

## Research Article

# Computing Simplicial Depth by Using Importance Sampling Algorithm and Its Application

Fanyu Meng,<sup>1</sup> Wei Shao ,<sup>2,3</sup> and Yuxia Su<sup>3</sup>

<sup>1</sup>School of Mathematics, Shandong University, Jinan 250100, China

<sup>2</sup>School of Management, Qufu Normal University, Rizhao 276826, China

<sup>3</sup>School of Statistics, Qufu Normal University, Qufu 273165, China

Correspondence should be addressed to Wei Shao; [wshao1031@gmail.com](mailto:wshao1031@gmail.com)

Received 30 December 2020; Revised 6 September 2021; Accepted 18 October 2021; Published 31 December 2021

Academic Editor: Harish Garg

Copyright © 2021 Fanyu Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Simplicial depth (SD) plays an important role in discriminant analysis, hypothesis testing, machine learning, and engineering computations. However, the computation of simplicial depth is hugely challenging because the exact algorithm is an NP problem with dimension  $d$  and sample size  $n$  as input arguments. The approximate algorithm for simplicial depth computation has extremely low efficiency, especially in high-dimensional cases. In this study, we design an importance sampling algorithm for the computation of simplicial depth. As an advanced Monte Carlo method, the proposed algorithm outperforms other approximate and exact algorithms in accuracy and efficiency, as shown by simulated and real data experiments. Furthermore, we illustrate the robustness of simplicial depth in regression analysis through a concrete physical data experiment.

## 1. Introduction

With the development of computer technology and multivariate statistical analysis, scientists deal with a large amount of multidimensional data in many fields, such as biogenetics and industrial engineering. The demand for multivariate data analysis tools has become increasingly urgent. As a powerful multivariate nonparametric and robust statistical tool, the statistical depth function extends the concept of one-dimensional data order statistics and provides the central-outward sorting of multivariate data [1–4]. In recent years, the interest of researchers in statistical depth has increased due to the extensive application of the statistical depth function in multivariate statistical analysis, robust estimation, discriminant analysis, hypothesis testing, machine learning, economics, and hydrological data analysis [5, 6].

The first statistical depth function concept, which was proposed by Tukey in 1975, is known as the halfspace depth (also known as the Tukey depth) [7–9]. The other concepts of the statistical depth function include projection depth [3, 10], simplicial depth (SD) [11, 12], and regression depth

[13, 14]. Zuo and Serfling defined a general structural property of the statistical depth function [1]. Among the many concepts of this statistical depth function, SD is a relatively attractive one not only because of its simple form and ability to achieve the maximum depth value in the center and satisfy monotonicity but also because of its important applications in sign test and centralization test [1, 12].

However, the computation of SD is complicated. The exact calculation of SD is an NP problem, which is only feasible when the dimension is no higher than three. Serfling and Wang emphasized that the computation of SD for higher-dimensional data still requires further study [12]. The computation and application of the statistical depth function are active research topics.

Similarly, Monte Carlo (MC) methods have become important statistical, computational tools that are widely used in finance, engineering computation, genetic biology, computational chemistry, and other related fields [15–18]. As a critical MC strategy, the importance sampling (IS) method concentrates most of the test samples in the important area of the objective function by introducing the transfer probability density function [15, 19]. This method

dramatically improves the computational efficiency and is an important MC acceleration algorithm. In this study, we apply an efficient IS algorithm to the approximate computation of SD and demonstrate the advantages of such an algorithm over other MC methods and exact algorithms through simulated and real data examples. Furthermore, we extend the SD to regression analysis and obtain a robust estimation of regression analysis. The results of a real physical data experiment show that the estimation based on the SD method is more robust than that based on the traditional least squares (LS) method.

The remainder of this paper is organized as follows. In Section 2, we review the preliminary concept and existing algorithms for SD. Section 3 describes the IS algorithm used for the computation of SD. The advantages of the IS algorithm are illustrated through simulated data examples in Section 4. The extension of SD to regression analysis and a real data experiment are presented in Section 5. Lastly, the conclusions are provided in Section 6.

## 2. Preliminary of SD and the State of the Art

In this section, we present the preliminary of SD and the existing algorithms for its computation.

Consider a sample set  $\mathbf{X}^n = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$  in  $\mathbb{R}^d$ , ( $d \geq 2$ ), where  $\mathbf{X}^n$  is one sample of size  $n$  in  $\mathbb{R}^d$  and  $\mathbf{x}$  is a given point in  $\mathbb{R}^d$ . The sample version [11] of SD of  $\mathbf{x}$  with respect to the sample set  $\mathbf{X}^n$  is expressed as

$$\text{SD}(\mathbf{x}, \mathbf{X}^n) = \frac{1}{C_n^{d+1}} \sum_{1 \leq i_1 < \dots < i_{d+1} \leq n} \mathbf{1}_{\{\mathbf{x} \in S[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d+1}}]\}}, \quad (1)$$

where  $\mathbf{1}_{\{A\}}$  denotes the indicator function of event  $A$ , and  $S[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d+1}}]$  denotes the simplex determined by the  $d+1$  sample points  $\{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d+1}}\}$ .

Serfling and Wang stated that no algorithms are faster than simply generating all simplices and counting the ones enclosing the given point (using  $O(n^{d+1})$  complex time) when dimension  $d \geq 5$  [12]. Therefore, designing an efficient approximate algorithm for the computation of SD is necessary.

A direct MC method for the computation of SD contains two steps: (1) randomly selecting  $d+1$  points from  $\mathbf{X}^n$  and then (2) taking the average of the points that enclose the given point  $\mathbf{x}$  (i.e., using  $\widehat{\text{SD}}(\mathbf{x}, \mathbf{X}^n)$  to estimate the true SD value  $\text{SD}(\mathbf{x}, \mathbf{X}^n)$ ).

$$\widehat{\text{SD}}(\mathbf{x}, \mathbf{X}^n) = \frac{1}{M} \sum_{i=1}^M \mathbf{1}_{\{\mathbf{x} \in S[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d+1}}]\}}, \quad (2)$$

where  $\{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d+1}}\}$  is randomly chosen from  $\mathbf{X}^n$  and  $M$  is the trying number for the estimation.

Another approach for the computation of SD is the use of the IS algorithm, which is the proposed method in this study. The computation of SD is an expectation computation. Therefore, SD can be estimated by the IS algorithm. The simple MC method uses the randomly selected  $d+1$  points to estimate the SD, whereas the IS approach selects  $d+1$  points with a high probability that they contain the given point  $\mathbf{x}$ . Theoretically, the results of the latter will have a

smaller variance than those of the former. The simulated data examples in Section 4 illustrate the advantage of the IS algorithm over the MC method.

## 3. New Algorithm for SD in $\mathbb{R}^d$

**3.1. Overview of the IS Algorithm.** Many engineering problems can be expressed as computations of a multidimensional integral. Using the MC method to compute the integral involves drawing samples from a uniform distribution on a regular area and using the sample mean to approximate the true integral. In higher-dimensional cases, the efficiency of the MC method is extremely low if the region where the target function is not equal to zero is extraordinarily sparse. On the contrary, the IS algorithm draws most samples in the important area. This strategy improves the efficiency of the integral computation. The IS algorithm plays an important role in the field of statistical physics, molecular simulation, and Bayesian statistics.

For example, we want to compute the integral of  $h(\mathbf{x})$  on region  $A$ ; that is,

$$\mu = \int_A h(\mathbf{x}) d\mathbf{x}, \quad (3)$$

and the integral computation (3) can be treated as an expectation calculation:

$$\mu = E_\pi \left( \frac{h(\mathbf{X})}{\pi(\mathbf{X})} \right), \quad (4)$$

where  $\mathbf{X}$  is a random variable (r.v.) with its own probability density function (p.d.f.)  $\pi(\mathbf{x})$ ; that is,  $\mathbf{X} \sim \pi(\mathbf{x})$ . If  $\mathbf{X}_1, \dots, \mathbf{X}_n$  denote samples with size  $n$  from  $\mathbf{X}$ , the MC method draws  $\mathbf{X}$  from a uniform distribution on region  $A$ . From the Law of Large Numbers [20], the sample mean can be used to estimate the expectation in (4) as

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n h(\mathbf{X}_i) \cdot S(A), \quad (5)$$

where  $S(A)$  is the area of  $A$  and  $\mathbf{X}$  is the r.v. from the uniform distribution on  $A$  ( $\mathbf{X} \sim U(\mathbf{x})$ ).

However, the efficiency of the MC method (5) will be extremely low if region  $A$  is extremely wide or sparse (especially in high-dimensional cases). By contrast, the IS method uses a special p.d.f.  $g(\mathbf{x})$  instead of  $\pi(\mathbf{x})$  in (4) to compute mean  $\mu$  and utilizes the corresponding sample mean to estimate the expectation in (4):

$$\tilde{\mu} = \frac{1}{n} \sum_{i=1}^n \frac{h(\mathbf{X}_i)}{g(\mathbf{X}_i)}, \quad (6)$$

and the variance  $\text{Var}(\tilde{\mu}) = 1/n \text{Var}(h(\mathbf{X}_i)/g(\mathbf{X}_i))$ , which means that we can choose appropriate  $g(\mathbf{x})$  close to  $h(\mathbf{x})$  to reduce the variance of  $\tilde{\mu}$ . In extreme situations, if we select  $g(\mathbf{x}) \propto h(\mathbf{x})$ , that is,  $g(\mathbf{x}) = c \cdot h(\mathbf{x})$  (where  $c = 1/\int_A h(\mathbf{x}) d\mathbf{x}$ ), the variance of  $\tilde{\mu}$  will drop to zero, and  $\tilde{\mu}$  is equal to the exact value  $\int_A h(\mathbf{x}) d\mathbf{x}$ . However, we cannot directly use the IS method defined in (6) during such an extreme situation because we do not know the exact value of  $\int_A h(\mathbf{x}) d\mathbf{x}$  in advance.

Nevertheless, it gives us a significant hint that the closer  $g(\mathbf{x})$  and  $h(\mathbf{x})$  are, the more accurate the IS method's result is. The steps of the IS method for the computation of integral (3) are listed as follows:

- (1) Draw the samples  $\infty$  from  $g(\mathbf{x})$ .
- (2) Compute the importance weights  $\omega_i = h(\mathbf{x}_i)/g(\mathbf{x}_i)$ .
- (3) Use the mean of the computed weights to estimate the integral in (3):

$$\tilde{\mu} = \frac{1}{n} \sum_{i=1}^n \omega_i. \quad (7)$$

The following theorem shows that the IS estimator in (7) is unbiased.

**Theorem 1.** *The IS estimator  $\tilde{\mu}$  in (7) is an unbiased estimator of  $\mu$ .*

*Proof.* To prove that the IS estimator is unbiased, we only need to show that the expectation of  $\tilde{\mu}$  is equal to  $\mu$ :

$$\begin{aligned} E(\tilde{\mu}) &= E\left(\frac{1}{n} \sum_{i=1}^n \omega_i\right) \\ &= \frac{1}{n} \sum_{i=1}^n E(\omega_i) \\ &= E(\omega_i). \end{aligned} \quad (8)$$

Because  $\omega_i$  is a r.v. and  $\omega_i = h(\mathbf{X}_i)/g(\mathbf{X}_i)$ ,

$$\begin{aligned} E(\omega_i) &= E\left(\frac{h(\mathbf{X}_i)}{g(\mathbf{X}_i)}\right) \\ &= \int_A \frac{h(\mathbf{x})}{g(\mathbf{x})} \cdot g(\mathbf{x}) d\mathbf{x} \\ &= \int_A h(\mathbf{x}) d\mathbf{x} \\ &= \mu. \end{aligned} \quad (9)$$

We obtain the expression  $E(\tilde{\mu}) = \mu$ , which verifies that the IS estimator  $\tilde{\mu}$  in (7) is unbiased. So we complete the proof of this theorem.

Aside from being an unbiased estimator of the integral presented in (3), the IS estimator exhibits a more efficient and powerful integral computation than the MC estimator defined in (5), especially in higher-dimensional cases.  $\square$

**3.2. IS Algorithm for SD Computation.** We use the previously described IS method to compute the SD. Using the definition of SD in (1) is not appropriate in computing the SD value of a data point with respect to a dataset. The MC method in (6) becomes extremely inefficient when dimension  $p$  or sample size  $n$  is excessively large because the number of simplices

containing the original data point decreases with the increase in  $p$  or  $n$ .

The IS algorithm can transform the original p.d.f. into a highly efficient one that can construct the simplex containing the original data point. In the computation of SD, the MC method randomly selects  $p$  data points to construct the simplex, whereas the IS method chooses the data points that are likely to let the original data point inside the simplex. Figure 1 is a 2D example that is composed of 20 sample data points. The data point  $x_0$  is used to compute the SD value. After sampling the two data points ( $x_1$  and  $x_2$ ), only two more ( $x_3$  or  $x_4$ ) are needed to construct the simplex that contains the original data point  $x_0$ . In this illustrated example, we do not need to count all the simplices after getting  $x_1$  and  $x_2$ ; only  $x_3$  or  $x_4$  is considered as the final vertex of the simplices containing  $x_0$ .

We list the details of the IS algorithm for the computation of SD in high-dimensional cases. Suppose that  $\mathbf{X}^n$  is a sample with size  $n$  in  $\mathbb{R}^d$  (i.e.,  $\mathbf{X}^n = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ ) and  $x$  is a given point in  $\mathbb{R}^d$ , ( $d \geq 2$ ). The data points are in general position (i.e., any  $d$  data points can define a unique  $d-1$ -dimensional hyperplane in  $\mathbb{R}^d$ ). The procedure of using the IS algorithm to compute SD (i.e., the computation of  $SD(\mathbf{x}, \mathbf{X}^n)$ ) is summarized as follows:

- (1) Set the IS parameters, including the number of samples tries  $N$ .
- (2) Let  $t = 1, 2, \dots, N$ . Compute the importance weight  $\omega_t$  for the  $t$ -th sample try.

- (i) Randomly choose  $d$  sample points from  $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ , and denote them as  $\{\mathbf{X}_1^t, \mathbf{X}_2^t, \dots, \mathbf{X}_d^t\}$ .
- (ii) Let  $k = 1, \dots, d$ , and compute the simplex data point set  $\mathcal{U}_k^t$  (i.e., the datasets consist of the possible data points that can construct a simplex containing the original data point  $\mathbf{x}$ ).

Replace the  $k$ -th data point  $\mathbf{X}_k^t$  with the original data point  $x$  to obtain a dataset  $P_k^t$  with size  $d$  (i.e.,  $P_k^t = \{\mathbf{X}_1^t, \dots, \mathbf{X}_{k-1}^t, x, \mathbf{X}_{k+1}^t, \dots, \mathbf{X}_d^t\}$ ). Compute the unique director  $\mathbf{d}_k^t$  which is perpendicular to the hyperplane determined by  $P_k^t$ .

Project all data points  $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$  and  $\mathbf{x}$  along  $\mathbf{d}_k^t$ , and compute the projected value  $\{x_{t,k}^l\}$ , where  $t = 1, \dots, N$ ;  $k = 1, \dots, d$ ; and  $l = 0, 1, \dots, n$ , and  $x_{t,k}^0$  is the projected value of  $\mathbf{x}$  along  $\mathbf{d}_k^t$ .

Compute the simplex data point set  $\mathcal{U}_k^t = \{\mathbf{X}_l | (x_{t,k}^l - x_{t,k}^0) \times (x_t - x_{t,k}^0) < 0, l = 1, \dots, n\}$ .

- (iii) Let  $\mathcal{U}^t = \cap_{k=1}^d \mathcal{U}_k^t$ , and set  $\omega_t = \# \{\mathcal{U}^t\} / (n-d)$ , where  $t = 1, \dots, N$ .

- (3) The sample mean of  $\omega_t$  ( $t = 1, \dots, N$ ) can be treated as the IS estimator of  $SD(x, \mathbf{X}_n)$ ; that is,

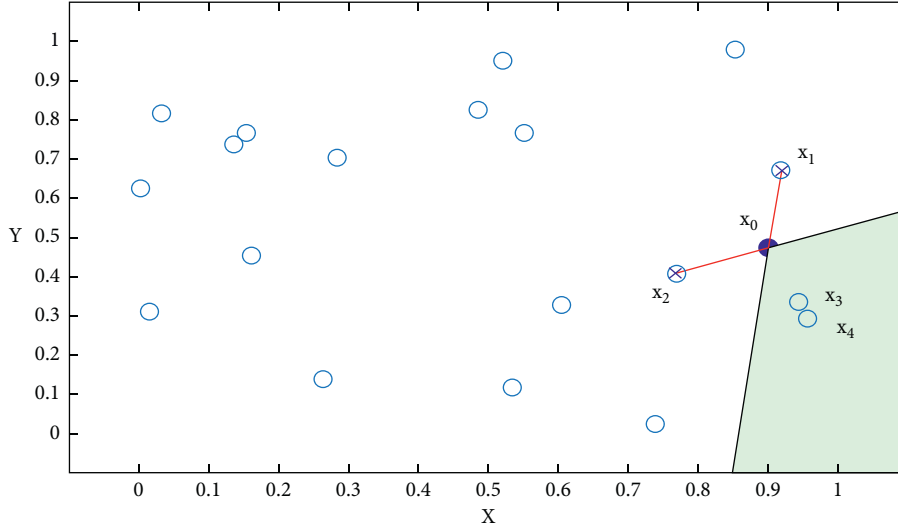


FIGURE 1: A 2D simulated example (sample size = 20) which indicates that only two possible trilaterals (or simplices) can contain the original data point  $x_0$  (the vertexes of the trilaterals are  $\{x_1, x_2, x_3\}$  and  $\{x_1, x_2, x_4\}$ ).

$$\widetilde{SD}(\mathbf{x}, \mathbf{X}_n) = \frac{1}{N} \sum_{t=1}^N \omega_t. \quad (10)$$

**Theorem 2.** *The computational complexity of using the IS algorithm to calculate SD is*

$$O(Nd^5n), \quad (11)$$

where  $N$  is the number of samples tries of the IS algorithm,  $d$  is the dimension of the sample data, and  $n$  is the sample size.

*Proof.* According to the steps for computing SD using the IS algorithm, we need to compute every  $\omega_i$  for  $i = 1, \dots, N$ . For every  $\omega_i$ , every selected sample data point  $\mathbf{X}_k^t$  for  $k = 1, \dots, d$  must be replaced. The computational complexity of finding the unique director perpendicular to the hyperplane is  $O(d^3)$ , whereas that of projecting all data points to the unique director is  $O(dn)$ . The total computational complexity is  $O(Nd^5n)$ . Then we complete the proof of this theorem.

Theorem 2 shows that the computational complexity of the IS algorithm for the computation of SD is a polynomial with dimension  $d$  and sample size  $n$  as its input arguments. While all other exact algorithms for the computation of SD are NP problems, especially when the dimension  $d \geq 5$ , there is no algorithm that can run faster than simply generating all simplices and computing the exact SD value (i.e., using  $O(n^{d+1})$  time) [12]. According to the definition of the IS algorithm in (7) and Theorem 1, the IS estimator defined in (10) is an unbiased estimator of  $SD(\mathbf{x}, \mathbf{X}_n)$ .  $\square$

#### 4. Performance Comparison

This section presents simulated and real data examples of SD computation. All results are obtained using *R* (version X64 3.6.2) and *MATLAB* (R2017a) on a Lenovo K42-80 laptop

computer (Intel(R) Core(TM) i7-6500U CPU@2.5 GHz, RAM 16.00 GB, Windows 10). The *R* and *MATLAB* codes for the results in this section are available upon request from the corresponding author.

**4.1. 2D Simulated Data Example.** In the simulated data experiment, we compare the computed SD results of the IS, exact, and approximate algorithms, including the MC method. The simulated dataset is sampled from a 2D multivariate normal distribution (i.e.,  $N(\vec{\mathbf{0}}_2, \mathbf{E}_2)$ , where  $\vec{\mathbf{0}}_2$  is 2D zeros vector and  $\mathbf{E}_2$  is a 2D unit matrix), and the sample size is 100.

We used the exact algorithm [21], the MC method, and the IS algorithm to compute the SD. The selected points  $\mathbf{x}$  are  $(0, 0)$ ,  $(0.5, 0.5)$ ,  $(1, 1)$ , and  $(2, 2)$ . We used the exact and approximate algorithms to compute the SD of  $\mathbf{x}$  with respect to the dataset. The number of random simplices was set to 100 for the MC and IS algorithms. All computations were repeated 50 times. The computed results (mean, standard deviation (sd), and total CPU time (s)) are summarized in Table 1 and Figure 2.

Since there is an exact algorithm for the SD computation in the 2D case, we can evaluate the accuracy of the IS and MC methods through their mean values and sd values. Moreover, the total CPU time consumed by every algorithm can reflect its efficiency. So, in this experiment, we use these three indicators (mean, sd, and total CPU time) to compare the performances of these algorithms (exact, MC, and IS methods) for the computation of SD.

The results reveal that (1) the exact algorithm consumes less CPU time (approximately 0.1 s), (2) the approximate algorithms (MC and IS) can achieve accurate results because their means are extremely close to the exact value, (3) IS performs better than MC as indicated by the smaller sd of the results of the former compared with those of the latter under the same CPU time, (4) all computed SD results from exact and approximate algorithms are zeros at point  $(2, 2)$ , which means that  $(2, 2)$  is outside the data cloud, and (5), with the

TABLE 1: Computed results (mean, sd, and total CPU time in s) from different algorithms (exact, MC, and IS) in 2D experiments.

		Exact	MC	IS
(0, 0)	Mean	0.2407	0.2380	0.2437
	sd	0	0.0420	0.0150
	Time	0.099	1.545	1.229
(0.5, 0.5)	Mean	0.1607	0.1528	0.1608
	sd	0	0.0326	0.0177
	Time	0.149	1.51	1.364
(1, 1)	Mean	0.0545	0.0594	0.0559
	sd	0	0.0245	0.0127
	Time	0.125	1.587	1.332
(2, 2)	Mean	0	0	0
	sd	0	0	0
	Time	0.223	1.594	1.492

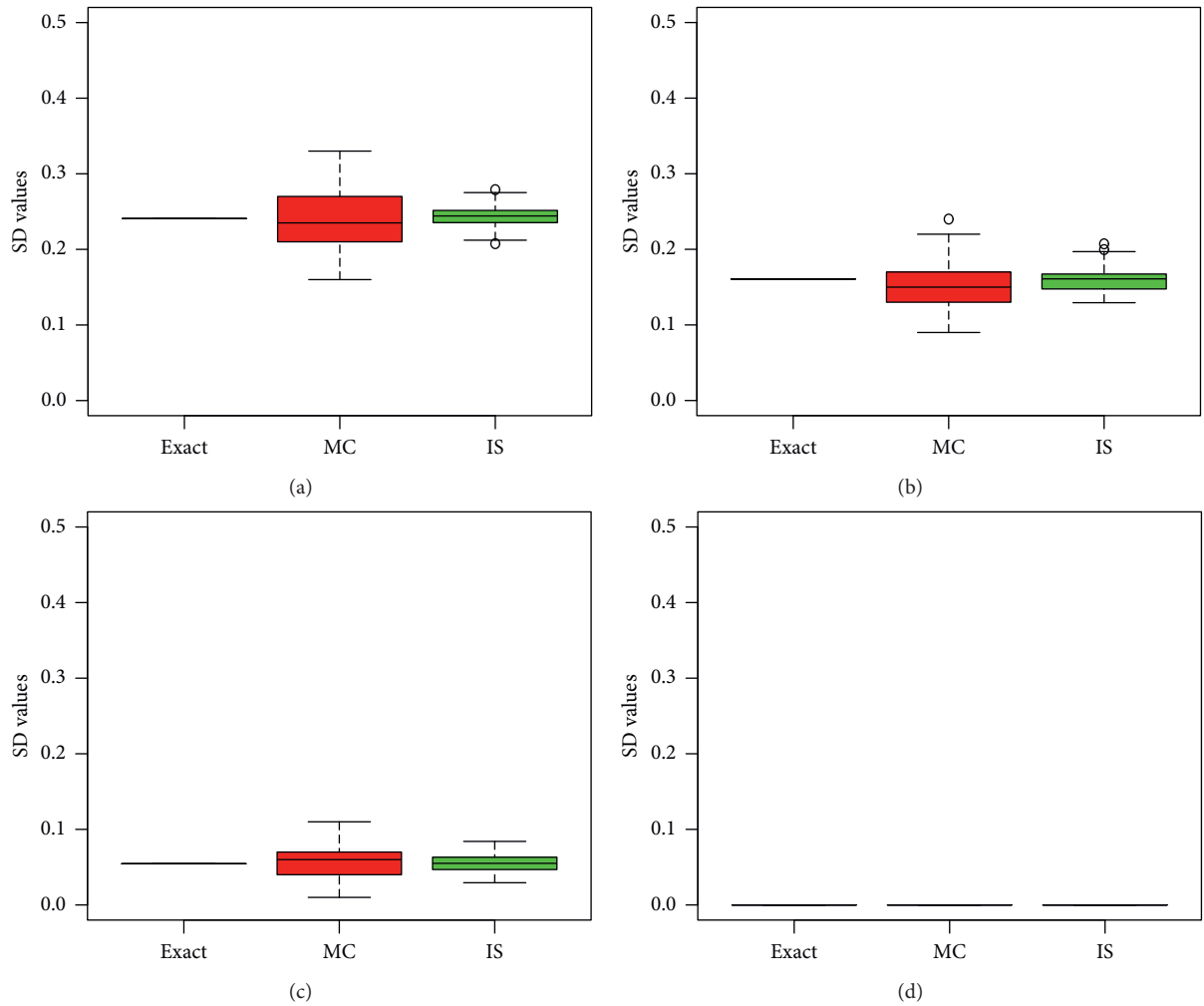


FIGURE 2: Boxplots of the results from different algorithms (exact, MC, and IS) in 2D experiments. (a–d) The results from the computed SD of (0, 0), (0.5, 0.5), (1, 1), and (2, 2), respectively.

exact algorithm, the simulated example also indicates that the IS algorithm can obtain highly accurate results.

**4.2. Higher-Dimensional Simulated Data Example.** In this subsection, we compute the SD of different data points by

using the MC and IS algorithms in 3D and five-dimensional simulated dataset. We did not use the exact algorithm [21] because it cannot obtain any result within three hours.

In the 3D case, the dataset was sampled from  $N(\vec{0}_3, E_3)$ , and the sample size was 1000. We used MC and IS methods to compute the SD of points (0, 0, 0), (0.5, 0.5, 0.5), and

TABLE 2: Results (mean, sd, and total CPU time in s) were obtained by MC and IS methods with respect to different data points in 3D experiments.

	(0, 0, 0)	(0.5, 0.5, 0.5)	(1, 1, 1)	Total CPU time
MC	0.1254 (0.0330)	0.0460 (0.0242)	0.0054 (0.0073)	5.13
IS	0.1280 (0.0121)	0.0449 (0.0078)	0.0050 (0.0033)	5.23

The sd values are listed in the parentheses behind the mean values.

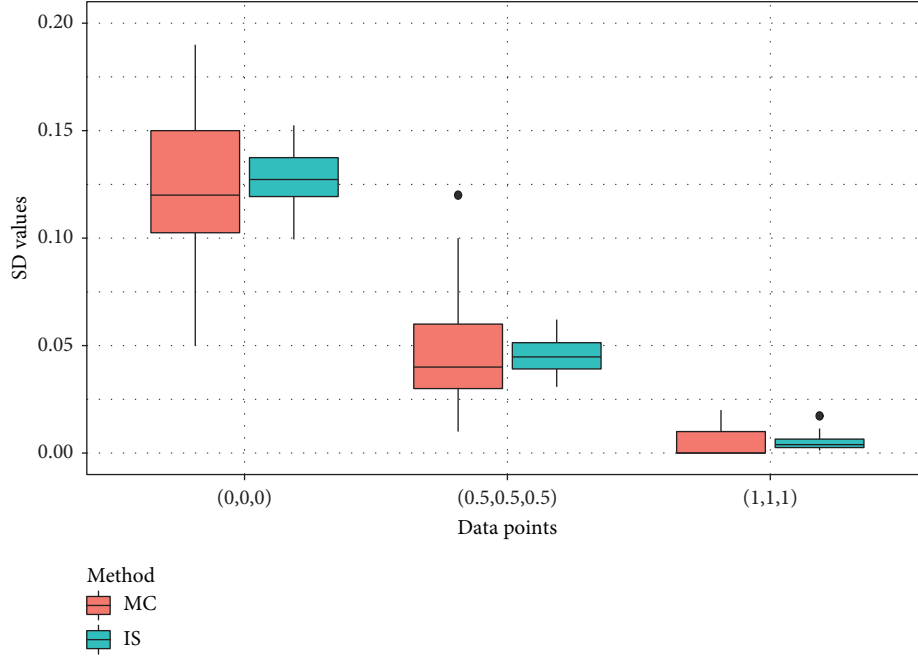


FIGURE 3: Boxplots of the results obtained by MC and IS methods in 3D experiments.

(1, 1, 1). We set the number of random simplices to 100 and repeated the computation 50 times. The computed results are summarized in Table 2 and Figure 3.

Because the exact algorithm cannot get any computed SD results within three hours when dimension  $d \geq 3$ , we can only use MC and IS methods for the computation of SD in this subsection. Three indicators (mean, sd, and total CPU time) are summarized for the evaluation of the approximate methods. The mean values can be seen as the final computed SD results and the sd reflects the accuracy of the method (the smaller, the more accurate). The total CPU time reflects the efficiency of the method because it is more efficient if the method consumes less CPU time in the same computation of SD.

Table 2 and Figure 3 indicate that (1) the computed SD results decrease when the data points are changed from (0, 0, 0) to (1, 1, 1); the data point (0, 0, 0) is deeper than the data point (1, 1, 1) with respect to the dataset; (2) the two methods have similar computational efficiencies because they consume almost the same total CPU time; (3) the sd obtained by the IS method is smaller than that calculated by the MC method, which means that the former is more accurate than the latter in this case.

In the five-dimensional case, the dataset was sampled from  $N(\vec{0}_5, E_5)$ , and the sample size was 1000. We used

MC and IS methods to compute the SD of points (0, 0, 0, 0, 0), (0.5, 0.5, 0.5, 0.5, 0.5), and (1, 1, 1, 1, 1). The number of random simplices was 100, and the computations were repeated 50 times. The computed results (mean, sd, and total CPU time in s) are presented in Table 3 and Figure 4.

Table 3 and Figure 4 show that (1) the computed SD values decrease when the data points are changed from (0, 0, 0, 0, 0) to (1, 1, 1, 1, 1), thereby suggesting that the former is deeper than the latter; (2) the SD values in the five-dimensional examples are slightly smaller than those in 3D examples because the sparsity of the data points increases when the dimension is increased from three to five; (3) the IS algorithm performs better than the MC approach as indicated by the smaller sd of the results of the former compared with those of the latter; (4) the two approximate algorithms consume almost the same CPU time; (5) even after using 100 random simplices, the MC algorithm cannot find any simplex containing point, whereas the IS algorithm can identify many simplices. In conclusion, the IS method outperforms the MC method in terms of accuracy in these simulated examples.

We also evaluated the MC and IS methods with other numbers of random samples tries in different datasets. The findings show that the result's accuracy increases with the



TABLE 3: Results (mean, sd, and total CPU time in s) were obtained by MC and IS methods in five-dimensional experiments.

	(0, 0, 0, 0, 0)	(0.5, 0.5, 0.5, 0.5, 0.5)	(1, 1, 1, 1, 1)	Total CPU time
MC	0.0360 (0.0160)	0.0036 (0.0066)	0 (0)	7.67
IS	0.0315 (0.0076)	0.0028 (0.0023)	$2.8 \times 10^{-6}$ ( $7.3 \times 10^{-6}$ )	8.42

The sd values are listed in the parentheses behind the mean values.

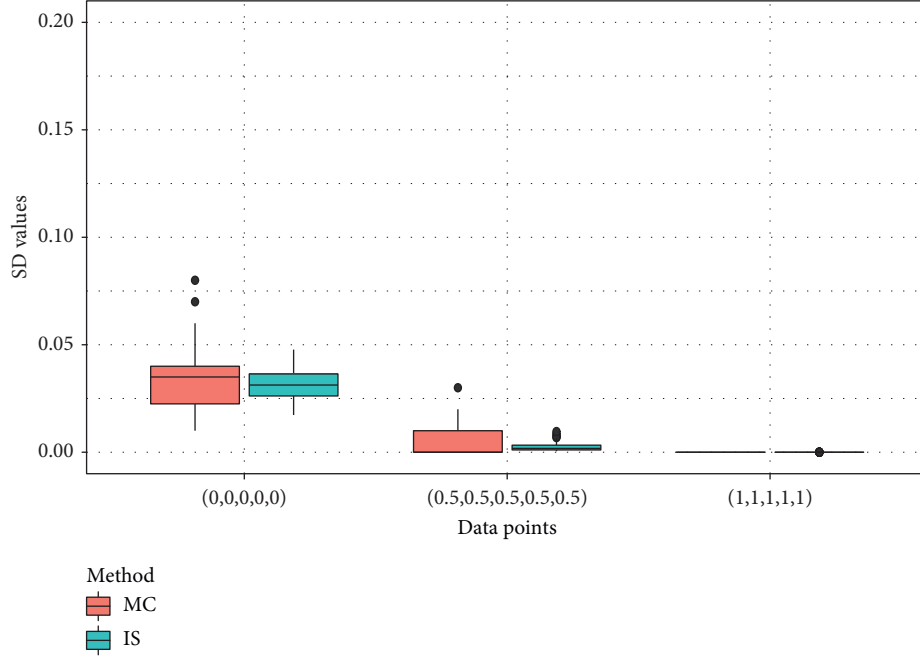


FIGURE 4: Boxplots of the results obtained by MC and IS methods in five-dimensional experiments.

increase in the number of random samples tries. In addition, the number of random samples tries can be used by IS method in lots of datasets. It is found in our experiments that, if we set the number of random samples tries  $N=1000$ , the IS method can obtain the computed SD results within one second when dimension  $d \leq 10$  and sample size  $n \leq 10000$ .

## 5. Application to Regression and Real Data Example

One of the most important extensions of SD is the robust estimation of regression based on SD. To demonstrate the relevant concept, we consider the linear regression model as follows:

$$Y = \alpha + \beta X + \varepsilon, \quad (12)$$

where random variables  $X$  and  $Y$  are in  $R^1$ ,  $\varepsilon \sim N(0, \sigma^2)$ , and  $\alpha, \beta$ , and  $\sigma^2$  are unknown parameters.

Considering that  $SD(\mathbf{x}, \mathbf{X}^n)$  can measure the depth of  $\mathbf{x}$  with respect to  $\mathbf{X}^n$ , we extend the definition of SD to regression (12) and determine the simplicial regression depth:

$$SD(\theta, \mathbf{W}^n) = \binom{n}{3}^{-1} \sum_{i < j < k} A(r_i(\theta), r_j(\theta), r_k(\theta)), \quad (13)$$

where  $\theta = (\alpha, \beta)$  are the parameters,  $\mathbf{W}^n = (Y^n, X^n)$  are the samples of the model defined in (12), and  $r_i(\theta) = Y_i - \alpha - \beta X_i$  is the residual based on the  $i$ -th sample and

$$A(r_i(\theta), r_j(\theta), r_k(\theta)) = \begin{cases} 1, & r_i(\theta), r_j(\theta), r_k(\theta) \text{ have alternating signs,} \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

The SD based estimator of (12) can be defined as the maximum of  $SD(\theta, \mathbf{W}^n)$ ; that is,

$$\hat{\theta}_{SD} = \arg \max_{\theta} SD(\theta, \mathbf{W}^n). \quad (15)$$

We consider the physical experiment data concerning the relationship between the atmospheric pressure and boiling point of water, which was discussed by a Scottish physicist named James D. Forbes [22]. In the mid-nineteenth century, this experiment can illustrate whether the simple measurement of the boiling point of water can substitute for the direct reading of the barometric pressure. The dataset was collected in the Alps in Scotland (Table 4 and Figure 5).

The linear regression model in (12) was used to fit the Forbes dataset. We used LS and SD methods to estimate the parameters of the model in (12). The function “lm” in R Stats package (“stat”) can be used to determine the LS estimator of

TABLE 4: Forbes data collected in Alps and Scotland. The temperature is measured using the boiling point ( $^{\circ}\text{F}$ ) and the pressure is obtained using barometric pressure (mmHg).

No.	Temperature	Pressure
1	194.5	20.79
2	194.3	20.79
3	197.9	22.40
4	198.4	22.67
5	199.4	23.15
6	199.9	23.35
7	200.9	23.89
8	201.1	23.99
9	201.4	24.02
10	201.3	24.01
11	203.6	25.14
12	204.6	26.57
13	209.5	28.49
14	208.6	27.76
15	210.7	29.04
16	211.9	29.88
17	212.2	30.06

the model in (12). For the SD based method, we combined quasi-Newton [23] and IS methods to find the maximum point of (15). Moreover, we performed three statistical tests (i.e., the  $R$  square value, normality test, and the test of goodness of fit [24]) for every fitted regression model to get a more insightful analysis. The  $R$  square (or adjusted  $R$  square) value from the significance test gives the percentage that the dependent variable ( $Y$ ) can be explained by the fitted model ( $\alpha + \beta X$ ) (see (12)). The normality test is used to test whether the residuals of the fitted model obey normal distribution which is the basis of other statistical tests. For example, under the assumption of normality, the  $F$  statistic value in the test of goodness of fit can be used to determine whether the fitted regression model makes sense.

We first used the LS method and SD approaches to compute the linear regression model with the original Forbes dataset (Table 4, denoted as original data in this section). The computed regression results are summarized in Table 5 and Figure 5(a); their corresponding statistical tests are summarized in Table 6 and Figure 6.

Table 5 and Figure 5(a) show that the LS and SD estimators obtained the very similar intercept parameter and slope parameter. This finding suggests that the SD method can capture the same accurate regression results compared with LS method.

The statistical test results have also confirmed the finding since the results from LS and SD methods were also very similar. They have very high  $R$  square values which indicate more than variance of the dependent variable that can be explained by the fitted model. Under significance level 0.01, we accept the assumption of normality and they pass the goodness of fit test (i.e., the  $p$  value of  $F$  statistic is almost zero). In addition, if one needs a higher level of significance (such as 0.05) in this example, then some statistical techniques (e.g., Box-Cox transformation or strong influence points detection) can be used to improve the regression model (see more details in [22]). However, this is another

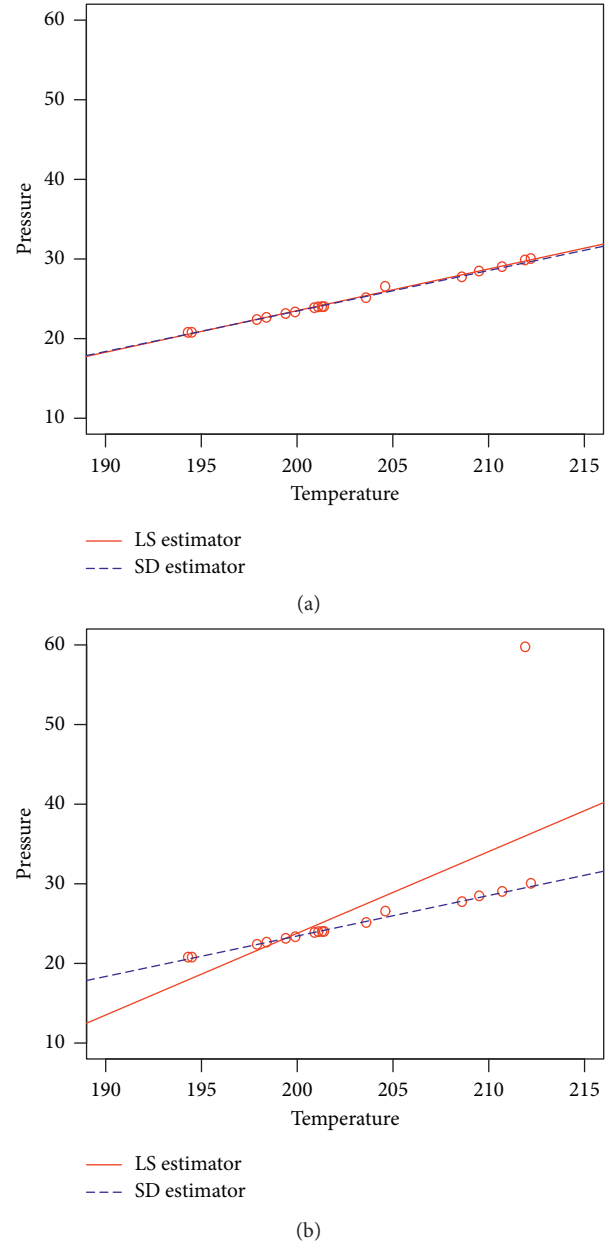


FIGURE 5: Regression analysis results from original (a) and contaminated (b) datasets.

research topic and there is a lack of sample points in this example; we only focus on the robustness of the regression model computed from different methods, especially when the dataset is contaminated, and that is what we do in the next experiment.

In the following experiment, we worked with a contaminated dataset from Forbes data. We intentionally changed the pressure of the 16th data point from 29.88 to 59.76. The new dataset was denoted as the contaminated data (Figure 5(b)). We compared the SD and LS methods' performances in the linear regression model with the contaminated dataset. The regression results are presented in Table 5 and Figure 5(b). Their corresponding statistical tests are summarized in Table 6 and Figure 7.



TABLE 5: Computed regression analysis results (original and contaminated) from Forbes data.

	Original data		Contaminated data	
	$\alpha$	$\beta$	$\alpha$	$\beta$
LS estimator	-81.0637	0.5229	-181.527	1.0266
SD estimator	-78.2631	0.5086	-78.2623	0.5085

TABLE 6: The statistical tests for regression analysis with original data and contaminated data using LS and SD methods.

	Original data			Contaminated data		
	$R^2$ (Adjusted $R^2$ )	$F$ ( $p$ -value)	$S-W$ ( $p$ -value)	$R^2$ (Adjusted $R^2$ )	$F$ ( $p$ -value)	$S-W$ ( $p$ -value)
LS	0.9944 (0.9941)	2677 (0)	0.8723 (0.024)	0.7650 (0.7493)	48.82 ( $4.37 \times 10^{-6}$ )	0.6473 ( $3.10 \times 10^{-5}$ )
SD	0.9922 (0.9918)	1928 (0)	0.8574 (0.0140)	0.9917 (0.9912)	1812 (0)	0.2861 ( $3.13 \times 10^{-8}$ )

Note: (Adjusted) is the adjusted  $R$  square value,  $F$  and  $p$  value are the  $F$  statistic value and its corresponding  $p$  value, and  $S-W$  and  $p$  value are the Shapiro-Wilk test statistic value and its corresponding  $p$  value.

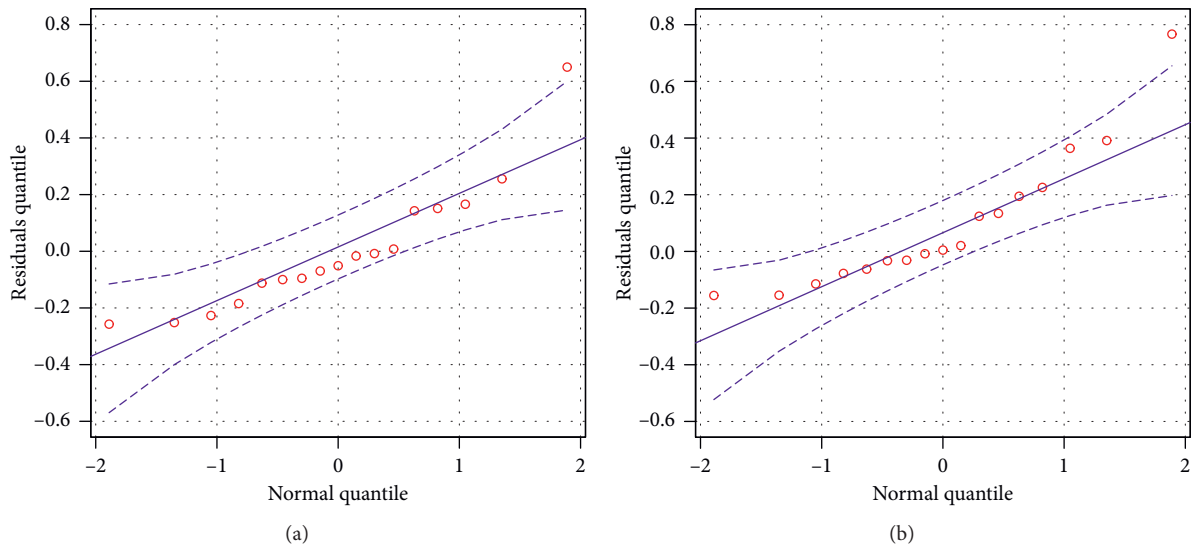


FIGURE 6: Quantile-quantile (QQ) plots of the residuals from LS and SD methods with the original data. (a) The QQ plot from the LS method. (b) The QQ plot from the SD method.

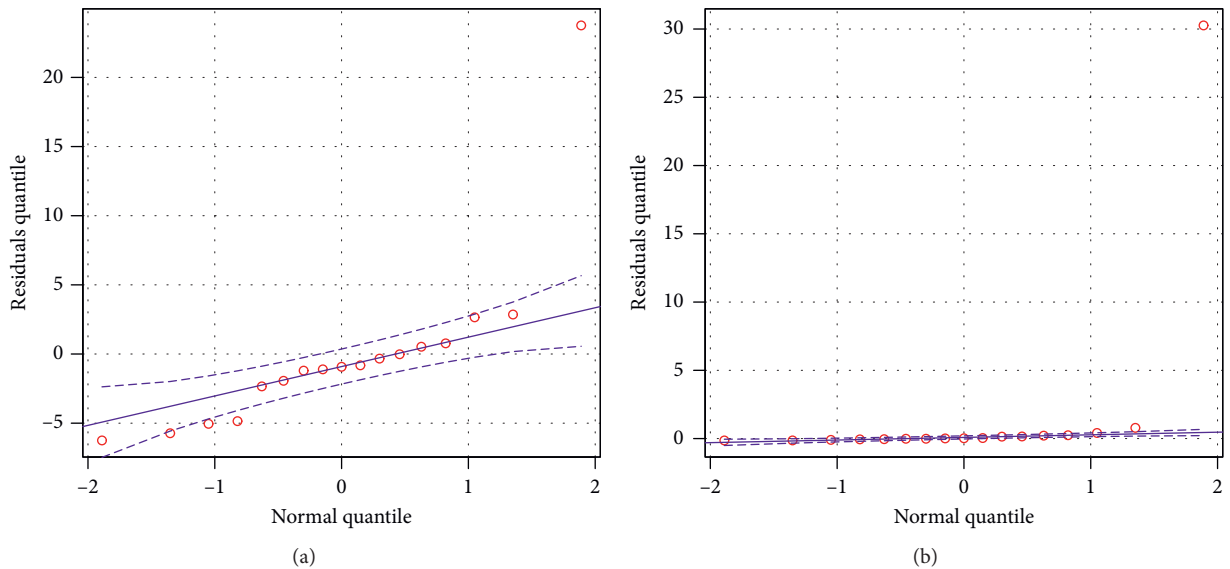


FIGURE 7: Quantile-quantile (QQ) plots of the residuals from LS and SD methods with the contaminated data. (a) The QQ plot from the LS method. (b) The QQ plot from the SD method.

The results show that the LS estimator is greatly influenced by the contaminated data point, whereas the SD estimator can maintain satisfactory performance. The slope parameter estimated by the LS estimator changes from 0.5229 to 1.0266, which cannot reflect the actual variation trend of the pressure-temperature curve. By contrast, the SD estimator is not affected by the contaminated data point and can still provide the actual variation trend. The estimated slope parameters obtained using SD method for two different datasets are 0.5086 and 0.5085, respectively. The statistical test results show that, under the influence of the contaminated data point, the residuals of the fitted models from the two methods do not pass the normality test. However, the  $R$  square (or Adjusted  $R$  square) value from the SD method (0.9917) is much larger than that of the LS method (0.7650) which means that the regression line from the SD method can explain more percentage of the variance of dependence variable compared with that of the LS method. These results imply that the SD estimator outperforms the LS estimator in the contaminated dataset experiment in terms of robustness.

## 6. Conclusions

The concept of statistical depth plays an important role in mathematical sciences, engineering, regression analysis, and life sciences. In this study, we computed the SD using the IS method and found that this new approach performs better than other exact and MC methods in terms of accuracy and efficiency. The simulated and real data examples illustrated the advantage of this new method. Finally, we tested the SD method based regression analysis through a concrete physical data example. The result indicated the excellent robustness of the proposed method compared with the LS estimation.

Given the many favorable properties of the proposed method, further research can be conducted on different angles. First, the IS parameter (i.e., number of sample tries  $N$ ) plays an important role in the computation of SD, so the determination of  $N$  before the performance of IS algorithm is yet to be thoroughly investigated. Second, the IS method for the SD computation can be improved by sampling the data points via other more important simplices (not the last data point in the possible simplices). Third, with the development of modern computer science, multicore high-performance computer is gaining popularity. Therefore, the IS method can be extended to a parallel computation based version. Lastly, the approximate algorithms (advanced MC methods) for other statistical depths (e.g., halfspace depth, projection depth, and regression depth) can be further explored.

## Data Availability

The experimental data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (11501320 and 11701318), the Natural Science Foundation of Shandong Province (ZR2021MA090 and ZR2019PG005), and the Postdoctoral Science Foundation of China (2020M682143).

## References

- [1] Y. Zuo and R. Serfling, "General notions of statistical depth function," *Annals of Statistics*, vol. 28, pp. 461–482, 2000.
- [2] Y. Zuo and R. Serfling, "Structural properties and convergence results for contours of sample statistical depth functions," *Annals of Statistics*, vol. 28, pp. 483–499, 2000.
- [3] Y. Zuo, "Projection-based depth functions and associated medians," *Annals of Statistics*, vol. 31, no. 5, pp. 1460–1490, 2003.
- [4] Y. Zuo, H. Cui, and X. He, "On the stahel-donoho estimators and depth-weighted means for multivariate data," *Annals of Statistics*, vol. 32, no. 1, pp. 167–188, 2004.
- [5] Y. Zuo, "On general notions of depth in regression," 2018, <https://arxiv.org/abs/1805.02046>.
- [6] W. Shao and Y. Zuo, "Computing the halfspace depth with multiple try algorithm and simulated annealing algorithm," *Computational Statistics*, vol. 35, no. 1, pp. 203–226, 2020.
- [7] J. W. Tukey, "Mathematics and the picturing of data," in *Proceedings of the International Congress of Mathematicians*, vol. 2, pp. 523–531, Vancouver, Canada, August 1974.
- [8] X. Liu, Y. Zuo, and Q. Wang, "Finite sample breakdown point of Tukey's halfspace median," *Science China Mathematics*, vol. 60, no. 5, pp. 861–874, 2017.
- [9] X. H. Liu, S. H. Luo, and Y. J. Zuo, "The limit of finite sample breakdown point of tukey's halfspace median for general data," *Acta Mathematica Sinica, English Series*, vol. 34, no. 9, pp. 1403–1416, 2018.
- [10] W. Shao and Y. Zuo, "Simulated annealing for higher dimensional projection depth," *Computational Statistics & Data Analysis*, vol. 56, no. 12, pp. 4026–4036, 2012.
- [11] R. Y. Liu, "On a notion of data depth based on random simplices," *Annals of Statistics*, vol. 18, no. 1, pp. 405–414, 1990.
- [12] R. Serfling and Y. Wang, "On liu's simplicial depth and randles' interdirections," *Computational Statistics & Data Analysis*, vol. 99, pp. 235–247, 2016.
- [13] P. J. Rousseeuw and M. Hubert, "Regression depth," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 388–402, 1999.
- [14] X. Liu and Y. Zuo, "Computing halfspace depth and regression depth," *Communications in Statistics - Simulation and Computation*, vol. 43, no. 5, pp. 969–985, 2014.
- [15] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, USA, 2001.
- [16] S. C. Kou, Q. Zhou, and W. H. Wong, "Eqri-energy sampler with allplications in statistical inference and statistical mechanics," *Annals of Statistics*, vol. 34, pp. 1581–1619, 2006.
- [17] F. Liang, C. Liu, and R. J. Carroll, *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*, John Wiley & Sons, NJ, USA, 2011.
- [18] W. Shao, G. Guo, G. Zhao, and F. Meng, "Simulated annealing for the bounds of Kendall's  $\tau$  and Spearman's  $\rho$ ," *Journal of Statistical Computation and Simulation*, vol. 84, no. 12, pp. 2688–2699, 2014.

- [19] W. Shao, G. Guo, F. Meng, and S. Jia, "An efficient proposal distribution for Metropolis-Hastings using a B-splines technique," *Computational Statistics & Data Analysis*, vol. 57, no. 1, pp. 465–478, 2013.
- [20] J. Shao, *Mathematical Statistics*, Springer, Berlin, 2003.
- [21] O. Pokotylo, P. Mozharovskyi, and R. Dyckerhoff, "Depth and depth-based classification with r-package ddalpha," 2016, <https://arxiv.org/abs/1608.04109>.
- [22] S. Weisberg, *Applied Linear Regression*, John Wiley & Sons, New Jersey NY USA, 1980.
- [23] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
- [24] R. R. Development Core Team, *A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020.