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

SIMULATION-BASED PREDICTIVE ANALYTICS FOR DYNAMIC QUEUEING SYSTEMS

Huiyin Ouyang

Innovation and Information Management, School of Business
The University of Hong Kong
Hong Kong

Barry L Nelson

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

ABSTRACT

Simulation and simulation optimization have primarily been used for static system design problems based on long-run average performance measures. Control or policy-based optimization has been a weakness, because it requires a way to predict future behavior based on current state and time information. This work is a first step in that direction with a focus on congestion measures for queueing systems. The idea is to fit predictive models to dynamic sample paths of the system state from a detailed simulation. We propose a two-step method to dynamically predict the probability that the system state belongs to a certain subset and test the performance of this method on two examples.

1 INTRODUCTION

Analysis methodology research has made significant advances on a number of output analysis problems that are important to practitioners: error estimation (e.g., confidence intervals); optimization; metamodeling; and variance reduction in settings for which it is essential (e.g., rare-event simulation). One thing common to all of these settings is identification of one or more summary performance measures of interest, such as expected delay in queue, long-run average system availability or the probability of default. Notice that the performance measures in these examples are all *static*—time matters only in the horizon over which averaging occurs—and *unconditional*, by which we mean they are not a dynamic function of the evolving system state (even in metamodeling the independent variables are usually static system design parameters). See Nelson (2016) for some history as to why analysis methodology research developed in this way.

Contrast this with research in machine/statistical learning, where the focus is typically on *conditional* statements; e.g., how likely is a customer to purchase product A given they have characteristics B, C, D and E? This is possible because of very large volumes of transactional data so that the observed (response, condition) pairs fill the space of interest.

Discrete-event simulations also generate large volumes of (dynamic) transactional data, and although not common now, detailed sample paths from large-scale simulations could be retained without approaching the dataset sizes that are routine in machine/statistical learning. As far as we know, all commercial simulation products are capable of generating a time-stamped trace of what happens at each event time. Although it is currently used primarily for debugging, we anticipate (and advocate) storing such a trace in a database in such a way that it could be easily queried. We see at least two types of analysis that this could support:

- Deeper understanding of a system's dynamic behavior: Having the best long-run average performance does not mean that a system design has acceptable hour-to-hour performance, and an on-average inferior system might have less variable, and more desirable, time-dependent behavior.
- Dynamic prediction of future system behavior based on current time and state information: Many (maybe most) real systems are *managed* to avoid undesirable behavior, such as excessive congestion or near system failure. At present, simulation optimization is better at static design decisions than it is at creating dynamic control policies (Nelson 2013). However, one precursor to system control is the ability to provide time- and state-dependent predictions of future system behavior, and this is facilitated by retaining sample output paths from simulation models.

In this paper we present some initial proof-of-concept results for developing such predictive models in the queueing network context; many simulation models are essentially networks of queues. We use a standard technology: logistic regression. However, the presence of both *time* and *state* as predictors creates a challenge that we address via an innovative two-pass metamodeling approach. We illustrate this idea on a simple network of two queues, using the results to identify strengths of the approach and where further work is needed.

2 MODEL

In this section we describe the specific problem of interest.

2.1 Problem Description

For a queueing system/network, possibly with multiple classes of customers, let $\vec{X}(t) = [X_1(t), \ldots, X_q(t)]$ denote the system state information observed at time $t \geq 0$. Suppose the system state is $\vec{x}_0$ at some given time t_0, and we would like to predict the probability that this system might be in trouble and need management intervention anytime in the future. Specifically, we are interested in the probability of an event $\{h(\vec{X}(t_0+t)) \in \mathscr{A}\}$, which depends on the state we observed at time t_0 and the future time t, and can be expressed as follows:

$$p(\vec{x}_0, t) = \Pr\{h(\vec{X}(t_0+t)) \in \mathscr{A} | \vec{X}(t_0) = \vec{x}_0\}. \tag{1}$$

For general queueing systems, we might be interested in the probability of the queue being blocked for a finite capacity queue; or the system having too many customers waiting for an infinite capacity queue; or identifying the bottleneck station in a queueing network (by predicting server utilization at each node); or the probability that the system will visit a specified set of states before time t (predicting the probability of first passage to the specified set of states is less than t), etc. These probabilities could be represented in our framework by defining the event expression $\{h(\vec{X}(t)) \in \mathscr{A}\}$ accordingly. The following are some examples of $\{h(\vec{X}(t)) \in \mathscr{A}\}$ when the state vector $\vec{X}$ represents the number of customers in each queue node:

- $\{X_i(t) \geq l$ for some node $i\}$, which indicates the total number of customers at a certain queue node exceeds a given threshold l. The value of l might be the number of servers at station i in which case $X_i(t) \geq l$ means all servers are busy, or l denotes the total capacity at station i and $X_i(t) \geq l$ means queue node i is blocked.
- $\{\sum_{i \in Q} X_i(t) \geq l\}$ for some collection of nodes Q in the network, which indicates the total number of customers at the collection of nodes exceeds a specified threshold.
- $\cup_{i \in Q}\{X_i(t) \geq l_i\}$ for some collection of nodes Q in the network, which indicates the number of customers at each node in the collection exceeds its specified threshold simultaneously.
- $I\{\sum_{i \in Q} X_i(\tau) \geq l$ for some $t_0 \leq \tau \leq t\} = 1$, which indicates that the system has exceeded the threshold before time t.

For the kinds of systems we simulate, the probability in (1) is very difficult to compute numerically, if not impossible. For simple queueing systems that can be modeled as finite-state Markov Chains (such as an $M/M/s/K$ queue) , we might be able to obtain these transient probabilities via numerical integration of Kolmogorov equations. However, for complicated systems with many queueing nodes, multiple classes of customers, infinite capacity or non-Markovian behavior, modeling as a Markov chain is infeasible. Hence, we would like to estimate the probability from simulation sample path data.

2.2 Sample Path Data from a Discrete-Event Simulation

Suppose a discrete-event simulation model of the system is readily available, and we keep the sample path data generated in every replication. More specifically, in each simulation run, we record the event time T_j and all or part of the system state information at theses times $\vec{X}(T_j)$ for $j = 1,2,\ldots,M$. Then, the sample path data for a typical simulation replication has the following form:

$$\{T_j, \vec{X}(T_j), j = 1,2,\ldots,M\}. \tag{2}$$

The idea is to predict (1) using N independent observations of (2), which could be obtained by running N simulation replications with different random seeds.

2.3 An Example: Erlang Loss System

We illustrate our problem by an example of the Erlang loss system, i.e., an $M/M/c/c$ queue. Suppose a system has c servers and no waiting space, and customers arrive to the system according to a Poisson process with rate a and the service times are exponentially distributed with rate 1. Customers arriving to a full system are rejected. For this Markovian queue the complete system state at time t can be denoted by an integer in $S = \{0,1,\ldots,c\}$ representing the number of customers in the system. Figure 1 shows a graphical view of six independent replications of sample path data generated in a simulation of an $M/M/10/10$ queue with arrival rate $a = 10$ and different initial states.

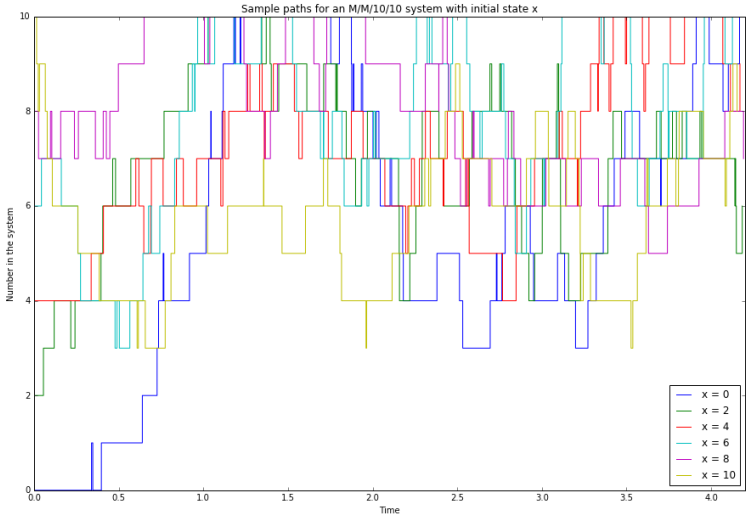

Figure 1: Simulation sample paths for an $M/M/10/10$ queue.

At a given time t_0, we observe the system in state x, and we would like to estimate the probability of the system being blocked t time units after t_0, i.e.,

$$p(x,t) = \Pr\{X(t_0+t) = c | X(t_0) = x\} \text{ for } t \geq 0, x \in S.$$

Of course, for this simple system, $p(x,t)$ can be computed from Kolmogorov Equations.

3 TWO-STEP PREDICTIVE METHOD

Estimating (1) directly by computing the empirical probabilities from simulation sample path data may not work since the combination of $(t_0, \vec{X}_0)$ of interest might never be observed (especially for a high-dimensional state space). Therefore, we propose a regression method, where the key idea is utilizing simulation to generate many sample paths and obtaining a regression model to compute the probability for real-time dynamic prediction for any state at any time.

Logistic regression is a well-understood and natural model for estimating probabilities. However, the probability in (1) depends on both the state variable $\vec{X}$ and the time variable t, and fitting $\vec{X}, t$ simultaneously will be noisy and difficult to identify contributions. The natural model of having time-dependent coefficients on basis functions of the state variables seems difficult to construct except in special cases.

When estimating the probability for a fixed time, logistic regression has been shown to be very effective with properly chosen basis functions of the system state variables. For example, Jiang et al. (2016) apply logistic regression in a fixed-time classification problem with simulation sample path data and show the effectiveness of such a method in a risk prediction problem. When dealing only with time, there are many time-series methods that provide predictions based on time alone. However, it is rare to deal with system state information and time together. In this paper, we propose a two-step prediction method to evaluate the effect of the two factors, where we use state variables as predictors in the first step for fixed time epochs, and then we regress on time in the second step for any given state.

To isolate the state variable and time, we first study how the probabilities depend on the state information at fixed times. We choose a set of time epochs $\mathscr{T}$, and predict the probabilities at those times from the states we observed. In this step, we perform logistic regressions and obtain maximum likelihood estimators of the parameters at every time epoch in $\mathscr{T}$. Obviously, the estimated parameters would depend on time, but analyzing the relation between the estimated parameters and time might be very difficult and would only work for the specific model (depending on what basis functions of state we choose), and hence it could not provide a general framework. Instead we evaluate the relation between the estimated probabilities obtained from the logistic regression models and time. We study how these estimated probabilities depend on time by a weighted least squares regression. To maintain the probability property, we use the logit function of the estimated probability (which ranges from $-\infty$ to ∞) as the response variable and perform a linear regression on basis functions of t in the second step. We use the weighted least squares method to obtain the parameters.

Our problem can be viewed as spatio-temporal metamodeling for which there is a substantial literature; see for instance Diggle (2013) and Mateu et al. (2015). At a high level this approach is a type of Gaussian process regression or kriging that extends the random field to include time. While potentially relevant to our problem, specification of covariance functions that represent the interaction of space and time is difficult, while our two-phase approach supports simple, easily estimable and testable metamodels in each phase, and is particularly useful when the response of interest is a probability.

Next, we describe the two-step method in detail in the following subsections.

3.1 Step One: Logistic Regression on System States

In the first step, we would like to predict the probability $p(\vec{X}, t)$ at a given set of time epochs $\{t_1, t_2, \ldots, t_m\}$. At each fixed time, we predict the probability from some basis functions of the state vector by logistic regression (LR), which is a natural approach for predicting probabilities. The basis functions of the state vector, denoted by $\vec{\Phi}(\vec{X}) = \left(\phi_1(\vec{X}), \phi_2(\vec{X}), \ldots, \phi_{d_\beta}(\vec{X})\right)$, are not obvious, and we will suggest some functions later in our examples.

Let $\{\vec{X}^{(r)}(T_j^{(r)}), j = 1, 2, \ldots, M^{(r)}\}$ denote the sample path data from replication r for $r = 1, 2, \ldots, N$. At any time t, the system state in replication r can be obtained by $\{\vec{X}^{(r)}(t) = \vec{X}^{(r)}(T_j^{(r)}) : T_j^{(r)} \le t < T_{j+1}^{(r)}\}$. Thus, we could obtain the system state at time t_0 for all replications; denote them by $\underline{\mathbf{X}} = (\vec{X}_0^{(1)}, \vec{X}_0^{(2)}, \ldots, \vec{X}_0^{(N)})'$ where

$\vec{X}_0^{(r)} = \vec{X}^{(r)}(t_0)$ for replication r. Similarly, the response vector $\vec{Y}_j = (y_{1j}, \ldots, y_{Nj})'$ at time $t_j \in \{t_1, t_2, \ldots, t_m\}$ can be obtained by

$$y_{rj} = \begin{cases} 1, & \text{if } h\left(\vec{X}^{(r)}(t_0 + t_j)\right) \in \mathscr{A}, \\ 0, & \text{otherwise} \end{cases}$$

for $r = 1, 2, \ldots, N$. We use logistic regression to fit the model for each $j = 1, 2, \ldots, m$:

$$\log \frac{p_j(\vec{X})}{1 - p_j(\vec{X})} = \vec{\Phi}(\vec{X})\vec{\beta}_j, \tag{3}$$

where $p_j(\vec{X}) = \Pr\{y_{rj} = 1\}$. We obtain the maximum likelihood estimators (MLEs) for the parameters with data $\underline{\mathbf{X}}$ and $\vec{Y}_j$ and denote the them by $\hat{\beta}_j$. Notice that there is a different coefficient estimator $\hat{\beta}_j$ for each time epoch t_j.

Finally, for any state $\vec{X}' \in S$, we could provide a step-one probability predictor at the chosen times t_j for $j = 1, 2, \ldots, m$:

$$\hat{p}_j(\vec{X}') = \frac{\exp\left(\vec{\Phi}(\vec{X})\hat{\beta}_j\right)}{1 + \exp\left(\vec{\Phi}(\vec{X})\hat{\beta}_j\right)}. \tag{4}$$

3.2 Step Two: Weighted Least Squares Regression on Time

In this step, our fundamental assumption is that the logit function of the real probability in (1) for a given state has a linear relationship with some basis functions of t, denoted by $\vec{\Theta}(t) = \left(\Theta_1(t), \Theta_2(t), \ldots, \Theta_{d_\gamma}(t)\right)$. We are not able to observe the real probabilities, and hence we use the estimated probability (4) from step one to fit the model. For any state $\vec{X}' \in S$ (which may or may not be observed in N replications of sample path data), we can obtain $\hat{g}(\vec{X}', t_j) = \log \frac{\hat{p}_j(\vec{X}')}{1 - \hat{p}_j(\vec{X}')}$ for $j = 1, 2, \ldots, m$ and fit the model

$$\hat{g}(\vec{X}', t) = \vec{\Theta}(t)\gamma.$$

We fit the model to minimize the weighted least squares, i.e.,

$$\min_{\vec{\gamma}} \quad \sum_j w_j(\vec{X}') \left[\hat{g}(\vec{X}', t_j) - \vec{\Theta}(t_j)\gamma\right]^2, \tag{5}$$

where the weight is given by $w_j(\vec{X}') = \hat{p}_j(\vec{X}')(1 - \hat{p}_j(\vec{X}'))$. The proposed weights indicate our intention to minimize the distance in the probability scale rather than the logit scale.

Let $\hat{\gamma}$ denote the minimizer of weighted least squares problem (5). Then, the predicted probability for state $\vec{X}'$ as a function of t is given by

$$\tilde{p}(\vec{X}', t) = \frac{\exp\left(\vec{\Theta}(t)\hat{\gamma}\right)}{1 + \exp\left(\vec{\Theta}(t)\hat{\gamma}\right)}. \tag{6}$$

Using (6) we could predict the probability in (1) for any state $\vec{X}' \in S$ and $t \geq 0$. Notice that we construct a new model for each state of interest $\vec{X}'$ in Step Two, but since the fit is via least squares it is fast even for large m.

4 EXPERIMENTS

In this section we perform the proposed two-step method on two queueing systems. The first system is the Erlang loss system where we can compute the real probabilities and the second system is a more complex queueing network with two classes of customers.

4.1 Erlang Loss System

First, we perform the method on an Erlang loss system which is introduced in Section 2.3. Let $c = 10$ servers with service rate $\mu = 1$ and the arrival rate $a = 10$. The state space is $S = \{0, 1, \ldots, 10\}$, so we could numerically compute the real probability $p(x,t)$ for any $x \in S$ and $t \geq 0$ (see the Appendix).

To fill up the state space as much as possible, we initialize the simulation differently, and run 50 replications starting from each initial state in the state space (and hence, we have a total of $N = 550$ replications of sample path data).

4.1.1 Step One: logistic regressions

Let $\mathscr{T} = \{0.1, 0.2, \ldots, 4\}$ be the set of time epochs that we select to perform logistic regressions. The selection of $\mathscr{T}$ is a design choice that we do not address in this paper. Let $t_j \in \mathscr{T}$ denote the jth time epoch in $\mathscr{T}$ for $j = 1, 2, \ldots, 40$, and the response vector at time t_j is $\vec{Y}_j = (y_{1j}, y_{2j}, \ldots, y_{Nj})$ where

$$y_{rj} = \begin{cases} 1, & \text{if the system is blocked at time } t_j \text{ in replication } r \\ 0, & \text{otherwise} \end{cases}$$

for $r = 1, 2, \ldots, N$. We suggest basis functions $\vec{\Phi}(x) = (1, x, \sqrt{x+1})$ and we obtain the MLEs $\hat{\beta}_j$ for $j = 1, 2, \ldots, 40$. In the end of Step One, we obtain a functional form of $\hat{p}_j(x)$ for $j = 1, \ldots, 40$.

4.1.2 Step Two: weighted least squares regressions

For an initial state $x' \in S$, we obtain $g(x', t_j)$ from Step One and perform weighted least squares regression on basis functions of t. For this queueing system, we know that the system converges to the steady state, and hence we propose some basis functions that go to 0 as $t \to \infty$. The simplest ones are the inverse of polynomial functions of t and we propose $\vec{\Theta}(t) = \left(1, \frac{1}{t+1}, \frac{1}{(t+1)^2}\right)$ to avoid the basis functions becoming unbounded around $t = 0$. After the weighted least squares regression, we obtain a functional form $\tilde{p}(x', t)$ for any $t \geq 0$.

4.1.3 Prediction results

Figure 2 shows the prediction results from the proposed two-step method, from which we can see that the final predicted probabilities in time (blue solid lines) are very close to the real probabilities (black lines). From Figure 2 we notice that for the plots for $x = 8$ and $x = 9$, the predicted probabilities are not estimating the real probabilities very well for small values of t. We think performing more logistic regressions around $t = 0$ and hence providing more information at small values of t might help improve the estimation.

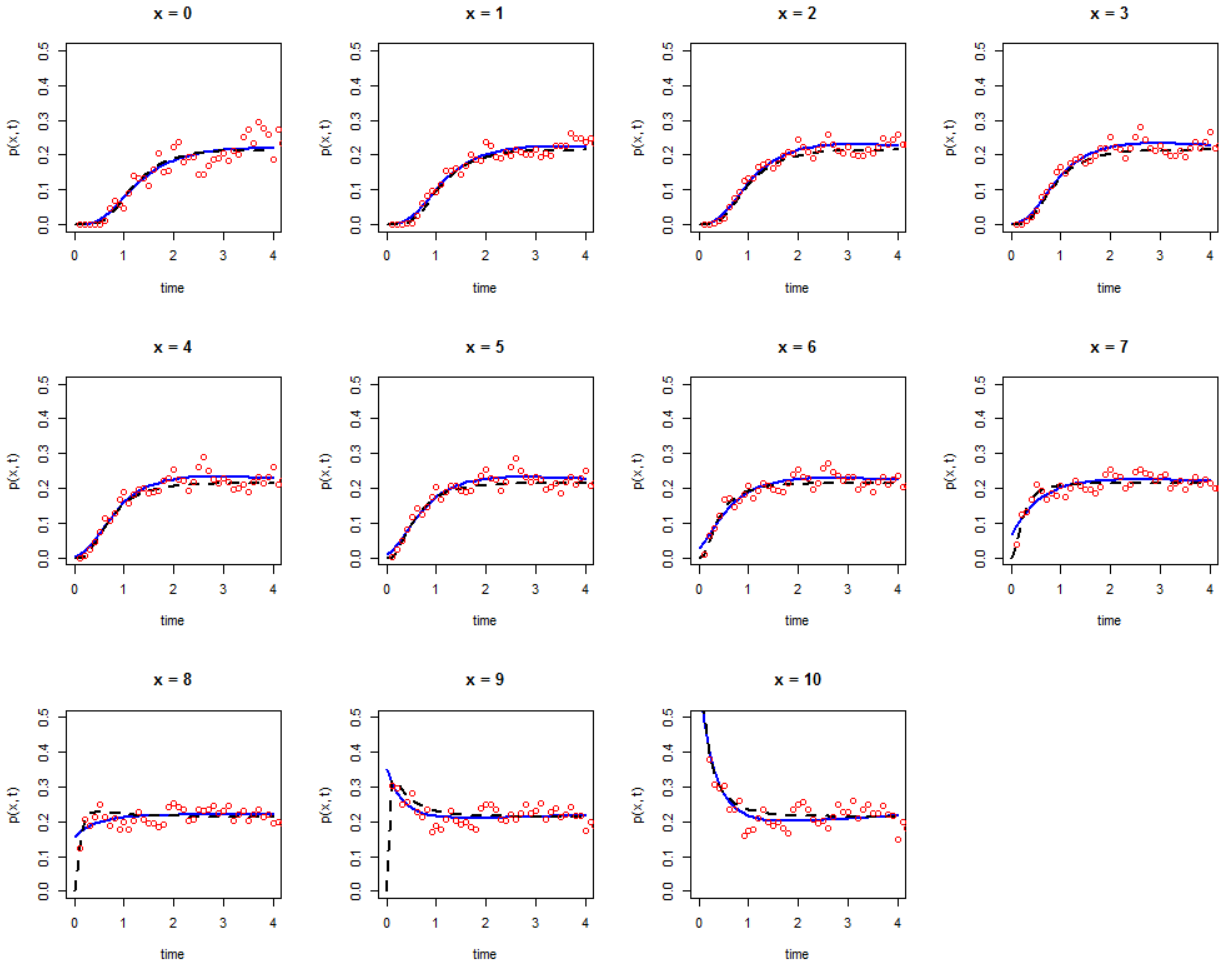

Figure 2: Two-step predictive method on the Erlang loss system, where the black dashed lines denote the true probabilities, the red dots are the estimated probability from Step-One logistic regression, and the blue solid lines represent the estimated probability from the proposed two-step method.

4.2 A Queueing Network with Two Classes of Customers

In this section, we consider a tandem queue with feedback. Figure 3 shows this model.

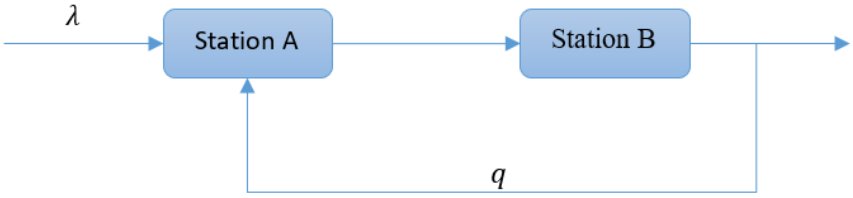

Figure 3: A tandem queue with feedback.

In this system, we have two stations A and B and two classes of customers. For station $I \in \{A,B\}$, we assume there are s_I identical servers, a total of c_I capacity ($c_I \geq s_I$), and the service times for class $k \in \{1,2\}$ customers are exponentially distributed with rate μ_{Ik}.

Customers of class k arrive to station A according to a Poisson process with rate λ_k, and after service completion at station A, the customers will continue to station B if there is an available space or otherwise they will stay at station A (without releasing the servers) until station B frees up a space. After service completion at station B, the customer will go back to station A with probability q for another service sequence or leave the system with probability $1-q$. The subset of the full system state that we consider is the 4-dimensional vector $\vec{X}(t) = (x_{A1}, x_{A2}, x_{B1}, x_{B2})$, where x_{Ik} denotes the number of class $k \in \{1,2\}$ customers at station $I \in \{A,B\}$. Notice that this is not a full characterization of the system state as it does not specify the class of customers in service. We do this to see how effectively we can predict with more aggregated predictors. The state space for the predictors is

$$S = \left\{(x_{A1}, x_{A2}, x_{B1}, x_{B2}) : x_{Ik} \in \{0,1,\ldots c_I\} \text{ and } x_{I1} + x_{I2} \leq c_I \text{ for } I \in \{A,B\} \text{ and } k \in \{1,2\}\right\}.$$

In this system, blocking at station A could be defined differently:

(i) If the capacity of station A is full, then no customers can enter station A.

(ii) If the capacity of station B is full, then no customers can leave station A (servers at station A might still be able to take new customers into service if not all servers are occupied).

(iii) If the capacity of station B is full and all servers at station A are occupied (either serving customers or have finished service but cannot release the customers due to no space available at station B), then station A could not take new customers into service.

We consider a system that has infinite capacity at the first station and has finite capacity $c_B = s_B$ at the second station, and we are interested in estimating the probability of Station A being blocked as described in case (iii) above, i.e., $\Pr\{x_{A1} + x_{A2} \geq s_A \text{ and } x_{B1} + x_{B2} = s_B\}$.

We simulate the system with the following set of parameters: $\lambda_1 = \lambda_2 = 4, q = 0.2, s_A = 4, c_A = \infty, s_B = c_B = 8, \mu_{A1} = 3, \mu_{A2} = 3.5, \mu_{B1} = 1.5, \mu_{B2} = 1$. In this simulation, we initialized the system randomly so that we could fill up the state space we could have possibly observed at time t_0 as much as possible. Let $t_0 = 1$ and we perform logistic regressions at time $t_0 + t_j$ for $t_j \in \mathscr{T}$ where $\mathscr{T} = \{0.1, 0.2, \ldots, 9\}$ in Step One. We suggest the basis functions to be $\vec{\Phi}(\vec{X}) = \left(1, x_{A1}, x_{A2}, x_{B1}, x_{B2}, \sqrt{x_{A1}+1}, \sqrt{x_{A2}+1}, \sqrt{x_{B1}+1}, \sqrt{x_{B2}+1}\right)$. In Step Two we suggest $\vec{\Theta}(t) = \left(1, \frac{1}{t+1}, \frac{1}{(t+1)^2}\right)$ as basis functions for t.

We could not compute the true probabilities numerically for this system due to the high-dimensional and infinite-size state space. Nor can we compute the steady-state probabilities for the same reason, even though we know this system will converge to steady state. Instead, we can estimate the steady-state probability of interest using simulation. We run the model for some extra replications for a longer time and truncate a specific length of time as a warm-up period, and estimate the steady-state probability and 95% confidence interval (C.I.). Notice the steady-state probability is independent of $\vec{X}$ and t. After simulation and computation, the estimation of the steady-state blocking probability is 0.67, with 95% C.I. of $(0.6534, 0.6865)$.

We predict the dynamic probability from nine specified states and show the results in Figure 4, where the red dots represent predictions from Step-One logistic regressions at given times, and the blue solid lines represent the predicted probabilities $p(\vec{X}, t)$ for given $\vec{X}$. The black dashed lines are the mean of the steady-state probabilities from simulation, and the gray areas represent the lower bound and upper bound of the 95% C.I..

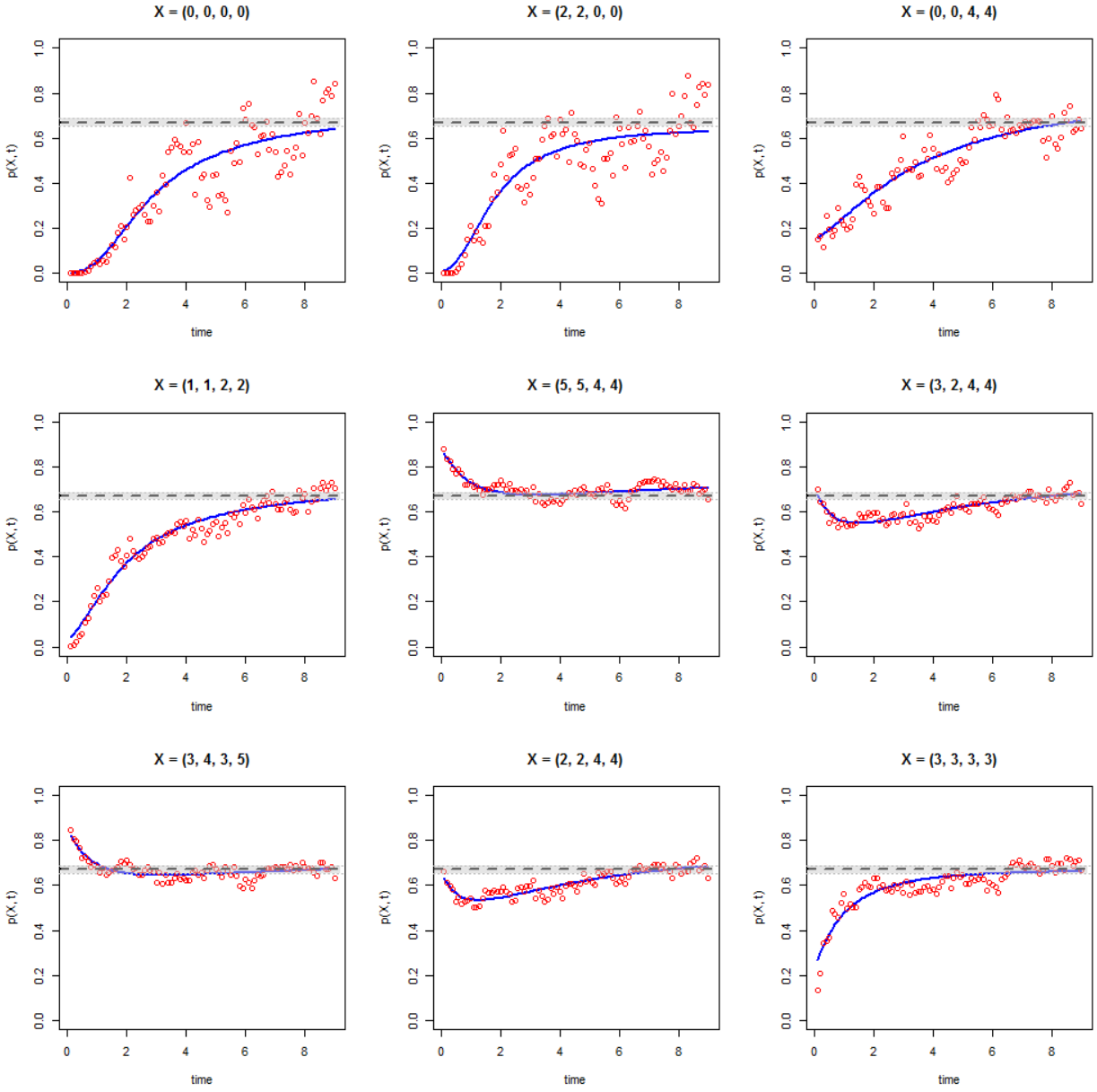

Figure 4: Two-step predictive method on the tandem queue with feedback, where the red dots denote the Step-One predictions, the blue lines denote the prediction from the two-step method, the black dashed lines represent the steady-state probability, and the gray areas represent the 95% C.I. of the steady-state probability.

In Figure 4, we can see that the predicted probabilities converge to the steady-state probability as t becomes large for all $\vec{X}$. For states where station A is blocked at the beginning (e.g., $(5,5,4,4)$ and $(3,4,3,5)$), the predicted blocking probabilities for small values of t are very large, and then decreasing to the steady state probability, while for states where station A is not blocked (e.g., $(0,0,0,0)$ and $(2,2,0,0)$), the predicted probabilities increase from 0 to steady-state probability as t increases. These predictions

capture both the long-run converging behavior and the short-term marginal behavior, and hence we conclude that the proposed two-step method works well for this queueing network.

5 CONCLUSIONS

The era of big data has created many opportunities and challenges in simulation. Instead of running a simulation model to obtain only some overall system performance measures, we can store all the useful information generated during the simulation at a low cost and utilize this information for dynamic system analysis. Our work is a first step toward using predictive analytics based on data generated during simulation.

It is difficult to deal with both time and state in a single regression model, especially with the interaction between them. We propose a two-step method to dynamically predict the probability of system state information at any future time based on the current observed state. We use simple models in each step and deal with state and time separately. The basis functions we suggest in the model are very simple functions of the state information and time, but they work well in our experiments. Although the examples we present in this paper consider only time-homogeneous systems, we do think our method will also work well in non-homogeneous systems. For instance, we have tested our method for an $M(t)/M/c/c$ queue where the arrival process is a non-homogeneous Poisson process with rate $a(t)$ and find that the two-step method estimates the true probabilities well after adding $1/a(t)$ into the basis functions in Step Two.

This paper is a proof-of-concept work towards application of analytics methods to simulation sample path data, where our main contributions include description of the problem context and motivation, and the two-step method in analyzing simulation sample paths that include both state and time information. We also suggest possible basis functions of state (when system states are described by the number of customers) and time (especially for converging systems).

This work could be extended theoretically in several ways. First, we select $\mathscr{T}$, the set of time epochs for logistic regressions in Step One, equally spaced from 0.1 to 4 by 0.1 in our experiments. However, this should be a design problem where we can investigate how the choice of $\mathscr{T}$ will affect our prediction and suggest "better" ways to choose time epochs in $\mathscr{T}$. Comparisons of different choices require us to be able to quantify the prediction error or the goodness-of-fit for our method. Secondly, we can provide statistical analysis of Step-Two weighted least squares regression, assuming the input, $\hat{g}$, are computed based on true probabilities. The analysis will enable us to evaluate the performance of suggested basis functions and select the best set of basis functions for the specific systems. Last but not least, the simulation sample path data from the same replication are auto-correlated, i.e., system states at different time epochs t and t' are not independent especially when t and t' are close to each other. Analysis of the auto-correlation in time might help improve the predictions.

ACKNOWLEDGMENTS

This research is supported by the National Science Foundation Grant CMMI-1537060 and SAS Institute.

A APPENDIX

We formulate the $M/M/c/c$ queue in Section 4.1 as a continuous time Markov chain (CTMC) and obtain the generator matrix as follows:

$$Q = \begin{bmatrix} -a & a & 0 & \dots & 0 & 0 & 0 \\ 1 & -(a+1) & a & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & c-1 & -(a+c-1) & a \\ 0 & 0 & 0 & \dots & 0 & c & -c \end{bmatrix}$$

Let $p_{xy}(t) = \Pr(X(t+t_0) = y \mid X(t_0) = x)$ and $P(t) = [p_{xy}(t)]_{x,y \in S}$. Then from the Kolmogorov backward Equations (see, e.g. Theorem 6.4 in Kulkarni (2009)), we have

$$\frac{d}{dt}P(t) = QP(t).$$

We can obtain $P(t)$ by numerically integrating the equations above with the initial condition $P(0) = I$. Then, $p(x,t) = P_{xc}(t)$ is our desired probability. Figure 5 shows the probability of blocking after t time units starting from state x_0 for such a queueing system.

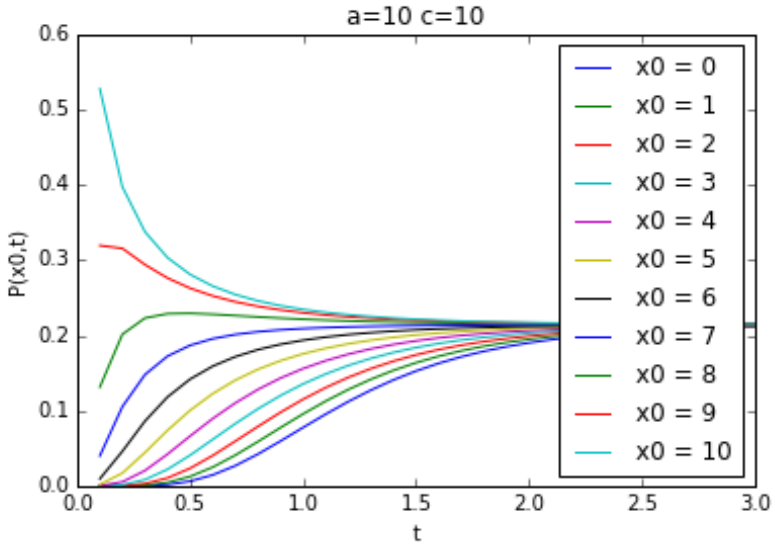

Figure 5: Probability of blocking after t time units starting from state x_0 for an $M/M/10/10$ queue.

REFERENCES

Diggle, P. J. 2013. *Statistical Analysis of Spatial and Spatio-temporal Point Patterns*. Boca Raton: CRC Press.

Jiang, G., L. J. Hong, and B. L. Nelson. 2016. "A Simulation Analytics Approach to Dynamic Risk Monitoring". In *Proceedings of the 2016 Winter Simulation Conference*, edited by P. Frazier, T. M. Roeder, R. Szechtman, and E. Zhou, 437–447: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kulkarni, V. G. 2009. *Modeling and Analysis of Stochastic Systems*. Boca Raton: CRC Press.

Mateu, J. et al. 2015. *Spatial and Spatio-temporal Geostatistical Modeling and Kriging*, Volume 998. Chichester: John Wiley & Sons.

Nelson, B. 2013. *Foundations and Methods of Stochastic Simulation: A First Course*. New York: Springer Science & Business Media.

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

AUTHOR BIOGRAPHIES

HUIYIN OUYANG is an assistant professor in the School of Business at the University of Hong Kong. She was a postdoctoral fellow in the Department of Industrial Engineering and Management Sciences at Northwestern University before joining the University of Hong Kong. She holds a Ph.D. in Statistics and

Operations Research from the University of North Carolina at Chapel Hill. Her research interests include modeling and analysis of stochastic systems, and simulation analytics with applications in service and health care management. Her email address is huiyin.ouyang@hku.hk.

BARRY L NELSON is the Walter P. Murphy Professor of the Department of Industrial Engineering and Management Sciences at Northwestern University. 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 email address is nelsonb@northwestern.edu.