Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

SIMULATION ANALYTICS FOR VIRTUAL STATISTICS VIA K NEAREST NEIGHBORS

Yujing Lin
Barry L. Nelson

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

ABSTRACT

"Virtual statistics" are performance measures that are conditional on the occurrence of an event; virtual waiting time of a customer arriving to a queue at time t is one example. In this paper, we describe a k-nearest-neighbor method for estimating virtual statistics post-simulation from the retained sample paths, examining both its small-sample and asymptotic properties. We implement leave-one-replication-out cross validation for tuning the parameter k, and compare the prediction performance of the k-nearest-neighbor estimator with a time-bucket estimator.

1 INTRODUCTION

At an abstract level, we are interested in estimating a class of time-dependent performance measures for a (possibly) non-stationary stochastic process using the output of stochastic computer simulation. The class of measures of interest we call *virtual performance* at time t_0, denoted by $V(t_0)$. Specifically, $V(t_0)$ is some aspect of system performance conditional on a particular event occurring at time t_0, where the time t_0 is independent of the system state; it may be specified arbitrarily or perhaps be for all t_0 in a range $T_{\text{start}} \leq t_0 \leq T_{\text{end}}$.

Let F_{t_0} represent the distribution of $V(t_0)$. A virtual performance measure is some property of $V(t_0)$, such as its mean $\mathrm{E}[V(t_0)]$, its q quantile $F_{t_0}^{-1}(q)$, or its entire distribution F_{t_0}. In this paper we focus on the mean.

The canonical example of $V(t_0)$ is the virtual waiting time of a customer arriving to a service system at time t_0; this case is considered by Smith and Nelson (2015). The "virtual" aspect of virtual performance reflects the fact that the particular event need not actually occur in the sample paths, and in many cases it has probability 0. Notice that this is different from a quantity such as the number of customers in the system at time t_0 which is directly observable, and is not generally the same as the average waiting time across time, especially when the system is non-stationary.

A second example of virtual performance is the response time to a serious fire that occurs at time t_0. This is the famous case described in Carter and Ignall (1975). In the queueing situation considered by Smith and Nelson (2015), one expects that arrivals near time t_0 do regularly occur and therefore the observed waiting times of those customers can be used to estimate properties of $V(t_0)$. In the situation considered by Carter and Ignall (1975), the event of interest is extremely rare, at time t_0 or any other time, so the ability to *compute* conditional performance measures at any time t_0 is critical. We are interested in the former situation.

The simulation context within which we want to estimate properties of $V(t_0)$ is when we have *retained sample path information* from many simulation replications. This is in contrast to a simulation experiment specifically designed to estimate virtual performance, or statistics computed "on the fly" as the simulation executes as in Smith and Nelson (2015). The context has a significant influence on the methods we create, and rules out the idea of inserting the event of interest into the simulation at time t_0, an approach that would not generalize if we wanted to estimate $V(t_0)$ over many times in a range $T_{\text{start}} \leq t_0 \leq T_{\text{end}}$ since frequent insertions would fundamentally change the system. Instead, we want to do the best we can with the sample path data that are retained. Our approach illustrates one of the benefits of retaining simulation sample paths rather than automatically summarizing them.

We will not introduce a specific data structure in this paper, but what we have in mind is that something like a time-stamped trace of all events and state changes throughout the simulation runs is retained; this sort of trace is available in nearly all commercial simulation languages, although the tools to query it are not (yet). Storing such detailed transactional information facilitates a deeper analysis of simulation results, an approach we call "simulation analytics." See Nelson (2016) for an argument in favor of a data analytics approach to simulation output analysis, and see Bertsimas and Kallus (2014) for a compelling argument to employ data analytics in stochastic optimization.

2 WHEN IS THE PROBLEM EASY?

There has been substantial work on virtual performance measures, including virtual waiting time, in the queueing literature. The famous result that "Poisson arrivals see time averages" or PASTA Wolff (1982) is of this type. Loosely translated in the queueing context, PASTA indicates that the distribution of the number of customers in the system observed by arrivals from a Poisson process is the same as the time-average number of customers in the system, provided the system in no way anticipates the customer's arrival.

More generally one can consider the *work* in the system at time t_0, where "work" is defined as some measure of service pending or in process at time t_0; see Wolff (1989), Chapters 5 and 10 for a thorough discussion. One relevant measure of work at time t_0 is the sum of the service times of all customers in the queue, plus the remaining service times of those customers in service at t_0. In a single channel queue with no overtaking the virtual work and the virtual waiting time for an arrival at time t_0 coincide. Like number in the system, virtual work is observable at any time t_0 so each simulated replication provides one unbiased observation.

Many real-world systems, however, are much more complex so that PASTA or virtual work cannot be applied directly. In fact, virtual waiting time in queueing theory is typically for stationary systems in steady state. Further, the customers in a multi-server, multi-station system could pass each other, so the above-mentioned measure of work at time t_0 might not coincide with the virtual waiting time for an arrival at t_0. The arrivals of interest may not even be from outside the system, they could be arrivals to an internal queue in the network that are departures from other queues. A non-queueing example is the virtual recovery time of a manufacturing system if a failure occurs at t_0. Although we use virtual waiting time for arrivals to a queue as our example throughout this paper, our goal is to develop a more general approach to estimate virtual performance measures for complex systems.

3 PROBLEM AND METHOD

For notation, let $t_{1j} \leq t_{2j} \leq \cdots \leq t_{M_j j}$ be the times in replication j that the events of interest (e.g., customer arrivals) actually do occur, and let $Y(t_{ij})$ be the system performance that is actually observed (e.g., waiting time of customer who arrived at time t_{ij}), for replication $j = 1,2,\ldots,n$. That is, our data are $\{Y(t_{ij}),t_{ij};\ i = 1,2,\ldots,M_j, j = 1,2,\ldots,n\}$, and our virtual performance measure is

$$v(t_0) = \mathrm{E}(Y(t) \mid t = t_0).$$

Notice that the M_j are random variables in general, and that this situation includes estimating probabilities as well as mean values by representing them as the expectation of indicator functions. What approaches might be used to estimate $v(t_0)$?

Smith and Nelson (2015) use the observed outputs $\{Y(t_{ij})\colon t_{ij} \in [t_L,t_U]\}$ to estimate properties of $v(t_0)$, where $t_L \leq t_0 \leq t_U$ and $[t_L,t_U]$ is a predefined time bucket. Their primary assumption is that the outputs within a time bucket on a particular replication j are approximately stationary. There is a bias-variance trade off in choosing $\Delta = t_U - t_L$: Large Δ reduces variance but increases bias; small Δ increases variance but may reduce bias, unless the probability of an empty bucket becomes too large. A modification is to use a window $[t_0 - \delta, t_0 + \delta]$ centered at t_0 instead. This is probably an improvement, and is feasible if all of the t_0 values of interest are known in advance, or if all outputs $(Y(t_{ij}), t_{ij})$ are retained. The same bias-variance trade off still exists.

In this paper, we propose to construct a *k-nearest-neighbor* (KNN) estimator from the simulation data, provided that all outputs $(Y(t_{ij}), t_{ij})$ are retained. In contrast to designing a time bucket in advance, as in Smith and Nelson (2015), our method simply uses the average waiting time of the k closest arrivals around t_0 to estimate $v(t_0)$.

From the perspective of data analytics, KNN is a nonparametric supervised learning approach that is suitable for problems with low dimension and independent observations. The dimension of a problem is determined by the number of predictors included in a KNN model, which affects the required computer memory, computation time and smoothness of the regression function. Since the time that the events of interest occur is the only predictor for virtual performance measures, the dimension is 1 which is ideal. Nevertheless, the independence assumption is usually violated because the observed outputs and predictors are obtained from a strongly dependent sample path within each replication. Thus, dealing with the correlation among observations is one of the most challenging issues. In fact, the observations from the same replication are dependent but the ones obtained across distinct replications are independent, so in general the k nearest neighbors are a mix of independent and dependent observations.

The KNN estimator of $v(t_0)$, $\tilde{V}(t_0)$, can be represented as

$$\tilde{V}(t_0) = \frac{1}{k} \sum_{\ell=1}^{k} Y(t_0^{(\ell)}) = \frac{1}{k} \sum_{\ell=1}^{k} \left[v(t_0^{(\ell)}) + \varepsilon(t_0^{(\ell)}) \right], \tag{1}$$

where $t_0^{(\ell)}$ is the ℓth nearest neighbor of t_0, and $Y(t_0^{(\ell)})$ and $\varepsilon(t_0^{(\ell)})$ are the corresponding observed output and noise, for $\ell = 1, 2, \ldots, k$. The number of nearest neighbors to include, k, is the single tuning parameter in KNN. In most situations the optimal $k^\star$ is not known because the true response surface is unknown and the data are noisy. A common approach is to evaluate many values of k and then choose the best one based on a bias-variance trade off such as empirical mean squared error.

In typical data analytics problems the chosen value of k is small, so a direct search, say starting with $k = 1$, is possible. However, because the superposition of arrivals from n replications may be very dense around t_0, and ε can be quite variable, using a large number of times $t_0^{(\ell)}$ close to t_0 to estimate $v(t_0)$ may not have a significant impact on bias but greatly reduce variance. Having good insights into how the value of k would be affected by various features of the simulation data could be very helpful for saving computation effort, a topic we address below.

The remainder of this paper is organized as follows. In Section 4, we present the asymptotic properties of the proposed KNN estimator, and we discuss two types of asymptotic consistency under different conditions on the system of interest and the growth rate of k. We discuss how to apply cross validation (CV) to our problem setting, and develop a stylized model to analyze how data features affect $k^\star$ in Section 5. To evaluate the prediction performance of the KNN estimator, we compare it with the time-bucket (TB) estimator described in Smith and Nelson (2015) using a concrete example in Section 6. Some conclusions and future work are offered in Section 7.

4 ASTMPTOTIC PROPERTIES OF THE KNN ESTIMATOR

In this section, we show that the proposed KNN estimator is asymptotically consistent and unbiased under some certain conditions on the growth rate of k, the observed event times t_{ij} and the output Y_{ij}.

4.1 Devroye's Results

Let $(\mathbf{X},Y),(\mathbf{X}_1,Y_1),\ldots,(\mathbf{X}_n,Y_n)$ be independent identically distributed $\Re^d\times\Re$-valued random vectors with $\mathrm{E}(|Y|)<\infty$. The regression function $m(\mathbf{x})=\mathrm{E}(Y|\mathbf{X}=\mathbf{x})$ is estimated by

$$m_n(\mathbf{x})=\sum_{i=1}^{n}W_{ni}(\mathbf{x})Y_i,$$

where $W_{ni}(\mathbf{x})=1/k$ if Y_i is one of the k nearest neighbors and $W_{ni}(\mathbf{x})=0$ otherwise. Devroye (1981) proves the *pointwise consistency* of the KNN estimator $m_n(\mathbf{x})$, that is, when $k,n\to\infty$ and $k/n\to 0$, $m_n(\mathbf{x})$ satisfies

$$\mathrm{E}\left(|m_n(\mathbf{x})-m(\mathbf{x})|^p\right)\to 0 \tag{2}$$

for almost all $\mathbf{x}$ whenever $\mathrm{E}(|Y|^p)<\infty$ for $p\geq 1$. Pointwise consistency of $m_n(\mathbf{x})$ applies for any realized value of $\mathbf{X}$, i.e., $\mathbf{X}=\mathbf{x}$, and the expectation in (2) is taken with respect to the k nearest neighbors of fixed $\mathbf{x}$ because they are randomly distributed around $\mathbf{x}$.

Devroye (1981) further shows the *strong consistency* of $m_n(\mathbf{x})$: If $|Y|\leq\gamma<\infty$, $k,n\to\infty$ and $k/n\to 0$, then

$$\mathrm{E}\left(|m_n(\mathbf{x})-m(\mathbf{x})|\right)\to 0 \quad \text{as} \quad n\to\infty, \quad \text{for almost all } \mathbf{x}, \tag{3a}$$

and

$$\mathrm{E}\left(|m_n(\mathbf{X})-m(\mathbf{X})|\right)\to 0 \quad \text{as} \quad n\to\infty. \tag{3b}$$

If, in addition, $k/\log n\to\infty$ as $n\to\infty$, then

$$|m_n(\mathbf{x})-m(\mathbf{x})|\to 0 \quad \text{a.s. as} \quad n\to\infty, \quad \text{for almost all } \mathbf{x}, \tag{4a}$$

and

$$\mathrm{E}\left(|m_n(\mathbf{X})-m(\mathbf{X})|\,|\,\mathbf{X}_1,Y_1,\ldots,\mathbf{X}_n,Y_n\right)\to 0 \quad \text{a.s. as} \quad n\to\infty. \tag{4b}$$

We see that (3a) is a specific case of (2) where $p=1$, but (3b) states the consistency on random vector $\mathbf{X}$ instead of a fixed $\mathbf{x}$. Similarly, the expectation in (3b) is with respect to the distribution of the k nearest neighbors of $\mathbf{X}$. Moreover, if k increases slower than n but faster than $\log n$ as $n\to\infty$, then we have almost sure consistency as shown in (4). From (3) and (4), we see that to have a strongly consistent estimator $m_n(x)$, the system output Y must be bounded, which is not assumable in general. But having a bounded $\mathrm{E}(|Y|^p)$ is a weaker assumption that many practical systems satisfy.

Another underlying assumption for the consistency results to hold is the independence among observations. This assumption will not be satisfied in many virtual statistics problems. To directly apply Devroye's results, the k nearest neighbors need to come from distinct replications so that their corresponding responses are independent, in the limit. How fast k can grow as $n\to\infty$ to assure independence is what we discuss in Section 4.2 below.

4.2 Asymptotics of the KNN Estimator with Poisson Arrivals

In our formulation the arrival time is the only "predictor" in the KNN model, and we are interested in predicting at a fixed time or times, denoted generically by t_0. In Devroye's notation, $\mathbf{x}=t_0$, $m(\mathbf{x})=v(t_0)$, and $m_n(\mathbf{x})=\tilde{V}(t_0)$. Here we establish conditions on k and n that allow (2) and (4a) to hold for the special case of non-stationary Poisson arrivals. We prove that if $k^2/n\to 0$ as both $k,n\to\infty$, then the KNN estimator is pointwise consistent and asymptotically unbiased. Even if k is fixed and only $n\to\infty$, then the KNN estimator is still asymptotically unbiased under mild conditions. The key ideas follow.

We assume that the arrival-counting process of a dynamic system, $\{N(t):t\geq 0\}$, is non-stationary Poisson with integrable rate function $\lambda(t)>0$, the response surface $Y(t)$ has finite variance for all t, and n independent sample paths (replications) are retained. Therefore the superposition of these n independent processes, $\{N^{\dagger}(t) : t \geq 0\}$, is also non-stationary Poisson with intensity function $n\lambda(t)$. Consider a symmetric interval around t_0 with width w_n, and let $t_0^{(\ell)}$ be the ℓth nearest neighbor to t_0 from the superposed set of $\{t_{ij}\}$ with its corresponding observed output $Y(t_0^{(\ell)})$, $\ell = 1, 2, \ldots, k$.

For any single replication, the probability that no arrival time falls in the interval $[t_0 - w_n/2, t_0 + w_n/2]$ is

$$p_0 = \Pr\left\{N\left(t_0 + \frac{w_n}{2}\right) - N\left(t_0 - \frac{w_n}{2}\right) = 0\right\} = e^{-[\Lambda(t_0+w_n/2)-\Lambda(t_0-w_n/2)]} = e^{-\Delta_\Lambda(t_0,w_n)}$$

where $\Lambda(t) = \int_0^t \lambda(s)ds$ and $\Delta_\Lambda(t,w) = \Lambda(t+w/2) - \Lambda(t-w/2)$. Similarly, the probabilities that a replication contributes one, or more than one, point to the interval are

$$p_1 = \Delta_\Lambda(t_0, w_n) \cdot e^{-\Delta_\Lambda(t_0,w_n)} \quad \text{and} \quad p_2 = 1 - p_0 - p_1$$

respectively. Hence, for any $w_n \in \mathbb{R}^+$ and $k \in \mathbb{Z}^+$, the probability that $Y(t_0^{(1)}), \ldots, Y(t_0^{(k)})$ are independent, conditioned on the interval $[t_0 - w_n/2, t_0 + w_n/2]$ containing k points, is

$$\begin{aligned}
P(n,k) &= \Pr\left\{Y(t_0^{(1)}), \ldots, Y(t_0^{(k)}) \text{ are independent} \middle| N^{\dagger}\left(t_0 + \frac{w_n}{2}\right) - N^{\dagger}\left(t_0 - \frac{w_n}{2}\right) = k\right\} \\
&= \Pr\left\{t_0^{(1)}, \ldots, t_0^{(k)} \text{ from distinct replications} \middle| N^{\dagger}\left(t_0 + \frac{w_n}{2}\right) - N^{\dagger}\left(t_0 - \frac{w_n}{2}\right) = k\right\} \\
&= \frac{\binom{n}{k} p_0^{n-k} p_1^k p_2^0}{e^{-n\Delta_\Lambda(t_0,w_n)} \left[n\Delta_\Lambda(t_0,w_n)\right]^k \Big/ k!} \\
&= \left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right).
\end{aligned}$$

If k is a fixed value, then $P(n,k) \to 1$ as $n \to \infty$. Since the conditional probability derived above is independent of the point of interest t_0 and interval width w_n, we can argue that if $n \to \infty$ with k fixed, then the probability that $Y(t_0^{(1)}), Y(t_0^{(2)}), \ldots, Y(t_0^{(k)})$ are independent converges to 1 for any t_0 and k.

If we further allow $k \to \infty$ as $n \to \infty$, which is required for pointwise consistency, then

$$\begin{aligned}
\lim_{\substack{n\to\infty \\ k^2/n \to 0}} P(n,k) &= \lim_{\substack{n\to\infty \\ k^2/n \to 0}} \left(1 - \frac{1}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right) \\
&= \lim_{\substack{n\to\infty \\ k^2/n \to 0}} \left[e^{-1/n} \cdots e^{-(k-1)/n} + o\left(\frac{k}{n}\right)\right] = \lim_{\substack{n\to\infty \\ k^2/n \to 0}} e^{-k^2/(2n)+k/(2n)} = 1.
\end{aligned}$$

Therefore, for any system with finite $\mathrm{E}(|Y|^p)$ and non-stationary Poisson arrival process, if $k^2/n \to 0$ as $k, n \to \infty$, then the probability that the observed waiting times of the k nearest neighbors are independent converges to 1 and the corresponding KNN estimator is pointwise consistent. If, in addition, the system output is bounded and we employ a sequence $k = k_n$ such that $k, n \to \infty$, $k^2/n \to 0$ and $k/\log n \to \infty$, then we get a strongly consistent KNN estimator satisfying (4a).

To prove asymptotic unbiasedness of the KNN estimator, we need one more condition on the response surface. Suppose the expected virtual performance measure $v(t)$ is Lipschitz continuous in $[0, T]$, where T is the length of the simulation run. Then for any $t_1, t_2 \in [0, T]$,

$$|v(t_1) - v(t_2)| \leq L \cdot |t_1 - t_2|$$

where $L > 0$ is a finite constant.

Now let W_n^k be the width of the smallest symmetric interval $[t_0 - W_n^k/2, t_0 + W_n^k/2]$ that contains the k nearest neighbors from the n superposed replications. Then the bias of the KNN estimator defined in (1) is

$$\mathrm{E}\left[\tilde{V}(t_0) - v(t_0)\right] = \frac{1}{k}\sum_{\ell=1}^{k} \mathrm{E}\left[v(t^{(\ell)}) - v(t_0)\right] \le \frac{L}{k}\sum_{\ell=1}^{k} \mathrm{E}\left[|t^{(\ell)} - t_0|\right].$$

Now since the distances from all the k nearest neighbors to t_0 must be less than or equal to $W_n^k/2$, the bias is bounded by $L \cdot \mathrm{E}[W_n^k/2]$. Therefore, if the virtual performance measure is Lipschitz continuous and the expected width of the interval that contains the k nearest neighbors converges to 0 as $n \to \infty$, then the KNN estimator is asymptotically unbiased.

In the stationary Poisson case, we can show that W_n^k has an Erlang distribution with parameters $n\lambda$ and k, and thus $\mathrm{E}(W_n^k) = k/(n\lambda)$ which converges to 0 if $k/n \to 0$. Using the Borel-Canteli Lemma we can show a stronger result, that $W_n^k \to 0$ almost surely if $k^2/n \to 0$. These results are easily extended to the non-stationary Poisson case.

Remark: Walk (2010) also provides consistency results for KNN applied to dependent data, but assuming that $\{(\mathbf{X}_i, Y_i), i = 1, 2, \ldots\}$ are identically distributed with either ρ-mixing or α-mixing dependence. Such conditions would be appropriate for a stationary queueing system in steady state.

5 PRACTICAL APPROACH

In data analytics, it is always preferred to have training data for model selection, and separate testing data for assessment to avoid overfitting. When this is not possible or desirable, CV is a very useful technique that tends to avoid overfitting. However, training the model with such methods can be computationally expensive. In this section we discuss how to apply CV in our problem setting and analyze how data features affect the choice of k by introducing a stylized model.

5.1 Cross Validation

A traditional K-fold CV approach randomly divides N observations into K subsets or “folds,” then uses any $K-1$ folds of data to train the model and the remaining fold as the testing data, where K for the number of folds should not be confused with k, the number of nearest neighbors to average in a KNN model. This procedure is repeated K times until all folds of data are tested, then a goodness-of-fit measure is used to compare alternative models. Ten-fold CV is a common choice, and it is sometimes preferred to leave-one-observation-out CV (which could be thought of as N-fold CV) because otherwise the folds are highly correlated since they share $N-1$ observations.

We should not directly apply traditional CV to virtual performance estimation due to the correlation among observations collected from within the same replication. Hart (1991) points out that CV performs poorly with correlated data. Thus, instead of leaving individual observations out, we propose to leave out *entire replications*, using the the remaining replications as the training data and the left-out replication as the testing data; recall that replications are independent. This guarantees independence of each training and testing set.

Unfortunately, since the training data now consist of all observations from multiple replications, the k nearest neighbors of any test point will not necessarily come from different replications, and therefore may be correlated. For this reason we further propose *leave-one-replication-out (LORO) CV*, which increases the likelihood that many of the k closest observations are from different replications, diminishing the effect of correlation among observations of Y. Stated differently, if each fold leaves out 1 replication for testing, then there are a large number $(n-1)$ replications for training and the k nearest neighbors should be well spread out among them. We provide a detailed algorithm in Section 6.

5.2 Approximation for Optimal $k^\star$

While estimator consistency is important, in reality the number of replications n is finite. A stylized model is introduced here under which we can derive the optimal tuning parameter $k^\star$, and this provides some insight into how to adjust k based on data features. Ultimately we hope to use this insight to obtain a good starting solution, say k^{start}, for a LORO CV.

Suppose the arrival process follows a stationary Poisson process with rate λ, and the expected virtual output (e.g., virtual waiting time), $v(t_0)$, of arrivals near t_0 is βt_0, where $\beta \neq 0$ is a constant. Further, the simulation noise ε within a replication has zero-mean, variance σ^2, and common correlation ρ. And finally, the arrival process and noise are independent. More specifically, if we have n independent replications that form the training-data set then this stylized model is

$$Y_{ij} = v(t_{ij}) + \varepsilon_{ij} = \beta t_{ij} + \varepsilon_{ij}, \quad i = 1, 2, \ldots, M_j,\ j = 1, 2, \ldots, n+1,$$

where $\text{Var}(\varepsilon_{ij}) = \sigma^2$ for all i, j, $\text{Corr}(\varepsilon_{ij}, \varepsilon_{ij'}) = 0$ for $j \neq j'$, and $\text{Corr}(\varepsilon_{ij}, \varepsilon_{i'j}) = \rho$ for $i \neq i'$ and all j. Notice that β determines the bias from using neighbors to estimate $v(t_0)$—large $|\beta|$ means large bias—while σ^2 represents the simulation stochastic noise and ρ the output correlation within replications. Under this stylized model we can derive $k^\star$ that minimizes the expected value of mean squared error of prediction, which we call *averaged mean-squared error* (AMSE), for various cases as a function of k.

We start the analysis of AMSE(k) for the simplest case where all the observations are independent, i.e., $\rho = 0$. Suppose k can take positive real values, then by taking the derivative of AMSE(k) with respect to k, we get the optimal k that minimizes AMSE(k) to be

$$k^\star = \sqrt{2 + \frac{12 n^2 \lambda^2 \sigma^2}{\beta^2}}. \tag{5}$$

We see that if β is large, then a smaller $k^\star$ is preferred to avoid a large bias. If the data are very noisy, i.e., σ^2 is large, then a larger $k^\star$ is better for variance reduction. Either larger n or larger λ leads to a denser training data set so that including more points nearby will not cause as much bias but reduce the variance significantly. This result also applies to the case where $\rho > 0$ but all the k nearest neighbors are from distinct replications.

On the other hand, if $\rho > 0$ when all the k nearest neighbors are from the *same replication*, then we can still obtain an expression for AMSE(k), which is

$$k^\star = \sqrt{2 + \frac{12(1-\rho) n^2 \lambda^2 \sigma^2}{\beta^2}}. \tag{6}$$

The interpretation of $k^\star$ is very similar to the independent-observations case. Notice that if $\rho = 1$, then $k^\star = \sqrt{2} < 2$, meaning that including more than one point will not provide additional AMSE reduction when the data are perfectly correlated. In general, the larger ρ is, the smaller $k^\star$ will be.

The optimal $k^\star$ given in (5) and (6) correspond to two extreme cases. For a more general situation where the k nearest neighbors come from a mix of common and independent replications an expression for AMSE(k) is difficult to obtain. As a compromise, we use a *weighted average covariance* to approximate the actual covariance introduced in AMSE(k). Using the same framework as in Section 4.2, let q_i be the probability that i out of k nearest neighbors come from the same replication conditioned that the interval $[t_0 - w_n/2, t_0 + w_n/2]$ containing k points:

$$q_i = \frac{k!}{(k-i)!i!}\left(1 - \frac{1}{n}\right)^{k-i}\left(\frac{1}{n}\right)^i = C_k^i \left(1 - \frac{1}{n}\right)^{k-i}\left(\frac{1}{n}\right)^i, \quad 0 \leq i \leq k.$$

Since each one of the k nearest neighbors has a weight of $1/k$, then the covariance contributed by a single replication which contributes i out of k nearest neighbors is $C_i^2 \cdot 2\rho\sigma^2/k^2$, for $i \geq 2$. Thus, the weighted average covariance contributed by each replication is

$$\sum_{i=2}^{k} C_i^2 \cdot \frac{2\rho\sigma^2}{k^2} \cdot q_i = \frac{k-1}{n^2 k}\rho\sigma^2.$$

Hence, the total weighted average covariance contributed by the n independent replications involved in the training-data set is $(k-1)\rho\sigma^2/(nk)$. The optimal k in this situation is

$$k^\star = \sqrt{2 + \frac{12(1-\rho/n)n^2\lambda^2\sigma^2}{\beta^2}} = \sqrt{2 + \frac{12(1-\tilde{\rho})n^2\lambda^2\sigma^2}{\beta^2}}, \tag{7}$$

where $\tilde{\rho} = \rho/n$ can be treated as an *average correlation*. We see that the impact of correlation ρ fades away at a rate of $1/n$, so if we have enough replications then the effect of correlation can be neglected. Notice that the weighted average covariance, $(k-1)\rho\sigma^2/(nk)$, is just one possible approximation, and it may be different from the exact covariance or other approximations.

These results provide insight on the effect of number of replications n, arrival process intensity λ, rate of change of the response surface β, output variability σ^2 and correlation ρ on the optimal choice of k for KNN. We hope, eventually, to use this insight to derive a starting value for a CV search for $k^\star$. Direct use of the formulas, say by trying to estimate each term, does not give consistently good starting values, particularly when a linear approximation to the response surface $\beta_0 + \beta t$ is inadequate. As shown in Section 6 below, $k^\star$ values in the hundreds or thousands are possible, making a good starting value for the search computationally necessary.

6 EXPERIMENTS

In this section we apply the proposed method to the airport check-in problem discussed in Smith and Nelson (2015), and compare the prediction performance of the KNN estimator to their TB estimator.

The characteristics of the simplified airport check-in system are the same as the ones provided in Smith and Nelson (2015), which has a non-stationary arrival process and time-dependent agent capacity. The difference is that only a single type of passenger is considered in this paper. Synchronously, the arrival rate and agent capacity change every hour. Refer to Figure 1 in Smith and Nelson (2015) for details.

To determine the optimal tuning-parameter k of the KNN model, we apply LORO CV to the retained simulation data. Before introducing a detailed LORO CV algorithm, we need to clarify the following notation. Let the testing-dataset and training-dataset to be denoted by S_{test} and S_{train}, respectively. As in Section 3, let $Y_{ij} = Y(t_{ij})$ be the recorded time in system (TIS) of the arrival occurring at t_{ij}, and M_j be the number of observations in the jth replication. We express the KNN estimator of $v(t_{ij})$ as $\tilde{V}(t_{ij}, k)$ and use average sum of squared error, $\text{ASSE}(k)$, as our goodness-of-fit measure, where k is the tuning parameter. Hence, for any value of k, we can implement LORO CV presented in Algorithm 6 to get the corresponding value of $\text{ASSE}(k)$, and find the optimal one by searching over a range over k, say $k \in [k_L, k_U]$.

The true TIS of an arrival occurring at time t_0, $v(t_0)$, is unknown, thus, we ran a side-experiment in Simio to get a nearly unbiased estimator for $v(t_0)$ so that we can measure the prediction error by using either TB or KNN estimator. For each point of interest t_0, we ran the airport-check-in model which exactly follows the original experiment design except for inserting a passenger arrival exactly at t_0, and repeated it for 1000 replications. In each replication, we can record the TIS of the inserted passenger. Let the TIS recorded from the jth replication be denoted by $Y_j(t_0)$, then the side-experiment estimator of $v(t_0)$ is $\sum_{j=1}^{1000} Y_j(t_0)/1000$. This is not an unbiased estimator of $v(t_0)$ because the arrivals at t_0 are not generated by the original arrival process but inserted externally, however the impact of bias is slight since we run enough

Algorithm 1 KNN method via leave-one-replication-out cross validation (LORO CV)

```
for $j = 1,2,\ldots,n$ do
    $S_{\text{test}} \leftarrow \{Y_{ij} : i = 1,2,\ldots,M_j\}$.
    $S_{\text{train}} \leftarrow$ all data except $S_{\text{test}}$.
    for $k \in [k_L, k_U]$ do
        for $i = 1,2,\ldots,M_j$ do
            Find the $k$ nearest points in $S_{\text{train}}$ to $t_{ij} \in S_{\text{test}}$.
            Compute the KNN estimator $\tilde{V}(t_{ij},k)$.
        end for
    end for
end for
for $k \in [k_L, k_U]$ do
    Compute $\text{ASSE}(k) = \left(\sum_{j=1}^{n}\sum_{i=1}^{M_j}[Y_{ij} - \tilde{V}(t_{ij},k)]^2\right) / \left(\sum_{j=1}^{n} M_j\right)$.
end for
Choose $k^\star$ that results in the minimum $\text{ASSE}(k)$.
```

replications and only one external arrival is inserted. Furthermore, to get nearly unbiased estimators for the TIS at a collection of t_0s, we repeat the entire procedure for each t_0 point individually since inserting many external arrivals in one replication significantly changes the original arrival process.

To compare the prediction performance between TB and KNN estimators, we use average predicted-squared error (APSE) as the criterion. Let the prediction of $v(t_0)$ from the rth macro-replication to be denoted by $\tilde{V}_r(t_0)$, then $\text{APSE}(t_0)$ is $\sum_{r=1}^{R}\left[\tilde{V}_r(t_0) - v(t_0)\right]^2/R$, where the unknown $v(t_0)$ is replaced by its nearly unbiased estimator obtained from the side-experiment in our situation.

This simplified airport check-in model is implemented in Simio. We made $R = 10$ macro-replications, and each of them has $n = 50$ replications. We saved all the arrival times and the corresponding TIS, and chose one-hour time buckets synchronized to the changes of arrival rate and agent capacity, that is, [6:00, 7:00], [7:00, 8:00], ..., [21:00, 22:00]. We used the average TIS of all arrivals within each time bucket as the TB estimator for that time bucket, as in Smith and Nelson (2015), and tested two different sets of t_0s: one of them contains t_0s that are close to the beginning of each time bucket: $t_0 = 6{:}06, 7{:}06, \ldots, 21{:}06$, and the other set contains the midpoints of each time bucket: $t_0 = 6{:}30, 7{:}30, \ldots, 21{:}30$.

The optimal k evaluated via LORO CV is 1980, which means that we would use the average TIS of the 1980 nearest neighbors of t_0 as the prediction for $v(t_0)$. We see that the KNN estimator works better than the TB estimator for most of t_0s when $t_0 = 6{:}06, 7{:}06, \ldots, 21{:}06$ from Figure 1.

For the second set of t_0s which is favorable to the TB estimator, the KNN estimator is still very competitive even with the single $k^\star = 1980$, as shown Figure 2. For the TB estimator, we have chosen the midpoints of time buckets so it is hard to further improve the prediction performance of TB estimator. However, the KNN estimator only uses a single $k^\star$ for all prediction points, which is not necessarily the optimal for any specific point t_0. This implies that if we used $k^\star(t_0)$ adapted to each t_0 of interest instead of a single optimal $k^\star$, then the KNN estimator could have even better prediction performance. Figure 3 illustrates how much the KNN estimator can improve by using an adaptive $k^\star(t_0)$ for all prediction points, where $t_0 = 6{:}30, 7{:}30, \ldots, 21{:}30$.

We find that $k^\star(t_0)$ is very different from $k^\star$ at some time points. For example, $k^\star(16{:}30) = 233 \ll k^\star = 1980$. By plotting one retained sample path shown in Figure 4, we see a strong decreasing trend in the TIS of those passengers who arrive from 15:30 to 18:00. Recall the optimal k derived from the stylized model in (7), a strong trend in TIS like what happen from 15:30 to 18:00 implies a large β in the stylized model, that is why a smaller $k^\star(16{:}30)$ is preferred to reduce bias. In fact, the TB estimator also did not predict well because it averaged too many observations to predict the TIS at $t_0 = 16{:}30$. The arrival rate from 16:00 to 17:00 is 75 per hour and 50 replications are involved, so on average 3750 observations were

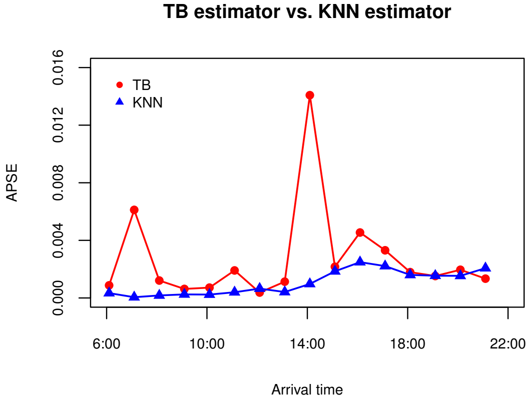

Figure 1: APSE(t_0) of TB estimator and KNN estimator with a single $k^\star$ at $t_0 = 6{:}06, 7{:}06, \ldots, 21{:}06$.

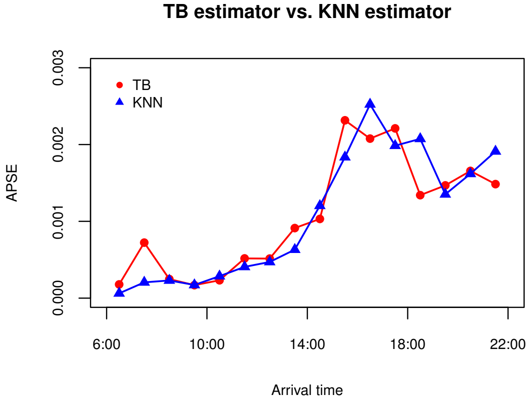

Figure 2: APSE(t_0) of TB estimator and KNN estimator with a single $k^\star$ at $t_0 = 6{:}30, 7{:}30, \ldots, 21{:}30$.

used to construct the TB estimator, which caused large bias so that its prediction performance is worse than the adaptive KNN estimator, according to Figure 3.

From this simplified airport check-in example, we see that the proposed KNN estimator is very competitive with the TB estimator even if we look at the midpoints of time buckets and only use a globally optimal k. Apparently, the prediction performance will be better if we look at t_0s which are not favorable to the TB estimator or let the KNN approach be adaptive to different time points.

Both TB estimation and KNN can be adapted to different time points of interest, but they adapt in different ways. A TB estimator only accounts for the density of arrivals in each time bucket, while an adaptive KNN estimator also takes the trend in the response into account. A TB estimator may predict as well as adaptive KNN if the trend is dominated by noise because averaging a large quantity of data without a strong trend reduces the variance. See the prediction performance and sample path at time $t_0 = 8{:}30$ in Figure 3 and Figure 4 as an example. However, if the trend dominates the noise, as at $t_0 = 16{:}30$, then the adaptive KNN works much better. Hence, even though the TB estimation is also a self-tuning approach, it may not tune in the right way while the KNN method is more adaptive because it accounts for more data features.

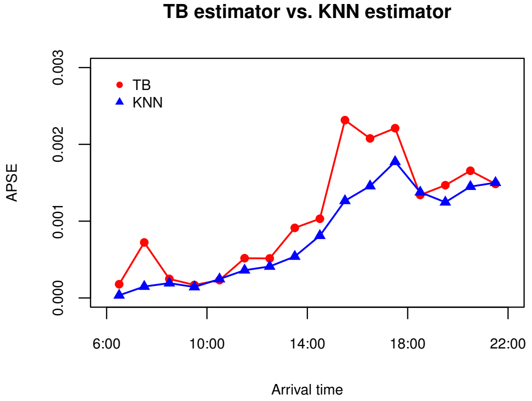

Figure 3: $\text{APSE}(t_0)$ of TB estimator and KNN estimator with adaptive $k^\star(t_0)$ at $t_0 = 6{:}30, 7{:}30, \ldots, 21{:}30$.

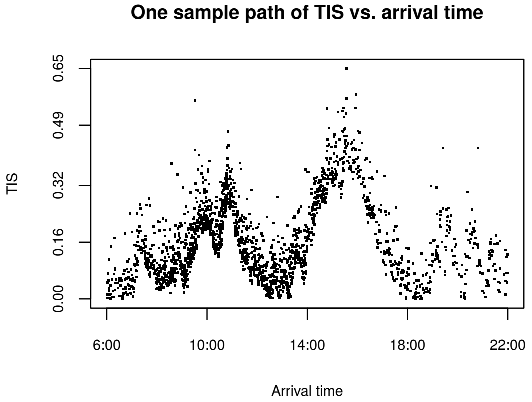

Figure 4: One retained sample path from airport check-in model.

Smith and Nelson (2015) envisioned the TB estimates being computed "on the fly" during the simulation run using predetermined time buckets; this requires little additional computation, but some reprogramming of the simulation. If computed after the run then both TB (single time bucket) and KNN (single value of k) require $O(N\log N)$ computational effort to sort the outputs by arrival time, where $N = \sum_{j=1}^{n} M_j$ is the total number of observations. However, to check all possible values of k via LORO CV is an $O(N^2)$ calculation, which is why trying to find a good starting value of k is so important.

Also notice that in this example arrival rate $\lambda(t)$ and agent capacity $s(t)$ change synchronously each hour, so it is very natural to make each time bucket one-hour long. Choosing a proper time bucket is not trivial in a general case when $\lambda(t)$ may change continuously or $\lambda(t)$ and $s(t)$ change asynchronously; the KNN estimator does not have such issues because k is data driven.

7 CONCLUSIONS

In this paper we propose a KNN method for estimating virtual statistics based on retained transactional data from simulation experiments. We show the asymptotic properties of the KNN estimator for the Poisson arrival case, and we will establish similar properties for general arrival processes as future work. The airport check-in example shows that even with a single $k^\star$, the KNN estimator can be as good as the TB estimator where the latter one does well, and can perform better where the TB estimator does poorly. Further, the KNN estimator can be improved with individually tuned $k^{\star}(t_0)$.

Our stylized approximation of the optimal tuning parameter $k^{\star}$ provides insight regarding how $k^{\star}$ should change according to the features of the output data, as illustrated in the airport check-in example. However, to make searching for $k^{\star}$ computationally feasible, we need a robust starting k, or at least a modest range of k within which to search, since an exhaustive search and CV calculation is $O(N^2)$. Our algorithm includes such a range, $[k_L, k_U]$, but in the experiments we tried all feasible values of k since setting this range, perhaps using our stylized model, is still an open problem.

ACKNOWLEDGEMENTS

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

REFERENCES

Bertsimas, D., and N. Kallus. 2014. "From Predictive to Prescriptive Analytics". *arXiv preprint arXiv:1402.5481*.

Carter, G., and E. J. Ignall. 1975. "Virtual Measures: A Variance Reduction Technique for Simulation". *Management Science* 21 (6): 607–616.

Devroye, L. 1981. "On the Almost Everywhere Convergence of Nonparametric Regression Function Estimates". *The Annals of Statistics* 9 (6): 1310–1319.

Hart, J. D. 1991. "Kernel Regression Estimation with Time Series Errors". *Journal of the Royal Statistical Society. Series B (Methodological)* 53 (1): 173–187.

Nelson, B. 2016. "'Some Tactical Problems in Digital Simulation' for the Next 10 Years". *Journal of Simulation* 10 (1): 2–11.

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.

Walk, H. 2010. "Strong Laws of Large Numbers and Nonparametric Estimation". In *Recent Developments in Applied Probability and Statistics*, 183–214. Springer.

Wolff, R. W. 1982. "Poisson Arrivals See Time Averages". *Operations Research* 30 (2): 223–231.

Wolff, R. W. 1989. *Stochastic Modeling and the Theory of Queues*. Prentice Hall.

AUTHOR BIOGRAPHIES

YUJING LIN is a Ph.D. candidate of the Department of Industrial Engineering and Management Sciences at Northwestern University. Her research interest is simulation analytics. 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. He is a Fellow of INFORMS and IIE. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems, and he is the author of *Foundations and Methods of Stochastic Simulation: A First Course*, from Springer. His e-mail address is nelsonb@northwestern.edu.