Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

VARIANCE AND DERIVATIVE ESTIMATION FOR VIRTUAL PERFORMANCE IN SIMULATION ANALYTICS

Yujing Lin
Barry L. Nelson

Department of Industrial Engineering & Management Sciences
Northwestern University
Evanston, IL 60208, USA

ABSTRACT

Virtual performance is a class of time-dependent performance measures conditional on a particular event occurring at timet_0 for a (possibly) nonstationary stochastic process; virtual waiting time of a customer arriving to a queue at time t_0 is one example. Virtual statistics are estimators of the virtual performance. In this paper, we go beyond the mean to propose estimators for the variance, and for the derivative of the mean with respect to time, of virtual performance, examining both their small-sample and asymptotic properties. We also provide a modified K-fold cross validation method for tuning the parameter k for the difference-based variance estimator, and evaluate the performance of both variance and derivative estimators via controlled studies. The variance and derivative provide useful information that is not apparent in the mean of virtual performance.

1 INTRODUCTION

"Virtual statistics," as we define them, are estimators for performance measures that are conditional on the occurrence of an event at a particular time, say t_0. The class of measures of interest we call virtual performance at time t_0, denoted by $V(t_0)$. Lin and Nelson (2016) and Lin et al. (2017) focus on estimating the mean of some time-dependent virtual performance, denoted by $v(t_0) = \mathrm{E}[V(t_0)]$, for a (possibly) nonstationary stochastic process using the output of computer simulation, and they propose a k-nearest-neighbors ($k$nn) estimator of it. In this paper, we extend the study of mean estimation to variance and derivative estimation for $V(t_0)$.

To motivate the study of virtual statistics, we consider a specific example of virtual waiting time described in Smith and Nelson (2015). Smith and Nelson (2015) consider the case of a traveler who wants to know how long it might take them to clear security if they arrive to the airport at, say, 7:30 AM. The $k$nn estimator proposed by Lin and Nelson (2016) estimates the mean of the virtual waiting time for the traveler. Of course, the traveler is unlikely to experience exactly the mean of the virtual waiting time. The standard deviation of the virtual waiting time provides a more complete description of the distribution of their delay. When the mean of virtual performance (i.e., the true regression function) is unknown, a typical approach is to first estimate the regression function, and then derive the response variance from the residuals; this is also called *residual variance estimation*. There exists substantial research on residual variance estimation when the true regression function is unknown. For example, Liitiäinen et al. (2010) describe a residual variance estimator using nearest neighbor statistics, and Liitiäinen et al. (2008) study variance estimation for a general setting that covers non-additive heteroscedastic noise under non-iid sampling.

Residual variance estimation requires estimating the unknown regression function first and then computing the sample variance based on the estimated regression function. If simulation users are only interested in the variance of virtual performance, then there is another class of variance estimator, called *difference-based* variance estimator, that does not require estimation of the unknown regression function. Rice (1984) presents a difference-based variance estimator for a fixed design. Gasser et al. (1986) provide a variation of difference-based variance estimator by introducing the concept of pseudo-residuals. *In this paper, we apply both residual variance and difference-based variance estimation schemes to the virtual statistics problem.*

Another performance measure of interest is the derivative of $v(t_0)$. The derivative information is quite useful since it reveals how the system will respond to a change in the time that the trigger event occurs. Take the airport check-in problem as an example. If the traveler arrives at the airport slightly earlier or later than their planned arrival time t_0, then the traveler probably wants to know whether or not this change would lead to a much longer expected waiting time; that is, is $v(t_0)$ changing rapidly? Additionally, simulation users can obtain some idea on how often they should estimate $v(t_0)$ from the derivative information. For example, if the derivative of $v(t_0)$ is close to 0 at some time $t = t_0$, then it is not necessary to estimate the mean of virtual performance at times close to t_0 because we know $v(t)$ changes very slowly.

The finite difference (FD) method has been widely used for derivative estimation in simulation. Although FD is well known, we show later why it is incompatible with a nonparametric *k*nn approach. In addition to FD, there are many other types of derivative estimation approaches. One of them is similar to the idea of the residual variance estimation scheme; that is, one should estimate the unknown regression function first by using some smooth functions such as polynomials or splines and then compute the estimator by taking the derivative of the estimated regression function with respect to time. For example, Zhou and Wolfe (2000) study the estimation of derivatives using spline estimators. Gasser and Müller (1979) and Gasser and Müller (1984) describe kernel-based derivative estimators. A more recent derivative estimation method is based on weighted slopes of symmetric observations around the time $t = t_0$ of interest. De Brabanter et al. (2013) and De Brabanter and Liu (2015) study this type of estimator and show its asymptotic properties. Although all of these approaches can apply to virtual performance settings, we focus on the weighted-slopes type of derivative estimator because it can be treated as an extension of our existing *k*nn mean estimation results; see Lin and Nelson (2016).

The remainder of this paper is organized as follows. We start with a summary of work on mean estimation for virtual performance in Section 2, which includes important assumptions and results from Lin et al. (2017). In Section 3, we formally define our variance and derivative estimators for virtual performance. The asymptotic properties of the proposed estimators under specific conditions on the system of interest and the growth rate of the tuning parameter k are offered in Section 4. We introduce a modified K-fold cross validation method for tuning the parameter of the difference-based variance estimator in Section 5. To evaluate the performance of the proposed variance and derivative estimators, we apply our method to controlled studies in Section 6, comparing the estimators with the true variance and derivative of virtual performance. Some conclusions are provided in Section 7.

2 *k*NN METHOD FOR THE MEAN

We first present the definition of virtual performance given in Lin et al. (2017). Consider a stochastic point process that begins at time $T_{\text{start}} \equiv 0$ and ends at time $T_{\text{end}} \equiv T$ where $\mathrm{E}(T^2) < \infty$. The random event times are $0 \le t_1 < t_2 < \cdots < t_M \le T$; in the simulation setting these will typically be the times that a common type of event occurs, such as "customer arrival" or "machine failure," although that is not essential. We will call all of these events "arrivals" later even though they may not be. Associated with event time t_i is a random performance variable $Y(t_i)$; in the simulation setting this might be the waiting time for a customer who arrives at time t_i, or the time until the system is restored after a failure that happens at time t_i. Thus, $\{(t_i, Y(t_i));\ i = 1,2,\ldots,M\}$ is a *marked point process* (typically with a complicated joint distribution).

For a fixed time $0 \le t_0 \le T$, let $i_0 = \mathrm{argmin}_i |t_i - t_0|$ (we handle the case of $M = 0$ events below). Then we define the *virtual performance at* t_0 to be $V(t_0) \stackrel{\mathscr{D}}{=} Y(t_{i_0}) \mid t_{i_0} = t_0$ and its mean to be $v(t_0) = \mathrm{E}(V(t_0))$. If $M = 0$, then define $Y(t_{i_0}) = 0$, and let $t_{i_0} = 0$ if $t_0 > T/2$, and $t_{i_0} = T$, otherwise.

Lin et al. (2017) propose a *k*nn method for estimating $v(t_0)$ from n independent simulation replications and provide two approaches for measuring the error of the *k*nn mean estimator. Therefore, the simulation data are $\{(t_{ij}, Y(t_{ij}));\ i = 1,2,\ldots,M_j\}$, $j = 1,2,\ldots,n$, where the subscript j denotes the jth replication. We assume that $\mathrm{E}[Y^2(t_{ij})] < \infty$ for all t_{ij}. For notational simplicity, we refer to this assumption as $\mathrm{E}[Y^2(t)] < \infty$ from here on. In this paper, we focus on the same type of stochastic point process but will study different virtual statistics. The development is based on some important results from Lin et al. (2017). Therefore, we restate the relevant assumptions and results in this section.

Denote the superposed process of all the observed arrivals by $\mathscr{T}_n = \left\{t_{ij} \colon i = 1,2,\ldots,M_j, j = 1,2,\ldots,n\right\}$. The $k$nn estimator of $v(t_0)$, $\bar{V}(t_0)$, proposed by Lin et al. (2017) is

$$\bar{V}(t_0) = \frac{1}{k}\sum_{\ell=1}^{k} Y(t_0^{(\ell,n)}), \quad t_0^{(\ell,n)} \in \mathscr{T}_n, \tag{1}$$

where $t_0^{(1,n)} < t_0^{(2,n)} < \cdots < t_0^{(k,n)}$ are the *sorted* k nearest neighbors to t_0 from the superposed process $\mathscr{T}_n$, and $Y(t_0^{(\ell,n)})$ is the corresponding observed output for $\ell = 1,2,\ldots,k$. Notice that the "closeness" here is based on $|t_0^{(\ell,n)} - t_0|$ regardless of replication and ties are broken arbitrarily.

The system of interest analyzed in this paper satisfies the same properties assumed in Lin et al. (2017). Let the arrival-counting process associated with t_{ij} from a generic replication of the dynamic system to be denoted by $\{N(t)\colon t \geq 0\}$. For any time interval $(t - w/2, t + w/2]$ with $w > 0$, let the number of arrivals within $(t-w/2, t+w/2]$ to be denoted by $N^w(t) = N(t+w/2) - N(t-w/2)$. If t_0 is very close to the endpoint 0, then $t - w/2$ might be negative so that $N(t-w/2)$ is not defined. A similar issue occurs for t_0 that is close to T. Thus, we further define $N(t) = N(0)$ for $t \leq 0$, and $N(t) = N(T)$ for $t \geq T$. For each replication, suppose $\{N(t)\colon t \geq 0\}$ satisfies the following properties for all $t \in [0,T]$:

$$\Pr\{N^w(t) \geq 1\} = \lambda_t w + o(w) \quad \text{and} \quad \Pr\{N^w(t) \geq 2\} = o(w), \tag{2}$$

where $\lambda_t > 0$ is the arrival process intensity at time t. Note that (2) is weaker than the condition for a Poisson arrival process because the latter also requires independent increments.

Lin et al. (2017) show that if $k/n \to 0$ as $n \to \infty$, then the *smallest symmetric* interval that contains the k nearest neighbors of t_0, denoted by $W_n^k(t_0)$, converges to 0 in L^2 norm; and the k nearest neighbors are asymptotically from distinct replications, implying that they are asymptotically independent. These results are used to prove consistency of $\bar{V}(t_0)$ for $v(t_0)$ in Lin et al. (2017).

3 VIRTUAL VARIANCE AND DERIVATIVE ESTIMATION

In this section, we define the variance and derivative for the virtual performance of our stochastic process, and propose our variance and derivative estimators.

3.1 Variance Estimation

The variance of the virtual performance $V(t_0)$ is $\sigma^2(t_0) = \mathrm{Var}(Y(t_0))$. We define a class of $k$nn variance estimator:

$$\widehat{\sigma}^2(t_0) = \sum_{(\ell,m)\in\mathscr{V}(t_0)} \phi_{\ell m}\left[Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)})\right]^2, \tag{3}$$

where the set $\mathscr{V}(t_0)$ contains the indices of the pairs $(t_0^{(\ell,n)}, t_0^{(m,n)})$ used for computing $\widehat{\sigma}^2(t_0)$. If $\mathscr{V}(t_0) = \{(\ell,m)|\ell \neq m \in \{1,2,\ldots,k\}\}$ (i.e., all the pairs of observations are used) and $\phi_{lm} = 1/(2k(k-1))$ for all (ℓ,m), then $\widehat{\sigma}^2(t_0)$ coincides with the *sample variance* of the k nearest neighbors, and it is also called a *residual-based* variance estimator, denoted by $\widehat{\sigma}^2_{\mathrm{RB}}(t_0)$ with $\mathscr{V}(t_0) = \mathscr{V}_{\mathrm{RB}}(t_0)$; i.e.,

$$\widehat{\sigma}^2_{\mathrm{RB}}(t_0) = \sum_{(\ell,m)\in\mathscr{V}_{\mathrm{RB}}(t_0)} \frac{1}{2k(k-1)}\left[Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)})\right]^2 = \frac{1}{k-1}\sum_{\ell=1}^{k}\left[Y(t_0^{(\ell,n)}) - \bar{V}(t_0)\right]^2. \tag{4}$$

Our residual-based variance estimator $\widehat{\sigma}^2_{\text{RB}}(t_0)$ is different from a typical sample variance which is computed from k *independent* measurements at $t = t_0$. Since it is very unlikely we will obtain any, much less multiple, observations at t_0 due to the nature of virtual performance, our proposed residual-based variance estimator is constructed based on the k nearest neighbors around t_0 and these k observations are usually dependent.

The residual-based variance estimator in (4) involves the pairs $(t_0^{(\ell,n)}, t_0^{(m,n)})$ from the k nearest neighbors. By contrast, Rice (1984) proposes a first-order *difference-based* variance estimator, denoted by $\widehat{\sigma}^2_{\text{DB}}(t_0)$, that only contains the pairs of any two *successive* observations such that the corresponding index set $\mathscr{V}(t_0)$ becomes $\mathscr{V}_{\text{DB}}(t_0) = \{(\ell,m) | m = \ell-1, \ell \in \{2,3,\ldots,k\}\}$ and the weight $\phi_{\ell m} = 1/(2(k-1))$, so

$$\widehat{\sigma}^2_{\text{DB}}(t_0) = \sum_{(\ell,m)\in\mathscr{V}_{\text{DB}}(t_0)} \frac{1}{2(k-1)} \left[Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)})\right]^2 = \frac{1}{2(k-1)} \sum_{\ell=2}^{k} \left[Y(t_0^{(\ell,n)}) - Y(t_0^{(\ell-1,n)})\right]^2. \quad (5)$$

Compared with the residual-based variance estimator defined in (4), a difference-based variance estimator like (5) removes the trend in the mean. There exist other variations of difference-based variance estimators. For example, Gasser et al. (1986) introduce pseudo-residuals to construct their difference-based variance estimator which assigns each squared difference its own weight based on their distances to the point of interest. Typically, equally-weighted difference-based variance estimators are applied for problems with equispaced design points, and many related papers like Rice (1984) assume independence among the observations. Nevertheless, the superposed arrivals in $\mathscr{T}_n$ could be very dense if either the arrival intensity or the number of replications n is large, so all observations within the superposed sample path $\mathscr{T}_n$ are close to each other such that the impact of the distance will be less significant. As for the independence assumption, we will establish the asymptotic independence for the k nearest neighbors around t_0 under certain conditions on the system and the growth rate of k. Therefore, we suggest the equally-weighted difference-based variance estimator defined in (5).

To further compare these two $k$nn variance estimators, $\widehat{\sigma}^2_{\text{RB}}(t_0)$ and $\widehat{\sigma}^2_{\text{DB}}(t_0)$, we establish their asymptotic properties in Section 4, and propose a parameter-tuning approach for $\widehat{\sigma}^2_{\text{DB}}(t_0)$ in Section 5.

3.2 Derivative Estimation

The derivative of $v(t)$ evaluated at $t = t_0$ is $v'(t_0) = dv(t)/dt|_{t=t_0}$. As mentioned in Section 1, the traditional FD method cannot be effectively used in our virtual statistics problem. If the FD δ is small, as it should be for low bias, then the arrival times t_{ij} in the interval $[t_0, t_0+\delta]$ or $[t_0-\delta, t_0+\delta]$ maybe nearly the *same*, and therefore cancel in a FD estimator.

A näive derivative estimator for $v'(t)$ at $t = t_0$ is $(Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)}))/(t_0^{(\ell,n)} - t_0^{(m,n)})$, where $(t_0^{(\ell,n)}, Y(t_0^{(\ell,n)}))$ and $(t_0^{(m,n)}, Y(t_0^{(m,n)}))$ are two observations near t_0. Motivated by this example, we define a class of derivative estimators for $v'(t_0)$:

$$\widehat{\beta}(t_0) = \sum_{(\ell,m)\in\mathscr{D}(t_0)} \omega_{\ell m} \left[\frac{Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)})}{t_0^{(\ell,n)} - t_0^{(m,n)}}\right], \quad \text{where} \quad \omega_{\ell m} = \frac{(t_0^{(\ell,n)} - t_0^{(m,n)})^2}{\sum_{(r,s)\in\mathscr{D}(t_0)} (t_0^{(r,n)} - t_0^{(s,n)})^2}. \quad (6)$$

The derivative estimator defined in (6) is the weighted average of the slopes of two neighbors within $\mathscr{D}(t_0)$, and the weight ω_{lm} is proportional to the difference between $t_0^{(\ell,n)}$ and $t_0^{(m,n)}$. Similar to the index set $\mathscr{V}(t_0)$ in the variance estimator, $\mathscr{D}(t_0)$ contains the indices of all pairs $(t_0^{(\ell,n)}, t_0^{(m,n)})$ used for computing $\widehat{\beta}(t_0)$.

A natural choice of $\mathscr{D}(t_0)$ is to employ the same k nearest neighbors used in the mean $\bar{V}(t_0)$. Then $\mathscr{D}(t_0) = \{(\ell,m) | \ell \neq m \in \{1,2,\ldots,k\}\}$. In this case, we can express $\widehat{\beta}(t_0)$ as

$$\widehat{\beta}(t_0) = \sum_{\ell\neq m}^{k} \frac{(t_0^{(\ell,n)} - t_0^{(m,n)})^2}{\sum_{r\neq s}^{k} (t_0^{(r,n)} - t_0^{(s,n)})^2} \cdot \frac{Y(t_0^{(\ell,n)}) - Y(t_0^{(m,n)})}{t_0^{(\ell,n)} - t_0^{(m,n)}} = \frac{\sum_{\ell=1}^{k} (t_0^{(\ell,n)} - \bar{t})(Y(t_0^{(\ell,n)}) - \bar{V}(t_0))}{\sum_{\ell=1}^{k} (t_0^{(\ell,n)} - \bar{t})^2},$$

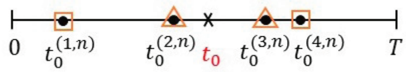

Figure 1: An example of constructing $\widehat{\beta}_{\text{SWD}}(t_0)$ based on 4 nearest neighbors.

where $\bar{t} = \sum_{\ell=1}^{k} t_0^{(\ell,n)}/k$. For this choice of $\mathscr{D}(t_0)$, the derivative estimator defined in (6) coincides with the ordinary least squares (OLS) estimator. Such a derivative estimator, denoted by $\widehat{\beta}_{\text{OLS}}(t_0)$ associated with $\mathscr{D}_{\text{OLS}}(t_0)$, can also be viewed as the estimated slope coefficient for a linear regression model of the k nearest neighbors to t_0.

De Brabanter et al. (2013) and De Brabanter and Liu (2015) propose a different choice of $\mathscr{D}(t_0)$ for constructing $\widehat{\beta}(t_0)$. Instead of using all $(t_0^{(\ell,n)}, t_0^{(m,n)})$, they only choose the pairs where $t_0^{(\ell,n)}$ and $t_0^{(m,n)}$ are symmetric around t_0. We call such a derivative estimator the *symmetric weighted difference* (SWD) estimator, and the corresponding index set becomes $\mathscr{D}_{\text{SWD}}(t_0) = \{(\ell,m) | \ell + m = 2\tilde{k}+1, \ell > m \in \{1,2,\ldots,\tilde{k}\}\}$, where $\tilde{k}$ is the number of involved pairs (i.e., slopes).

A simple illustration for constructing $\widehat{\beta}_{\text{SWD}}(t_0)$ is shown in Figure 1. Suppose we use 4 nearest neighbors around t_0 to construct $\widehat{\beta}(t_0)$, then $\widehat{\beta}_{\text{OLS}}(t_0)$ will involve all $4 \times (4-1) = 12$ pairs of $(t_0^{(\ell,n)}, t_0^{(m,n)})$ while $\widehat{\beta}_{\text{SWD}}(t_0)$ will only include two pairs: $(t_0^{(4,n)}, t_0^{(1,n)})$ and $(t_0^{(3,n)}, t_0^{(2,n)})$.

The number of involved slopes $\tilde{k}$ must satisfy $\tilde{k} \leq k/2$. In the simple example shown above, $\tilde{k} = 2$ when 4 nearest neighbors are chosen, which is the best situation. If $t_0^{(2,n)}$ also locates on the same side of t_0 as $t_0^{(3,n)}$ and $t_0^{(4,n)}$, then $\widehat{\beta}_{\text{SWD}}(t_0)$ will only contain one slope computed from $(t_0^{(2,n)}, t_0^{(1,n)})$. The worst case is that all these 4 nearest neighbors are on one side of t_0 such that we cannot construct $\widehat{\beta}_{\text{SWD}}(t_0)$ according to its definition. Therefore, to construct a SWD estimator $\widehat{\beta}_{\text{SWD}}(t_0)$, we do not use the original k nearest neighbors. Instead, we choose the k nearest neighbors to t_0 from $[0, t_0]$ and another k nearest neighbors to t_0 from $[t_0, T]$, and sort these $2k$ neighbors as $t_{0,\text{SWD}}^{(1,n)} < t_{0,\text{SWD}}^{(2,n)} < \cdots < t_{0,\text{SWD}}^{(2k,n)}$. Then the index set $\mathscr{D}(t_0)$ for $\widehat{\beta}_{\text{SWD}}(t_0)$ is $\mathscr{D}_{\text{SWD}}(t_0) = \{(\ell,m) | \ell + m = 2k+1, \ell > m \in \{1,2,\ldots,k\}\}$ such that $\widehat{\beta}_{\text{SWD}}(t_0)$ is constructed on k pairs of symmetric observations around t_0. Note that these $2k$ neighbors might not be the $2k$ *nearest* neighbors to t_0. The asymptotic properties of $\widehat{\beta}_{\text{OLS}}(t_0)$ and $\widehat{\beta}_{\text{SWD}}(t_0)$ are established in Section 4.

4 ASYMPTOTIC PROPERTIES OF VARIANCE AND DERIVATIVE ESTIMATORS

In this section we establish the asymptotic properties of the proposed variance and derivative estimators. The proofs for all the asymptotic results are provided in Lin and Nelson (2017).

Theorem 1 Suppose that the system of interest satisfies $\mathrm{E}[Y^2(t)] < \infty$ and its arrival-counting process satisfies (2), and that the true response surface $v(t)$ and the marginal variance $\sigma^2(t)$ are Lipschitz continuous with finite Lipschitz constants $L_1, L_2 > 0$ for any $t_1, t_2 \in [0,T]$. If $k/n \to 0$ as $k, n \to \infty$, then

(i) the residual-based variance estimator $\widehat{\sigma}^2_{\text{RB}}(t_0)$ is asymptotically unbiased and consistent for $\sigma^2(t_0)$;

(ii) the difference-based variance estimator $\widehat{\sigma}^2_{\text{DB}}(t_0)$ is asymptotically unbiased for $\sigma^2(t_0)$;

(iii) if in addition, $\mathrm{E}[T^4] < \infty$ and the fourth moment of $Y(t)$ is also Lipschitz continuous with finite Lipschitz constant $L_3 > 0$ for any $t_1, t_2 \in [0,T]$, then $\widehat{\sigma}^2_{\text{DB}}(t_0)$ is asymptotically consistent for $\sigma^2(t_0)$.

De Brabanter and Liu (2015) show the asymptotic unbiasedness and consistency for $\widehat{\beta}_{\text{SWD}}(t_0)$, but they only consider cases where all observations are independent and homoscedastic. We employ the key part of their proof and then extend it to our problem in which the observations might be dependent and heteroscedastic. Before establishing the asymptotic properties for $\widehat{\beta}_{\text{SWD}}(t_0)$, we need to establish the following lemma.

Lemma 1 Suppose that the system of interest satisfies $\text{E}[Y^2(t)] < \infty$ and its arrival-counting process satisfies (2). Let $t_{0,\text{SWD}}^{(1,n)} < t_{0,\text{SWD}}^{(2,n)} < \cdots < t_{0,\text{SWD}}^{(2k,n)}$ be the sorted $2k$ observations used for computing $\widehat{\beta}_{\text{SWD}}(t_0)$. Define $W_{\text{SWD}}^{2k}(t_0) = t_{0,\text{SWD}}^{(2k,n)} - t_{0,\text{SWD}}^{(1,n)}$ as the smallest interval that contains these $2k$ observations, and

$$I_{\text{SWD}}^{2k}(t_0) = \begin{cases} 1, & \text{if } t_{0,\text{SWD}}^{(1,n)}, t_{0,\text{SWD}}^{(2,n)}, \cdots, t_{0,\text{SWD}}^{(2k,n)} \text{ are from distinct replications} \\ 0, & \text{otherwise.} \end{cases}.$$

If $k/n \to 0$ as $n \to \infty$, then

(i) $W_{\text{SWD}}^{2k}(t_0) \xrightarrow{L^2} 0$, implying that $\lim_{\substack{n\to\infty \\ k/n\to 0}} \text{E}[(W_{\text{SWD}}^{2k}(t_0))^2] = 0$;

(ii) and $\Pr\{I_{\text{SWD}}^{2k}(t_0) = 1\} \to 1$; that is, $\{Y(t_{0,\text{SWD}}^{(1,n)}), Y(t_{0,\text{SWD}}^{(2,n)}), \cdots, Y(t_{0,\text{SWD}}^{(2k,n)})\}$ are asymptotically independent.

Theorem 2 Suppose that the system of interest satisfies $\text{E}[Y^2(t)] < \infty$ and its arrival-counting process satisfies (2), and that $v(t)$ is twice continuously differentiable with $v''(t) < \infty$ and $\sup_{t\in[0,T]} \sigma^2(t) = \sigma^2_{\text{sup}} < \infty$. If $k/n \to 0$ as $k, n \to \infty$, then

(i) $\widehat{\beta}_{\text{SWD}}(t_0)$ is asymptotically unbiased for $v'(t_0)$;

(ii) if in addition, $k^{3/2}/n \to \infty$ as $k, n \to \infty$, then $\widehat{\beta}_{\text{SWD}}(t_0)$ is asymptotically consistent for $v'(t_0)$.

We can use the same proof of asymptotic unbiasedness of $\widehat{\beta}_{\text{SWD}}(t_0)$ from De Brabanter and Liu (2015) for proving part (i) in Theorem 2, since neither the independence nor homoscedasticity assumption is required for showing asymptotic unbiasedness. The proof for part (ii) is also based on De Brabanter and Liu (2015), but we need to transform our problem into their situation where both the independence and homoscedasticity assumption are required. The proof for Theorem 3 is similar to the one for Theorem 2.

Theorem 3 Suppose that the system of interest satisfies $\text{E}[Y^2(t)] < \infty$ and its arrival-counting process satisfies (2), and that $v(t)$ is twice continuously differentiable with $v''(t) < \infty$ and $\sup_{t\in[0,T]} \sigma^2(t) = \sigma^2_{\text{sup}} < \infty$. If $k/n \to 0$ as $k, n \to \infty$, then

(i) $\widehat{\beta}_{\text{OLS}}(t_0)$ is asymptotically unbiased for $v'(t_0)$;

(ii) if in addition, $k^2/n \to \infty$ as $k, n \to \infty$, then $\widehat{\beta}_{\text{OLS}}(t_0)$ is asymptotically consistent for $v'(t_0)$.

From Theorems 2–3, we see that k should not increase faster than n but should not increase too slowly either. The growth rate of k affects the width of the interval $W_{\text{SWD}}^{2k}(t_0)$. If k grows too slowly, then $W_{\text{SWD}}^{2k}(t_0)$ might be too narrow such that the observations are too close to each other, which is harmful in derivative estimation. Specifically, the number of nearest neighbors k for $\widehat{\beta}_{\text{SWD}}(t_0)$ should increase faster than the k for $\widehat{\beta}_{\text{OLS}}(t_0)$. This is because $\widehat{\beta}_{\text{OLS}}(t_0)$ uses many more weighted slopes so its variance can be better controlled.

5 PRACTICAL APPROACH

In practice, we need to determine the tuning parameter k to construct good variance and derivative estimators based on finite sample paths. We discuss how to tune the parameter k in this section.

We know $\widehat{\sigma}^2_{\text{RB}}(t_0)$ is the sample variance of the k nearest neighbors, so it is natural to use the same optimal $k^\star$, denoted by $k^\star_{\text{mean}}$, tuned from the mean estimation procedure. Lin et al. (2017) introduce a leave-one-replication-out cross validation (LORO CV) method to obtain $k^\star_{\text{mean}}$. For the difference-based variance estimator $\widehat{\sigma}^2_{\text{DB}}(t_0)$, we suggest two k values: one is $k^\star_{\text{mean}}$ if estimating $v(t_0)$ is also of interest; the other one is to tune k directly without the mean estimation, as described in Algorithm 1.

Algorithm 1 $k$nn method via K-fold cross validation for $\widehat{\sigma}^2_{\text{DB}}(t_0)$

Input fixed test vector $\mathbf{t}_{\text{test}} = \{t_1, t_2, \ldots, t_{M_{\text{test}}}\}$ and search range $k_L < k_U$, NN = "nearest neighbors."
Randomly divide the n replications into K folds of approximately equal size.
for $\ell = 1, 2, \ldots, K$ **do**
  $S_{\text{test}} \leftarrow \{\mathbf{Y}_j, \mathbf{t}_j; j = 1, 2, \ldots, n_\ell\}$, where $\mathbf{t}_j = \{t_{1j}, t_{2j}, \ldots, t_{M_j j}\}$, $\mathbf{Y}_j = \{Y(t_{1j}), Y(t_{2j}), \ldots, Y(t_{M_j j})\}$, and n_ℓ is the number of replications in the ℓ^{th} fold.
  $S_{\text{train}} \leftarrow$ all data except S_{test}.
  Find the one nearest neighbor from each $\mathbf{t}_j \in S_{\text{test}}$ for each $t_m \in \mathbf{t}_{\text{test}}$.
  Compute the sample variance $S^2_\ell(t_m)$ using these independent n_ℓ observations for each $t_m \in \mathbf{t}_{\text{test}}$.
  Find k_U NN in S_{train} to each $t_m \in \mathbf{t}_{\text{test}}$.
  Store the indices of the k_U NN to each $t_m \in \mathbf{t}_{\text{test}}$ into an index matrix $\mathbf{M}_{\text{ind}} \in \Re^{M_{\text{test}} \times k_U}$, where the ith row in $\mathbf{M}_{\text{ind}}$ contains the indices of the k_U NN to $t_m \in \mathbf{t}_{\text{test}}$.
  for $k \in [k_L, k_U]$ **do**
    Extract the first k columns from $\mathbf{M}_{\text{ind}}$.
    Find the k NN to each $t_m \in \mathbf{t}_{\text{test}}$ and compute the difference-based estimator $\widehat{\sigma}^2_{\text{DB},\ell}(t_m, k)$.
  end for
end for
for $k \in [k_L, k_U]$ **do**
  Compute $\text{EMSE}(k) = \left(\sum_{\ell=1}^{K} \sum_{m=1}^{M_{\text{test}}} [S^2_\ell(t_m) - \widehat{\sigma}^2_{\text{DB},\ell}(t_m, k)]^2\right) / (M_{\text{test}} \times K)$.
end for
Choose $k^\star_{\text{db}}$ that results in the minimum $\text{EMSE}(k)$.

A simple example to illustrate how this algorithm works is provided in Lin and Nelson (2017). We find that tuning the parameter for $\widehat{\sigma}^2_{\text{DB}}(t_0)$ is computationally cheaper than for $\widehat{\sigma}^2_{\text{RB}}(t_0)$. For a single k value, say k_0, the computational effort required for computing the $\text{EMSE}(k_0)$ of $\widehat{\sigma}^2_{\text{DB}}(t_0)$ is $O(KM_{\text{test}} \log(\sum_{j=1}^{n} M_j))$, where M_{test} is the number of test points chosen in Algorithm 1. On the other hand, the computational effort required by $\widehat{\sigma}^2_{\text{DB}}(t_0)$ depends on the mean estimation procedure, which requires $O((\sum_{j=1}^{n} M_j) \log(\sum_{j=1}^{n} M_j))$ for computing $\text{EMSE}(k_0)$. Typically, we have $\sum_{j=1}^{n} M_j \gg KM_{\text{test}}$ because $\sum_{j=1}^{n} M_j$ increases fast as we increase the number of replications n or have a very dense arrival counting process, while the number of folds K is usually 10 and M_{test} is often chosen to be much smaller than any M_j. Take one queueing system with $n = 100$ replications we are going to analyze in Section 6 as an example, $\sum_{j=1}^{100} M_j = 30852$ while $KM_{\text{test}} = 10 \times 15 = 150$. Hence, if one is only interested in the variance of $V(t_0)$, then obtaining a difference-based variance estimator $\widehat{\sigma}^2_{\text{DB}}(t_0)$ from Algorithm 1 is much cheaper.

As for the two derivative estimators, we propose to use the same optimal $k^\star_{\text{mean}}$ value. That is, we use the same $k^\star_{\text{mean}}$ nearest neighbors to fit a linear regression model and the estimated slope coefficient is $\widehat{\beta}_{\text{OLS}}(t_0)$. For $\widehat{\beta}_{\text{SWD}}(t_0)$, we choose $k^\star_{\text{mean}}$ nearest neighbors to t_0 from each side of t_0 and then use these two $2k^\star_{\text{mean}}$ neighbors to compute $\widehat{\beta}_{\text{SWD}}(t_0)$. Note that there might be fewer than $k^\star_{\text{mean}}$ observations (e.g., only $\tilde{k} < k^\star_{\text{mean}}$ observations) on one side of t_0 if t_0 is close to the endpoints 0 or T. If that happens, then we only choose $\tilde{k}$ nearest neighbors from each side of t_0 such that $\widehat{\beta}_{\text{SWD}}(t_0)$ will be constructed on $\tilde{k}$ slopes.

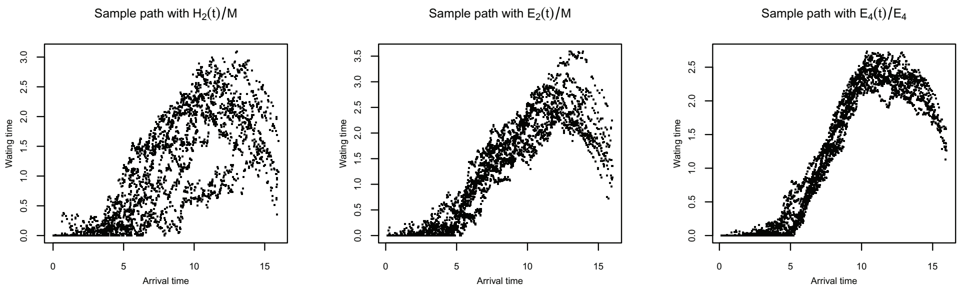

Figure 2: Sample paths of 10 replications for the three queueing systems.

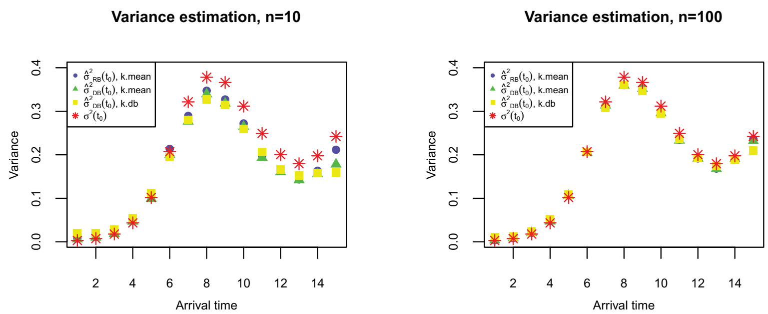

Figure 3: Performance of $\widehat{\sigma}^2_{\mathrm{RB}}(t_0)$ vs. $\widehat{\sigma}^2_{\mathrm{DB}}(t_0)$ for $H_2(t)/M/1/c$ system.

6 EXPERIMENTS

Lin et al. (2017) study the virtual waiting times for a series of phase-type queueing models to evaluate the performance of $\bar{V}(t_0)$. In this paper, we use the same phase-type queueing models to evaluate the performance of the proposed variance and derivative estimators.

We study three phase-type FIFO queueing models: $H_2(t)/M/s/c$, $E_2(t)/M/s/c$, and $E_4(t)/E_4/s/c$, where H_2 stands for two-phase hyperexponential distribution, E_2 (or E_4) stands for two-phase (or four-phase) Erlang distribution, and M stands for exponential distribution. The nonstationary arrival rate functions are piecewise linear, the service rate $\mu = 20$, the number of servers $s = 1$, the system capacity $c = 50$, and the mixing probability p within the $H_2(t)$ distribution is 0.4. The 10-replications sample paths which illustrate the trend and variability for these three systems are shown in Figure 2.

The reason we choose these phase-type queueing models for the empirical study is that we can compute the virtual performance measures of interest. Lin et al. (2017) describe how to compute the expected virtual waiting time using Kolmogorov forward equations (KFEs), and we can compute the variance and derivative based on the same technique. Refer to Lin and Nelson (2017) for more details. *Overall, the proposed variance and derivative estimators turn out to estimate the true values very well for all three systems.*

We first present the simulation results for the variance estimators. In Section 5, we have discussed how to choose appropriate k values for the $k$nn variance estimators. We use $k^{\star}_{\mathrm{mean}}$ tuned from LORO CV for $\widehat{\sigma}^2_{\mathrm{RB}}(t_0)$; and we try two k values for $\widehat{\sigma}^2_{\mathrm{DB}}(t_0)$: one is $k^{\star}_{\mathrm{mean}}$ and the other is $k^{\star}_{\mathrm{db}}$ tuned directly from Algorithm 1. The performance of the variance estimators averaged across 100 macro-replications for the three systems are presented in Figures 3–5, where n indicates the number of replications within each macro-replication. Overall, the performance of the variance estimators is good, and the difference-based variance estimators are very close to the true variance when the system has low variability, as in Figure 5. On the other hand,

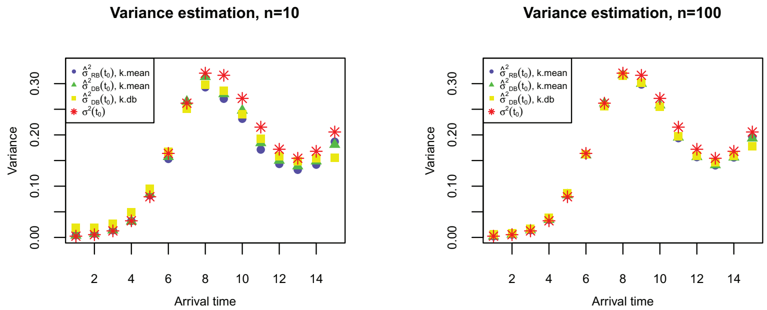

Figure 4: Performance of $\widehat{\sigma}^2_{\mathrm{RB}}(t_0)$ vs. $\widehat{\sigma}^2_{\mathrm{DB}}(t_0)$ for $E_2(t)/M/1/c$ system.

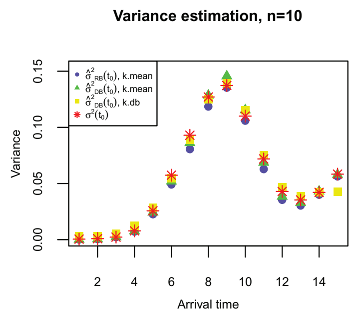

Figure 5: Performance of $\widehat{\sigma}^2_{\mathrm{RB}}(t_0)$ vs. $\widehat{\sigma}^2_{\mathrm{DB}}(t_0)$ for $E_4(t)/E_4/1/c$ system.

the variance estimators are more biased when the system is highly variable and n is small. For example, both $H_2(t)/M/1/c$ and $E_2(t)/M/1/c$ are more variable than $E_4(t)/E_4/1/c$ according to Figure 2, especially during the time period of $t = 7$ to $t = 10$, and we find these variance estimators become more biased in this time period when only 10 replications are used, but the bias is effectively reduced as n increases from 10 to 100 (Figures 3–4).

We find both $k^\star_{\mathrm{mean}}$ and $k^\star_{\mathrm{db}}$ are larger than the number of replications n such that there always exist dependence among the k nearest neighbors. Thus, the variance estimators underestimate the variance due to the positive correlation, especially when the system has high variability; in other words, the more variable the system is, the more biased the variance estimators could be if n is too small. This is because the optimal k value tuned from either LORO CV or K-fold CV is larger when the system has higher variability (e.g., $k^\star_{\mathrm{mean}} \approx 200$ for $H_2(t)/M/1/c$ and $k^\star_{\mathrm{mean}} \approx 50$ for $E_4(t)/E_4/1/c$ when $n = 10$ for both of these two systems),

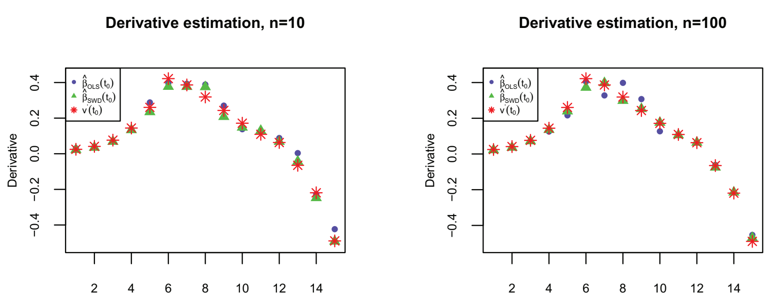

Figure 6: Performance of $\widehat{\beta}_{\mathrm{OLS}}(t_0)$ vs. $\widehat{\beta}_{\mathrm{SWD}}(t_0)$ for $H_2(t)/M/1/c$ system.

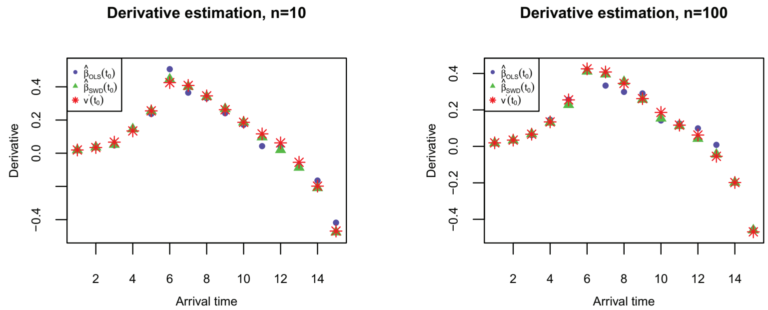

Figure 7: Performance of $\widehat{\beta}_{\text{OLS}}(t_0)$ vs. $\widehat{\beta}_{\text{SWD}}(t_0)$ for $E_2(t)/M/1/c$ system.

so the dependence issue is more severe for the more variable system such that the bias of the corresponding variance estimators is larger. Even for a single system, the bias of the variance estimators increases as the variability of the system increases. Take $H_2(t)/M/1/c$ as an example: the variance estimators become most biased at $t_0 = 8$ when the system variability itself reaches its peak value. For the two more variable systems $H_2(t)/M/1/c$ and $E_2(t)/M/1/c$, increasing n is an efficient way to reduce the bias of the variance estimators, because we find that the optimal k value tuned from CV does not increase as fast as n so that the dependence issue becomes less severe as n increases.

The role of CV is to balance bias and variance. To assess the variability of these variance estimators, we run 100 macro-replications for all scenarios so we can obtain an estimator for the variability of the variance estimators. Take $\widehat{\sigma}^2_{\text{DB}}(t_0)$ as an example: its variance estimator is computed as $\sum_{r=1}^{R}[\widehat{\sigma}^2_{\text{DB,r}}(t_0) - \overline{\widehat{\sigma}^2_{\text{DB}}}(t_0)]^2/(R-1)$, where $\widehat{\sigma}^2_{\text{DB,r}}(t_0)$ is the difference-based variance estimator computed from the rth macro-replication and $\overline{\widehat{\sigma}^2_{\text{DB}}}(t_0) = \sum_{r=1}^{R}\widehat{\sigma}^2_{\text{DB,r}}(t_0)/R$. We find that the variance of $\widehat{\sigma}^2_{\text{RB}}(t_0)$ is very close to the variance of $\widehat{\sigma}^2_{\text{DB}}(t_0)$ with $k^\star_{\text{mean}}$. Even though $\widehat{\sigma}^2_{\text{RB}}(t_0)$ includes many more pairs of observations, $\widehat{\sigma}^2_{\text{DB}}(t_0)$ removes the trend in the mean response function such that the variance caused by the regression function can be effectively reduced. As for the other $k$nn difference-based variance estimator, $\widehat{\sigma}^2_{\text{DB}}(t_0)$ with $k^\star_{\text{db}}$, the optimal $k^\star_{\text{db}}$ tuned from Algorithm 1 is much larger than $k^\star_{\text{mean}}$. Hence, the variance of $\widehat{\sigma}^2_{\text{DB}}(t_0)$ with $k^\star_{\text{db}}$ can be further reduced and it is smaller than the variance of the other two estimators.

The performance of the derivative estimators is provided in Figures 6–8. Overall, both $\widehat{\beta}_{\text{SWD}}(t_0)$ and $\widehat{\beta}_{\text{OLS}}(t_0)$ estimate the true derivative well, but $\widehat{\beta}_{\text{SWD}}(t_0)$ performs better than $\widehat{\beta}_{\text{OLS}}(t_0)$ in terms of both bias and variance. Specifically, we find that $\widehat{\beta}_{\text{OLS}}(t_0)$ is more biased when the variability dominates the trend in the system, e.g., during the time period of $t = 7$ to $t = 10$. This is because the true regression function is not necessarily a linear function and $\widehat{\beta}_{\text{OLS}}(t_0)$ assigns non-zero weight to every single pair of $(t_0^{(\ell,n)}, t_0^{(m,n)})$ for $(\ell,m) \in \mathscr{D}_{\text{OLS}}(t_0)$ so that the bias is very likely to be increased due to lack of symmetry. As for the variance estimators computed from 100 macro-replications for these derivative estimators, even though $\widehat{\beta}_{\text{OLS}}(t_0)$ includes many more slopes, the slopes used in $\widehat{\beta}_{\text{SWD}}(t_0)$ are less variable and less biased because the pairs $(t_{0,\text{SWD}}^{(\ell,n)}, t_{0,\text{SWD}}^{(m,n)})$ are well spread and symmetric around t_0.

Different from the variance estimation, the positive correlation is not that harmful for derivative estimation. Think about an extreme case where the true waiting time is a linear function of time and all the k nearest neighbors are from a single replication. If these k nearest neighbors are perfectly correlated (i.e., $\rho = 1$), then the derivative estimator is unbiased but the variance estimator is very poor. Thus, the positive correlation actually improves the performance of the derivative estimators in this situation.

In addition to the graphical presentation, we also display the *mean squared error* (MSE) for one case, the $E_2(t)/M/1/c$ system; see Table 1. Overall, the MSE of all the variance estimators and the $\widehat{\beta}_{\text{SWD}}(t_0)$

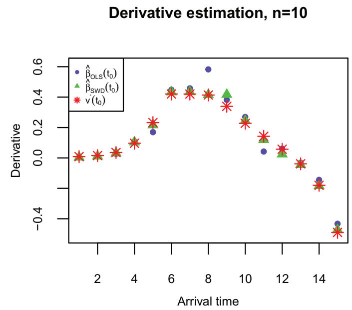

Figure 8: Performance of $\widehat{\beta}_{\text{OLS}}(t_0)$ vs. $\widehat{\beta}_{\text{SWD}}(t_0)$ for $E_4(t)/E_4/1/c$ system.

Table 1: MSE of variance and derivative estimators for $E_2(t)/M/1/c$ with $n = 100$.

	true	MSE			true	MSE	
t_0	$\sigma^2(t_0)$	$\widehat{\sigma}^2_{\text{RB}}(t_0)$	$\widehat{\sigma}^2_{\text{DB}}(t_0)$ with $k^\star_{\text{mean}}$	$\widehat{\sigma}^2_{\text{DB}}(t_0)$ with $k^\star_{\text{db}}$	$\nu'(t_0)$	$\widehat{\beta}_{\text{OLS}}(t_0)$	$\widehat{\beta}_{\text{SWD}}(t_0)$
1	0.0024	9.60×10^{-7}	9.35×10^{-7}	1.37×10^{-5}	0.0195	3.76×10^{-4}	8.03×10^{-5}
2	0.0054	4.09×10^{-6}	4.44×10^{-6}	9.02×10^{-6}	0.0344	1.77×10^{-3}	3.60×10^{-4}
3	0.0128	1.92×10^{-5}	1.95×10^{-5}	3.18×10^{-5}	0.0668	6.01×10^{-3}	1.49×10^{-3}
4	0.0325	6.70×10^{-5}	6.97×10^{-5}	8.84×10^{-5}	0.1343	2.76×10^{-2}	4.33×10^{-3}
5	0.0794	2.66×10^{-4}	2.59×10^{-4}	2.64×10^{-4}	0.2551	5.93×10^{-2}	1.09×10^{-2}
6	0.1639	8.51×10^{-4}	8.99×10^{-4}	6.11×10^{-4}	0.4254	1.34×10^{-1}	2.51×10^{-2}
7	0.2618	1.65×10^{-3}	1.86×10^{-3}	1.13×10^{-3}	0.4081	3.32×10^{-1}	3.87×10^{-2}
8	0.3204	1.60×10^{-3}	1.87×10^{-3}	1.40×10^{-3}	0.3455	4.10×10^{-1}	3.59×10^{-2}
9	0.3162	2.15×10^{-3}	2.34×10^{-3}	1.58×10^{-3}	0.2621	2.64×10^{-1}	4.30×10^{-2}
10	0.2712	1.81×10^{-3}	2.15×10^{-3}	1.51×10^{-3}	0.1862	1.67×10^{-1}	2.95×10^{-2}
11	0.2151	1.75×10^{-3}	1.72×10^{-3}	1.32×10^{-3}	0.1166	1.98×10^{-1}	2.51×10^{-2}
12	0.1720	1.14×10^{-3}	1.22×10^{-3}	8.67×10^{-4}	0.0623	1.36×10^{-1}	1.56×10^{-2}
13	0.1544	7.27×10^{-4}	7.11×10^{-4}	5.02×10^{-4}	-0.0546	6.03×10^{-2}	1.18×10^{-2}
14	0.1680	8.38×10^{-4}	8.85×10^{-4}	6.64×10^{-4}	-0.1982	5.07×10^{-2}	5.99×10^{-3}
15	0.2057	1.24×10^{-3}	1.32×10^{-3}	1.55×10^{-3}	-0.4689	2.48×10^{-2}	4.28×10^{-3}

derivative estimator are at least an order of magnitude smaller than the quantity being estimated. Notice that $\widehat{\beta}_{\text{SWD}}(t_0)$ has substantially smaller MSE than $\widehat{\beta}_{\text{OLS}}(t_0)$ for some t_0, which is what we expect due to the symmetry of the observations involved in $\widehat{\beta}_{\text{SWD}}(t_0)$.

To better interpret the simulation results, we choose $t_0 = 6$ for the $E_2(t)/M/1/c$ system as an illustration. If a customer arrives at this system at $t_0 = 6$, then the mean estimator of the waiting time in the queue for this customer is 0.71 minutes (obtained from Lin et al. (2017)), the variance estimator for the waiting time is 0.16, i.e., the standard deviation is 0.4 minutes (Figure 4). The SWD estimator $\widehat{\beta}_{\text{SWD}}(6) \approx 0.45$, meaning that the rate of change in the waiting time at $t_0 = 6$ is 0.45 minutes per time unit.

7 CONCLUSIONS

In this paper we propose two variance estimators and two derivative estimators for the virtual performance based on retained sample paths from simulation experiments. We show the asymptotic properties of these virtual statistics and propose a parameter tuning algorithm for the $k$nn difference-based variance estimator. The controlled studies show that even with the global optimal $k^\star_{\text{mean}}$ obtained from mean estimation, the performance of all these virtual statistics is good.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Foundation under Grant Number CMMI-1537060 and GOALI co-sponsor SAS Institute.

REFERENCES

De Brabanter, K., J. De Brabanter, B. De Moor, and I. Gijbels. 2013. "Derivative estimation with local polynomial fitting.". *Journal of Machine Learning Research* 14 (1): 281–301.

De Brabanter, K., and Y. Liu. 2015. "Smoothed nonparametric derivative estimation based on weighted difference sequences". In *Stochastic Models, Statistics and Their Applications*, 31–38. Springer.

Gasser, T., and H.-G. Müller. 1979. *Nonparametric estimation of regression functions and their derivatives*. Univ., Sonderforschungsbereich 123.

Gasser, T., and H.-G. Müller. 1984. "Estimating regression functions and their derivatives by the kernel method". *Scandinavian Journal of Statistics*:171–185.

Gasser, T., L. Sroka, and C. Jennen-Steinmetz. 1986. "Residual variance and residual pattern in nonlinear regression". *Biometrika*:625–633.

Liitiäinen, E., F. Corona, and A. Lendasse. 2008. "On nonparametric residual variance estimation". *Neural Processing Letters* 28 (3): 155–167.

Liitiäinen, E., F. Corona, and A. Lendasse. 2010. "Residual variance estimation using a nearest neighbor statistic". *Journal of Multivariate Analysis* 101 (4): 811–823.

Lin, Y., and B. L. Nelson. 2016. "Simulation analytics for virtual statistics via k nearest neighbors". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 448–459. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lin, Y., and B. L. Nelson. 2017. "Variance and derivative estimation for virtual performance in simulation analytics". Technical report, Northwestern University.

Lin, Y., B. L. Nelson, and L. Pei. 2017. "Virtual Statistics in Simulation via k Nearest Neighbors". Technical report, Northwestern University.

Rice, J. 1984. "Bandwidth choice for nonparametric regression". *The Annals of Statistics*:1215–1230.

Smith, J. S., and B. L. Nelson. 2015. "Estimating and interpreting the waiting time for customers arriving to a non-stationary queueing system". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2610–2621. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhou, S., and D. A. Wolfe. 2000. "On derivative estimation in spline regression". *Statistica Sinica*:93–108.

AUTHOR BIOGRAPHIES

YUJING LIN received her Ph.D. from the Department of Industrial Engineering and Management Sciences at Northwestern University in 2017. She then joined Amazon as a research scientist in the Supply Chain & Optimization Technologies team. Her research interest is simulation analytics, and her e-mail address is yujinglin2013@u.northwestern.edu.

BARRY L. NELSON is the Walter P. Murphy Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University and a Distinguished Visiting Scholar in the Lancaster University Management School. His e-mail address is nelsonb@northwestern.edu.