Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

CALIBRATING INFINITE SERVER QUEUEING MODELS DRIVEN BY COX PROCESSES

Ruixin Wang
Harsha Honnappa

School of Industrial Engineering
Purdue University
West Lafayette, IN 47906, USA

ABSTRACT

This paper studies the problem of calibrating a Cox$/G/\infty$ infinite server queue to a dataset consisting of the number in the system and the age of the jobs currently in service, sampled at discrete time points. This calibration problem is complicated owing to the fact that the arrival intensity and the service time distribution must be jointly calibrated. Furthermore, maximizing the finite dimensional distribution (FDD) of the number-in-system process (which is the natural calibration objective) is intractable in this setting, since the computation of the FDDs involves an intractable integration over the path measure of the Cox input process. We derive an approximate inference procedure that maximizes a lower bound to the FDDs using stochastic gradient descent. This lower bound is tight when the calibrated parameters coincide with those of the 'true' model. We present extensive numerical experiments that demonstrate the efficacy and validity of the proposed method.

1 INTRODUCTION

Calibrating stochastic models to a dataset is an important part of modeling and simulation practice. In this paper, we consider the problem of calibrating infinite server queueing models that operate in random environments. Specifically, we assume that the input to the queue is well-modeled by a doubly stochastic Poisson, or 'Cox', process and that the service times of arriving jobs are identically distributed with general service distribution; that is, Cox$/G/\infty$ queues.

These so-called Cox$/G/\infty$ queues have received significant study in the recent past, with particular focus on asymptotic analysis; see Honnappa et al. (2020) and references therein. Infinite server queues are often viewed as crude approximations (after all, a free server is always available and no arriving job must await its turn for service), but they are remarkably useful models for performance analysis of large-capacity service systems. For instance, early in the coronavirus pandemic, ICU capacity for COVID-19 patients (in a hospital network) was essentially unbounded as new capacity was created to accommodate incoming patients. Alternatively, infinite server queues are also useful as demand models in ride-sharing service systems. Consider a customer requesting a taxi ride through a smartphone application-based service. Ride requests can be modeled as a pure delay queue, wherein these requests are immediately put into service, and the (stochastic) amount of time taken for a customer to be matched with a driver represents the "service time" of the passenger in the matching system. It has been empirically observed that traffic in these systems exhibits high variability and over dispersion, suggesting that the traffic intensity (i.e., the *average* number of jobs to arrive per unit time) is stochastic itself (Wang, Jaiwal, and Honnappa 2020; Zhang, Hong, and Zhang 2014). Cox processes represent a tractable family of counting process models, wherein the stochastic traffic intensity is assumed to be 'trivially' adapted to the initial filtration of the counting process.

Model calibration of stationary queueing models is facilitated by the fact that these models are finitely parameterized and, therefore, if precise inter-arrival times and service time data are available, calibration

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

is straightforward. In practice, however data are typically censored, wherein either the inter-arrival or service times are unknown to precision, or only the state of the queue (either the number-in-system or the workload) are available; see Asanjarani et al. (2021) for a classification of the queue observation methods. For instance, in the infinite server setting, it is typically the case that while the number-in-system can be observed at discrete points in time, it may only be possible to record the age of jobs in service, while the precise departure epoch is unknown. This coincides with the suggestion in Pang and Whitt (2010) that the elapsed service time (age) of the customers should be modeled and used. In practice, of course, queues are nonstationary and potentially subject to exogeneous stochasticity, and calibrating models in this setting is a far more complicated problem. In this paper, we consider the problem of jointly calibrating the traffic intensity model and the service time distribution of a Cox$/G/\infty$ queue, where data consists of observations of the number-in-system and (potentially) the age of the jobs in service, collected at discrete time points.

Estimation and calibration of infinite server queueing models with nonstationary input has been studied in the recent literature (Li, Hu, Wang, and Yu 2019; Goldenshluger and Koops 2019). In Li et al. (2019), a maximum likelihood estimator (MLE) of the arrival intensity and service time distributions of an $M_t/G/\infty$ queue. Goldenshluger and Koops (2019) consider the same setting, but focus on the problem of estimating the service time distribution assuming the arrival intensity is known. The authors introduce a nonparametric estimator of the service time distribution (using kernel density estimation methods) and demonstrate that the method achieves the minimax rate. The standard setting of stationary $M/G/\infty$ queues has received significantly more attention in the literature; see Goldenschluger et al. (2016), Schweer and Wichelhaus (2020), Schweer and Wichelhaus (2015) for recent work in this direction and Asanjarani et al. (2021) for a more extensive survey. Furthermore, all of this existing literature considers Markovian traffic models (i.e., $M_t/G/\infty$ queues) with the exception Schweer and Wichelhaus (2015), Schweer and Wichelhaus (2020) where a renewal traffic model is considered. In this paper we consider the considerably harder problem of jointly estimating the *stochastic* arrival intensity model of a Cox input process model and the service time distribution.

Our approach builds on our prior work Wang et al. (2020) that focused on estimating the stochastic intensity process of a Cox process alone. While there are many possible choices for the model of the stochastic intensity, we again assume a Markov diffusion model of the intensity. In other words, we assume that the intensity is well-modeled as the solution of an Ito stochastic differential equation (SDE). Furthermore, we will also assume that the intensity process is ergodic in order to highlight some implications for the calibration method. We assume that the drift and diffusion coefficients of the model are unknown, but modeled by a universal function approximator (UFA) that can be optimized by a gradient-based optimization method. On the other hand, we also model the survival function of the service times by a UFA; more precisely, we model the logarithm of the survival function (also known as the integrated hazard function) by a UFA. In this paper, we will use artificial neural networks (ANNs) as the UFAs and optimize/train the parameters of the ANNs using stochastic gradient descent (SGD). Nonetheless, it should be noted that other choices of UFAs are possible.

We will jointly calibrate the arrival intensity and service time models by maximizing the finite dimensional distributions of the number-in-system process. However, unlike the $M_t/G/\infty$ queue setting extensively studied in the literature, this is not a straightforward likelihood maximization problem, since the number-in-system likelihood function must be further integrated over the path measure corresponding to the stochastic intensity process. In general, this integration is intractable, and direct inference of a closed-form expression is impossible. Building on our prior work in Wang et al. (2020), we present an *approximate inference* method, wherein we introduce a tractable surrogate objective to the integral which can be leveraged to approximately calibrate the infinite server model. To be precise, we introduce a lower bound on the logarithm of the finite dimensional distributions, which is optimize over the parameters of the UFAs. This lower bound is tight precisely when the optimized parameters correspond to the 'true' arrival intensity model and service time distribution. In this paper, we carefully derive the lower bound and then demonstrate how it can be optimized using simualtion optimization. We then provide extensive numerical experimentation that demonstrates the accuracy and efficacy of the proposed method.

The paper is organized as follows: we briefly review the Cox process in Section 2, followed by a discussion of the number-in-system process for the Cox/G/∞ queue in Section 3. We then describe the statistical estimation problem and variational inference in section 4. In Section 5, we present our method by deriving the evidence lower bound (ELBO) and introducing the detailed training procedure. In Section 6, we present our simulation results and discuss prediction accuracy under different model settings. We end with concluding remarks in Section 7.

2 COX PROCESSES

We start by recalling the definition of Cox process. Let $(X(t) : t \geq 0)$ be a non-decreasing $\mathbb{Z}_+$-valued point process, $(X|Z)$ represent the process conditioned on the stochastic process $(Z(t) : t \geq 0)$, and Poi(Λ) represent a Poisson process with integrated intensity function $(\Lambda(t) : t \geq 0)$. Formally, a Cox, or doubly stochastic Poisson process (DSPP), is defined as:

Definition 1 Let $(Z(t) : t \geq 0)$ be a non-negative stochastic process such that with probability one $t \mapsto Z(t)$ is locally integrable. Then, $(X(t) : t \geq 0)$ is a Cox process driven by $(Z(t) : t \geq 0)$ if $(X|Z) \sim$ Poi$(\mathbf{Z})$, where $\mathbf{Z}$ is the integrated process defined as $\mathbf{Z}(s,t) := \int_s^t Z(r)dr$ for any $s < t$.

That is, for any set of points $\{t_0, t_1, \ldots, t_d\} \subset (0, \infty)$, where $0 < t_0 \leq t_1 \leq \cdots \leq t_d < \infty$, the finite dimensional distributions of $(X|Z)$ satisfy

$$\begin{aligned}&\mathbb{P}(X(t_0) = k_0, X(t_1) = k_1, \ldots, X(t_d) = k_d | Z_{0:t_d}\}) \qquad (1)\\ &\qquad = \frac{\exp(-\mathbf{Z}(0,t_0))(\mathbf{Z}(0,t_0))^{k_0}}{k_0!} \prod_{i=0}^{d-1} \frac{\exp(-\mathbf{Z}(t_i,t_{i+1}))(\mathbf{Z}(t_i,t_{i+1}))^{k_{i+1}-k_i}}{(k_{i+1}-k_i)!},\end{aligned}$$

where $Z_{0:t} \equiv (Z(s) : 0 \leq s \leq t)$. Formally, the path measure induced by $(X(t) : t \geq 0)$ is defined as $\int Poi(\mathbf{Z})dP(Z)$, where $P(\cdot)$ is the path measure induced by the stochastic process $(Z(t) : t \geq 0)$), so that the finite dimensional distribution of $X(t)$ (at any fixed $t \geq 0$) satisfies

$$\mathbb{P}\left(X(t_0) = k_0, \ldots, X(t_d) = k_d\right) = \int \mathbb{P}(X(t_0) = k_0, X(t_1) = k_1, \ldots, X(t_d) = k_d | Z_{0:t_d}\}) dP(Z_{0:t_d}). \qquad (2)$$

Note that we are deliberately being less than rigorous in our description of this path measure so as to avoid a heavier notational burden that distracts from the primary message of this paper.

3 INFINITE SERVER QUEUES

We consider a Cox/G/∞ queue where arrivals follow a Cox process with intensity $Z(t)$ and service time has general distribution G. Let $X(t)$ represent the number-in-system of the infinite server queue. It is well known (Eick, Massey, and Whitt 1993) that, starting from an empty system, conditioned on the intensity process $(Z(s) : 0 \leq s \leq t)$, the mean queue length is given by $E[X(t)|(Z(s) : 0 \leq s \leq t)] = \int_0^t [1 - G(t-s)]Z(s)ds$, where $\bar{G}(t-s) =: 1 - G(t-s)$ is the tail probability of the service times. Furthermore, the probability of observing k customers in the system at time t is given by

$$\mathbb{P}(X(t) = k | Z_{0:t}) = \frac{e^{-\int_0^t Z(s)\bar{G}(t-s)ds} \left(\int_0^t Z(s)\bar{G}(t-s)ds\right)^k}{k!},$$

and $Z(s)\bar{G}(t-s)$ can be interpreted as the "survival intensity" at time s.

Consider a Cox$/M/\infty$ queue with exponentially distributed service time. The finite dimensional distribution of the Cox$/M/\infty$ can be derived in multiple ways, see Hillestad and Carrillo (1980) for instance. Let k_1 be the number in the system at t_1. Then, one can easily deduce that for $t_1 < t_2$, the probability that i out of k_1 customers survive from t_1 to t_2 follow a binomial distribution with parameters $(k_1, \bar{G}(t_2 - t_1))$. Let $Y := (X(t_1), X(t_2))$, then the joint distribution for the two-dimensional case can be observed to have conditional distribution $\mathbb{P}\left(Y = (k_1, k_2) | Z_{0:t_2}\right)$

$$=\text{Pois}\left(k_1; \int_0^{t_2} Z(t)[\bar{G}(t_1 - t)]dt\right)\left[\sum_{i=0}^{\min\{k_1,k_2\}} \text{B}(i; \bar{G}(t_2 - t_1))\text{Pois}\left(k_2 - i; \int_{t_1}^{t_2} Z(t)[\bar{G}(t_2 - t)]dt\right)\right], \tag{3}$$

where $\text{B}(\cdot; p)$ refers to the binomial PMF with parameter p and $\text{Pois}(\cdot; \lambda)$ is the Poisson PMF with mean λ.

Before moving to the non-memoryless case, we recall the definition of the Poisson-Binomial distribution

Definition 2 Poisson-Binomial distribution is a discrete probability distribution of the sum of independent Bernoulli trials that are not necessarily identically distributed. Probability mass function (PMF) of the P-B distributed random variable B is given by

$$\mathbb{P}(B = k) = \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j),$$

where F_k is the set of all subsets of k integers that can be selected from the set $\{1, 2, 3, ..., n\}$.

For the Cox$/G/\infty$ system, the conditional distribution is much more complicated due to the non-memoryless nature of the service time distribution. Additional information is required for the joint distribution to be derived explicitly. Let $S_{1:X(t_1)} := \{S_1, ..., S_{X(t_1)}\}$ be the age of the customers in the system at time t_1. Treating $S_{1:X(t_1)}$ as a 'local latent' variable, the joint distribution can be expressed as

$$\begin{aligned}
&\mathbb{P}\left(Y = (k_1, k_2) | Z_{0:t_2}\right) \\
=& \int \mathbb{P}(X(t_1) = k_1, X(t_2 = k_2), S_{1:k_1} \in d\boldsymbol{s} | Z_{0:t_2}) \\
=& \int \mathbb{P}(X(t_2) = k_2 | X(t_1) = k_1, S_{1:k_1} = \boldsymbol{s}, Z_{0:t_2}) \mathbb{P}(X(t_1) = k_1, S_{1:k_1} \in d\boldsymbol{s} | Z_{0:t_2}) \\
=& \int \mathbb{P}(X(t_2) = k_2 | X(t_1) = k_1, S_{1:k_1} = \boldsymbol{s}, Z_{0:t_2}) \mathbb{P}(X(t_1) = k_1 | Z_{0:t_2}) \mathbb{P}(S_{1:k_1} \in d\boldsymbol{s} | X(t_1) = k_1, Z_{0:t_2}).
\end{aligned}$$

To be precise, suppose that there are k_1 customers at time t_1 and let $\boldsymbol{s} := \{s_1, s_2, ...s_{k_1}\}$ be the age of those customers, then, conditioned on the fact that customer i has already survived for s_i, the customer has a further survival probability of $p_i = \mathbb{P}(R \geq t_2 - t_1 + s_i | R \geq s_i) = \frac{\mathbb{P}(R \geq t_2 - t_2 + s_i)}{\mathbb{P}(R \geq s_i)} = \frac{\bar{G}(t_2 - t_1 + s_i)}{\bar{G}(s_i)}$ of surviving to t_2. This fact leads us to treat the existing customers and the new arrivals separately, using the Poisson Binomial distribution (Definition 2). Let X_{t_1,t_2} and V_{t_2} be the new arrivals from t_1 to t_2 and existing customers at time t_1 that are in the system at time t_2 (respectively). We know that V_{t_2} follows Poisson Binomial distribution with parameters $\{p_1, p_2, ...p_{k_1}\}$, and X_{t_1,t_2} follows Poisson distribution with parameter $\int_{t_1}^{t_2} Z(t)[1 - G(t_1 - t)]dt$. It is straight forward that $X_{t_1,t_2} \leq k_2$ since the number of new surviving arrivals cannot exceed the total number in the system at t_2. Meanwhile, we have $V_{t_2} \leq k_1$ by its definition. It follows that,

$$\mathbb{P}\left(X(t_2) = k_2 | X(t_1) = k_1, S_{1:k_1} = s_{1:k_1}\right) = \sum_{i=0}^{\min\{k_1,k_2\}} \text{PB}(i; p_{1:k_1}) \text{Pois}\left(k_2 - i; \int_{t_1}^{t_2} Z(t)\bar{G}(t_2 - t)dt\right).$$

In this paper, we will consider the situation where $S_{1:k_1}$ is observable, implying $\mathbb{P}(S_{1:k_1} = \boldsymbol{s} | X(t_1) = k_1, Z_{0:t_2}) = \delta_{s_{1:k_1}}(\boldsymbol{s})$, where δ is the Dirac measure. The integral above simplifies to

$$\text{Pois}\left(k_1; \int_0^{t_1} Z(t)\bar{G}(t_1 - t)dt\right) \left[\sum_{i=0}^{\min\{k_1,k_2\}} \text{PB}(i; p_{1:k_1})\text{Pois}\left(k_2 - i; \int_{t_1}^{t_2} Z(t)\bar{G}(t_2 - t)dt\right)\right], \tag{4}$$

where $\text{PB}(\cdot; \boldsymbol{p})$ refers to PMF of the Poisson binomial with successs probability vector $\boldsymbol{p}$. In general, computing the Poisson-Binomial is computationally intensive and approximations may be necessary.
Poisson Approximation. The Poisson Binomial can be approximated by a Poisson distribution with parameter $\mu := \sum_{i=1}^{k_1} p_i$ (Le Cam et al. 1960). Error bounds are given under condition $\max_{p_i \in \boldsymbol{p}} p_i \leq 1/4$, indicating that the Poisson approximation works well when the success probabilities are small.
Normal Approximation. The normal approximation of the Poisson Binomial distribution is based on the central limit theorem, and can be a poor approximation if k_1 is small. Let $\sigma := \left[\sum_{i=1}^{k_1} p_i(1 - p_i)\right]^{1/2}$ and Φ be the cdf of the standard normal distribution. Applying to our specific case, we obtain

$$\mathbb{P}(X_{t_2} = V_{t_2} + X_{t_1,t_2} = k_2 | z_{0:t_2}, s_{1:k_1}) \approx \sum_{i=0}^{k_1} \Phi'(\frac{i + 0.5 - \mu}{\sigma}) P(X_{t_1:t_2} = k_2 - i | z_{t_1:t_2}).$$

4 THE MODEL ESTIMATION PROBLEM AND APPROXIMATE INFERENCE

In the setting of a infinite server queue, the model estimation problem amounts to jointly estimating the drift and the service time distribution, parameterized by θ and θ_1, respectively. For simplicity, consider the marginal distribution of $X(t)$

$$\log P_{\theta_1,\theta}\left(X(t) = k\right) = \log \int P_{\theta_1}(X(t) = k | Z_{0:t}) dP_\theta(Z_{0:t}) \tag{5}$$

where P_θ is the path measure corresponding to the parameters θ. The difficulties of estimating the drift and diffusion coefficient using MLE are addressed concisely in Section 3 of Wang, Jaiwal, and Honnappa (2020). Introducing a new measure $P_{\phi,k}(Z_{0:t})$ (equivalent to P_θ), Jensen's inequality implies that

$$\begin{aligned} \log P_{\theta_1,\theta}(X(t) = k) &= \log \int P_{\theta_1}(X(t) = k | Z_{0:t}) \frac{dP_\theta(Z_{0:t})}{dP_{\phi,k}(Z_{0:t})} dP_{\phi,k}(Z_{0:t}) \\ &\geq \int \log\left(P_{\theta_1}(X(t) = k | Z_{0:t}) \frac{dP_\theta(Z_{0:t})}{dP_{\phi,k}(Z_{0:t})}\right) dP_{\phi,k}(Z_{0:t}). \end{aligned} \tag{6}$$

While this is a lower bound, observe that the inequality can be tightened by maximizing over both θ and ϕ (the conditional measure $P(Z_{0:t} | X(t) = k)$ is the "optimal" choice that achieves equality). The lower bound, however, is highly non-concave in these parameters and consequently we can only guarantee the computation of a local optimum. Furthermore, the choice of parameterization will, in general, imply that the class of measures being optimized over may not include the 'true' measures, resulting in an approximation. Therefore, this procedure of optimizing over path measures is an example of *approximate inference*, used extensively in the machine learning literature for approximately solving high dimensional and large sample statistical inference problems, particularly with Bayesian models. See our description of approximate inference in a more general setting in Wang, Jaiwal, and Honnappa (2020).

[Deep latent models (DLMs) (Goodfellow et al. 2016, Ch. 19, 20) parameterize the probability measures by deep neural networks (DNNs). Variational autoencoders (VAEs) (Kingma and Welling 2019) are an example of DLMs in the multivariate setting where the sequence of prior distributions are known only up to the parameters of an appropriately chosen DNN modeling these parameters. In the VAE literature this sequence of prior distributions are also known as *decoders*. The approximating measures $\mathcal{Q}_n$, entitled *encoders* in the VAE literature, are also parameterized using DNNs. Given the ensemble $\mathbf{Y}_n$, the DNN parameters of both the encoder and decoder are estimated using stochastic gradient descent (SGD). Our current setting, of course, is far more complicated than the VAE setting since the DNNs model the drift and diffusion coefficients of SDEs leading to a more involved training procedure.

5 DLMs FOR INFINITE SERVER QUEUES

We assume access to n independent and identically distributed (i.i.d.) observations of a stochastic process $(X(t), S_{1:X(t)} : 0 \leq t \leq T)$. In many service systems, such as hospitals and call centers, traffic counts are collected at fixed, regular intervals; for instance, in many large call centers, this is typically at intervals of length 30 seconds to 1 minute. It has been observed (Zhang et al. 2014) that a Cox process, with CIR-type ergodic diffusion process as intensity, is an appropriate model of the traffic counts at operational time-scales (typically of the order of 10 minutes). The time interval $[0, T]$ in our model represents this operational time-scale.

For clarity of exposition, we will describe our method assuming (i) the traffic counts are observed at the time epochs $T/2$ and T; and (ii) a single sample $n = 1$. These can be extended to more observation instants and samples at the expense of a more burdensome notation, but our method will not change. We model the unknown stochastic intensity process by the SDE

$$dZ(t) = b(Z(t), t; \theta)dt + \eta\sqrt{Z(t)}dW(t), \quad t \leq T \tag{7}$$

where $\{W(t), t \geq 0\}$ is the standard Brownian motion, $b(\cdot, t; \theta) : C_b[0, T] \times [0, T] \mapsto \mathbb{R}$ is the drift and $\eta\sqrt{(\cdot)}$ with $\eta > 0$ is the diffusion coefficient. $C_b[0, T]$ denotes the space of all continuous and bounded function on the interval $[0, T]$. Here, the unknown drift function is modeled using a DNN parameterized by θ, and to avoid getting bogged down in technical detail, we assume the existence of a strong solution to (7). For technical reasons (required by the Girsanov's theorem) we will, for now, assume that the diffusion coefficient is known. We denote the measure induced by the solution of this SDE as $P_\theta(\cdot)$. The independent increments property implies that the joint distribution of the arrival count random vector $Y_1 := \left(X\left(\frac{T}{2}\right), X(T)\right)$, conditional on the intensity process $Z_{0:T}$, can be expressed as: $P_{\theta_1}\left(Y_1 = (k_1, k_2) | Z_{0:T}\right)$

$$= \text{Pois}\left(k_1; \int_0^{\frac{T}{2}} Z(t)\bar{G}(\frac{T}{2} - t; \theta_1)dt\right) \left[\sum_{i=0}^{\min\{k_1,k_2\}} \text{PB}(i; p_{1:k_1}^{\theta_1})\text{Pois}\left(k_2 - i; \int_{\frac{T}{2}}^{T} Z(t)\bar{G}(T - t; \theta_1)dt\right)\right]. \tag{8}$$

Note that we assume access to the age information, i.e., $\mathbb{P}(S_{1:k_1} = \boldsymbol{s} | X(\frac{T}{2}) = k_1, Z_{0:T}) = \delta_{s_{1:k_1}}(\boldsymbol{s})$ as in Equation (4).

5.1 A DLM for the Infinte Server Queue

By definition, the variational family $\mathcal{Q}$ must consist of measures that are absolutely continuous with respect to the 'prior' measure P_θ. In our current setting, $\mathcal{Q}$ is the class of equivalent measures induced by the solutions of SDEs that have the same diffusion coefficient as (7). To be precise, consider the SDE

$$dZ(t) = \bar{b}_{k_1,k_2}(Z(t), t : \phi)dt + \eta\sqrt{Z(t)}dW(t), \text{ for } t \leq T, \tag{9}$$

where for each $(k_1, k_2) \in \{0, 1, \ldots\}^2$ the drift function $\bar{b}_{k_1,k_2}(\cdot, \cdot; \phi)$ is modeled using a DNN with parameter ϕ. We denote the measure induced by the solution of this SDE as Q_ϕ. Figure 1 illustrates the use of deep latent models in defining measures P_θ and Q_ϕ and consequently ELBO. Next, we derive the ELBO for the observation random vector Y_1. The proof, omitted for space reasons, follows from Girsanov's theorem.

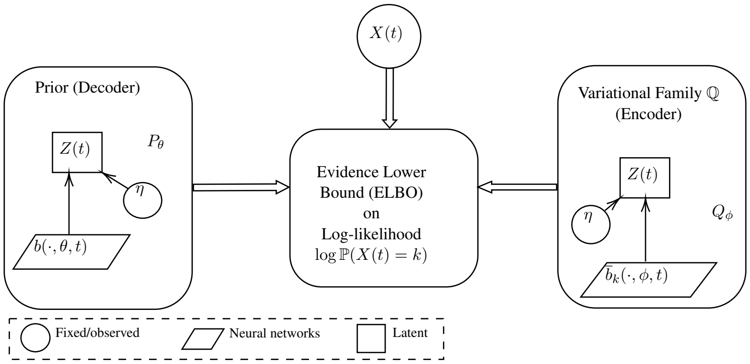

$X(t)$
Prior (Decoder)
$Z(t)$
P_θ
η
$b(\cdot,\theta,t)$
Evidence Lower Bound (ELBO) on Log-likelihood $\log\mathbb{P}(X(t)=k)$
Variational Family $\mathbb{Q}$ (Encoder)
$Z(t)$
η
Q_ϕ
$\bar{b}_k(\cdot,\phi,t)$
Fixed/observed
Neural networks
Latent

Figure 1: An illustration of the deep latent modeling framework.

Theorem 1 Define $u_{k_1,k_2}(Z(t),t;\theta,\phi) := (\eta\sqrt{Z(t)})^{-1}\left(\bar{b}_{k_1,k_2}(Z(t),t;\phi) - b(Z(t),t;\theta)\right)$ and suppose that u_{k_1,k_2} satisfies a *strong* Novikov's condition, $\mathbb{E}\left[\exp\left(\frac{1}{2}\int_0^{t_2}|u_{k_1,k_2}(Z(t),t;\theta,\phi)|^2dt\right)\right] < +\infty\ \forall\theta,\phi$. Then,

$$\hat{W}_t := \int_0^t u_{k_1,k_2}(Z(s),s;\theta,\phi)ds + W(t) \tag{10}$$

is a Brownian motion w.r.t. Q_ϕ , $dZ(t) = b(Z(t),t;\theta)dt + \eta\sqrt{Z}(t)d\hat{W}_t$, and

$$\log\mathbb{P}(Y_1 = (k_1,k_2)) \geq \mathbb{E}_{Q_\phi}\left[\log\mathbb{P}(Y_1 = (k_1,k_2)|Z_{0:T}) - \tfrac{1}{2}\int_0^T u^2_{k_1,k_2}(Z(s),s;\theta,\phi)ds\right] := \text{ELBO}. \tag{11}$$

Here, the likelihood on the right hand side is given by (3). Notice that we must assume that Novikov's condition holds for all possible parameterizations of the functions $\bar{b}$ and b. This is a strong condition that is satisfied for the class of DNNs that we work with in this paper, since the output of the DNN is bounded by definition. However, more analysis is required on sufficient conditions for DNNs to satisfy Novikov's condition. Another observation is that the latent process in time $(0,\frac{T}{2}]$ depends on both observations k_1 and k_2; note the integrand in (11). Therefore, this is a smoothing estimation problem.

5.2 TRAINING THE DLM

Our objective is to train the neural networks $b(Z(t),t;\theta)$, $u_{k_1,k_2}(Z(t),t;\theta,\phi)$ and $\bar{G}(t;\theta_1)$ by maximizing the ELBO. We fix $u_{k_1,k_2}(Z(t),t;\theta,\phi)$ to be a deterministic neural network defined as $\tilde{u}(k_1,k_2,t;\beta)$ with parameters β. Combined with (10), this additional restriction imposed on $u_{k_1,k_2}(Z(t),t;\theta,\phi)$ ensures that the process $\hat{W}_t$ has independent increments. In the variational inference literature (Blei et al. 2017), this assumption is also known as the mean-field approximation; that is, each partition of the unknown latent variable is independent of the other. A similar assumption on the latent process was used in Tzen and Raginsky (2019), where the authors call it a path-space analog of the mean-field approximation. Now substituting $u_{k_1,k_2}(Z(t),t;\theta,\phi) = \tilde{u}(k_1,k_2,t;\beta)$ in (10) and from the observation $dZ(t) = b(Z(t),t;\theta)dt + \eta\sqrt{Z}(t)d\hat{W}_t$, it follows that we can simulate the SDE using $W(t)$ instead of $\hat{W}(t)$; that is,

$$dZ(t) = b(Z(t),t;\theta)dt + \sqrt{Z(t)}\tilde{u}(k_1,k_2,t;\beta)dt + \sqrt{Z(t)}dW(t) \text{ and } Z(0) = 0. \tag{12}$$

We denote the measure induced by the above SDE as $Q_{\beta,\theta}$. For simplicity we fix $\eta = 1$.

We use stochastic gradient descent (SGD) to maximize the objective in (11) to learn the unknown neural network parameters θ and β. In order to use SGD, we first need to generate sample paths of the latent process $Z(t)$ (12), which we do using the Euler-Maruyama discretization method. We partition the time interval $[0,T]$ in N equal sub-intervals, denoted as $\{t_0,t_1,...t_N\}$, with $t_0 = 0$ and $t_N = T$, set

$Z(t_0) = Z(0)$, and simulate $\{Z(t_m)\}_{0\le m\le N}$ using the recursive equation

$$Z(t_{m+1}) = Z(t_m) + b(Z(t_m), t_m, \theta)(t_{m+1} - t_m) + \sqrt{Z(t_m)}\tilde{u}(k_1, k_2, t_m; \beta)(t_{m+1} - t_m) + \sqrt{Z(t_m)}\Delta W_m \tag{13}$$

where $\{\Delta W_m := W(t_{m+1}) - W(t_m)\}_{0\le m<N}$ are N i.i.d.standard Gaussians.

In order to use SGD we also need to compute the gradient of the objective function (11) with respect to the parameters θ, θ_1 and β. Notice that the expectation in ELBO is with respect to the measure induced by SDE in (12) denoted as $Q_{\beta,\theta}$. Observe that the only source of randomness in generating $Z(t)$ is from the Brownian motion $W(t)$, which does not depend on either β or θ. Therefore we interchange the differential operator with respect to the parameters and the expectation in (11). To make the dependence of $Z(t)$ on β and θ explicit, we write $Z(t)$ as $Z^{\beta,\theta}(t)$. In particular for given values of parameters θ and β_{-j} (all components of parameter β except β^j) observe that $\frac{\partial}{\partial\beta^j}\mathbb{E}\left[\log\mathbb{P}(Y_1 = (k_1, k_2)|Z^{\beta,\theta}_{0:T}) - \frac{1}{2}\int_0^T \tilde{u}^2(k_1, k_2, s; \beta)ds\right] =$

$$\begin{aligned}&\mathbb{E}\Bigg[\frac{\partial}{\partial\beta^j}\log\Bigg(\frac{e^{-\int_0^{\frac{T}{2}} Z^{\beta,\theta}(t)\bar{G}(\frac{T}{2}-t;\theta_1)dt}\left(\int_0^{\frac{T}{2}} Z^{\beta,\theta}(t)\bar{G}(\frac{T}{2}-t;\theta_1)dt\right)^{k_1}}{k_1!}\Bigg)\times\\ &\Bigg(\sum_{i=0}^{\min\{k_1,k_2\}}\mathrm{PB}(i;p^{\theta_1}_{1:k_1})\frac{e^{-\int_{\frac{T}{2}}^{T} Z^{\beta,\theta}(t)\bar{G}(T-t;\theta_1)dt}\left(\int_{\frac{T}{2}}^{T} Z^{\beta,\theta}(t)\bar{G}(T-t;\theta_1)dt\right)^{k_2-i}}{(k_2-i)!}\Bigg) - \frac{\partial}{\partial\beta^j}\int_0^T \tilde{u}^2(k_1, k_2, s; \beta)ds\Bigg],\end{aligned} \tag{14}$$

where we use the likelihood expression from (3). Also note that, to avoid any confusion, we have omitted subscript $Q_{\beta,\theta}$ from $\mathbb{E}[\cdot]$ above. Now applying straightforward product differentiation rule and subsequently interchanging the integral and $\frac{\partial}{\partial\beta^j}$, would result into an expression requiring us to compute the derivative of the process $Z^{\beta,\theta}(t)$ with respect to β^j. To compute the derivative process, it follows from Kunita (1984), Theorem 3.1) that under certain regularity condition on the drift and diffusion coefficient of the process $Z^{\beta,\theta}(t)$ (12) the derivative process $\frac{\partial}{\partial\beta^j}Z^{\beta,\theta}(t)$ is the solution of the following SDE

$$\begin{aligned}\frac{\partial Z^{\beta,\theta}(t)}{\partial\beta^j} &= \int_0^t\left(\frac{\partial b(Z^{\beta,\theta}(s), s; \theta)}{\partial Z^{\beta,\theta}(s)}\frac{\partial Z^{\beta,\theta}(s)}{\partial\beta^j} + \frac{\tilde{u}(k_1, k_2, s; \beta)}{2\sqrt{Z^{\beta,\theta}(s)}}\frac{\partial Z^{\beta,\theta}(s)}{\partial\theta^j} + \sqrt{Z^{\beta,\theta}(s)}\frac{\partial u_s}{\partial\beta^j}\right)ds\\ &+ \int_0^t\left(\frac{1}{2\sqrt{Z^{\beta,\theta}(s)}}\frac{\partial Z^{\beta,\theta}(s)}{\partial\beta^j}\right)dW_s \text{ and } \frac{\partial Z^{\beta,\theta}(0)}{\partial\beta^j} = 0.\end{aligned} \tag{15}$$

The derivative of the ELBO with respect to the θ_1 can be obtained directly from the automatic differentiation.

6 NUMERICAL RESULTS

We conducted a number of experiments to demonstrate performance of the DLM. We start by describing the setting for the experiments. The code is written in Python using PyTorch. The time complexity for each iteration of the gradient update is $\mathcal{O}\left(N((k+n)T(b) + T(\tilde{u}))\right)$, where k and n are number of parameters in β and θ, respectively, $T(f)$ is the time complexity for computing f, and N is the number of time steps in the time discretization.

Observe that training the neural network by maximizing the ELBO entails solving a stochastic optimization problem in (11). We use a sample average approximation (SAA) of (11) for which we simulate m independent sample paths of $(Z(t) : t \in [0, T])$. We integrate the SDE using Euler-Maruyama discretization, as noted in the previous section.

Next, there are multiple ways of parameterizing the service time distribution. We model the integrated hazard function (IHF) $H(t; \theta_1)$, where θ_1 denote the parameters of a neural network. It can be easily seen that IHF of the exponential distribution is precisely linear. Once we learn the integrated hazard function, CDF of the service time distribution is simply $G(t;\theta_1) = 1 - e^{-H(t;\theta_1)}$. To simplify the notation, we define $F(t;\theta_1) \equiv e^{-H(t_i - t;\theta_1)}$ for $t \in (t_i, t_{i+1}]$, $i = 1, 2, ..., k-1$, where k is the total number of observation epochs.

The architecture of the neural networks is

- $b(Z(t), t; \theta) : R^2 \rightarrow R$ is a feedforward neural network with 10 fully connected layers of size 10. The activation function is chosen as $tanh$. The inputs are time epoch and the current intensity.
- $\tilde{u}(k, t; \beta) : R^2 \rightarrow R$ is also a feedforward neural network with 10 fully connected layers of size 10. The activation function is chosen as $tanh$. The inputs are time epoch and the state at time T.
- $H(t, \theta_1) : R \rightarrow R$ is a feedforward neural network with 3 fully connected layers of size 5. The activation function is chosen as $tanh$. The only input is the time. In order to ensure that the parameters θ_1 descent to the correct region, we enforce linear constraints $at \leq H(t, \theta_1) \leq bt$. In our simulations, we set $a = 0$ and $b = 4$.

"Better" architectures can be achieved through hyperparameter optimization, which we do not pursue here.

We assume that the true latent intensity process is a standard CIR process:

$$dZ(t) = 0.3(80 - Z(t))dt + \sqrt{Z(t)}dW(t), \tag{16}$$

where $Z(0) = 5$. We set the simulation horizon to be $T = 2$, and uniformly partition the interval $[0, T]$ into the grid $\mathcal{P} = \{t_1, t_2, .., t_M\}$ with $t_{k+1} - t_k = 1/50$, $t_1 = 0$ and $t_M = 2$. The training data consists of $n = 200$ sample paths of the Cox process generated using the theoretical model (16). This data is further divided into 'mini-batches' of size 10 and then fed into the Adam solver (Kingma and Ba 2014). We run the code for 35 epochs (350 gradient updates in total). The learning rate for $b(Z(t), t; \theta)$ and $\tilde{u}(k, t; \beta) : R^2 \rightarrow R$ are both set to be 0.01.

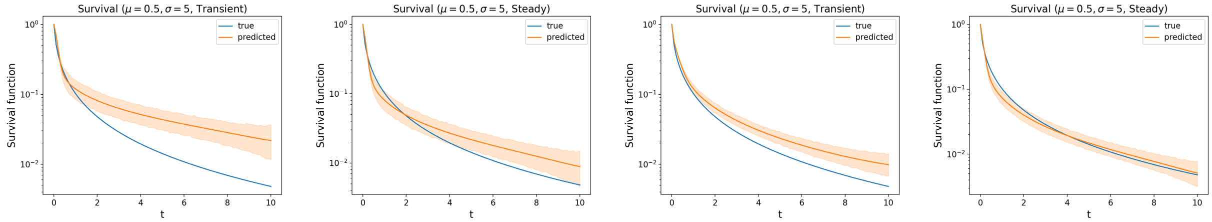
6.1 Learning the Tail of the Service Time Distribution

We consider the problem of jointly estimating the service time distribution and the intensity process, when the intensity is in steady state. Following Equation (4), the likelihood becomes $\mathbb{P}\left(Y = (k_1, k_2) | Z_{t_0:t_2}, s^0_{1:k_0}, s^1_{1:k_1}\right)$

$$\begin{aligned} = &\left[\sum_{i=0}^{\min\{k_0,k_1\}} \text{PB}(i; p_{1:k_0}) \text{Pois}\left(k_1 - i; \int_{t_0}^{t_1} Z(t)\bar{G}(t_1 - t; \theta_1)dt\right)\right] \times \\ &\left[\sum_{i=0}^{\min\{k_1,k_2\}} \text{PB}(i; p_{1:k_1}) \text{Pois}\left(k_2 - i; \int_{t_1}^{t_2} Z(t)\bar{G}(t_2 - t; \theta_1)dt\right)\right], \end{aligned} \tag{17}$$

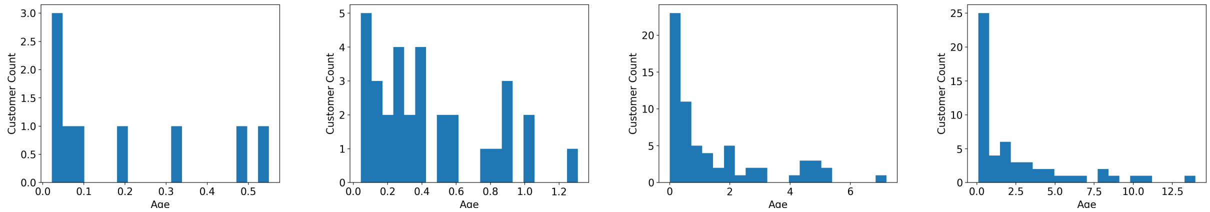
where $\{t_0, t_1, t_2\} = \{20, 21, 22\}$, $s^0_{1:k_0}$ is the age of customers that are in the system at time $t_0 = 20$ and $s^1_{1:k_1}$ is the age of customers that are in the system at time $t_1 = 21$. Suppose the service times are log-normally distributed with mean 0.5 and variance 5 (the variance is chosen to be large deliberately to emphasize the tail behavior). Figure 2a and 2b shows that the tail prediction is significantly better in the steady-state case. In addition, the total variation distance for the transient and steady case are 0.167 and 0.103, respectively, implying that the overall prediction for the steady state system is better.

This phenomenon is due to lack of the age information in the transient setting. Recall that the customer arriving at t_1 has a survival probability of $p_i = \mathbb{P}(S \geq t_2 - t_1 + s_i | S \geq s_i) = \frac{\bar{G}(t_2 - t_1 + s_i; \theta_i)}{\bar{G}(s_i; \theta)}$ up until time t_2. If learning from an empty system, then no information of the customers surviving beyond t_2 enters the likelihood. Figure 3 is a histogram that demonstrates the age information at different time epochs. We also run the experiments for different variance. It can be seen that the L1-distance between the true and learned service time CDF is always smaller in the steady-state case.



(a) Unknown Intensity (b) Unknown Intensity (c) Known Intensity (d) Known Intensity

Figure 2: Survival function prediction with 95% CI.



(a) At t=1 (b) At t=2 (c) At t=10 (d) At t=20

Figure 3: Histogram of the customer age.

6.2 Estimating the Service Time Distribution with Known Intensity

In this section, we assume access to the true intensity and estimate the service time distribution alone. Specifically, we set the latent drift $b_{k_1,k_2}(Z(t), t : \phi)dt = 0.3(80 - Z(t))$ and then optimize the ELBO objective over θ and θ_1. As in the previous section, we let the true service time distribution to be log-normal with mean 0.5 and variance 5. Comparing Figure 2a-2b to Figure 2c-2d, it is apparent that assuming knowledge of the true intensity slightly improves the overall prediction. Table 1 summaries estimation of the service time distribution in terms of the L1-distance. We see from the table that learning of the service time CDF is more accurate if we learn the service time distribution only and if the system is already in steady state. This pattern is consistent under different variance of the service time distribution.

Table 1: Simulation result for the L1 distance. (log-normal)

$\mu = 0.5$	σ^2	0.1	1	10	100
Transient	CDF	0.111	0.127	0.131	0.120
	CDF & Intensity	0.134	0.174	0.167	0.151
Steady	CDF	0.083	0.116	0.095	0.103
	CDF & Intensity	0.110	0.152	0.136	0.128

6.3 Estimating Different Service Time Distributions

Next, we explore the prediction under different service time distributions. We are interested in log-normal and Pareto service time distributions. The mean of the service time is set to be 0.5 and we increase the variance from 0.1 to 100. We consider the transient case where the observations are made at time $t_1 = 1$ and $t_2 = 2$, and make 500 predictions (independent from the training sample) of the queue length at time $t_2 = 2$. The simulation is summarized in Table 2. Overall, prediction of the queue length is acceptable for all the service time distributions. We note that mean of the predictions is slightly off when variance of the true distribution is large. It can be seen from the central limit theorem that, to achieve the same variance of the mean prediction, a larger sample size is required for distribution with higher variance. On the other hand, variance of the predictions is slightly overestimated. Our observation is that there is a slight identification issue between estimation of the service time distribution and the SDE, i.e., the drift of $Z(t)$ can be underestimated slightly so that the random component of the integral $\int_0^t Z(s)\bar{G}(t-s)ds$ is overestimated. We believe this can be improved by better choice of the hyper-parameters. But we don't further pursuit this. For most of the service time distribution, the true variance is covered in the 95% confidence interval of the predicted one.

Table 2: Simulation for different service time distribution.

$\mu = 0.5$	σ^2	**0.1**		**1**		**10**		**100**	
		True	Predicted	True	Predicted	True	Predicted	True	Predicted
Log-normal	Mean with CI	14.63 [14.25 15.01]	14.55 [13.88 15.12]	10.84 [10.51 11.16]	10.47 [9.23 11.01]	6.92 [6.74 7.10]	6.97 [6.58 7.36]	4.46 [4.26 4.65]	3.96 [3.66 4.25]
	Variance with CI	18.28 [16.21 20.78]	16.61 [13.78 20.42]	13.63 [12.08 15.49]	15.20 [12.69 18.80]	8.22 [7.54 8.99]	7.79 [6.46 7.57]	4.98 [4.41 5.66]	4.48 [3.72 5.51]
Pareto	Mean with CI	14.90 [14.51 15.28]	15.31 [14.65 15.97]	14.08 [13.70 14.45]	13.97 [13.34 14.59]	13.83 [13.46 14.21]	13.43 [12.81 14.04]	13.63 [13.29 13.98]	13.24 [12.63 13.84]
	Variance with CI	19.44 [17.24 22.10]	22.42 [18.60 27.56]	18.44 [16.35 20.96]	19.73 [16.37 24.26]	18.21 [16.15 20.70]	19.15 [15.89 23.54]	15.57 [13.80 17.69]	18.75 [15.55 23.04]

6.4 Approximations to the Poisson Binomial Distribution

In this section, we demonstrate how different approximation techniques influence the prediction result. We follow the same setting as in the previous section while focus only on the log-normal distribution. Observe from Table 3 that normal approximation is slightly off when the queue length at observation epoch is small. See prediction of the mean for $\sigma^2 = 100$ at T for instance. Our interpretation is that Normal approximation is based on the central limit theorem, see Neammanee (2005) for instance. The approximation can be poor when number of the surviving customers at the previous observation epoch is small. Otherwise, Normal approximation makes reasonable prediction on both the mean and variance. On the other hand, Poisson approximation works well in almost all the predictions. However, one shall be extra cautious when the overall survival probability is large. It is pointed out in Le Cam et al. (1960) that the Poisson approximation is guaranteed to work well only if the maximum survival probability is smaller than $\frac{1}{4}$.

Table 3: Simulation for different approximation schemes. (log-normal distribution)

$\mu = 0.5$	σ^2	**0.1**		**1**		**10**		**100**	
		T/2	T	T/2	T	T/2	T	T/2	T
TRUE	Mean with CI	6.92 [6.66 7.17]	14.63 [14.25 15.01]	5.17 [4.95 5.38]	10.84 [10.51 11.16]	3.21 [3.09 3.32]	6.92 [6.74 7.10]	2.06 [1.93 2.19]	4.46 [4.26 4.65]
	Variance with CI	8.38 [7.43 9.52]	18.28 [16.21 20.78]	5.88 [5.21 6.68]	13.63 [12.08 15,49]	3.26 [2.99 3.57]	8.22 [7.54 8.99]	2.12 [1.88 2.41]	4.98 [4.41 5.66]
PB exact	Mean with CI	6.96 [6.56 7.36]	14.55 [13.98 15.12]	5.28 [4.95 5.61]	10.47 [9.23 11.01]	3.85 [3.59 4.10]	6.97 [6.58 7.36]	1.99 [1.79 2.19]	3.96 [3.66 4.25]
	Variance with CI	8.31 [6.89 10.21]	16.61 [13.78 20.42]	5.48 [4.54 6.73]	15.20 [12.69 18.80]	3.42 [2.84 4.21]	7.79 [6.46 7.57]	2.00 [1.66 2.46]	4.48 [3.72 5.51]
Gaussian Approx	Mean with CI	6.87 [6.61 8.32]	14.17 [13.80 14.54]	4.77 [4.57 4.97]	10.40 [10.11 10.70]	2.82 [2.67 2.97]	6.32 [6.06 6.57]	1.93 [1.80 2.05]	5.46 [5.27 5.65]
	Variance with CI	8.32 [7.38 9.46]	17.41 [15.44 19.79]	5.09 [4.51 5.79]	11.24 [9.97 12.78]	2.87 [2.54 3.26]	8.35 [7.40 7.49]	1.96 [1.73 2.22	4.72 [4.19 5.37]
Poisson Approx	Mean with CI	6.89 [6.63 7.14]	14.37 [14.00 14.73]	5.27 [5.06 5.47]	10.52 [10.18 10.85]	3.76 [3.60 3.91]	6.88 [6.63 7.12]	2.12 [1.99 2.24]	4.77 [4.58 4.95]
	Variance with CI	8.42 [7.46 9.57]	17.20 [15.25 19.55]	5.60 [4.96 6.36]	14.51 [12.86 16.49]	3.19 [2.82 3.62]	7.57 [6.71 8.60]	2.05 [1.81 2.33]	4.60 [4.07 5.22]

7 CONCLUSIONS AND COMMENTARY

We present a machine learning based method for estimation of the arrival intensity and service time distribution for the Cox$/G/\infty$ queues. We demonstrate effectiveness of our method by comparing the prediction accuracy under: (1) steady/transient queue; (2) different learning objectives; (3) different true service time CDF; (4) different approximations of Poisson Binomial distribution. While the predictions are accurate in almost all the cases, we provide insight on favorable conditions under which our method works exceptionally well. In future work, we plan to relax the requirement that the age information be available. We are also interested in establishing statistical guarantees of the estimation. In ongoing work, we aim to show that, if the architecture of the neural networks are chosen properly, then the DLM is consistent. We will present the consistency result in the future.

REFERENCES

Asanjarani, A., Y. Nazarathy, and P. Taylor. 2021. "A survey of parameter and state estimation in queues". *Queueing Systems* 97(1):39–80.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe. 2017. "Variational inference: A review for statisticians". *Journal of the American statistical Association* 112(518):859–877.

Eick, S. G., W. A. Massey, and W. Whitt. 1993. "The physics of the Mt/G/∞ queue". *Operations Research* 41(4):731–742.

Goldenschluger, A. et al. 2016. "Nonparametric estimation of the service time distribution in the M/G/∞ queue". *Advances in Applied Probability* 48(4):1117–1138.

Goldenshluger, A., and D. T. Koops. 2019. "Nonparametric estimation of service time characteristics in infinite-server queues with nonstationary Poisson input". *Stochastic Systems* 9(3):183–207.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. MIT press.

Hillestad, R. J., and M. J. Carrillo. 1980. "Models and Techniques for Recoverable Item Stockage when Demand and the Repair Process are Nonstationary. Part I. Performance Measurement.". Technical report, RAND CORP SANTA MONICA CA.

Honnappa, H., Y. Liu, S. Tindel, and A. Yip. 2020. "Infinite server queues in a random fast oscillatory environment". *arXiv preprint arXiv:2004.05034*.

Kingma, D. P., and J. Ba. 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., and M. Welling. 2019. "An Introduction to Variational Autoencoders". *Foundations and Trends® in Machine Learning* 12(4):307–392.

Kunita, H. 1984. "Stochastic differential equations and stochastic flows of diffeomorphisms". In *Ecole d'été de probabilités de Saint-Flour XII-1982*, 143–303. Springer.

Le Cam, L. et al. 1960. "An approximation theorem for the Poisson binomial distribution.". *Pacific Journal of Mathematics* 10(4):1181–1197.

Li, D., Q. Hu, L. Wang, and D. Yu. 2019. "Statistical inference for Mt/G/Infinity queueing systems under incomplete observations". *European Journal of Operational Research* 279(3):882–901.

Neammanee, K. 2005. "A refinement of Normal approximation to Poisson Binomial". *International Journal of Mathematics and Mathematical Sciences* 2005(5):717–728.

Pang, G., and W. Whitt. 2010. "Two-parameter heavy-traffic limits for infinite-server queues". *Queueing Systems* 65(4):325–364.

Schweer, S., and C. Wichelhaus. 2015. "Nonparametric estimation of the service time distribution in the discrete-time GI/G/∞ queue with partial information". *Stochastic Processes and their Applications* 125(1):233–253.

Schweer, S., and C. Wichelhaus. 2020. "Nonparametric estimation of the service time distribution in discrete-time queueing networks". *Stochastic Processes and their Applications* 130(8):4643–4666.

Tzen, B., and M. Raginsky. 2019. "Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit". *arXiv preprint arXiv:1905.09883*.

Wang, R., P. Jaiwal, and H. Honnappa. 2020. "Estimating Stochastic Poisson Intensities Using Deep Latent Models". *arXiv preprint arXiv:2007.06037*.

Zhang, X., L. J. Hong, and J. Zhang. 2014. "Scaling and modeling of call center arrivals". In *Proceedings of the Winter Simulation Conference 2014*, 476–485. IEEE.

AUTHOR BIOGRAPHIES

Ruixin Wang is a Ph.D. student in the School of Industrial Engineering at Purdue University. His research interests lie in approximate dynamic programming, electricity market modeling and stochastic simulation. His email address is wang2252@purdue.edu.

HARSHA HONNAPPA is an assistant professor in the School of Industrial Engineering at Purdue University. His research interests are in applied probability, game theory, and machine learning. He is a member of INFORMS, IEEE, and SIAM, and serves as an associate editor for Operations Research and Operations Research Letters. His email address is honnappa@purdue.edu.