotic distribution of minimum statistic, we can write:

$$P(M^*_{n_1,\ldots,n_k} \le x) \to P\left\{\max_{1\le i<j\le k}(c_{i,j}Z_i + \tilde{c}_{i,j}Z_j) \le x\right\}, \tag{5.6}$$

where $Z_1, Z_2, \ldots, Z_k$ are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$ with $c_{i,j}^2 + \tilde{c}_{i,j}^2 = 1$.

For example, when $k = 3$, the Minimum Statistic can be expanded as

$$\begin{aligned} M^*_{n_1,n_2,n_3} = \max\{&[\frac{1}{12}(\frac{1}{n_1} + \frac{1}{n_2})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(1)}_{n_1}, F^{(2)}_{n_2})], [(\frac{1}{12}(\frac{1}{n_1} + \frac{1}{n_3})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(1)}_{n_1}, F^{(3)}_{n_3})],\\ &[(\frac{1}{12}(\frac{1}{n_2} + \frac{1}{n_1})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(2)}_{n_2}, F^{(1)}_{n_1})], [(\frac{1}{12}(\frac{1}{n_2} + \frac{1}{n_3})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(2)}_{n_2}, F^{(3)}_{n_3})],\\ &[(\frac{1}{12}(\frac{1}{n_3} + \frac{1}{n_2})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(3)}_{n_3}, F^{(2)}_{n_2})], [(\frac{1}{12}(\frac{1}{n_3} + \frac{1}{n_1})]^{-\frac{1}{2}}[\frac{1}{2} - Q(F^{(3)}_{n_3}, F^{(1)}_{n_1})]\}. \end{aligned}$$

and

$$P(M^*_{n_1,n_2,n_3} \le x) \to P\{-x \le (c_{1,2}Z_1 + \tilde{c}_{1,2}Z_2) \le x, -x \le (c_{1,3}Z_1 + \tilde{c}_{1,3}Z_3) \le x, \tag{5.7}$$

$$-x \le (c_{2,3}Z_2 + \tilde{c}_{2,3}Z_3) \le x\}, \tag{5.8}$$

where Z_1, Z_2, Z_3 are independent from $N(0,1)$, $c_{i,j} = \lim n_i^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$, and $\tilde{c}_{i,j} = \lim n_j^{-1/2}(n_i^{-1} + n_j^{-1})^{-1/2}$ with $c_{i,j}^2 + \tilde{c}_{i,j}^2 = 1$.

More properties of multi-sample minimum statistic are left to further research.

In addition, we can also find the generalized product and sum statistics and study their asymptotic properties. For product and sum statistics, denote as $P_{n_1,n_2,\ldots,n_k}$ and $S_{n_1,n_2,\ldots,n_k}$ for k samples, their k-sample comparisons can be written as

$$P_{n_1,\ldots,n_k} = \prod_{1\le i_1,\ldots,i_{k-1}\le k, i_{\ell_1}\neq i_{\ell_2}, 1\le \ell_1,\ell_2\le k-1} P_{n_{i_1},\ldots,n_{i_{k-1}}} \tag{5.9}$$

and

$$S_{n_1,\ldots,n_k} = \sum_{1\le i_1,\ldots,i_{k-1}\le k, i_{\ell_1}\neq i_{\ell_2}, 1\le \ell_1,\ell_2\le k-1} S_{n_{i_1},\ldots,n_{i_{k-1}}}, \tag{5.10}$$

where $P_{n_{i_1},\ldots,n_{i_{k-1}}}$ and $S_{n_{i_1},\ldots,n_{i_{k-1}}}$ are product and sum statistics for $k-1$ samples.

For three-sample cases $k = 3$, the product and sum statistics are

$$P_{n_1,n_2,n_3} = P_{n_1,n_2} P_{n_1,n_3} P_{n_2,n_3}$$

and

$$S_{n_1,n_2,n_3} = S_{n_1,n_2} + S_{n_1,n_3} + S_{n_2,n_3}$$

5.2 High-dimensional Data

For high-dimensional data, with a large number of dimension p and small sample size n, homogeneity tests on these type of data are challenging. As seen from data analysis on Ramen spectrum, with a large dimension such as 27M and 15M, the inverse of covariance matrix is not available, leading to invalid calculation on Mahalanobis depth and Spatial depth. This section will focus on how to solve this issue.

The original Mahalanobis depth [31] for any point x is defined as:

$$D(x; F) = \frac{1}{1 + (x-\mu)^T \Sigma^{-1} (x-\mu)},$$

where μ represents the mean of distribution F, and Σ is the covariance matrix.

A modified version of Mahalanobis depth is proposed by [9] to improve the estimation of covariance matrix through oracle approximating shrinkage estimator, which wrote the modified covariance matrix as

$$\hat{\Sigma} = \rho \hat{F} + (1-\rho)\Sigma,$$

where ρ is a control parameter, with range $\rho \in (0, 1)$; $\hat{F}$ is target matrix with $\hat{F} = Tr(\Sigma/p) I_{p\times p}$, where $I_{p\times p}$ is p-dimensional identity matrix [49].

Based on the code by [9] and [49], ρ is calculated through the following steps:

(1) First, calculate the two values A and B, with

$$A = (1 - \frac{2}{p}) Tr(\Sigma^2) + Tr(\Sigma)^2,$$

$$B = (n + 1 - \frac{2}{p})(Tr(\Sigma^2) - \frac{Tr(\Sigma)^2}{p}),$$

(2) ρ is based on the value of A and B,

$$\rho = \max(\min(\frac{A}{B}, 1), 0).$$

By this modification, the extended Mahalanobis depth can have more robust depth values. This modified Mahalanobis depth can be applied to Q Statistics and have the corresponding Product and Sum Statistics under Mahalanobis depth. Through this modification, the proposed Product and Sum Statistics can be applied on high-dimensional data.

For power comparison, we can also use modified band depth (MBD) statistics [32], designed for high-dimensional data. Let $f_1, \ldots, f_n$ be a set of continuous function on interval I, the MBD on any f is defined as

$$\text{MBD}(f) = \binom{n}{2}^{-1} \frac{1}{\lambda(I)} \sum_{1 \le i_1 < i_2 \le n} \lambda(A(f; f_{i_1}, f_{i_2})),$$

where $A(f; f_{i_1}, f_{i_2}) = \{t \in I : \min_{r=i_1,i_2} f_r(t) \le f(t) \le \max_{r=i_1,i_2} f_r(t)\}$ and λ is the Lebesgue measure in $\mathbb{R}$.

For a d-dimensional point $\boldsymbol{y}$ in sample $\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}$, the finite-dimensional MBD is defined as:

$$\text{MBD}_d(\boldsymbol{y}) = \binom{n}{2}^{-1} \sum_{1 \le i_1 < i_2 \le n} d^{-1} \times \sum_{k=1}^{d} I_{\{\min\{y_{i_1}(k), y_{i_2}(k)\} \le y(k) \le \max\{y_{i_1}(k), y_{i_2}(k)\}\}}.$$

5.3 Conclusion

This thesis aims to propose a novel DEEPEAST technique to retain the power when combining two or more Q Statistics and introduce two new test statistics $P_{m,n}$ and $S_{m,n}$ to test the homogeneity of two samples. To explain how to enhance statistical power without losing information when projection multivariate data to one-dimensional data, we define the same-attraction function. Our new test statistics $P_{m,n}$ and $S_{m,n}$ share a 'common attractor' and are applicable across all depth functions. In addition, we propose a proposition to compare the asymptotic power of test statistics. By introducing several examples, our proposed test statistic are asymptotically more powerful than some existing statistics. The asymptotic distributions of $P_{m,n}$ and $S_{m,n}$ under one-dimensional Euclidean depth were derived and substantiated through extensive simulations, involving comparisons of density profiles and tail probabilities. Our derivation utilizes Hoeffding decomposition of V-Statistics, our technique could be further extended to multidimensional case for all depths. The strategic block permutation algorithm was developed to facilitate the comprehensive application of the DEEPEAST technique across various depth functions. Our extensive simulated power

comparisons reveal that $P_{m,n}$ and $S_{m,n}$ exhibit superior performance in multivariate distributions and are competitive in one-dimensional Euclidean depth.

In addition, we apply the DEEPEAST technique for comparing different spectral samples and conclude that our proposed $P_{m,n}$ and $S_{m,n}$ have superior performance in discerning the difference between two samples of varying dimensions, outperforming other test statistics in this regard. Moreover, we conducted more real data analyses on sloan digital sky survey data and skull data through detailed two-sample and three-sample comparisons in Chapter 4. By finding the estimated p-values and asymptotic p-values, we can test if two or more groups of data are statistically the same through our proposed test statistics and existing methods.

Although our research provides significant contributions to two-sample tests, there is an opportunity to extend this methodology to multi-sample tests. In line with approaches similar to [41], a generalized version of $P_{m,n}$ and $S_{m,n}$ could be developed. More simulations could be performed to shows its statistical power. This extension represents an intriguing and challenging avenue for future theoretical research, promising to further enhance the use and applicability of our findings in broader statistical contexts. More applications and extensions of data depth on high-dimensional data is also a future work.

Bibliography

[1] T. Anderson. 1962. On the Distribution of the Two-Sample Cramér-von Mises Criterion. *The Annals of Mathematical Statistics* 33, 3 (1962), 1148–1159.

[2] G. W. Auner, S. K. Koya, H. Huang, B. Broadbent, M. Trexler, Z. Auner, A. Elias, K. C. Mehne, and M. A. Brusatori. 2018. Applications of Raman spectroscopy in cancer diagnosis. *Cancer and Metastasis Reviews* 37, 4 (2018), 691–717.

[3] M. S. Barale and D. T. Shirke. 2021. A test based on data depth for testing location-scale of the two multivariate populations. *Journal of Statistical Computation and Simulation* 91, 4 (2021), 768–785.

[4] W. Baumgartner, P. Weiß, and H. Schindler. 1998. A Nonparametric Test for the General Two-Sample Problem. *Biometrics* 54, 3 (1998), 1129–1135.

[5] C. V. Berenguer, Pereira. F., J. S. Câmara, and J. A. M. Pereira. 2023. Underlying Features of Prostate Cancer—Statistics, Risk Factors, and Emerging Methods for Its Diagnosis. *Current Oncology* (2023), 2300–2321.

[6] J. M. Brewer. 2023. *Automated processing of prostate-based Raman spectra.* Master's thesis. University of British Columbia.

[7] B. M. Brown. 1958. Statistical use of spatial median. *Journal of the Royal Statistical Society: Series B* 53 (1958), 448–456.

[8] Y. Chen, W. Lin, and X. Shi. 2023. Multivariate two-sample test statistics based on data depth. (2023).

[9] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero. 2010. Shrinkage algorithms for MMSE covariance estimation. *Trans. Sig. Proc.* 58, 10 (oct 2010), 5016–5029.

[10] S. Chenouri and C. G. Small. 2012. A nonparametric multivariate multisample test based on data depth. *Electronic Journal of Statistics* 6 (2012), 760 – 782.

[11] E. Chung and J. Romano. 2016. Asymptotically valid and exact permutation tests based on two-sample U-statistics. *Journal of Statistical Planning and Inference* 168 (2016), 97–105.

[12] S. Corsetti, T. Rabl, D. McGloin, and G. Nabi. 2018. Raman spectroscopy for accurately characterizing biomolecular changes in androgen-independent prostate cancer cells. *Journal of Biophotonics* 11, 3 (2018).

[13] D. Cox and N. Reid. 1987. Approximations to Noncentral Distributions. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 15, 2 (1987), 105–14.

[14] C. Craig. 1936. On the Frequency of the function xy. *The Annals of Mathematical Statistics* 7, 1 (1936), 1–15.

[15] S. Devpura, K.N. Barton, S. L. Brown, O. Palyvoda, S. Kalkanis, V. N. Naik, F. Siddiqui, R. Naik, and I. J. Chetty. 2014. Vision 20/20: The role of Raman spectroscopy in early stage cancer detection and feasibility for application in radiation therapy response assessment: Raman spectroscopy for cancer detection/radiation therapy response assessment. *Medical Physics* 41, 5 (2014), 050901.

[16] T. S. Ferguson. 2005. U-STATISTICS: Notes for Statistics 200C, Spring 2005. (2005).

[17] R. A. Fisher. 1936. *Design of experiments*. Vol. 1. British Medical Journal. 554 pages.

[18] A. M. Fuentes, A. Narayan, K. Milligan, J. J. Lum, A. G. Brolo, J. L. Andrews, and A. Jirasek. 2023. Raman spectroscopy and convolutional neural networks for monitoring biochemical radiation response in breast tumour xenografts. *Scientific Reports* 13, 1 (2023), 1530.

[19] F. Gnettner, C. Kirch, and A. Nieto-Reyes. 2023. Variations of the depth based Liu-Singh two-sample test including functional spaces. (2023).

[20] J. C. Gower. 1974. Algorithm as 78: The mediancentre. *Journal of the Royal Statistical Society: Series C* 23 (1974), 466–470.

[21] W. Hoeffding. 1948. A Non-Parametric Test of Independence. *The Annals of Mathematical Statistics* 19, 4 (1948), 546 – 557.

[22] W. Hoeffding. 1948. A Class of Statistics with Asymptotically Normal Distribution. *The Annals of Mathematical Statistics* 19, 3 (1948), 293–325.

[23] H. Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936), 321.

[24] S. B. Jiang, J. Wolfgang, and G. S. Mageras. 2008. Quality Assurance Challenges for Motion-Adaptive Radiation Therapy: Gating, Breath Holding, and Four-Dimensional Computed Tomography. *International Journal of Radiation Oncology*Biology*Physics* 71, 1 (2008), 103–107.

[25] I. Kim, S. Balakrishnan, and L. Wasserman. 2020. Robust multivariate nonparametric tests via projection averaging. *The Annals of Statistics* 48, 6 (2020), 3417 – 3441.

[26] D. Kosiorowski and Z. Zawadzki. 2017. DepthProc: An R Package for Robust Exploration of Multidimensional Economic Phenomena. (2017).

[27] A. J. Lee. 1990. *U-Statistics: Theory and Practice. Statistics: Textbooks and Monographs.* Dekker, Inc., New York.

[28] J. Liu, S. Ma, W. Xu, and L. Zhu. 2022. A generalized Wilcoxon–Mann–Whitney type test for multivariate data through pairwise distance. *Journal of Multivariate Analysis* 190 (2022), 104946.

[29] R. Y. Liu. 1992. Data depth and multivariate rank tests. *In L_1-Statistics and Related Methods (Y. Dodge, ed.)* (1992), 279–294.

[30] R. Y. Liu, M. P. Jesse, and S. Kesar. 1999. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics* (1999), 783–858.

[31] R. Y. Liu and K. Singh. 1993. A quality index based on data depth and multivariate rank tests. *J. Amer. Statist. Assoc.* 88, 421 (1993), 252–260.

[32] S. López-Pintado, J. Romo, and A. Torrente. 2010. Robust depth-based tools for the analysis of gene expression data. *Biostatistics* 11 2 (2010), 254–64.

[33] A. Mahadevan-Jansen and RR. Richards-Kortum. 1996. Raman spectroscopy for the detection of cancers and precancers. *Journal of Biomedical Optics* 1, 1 (1996), 31–70.

[34] K. Milligan, X. Deng, P. Shreeves, R. Ali-Adeeb, Q. Matthews, A. Brolo, J. J. Lum, J. L. Andrews, and A. Jirasek. 2021. Raman spectroscopy and group and basis-restricted non negative matrix factorisation identifies radiation induced metabolic changes in human cancer cells. *Scientific Reports* 11, 1 (2021), 3853.

[35] K. Milligan, S. J. Van Nest, X. Deng, R. Ali-Adeeb, P. Shreeves, S. Punch, N. Costie, N. Pavey, J. M. Crook, D. M. Berman, A. G. Brolo, J. J. Lum, J. L. Andrews, and A. Jirasek. 2022. Raman spectroscopy and supervised learning as a potential tool to identify high-dose-rate-brachytherapy induced biochemical profiles of prostate cancer. *Journal of Biophotonics* 15, 11 (2022).

[36] Z. Movasaghi, S. Rehman, and I. U. Rehman. 2007. Raman Spectroscopy of Biological Tissues. *Applied Spectroscopy Reviews* 42, 5 (2007), 493–541.

[37] H. Murakami. 2012. Modified Baumgartner statistics for the two-sample and multisample problems: a numerical comparison. *Journal of Statistical Computation and Simulation* 82, 5 (2012), 711–728.

[38] K. C. S. Pillai. 1955. Some New Test Criteria in Multivariate Analysis. *Annals of Mathematical Statistics* 26 (1955), 117–121.

[39] SAS. 2009. *SAS/STAT® 9.2 User's Guide Second Edition*. SAS Publishing. 101 pages.

[40] R. Serfling and Y. Zuo. 2000. General notions of statistical depth function. *The Annals of Statistics* 28, 2 (2000), 461–482.

[41] X. Shi, Y. Zhang, and Y. Fu. 2023. Two-sample tests based on data depth. *Entropy* 25, 2 (2023), 238.

[42] G. J. Székely and M. L. Rizzo. 2004. Testing for equal distributions in high dimension. (2004).

[43] G. J. Székely and M. L. Rizzo. 2013. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference* 143, 8 (2013), 1249–1272.

[44] J. W. Tukey. 1974. Mathematics and the picturing of data. *Canadian Mathematical Congress* 2 (1974), 523–531.

[45] R. von Mises. 1947. On the asymptotic distribution of differentiable statistical functions. *The Annals of Mathematical Statistics* 18, 2 (1947), 309–348.

[46] W. Welch. 1990. Construction of permutation tests. *J. Amer. Statist. Assoc.* 85, 411 (1990), 693–698.

[47] F. Wilcoxon. 1992. Individual comparisons by ranking methods. *Springer New York* (1992), 196–202.

[48] S. S. Wilks. 1935. On the independence of k sets of normally distributed statistical variables. *Econometrica* 3, 3 (1935), 309–326.

[49] Y. Zhang. 2024. *Ranking functions of data.* Master's thesis. University of British Columbia.

[50] Y. Zuo and X. He. 2006. On the limiting distributions of multivariate depth-based rank sum statistics and related tests. *The Annals of Statistics* 24, 6 (2006), 2879–2896.

Appendix A

Code

A.1 Chapter 2

A.1.1 Density Plot of Sum Statistic

```
mu=0
sd=1
library(MASS)

Sum_stat=function(a,b){
  x=rnorm(a,mu,sd)
  y=rnorm(b,mu,sd)

  D_xf=1/(1+(x-mean(x))^2)
  D_yf=1/(1+(y-mean(x))^2)
  D_xg=1/(1+(x-mean(y))^2)
  D_yg=1/(1+(y-mean(y))^2)

  s1=c()
  for (i in 1:a){
    s1[i]=sum(D_xf[i]<=D_yf)
  }
  #sum(s1)

  s2=c()
  for (i in 1:b){
    s2[i]=sum(D_yg[i]<=D_xg)
  }
  #sum(s2)

  S=-a*b*(sum(s1)/a/b + sum(s2)/a/b -1)/(a+b)

  return(S)
}

set.seed(1)
loop=10000
table=matrix(NA,10,loop)

```

```
#m,n same size
m=n=c(1:10)*100
for (i in 1:length(m)){
  for (j in 1:loop){
    table[i,j]=Sum_stat(m[i],n[i])
  }
}

library(calculus)
library(mvtnorm)
x=seq(-2, 4, 0.01)
x=x[!(x==0)]
f1=function(z){
  exp(-(z^2+sqrt(3)*pi*x^2/2/z^2)/(2-sqrt(3)/pi))/z
}
y=integral(f1,bounds=list(z=c(0,Inf)), method='hcubature')$value *exp(sqrt
    (3)*x/(2-sqrt(3)/pi))/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))
plot(x,y, type='l',ylab='',main='Sum Statistics density plot',col='blue')
lines(density(table[10,]),col='red')
```

A.1.2 Density Plot of Product Statistic

```
Prod_stat=function(a,b){
  x=rnorm(a,mu,sd)
  y=rnorm(b,mu,sd)

  D_xf=1/(1+(x-mean(x))^2)
  D_yf=1/(1+(y-mean(x))^2)
  D_xg=1/(1+(x-mean(y))^2)
  D_yg=1/(1+(y-mean(y))^2)

  s1=c()
  for (i in 1:a){
    s1[i]=sum(D_xf[i]<=D_yf)
  }
  #sum(s1)

  s2=c()
  for (i in 1:b){
    s2[i]=sum(D_yg[i]<=D_xg)
  }
  #sum(s2)

  P=-a*b*( (sum(s1)/a/b) * (sum(s2)/a/b) -1/4)/(a+b)

  return(P)
}

set.seed(1)
loop=10000
table=matrix(NA,10,loop)

#m,n same size
m=n=c(1:10)*100

for (i in 1:length(m)){
```

```
35   for (j in 1:loop){
36     table[i,j]=Prod_stat(m[i],n[i])
37   }
38 }
39
40 x=seq(-2, 4, 0.01)
41 x=x[!(x==0)]
42 f1=function(z1,z2){
43   12*dchisq(12*x+6*z1*z2,1) *exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-
       sqrt(3)/pi))/2/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))
44 }
45 y=integral(f1,bounds=list(z1=c(-Inf,Inf),z2=c(-Inf,Inf)), method='hcubature
       ')$value
46 plot(x[-c(151,250)],y[-c(151,250)],type='l',xlab='x', ylab='',main='Product
       Statistics density plot', col='blue')
47 lines(density(table[10,]), col='red',xlim=c(-2, 4))
```

A.1.3 Empirical quantiles of Sum Statistic

```
1  set.seed(1)
2  loop=10000
3  table=matrix(NA,10,loop)
4
5  #m,n same size
6  m=n=c(1:10)*100
7
8  for (i in 1:length(m)){
9    for (j in 1:loop){
10     table[i,j]=Sum_stat(m[i],n[i])
11   }
12 }
13
14 #quantile 0.20 0.1 0.05 0.01
15 v1=v2=v3=v4=c()
16 for (i in 1:length(m)){
17   v1[i]=quantile(table[i,], 0.8)
18   v2[i]=quantile(table[i,], 0.9)
19   v3[i]=quantile(table[i,], 0.95)
20   v4[i]=quantile(table[i,], 0.99)
21 }
22
23 #m,n different
24 m=c(1:10)*100
25 n=m/2
26 for (i in 1:length(m)){
27   for (j in 1:loop){
28     table[i,j]=Sum_stat(m[i],n[i])
29   }
30 }
31
32 v12=v22=v32=v42=c()
33 for (i in 1:length(m)){
34   v12[i]=quantile(table[i,], 0.8)
35   v22[i]=quantile(table[i,], 0.9)
36   v32[i]=quantile(table[i,], 0.95)
37   v42[i]=quantile(table[i,], 0.99)
```

```
}


#find critical value c for each quantile
#0.8
f=function(x){integral(function(z1,z2){
  exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-sqrt(3)/pi))/(2*pi*sqrt((2*
    pi-sqrt(3))/sqrt(3)/pi^2))*as.numeric(-z1*z2<x)},bounds=list(z1=c(-Inf,
    Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.8}
uniroot(f, lower=-2,upper=4)$root #0.6416644

#0.9
f=function(x){integral(function(z1,z2){
  exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-sqrt(3)/pi))/(2*pi*sqrt((2*
    pi-sqrt(3))/sqrt(3)/pi^2))*as.numeric(-z1*z2<x)},bounds=list(z1=c(-Inf,
    Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.9}
uniroot(f, lower=-2,upper=4)$root #1.131159

#0.95
f=function(x){integral(function(z1,z2){
  exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-sqrt(3)/pi))/(2*pi*sqrt((2*
    pi-sqrt(3))/sqrt(3)/pi^2))*as.numeric(-z1*z2<x)},bounds=list(z1=c(-Inf,
    Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.95}
uniroot(f, lower=-2,upper=4)$root #1.656609

#0.99
f=function(x){integral(function(z1,z2){
  exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-sqrt(3)/pi))/(2*pi*sqrt((2*
    pi-sqrt(3))/sqrt(3)/pi^2))*as.numeric(-z1*z2<x)},bounds=list(z1=c(-Inf,
    Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.99}
uniroot(f, lower=-2,upper=4)$root #2.960827



#plot
par(mfrow = c(2, 2))
plot(m, v1,type='l', ylim=c(0.5,0.7),xlab='m(n=m)',ylab='Sum Statistics')
abline(h=0.6416644, col='red')
plot(m, v12,type='l', ylim=c(0.5,0.7),xlab='m(n=m/2)',ylab='Sum Statistics')
abline(h=0.6416644, col='red')

plot(m, v2,type='l', ylim=c(1,1.2),xlab='m(n=m)',ylab='Sum Statistics')
abline(h=1.131159, col='red')
plot(m, v22,type='l', ylim=c(1,1.2),xlab='m(n=m/2)',ylab='Sum Statistics')
abline(h=1.131159, col='red')


par(mfrow = c(2, 2))
plot(m, v3,type='l', ylim=c(1.5,1.8),xlab='m(n=m)',ylab='Sum Statistics')
abline(h=1.656609, col='red')
plot(m, v32,type='l', ylim=c(1.5,1.8),xlab='m(n=m/2)',ylab='Sum Statistics')
abline(h=1.656609, col='red')
plot(m, v4,type='l', ylim=c(2.7,3.1),xlab='m(n=m)',ylab='Sum Statistics')
abline(h=2.960827, col='red')
plot(m, v42,type='l', ylim=c(2.7,3.1),xlab='m(n=m/2)',ylab='Sum Statistics')
abline(h=2.960827, col='red')
```

A.1.4 Empirical quantiles of Product Statistic

```
set.seed(1)
loop=10000
table=matrix(NA,10,loop)

#m,n same size
m=n=c(1:10)*100

for (i in 1:length(m)){
  for (j in 1:loop){
    table[i,j]=Prod_stat(m[i],n[i])
  }
}

#quantile 0.20 0.1 0.05 0.01
v1=v2=v3=v4=c()
for (i in 1:length(m)){
  v1[i]=quantile(table[i,], 0.8)
  v2[i]=quantile(table[i,], 0.9)
  v3[i]=quantile(table[i,], 0.95)
  v4[i]=quantile(table[i,], 0.99)
}

#m,n different
set.seed(1)
m=c(1:10)*100
n=m/2
for (i in 1:length(m)){
  for (j in 1:loop){
    table[i,j]=Prod_stat(m[i],n[i])
  }
}

v12=v22=v32=v42=c()
for (i in 1:length(m)){
  v12[i]=quantile(table[i,], 0.8)
  v22[i]=quantile(table[i,], 0.9)
  v32[i]=quantile(table[i,], 0.95)
  v42[i]=quantile(table[i,], 0.99)
}


#find critical value c for each quantile
#0.8
f=function(x){integral(function(z1,z2){
  pchisq(12*x+6*z1*z2,1) *exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-
    sqrt(3)/pi))/2/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))},bounds=list(z1=c
    (-Inf,Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.8}
uniroot(f, lower=-2,upper=4)$root #0.4379003

#0.9
f=function(x){integral(function(z1,z2){
  pchisq(12*x+6*z1*z2,1) *exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-
    sqrt(3)/pi))/2/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))},bounds=list(z1=c
    (-Inf,Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.9}
uniroot(f, lower=-2,upper=4)$root #0.6818296

```

```
#0.95
f=function(x){integral(function(z1,z2){
  pchisq(12*x+6*z1*z2,1) *exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-
    sqrt(3)/pi))/2/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))},bounds=list(z1=c
    (-Inf,Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.95}
uniroot(f, lower=-2,upper=4)$root #0.9383627

#0.99
f=function(x){integral(function(z1,z2){
  pchisq(12*x+6*z1*z2,1) *exp(-(z1^2+sqrt(3)*z1*z2+sqrt(3)*pi*z2^2/2)/(2-
    sqrt(3)/pi))/2/pi/sqrt((2*pi-sqrt(3))/(sqrt(3)*pi^2))},bounds=list(z1=c
    (-Inf,Inf),z2=c(-Inf,Inf)), method='hcubature')$value-0.99}
uniroot(f, lower=-2,upper=4)$root #1.570562



#plot
par(mfrow = c(2, 2))
plot(m, v1,type='l', ylim=c(0.3,0.5),xlab='m(n=m)',ylab='Product Statistics
    ')
abline(h=0.4379003, col='red')
plot(m, v12,type='l', ylim=c(0.3,0.5),xlab='m(n=m/2)',ylab='Product
    Statistics')
abline(h=0.4379003, col='red')

plot(m, v2,type='l', ylim=c(0.6,0.8),xlab='m(n=m)',ylab='Product Statistics
    ')
abline(h=0.6818296, col='red')
plot(m, v22,type='l', ylim=c(0.6,0.8),xlab='m(n=m/2)',ylab='Product
    Statistics')
abline(h=0.6818296, col='red')


par(mfrow = c(2, 2))
plot(m, v3,type='l', ylim=c(0.8,1),xlab='m(n=m)',ylab='Product Statistics')
abline(h=0.9383627, col='red')
plot(m, v32,type='l', ylim=c(0.8,1),xlab='m(n=m/2)',ylab='Product Statistics
    ')
abline(h=0.9383627, col='red')

plot(m, v4,type='l', ylim=c(1.35,1.65),xlab='m(n=m)',ylab='Product
    Statistics')
abline(h=1.570562, col='red')
plot(m, v42,type='l', ylim=c(1.35,1.65),xlab='m(n=m/2)',ylab='Product
    Statistics')
abline(h=1.570562, col='red')
```

A.1.5 Functions and null hypothesis

```
library(ddalpha)
library(MASS)
library(matrixStats)
library(mvtnorm)

# Five stats
QQ_test=function(Fm,Gn,type.depth){
```

```
  if(type.depth==1){
  depth_Fm_F=depth.Mahalanobis(x=Fm,data=Fm) # ddalpha
  depth_Gn_F=depth.Mahalanobis(x=Gn,data=Fm)
  depth_Fm_G=depth.Mahalanobis(x=Fm,data=Gn)
  depth_Gn_G=depth.Mahalanobis(x=Gn,data=Gn)
  }
  if(type.depth==2){
  depth_Fm_F=depth.spatial(x=Fm,data=Fm) # ddalpha
  depth_Gn_F=depth.spatial(x=Gn,data=Fm)
  depth_Fm_G=depth.spatial(x=Fm,data=Gn)
  depth_Gn_G=depth.spatial(x=Gn,data=Gn)
  }
  if(type.depth==3){
  depth_Fm_F=depth.projection(x=Fm,data=Fm) # ddalpha
  depth_Gn_F=depth.projection(x=Gn,data=Fm)
  depth_Fm_G=depth.projection(x=Fm,data=Gn)
  depth_Gn_G=depth.projection(x=Gn,data=Gn)
  }
if(type.depth==4){
  depth_Fm_F=depth.Mahalanobis(x=Fm,data=Fm, mah.estimate = "MCD") # ddalpha
  depth_Gn_F=depth.Mahalanobis(x=Gn,data=Fm, mah.estimate = "MCD")
  depth_Fm_G=depth.Mahalanobis(x=Fm,data=Gn, mah.estimate = "MCD")
  depth_Gn_G=depth.Mahalanobis(x=Gn,data=Gn, mah.estimate = "MCD")
  }

  V_q=c()
  for (i in 1:length(depth_Gn_F)) {
    V_q[i]=(sum(depth_Gn_F[i]>=depth_Fm_F))/length(depth_Fm_F)
  }
  Q_test_rF=mean(V_q)

  V_q=c()
  for (i in 1:length(depth_Fm_G)) {
    V_q[i]=(sum(depth_Fm_G[i]>=depth_Gn_G))/length(depth_Gn_G)
  }
  Q_test_rG=mean(V_q)
  return(c(Q_test_rF,Q_test_rG))
}

# Compute Q_F_Chi,Q_G_Chi,A,M,W
# compute D (difference), Min stat, absolute value
five_in_all=function(Fm,Gn,type.depth){
  m=dim(Fm)[1]
  n=dim(Gn)[1]
  wn=n/(n+m)
  wm=m/(n+m)
  QQ=QQ_test(Fm,Gn,type.depth)
  Q_F=QQ[1]
  Q_G=QQ[2]
  Q_F_Chi=((1/n+1/m)*(1/12))^(-1)*(Q_F-1/2)^2 ####
  Q_G_Chi=((1/n+1/m)*(1/12))^(-1)*(Q_G-1/2)^2
  A=0.5*(((1/n+1/m)*(1/12))^(-1))*((Q_G-0.5)^2+(Q_F-0.5)^2)
  M=(((1/n+1/m)*(1/12))^(-1))*max((Q_G-0.5)^2,(Q_F-0.5)^2) #max
  W=(1/(wn+wm))*(((1/n+1/m)*(1/12))^(-1))*(wn*(Q_G-0.5)^2+wm*(Q_F-0.5)^2)
  Mn=((1/n+1/m)*(1/12))^(-1/2)*(1/2-min(Q_G,Q_F)) #min
  P=-m*n*(Q_F*Q_G-1/4)/(m+n) #Product
  S=-m*n*(Q_F+Q_G-1)/(m+n) #sum
  return(c(Q_F_Chi,Q_G_Chi,A,M,W,Mn,P,S))
```

```
}


### H stat
rtable=function(data1,data2,ref,type.depth){
  data_all=rbind(data1,data2)
  if(type.depth==1){
  depth_data1=depth.Mahalanobis(x=data1,data=ref)
  depth_data2=depth.Mahalanobis(x=data2,data=ref)
  depth_all=depth.Mahalanobis(x=data_all,data=ref)
  }
  if(type.depth==2){
  depth_data1=depth.spatial(x=data1,data=ref)
  depth_data2=depth.spatial(x=data2,data=ref)
  depth_all=depth.spatial(x=data_all,data=ref)
  }
  if(type.depth==3){
  depth_data1=depth.projection(x=data1,data=ref)
  depth_data2=depth.projection(x=data2,data=ref)
  depth_all=depth.projection(x=data_all,data=ref)
  }
  col_1=c()
  for (i in 1:length(depth_data1)) {
    col_1[i]=sum(depth_data1[i]>=depth_all)
  }
  col_2=c()
  for (i in 1:length(depth_data2)) {
    col_2[i]=sum(depth_data2[i]>=depth_all)
  }
  v_length <- max(length(col_1), length(col_2))
  length(col_1)=v_length
  length(col_2)=v_length
  output=cbind(col_1,col_2)
  return(output)
}


H_test<-function(data1,data2,type.depth){
  table_Rij=rtable(data1,data2,ref=data1,type.depth)
  R_3=sum(table_Rij[,1],na.rm=TRUE)
  R_4=sum(table_Rij[,2],na.rm=TRUE)

  n_3=sum(!is.na(table_Rij[,1]))
  n_4=sum(!is.na(table_Rij[,2]))

  table_Rij=rtable(data1,data2,ref=data2,type.depth)
  table_Rij
  R_5=sum(table_Rij[,1],na.rm=TRUE)
  R_6=sum(table_Rij[,2],na.rm=TRUE)

  n_5=sum(!is.na(table_Rij[,1]))
  n_6=sum(!is.na(table_Rij[,2]))
  n=n_5+n_6
  H=12/(n*(n+1)*2)*(R_3^2/n_3+R_4^2/n_4+R_5^2/n_5+R_6^2/n_6)-3*(n+1)
  return(H)
}


```

```
BDBR=function(data1, data2, type.depth){
  table_Rij1=rtable(data1, data2, ref=data1, type.depth)
  table_Rij2=rtable(data1, data2, ref=data2, type.depth)
  n1=sum(!is.na(table_Rij1[,1]))
  n2=sum(!is.na(table_Rij2[,2]))
  N=n1+n2

  R_F1=sort(table_Rij1[,2])
  R_F2=sort(table_Rij2[,1])

  B_F1=rep(0,n2)
  B_F2=rep(0,n1)

  for (i in 1:n2){
    E=(N+1)*i/(n2+1)
    V=i*(1-i/(n2+1))*n1*(N+1)/(n2+1)/(n2+2)
    B_F1[i]=(R_F1[i]-E)^2/V
  }
  BF1=sum(B_F1)/n2

  for (j in 1:n1){
    E=(N+1)*j/(n1+1)
    V=j*(1-j/(n1+1))*n2*(N+1)/(n1+1)/(n1+2)
    B_F2[j]=(R_F2[j]-E)^2/V
  }
  BF2=sum(B_F2)/n1

  B=max(BF1, BF2)

  return(B)
}
```

```
# Empirical Distribution
set.seed(1)
mu1 <- c(0,0)
mu2 <- c(0,0)
sigma1 <- matrix(c(1,0,0,1), nc = 2)
sigma2 <- matrix(c(1,0,0,1), nc = 2)
Loop=10000
m=n=c(1:10)*50
Type.Depth=c(1,2,3)
for(iT in 1:length(Type.Depth)){
  for(im in 1:length(m)){
    five_table=c()
    for (i in 1:Loop) {
      Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
      Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
      five_table=rbind(five_table,five_in_all(Fm,Gn,Type.Depth[iT]))
    }
    colnames(five_table)=c('Q_F_Chi','Q_G_Chi','A','M','W','Mn','P','S')
    # Save an object to a file
    my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/five_in_all/null",
                  Type.Depth[iT],m[im],n[im], ".rds",sep = "-")
    saveRDS(five_table, file = my_data)
  }
}

```

```
26
27
28 # Empirical Distribution
29 set.seed(1)
30 mu1 <- c(0,0)
31 mu2 <- c(0,0)
32 sigma1 <- matrix(c(1,0,0,1), nc = 2)
33 sigma2 <- matrix(c(1,0,0,1), nc = 2)
34 Loop=10000
35 m=c(1:10)*100
36 n=m/2
37 Type.Depth=c(1,2,3)
38 for(iT in 1:length(Type.Depth)){
39   for(im in 1:length(m)){
40     five_table=c()
41     for (i in 1:Loop) {
42       Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
43       Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
44       five_table=rbind(five_table,five_in_all(Fm,Gn,Type.Depth[iT]))
45     }
46     colnames(five_table)=c('Q_F_Chi','Q_G_Chi','A','M','W','Mn','P','S')
47     # Save an object to a file
48     my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
       hypothesis/five_in_all/null",
49                   Type.Depth[iT],m[im],n[im], ".rds",sep = "-")
50     saveRDS(five_table, file = my_data)
51   }
52 }
53
54
55 # Empirical Distribution
56 set.seed(1)
57 mu1 <- c(0,0)
58 mu2 <- c(0,0)
59 sigma1 <- matrix(c(1,0,0,1), nc = 2)
60 sigma2 <- matrix(c(1,0,0,1), nc = 2)
61 Loop=10000
62 m=n=c(1:10)*100
63 Type.Depth=c(1,2,3)
64 for(iT in 1:length(Type.Depth)){
65   for(im in 1:length(m)){
66     h=c()
67     for (i in 1:Loop) {
68       Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
69       Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
70       h=rbind(h,H_test(Fm,Gn,Type.Depth[iT]))
71     }
72     colnames(h)=c('DbR')
73     # Save an object to a file
74     my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
       hypothesis/dbr/null",
75                   Type.Depth[iT],m[im],n[im], "DbR.rds",sep = "-")
76     saveRDS(h, file = my_data)
77   }
78 }
79
80
81
```

```
# Empirical Distribution
set.seed(1)
mu1 <- c(0,0)
mu2 <- c(0,0)
sigma1 <- matrix(c(1,0,0,1), nc = 2)
sigma2 <- matrix(c(1,0,0,1), nc = 2)
Loop=10000
m=c(1:10)*50
n=m/2
Type.Depth=c(1,2,3)
for(iT in 1:length(Type.Depth)){
  for(im in 1:length(m)){
    h=c()
    for (i in 1:Loop) {
      Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
      Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
      h=rbind(h,H_test(Fm,Gn,Type.Depth[iT]))
    }
    colnames(h)=c('DbR')
    # Save an object to a file
    my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/dbr/null",
                  Type.Depth[iT],m[im],n[im], "DbR.rds",sep = "-")
    saveRDS(h, file = my_data)
  }
}

# Empirical Distribution
#n=m
set.seed(1)
mu1 <- c(0,0)
mu2 <- c(0,0)
sigma1 <- matrix(c(1,0,0,1), nc = 2)
sigma2 <- matrix(c(1,0,0,1), nc = 2)
Loop=10000
m=n=c(1:10)*50
Type.Depth=c(1,2,3)
for(iT in 1:length(Type.Depth)){
  for(im in 1:length(m)){
    h=c()
    for (i in 1:Loop) {
      Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
      Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
      h=rbind(h,BDBR(Fm,Gn,Type.Depth[iT]))
    }
    colnames(h)=c('BDbR')
    # Save an object to a file
    my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/bdbr/null",
                  Type.Depth[iT],m[im],n[im], "BDbR.rds",sep = "-")
    saveRDS(h, file = my_data)
  }
}

#n=m/2
set.seed(1)
mu1 <- c(0,0)
mu2 <- c(0,0)
```

```
sigma1 <- matrix(c(1,0,0,1), nc = 2)
sigma2 <- matrix(c(1,0,0,1), nc = 2)
Loop=10000
m=c(1:10)*50
n=m/2
Type.Depth=c(1,2,3)
for(iT in 1:length(Type.Depth)){
  for(im in 1:length(m)){
    h=c()
    for (i in 1:Loop) {
      Fm <- rbind(mvrnorm(m[im], mu1 ,sigma1))
      Gn <- rbind(mvrnorm(n[im], mu2 ,sigma2))
      h=rbind(h,BDBR(Fm,Gn,Type.Depth[iT]))
    }
    colnames(h)=c('BDbR')
    # Save an object to a file
    my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/bdbr/null",
                  Type.Depth[iT],m[im],n[im], "BDbR.rds",sep = "-")
    saveRDS(h, file = my_data)
  }
}
```

A.1.6 Empirical quantiles of Minimum Statistic

```
m=n=c(1:10)*100
Type.Depth=c(1,2,3)
Q=array(NA,dim=c(length(Type.Depth),length(m),8))
for(iT in 1:length(Type.Depth)){
for(im in 1:length(m)){

my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/five_in_all/null",
Type.Depth[iT],m[im],n[im], ".rds",sep = "-")

five_table=readRDS(file =my_data)
five_table=five_table[1:10000,]
Q[iT,im,] <- apply(five_table,2,function (x) quantile(x,probs = 0.95))
}
}

par(mar=c(4,4,0,0)+0.1,fig=c(0,0.5,1/4,2/4) , new=TRUE)
library("fdrtool")
plot(m,Q[1,,7],ylim=c(1.95, 2.55),type='l',col='black',lwd=2,
     #main="Log of Normalizing Constant Approximation",
     ylab="Minimum Statistics",
     xlab="m (n=m)",cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(h=qhalfnorm(0.95),col="red",lwd=2)

lines(m,Q[2,,7],lty=2,col="blue",lwd=2)
lines(m,Q[3,,7],lty=3,col="purple",lwd=2)
legend("topright",
       c("Mahalanobis","Spatial","Projection"),
       lty=c(1,2,3),col=c("black","blue","purple"),bty='n',lwd=2,cex=1.5,x.
    intersp=2)

```

```
m=n=c(1:10)*100
n=m/2
Type.Depth=c(1,2,3)
Q=array(NA,dim=c(length(Type.Depth),length(m),8))
for(iT in 1:length(Type.Depth)){
for(im in 1:length(m)){

my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/five_in_all/null",
Type.Depth[iT],m[im],n[im], ".rds",sep = "-")

five_table=readRDS(file =my_data)
five_table=five_table[1:10000,]
Q[iT,im,] <- apply(five_table,2,function (x) quantile(x,probs = 0.95))
}
}

par(mar=c(4,4,0,0)+0.1,fig=c(0.5,1,1/4,2/4), new=TRUE)
library("fdrtool")
plot(m,Q[1,,7],ylim=c(1.95,2.55),type='l',col='black',lwd=2,
     #main="Log of Normalizing Constant Approximation",
     ylab="Minimum Statistics",
     xlab="m (n=m/2)",cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(h=qhalfnorm(0.95),col="red",lwd=2)

lines(m,Q[2,,7],lty=2,col="blue",lwd=2)
lines(m,Q[3,,7],lty=3,col="purple",lwd=2)
legend("topright",
       c("Mahalanobis","Spatial","Projection"),
       lty=c(1,2,3),col=c("black","blue","purple"),bty='n',lwd=2,cex=1.5,x.
    intersp=2)
```

A.2 Chapter 3

A.2.1 Univariate distribution

```
library(MASS)
QQ_test=function(Fm,Gn){
  depth_Fm_F=1/(1+(Fm-mean(Fm))^2)
  depth_Gn_F=1/(1+(Gn-mean(Fm))^2)
  depth_Fm_G=1/(1+(Fm-mean(Gn))^2)
  depth_Gn_G=1/(1+(Gn-mean(Gn))^2)

  V_q=c()
  for (i in 1:length(depth_Gn_F)) {
    V_q[i]=(sum(depth_Gn_F[i]>=depth_Fm_F))/length(depth_Fm_F)
  }
  Q_test_rF=mean(V_q)

  V_q=c()
  for (i in 1:length(depth_Fm_G)) {
    V_q[i]=(sum(depth_Fm_G[i]>=depth_Gn_G))/length(depth_Gn_G)
  }
  Q_test_rG=mean(V_q)
  return(c(Q_test_rF,Q_test_rG))
```

```
}

five_in_all=function(Fm,Gn){
  m=length(Fm)
  n=length(Gn)

  QQ=QQ_test(Fm,Gn)
  Q_F=QQ[1]
  Q_G=QQ[2]

  M=(((1/n+1/m)*(1/12))^(-1))*max((Q_G-0.5)^2,(Q_F-0.5)^2) #max
  Mn=((1/n+1/m)*(1/12))^(-1/2)*(1/2-min(Q_G,Q_F)) #min
  P=-m*n*(Q_F*Q_G-1/4)/(m+n) #Product
  S=-m*n*(Q_F+Q_G-1)/(m+n) #sum
  return(c(M,Mn,P,S))
}


rtable=function(data1,data2,ref){
  data_all=c(data1,data2)
  depth_data1=1/(1+(data1-mean(ref))^2)
  depth_data2=1/(1+(data2-mean(ref))^2)
  depth_all=1/(1+(data_all-mean(ref))^2)

  col_1=c()
  for (i in 1:length(depth_data1)) {
    col_1[i]=sum(depth_data1[i]>=depth_all)
  }
  col_2=c()
  for (i in 1:length(depth_data2)) {
    col_2[i]=sum(depth_data2[i]>=depth_all)
  }
  v_length <- max(length(col_1), length(col_2))
  length(col_1)=v_length
  length(col_2)=v_length
  output=cbind(col_1,col_2)
  return(output)
}

H_test<-function(data1,data2){
  table_Rij=rtable(data1,data2,ref=data1)
  R_3=sum(table_Rij[,1],na.rm=TRUE)
  R_4=sum(table_Rij[,2],na.rm=TRUE)

  n_3=sum(!is.na(table_Rij[,1]))
  n_4=sum(!is.na(table_Rij[,2]))

  table_Rij=rtable(data1,data2,ref=data2)
  table_Rij
  R_5=sum(table_Rij[,1],na.rm=TRUE)
  R_6=sum(table_Rij[,2],na.rm=TRUE)

  n_5=sum(!is.na(table_Rij[,1]))
  n_6=sum(!is.na(table_Rij[,2]))
  n=n_5+n_6
  H=12/(n*(n+1)*2)*(R_3^2/n_3+R_4^2/n_4+R_5^2/n_5+R_6^2/n_6)-3*(n+1)
  return(H)
}
```

```
78
79
80 BDBR=function(data1, data2){
81   table_Rij1=rtable(data1, data2, ref=data1)
82   table_Rij2=rtable(data1, data2, ref=data2)
83   n1=sum(!is.na(table_Rij1[,1]))
84   n2=sum(!is.na(table_Rij2[,2]))
85   N=n1+n2
86
87   R_F1=sort(table_Rij1[,2])
88   R_F2=sort(table_Rij2[,1])
89
90   B_F1=rep(0,n2)
91   B_F2=rep(0,n1)
92
93   for (i in 1:n2){
94     E=(N+1)*i/(n2+1)
95     V=i*(1-i/(n2+1))*n1*(N+1)/(n2+1)/(n2+2)
96     B_F1[i]=(R_F1[i]-E)^2/V
97   }
98   BF1=sum(B_F1)/n2
99
100  for (j in 1:n1){
101    E=(N+1)*j/(n1+1)
102    V=j*(1-j/(n1+1))*n2*(N+1)/(n1+1)/(n1+2)
103    B_F2[j]=(R_F2[j]-E)^2/V
104  }
105  BF2=sum(B_F2)/n1
106
107  B=max(BF1, BF2)
108
109  return(B)
110 }
111
112
113 #two sample wilcoxon mann whitney
114 Wilcoxon=function(Fm,Gn){
115   m=length(Fm)
116   n=length(Gn)
117
118   u=c()
119   for (i in 1:m){
120     u[i]=sum(Fm[i]<Gn)/n
121   }
122   U=mean(u)
123   return(U)
124 }
125
126 Permu=function(Fm,Gn,m,n,size){
127   T_star=five_in_all(Fm, Gn)
128   B=(m+n)/size
129   Loop=200
130   T_b=matrix(NA,Loop,4)
131   pvalue=c()
132   for(loop in 1:Loop){
133
134     ind=sample( B, m/size)
135
```

```
136     ind=sort(ind)
137
138     Fm.ind=NULL
139     for(i in 1:length(ind))
140       Fm.ind=c(Fm.ind, ((ind[i]-1)*size+1): (ind[i]*size) )
141
142     Fm.b=c(Fm,Gn)[Fm.ind]
143     Gn.b=c(Fm,Gn)[-Fm.ind]
144     T_b[loop,]=five_in_all(Fm.b, Gn.b)
145   }
146
147   # pvalue=sum(T_b>T_star)/Loop
148   pvalue[1]=sum(T_b[,3]>T_star[3])/Loop #prod
149   pvalue[2]=sum(T_b[,4]>T_star[4])/Loop #sum
150
151   return(pvalue)
152 }
153
154
155 rept=1000
156 m=c(1:10)*50
157 n=m
158 p_quantile_sum=p_quantile_prod=c()
159 for (im in 1:10){
160   p_matrix=matrix(NA,rept,2)
161   for (loop in 1:rept){
162     Fm=rnorm(m[im],0,1)
163     Gn=rnorm(n[im],0,1)
164     p_matrix[loop,]=Permu(Fm,Gn,m[im], n[im], 25)
165   }
166   p_quantile_prod[im]=quantile(p_matrix[,1],0.05)
167   p_quantile_sum[im]=quantile(p_matrix[,2],0.05)
168 }
169 p_quantile_sum
170 p_quantile_prod
171 saveRDS(p_quantile_sum,"C:/Users/ychen462/Desktop/Data depth new/1 dim
       Euclidean/Permutation/p-value-sum-samesize.rds")
172 saveRDS(p_quantile_prod,"C:/Users/ychen462/Desktop/Data depth new/1 dim
       Euclidean/Permutation/p-value-prod-samesize.rds")
173
174 rept=1000
175 m=c(1:10)*50
176 n=m/2
177 p_quantile_sum=p_quantile_prod=c()
178 for (im in 1:10){
179   p_matrix=matrix(NA,rept,2)
180   for (loop in 1:rept){
181     Fm=rnorm(m[im],0,1)
182     Gn=rnorm(n[im],0,1)
183     p_matrix[loop,]=Permu(Fm,Gn,m[im], n[im], 25)
184   }
185   p_quantile_prod[im]=quantile(p_matrix[,1],0.05)
186   p_quantile_sum[im]=quantile(p_matrix[,2],0.05)
187 }
188 p_quantile_sum
189 p_quantile_prod
190 saveRDS(p_quantile_sum,"C:/Users/ychen462/Desktop/Data depth new/1 dim
       Euclidean/Permutation/p-value-sum-diffsize.rds")
```

```
saveRDS(p_quantile_prod,"C:/Users/ychen462/Desktop/Data depth new/1 dim
    Euclidean/Permutation/p-value-prod-diffsize.rds")
```

Scale change

```
set.seed(1)
#set m,n values, mu and sigma
m=c(1:10)*50
n=m
mu1 <- 0.25
mu2 <- 0.25
sigma1 <- 1
sigma2 <- 1.15

rept=1000 #number of repetitions
Q=matrix(NA, length(m),6 )
powers=matrix(NA, length(m),7 )


for (im in 1:10){
  for (loop in 1:rept){
    Fm=rnorm(m[im],mu1, sigma1)
    Gn=rnorm(n[im],mu2, sigma2)

    Fm_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    Permutation/Fm Gn1-1/",
                  m[im],n[im],loop, "Fm.rds",sep = "-")
    Gn_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    Permutation/Fm Gn1-1/",
                  m[im],n[im],loop, "Gn.rds",sep = "-")
    saveRDS(Fm, file=Fm_data)
    saveRDS(Gn, file=Gn_data)
  }

  #read data from null hypothesis
  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    null hypothesis/five_in_all/null",
                m[im],n[im],".rds",sep = "-")
  five_table=readRDS(file =my_data)

  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    null hypothesis/dbr/null",
                m[im],n[im],"DbR.rds",sep = "-")
  DbR_table=readRDS(file =my_data)

  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    null hypothesis/bdbr/null",
                m[im],n[im],"BDbR.rds",sep = "-")
  BDbR_table=readRDS(file =my_data)

  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
    null hypothesis/Wilcoxon/null",
                m[im],n[im],"Wilcoxon.rds",sep = "-")
  Wilcoxon_table=readRDS(file =my_data)


```

```
  Q[im, ]=c(apply(five_table,2,function (x) quantile(x,probs = 0.95))[c(1,2)],
            quantile(DbR_table,probs = 0.95),
            quantile(BDbR_table,prob=0.95),
            quantile(Wilcoxon_table,prob=c(0.025,0.975)))


  #powers
  h=c()
  temp=matrix(NA,rept,2)
  for (loop in 1:rept){
    Fm=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/Permutation/Fm Gn1-1/",
                     m[im],n[im],loop, "Fm.rds",sep = "-"))
    Gn=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/Permutation/Fm Gn1-1/",
                     m[im],n[im],loop, "Gn.rds",sep = "-"))
    h=rbind(h,c(five_in_all(Fm,Gn)[c(1,2)],
                H_test(Fm,Gn),
                BDBR(Fm,Gn),
                Wilcoxon(Fm,Gn)))
    temp[loop,]=Permu(Fm,Gn,m[im], n[im], 25)
  }

  for(j in 1:4){
    powers[im,j]=mean(h[,j]>=Q[im,j])
  }
  powers[im,5]=mean((h[,5]<=Q[im,5])+(h[,5]>=Q[im,6]))


  p_quantile_sum=readRDS("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/Permutation/p-value-sum-samesize.rds")
  p_quantile_prod=readRDS("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/Permutation/p-value-prod-samesize.rds")


  #powers for prod, sum
  powers[im,6]=sum(temp[,1]<p_quantile_prod[im])/rept
  powers[im,7]=sum(temp[,2]<p_quantile_sum[im])/rept


}
powers
my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/Permutation/power1-1.rds",sep = "-")
saveRDS(powers, file = my_data)



####n=m/2
set.seed(1)
#set m,n values, mu and sigma
m=c(1:10)*50
n=m/2
mu1 <- 0.25
mu2 <- 0.25
sigma1 <- 1
sigma2 <- 1.15
```

```
98
99 rept=1000 #number of repetitions
100 Q=matrix(NA, length(m),6 )
101 powers=matrix(NA, length(m),7 )
102
103 for (im in 1:10){
104   for (loop in 1:rept){
105     Fm=rnorm(m[im],mu1, sigma1)
106     Gn=rnorm(n[im],mu2, sigma2)
107
108     Fm_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      Permutation/Fm Gn1-2/",
109                   m[im],n[im],loop, "Fm.rds",sep = "-")
110     Gn_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      Permutation/Fm Gn1-2/",
111                   m[im],n[im],loop, "Gn.rds",sep = "-")
112     saveRDS(Fm, file=Fm_data)
113     saveRDS(Gn, file=Gn_data)
114   }
115
116
117   #read data from null hypothesis
118   my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      null hypothesis/five_in_all/null",
119                 m[im],n[im],".rds",sep = "-")
120   five_table=readRDS(file =my_data)
121
122   my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      null hypothesis/dbr/null",
123                 m[im],n[im],"DbR.rds",sep = "-")
124   DbR_table=readRDS(file =my_data)
125
126   my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      null hypothesis/bdbr/null",
127                 m[im],n[im],"BDbR.rds",sep = "-")
128   BDbR_table=readRDS(file =my_data)
129
130   my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      null hypothesis/Wilcoxon/null",
131                 m[im],n[im],"Wilcoxon.rds",sep = "-")
132   Wilcoxon_table=readRDS(file =my_data)
133
134
135
136   Q[im, ]=c(apply(five_table,2,function (x) quantile(x,probs = 0.95))[c(1,2)
      ],
137             quantile(DbR_table,probs = 0.95),
138             quantile(BDbR_table,prob=0.95),
139             quantile(Wilcoxon_table,prob=c(0.025,0.975)))
140
141
142   #powers
143   h=c()
144   temp=matrix(NA,rept,2)
145   for (loop in 1:rept){
146     Fm=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/1 dim
      Euclidean/Permutation/Fm Gn1-2/",
147                      m[im],n[im],loop, "Fm.rds",sep = "-"))
```

```
148     Gn=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/1 dim
        Euclidean/Permutation/Fm Gn1-2/",
149                      m[im],n[im],loop, "Gn.rds",sep = "-"))
150     h=rbind(h,c(five_in_all(Fm,Gn)[c(1,2)],
151                 H_test(Fm,Gn),
152                 BDBR(Fm,Gn),
153                 Wilcoxon(Fm,Gn)))
154     temp[loop,]=Permu(Fm,Gn,m[im], n[im], 25)
155   }
156
157   for(j in 1:4){
158     powers[im,j]=mean(h[,j]>=Q[im,j])
159   }
160   powers[im,5]=mean((h[,5]<=Q[im,5])+(h[,5]>=Q[im,6]))
161
162
163   p_quantile_sum=readRDS("C:/Users/ychen462/Desktop/Data depth new/1 dim
        Euclidean/Permutation/p-value-sum-diffsize.rds")
164   p_quantile_prod=readRDS("C:/Users/ychen462/Desktop/Data depth new/1 dim
        Euclidean/Permutation/p-value-prod-diffsize.rds")
165
166   #powers for prod, sum
167   powers[im,6]=sum(temp[,1]<p_quantile_prod[im])/rept
168   powers[im,7]=sum(temp[,2]<p_quantile_sum[im])/rept
169 }
170
171 powers
172 my_data=paste("C:/Users/ychen462/Desktop/Data depth new/1 dim Euclidean/
      permutation/power1-2.rds",sep = "-")
173 saveRDS(powers, file = my_data)
```

Mean change

```
1 mu1 <- 0
2 mu2 <- 0.25
3 sigma1 <- 1
4 sigma2 <- 1
```

Both change

```
1 mu1 <- 0.25
2 mu2 <- 0
3 sigma1 <- 1
4 sigma2 <- 1.15
```

A.2.2 Multivariate distribution

```
1 Permu=function(Fm,Gn,m,n,size,type.depth){
2   T_star=five_in_all(Fm, Gn,type.depth)
3   B=(m+n)/size
4   Loop=200
5   T_b=matrix(NA,Loop,8)
```

```
  pvalue=c()
  for(loop in 1:Loop){

    ind=sample( B, m/size)

    ind=sort(ind)

    Fm.ind=NULL
    for(i in 1:length(ind))
      Fm.ind=c(Fm.ind, ((ind[i]-1)*size+1): (ind[i]*size) )

    Fm.b=rbind(Fm,Gn)[Fm.ind,]
    Gn.b=rbind(Fm,Gn)[-Fm.ind,]
    T_b[loop,]=five_in_all(Fm.b, Gn.b,type.depth)
  }

  # pvalue=sum(T_b>T_star)/Loop
  pvalue[1]=sum(T_b[,7]>T_star[7])/Loop #prod
  pvalue[2]=sum(T_b[,8]>T_star[8])/Loop #sum

  return(pvalue)
}

mu <- c(0,0)
sigma <- matrix(c(1,0,0,1), nc = 2)

rept=1000
m=c(1:10)*50
type.depth=c(1,2,3)
n=m
p_quantile_sum=p_quantile_prod=matrix(NA,10,3)

for (iT in type.depth){
for (im in 1:10){
  p_matrix=matrix(NA,rept,2)
  for (loop in 1:rept){
    Fm=mvrnorm(m[im],mu, sigma)
    Gn=mvrnorm(n[im],mu, sigma)
    p_matrix[loop,]=Permu(Fm,Gn,m[im], n[im], 25,iT)
  }
  p_quantile_prod[im,iT]=quantile(p_matrix[,1],0.05)
  p_quantile_sum[im,iT]=quantile(p_matrix[,2],0.05)
}
}
p_quantile_sum
p_quantile_prod
saveRDS(p_quantile_sum,"C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/p-value-sum-samesize.rds")
saveRDS(p_quantile_prod,"C:/Users/ychen462/Desktop/Data depth new/two sample
    /Permutation/p-value-prod-samesize.rds")


rept=1000
m=c(1:10)*50
type.depth=c(1,2,3)
n=m/2
p_quantile_sum=p_quantile_prod=matrix(NA,10,3)

```

```
for (iT in type.depth){
  for (im in 1:10){
    p_matrix=matrix(NA,rept,2)
    for (loop in 1:rept){
      Fm=mvrnorm(m[im],mu, sigma)
      Gn=mvrnorm(n[im],mu, sigma)
      p_matrix[loop,]=Permu(Fm,Gn,m[im], n[im], 25,iT)
    }
    p_quantile_prod[im,iT]=quantile(p_matrix[,1],0.05)
    p_quantile_sum[im,iT]=quantile(p_matrix[,2],0.05)
  }
}
p_quantile_sum
p_quantile_prod
saveRDS(p_quantile_sum,"C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/p-value-sum-diffsize.rds")
saveRDS(p_quantile_prod,"C:/Users/ychen462/Desktop/Data depth new/two sample
    /Permutation/p-value-prod-diffsize.rds")
```

Scale change

```
set.seed(1)
#set m,n values, mu and sigma
m=c(1:10)*50
n=m
type.depth=c(1,2,3)
mu1 <- c(0,0)
mu2 <- c(0,0)
sigma1 <- matrix(c(1,0,0,1), nc = 2)
sigma2 <- matrix(c(1,0.5,0.5,1), nc = 2)



rept=1000 #number of repetitions

Q=array(NA,dim=c(length(type.depth),length(m),4))
powers=array(NA,dim=c(length(type.depth),length(m),6))

for (iT in type.depth){
  for (im in 1:10){
    for (loop in 1:rept){
      Fm=mvrnorm(m[im],mu1, sigma1)
      Gn=mvrnorm(n[im],mu2, sigma2)

      Fm_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Fm Gn1-1/",
                    iT,m[im],n[im],loop, "Fm.rds",sep = "-")
      Gn_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Fm Gn1-1/",
                    iT,m[im],n[im],loop, "Gn.rds",sep = "-")
      saveRDS(Fm, file=Fm_data)
      saveRDS(Gn, file=Gn_data)
    }


    #read data from null hypothesis
```

```
34  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/five_in_all/null",
35                iT,m[im],n[im],".rds",sep = "-")
36  five_table=readRDS(file =my_data)
37
38  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/dbr/null",
39                iT,m[im],n[im],"DbR.rds",sep = "-")
40  DbR_table=readRDS(file =my_data)
41
42  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
    hypothesis/bdbr/null",
43                iT,m[im],n[im],"BDbR.rds",sep = "-")
44  BDbR_table=readRDS(file =my_data)
45
46  Q[iT, im, 1]=apply(five_table,2,function (x) quantile(x,probs = 0.95))
    [4] #max
47  Q[iT, im, 2]=apply(five_table,2,function (x) quantile(x,probs = 0.95))
    [6] #min
48  Q[iT, im, 3]=quantile(DbR_table,probs = 0.95)
49  Q[iT, im, 4]=quantile(BDbR_table,probs = 0.95)
50
51
52  #powers for max, min, dbr, bdbr
53  h=c()
54  temp=matrix(NA,rept,2)
55  for (loop in 1:rept){
56    Fm=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Fm Gn1-1/",
57                     iT,m[im],n[im],loop, "Fm.rds",sep = "-") )
58    Gn=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Fm Gn1-1/",
59                     iT,m[im],n[im],loop, "Gn.rds",sep = "-"))
60    h=rbind(h,c(five_in_all(Fm,Gn,iT)[c(4,6)],
61                H_test(Fm,Gn,iT),
62                BDBR(Fm,Gn,iT)))
63    temp[loop,]=Permu(Fm,Gn,m[im], n[im], 25 ,iT)
64  }
65
66  temp_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Permu temp/Scale change/",
67                  iT,m[im],n[im], "PS.rds",sep = "-")
68  saveRDS(temp, file=temp_data)
69  h_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/Permu temp/Scale change/",
70               iT,m[im],n[im], "MaxDbr.rds",sep = "-")
71  saveRDS(h, file=h_data)
72
73
74  #powers for max, min, dbr, bdbr
75  for(j in 1:4){
76    powers[iT,im,j]=mean(h[,j]>=Q[iT,im,j])
77  }
78
79  p_quantile_sum=readRDS("C:/Users/ychen462/Desktop/Data depth new/two
    sample/Permutation/p-value-sum-samesize.rds")
80  p_quantile_prod=readRDS("C:/Users/ychen462/Desktop/Data depth new/two
    sample/Permutation/p-value-prod-samesize.rds")
```

```
81
82      #powers for prod, sum
83      powers[iT,im,5]=sum(temp[,1]< p_quantile_prod[im,iT])/rept
84
85      powers[iT,im,6]=sum(temp[,2]< p_quantile_sum[im,iT])/rept
86
87
88    }
89  }
90
91  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/new power1-1.rds",sep = "-")
92  saveRDS(powers, file = my_data)
93
94
95
96  ####n=m/2
97  set.seed(1)
98  #set m,n values, mu and sigma
99  m=c(1:10)*50
100 n=m/2
101 type.depth=c(1,2,3)
102 mu1 <- c(0,0)
103 mu2 <- c(0,0)
104 sigma1 <- matrix(c(1,0,0,1), nc = 2)
105 sigma2 <- matrix(c(1,0.5,0.5,1), nc = 2)
106
107
108
109 rept=1000 #number of repetitions
110
111 Q=array(NA,dim=c(length(type.depth),length(m),4))
112 powers=array(NA,dim=c(length(type.depth),length(m),6))
113
114 for (iT in type.depth){
115   for (im in 1:10){
116     for (loop in 1:rept){
117       Fm=mvrnorm(m[im],mu1, sigma1)
118       Gn=mvrnorm(n[im],mu2, sigma2)
119
120       Fm_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Fm Gn1-2/",
121                     iT,m[im],n[im],loop, "Fm.rds",sep = "-")
122       Gn_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Fm Gn1-2/",
123                     iT,m[im],n[im],loop, "Gn.rds",sep = "-")
124       saveRDS(Fm, file=Fm_data)
125       saveRDS(Gn, file=Gn_data)
126     }
127
128
129     #read data from null hypothesis
130     my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
        hypothesis/five_in_all/null",
131                   iT,m[im],n[im],".rds",sep = "-")
132     five_table=readRDS(file =my_data)
133
```

```
134     my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
        hypothesis/dbr/null",
135                   iT,m[im],n[im],"DbR.rds",sep = "-")
136     DbR_table=readRDS(file =my_data)
137
138     my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/null
        hypothesis/bdbr/null",
139                   iT,m[im],n[im],"BDbR.rds",sep = "-")
140     BDbR_table=readRDS(file =my_data)
141
142     Q[iT, im, 1]=apply(five_table,2,function (x) quantile(x,probs = 0.95))
        [4] #max
143     Q[iT, im, 2]=apply(five_table,2,function (x) quantile(x,probs = 0.95))
        [6] #min
144     Q[iT, im, 3]=quantile(DbR_table,probs = 0.95)
145     Q[iT, im, 4]=quantile(BDbR_table,probs = 0.95)
146
147
148     #powers for max, min, dbr, bdbr
149     h=c()
150     temp=matrix(NA,rept,2)
151     for (loop in 1:rept){
152       Fm=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Fm Gn1-2/",
153                        iT,m[im],n[im],loop, "Fm.rds",sep = "-") )
154       Gn=readRDS(paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Fm Gn1-2/",
155                        iT,m[im],n[im],loop, "Gn.rds",sep = "-"))
156       h=rbind(h,c(five_in_all(Fm,Gn,iT)[c(4,6)],
157                   H_test(Fm,Gn,iT),
158                   BDBR(Fm,Gn,iT)))
159       temp[loop,]=Permu(Fm,Gn,m[im], n[im], 25 ,iT)
160     }
161
162     temp_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Permu temp/Scale change/",
163                     iT,m[im],n[im], "PS.rds",sep = "-")
164     saveRDS(temp, file=temp_data)
165     h_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
        Permutation/Permu temp/Scale change/",
166                  iT,m[im],n[im], "MaxDbr.rds",sep = "-")
167     saveRDS(h, file=h_data)
168
169
170     #powers for max, min, dbr, bdbr
171     for(j in 1:4){
172       powers[iT,im,j]=mean(h[,j]>=Q[iT,im,j])
173     }
174
175
176     p_quantile_sum=readRDS("C:/Users/ychen462/Desktop/Data depth new/two
        sample/Permutation/p-value-sum-diffsize.rds")
177     p_quantile_prod=readRDS("C:/Users/ychen462/Desktop/Data depth new/two
        sample/Permutation/p-value-prod-diffsize.rds")
178
179     #powers for prod, sum
180     powers[iT,im,5]=sum(temp[,1]< p_quantile_prod[im,iT])/rept
181     powers[iT,im,6]=sum(temp[,2]< p_quantile_sum[im,iT])/rept
```

```
182
183
184
185   }
186 }
187
188 my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
      Permutation/new power1-2.rds",sep = "-")
189 saveRDS(powers, file = my_data)
190
191
192
193
194
195
196
197 ####################plot
198 add_legend <- function(...) {
199   opar <- par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0),
200               mar=c(0, 0, 0, 0), new=TRUE)
201   on.exit(par(opar))
202   plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
203   legend(...)
204 }
205 m=n=c(1:10)*50
206 my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
      Permutation/new power1-1.rds",sep = "-")
207 six_table=readRDS(file =my_data)
208 a=0.06
209
210 plotchar=c(8,18,4,5,3,12,6,7)
211 COL=c( "black", "red", 'darkgreen','orange', "blue", "purple","brown")
212 par(mfrow=c(3,2) )
213 par(mar=c(4,4,4,0)+0.1,fig=c(0,0.5,2/3-a,1) )
214
215 yrange=c(0,1)
216 plot(m,six_table[1,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
      =2,
217      #main="Log of Normalizing Constant Approximation",
218      ylab="Power (Mahalanobis)",
219      xlab="m (n=m)") #max
220
221 lines(m,six_table[1,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
222 lines(m,six_table[1,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2) #dbr
223 lines(m,six_table[1,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2) #bdbr
224 lines(m,six_table[1,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2) #prod
225 lines(m,six_table[1,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2) #sum
226
227 par(mar=c(4,4,0,0)+0.1,fig=c(0,0.5,1/3-a/2,2/3-a) , new=TRUE)
228
229 plot(m,six_table[2,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
      =2,
230      #main="Log of Normalizing Constant Approximation",
231      ylab="Power (Spatial)",
232      xlab="m (n=m)") #max
233
234
235 lines(m,six_table[2,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
```

```
lines(m,six_table[2,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2)
lines(m,six_table[2,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2)
lines(m,six_table[2,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2)
lines(m,six_table[2,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2)


par(mar=c(4,4,0,0)+0.1,fig=c(0,0.5,0,1/3-a/2) , new=TRUE)


plot(m,six_table[3,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
    =2,
     #main="Log of Normalizing Constant Approximation",
     ylab="Power (Projection)",
     xlab="m (n=m)")

lines(m,six_table[3,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
lines(m,six_table[3,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2)
lines(m,six_table[3,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2)
lines(m,six_table[3,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2)
lines(m,six_table[3,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2)

#####
m=c(1:10)*50
my_data=paste("C:/Users/ychen462/Desktop/Data depth new/two sample/
    Permutation/new power1-2.rds",sep = "-")
six_table=readRDS(file =my_data)


par(mar=c(4,4,4,0)+0.1,fig=c(0.5,1,2/3-a,1) , new=TRUE)

yrange=c(0,1)
plot(m,six_table[1,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
    =2,
     #main="Log of Normalizing Constant Approximation",
     ylab="Power (Mahalanobis)",
     xlab="m (n=m/2)") #max

lines(m,six_table[1,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
lines(m,six_table[1,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2) #dbr
lines(m,six_table[1,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2) #bdbr
lines(m,six_table[1,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2) #prod
lines(m,six_table[1,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2) #sum

par(mar=c(4,4,0,0)+0.1,fig=c(0.5,1,1/3-a/2,2/3-a) , new=TRUE)

plot(m,six_table[2,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
    =2,
     #main="Log of Normalizing Constant Approximation",
     ylab="Power (Spatial)",
     xlab="m (n=m/2)") #max


lines(m,six_table[2,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
lines(m,six_table[2,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2)
lines(m,six_table[2,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2)
lines(m,six_table[2,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2)
lines(m,six_table[2,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2)

```

```
290
291 par(mar=c(4,4,0,0)+0.1,fig=c(0.5,1,0,1/3-a/2) , new=TRUE)
292
293
294 plot(m,six_table[3,,1],ylim=yrange,type='b',pch=plotchar[1],col=COL[1],lwd
      =2,
295      #main="Log of Normalizing Constant Approximation",
296      ylab="Power (Projection)",
297      xlab="m (n=m/2)")
298
299 lines(m,six_table[3,,2],type='b',pch=plotchar[2],col=COL[2],lwd=2) #min
300 lines(m,six_table[3,,3],type='b',pch=plotchar[3],col=COL[3],lwd=2)
301 lines(m,six_table[3,,4],type='b',pch=plotchar[4],col=COL[4],lwd=2)
302 lines(m,six_table[3,,5],type='b',pch=plotchar[5],col=COL[5],lwd=2)
303 lines(m,six_table[3,,6],type='b',pch=plotchar[6],col=COL[6],lwd=2)
304
305
306
307
308
309
310 add_legend("top",
311            legend=c(expression(M["m,n"]),
312                     expression(M["m,n"]^'*'),"DbR", 'BDbR',expression(P["m,n
      "]),
313                     expression(S["m,n"])),
314            pch=plotchar[c(1:6)],col=COL[c(1:6)],bty='n', horiz=TRUE,lwd=2)
```

Mean change

```
1 mu1 <- c(0,0)
2 mu2 <- c(0.3,0.3)
3 sigma1 <- matrix(c(1,0,0,1), nc = 2)
4 sigma2 <- matrix(c(1,0,0,1), nc = 2)
```

Both change

```
1 mu1 <- c(0,0)
2 mu2 <- c(0.2,0.2)
3 sigma1 <- matrix(c(1,0,0,1), nc = 2)
4 sigma2 <- matrix(c(1,0.4,0.4,1), nc = 2)
```

A.3 Chapter 4

A.3.1 Ramen spectrum

```
1 name="xxx/Data depth/785_830_100%
      _30s_Renishaw_1070centre_15stepsize_8x6_Slice1_Region3_KirstySlice_KirstyZfirst
      .txt"
2
3 data <- read.delim(name)
```

```
4
5 wave=data[,4]
6 intens=data[,5]
7
8 plot(wave)
9 plot(intens)
10
11 table(wave)
12 N=48
13 wave1=as.numeric(names(table(wave)))
14 T=length(wave1)
15 wave1=wave1[T:1]
16 samples=NULL
17 for(i in 1:48)
18   samples=rbind(samples,intens[((i-1)*T+1):(i*T)])
19
20 samples=samples[-c(17,18),]
21
22 x=c(1:length(wave1[219:245]))
23 x0=14
24 newx=(x-x0)^2/(length(x))^2
25 ind1=ind2=NULL
26 Res=matrix(NA, 46,27)
27 for(i in 1:46){
28   #Sys.sleep(1)
29   y=log(samples[i,219:245])
30   fit=lm(y~newx)
31   Res[i,]=summary(fit)$res
32   rs=summary(fit)$r.squared
33   if(rs>0.5) {plot(wave1[219:245],y,ylab=i,xlab="Group 2",type="l");ind2=c(
       ind2,i)}
34   else {plot(wave1[219:245],y,xlab="Group 1",ylab=i,type="l");ind1=c(ind1,i)
       }
35   abline(v=1524,col="red")
36 }
37
38 Fm=samples[ind1,219:245]
39 Gn=samples[ind2,219:245]
40
41 #plot
42 add_legend <- function(...) {
43   opar <- par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0),
44               mar=c(0, 0, 0, 0), new=TRUE)
45   on.exit(par(opar))
46   plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
47   legend(...)
48 }
49
50 xx=wave1[219:245]
51
52 par(mfrow=c(4,2) )
53
54 a=0.01;b=(1-3*a)/2;c0=0.017
55
56 par(mar=c(-0.1,4,0.2,0)+0.1,fig=c(0+a,0+a+b,3/4+c0,1) )
57
58 plot(xx,Fm[1,],type="l",ylab="Counts",xlab="",xaxt = 'n')
59 abline(v=1523.71,lty=2)
```

```
60
61 par(mar=c(-0.1,4,0.2,0)+0.1,fig=c(0+2*a+b,0+2*a+2*b,3/4+c0,1), new=TRUE)
62 plot(xx,Gn[1,],type="l",ylab="Counts",xlab="",xaxt = 'n')
63 abline(v=1523.71,lty=2)
64
65 par(mar=c(-0.1,4,-0.1,0)+0.1,fig=c(0+a,0+a+b,2/4+2*c0,3/4+c0), new=TRUE)
66 plot(xx,Fm[2,],type="l",ylab="Counts",xlab="",xaxt = 'n')
67 abline(v=1523.71,lty=2)
68
69 par(mar=c(-0.1,4,-0.1,0)+0.1,fig=c(0+2*a+b,0+2*a+2*b,2/4+2*c0,3/4+c0), new=
       TRUE)
70 plot(xx,Gn[2,],type="l",ylab="Counts",xlab="",xaxt = 'n')
71 abline(v=1523.71,lty=2)
72
73 par(mar=c(-0.1,4,-0.1,0)+0.1,fig=c(0+a,0+a+b,1/4+3*c0,2/4+2*c0), new=TRUE)
74 plot(xx,Fm[3,],type="l",ylab="Counts",xlab="",xaxt = 'n')
75 abline(v=1523.71,lty=2)
76
77 par(mar=c(-0.1,4,-0.1,0)+0.1,fig=c(0+2*a+b,0+2*a+2*b,1/4+3*c0,2/4+2*c0), new
       =TRUE)
78 plot(xx,Gn[3,],type="l",ylab="Counts",xlab="",xaxt = 'n')
79 abline(v=1523.71,lty=2)
80
81 par(mar=c(4,4,-0.1,0)+0.1,fig=c(0+a,0+a+b,0,1/4+3*c0), new=TRUE)
82 plot(xx,Fm[4,],type="l",ylab="Counts",xlab=expression("cm"^"-1") )
83 abline(v=1523.71,lty=2)
84
85 par(mar=c(4,4,-0.1,0)+0.1,fig=c(0+2*a+b,0+2*a+2*b,0,1/4+3*c0), new=TRUE)
86 plot(xx,Gn[4,],type="l",ylab="Counts",xlab=expression("cm"^"-1") )
87 abline(v=1523.71,lty=2)
88
89
90
91 #27M
92 #p-value: prod, sum
93 size=2
94 Fm=samples[ind1,219:245]
95 Gn=samples[ind2,219:245]
96 Permu(Fm,Gn,m,size,1,1000) #NA
97 Permu(Fm,Gn,m,size,2,1000) #NA
98 Permu(Fm,Gn,m,size,3,1000) #0.006 0.006
99 #take log
100 Fm=log(samples[ind1,219:245])
101 Gn=log(samples[ind2,219:245])
102 Permu(Fm,Gn,m,size,1,1000) #NA
103 Permu(Fm,Gn,m,size,2,1000) #NA
104 Permu(Fm,Gn,m,size,3,1000) #0 0
105
106 #p-value: max,min,dbr,bdbr
107 Fm=samples[ind1,219:245]
108 Gn=samples[ind2,219:245]
109 all_data=samples[,219:245]
110
111 mean_Fm=colMeans(Fm)
112 v_Fm=cov(Fm)
113 mean_Gn=colMeans(Gn)
114 v_Gn=cov(Gn)
115 mean_all=colMeans(all_data)
```

```
v_all=cov(all_data)

#max min
M1=five_in_all(Fm,Gn,1)[c(4,6)] #NA
M2=five_in_all(Fm,Gn,2)[c(4,6)] #NA
M3=five_in_all(Fm,Gn,3)[c(4,6)] #10.167984  3.188728
#dbr
H1=H_test(Fm,Gn,1) #NA
H2=H_test(Fm,Gn,2) #NA
H3=H_test(Fm,Gn,3) #6.190163
#bdbr
B1=BDBR(Fm,Gn,1) #NA
B2=BDBR(Fm,Gn,2) #NA
B3=BDBR(Fm,Gn,3) #8.645859

#Simulation
set.seed(123)
Type.Depth=c(3)
n=1000
for(iT in 1:length(Type.Depth)){
  MV=c()
  for (i in 1:n) {
    Fm <- rbind(mvrnorm(35, mean_all, v_all))
    Gn <- rbind(mvrnorm(11, mean_all, v_all))
    MV=rbind(MV,c(five_in_all(Fm,Gn,Type.Depth[iT])[c(4,6)] ,
                  H_test(Fm,Gn,Type.Depth[iT]),
                  BDBR(Fm,Gn,Type.Depth[iT])))
  }
  # Save an object to a file
  my_data=paste("xxx/Data depth/spectra/Data/",
                Type.Depth[iT],27,".rds",sep = "-")
  saveRDS(MV, file = my_data)
}

#projection
my_data=paste("xxx/Data depth/spectra/Data/",
              3,27,".rds",sep = "-")
Q1=readRDS(file = my_data)
mean(Q1[,1]>=M3[1]) #0.008
mean(Q1[,2]>=M3[2]) #0.008
mean(Q1[,3]>=H3) #0.006
mean(Q1[,4]>=B3) #0.004


#15M
Fm=samples[ind1,225:239]
Gn=samples[ind2,225:239]


#5L
Fm=samples[ind1,c(228,229,230,231,232)]
Gn=samples[ind2,c(228,229,230,231,232)]

#5R
Fm=samples[ind1,232:236]
Gn=samples[ind2,232:236]


```

```
174
175 #Scale curve
176 Fm=samples[ind1,c(228,229,230,231,232)]
177 Gn=samples[ind2,c(228,229,230,231,232)]
178
179 Fm=log(samples[ind1,c(228,229,230,231,232)])
180 Gn=log(samples[ind2,c(228,229,230,231,232)])
181
182 s1<-scaleCurve(Fm, depth_params = list(method = "Mahalanobis"), name = "
        Group 1",title="")
183 s2<-scaleCurve(Gn, depth_params = list(method = "Mahalanobis"), name = "
        Group 2")
184 sc_list <- combineDepthCurves(.list=list(s1, s2))
185 plot(sc_list)
186
187 s1<-scaleCurve(Fm, depth_params = list(method = "Projection"), name = "Group
         1",title="")
188 s2<-scaleCurve(Gn, depth_params = list(method = "Projection"), name = "Group
         2")
189 sc_list <- combineDepthCurves(.list=list(s1, s2))
190 plot(sc_list)
```

A.3.2 Sloan Digital Sky Survey Data

```
 1 library(ddalpha)
 2 library(MASS)
 3 library(matrixStats)
 4 library(openxlsx)
 5 library(mvtnorm)
 6 library(palmerpenguins)
 7 library(DepthProc)
 8
 9 library(astrodatR)
10 data("SDSS_ptsrc_train")
11 df=SDSS_ptsrc_train
12
13 my_tab=table(df[,5])
14 df_name=names(my_tab)
15 df_class1<-which(df[,5]==df_name[1])
16 df_class2<-which(df[,5]==df_name[2])
17 df_class3<-which(df[,5]==df_name[3])
18
19 df_class1_num<-df[df_class1,1:4]
20
21 df_class2_num<-df[df_class2,1:4]
22
23 df_class3_num<-df[df_class3,1:4]
24
25 length(df_class1_num[,1])#2000
26 length(df_class2_num[,1])#5000
27 length(df_class3_num[,1])#2000
28
29 scaleCurve1=function (x, y = NULL, alpha = seq(0, 1, 0.01), name = "X",
       name_y = "Y",
30           title = "Scale Curve", depth_params = list(method = "Projection"))
31 {
```

```
  x <- na.omit(x)
  if (is.data.frame(x)) {
    x <- as.matrix(x)
  }
  if (!is.matrix(x)) {
    stop("x must be a matrix or data frame!")
  }
  if (!is.null(y)) {
    if (is.data.frame(y)) {
      y <- as.matrix(y)
    }
    if (!is.matrix(y)) {
      stop("y must be a matrix or data frame!")
    }
  }
  dim_x <- dim(x)[2]
  uxname_list <- list(u = x, X = x)
  depth_est <- do.call(depth, c(uxname_list, depth_params))
  k <- length(alpha)
  vol <- 1:k
  alpha_border <- ecdf(depth_est)(depth_est)
  for (i in 1:k) {
    tmp_x <- x[alpha_border >= alpha[i], ]
    np <- nrow(unique(as.matrix(tmp_x)))
    if (np > dim_x) {
      vol[i] <- convhulln(tmp_x, options = "FA")$vol
    }
    else {
      vol[i] <- 0
    }
  }
  vol=log(1+vol)
  scale_curve <- new("ScaleCurve", rev(vol), alpha = alpha,
                     depth = depth_est, name = name, title = title)
  if (!is.null(y)) {
    name <- name_y
    sc_tmp <- scaleCurve(x = y, y = NULL, alpha = alpha,
                         name = name, name_y = "Y", depth_params =
    depth_params)
    scale_curve <- combineDepthCurves(scale_curve, sc_tmp)
  }
  return(scale_curve)
}

#Mahalanobis ScaleCurve
s1<-scaleCurve1(df_class1_num,depth_params = list(method = "Mahalanobis"),
    name="Class1",title="",name_y ='log(1+Volume)')
s2<-scaleCurve1(df_class2_num,depth_params = list(method = "Mahalanobis"),
    name = "Class2")
s3<-scaleCurve1(df_class3_num, depth_params = list(method = "Mahalanobis"),
    name = "Class3")

sc_list <- combineDepthCurves(.list=list(s1, s2))
plot(sc_list)

sc_list <- combineDepthCurves(.list=list(s1, s3))
plot(sc_list)

```

```
sc_list <- combineDepthCurves(.list=list(s2, s3))
plot(sc_list)

#functions
percentile<-function(x,v){
  g=0
  for (i in v){
    if (i>=x){
      g=g+1
    }
  }
  return(g/length(v))
}

#class 1, class 2
M1=five_in_all(df_class1_num,df_class2_num,1)[c(4,6)] #4078.73497   63.86497
M2=five_in_all(df_class1_num,df_class2_num,2)[c(4,6)] #4123.11283   64.21147
M3=five_in_all(df_class1_num,df_class2_num,3)[c(4,6)] #3847.57551   62.02883

H1=H_test(df_class1_num,df_class2_num,1) #2186.856
H2=H_test(df_class1_num,df_class2_num,2) #2275.329
H3=H_test(df_class1_num,df_class2_num,3) #3489.914

B1=BDBR(df_class1_num,df_class2_num,1) #8148.935
B2=BDBR(df_class1_num,df_class2_num,2) #8580.996
B3=BDBR(df_class1_num,df_class2_num,3) #8021.614


m=colMeans(rbind(df_class1_num,df_class2_num))
v=cov(rbind(df_class1_num,df_class2_num))

#simulated data
set.seed(123)
MV=c()
Type.Depth=c(1,2,3)
n=1000
for(iT in 1:length(Type.Depth)){
  for (i in 1:n) {
    Fm <- rbind(mvrnorm(2000, m ,v))
    Gn <- rbind(mvrnorm(5000, m ,v))
    MV=rbind(MV,c(five_in_all(Fm,Gn,Type.Depth[iT])[c(4,6)],
                  H_test(Fm,Gn,Type.Depth[iT]),
                  BDBR(Fm,Gn,Type.Depth[iT])))
  }
  # Save an object to a file
  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
                Type.Depth[iT],2000,5000,".rds",sep = "-")
  saveRDS(MV, file = my_data)
}



#maha
my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              1,2000,5000,".rds",sep = "-")
Q1=readRDS(file = my_data)
```

```
#simulated quant
quantile(Q1[,1],0.95) #max
quantile(Q1[,2],0.95) #min
quantile(Q1[,3],0.95) #dbr
quantile(Q1[,4],0.95) #bdbr
#percentile
percentile(M1[1],Q1[,1]) #0
percentile(M1[2],Q1[,2]) #0
percentile(H1,Q1[,3]) #0
percentile(B1,Q1[,4]) #0


#Spatial
my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              2,2000,5000,".rds",sep = "-")
Q2=readRDS(file = my_data)
#percentile
percentile(M2[1],Q2[,1]) #0
percentile(M2[2],Q2[,2]) #0
percentile(H2,Q2[,3]) #0
percentile(B2,Q2[,4]) #0


#Projection
my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              3,2000,5000,".rds",sep = "-")
Q3=readRDS(file = my_data)
#percentile
percentile(M3[1],Q3[,1]) #0
percentile(M3[2],Q3[,2]) #0
percentile(H3,Q3[,3]) #0
percentile(B3,Q3[,4]) #0

####p-value prod sum
Permu=function(Fm,Gn,m,n,size,type.depth){
  T_star=five_in_all(Fm, Gn,type.depth)
  B=(m+n)/size
  Loop=200
  T_b=matrix(NA,Loop,8)
  pvalue=c()
  for(loop in 1:Loop){

    ind=sample( B, m/size)

    ind=sort(ind)

    Fm.ind=NULL
    for(i in 1:length(ind))
      Fm.ind=c(Fm.ind, ((ind[i]-1)*size+1): (ind[i]*size) )

    Fm.b=rbind(Fm,Gn)[Fm.ind,]
    Gn.b=rbind(Fm,Gn)[-Fm.ind,]
    T_b[loop,]=five_in_all(Fm.b, Gn.b,type.depth)
  }

  # pvalue=sum(T_b>T_star)/Loop
```

```
  pvalue[1]=sum(T_b[,7]>T_star[7])/Loop #prod
  pvalue[2]=sum(T_b[,8]>T_star[8])/Loop #sum

  return(pvalue)
}
size=100
Permu(df_class1_num,df_class2_num,2000,5000,size,1) #0 0
Permu(df_class1_num,df_class2_num,2000,5000,size,2) #0 0
Permu(df_class1_num,df_class2_num,2000,5000,size,3) #0 0

##asymptotical p value
library(cubature)

c12=2000^(-1/2)*(2000^(-1)+5000^(-1))^(-1/2)
c12t=5000^(-1/2)*(2000^(-1)+5000^(-1))^(-1/2)

#max
x0=M1[1]
#M2[1], M3[1]
lower <- rep(-Inf,2)
upper <- rep(Inf,2)

# First implementation (modified)
fxyz <- function(w) {
  x <- w[1]
  y <- w[2]
  (2*pi)^(-2/2)*exp(-(x^2+y^2)/2)*as.numeric((c12*x+c12t*y)^2 <=x0)
}

adaptIntegrate(f=fxyz,lowerLimit=lower,upperLimit=upper,doChecking=TRUE,
               maxEval=2000000,absError=10e-10,tol=1e-10)
1-1

#min
x0=M1[2]
lower <- rep(-Inf,2)
upper <- rep(Inf,2)

# First implementation (modified)
fxyz <- function(w) {
  x <- w[1]
  y <- w[2]
  (2*pi)^(-2/2)*exp(-(x^2+y^2)/2)*as.numeric((c12*x+c12t*y) <=x0) *as.
    numeric((c12*x+c12t*y) >=-x0)
}

adaptIntegrate(f=fxyz,lowerLimit=lower,upperLimit=upper,doChecking=TRUE,
               maxEval=2000000,absError=10e-10,tol=1e-10)
1-1
```

A.3.3 Skull Data

```
library(HSAUR)
data("skulls", package = "HSAUR")
levels(skulls$epoch)
skull=scale(skulls[c(1:150),2:5])
```

```
l1=skull[c(1:30),]
l2=skull[c(31:60),]
l3=skull[c(61:90),]
l4=skull[c(91:120),]
l5=skull[c(121:150),]



s1<-scaleCurve(skulls[c(1:30),2:5], depth_params = list(method = "
    Mahalanobis"),name="4000 B.C.",title="")
s2<-scaleCurve(skulls[c(31:60),2:5], depth_params = list(method = "
    Mahalanobis"), name = "3300 B.C.",title="")
s3<-scaleCurve(skulls[c(61:90),2:5], depth_params = list(method = "
    Mahalanobis"), name = "1850 B.C.",title="")
s4<-scaleCurve(skulls[c(91:120),2:5], depth_params = list(method = "
    Mahalanobis"), name = "200 B.C.",title="")
s5<-scaleCurve(skulls[c(121:150),2:5], depth_params = list(method = "
    Mahalanobis"), name = "150 A.D.",title="")


###1850 200
sc_list <- combineDepthCurves(.list=list(s3, s4))
plot(sc_list)

###150 3300
sc_list <- combineDepthCurves(.list=list(s2,s5))
plot(sc_list)

###150 200
sc_list <- combineDepthCurves(.list=list(s4,s5))
plot(sc_list)



###1850 200
m=colMeans(skull[c(61:120),])
v=cov(skull[c(61:120),])

M1=five_in_all(l3,l4,1)[c(4,6)] #3.813556 1.952833
M2=five_in_all(l3,l4,2)[c(4,6)] #4.293556 2.072090
M3=five_in_all(l3,l4,3)[c(4,6)] #2.357556 1.535433

H1=H_test(l3,l4,1) #1.961202
H2=H_test(l3,l4,2) #2.304372
H3=H_test(l3,l4,3) #2.665464

B1=BDBR(l3,l4,1) #2.189558
B2=BDBR(l3,l4,2) #2.492129
B3=BDBR(l3,l4,3) #2.515559



#simulated data
set.seed(123)
Type.Depth=c(1,2,3)
n=1000
for(iT in 1:length(Type.Depth)){
  MV=c()
```

```
  for (i in 1:n) {
    Fm <- rbind(mvrnorm(30, m, v))
    Gn <- rbind(mvrnorm(30, m, v))
    MV=rbind(MV,c(five_in_all(Fm,Gn,Type.Depth[iT])[c(4,6)],
                  H_test(Fm,Gn,Type.Depth[iT]),
                  BDBR(Fm,Gn,Type.Depth[iT])))
  }
  # Save an object to a file
  my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
                Type.Depth[iT],"3-4.rds",sep = "-")
  saveRDS(MV, file = my_data)
}



my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              1,"3-4.rds",sep = "-")
Q1=readRDS(file = my_data)
mean(Q1[,1]>=M1[1]) #0.312
mean(Q1[,2]>=M1[2]) #0.312
mean(Q1[,3]>=H1) #0.351
mean(Q1[,4]>=B1) #0.379


my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              2,"3-4.rds",sep = "-")
Q2=readRDS(file = my_data)
mean(Q2[,1]>=M2[1]) #0.396
mean(Q2[,2]>=M2[2]) #0.396
mean(Q2[,3]>=H2) #0.411
mean(Q2[,4]>=B2) #0.447


my_data=paste("C:/Users/ychen462/Desktop/Data depth new/data analysis/Data
    /",
              3,"3-4.rds",sep = "-")
Q3=readRDS(file = my_data)
mean(Q3[,1]>=M3[1]) #0.369
mean(Q3[,2]>=M3[2]) #0.365
mean(Q3[,3]>=H3) #0.145
mean(Q3[,4]>=B3) #0.212


size=5 #s=5,c=1000
Permu(l3,l4,30,30,size,1) #0.177 0.127
Permu(l3,l4,30,30,size,2) #0.249 0.180
Permu(l3,l4,30,30,size,3) #0.163 0.137


##asymptotical p value
library(cubature)

c12=30^(-1/2)*(30^(-1)+30^(-1))^(-1/2)
c12t=30^(-1/2)*(30^(-1)+30^(-1))^(-1/2)

```

```
#max
x0=M1[1]
#x0=M2[1], x0=M3[1]
lower <- rep(-Inf,2)
upper <- rep(Inf,2)

# First implementation (modified)
fxyz <- function(w) {
  x <- w[1]
  y <- w[2]
  (2*pi)^(-2/2)*exp(-(x^2+y^2)/2)*as.numeric((c12*x+c12t*y)^2 <=x0)
}

adaptIntegrate(f=fxyz,lowerLimit=lower,upperLimit=upper,doChecking=TRUE,
               maxEval=2000000,absError=10e-10,tol=1e-10)
1-0.9491544
0.0508456 #maha
1-0.9617267
0.0382733 #spatial
1-0.8753175
0.1246825 #proj


#min
x0=M1[2]
#x0=M2[2], x0=M3[2]
lower <- rep(-Inf,2)
upper <- rep(Inf,2)

# First implementation (modified)
fxyz <- function(w) {
  x <- w[1]
  y <- w[2]
  (2*pi)^(-2/2)*exp(-(x^2+y^2)/2)*as.numeric((c12*x+c12t*y) <=x0) *as.
    numeric((c12*x+c12t*y) >=-x0)
}

adaptIntegrate(f=fxyz,lowerLimit=lower,upperLimit=upper,doChecking=TRUE,
               maxEval=2000000,absError=10e-10,tol=1e-10)
1-0.9491544
0.0508456 #maha
1-0.9617267
0.0382733 #spatial
1-0.8753175
0.1246825 #proj
```