IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 8, 2019, accepted April 25, 2019, date of publication May 8, 2019, date of current version May 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915642

Fast and Unbiased Estimation of Volume Under Ordered Three-Class ROC Surface (VUS) Based on Dynamic Programming

SHUN LIU[1], **WEICHAO XU**[1], **(Member, IEEE), XU SUN**[2], **(Member, IEEE), AND YUN ZHANG**[1]

[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[2]Baidu, Inc., Beijing 100085, China

Corresponding author: Weichao Xu (wcxu@gdut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Project 61771148 and Project U1501251.

ABSTRACT Receiver operating characteristic (ROC) analysis has been widely used in two-class problems. However, in practice, three-class problems are frequently encountered, especially in the area of medicine. To evaluate the performance of three-class classifiers, researchers have proposed the volume under the three-class ROC surface (VUS) as a figure-of-merit. Unfortunately, to the best of our knowledge, however, all the existing methods suffer heavy computational loads. In this paper, to overcome such an unsatisfactory problem, we develop an efficient dynamic programming-based algorithm for unbiased estimation of the VUS and the corresponding variance. The Monte Carlo simulations verified both the unbiasedness and computing efficiency of our algorithm compared with the state-of-the-art work proposed by Waegeman and co-authors.

INDEX TERMS Receiver operating characteristic (ROC), volume under the ROC surface (VUS), dynamic programming, fast algorithm.

NOMENCLATURE

θ	Population version of VUS
$\hat{\theta}$	Sample version of VUS
$\hat{\sigma}^2_{\hat{\theta}}$	Unbiased estimation of the variance of $\hat{\theta}$
$F_i(\cdot)$	Cumulative distribution function of Class i
$\mathbb{P}(\cdot)$	Probability of occurrence of the event in ( )
$\mathbb{E}(\cdot)$	Expectation of random variables
$\mathbb{V}(\cdot)$	Variance of random variables
$\mathcal{E}(\cdot)$	Number of events satisfying the relationship in ( )
$\mathcal{I}(\cdot)$	Indicator function that returns unity (zero) when its argument is true (false)
$\mathcal{O}(\cdot)$	Big-Oh notation for time complexity
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\mathcal{U}(a, b)$	Uniform distribution with parameters a and b
$\mathcal{L}(\mu, \sigma)$	Laplace distribution with parameters μ and σ
$\mathcal{R}(\sigma^2)$	Rayleigh distribution with parameter σ^2
$\triangleq$	Means 'is defined as'
n'	$\triangleq n-1$
$n^{[2]}$	$\triangleq n(n-1)$

The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Md Asaduzzaman.

I. INTRODUCTION

Receiver operating characteristic (ROC) analysis is an indispensable framework in signal detection or medical decision making, with a major application for characterizing the performance of binary classifications [1]–[4]. In essence, ROC analysis is a supervised methodology requiring the prior knowledge of the sample membership (abnormal vs. normal). Given such knowledge, an ROC curve, which is a plot of false positive rate against true positive rate (sensitivity), can be defined according to various decision threshold settings. The area under the ROC curve (AUC) can then be computed, either analytical or empirically, as an index to summarize the overall performance of the binary classifier.

In many scenarios, especially in the area of medicine, the diagnostic tasks involve three outcomes, namely, abnormalities are two-sided. For example, the heart signal is classified as Bradycardia (slower rhythm), normal, and Tachycardia (faster rhythm); and the blood pressure is classified as Hypertension (lower pressure), normal, and Hypertension (higher pressure). In communication, the amplitudes of transmitted signals fall into three categories, as negative (binary ''0''), idle (baseline), and positive (binary ''1''). For such cases with three ordered alternatives, Scurfield [5] extended ROC curve and AUC to ROC surface and volume under the surface (VUS) in a parallel manner. Following this direction, other researchers have proposed various methods to estimate the mean and variance of VUS [6]–[9]. Besides the above-mentioned methods focusing on one-dimensional ordered three-class measurements, other techniques for ROC analysis of high-dimensional data have also been proposed, including Mossman's three-way method [10], He's likelihood ratio based framework [11] and Dreiseitl's nonparametric algorithms [12].

From the viewpoint of computation, all the existing methods are unsatisfactory. In other words, the time complexity of all methods is polynomial, ranging from quintic order [6]–[8], [11], [12], to quadratic order (the state-of-the-art) [9]. Moreover, as shown later on in Section V, the estimator for the variance of VUS in [9] is biased, which might be misleading in practice. Motivated by such unsatisfactory situation, in this paper, we derive a fast and unbiased estimator for the variance of VUS based on the formulas of Dreiseitl et al. [12] and Nakas and Yiannoutsos [6]. Extending our previous work [13] as well as that of Waegeman *et al.* [9], we first reformulate an unbiased estimator, based on [12] and [6], into a mathematically equivalent recursive structure. We then apply the well-known dynamic programming technique [14] that widely used in the literature, such as optimal control [15], speech recognition [16], communication [17], energy management in electric bus [18], just to name a few. As shown in the analysis of time complexity later on, the new algorithm is capable of reducing the time complexity from a quintic order down to a linearithmic order, which is the major contribution of this work.

The rest of this paper is structured as follows. Section II describes the basic idea of three-class ROC surface. In Section III, we present the basic definitions of the VUS as well as the associated unbiased estimators. Section IV is devoted to developing a linearithmic algorithm based on dynamic programming. In Section V, numerical experiments are undertaken to demonstrate the efficiency and unbiasedness of our algorithm. Section VI gives a discussion regarding the extension of our algorithm to multi-class scenarios, where the class number is greater than three. Finally, we draw our conclusion in Section VII.

II. ORDERED THREE-CLASS ROC SURFACE

Let $\{X_{1i}\}_{i=1}^{n_1}$, $\{X_{2j}\}_{j=1}^{n_2}$, $\{X_{3k}\}_{k=1}^{n_3}$ be three independent and identically distributed (i.i.d.) samples drawn from three continuous populations with cumulative distribution functions (cdfs hereafter) $F_1(x)$, $F_2(y)$ and $F_3(z)$, respectively. Suppose that we are going to design a classifier to discriminate the three classes based on two thresholds $th_1 = x$ and $th_2 = z$, where $-\infty < x < z < +\infty$. As illustrated in Figure 1, three decision regions are defined by the two thresholds. For a newly observed value w, a natural criterion is: decide w to X_1, X_2, and X_3 if w falls in the region of $(-\infty, x)$, (x, z), and $(z, +\infty)$, respectively. Write

$$P_1(x, z) \triangleq \mathbb{P}(X_1 < x), \tag{1}$$

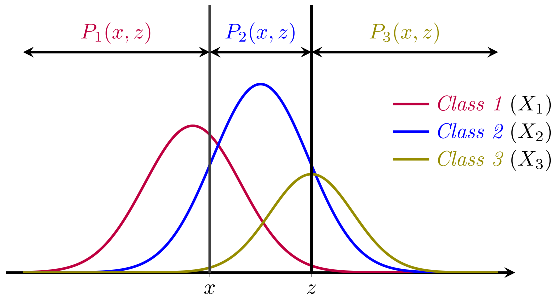


FIGURE 1. Schematic illustration of the double-threshold classifier. The three probabilities P_1,P_2 and P_3 are defined in (1)–(3), respectively.

$$P_2(x, z) \triangleq \mathbb{P}(x < X_2 < z), \tag{2}$$
$$P_3(x, z) \triangleq \mathbb{P}(z < X_3). \tag{3}$$

It is obvious that P_1, P_2 and P_3 are the probabilities that the classifier correctly classifies each sample to its true class (Fig. 1). For each pair of (x, z), there exists a corresponding point (P_1, P_2, P_3) in the three-dimensional space. With different decision criteria, i.e. x and z, a surface, called ROC surface, can be described by the simultaneous equations (1)–(3) (Fig. 2).

III. VOLUME UNDER THE SURFACE

A. DEFINITION OF VUS

As illustrated in Fig. 2, the volume formed by the surface and the three plenary walls, i.e., the volume under the ROC surface (VUS), is determined by [19]

$$\theta = \int_0^1 \int_0^1 P_2(x, z) dP_1(x, z) dP_3(x, z) \tag{4}$$

which can also be interpreted as the probability of the triplet X_3, X_2, X_1 being in descending order [5], i.e.,

$$\theta = \mathbb{P}(X_3 > X_2 > X_1). \tag{5}$$

Given the probabilistic interpretation (5) above, a natural sample version can be constructed, as [6], [9]

$$\hat{\theta} \triangleq \frac{1}{n_1\, n_2\, n_3} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \tag{6}$$

where $\mathcal{I}(\cdot)$ is the indicator function returning unity (zero) when its argument is true (false), and n_1, n_2, n_3 are the sample sizes with respect to three classes.

Remark 1: The sample version $\hat{\theta}$ defined in (6) is an unbiased estimator of θ defined in (5), since, taking expectations on both sides of (6) and imposing the i.i.d. assumption, we have

$$\begin{aligned}\mathbb{E}(\hat{\theta}) &= \frac{1}{n_1\, n_2\, n_3} n_1\, n_2\, n_3 \mathbb{E}[\mathcal{I}(X_3 > X_2 > X_1)] \\ &= \iiint_{x_3 > x_2 > x_1} dF_1(x_1) dF_2(x_2) dF_3(x_3) \\ &= \mathbb{P}(X_3 > X_2 > X_1) \\ &= \theta \textit{ that defined in (5).}\end{aligned}$$

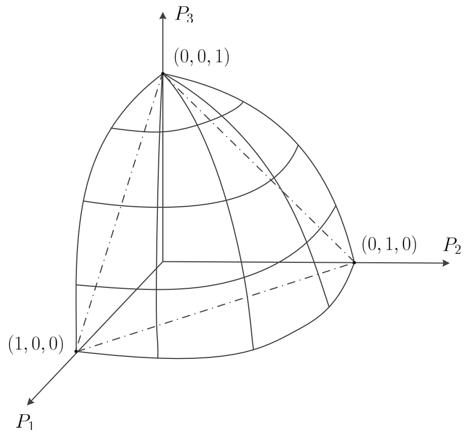


FIGURE 2. **Schematic illustration of the geometry of ROC surface. The three axes correspond to P_1, P_2 and P_3 defined in (1)–(3), respectively.**

Remark 2: In [7], the authors compared several nonparametric smoothing methods based on kernel density estimation, for the point estimate of θ defined in (5). Numerical results suggest that, in terms of unbiasedness, the sample version of (6) outperform the nonparametric smoothing methods. Because unbiasedness is a critical feature for estimators, we only focus on the statistical properties of $\hat{\theta}$ in (6) throughout this work.

B. VARIANCE OF $\hat{\theta}$

Given the sample version of (6), it is necessary to estimate its variance, which is needed to calculate the confidence interval. In a similar procedure as Dreiseitl *et al.* [12], it follows that the variance of $\hat{\theta}$ is

$$\begin{aligned}\mathbb{V}(\hat{\theta}) = \frac{1}{n_1 n_2 n_3} \times \big[&\theta(1-\theta) \\ &+(n_3-1)(q_{12}-\theta^2) \\ &+(n_2-1)(q_{13}-\theta^2) \\ &+(n_1-1)(q_{23}-\theta^2) \\ &+(n_2-1)(n_3-1)(q_1-\theta^2) \\ &+(n_1-1)(n_3-1)(q_2-\theta^2) \\ &+(n_1-1)(n_2-1)(q_3-\theta^2)\big] \end{aligned} \tag{7}$$

where

$$q_{12} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3' > X_2 > X_1) \tag{8}$$
$$q_{13} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3 > X_2' > X_1) \tag{9}$$
$$q_{23} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3 > X_2 > X_1') \tag{10}$$
$$q_{1} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3' > X_2' > X_1) \tag{11}$$
$$q_{2} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3' > X_2 > X_1') \tag{12}$$
$$q_{3} = \mathbb{P}(X_3 > X_2 > X_1 \cap X_3 > X_2' > X_1') \tag{13}$$

with X' being an i.i.d. copy of X. Note that the q-terms above are all probabilities of two three-tuples that are simultaneously ordered as indicated by the two inequalities in the parentheses.

C. UNBIASED ESTIMATOR OF $\mathbb{V}(\hat{\theta})$–SLOW VERSION

Theorem 1: Let $\hat{\theta}$ be defined as in (6) with respect to three i.i.d. samples $\{X_{1i}\}_{i=1}^{n_1}$, $\{X_{2j}\}_{j=1}^{n_2}$, $\{X_{3k}\}_{k=1}^{n_3}$ drawn from three continuous distributions, respectively. Let $\sigma_{\hat{\theta}}^2$ be a compact notation of $\mathbb{V}(\hat{\theta})$. Denote by $n_i' \triangleq n_i - 1$, $i = 1, 2, 3$. Then, an unbiased estimator of $\mathbb{V}(\hat{\theta})$ in (7), denoted by $\hat{\sigma}_{\hat{\theta}}^2$, can be established, as

$$\begin{aligned}\hat{\sigma}_{\hat{\theta}}^2 = \frac{1}{n_1' n_2' n_3'}\big[&\hat{q}_0 + n_3'(\hat{q}_{12}-\hat{\theta}^2) + n_2'(\hat{q}_{13}-\hat{\theta}^2) + n_1'(\hat{q}_{23}-\hat{\theta}^2) \\ &+ n_2' n_3'(\hat{q}_1-\hat{\theta}^2) + n_1' n_3'(\hat{q}_2-\hat{\theta}^2) + n_1' n_2'(\hat{q}_3-\hat{\theta}^2)\big]\end{aligned} \tag{14}$$

where

$$\hat{q}_0 = \hat{\theta}(1-\hat{\theta}) \tag{15}$$

$$\hat{\theta} = \frac{1}{n_1 n_2 n_3}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\mathcal{I}(X_{3k} > X_{2j} > X_{2i}) \tag{16}$$

$$\hat{q}_{12} = \frac{1}{n_1\, n_2\, n_3\, n_3'}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k\neq k'=1}^{n_3}\sum^{n_3}\big[\mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \times \mathcal{I}(X_{3k'} > X_{2j} > X_{1i})\big] \tag{17}$$

$$\hat{q}_{13} = \frac{1}{n_1 n_2\, n_2' n_3}\sum_{i=1}^{n_1}\sum_{j\neq j'=1}^{n_2}\sum^{n_2}\sum_{k=1}^{n_3}\big[\mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \times \mathcal{I}(X_{3k} > X_{2j'} > X_{1i})\big] \tag{18}$$

$$\hat{q}_{23} = \frac{1}{n_1\, n_1' n_2 n_3}\sum_{i\neq i'=1}^{n_1}\sum^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\big[\mathcal{I}(X_{1k} > X_{2j} > X_{1i}) \times \mathcal{I}(X_{3k} > X_{2j} > X_{1i'})\big] \tag{19}$$

$$\hat{q}_1 = \frac{1}{n_1 n_2\, n_2' n_3\, n_3'}\sum_{i=1}^{n_1}\sum_{j\neq j'=1}^{n_2}\sum^{n_2}\sum_{k\neq k'=1}^{n_3}\sum^{n_3}[\mathcal{I}(X_{3k}>X_{2j}>X_{1i}) \times \mathcal{I}(X_{3k'}>X_{2j'}>X_{1i})] \tag{20}$$

$$\hat{q}_2 = \frac{1}{n_1\, n_1' n_2 n_3\, n_3'}\sum_{i\neq i'=1}^{n_1}\sum^{n_1}\sum_{j=1}^{n_2}\sum_{k\neq k'=1}^{n_3}\sum^{n_3}\big[\mathcal{I}(X_{3k}>X_{2j}>X_{1i}) \times \mathcal{I}(X_{3k'}>X_{2j}>X_{1i'})\big] \tag{21}$$

$$\hat{q}_3 = \frac{1}{n_1\, n_1' n_2\, n_2' n_3}\sum_{i\neq i'=1}^{n_1}\sum^{n_1}\sum_{j\neq j'=1}^{n_2}\sum^{n_2}\sum_{k=1}^{n_3}\big[\mathcal{I}(X_{3k}>X_{2j}>X_{1i}) \times \mathcal{I}(X_{3k}>X_{2j'}>X_{1i'})\big]. \tag{22}$$

Proof: To show that $\mathbb{E}(\hat{\sigma}_{\hat{\theta}}^2) = \sigma_{\hat{\theta}}^2$, it suffices to evaluate the expectations of the $\hat{q}$-terms in the numerator of (14). It is obvious that

$$\mathbb{E}(\hat{q}_\zeta) = q_\zeta \tag{23}$$

where $\zeta \in \{12, 13, 23, 1, 2, 3\}$ stands for the subscripts of q-terms. Applying the relationship of $\sigma_{\hat{\theta}}^2 = \mathbb{E}(\hat{\theta}^2) - \theta^2$ yields

$$\mathbb{E}(\hat{\theta}^2) = \theta^2 + \sigma_{\hat{\theta}}^2\,. \tag{24}$$

TABLE 1. Quantities needed in the fast algorithm.

Quantities	Numbers of events satisfying the relation inside ( )
S_1	$\mathcal{E}(X_3 > X_2 > X_1)$
S_2	$\mathcal{E}(X_3 > X_3' > X_2 > X_1)$ or $\mathcal{E}(X_3' > X_3 > X_2 > X_1)$
S_3	$\mathcal{E}(X_3 > X_2 > X_2' > X_1)$ or $\mathcal{E}(X_3 > X_2' > X_2 > X_1)$
S_4	$\mathcal{E}(X_3 > X_2 > X_1 > X_1')$ or $\mathcal{E}(X_3 > X_2 > X_1' > X_1)$
S_5	$\mathcal{E}(X_3 > X_3' > X_2 > X_2' > X_1)$ or $\mathcal{E}(X_3' > X_3 > X_2 > X_2' > X_1)$ $\mathcal{E}(X_3 > X_3' > X_2' > X_2 > X_1)$ or $\mathcal{E}(X_3' > X_3 > X_2' > X_2 > X_1)$
S_6	$\mathcal{E}(X_3 > X_2 > X_3' > X_2' > X_1)$ or $\mathcal{E}(X_3' > X_2' > X_3 > X_2 > X_1)$
S_7	$\mathcal{E}(X_3 > X_3' > X_2 > X_1 > X_1')$ or $\mathcal{E}(X_3' > X_3 > X_2 > X_1 > X_1')$ $\mathcal{E}(X_3 > X_3' > X_2 > X_1' > X_1)$ or $\mathcal{E}(X_3' > X_3 > X_2 > X_1' > X_1)$
S_8	$\mathcal{E}(X_3 > X_2 > X_2' > X_1 > X_1')$ or $\mathcal{E}(X_3 > X_2' > X_2 > X_1 > X_1')$ $\mathcal{E}(X_3 > X_2 > X_2' > X_1' > X_1)$ or $\mathcal{E}(X_3 > X_2' > X_2 > X_1' > X_1)$
S_9	$\mathcal{E}(X_3 > X_2 > X_1 > X_2' > X_1')$ or $\mathcal{E}(X_3 > X_2' > X_1' > X_2 > X_1)$

Taking expectation of both sides of (14) and using (24), it follows that

$$\mathbb{E}(\hat{q}_0) = \theta(1-\theta) - \sigma^2_{\hat{\theta}} \tag{25}$$

and

$$\mathbb{E}(\hat{q}_\zeta - \hat{\theta}^2) = q_\zeta - \theta^2 - \sigma^2_{\hat{\theta}} \tag{26}$$

Substituting (25) and (26) into the expectation of (14) along with some straightforward algebra, we have

$$\mathbb{E}(\hat{\sigma}^2_{\hat{\theta}}) = \sigma^2_{\hat{\theta}},$$

and the theorem thus follows. □

IV. EFFICIENT ALGORITHM

A. UNBIASED ESTIMATOR OF $\mathbb{V}(\hat{\theta})$—FAST VERSION

It is noteworthy that, although being unbiased, the naive implementation of the algorithm based on (14)–(22) is computationally very inefficient, especially for large samples, due to the quintic order, i.e., $\mathcal{O}[n_1 n_2 n_3(n_1 n_2 + n_1 n_3 + n_2 n_3)]$ of the time complexity. Fortunately, a linearithmic algorithm is available after rewriting (17)–(22) in terms of S_1–S_9 listed in Table 1. As will be shown later on, these S-terms, which represent the number of events satisfying the relation inside respective brackets, can all be computed by dynamic programming.

Theorem 2: Let $\hat{\theta}$ be defined as in (6) with respect to three i.i.d. samples $\{X_{1i}\}_{i=1}^{n_1}$, $\{X_{2j}\}_{j=1}^{n_2}$, $\{X_{3k}\}_{k=1}^{n_3}$ drawn from three continuous distributions, respectively. Let n_i', $i = 1, 2, 3$ be the same as in Theorem 1. Then the estimator $\sigma^2_{\hat{\theta}}$ in Theorem 1 is equivalent to

$$\begin{aligned}\hat{\sigma}^2_{\hat{\theta}} = \hat{\varsigma}^2_{\hat{\theta}} = \frac{1}{n_1' n_2' n_3'}[&\hat{\theta}(1-\hat{\theta}) \\ &+ n_3'(\hat{\mathcal{Q}}_{12} - \hat{\theta}^2) + n_2'(\hat{\mathcal{Q}}_{13} - \hat{\theta}^2) + n_1'(\hat{\mathcal{Q}}_{23} - \hat{\theta}^2) \\ &+ n_2' n_3'(\hat{\mathcal{Q}}_1 - \hat{\theta}^2) + n_1' n_3'(\hat{\mathcal{Q}}_2 - \hat{\theta}^2) + n_1' n_2'(\hat{\mathcal{Q}}_3 - \hat{\theta}^2)]\end{aligned} \tag{27}$$

where

$$\hat{\theta} = \frac{1}{n_1 n_2 n_3}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\mathcal{I}(X_{3k} > X_{2j} > X_{1i}) = \frac{S_1}{n_1 n_2 n_3} \tag{28}$$

$$\hat{\mathcal{Q}}_{12} = \hat{q}_{12} = \frac{2S_2}{n_1 n_2 n_3 (n_3 - 1)} \tag{29}$$

$$\hat{\mathcal{Q}}_{13} = \hat{q}_{13} = \frac{2S_3}{n_1 n_2 (n_2 - 1) n_3} \tag{30}$$

$$\hat{\mathcal{Q}}_{23} = \hat{q}_{23} = \frac{2S_4}{n_1 (n_1 - 1) n_2 n_3} \tag{31}$$

$$\hat{\mathcal{Q}}_1 = \hat{q}_1 = \frac{4S_5 + 2S_6}{n_1 n_2 (n_2 - 1) n_3 (n_3 - 1)} \tag{32}$$

$$\hat{\mathcal{Q}}_2 = \hat{q}_2 = \frac{4S_7}{n_1 (n_1 - 1) n_2 n_3 (n_3 - 1)} \tag{33}$$

$$\hat{\mathcal{Q}}_3 = \hat{q}_3 = \frac{4S_8 + 2S_9}{n_1 (n_1 - 1) n_2 (n_2 - 1) n_3}. \tag{34}$$

Proof: For compactness, write $n^{[2]} \triangleq n(n-1)$. Then from Table 1, it follows readily that

$$\begin{aligned}\hat{\theta} &= \frac{1}{n_1 n_2 n_3}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \\ &= \frac{\mathcal{E}(X_3 > X_2 > X_1)}{n_1 n_2 n_3} = \frac{S_1}{n_1 n_2 n_3}\end{aligned} \tag{35}$$

$$\begin{aligned}\hat{\mathcal{Q}}_{12} &= \frac{1}{n_1 n_2 n_3^{[2]}}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k \neq k'=1}^{n_3}\sum^{n_3}\mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \\ &\quad \times \mathcal{I}(X_{3k'} > X_{2j} > X_{1i}) \\ &= \frac{1}{n_1 n_2 n_3^{[2]}}[\mathcal{E}(X_3 > X_3' > X_2 > X_1) \\ &\quad + \mathcal{E}(X_3' > X_3 > X_2 > X_1)] \\ &= \frac{2S_2}{n_1 n_2 n_3^{[2]}}\end{aligned} \tag{36}$$

which are the statements in (28) and (29), respectively. The results of (30) and (31) can be verified in a similar way.

For $\hat{Q}_1$, it follows that

$$\begin{aligned}
\hat{Q}_1 &= \frac{1}{n_1\, n_2^{[2]} n_3^{[2]}} \sum_{i=1}^{n_1}\sum_{j\neq j'=1}^{n_2}\sum^{n_2}\sum_{k\neq k'=1}^{n_3}\sum^{n_3} \mathcal{I}(X_{3k} > X_{2j} > X_{1i}) \\
&\quad \times \mathcal{I}(X_{3k'} > X_{2j'} > X_{1i}) \\
&= \frac{1}{n_1\, n_2^{[2]} n_3^{[2]}} [\mathcal{E}(X_3 > X_3' > X_2 > X_2' > X_1) \\
&\quad + \mathcal{E}(X_3 > X_3' > X_2' > X_2 > X_1) \\
&\quad + \mathcal{E}(X_3' > X_3 > X_2 > X_2' > X_1) \\
&\quad + \mathcal{E}(X_3' > X_3 > X_2' > X_2 > X_1) \\
&\quad + \mathcal{E}(X_3 > X_2 > X_3' > X_2' > X_1 \\
&\quad + \mathcal{E}(X_3' > X_2' > X_3 > X_2 > X_1)] \\
&= \frac{4S_5 + 2S_6}{n_1 n_2^{[2]} n_3^{[2]}} \qquad (37)
\end{aligned}$$

which confirms the result in (32). In a similar manner, we also have the results in (33) and (34), respectively. Hence the theorem follows. □

B. EFFICIENT COMPUTATIONS OF S_1 TO S_9 BASED ON DYNAMIC PROGRAMMING

Let $\mathcal{D}_1, \ldots, \mathcal{D}_N, N = n_1 + n_2 + n_3$, be a combined sequence of $X_{11}, \cdots, X_{1n_1}, X_{21}, \cdots, X_{2n_2}$ and $X_{31}, \cdots, X_{3n_3}$. Sorting this $\mathcal{D}$-sequence in ascending order yields the sequence of order statistics [20]–[22]

$$\begin{aligned}
&\underbrace{\mathcal{D}_{(1)} = \cdots = \mathcal{D}_{(1)}}_{\text{Block}_1} < \cdots < \underbrace{\mathcal{D}_{(J)} = \cdots = \mathcal{D}_{(J)}(= D_i)}_{\text{Block}_J} \\
&\qquad < \cdots < \underbrace{\mathcal{D}_{(K)} = \cdots = \mathcal{D}_{(K)}}_{\text{Block}_K}. \qquad (38)
\end{aligned}$$

Suppose that the elements of Block_J are all equal to D_i. Let a_i, b_i, and c_i be the number of X_1's, X_2's and X_3's equaling to $\mathcal{D}_{(i)}$, respectively, for $i = 1, \ldots, K$. Then we can obtain three count vectors $\mathcal{C}_{X_1} \triangleq [a_1 \ \ldots \ a_K]$, $\mathcal{C}_{X_2} \triangleq [b_1 \ \ldots \ b_K]$ and $\mathcal{C}_{X_3} \triangleq [c_1 \ \ldots \ c_K]$, each is based on the $\mathcal{D}_{(i)}$-sequence in Eq. (38), which, as shown in Fig. 3, can be obtained in a linearithmic time, i.e., $\mathcal{O}[(n_1 + n_2 + n_3)\log(n_1 + n_2 + n_3)]$, by using some efficient and popular sorting algorithms in the text book [23]. All the S-terms can be computed via $\mathcal{C}_{X_1}$, $\mathcal{C}_{X_2}$ and $\mathcal{C}_{X_3}$ in linear time $\mathcal{O}(K)$, where $K \leq n_1 + n_2 + n_3$. Next, we will explain the computing structure by investigating the definitions of S_1 and S_8 respectively. The algorithms for the rest terms can be constructed in a similar and straightforward manner, thus omitted for brevity.

We start from the computation of S_1. From Table 1, the definition of S_1 is

$$\begin{aligned}
S_1 &= \mathcal{E}(X_3 > X_2 > X_1) \\
&= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \mathcal{I}(X_{3k} > X_{2j} > X_{1i})
\end{aligned}$$

Algorithm 1: Procedure of Computing Count Vectors

Data: $X_{11}, \cdots, X_{1n_1}, X_{21}, \cdots, X_{2n_2}$ and $X_{31}, \cdots, X_{3n_3}$
Result: count vectors $\mathcal{C}_{X_1}, \mathcal{C}_{X_2}$, and $\mathcal{C}_{X_3}$

1 **begin**
2 $\quad \mathcal{D} \longleftarrow X_{11}, \cdots, X_{1n_1}, X_{21}, \cdots, X_{2n_2}, X_{31}, \cdots, X_{3n_3}$;
3 $\quad \mathcal{W} \longleftarrow$ sorted version of $\mathcal{D}$ in ascending order;
4 $\quad \mathcal{L} \longleftarrow$ labels of elements in $\mathcal{W}$;
5 $\quad \mathcal{W} \longleftarrow \{\mathcal{W}, \mathcal{W}_{n_1+n_2+n_3} + 1\}$;
6 $\quad i \longleftarrow 1$;
7 $\quad k \longleftarrow 1$;
8 $\quad \mathrm{CX}_k \longleftarrow 0$;
9 $\quad \mathrm{CY}_k \longleftarrow 0$;
10 $\quad \mathrm{CZ}_k \longleftarrow 0$;
11 $\quad$ **while** $i \leq n_1 + n_2 + n_3$ **do**
12 $\quad\quad s \longleftarrow i$;
13 $\quad\quad j \longleftarrow s$;
14 $\quad\quad$ **while** $\mathcal{W}_j = \mathcal{W}_s$ **do**
15 $\quad\quad\quad$ **if** $\mathcal{L}_j =$ 'X_1' **then**
16 $\quad\quad\quad\quad \mathrm{CX}_k \longleftarrow \mathrm{CX}_k + 1$;
17 $\quad\quad\quad$ **else if** $\mathcal{L}_j =$ 'X_2' **then**
18 $\quad\quad\quad\quad \mathrm{CY}_k \longleftarrow \mathrm{CY}_k + 1$;
19 $\quad\quad\quad$ **else** $\mathrm{CZ}_k \longleftarrow \mathrm{CZ}_k + 1$;
20 $\quad\quad\quad$;
21 $\quad\quad$ **end**
22 $\quad\quad i \longleftarrow j$;
23 $\quad\quad k \longleftarrow k + 1$;
24 $\quad\quad \mathrm{CX}_k \longleftarrow 0$;
25 $\quad\quad \mathrm{CY}_k \longleftarrow 0$;
26 $\quad\quad \mathrm{CZ}_k \longleftarrow 0$;
27 $\quad$ **end**
28 $\quad K \longleftarrow k - 1$;
29 $\quad \mathcal{C}_{X_1} \longleftarrow \mathrm{CX}_1, \ldots, \mathrm{CX}_K$;
30 $\quad \mathcal{C}_{X_2} \longleftarrow \mathrm{CY}_1, \ldots, \mathrm{CY}_K$;
31 $\quad \mathcal{C}_{X_3} \longleftarrow \mathrm{CZ}_1, \ldots, \mathrm{CZ}_K$;
32 **end**

FIGURE 3. Fast algorithm for computing the three count vectors $\mathcal{C}_{X_1}$, $\mathcal{C}_{X_2}$ and $\mathcal{C}_{X_3}$. In Line 3, $\mathcal{W}$ is the ordered $\mathcal{D}$-sequence in (38); whereas in Line 4, $\mathcal{L}$ contains the labels of $\mathcal{W}_i$, $i = 1, \ldots, n_1 + n_2 + n_3$. Specifically, $\mathcal{L}_i =$ 'X_1' if $\mathcal{W}_i$ comes from X_1-class, and $\mathcal{L}_i =$ 'X_2' and $\mathcal{L}_i =$ 'X_3' if $\mathcal{W}_i$ and $\mathcal{W}_i$ come from X_2-class and X_3-class respectively. Line 5 appends an extra element $\mathcal{W}_{m+n+l} + 1$ $(= \max(\mathcal{Z}) + 1)$ to $\mathcal{W}$ in order to prevent overflow in Line 14. After Lines 11 to 26, we obtain three lists, $\mathrm{CX}_1, \ldots, \mathrm{CX}_k$, $\mathrm{CY}_1, \ldots, \mathrm{CY}_k$ and $\mathrm{CZ}_1, \ldots, \mathrm{CZ}_k$. Finally, in Lines 28 to 30, the extra (last) elements (due to Line 5) in CX-list, CY-list and CZ-list are removed and the rest are stored in $\mathcal{C}_{X_1}$, $\mathcal{C}_{X_2}$ and $\mathcal{C}_{X_3}$, respectively. It is noteworthy that the most time consuming procedure is the sorting operation in Line 3, which can be accomplished by any efficient sorting algorithms, such as the familiar *quick sort* and *merge sort* that are available in the textbook [23].

$$= \sum_{k=3}^{K}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_k b_j a_i. \qquad (39)$$

It follows that (39) can be implemented via a dynamic programming structure. Specifically, we first construct a $3 \times K$ count matrix $\mathbf{C1}$ via stacking $\mathcal{C}_{X_3}$ (Row_1), $\mathcal{C}_{X_2}$ (Row_2) and $\mathcal{C}_{X_1}$ (Row_3) aforementioned. We further set $\mathbf{C1}_{[1,2]}$ and $\mathbf{C1}_{[2,1]}$ to

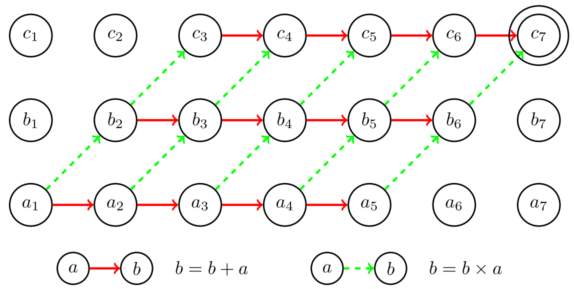


FIGURE 4. Diagram for computing S_1 defined in (39), where $K = 7$ is just for purpose of demonstration.

be 0. Then, as illustrated in Fig. 4, the programming path goes from the southwest corner towards the northeast corner in a linear time $\mathcal{O}(3K)$, with the update rule of

$$\mathbf{C}_{[I,J]} = \begin{cases} \mathbf{C}_{[I,J]} + \mathbf{C}_{[I,J-1]} & I = 3, 2 \le J \le K-2 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 2, 2 \le J \le K-1 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 1, 3 \le J \le K \end{cases} \tag{40}$$

As the indexes I, J running from 3 to 1 and 2 to K respectively, the final desired result is stored in the cell of $\mathbf{C1}_{[1,K]}$.

As to S_8 defined in Table 1, it follows that

$$\begin{aligned} S_8 &= \mathcal{E}(X_3 > X_3' > X_2 > X_1 > X_1') \\ &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\sum_{k=1}^{n_2}\sum_{o=1}^{n_2}\sum_{p=1}^{n_3} \mathcal{I}(X_{3p} > X_{2o} > X_{2k} > X_{1j} > X_{1i}) \\ &= \sum_{p=5}^{K}\sum_{o=4}^{p-1}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_p b_o b_k a_j a_i. \end{aligned} \tag{41}$$

We need to construct a $5 \times K$ count matrix $\mathbf{C8}$, with Row_5 (bottom) being $\mathcal{C}_{X_1}$, Row_4 and Row_3 both being $\mathcal{C}_{X_2}$, and Row_2 and Row_1 (top) both being $\mathcal{C}_{X_3}$. Then, after setting $\mathbf{C8}_{[1,4]}$, $\mathbf{C8}_{[2,3]}$, $\mathbf{C8}_{[3,2]}$, and $\mathbf{C8}_{[4,1]}$ to be 0, the dynamic programming path goes from the southwest corner towards the northeast corner in a linear time $\mathcal{O}(5K)$ (Figure 5), with the update rule of

$$\mathbf{C}_{[I,J]} = \begin{cases} \mathbf{C}_{[I,J]} + \mathbf{C}_{[I,J-1]} & I = 5, 2 \le J \le K-4 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 4, 2 \le J \le K-3 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 3, 3 \le J \le K-2 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 2, 4 \le J \le K-1 \\ \mathbf{C}_{[I,J]} \cdot \mathbf{C}_{[I+1,J-1]} + \mathbf{C}_{[I,J-1]} & I = 1, 5 \le J \le K \end{cases} \tag{42}$$

When the updating finished, the desired value of S_8 in (41) is stored in the cell $\mathbf{C8}_{[1,K]}$.

The rest S-terms can also be computed in a similar way with different count matrices all constructed by $\mathcal{C}_{X_1}$, $\mathcal{C}_{X_2}$ and $\mathcal{C}_{X_3}$. It follows again from Table 1 that S_2–S_7 and S_9 can

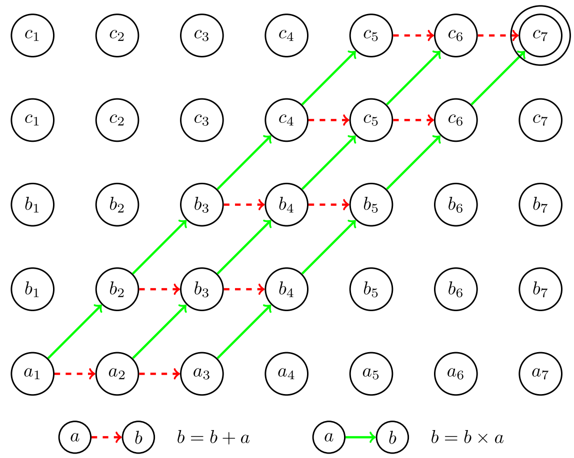


FIGURE 5. Diagram for computing S_8 defined in (41), where $K = 7$ is just for purpose of demonstration.

be expressed as

$$\begin{aligned} S_2 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{o=1}^{n_3} \mathcal{I}(X_{3o} > X_{3k} > X_{2j} > X_{1i}) \\ &= \sum_{o=4}^{K}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_o c_k b_j a_i \end{aligned} \tag{43}$$

$$\begin{aligned} S_3 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_2}\sum_{o=1}^{n_3} \mathcal{I}(X_{3o} > X_{2k} > X_{2j} > X_{1i}) \\ &= \sum_{o=4}^{K}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_o b_k b_j a_i \end{aligned} \tag{44}$$

$$\begin{aligned} S_4 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\sum_{k=1}^{n_2}\sum_{o=1}^{n_3} \mathcal{I}(X_{3o} > X_{2k} > X_{1j} > X_{1i}) \\ &= \sum_{o=4}^{K}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_o b_k a_j a_i \end{aligned} \tag{45}$$

$$\begin{aligned} S_5 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_2}\sum_{o=1}^{n_3}\sum_{p=1}^{n_3} \mathcal{I}(X_{3p} > X_{3o} > X_{2k} > X_{2j} > X_{1i}) \\ &= \sum_{p=5}^{K}\sum_{o=4}^{p-1}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_p c_o b_k b_j a_i \end{aligned} \tag{46}$$

$$\begin{aligned} S_6 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{o=1}^{n_2}\sum_{p=1}^{n_3} \mathcal{I}(X_{3p} > X_{2o} > X_{3k} > X_{2j} > X_{1i}) \\ &= \sum_{p=5}^{K}\sum_{o=4}^{p-1}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_p b_o c_k b_j a_i \end{aligned} \tag{47}$$

$$\begin{aligned} S_7 &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\sum_{k=1}^{n_2}\sum_{o=1}^{n_3}\sum_{p=1}^{n_3} \mathcal{I}(X_{3p} > X_{3o} > X_{2k} > X_{1j} > X_{1i}) \\ &= \sum_{p=5}^{K}\sum_{o=4}^{p-1}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_p c_o b_k a_j a_i \end{aligned} \tag{48}$$

$$S_9 = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_1}\sum_{o=1}^{n_2}\sum_{p=1}^{n_3}\mathcal{I}(X_{3p}>X_{2o}>X_{1k}>X_{2j}>X_{1i})$$
$$= \sum_{p=5}^{K}\sum_{o=4}^{p-1}\sum_{k=3}^{o-1}\sum_{j=2}^{k-1}\sum_{i=1}^{j-1} c_p b_o a_k b_j a_i. \quad (49)$$

Given (39)–(49) that can all be calculated by dynamic programming, the slow version (14) of quintic time can thus be converted into the fast version (27) with a linearithmic time.

V. NUMERICAL RESULTS

This section purports to verify the computational efficiency as well as the unbiasedness of the dynamic programming based algorithm (Theorem 2), denoted by $\mathbb{V}_{DP}$ in the sequel. The state-of-the-art algorithm developed by Waegeman *et al.* [9] is to be denoted by $\mathbb{V}_{WBB}$. Throughout this section, Monte Carlo simulations are undertaken for sample sizes from 10 to 200 with an increment of 10. The number of trials is set to be 10^6 for accuracy. All samples of random variables following various distributions are generated by functions in Matlab `Statistics Toolbox`™.

A. COMPARISON OF COMPUTATIONAL LOADS

To illustrate the computational efficiency of our proposed algorithm, we generate three one-dimensional normal samples, with $\{X_{1i}\}_{i=1}^{n_1}$ following the normal distribution $\mathcal{N}(0, 0.6)$, $\{X_{2j}\}_{j=1}^{n_2}$ following $\mathcal{N}(1, 0.4)$, and $\{X_{3k}\}_{k=1}^{n_3}$ following $\mathcal{N}(2, 0.2)$, respectively. Here the notation $\mathcal{N}(\mu, \sigma^2)$ stands for a normal distribution with mean μ and variance σ^2. Since the parameters have little effect on the computational speed, they are chosen rather arbitrarily. Based on the analysis of Waegeman *et al.* [9], it follows that the computational complexity of $\mathbb{V}_{WBB}$ is $\mathcal{O}(8[n_1 + n_2 + n_3]^2)$. On the other hand, from Theorem 2, the computational complexity of our algorithm (2) is dominated by the procedure of attaining $\mathcal{C}_{X_1}$, $\mathcal{C}_{X_2}$ and $\mathcal{C}_{X_3}$, whose time complexity is linearithmic, i.e., $\mathcal{O}[(n_1+n_2+n_3)\log(n_1+n_2+n_3)]$. Since each dynamic programming procedure for S-terms is in linear time, the overall time complexity of our algorithm is $\mathcal{O}[(n_1+n_2+n_3)\log(n_1+n_2+n_3)]$. As shown in Fig. 6, the linearithmic algorithm $\mathbb{V}_{DP}$ proposed in this work does outperform $\mathbb{V}_{WBB}$ in terms of computational efficiency.

B. COMPARISON OF UNBIASEDNESS

In this subsection, we verify the unbiasedness of our algorithm in (27) with samples drawn from four continuous distributions, including normal, uniform, Laplace, and Rayleigh distributions. The empirical variance of VUS calculated from 10^6 Monte Carlo trials is considered to be the ground truth and denoted by $\mathbb{V}_{Emp}$. It is observed from Figs. 7 to 10 that

1) the results of our algorithm $\mathbb{V}_{DP}$ agree well with those of $\mathbb{V}_{Emp}$, demonstrating its unbiasedness;
2) the results of $\mathbb{V}_{WBB}$ deviate from those of $\mathbb{V}_{Emp}$, especially for small samples;

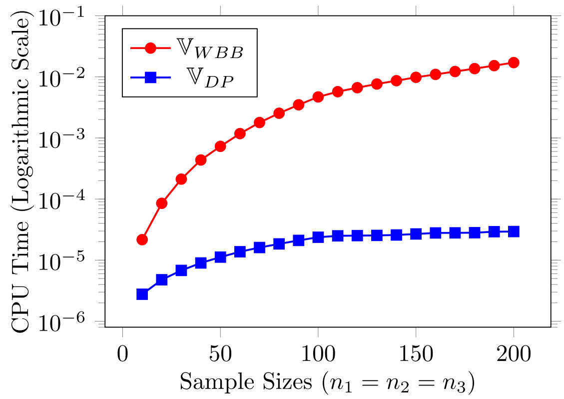


FIGURE 6. The contrast of computational speeds between the two algorithms, i.e., $\mathbb{V}_{DP}$ and $\mathbb{V}_{WBB}$. For simplicity, here the sample sizes are set to be equal, namely, $n_1 = n_2 = n_3 = 10(10)200$. All samples are drawn from normal distributions with arbitrary parameters since they have little effect on the computational speed comparison. A logarithmic scale is used for better visual effect.

3) with increase of sample size, the deviation of $\mathbb{V}_{WBB}$ to $\mathbb{V}_{Emp}$ is becoming less and less noticeable, suggesting that $\mathbb{V}_{WBB}$ is only asymptotically unbiased.

VI. DISCUSSION

A. EXTENSION TO K-CLASS CASES–UNBIASED ESTIMATOR

Thus far we have developed an unbiased estimator of $\mathbb{V}(\hat{\theta})$ which can be computed in linearithmic time. In this section, we discuss the scalability of our method to more general cases, that is when the class number $k > 3$. we first develop an unbiased estimator, base on the formulas of Nakas *et al.* [6], as

$$\sigma^2_{\hat{\theta}} = \frac{1}{n_1\cdots n_k}\Big[\theta(1-\theta) + \sum_{i=1}^{k}(n_i-1)(q_i-\theta)$$
$$+ \sum_{i_1\neq i_2=1}^{k}\sum^{k}(n_{i_1}-1)(n_{i_2}-1)(q_{i_1i_2}-\theta^2)$$
$$+ \cdots + \sum_{i_1\neq\cdots\neq i_{k-1}=1}^{k}\cdots\sum^{k}(n_{i_1}-1)\cdots(n_{i_{k-1}}-1)$$
$$\times (q_{i_1i_2\cdots i_{k-1}}-\theta^2)\Big] \quad (50)$$

where

$$\theta = \mathbb{P}(X_k > \cdots > X_1) \quad (51)$$
$$q_i = \mathbb{P}(X_k > \cdots > X_i > \cdots > X_1 \cap X_k > \cdots > X_i' > \cdots > X_1) \quad (52)$$
$$q_{i_1i_2} = \mathbb{P}(X_k>\cdots>X_{i_1}>\cdots>X_{i_2}>\cdots>X_1 \cap X_k>\cdots>X_{i_1}'>\cdots>X_{i_2}' > \cdots>X_1)$$
$$\vdots \quad (53)$$

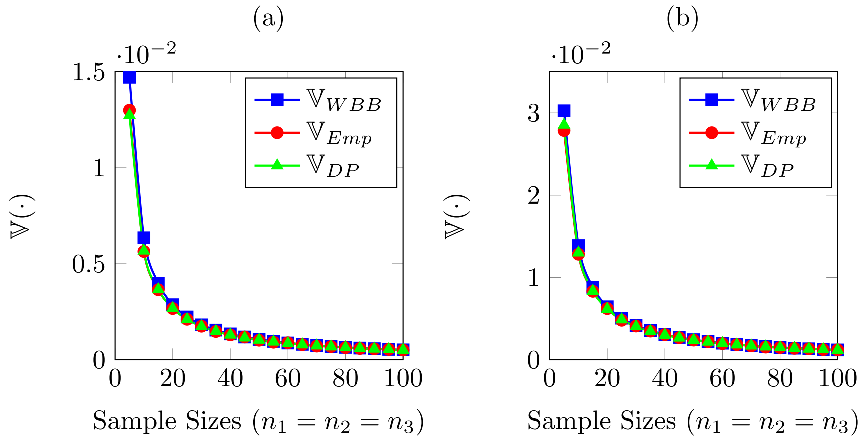


FIGURE 7. **Verification of unbiasedness of the estimator in (27) with normal distribution $\mathcal{N}(\mu, \sigma^2)$. For simplicity, here the sample sizes are set to be equal, namely, $n_1 = n_2 = n_3 = 10(10)100$. (a) Null case under normal distribution, where both X_1, X_2 and X_3 are following $\mathcal{N}(0, 1)$. (b) Non-null case under normal distribution, where X_1 follows $\mathcal{N}(0, 1)$, X_2 follows $\mathcal{N}(1, 1)$, and X_3 follows $\mathcal{N}(2, 1)$, respectively.**

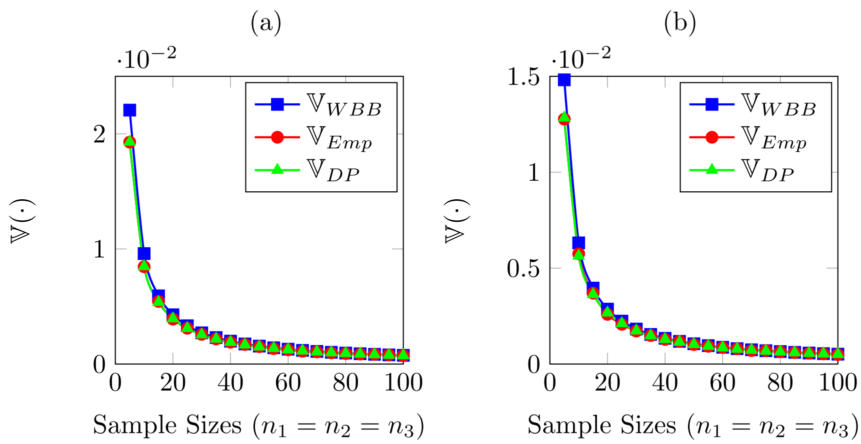


FIGURE 8. **Verification of unbiasedness of the estimator in (27) with uniform distribution $\mathcal{U}(a, b)$. (a) Null case under uniform distribution, where both X_1, X_2 and X_3 are following $\mathcal{U}(0, 2)$. For simplicity, here the sample sizes are set to be equal, namely, $n_1 = n_2 = n_3 = 10(10)100$. (b) Non-null case under uniform distribution, where X_1 follows $\mathcal{U}(0, 2)$, X_2 follows $\mathcal{U}(1, 3)$, and X_3 follows $\mathcal{U}(2, 4)$, respectively.**

with X' being an i.i.d. copy of X. Given (50)–(53) above, an unbiased estimator can be obtained, as stated in Theorem 3 below.

Theorem 3: Let $\{X_{1i_1}\}_{i_1=1}^{n_1}$, $\{X_{2i_2}\}_{i_2=1}^{n_2}$, ..., $\{X_{ki_k}\}_{i_k=1}^{n_k}$ be k i.i.d. samples drawn respectively from k continuous populations. Then $\mathbb{E}(\hat{\sigma}^2_{\hat{\theta}}) = \sigma^2_{\hat{\theta}}$, with $\hat{\sigma}^2_{\hat{\theta}}$ being

$$\begin{aligned}\hat{\sigma}^2_{\hat{\theta}} = \frac{1}{(n_1-1)\cdots(n_k-1)}\Big[&\hat{\theta}(1-\hat{\theta}) + \sum_{i=1}^{k}(n_i-1)(\hat{q}_i-\hat{\theta}) \\ &+ \sum_{i_1\neq i_2=1}^{k}\sum^{k}(n_{i_1}-1)(n_{i_2}-1)(\hat{q}_{i_1i_2}-\hat{\theta}^2) \\ &+\cdots+\sum_{i_1\neq\cdots\neq i_{k-1}=1}^{k}\cdots\sum^{k}(n_{i_1}-1)\cdots(n_{i_{k-1}}-1)(\hat{q}_{i_1i_2\cdots i_{k-1}}-\hat{\theta}^2)\Big]\end{aligned} \tag{54}$$

where

$$\hat{\theta} = \frac{1}{n_1\ldots n_k}\sum_{j_1=1}^{n_1}\cdots\sum_{j_k=1}^{n_k}\mathcal{I}(X_{kj_k} > \cdots > X_{1j_1}) \tag{55}$$

$$\begin{aligned}\hat{q}_i = \frac{1}{n_1\cdots n_i(n_i-1)\cdots n_k}&\sum_{j_1=1}^{n_1}\cdots\sum_{j_\iota\neq j_i=1}^{n_i}\sum^{n_i}\cdots\sum_{j_k=1}^{n_k} \\ &\times\big[\mathcal{I}(X_{kj_k} > \cdots > X_{ij_i} > \cdots > X_{1j_1}) \\ &\times\mathcal{I}(X_{kj_k} > \cdots > X_{ij_\iota} > \cdots > X_{1j_1})\big]\end{aligned} \tag{56}$$

and so on.

Proof: Based on the relationship $\mathbb{V}(\hat{\theta}) = \mathbb{E}(\hat{\theta}^2) - \theta^2$, we have

$$\mathbb{E}(\hat{\theta}^2) = \theta^2 + \sigma^2_{\hat{\theta}}. \tag{57}$$

Moreover, due to the i.i.d. assumption, it follows readily that all $\hat{q}$-terms above are unbiased estimates of the corresponding

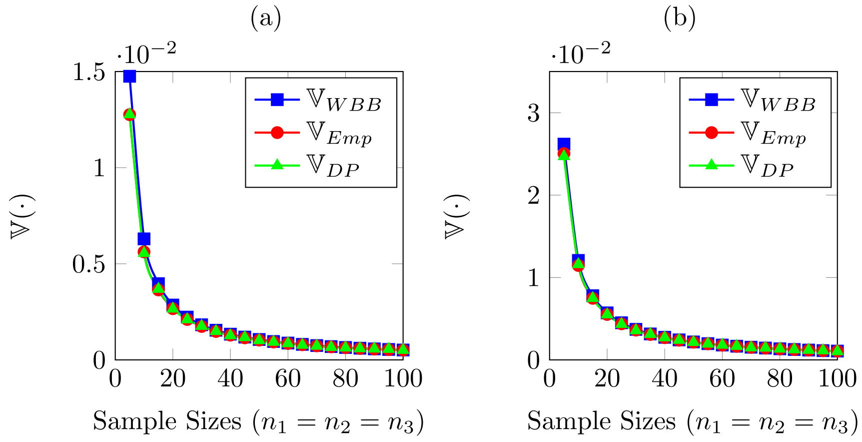


FIGURE 9. Verification of unbiasedness of the estimator in (27) with Laplace distribution $\mathcal{L}(\mu, b)$. (a) Null case under uniform distribution, where both X_1, X_2 and X_3 are following $\mathcal{L}(0, 1)$. For simplicity, here the sample sizes are set to be equal, namely, $n_1 = n_2 = n_3 = 10(10)100$. (b) Non-null case under uniform distribution, where X_1 follows $\mathcal{L}(0, 1)$, X_2 follows $\mathcal{L}(1, 2)$, and X_3 follows $\mathcal{L}(2, 3)$, respectively.

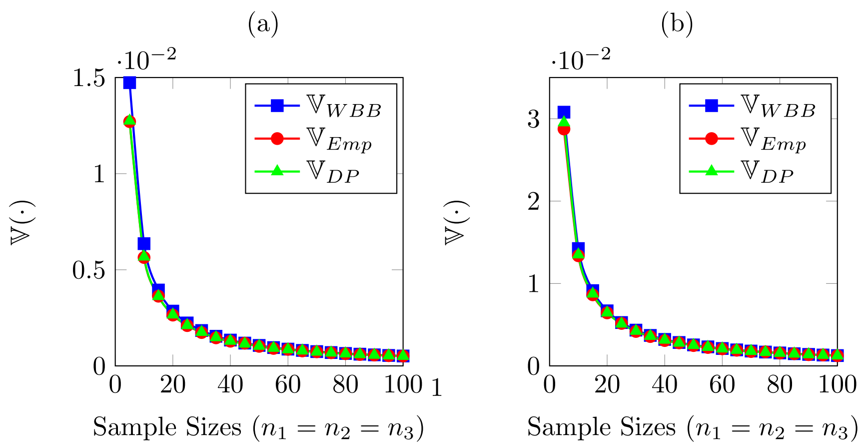


FIGURE 10. Verification of unbiasedness of the estimator in (27) with Rayleigh distribution $\mathcal{R}(\sigma^2)$. (a) Null case under uniform distribution, where both X_1, X_2 and X_3 are following $\mathcal{R}(1)$. For simplicity, here the sample sizes are set to be equal, namely, $n_1 = n_2 = n_3 = 10(10)100$. (b) Non-null case under uniform distribution, where X_1 follows $\mathcal{R}(1)$, X_2 follows $\mathcal{R}(2)$, and X_3 follows $\mathcal{R}(3)$, respectively.

q-terms in (50)–(53), i.e.,

$$\mathbb{E}(\hat{q}_i) = q_i \tag{58}$$
$$\mathbb{E}(\hat{q}_{i_1 i_2}) = q_{i_1 i_2} \tag{59}$$
$$\vdots$$
$$\mathbb{E}(\hat{q}_{i_1 \cdots i_{k-1}}) = q_{i_1 \cdots i_{k-1}}. \tag{60}$$

Taking expectation of both sides of (54) and substituting (50), (57)–(60) thereafter along with some tedious but straightforward algebra, we arrive at

$$\mathbb{E}(\hat{\sigma}^2_{\hat{\theta}}) = \frac{\sigma^2_{\hat{\theta}}}{(n_1 - 1)\cdots(n_k - 1)}\Big[-1 - \sum_{i=1}^{k}(n_i - 1) - \sum_{i_1 \neq i_2 = 1}^{k}\sum^{k}(n_{i_1} - 1)(n_{i_2} - 1) - \cdots - \sum_{i_1 \neq \cdots \neq i_{k-1} = 1}^{k} \cdots \sum^{k}(n_{i_1} - 1)\cdots(n_{i_{k-1}} - 1) + n_1 \cdots n_k\Big]. \tag{61}$$

For convenience, denote by T_k the quantity inside the brackets on the right side of (61). Then we only need to show that

$$T_k = (n_1 - 1)\cdots(n_k - 1) \tag{62}$$

which can be proved by mathematical induction, as follows.

- Step 1: For $k = 2$, it is easy to verify that (62) holds true, since
$$\begin{aligned} T_2 &= -1 - [(n_1 - 1) + (n_2 - 1)] + n_1 n_2 \\ &= (n_1 - 1)(n_2 - 1). \end{aligned} \tag{63}$$
- Step 2: Assume that for $k = r$, both sides of (62) are equal, that is,
$$T_r = (n_1 - 1)\cdots(n_r - 1). \tag{64}$$

- Step 3: When $k = r + 1$, we have, from (61),

$$\begin{aligned}
T_{r+1} = & T_r - (n_{r+1} - 1) - . \sum_{i=1}^{r} (n_{r+1} - 1)(n_i - 1) \\
& - - \sum_{i_1}^{r} \cdots \sum_{i_{r-2}}^{r} (n_{r+1}-1)(n_{i_1}-1) \cdots (n_{i_{r-2}}-1) \\
& + n_1 \cdots n_r (n_{r+1} - 1) \\
& - \sum_{i_1}^{r+1} \cdots \sum_{i_r}^{r+1} (n_{i_1} - 1) \cdots (n_{i_r} - 1) \\
= & T_r + (n_1 - 1) \cdots (n_r - 1)(n_{r+1} - 1) \\
& -(n_1 - 1) \cdots (n_r - 1)
\end{aligned} \tag{65}$$

which becomes

$$T_{r+1} = (n_1 - 1) \cdots (n_{r+1} - 1) \tag{66}$$

upon substitution of (64) into the last step in (65).

And the theorem thus follows. □

B. EXTENSION TO K-CLASS CASES—TIME COMPLEXITY ANALYSIS

It follows from Theorem 2 that the proposed algorithm for three-class problem is in linearithmic time, that is, the time complexity is of order $\mathcal{O}(N \log N + 9 \times 5N)$, where N is the sum of sample sizes. For general k-class problems ($k > 3$), the time complexity can be expresses as

$$\mathcal{O}[N \log N + \lambda(k)N] \tag{67}$$

where $\lambda(k)$ grows exponentially with increase of k. On the other hand, the algorithm proposed by Waegeman *et al.* [9], has a time complexity of

$$\mathcal{O}(2^k N^2). \tag{68}$$

It follows from a comparison of (67) and (68) that for small k, the dynamic programming based algorithm runs faster than the algorithm of Waegeman *et al.* [9]; whereas for large k, the former might underperform the latter. The determination of the breaking point of k is beyond the scope of this paper and will be addressed in our future work.

VII. CONCLUSION

In this paper, we have proposed an efficient dynamic-programming based algorithm for unbiased estimation of the variance of VUS. Theoretical and empirical results suggest that (a) our algorithm is in linearithmic time, faster than the state-of-the-art method developed by Waegeman *et al*, which is in quadratic time; and (b) our estimator is unbiased, compared with Waegeman's method, which is only asymptotically unbiased. Besides these advantages, the structure of our algorithm can be easily extended to multi-class cases (the number of class is greater than three) based on the results in [6]. Moreover, the dynamic programming structure can be implemented by VLSI circuits [24], [25], which means that the computational speed could be further accelerated. The methodology established in this work is believed to shed new light on the topic of ROC analysis, which is an indispensable tool in many scientific and engineering areas.

REFERENCES

[1] K. H. Zou, A. J. O'Malley, and L. Mauri, "Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models," *Circulation*, vol. 115, no. 5, pp. 654–657, Feb. 2007.

[2] W. Peterson, T. Birdsall, and W. Fox, "The theory of signal detectability," *Trans. IRE Prof. Group Inf. Theory*, vol. 4, no. 4, pp. 171–212, Sep. 1954.

[3] M. J. Zolghadri and E. G. Mansoori, "Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis," *Inf. Sci.*, vol. 177, no. 11, pp. 2296–2307, Jun. 2007.

[4] C. Catal, O. Alan, and K. Balkan, "Class noise detection based on software metrics and ROC curves," *Inf. Sci.*, vol. 181, no. 21, pp. 4867–4877, Nov. 2011.

[5] B. K. Scurfield, "Multiple-event forced-choice tasks in the theory of signal detectability," *J. Math. Psychol.*, vol. 40, no. 3, pp. 253–269, Sep. 1996.

[6] C. T. Nakas and C. T. Yiannoutsos, "Ordered multiple-class ROC analysis with continuous measurements," *Statist. Med.*, vol. 23, no. 22, pp. 3437–3449, Nov. 2004.

[7] L. Kang and L. Tian, "Estimation of the volume under the ROC surface with three ordinal diagnostic categories," *Comput. Statist. Data Anal.*, vol. 62, pp. 39–51, Jun. 2013.

[8] M. P. Sampat *et al.*, "Indexes for three-class classification performance assessment—An empirical comparison," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 3, pp. 300–312, May 2009.

[9] W. Waegeman, B. de Baets, and L. Boullart, "On the scalability of ordered multi-class ROC analysis," *Comput. Statist. Data Anal.*, vol. 52, no. 7, pp. 3371–3388, Mar. 2008.

[10] D. Mossman, "Three-way ROC," *Med. Decis. Making*, vol. 19, no. 1, pp. 78–89, Jan. 1999.

[11] X. He and E. C. Frey, "The meaning and use of the volume under a three-class ROC surface (VUS)," *IEEE Trans. Med. Imag.*, vol. 27, no. 5, pp. 577–588, May 2008.

[12] S. Dreiseitl, L. Ohno-Machado, and M. Binder, "Comparing three-class diagnostic tests by three-way ROC analysis," *Med. Decis. Making*, vol. 20, no. 3, pp. 323–331, Jul. 2000.

[13] W. Xu, S. Liu, X. Sun, S. Liu, and Y. Zhang, "A fast algorithm for unbiased estimation of variance of AUC based on dynamic programming," *IEEE Access*, vol. 4, pp. 9553–9560, 2016.

[14] R. Bellman, *Dynamic Programming*. Chelmsford, MA, USA: Courier Corporation, 2013.

[15] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2005, vol. 1, no. 3.

[16] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[17] J. A. Ayala-Romero, J. J. Alcaraz, and J. Vales-Alonso, "Energy saving and interference coordination in HetNets using dynamic programming and CEC," *IEEE Access*, vol. 6, pp. 71110–71121, 2018.

[18] W. Wang, Z. Zhang, J. Shi, C. Lin, and Y. Gao, "Optimization of a dual-motor coupled powertrain energy management strategy for a battery electric bus based on dynamic programming method," *IEEE Access*, vol. 6, pp. 32899–32909, 2018.

[19] W. Xu, P. Chen, Y. Hung, and S. Kung, "A novel three-class ROC method for eQTL analysis," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 6., Jul. 2010, pp. 3056–3061.

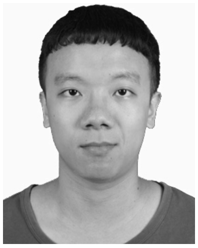
[20] W. Xu, C. Chang, Y. S. Hung, S. K. Kwan, and P. C. W. Fung, "Order statistics correlation coefficient as a novel association measurement with applications to biosignal analysis," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5552–5563, Dec. 2007.

[21] W. Xu, J. Dai, Y. Hung, and Q. Wang, "Estimating the area under a receiver operating characteristic (ROC) curve: Parametric and nonparametric ways," *Signal Process.*, vol. 93, no. 11, pp. 3111–3123, Nov. 2013.

[22] W. Xu, Y. Hung, M. Niranjan, and M. Shen, "Asymptotic mean and variance of Gini correlation for bivariate normal samples," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 522–534, Feb. 2010.

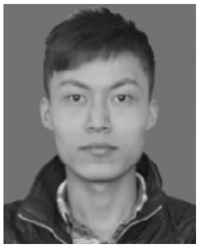
[23] U. Manber, *Introduction to Algorithms: A Creative Approach*. Reading, MA, USA: Addison-Wesley, 1989, vol. 4.

[24] P. K. Tripathy, R. K. Dash, and C. R. Tripathy, "A dynamic programming approach for layout optimization of interconnection networks," *Eng. Sci. Technol., Int. J.*, vol. 18, no. 3, pp. 374–384, Sep. 2015.
[25] M. Shupletsov and M. Avtaikina, "Dynamic programming algorithms for large-scale equivalence checking and functional correction," in *Proc. IEEE Conf. Russian Young Researchers Elect. Electron. Eng.*, Feb. 2017, pp. 1032–1035.



SHUN LIU received the B.Eng. degree from the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include statistical signal processing, pattern recognition, and machine learning.

WEICHAO XU (M'06) received the B.Eng. and M.Eng. degrees in electrical engineering from the University of Science and Technology of China, Hefei, in 1993 and 1996, respectively, and the Ph.D. degree in biomedical engineering from The University of Hong Kong, Hong Kong, in 2002, where he was a Research Associate with the Department of Electrical and Electronic Engineering, from 2002 to 2010. In 2011, he entered the 100-Talent Scheme in the Guangdong University of Technology, where he is currently a Professor. His research interests include the areas of mathematical statistics, computational statistics, information theory, machine learning, optimization theory, and digital signal processing and its applications.



XU SUN received the B.Eng. and M.Eng. degrees from the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2013 and 2016, respectively. He is currently a Senior Engineer with Baidu, Inc. His research interests include statistical signal processing, statistical learning theory, and intelligent systems.

YUN ZHANG received the B.S. and M.S. degrees in automatic engineering from Hunan University, Changsha, China, in 1982 and 1986, respectively, and the Ph.D. degree in automatic engineering from the South China University of Science and Technology, Guangzhou, China, in 1998. He is currently a Professor with the School of Automation, Guangdong University of Technology, Guangzhou. His research interests include intelligent control systems, network systems, and signal processing.