A Machine-Learning-Based Importance Sampling Method to Compute Rare Event Probabilities[*]

Vishwas Rao[1], Romit Maulik[1], Emil Constantinescu[1], and Mihai Anitescu[1,2]

[1] Argonne National Laboratory, Lemont, IL, USA
[2] University of Chicago, Chicago, IL, USA
{vhebbur,rmaulik,emconsta,anitescu}@anl.gov

Abstract. We develop a novel computational method for evaluating the extreme excursion probabilities arising from random initialization of nonlinear dynamical systems. The method uses excursion probability theory to formulate a sequence of Bayesian inverse problems that, when solved, yields the biasing distribution. Solving multiple Bayesian inverse problems can be expensive; more so in higher dimensions. To alleviate the computational cost, we build machine-learning-based surrogates to solve the Bayesian inverse problems that give rise to the biasing distribution. This biasing distribution can then be used in an importance sampling procedure to estimate the extreme excursion probabilities.

Keywords: Machine learning · Rice's formula · Gaussian processes.

1 Motivation

Characterizing high-impact rare and extreme events such as hurricanes, tornadoes, and cascading power failures are of great social and economic importance. Many of these natural phenomena and engineering systems can be modeled by using dynamical systems. The models representing these complex phenomena are approximate and have many sources of uncertainties. For example, the exact initial and boundary conditons or the external forcings that are necessary to fully define the underlying model might be unknown. Other parameters that are set based on experimental data may also be uncertain or only partially known. A probabilistic framework is generally used to formulate the problem of quantifying various uncertainties in these complex systems. By definition, the outcomes of interest that correspond to high-impact rare and extreme events reside in the tails of the probability distribution of the associated event space. Fully characterizing the tails requires resolving high-dimensional integrals over irregular domains. The most commonly used method to determine the probability

of rare and extreme events is Monte Carlo simulation (MCS). Computing rare-event probabilities via MCS involves generating several samples of the random variable and calculating the fraction of the samples that produce the outcome of interest. For small probabilities, however, this process is expensive. For example, consider an event whose probability is around 10^{-3} and the underlying numerical model for the calculation requires ten minutes per simulation. With MCS, estimating the probability of such an event to an accuracy of 10% will require two years of serial computation. Hence, alternative methods are needed that are computationally efficient.

Important examples of extreme events are rogue waves in the ocean [14], hurricanes, tornadoes [29], and power outages [3]. The motivation for this work comes from the rising concern surrounding transient security in the presence of uncertain initial conditions identified by North American Electric Reliability Corporation in connection with its long-term reliability assessment [32]. The problem can be mathematically formulated as a dynamical system with uncertain initial conditions. In this paper, the aim is to compute the extreme excursion probability: the probability that the transient due to a sudden malfunction exceeds preset safety limits. Typically, the target safe limit exceedance probabilities are in the range $10^{-4} - -10^{-5}$. We note that the same formulation is applicable in other applications such as data assimilation, which is used extensively for medium- to long-term weather forecasting. For example, one can potentially use the formulation in this paper to determine the likelihood of temperature levels at a location exceeding certain thresholds or the likelihood of precipitation levels exceeding safe levels in a certain area.

In [25], we presented an algorithm that uses ideas from excursion probability theory to evaluate the probability of extreme events [1]. In particular we used Rice's formula [27], which was developed to estimate the average number of upcrossings for a generic stochastic process. Rice's formula is given by

$$\mathbb{E}\left\{N_u^+(0,T)\right\} = \int_0^T \int_0^\infty y\varphi_t(u,y)\,\mathrm{d}y\,\mathrm{d}t\,, \tag{1}$$

where the left-hand side denotes the expected number of upcrossings of level u, y is the derivative of the stochastic process in a mean squared sense, and $\varphi_t(u,y)$ represents the joint probability distribution of the process and its derivative. In this paper, we build on our recent algorithm [25], which we used to construct an importance biasing distribution (IBD) to accelerate the computation of extreme event probabilities. A key step in the algorithm presented in [25] involves solving multiple Bayesian inverse problems, which can be expensive in high dimensions. Here, we propose to use machine-learning-based surrogates to obtain the inverse maps and hence alleviate the computational costs.

1.1   Mathematical setup and overview of the method

The mathematical setup used in this paper consists of a nonlinear dynamical system that is excited by a Gaussian initial state and that results in a non-Gaussian stochastic process. We are interested in estimating the probability

of the stochastic process exceeding a preset threshold. Moreover, we wish to estimate the probabilities when the underlying event of the process exceeding the threshold is a *rare event*. The rare events typically lie in the tails of the underlying event distribution. To characterize the tail of the resulting stochastic process, we use ideas from theory excursion probabilities [1]. Specifically, we use Rice's formula (1) to estimate the expected number of upcrossings of a stochastic process. For a description of the mathematically rigorous settings used for the rare event problem, we refer interested readers to [25, §2] and references therein.

Evaluating $\varphi_t(u, y)$, the joint probability distribution of the stochastic process and its derivative, is central to evaluating the integral in Rice's formula. However, $\varphi_t(u, y)$ is analytically computable only for Gaussian processes. Since our setup results in a non-Gaussian stochastic process, we linearize the nonlinear dynamical system variation around the trajectories starting at the mean of the initial state. We thus obtain a Gaussian approximation to the system trajectory distribution. In [25], we solve a sequence of Bayesian inverse problems to determine a biasing distribution to accelerate the convergence of the probability estimates. Forr high-dimensional problems, however, solving multiple Bayesian inverse problems can be expensive. In this work, we propose to replace multiple solutions to Bayesian inverse problems with machine-learning-based surrogates to alleviate the computational burden.

1.2   Organization

The rest of the paper is organized as follows. In §2 we review the existing literature for estimating rare event probabilities. In §3 we reformulate the problem of determining the IBD as a Bayesian inference problem, and in §4 we develop a machine-learning-based surrogate to approximate the solution to the Bayesian inference problem. In §5 we demonstrate this methodology on a simple nonlinear dynamical system excited by a Gaussian distribution. In §6 we present our conclusions and potential future research directions.

2   Existing literature

2.1   Monte Carlo and importance sampling

Most of the existing methods to compute the probabilities of rare events use MCS directly or indirectly. The MCS approach was developed by Metropolis and his collaborators to solve problems in mathematical physics [22]. Since then, it has been used in a variety of applications [21, 28]. When evaluating rare event probabilities, the MCS method basically counts the fraction of the random samples that cause the rare event. For a small probabilty P of the underlying event, the number of samples required to obtain an accuracy of $\epsilon \ll 1$ is $\mathcal{O}(\epsilon^{-2}P^{-1})$. Hence MCS becomes impractical for estimating rare event probabilities.

A popular sampling technique that is employed to compute rare event probabilties is importance sampling (IS). IS is a variance reduction technique developed in the 1950s [17] to estimate the quantity of interest by constructing

estimators that have smaller variance than MCS. In MCS, simulations from most of the samples do not result in the rare event and hence do not play a part in probability calculations. IS, instead, uses problem-specific information to construct an IBD; computing the rare event probability using the IBD requires fewer samples. Based on this idea, several techniques for constructing IBDs have been developed [8]. For a more detailed treatment of IS, we direct interested readers to [2, 13]. One of the major challenges involved with importance sampling is the construction of an IBD that results in a low-variance estimator. We note that the approach may sometimes be inefficient for high-dimensional problems [19]. A more detailed description of MCS and IS in the context of rare events can be found in [25, §2] and references therein.

2.2   Nested subset methods

Other methods use the notion of conditional probability over a sequence of nested subsets of the probability space of interest. For example, one can start with the entire probability space and progressively shrink to the region that corresponds to the rare event. Furthermore, one can use the notion of conditional probability to factorize the event of interest as a product of conditional events. Subset simulation (SS) [4] and splitting methods [16] are ideas that use this idea. Several modifications and improvements have been proposed to both SS [6, 9, 10, 18, 33] and splitting methods [5, 7]. Evaluating the conditional probabilities forms a major portion of the computational load. Compute the conditional probabilities for different nested subsets concurrently is nontrivial.

2.3   Other approaches

Large deviation theory (LDT) is an efficient approach for estimating rare events in cases when the event space of interest is dominated by few elements such as rogue waves of a certain height. LDT also has been used to estimate the probabilities of extreme events in dynamical systems with random components [11, 12]. A sequential sampling strategy has been used to compute extreme event statistics [23, 24].

3   A Bayesian inference formulation to construct IBD

Most of the work in this section is a review of our approach described in [25, §3]. Here we reformulate the problem of constructing an IBD as a sequence of Bayesian inverse problems. Consider the following dynamical system,

$$\mathbf{x}' = f(t, \mathbf{x}), \quad t = [0, T] \tag{2}$$
$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}_0 \sim p, \quad \mathbf{x} \in \Omega,$$

where $\mathbf{c}$ is a canonical basis vector and $\mathbf{x}_0$, the initial state of the system, is uncertain and has a probability distribution p. The problem of interest is to

estimate the probability that $\mathbf{c}^\top \mathbf{x}(t)$ exceeds the level u for $t \in [0, T]$. That is, we seek to estimate the following *excursion probability*,

$$P_T(u) := \mathbb{P}\left(\sup_{0 \leq t \leq T} \mathbf{c}^\top \mathbf{x}(t, \mathbf{x}_0) \geq u, \;\; t \in [0, T]\right), \tag{3}$$

where $\mathbf{x}(t, \mathbf{x}_0)$ represents the solution of the dynamical system (2) for a given initial condition $\mathbf{x}_0$. We note that

$$P_T(u) = \mu(\Omega(u)), \tag{4}$$

where μ is the respective measure transformation subject to (2) and $\Omega(u) \subset \Omega$ represents the *excursion set*

$$\Omega(u) := \left\{\mathbf{x}_0 : \sup_{0 \leq t \leq T} \mathbf{c}^\top \mathbf{x}(t, \mathbf{x}_0) \geq u\right\}. \tag{5}$$

Hence, estimating $\Omega(u)$ will help us in estimating the excursion probability $P_T(u)$. In general, however, estimating the excursion set $\Omega(u)$ analytically is difficult. Rice's formula, (1) gives us insights about the excursion set and can be used to construct an approximation to the excursion set.

Recall that in Rice's formula (1), $\varphi_t(u, y)$ represents the joint probability density of $\mathbf{c}^\top \mathbf{x}$ and its derivative $\mathbf{c}^\top \mathbf{x}'$ for an excursion level u. The right-hand side of (1) can be interpreted as the summation of all times and slopes at which an excursion occurs. One can sample from $y\varphi_t(u, y)$ to obtain a slope-time pair (y_i, t_i) at which the sample paths of the stochastic process cause an excursion. Now consider the map $\mathcal{G} : \mathbb{R}^{d \times 1} \to \mathbb{R}^2$ that evaluates the vector $\begin{bmatrix} \mathbf{c}^\top \mathbf{x}(t) \\ \mathbf{c}^\top \mathbf{x}'(t) \end{bmatrix}$ based on the dynamical system (2), given an initial state $\mathbf{x}_0$ and a time t. By definition of the excursion set $\Omega(u)$, there exists an element $\mathbf{x}_i \in \Omega(u)$ that satisfies the following relationship,

$$\mathcal{G}(\mathbf{x}_i, t_i) = \begin{bmatrix} u + \varepsilon_i \\ y_i \end{bmatrix}, \tag{6}$$

where $\varepsilon > 0$. We can use this insight to construct an approximation of $\Omega(u)$ by constructing the preimages of multiple slope-time pairs. Observe that the problem of finding the preimage of a sample (y_i, t_i) is ill-posed since there could be multiple $\mathbf{x}_i$'s that map to $\begin{bmatrix} u + \varepsilon_i \\ y_i \end{bmatrix}$ at t_i via operator $\mathcal{G}$. We define the set

$$X_i := \left\{\mathbf{x}_i \in \Omega : \mathcal{G}(\mathbf{x}_i, t_i) = \begin{bmatrix} u + \varepsilon_i \\ y_i \end{bmatrix}\right\}, \tag{7}$$

and an approximation $\widehat{\Omega}(u)$ to $\Omega(u)$ can be written as

$$\widehat{\Omega}(u) := \bigcup_{i=1}^{N} X_i. \tag{8}$$

Note that the approximation (8) improves as we increase N. For a discussion on the choice of ε_i, we refer interested readers to [25, §3.3].

The underlying computational framework to approximate $\widehat{\Omega}(u)$ consists of the following stages:

- Draw samples from unnormalized $y\varphi_t(u, y)$
- Find the preimages of these samples to approximate $\Omega(u)$.

We use MCMC to draw samples from unnormalized $y\varphi_t(u, y)$. We note that irrespective of the size of the dynamical system, $y\varphi_t(u, y)$ represents an unnormalized density in two dimensions; hence, using MCMC is an effective means , draw samples from it. Drawing samples from $y\varphi_t(u, y)$ requires evaluating it repeatedly, and in the following section we discuss the means to do so.

3.1   Evaluating $y\varphi_t(u, y)$

We note that $y\varphi_t(u, y)$ can be evaluated analytically only for special cases. Specifically, when $\varphi_t(u, y)$ is a Gaussian process, then the joint density function $y\varphi_t(u, y)$ is analytically computable. Consider the dynamical system described by (2). When p is Gaussian and f is linear, we have

$$\mathbf{x}' = A\,\mathbf{x}(t) + b\,, \quad \mathbf{x}(t_0) = \mathbf{x}_0\,, \quad \mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}}_0, \Sigma)\,. \tag{9}$$

Assuming A is invertible, $\mathbf{x}(t)$ can be written as

$$\mathbf{x}(t) = \exp(A(t - t_0))\,\mathbf{x}_0 - (I - \exp(A(t - t_0)))\,A^{-1}b\,, \tag{10}$$

where I represents an identity matrix of the appropriate size. Given that $\mathbf{x}_0$ is normally distributed, it follows that $\mathbf{x}(t)$ is a Gaussian process:

$$\mathbf{x}(t) \sim \mathcal{GP}\left(\overline{\mathbf{x}}, \mathrm{cov}_{\mathbf{x}}\right)\,, \text{ where} \tag{11}$$
$$\overline{\mathbf{x}} = \exp(A(t - t_0))\overline{\mathbf{x}}_0 - (I - \exp(A(t - t_0)))\,A^{-1}b \text{ and}$$
$$\mathrm{cov}_{\mathbf{x}} = \exp(A(t - t_0))\Sigma\left(\exp(A(t - t_0))\right)^{\top}\,.$$

The joint probability density function (PDF) of $\mathbf{c}^{\top}\mathbf{x}(t)$ and $\mathbf{c}^{\top}\mathbf{x}'(t)$ is given by [26, equation 9.1]

$$\begin{bmatrix}\mathbf{c}^{\top}\mathbf{x} \\ \mathbf{c}^{\top}\mathbf{x}'\end{bmatrix} \sim \mathcal{GP}\left(\overline{\mathbf{x}}^{\varphi}, \begin{bmatrix}\mathbf{c}^{\top}\Phi\mathbf{c} & \mathbf{c}^{\top}\Phi A^{\top}\mathbf{c} \\ \mathbf{c}^{\top}A\Phi^{\top}\mathbf{c} & \mathbf{c}^{\top}A\Phi A^{\top}\mathbf{c}\end{bmatrix}\right)\,, \tag{12}$$

where

$$\overline{\mathbf{x}}^{\varphi} := \begin{bmatrix}\mathbf{c}^{\top}\overline{\mathbf{x}} \\ \mathbf{c}^{\top}(A\overline{\mathbf{x}} + b)\end{bmatrix}$$

and

$$\Phi := \exp(A(t - t_0))\Sigma\left(\exp(A(t - t_0))\right)^{\top}\,.$$

We now can evaluate $y\varphi_t(u, y)$ for arbitrary values of u_i, y_i, and t_i as

$$y_i\varphi_{t_i}(u_i, y_i) = \frac{y_i}{2\pi \mid \Upsilon \mid} \exp\left(-\frac{1}{2}\left\| \begin{bmatrix} u_i \\ y_i \end{bmatrix} - \overline{\mathbf{x}}^\varphi \right\|_{\Upsilon^{-1}}^2\right), \tag{13}$$

where $\Upsilon := \begin{bmatrix} \mathbf{c}^\top \Phi \mathbf{c} & \mathbf{c}^\top \Phi A^\top \mathbf{c} \\ \mathbf{c}^\top A \Phi^\top \mathbf{c} & \mathbf{c}^\top A \Phi A^\top \mathbf{c} \end{bmatrix}$ and $\mid \Upsilon \mid$ denotes the determinant of Υ. Note that the right-hand side in (13) is dependent on t_i via Υ.

3.2   Notes for nonlinear f

When f is nonlinear, $y\varphi_t(u, y)$ cannot be computed analytically—a key ingredient for our computational procedure. We approximate the nonlinear dynamics by linearizing f around the mean of the initial distribution. Assuming that the initial state of the system is normally distributed as described by equation (9), linearizing around the mean of the initial state gives

$$\mathbf{x}' \approx \mathbf{F} \cdot (\mathbf{x} - \overline{\mathbf{x}}_0) + f(\overline{\mathbf{x}}_0, 0), \tag{14}$$

where $\mathbf{F}$ represents the Jacobian of f at $t = 0$ and $\mathbf{x} = \overline{\mathbf{x}}_0$; this reduces the nonlinear dynamical system to a form that is similar to equation (9). Thus, we can now use equations (11), (12), and (13) to approximate $y\varphi_t(u, y)$ for nonlinear f.

4   Machine-learned inverse maps

In [25] we formulated the problem of determining preimages (7) as a Bayesian inverse problem. However, solving multiple Bayesian inverse problems can be expensive. Hence we approximated our IBD by using the solutions of a small number of Bayesian inverse problems. In this section we build a simple data-driven surrogate for approximating the preimages X_i described in equation (7). Using the surrogate, we can approximate the preimages of several $\mathbf{y}_i$'s obtained by sampling from $y\varphi_t(u, y)$. The surrogate developed here approximates the inverse of the map defined in equation (6). To that end, we wish to approximate the map

$$\mathcal{G}^{-1} : \mathbb{R}^2 \to \mathbb{R}^d, \tag{15}$$

where the input space corresponds to $(u + \varepsilon_i, y)\mid_{t_i}$ and the output lives in the domain of the state space (Ω here). This is equivalent to augmenting t_i as an additional input variable and building a surrogate that maps from $\mathbb{R}^3 \to \mathbb{R}^d$. We utilize a fully connected deep neural network to approximate this map. A one-layered neural network can be expressed as

$$\xi_j = F\left(\sum_{\ell=1}^{L} c_m^\ell x_\ell + \epsilon_m\right), \tag{16}$$

where F is a differentiable activation function that imparts nonlinearity to this transformation; L is the input dimension of an incoming signal; M is the number of hidden-layer neurons (in machine learning terminology); $c_m^\ell \in \mathbb{R}^{M \times L}$ are the weights of this map; $\epsilon_m \in \mathbb{R}^M$ are the biases; and $\xi_j \in \mathbb{R}^J$ is the nonlinear output of this map, which may be matched to targets available from data or "fed-forward" into future maps. Note that ξ_j is the postactivation value of each neuron in a hidden layer of J neurons. In practice, multiple compositions of this map may be used to obtain nonlinear function approximators, called deep neural networks, that are very expressive. For nonlinear activation, we utilize

$$F(\xi) = \max(\xi, 0), \tag{17}$$

for all its activation functions. In addition, we concatenate three such maps as shown in equation 16 to ultimately obtain an approximation for $\mathcal{G}^{-1}$. Two such submaps have J fixed at 256, and a final transformation utilizes $J = 3$. We note that the function F for the final transformation is the identity, as is common in machine learning algorithms. A schematic of this network architecture is shown in Figure 1. The trainable parameters (c_m^ℓ and ϵ_m for each transformation) are optimized with the use of backpropagation [30], an adjoint calculation technique that obtains gradients of the loss function with respect to these parameters. A stochastic gradient optimization technique, ADAM, is used to update these parameters [20] with a learning rate of 0.001. Our loss function is given by the L_2-distance between the prediction of the network and the targets (i.e., the mean-squared error). Our network also incorporates a regularization strategy, called dropout [31], that randomly switches off certain units ξ_j (here we utilize a dropout probability of 0.1) in the forward propagation of the map (i.e., from $d \to 2$). Through this approach, memorization of data is avoided, while allowing for effective exploration of a complex nonconvex loss surface.

Our map is trained for 500 epochs with a batch size of 256; in other words, a weight update is performed after a loss is computed for 256 samples. Each epoch is completed when the losses from the entire data set are used for gradient update. During the network training, we set aside a random subset of the data for validation. Losses calculated from this data set are used only to monitor the learning of the framework for unseen data. These are plotted in Figure 2, where one can see that both training and validation losses are reduced to an equal magnitude. Figure 3 also shows scatter plots for this validation data set where a good agreement between the true and predicted quantities can be seen. We may now use this map for approximating the IBD.

### 4.1   Using the machine-learned inverse map to construct IBD

The following procedure is used to construct the IBD.

1. Obtain different realizations of the initial conditions of the dynamical system by sampling from the initial PDF p.
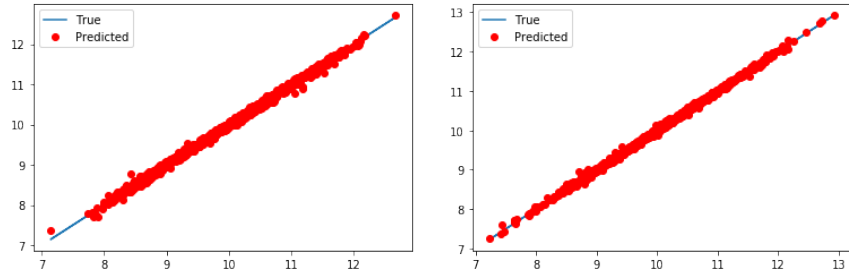2. Use $\mathcal{G}$ to obtain the forward maps of these realizations.
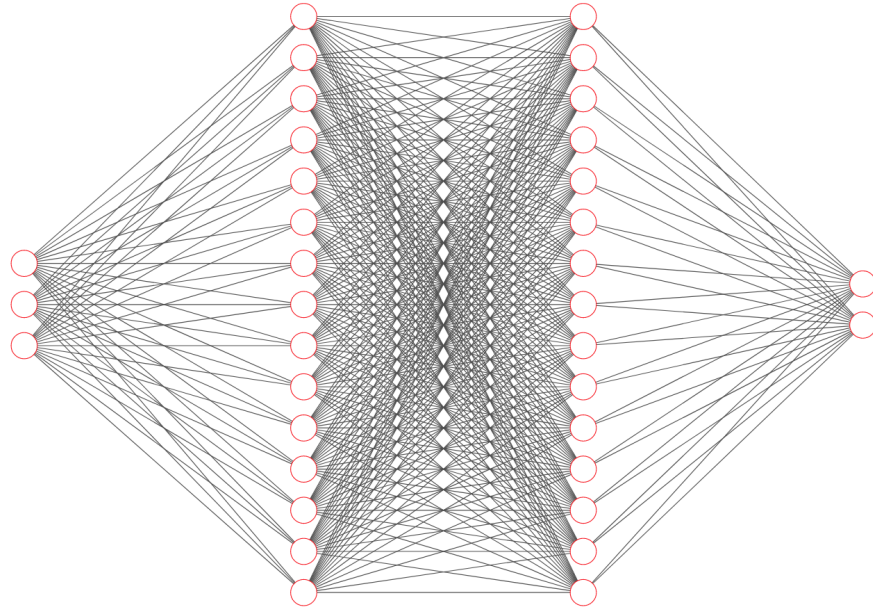
Fig. 1. Schematic of our neural network architecture. Note that the number of hidden layer units are not representative since this study utilizes 256 such units.
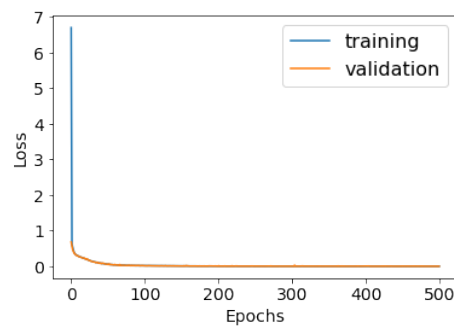
Fig. 2. Convergence of training for our network. Note how both training and validation losses diminish in magnitude concurrently.

Fig. 3. Scatter plots between truth and predicted quantites of the inverse map with dimension 1 (left) and dimension 2 (right). These results are from unseen data.

3. Use the forward maps and the corresponding random realizations of the initial conditions to train the inverse map $\mathcal{G}^{-1}$.
4. We now apply this trained inverse map on samples generated from $y\varphi_t(u, y)$ to obtain the approximate preimages of samples $\mathbf{y}_i$.
5. Use a Gaussian approximation of these inverse maps is used as an IBD. Assume that this Gaussian approximation has PDF p^{IBD}.
6. Sample from the IBD, and use importance sampling to estimate the probabilities.

4.2   Using IBD to estimate rare event probability

We can now estimate $P_T(u)$ using the IBD as follows:

$$P_T^{\mathrm{IS}}(u)(\widehat{\mathbf{x}}_0^1, \ldots, \widehat{\mathbf{x}}_0^M) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}(\widehat{\mathbf{x}}_0^i)\psi(\widehat{\mathbf{x}}_0^i)\,, \tag{18}$$

where $\widehat{\mathbf{x}}_0^1, \ldots, \widehat{\mathbf{x}}_0^M$ are sampled from the biasing distribution p^{IBD} and $\mathbb{I}(\widehat{\mathbf{x}}_0^i)$ represents the indicator function given by

$$\mathbb{I}(\widehat{\mathbf{x}}_0^i) = \begin{cases} 1\,, & \sup_{0 \leq t \leq T} \mathbf{c}^\top \mathbf{x}(t, \widehat{\mathbf{x}}_0^i) \geq u\,, & t \in [0, T]\,, \\ 0\,, & \sup_{0 \leq t \leq T} \mathbf{c}^\top \mathbf{x}(t, \widehat{\mathbf{x}}_0^i) < u\,, & t \in [0, T]\,. \end{cases} \tag{19}$$

Also, $\psi(\widehat{\mathbf{x}}_0^i)$ represents the importance weights. The importance weight for an arbitrary $\widehat{\mathbf{x}}_0^i$ is given by

$$\psi(\widehat{\mathbf{x}}_0^i) = \frac{p(\widehat{\mathbf{x}}_0^i)}{p^{\mathrm{IBD}}(\widehat{\mathbf{x}}_0^i)}\,. \tag{20}$$

5   Numerical experiments

We demonstrate the application ofthe procedure described in Section 3 and Section 4 for nonlinear dynamical systems excited by a Gaussian distribution. We

use the Lotka-Volterra equations as a test problem. These equations, also known as the predator-prey equations, are a pair of first-order nonlinear differential equations and are used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as a prey. The populations change through time according to the following pair of equations,

$$\frac{dx_1}{dt} = \alpha x_1 - \beta x_1 x_2 \,, \tag{21}$$
$$\frac{dx_2}{dt} = \delta x_1 x_2 - \gamma x_2 \,,$$

where x_1 is the number of prey, x_2 is the number of predators, and $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ represent the instantaneous growth rates of the two populations. We assume that the initial state of the system at time $t = 0$ is a random variable that is normally distributed:

$$\mathbf{x}(0) \sim \mathcal{N}\left(\begin{bmatrix} 10 \\ 10 \end{bmatrix}, 0.8 \times I_2\right),$$

and we are interested in estimating the probability of the event $P(\mathbf{c}^\top \mathbf{x} \geq u)$, where $\mathbf{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $t \in [0, 10]$, and $u = 17$. The first step of our solution procedure involves sampling from $y\varphi_t(u, y)$ to generate observations $\mathbf{y}_i$. We linearize the dynamical system about the mean of the distribution of $\mathbf{x}_0$ (equation (14)) and express $\varphi_t(u, y)$ as a function of t and y as described by equation (12). We compute $y\varphi_t(u, y)$ as shown in equation (13). We use the delayed rejection adaptive Metropolis (DRAM) Markov chain Monte Carlo (MCMC) method to generate samples from $y\varphi_t(u, y)$. (For more details about DRAM, see [15].) To minimize the effect of the initial guess on the posterior inference, we use a burn-in of 1,000 samples. Figure 4 shows the contours of $y\varphi_t(u, y)$ and samples drawn from it by using DRAM MCMC. In [25] we then solved the Bayesian inverse problem by using both MCMC and Laplace approximation at MAP to construct a distribution that approximately maps to likelihood constructed around $\mathbf{y}_i$. Here, we replace the solution to the Bayesian inverse problem with a machine-learned inverse map described in Section 4. Multiple samples generated from $y\varphi_t(u, y)$ can be used to construct the IBD, as described in §4.1, and the IBD can be used to estimate $P_T(u)$, as explained in §4.2. Figure 5 compares the results between the conventional MCS and Machine Learning based Importance Sampling (ML-based IS) methods. Note that we use an MCS estimate with 10 Million samples as a proxy for the true probabilities. The "true" probability is 3.28×10^{-5}. ML-based IS gives fairly good estimate even with small number of model evaluations. When the training dataset size is large enough, the improvements are dramatic. Notice that for a true probability of the order of 10^{-5} we obtain an estimate that has a relative error of less than 1%. Notice that our method gives the same (or better) accuracy as the MCS with hundred times lesser computational cost. The convergence with just 5000 training samples is acceptable and these results improve
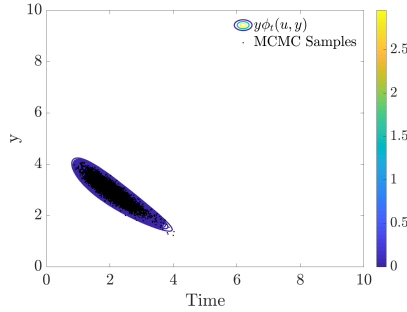
Fig. 4. Contours of $y\varphi_t(u, y)$ and samples drawn from it using DRAM MCMC.
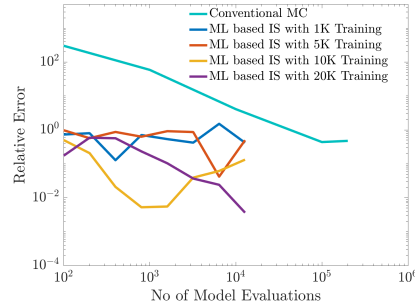
Fig. 5. Comparison between conventional MCS and ML-based IS. We observe even with a small amount of training data, we obtain fairly accurate estimates; and as we increase the training data, the accuracy improves dramatically.

dramatically for 10000 and 20000 training samples. We believe the results could be even better when we use a Gaussian mixture to represent the IBD instead of a simple Gaussian approximation.

5.1   Computational Cost

In Figure 5, we havent included the costs associated with generating training data, training costs, and cost for approximating the inverse map as these costs are almost negligible when compared to the overall costs. Note that generating 20000 samples is approximately equivalent to 400 model evaluations (this is because a single model evaluation can be used to generate the slope and state at 50 different times and each of them can be used as a training sample). The training of the ML framework, for this problem, required very little compute time. Each training was executed on an 8th-generation Intel Core-I7 machine with Python 3.6.8 and Tensorflow 1.14 and took less than 180 seconds for training 20000 samples (this is less than 50 model evaluations). Inference (for 20000 prediction points) costs were less than 2 seconds, on average.

6   Conclusions and future work

In this work we developed a ML-based IS to estimate rare event probabilities and we demostrated the algorithm on the prey-predator system. The method developed here builds on the approach in [25] and replaces the expensive Bayesian inference with a Machine learning based surrogate. This approach yields fairly accurate estimate of the probabilities and for a given accuracy requires atleast three orders of magnitude lesser computational effort than the traditional MCS.

In future, we aim to test this algorithm for larger problems and also use an active learning based approach to pick the training samples. Scaling this algorithm to high dimensions (say $\mathcal{O}(1000)$) could be challenging and to address it, we will use state-of-the-art techniques developed by machine learning and deep learning community in the future.

References

1. Adler, R.J.: The geometry of random fields. SIAM (2010)
2. Asmussen, S., Glynn, P.W.: Stochastic simulation: algorithms and analysis, vol. 57. Springer Science & Business Media (2007)
3. Atputharajah, A., Saha, T.K.: Power system blackouts-literature review. In: Industrial and Information Systems (ICIIS), 2009 International Conference on. pp. 460–465. IEEE (2009)
4. Au, S.K., Beck, J.L.: Estimation of small failure probabilities in high dimensions by subset simulation. Probabilistic engineering mechanics **16**(4), 263–277 (2001)
5. Beck, J.L., Zuev, K.M.: Rare-event simulation. Handbook of uncertainty quantification pp. 1–26 (2016)
6. Bect, J., Li, L., Vazquez, E.: Bayesian subset simulation. SIAM/ASA Journal on Uncertainty Quantification **5**(1), 762–786 (2017)
7. Botev, Z.I., Kroese, D.P.: Efficient monte carlo simulation via the generalized splitting method. Statistics and Computing **22**(1), 1–16 (2012)
8. Bucklew, J.: Introduction to rare event simulation. Springer Science & Business Media (2013)
9. Ching, J., Beck, J.L., Au, S.: Hybrid subset simulation method for reliability estimation of dynamical systems subject to stochastic excitation. Probabilistic engineering mechanics **20**(3), 199–214 (2005)
10. Ching, J., Au, S.K., Beck, J.L.: Reliability estimation for dynamical systems subject to stochastic excitation using subset simulation with splitting. Computer methods in applied mechanics and engineering **194**(12-16), 1557–1579 (2005)
11. Dematteis, G., Grafke, T., Vanden-Eijnden, E.: Rogue waves and large deviations in deep sea. Proceedings of the National Academy of Sciences **115**(5), 855–860 (2018)
12. Dematteis, G., Grafke, T., Vanden-Eijnden, E.: Extreme event quantification in dynamical systems with random components. SIAM/ASA Journal on Uncertainty Quantification **7**(3), 1029–1059 (2019)
13. Dunn, W.L., Shultis, J.K.: Exploring Monte Carlo methods. Elsevier (2011)
14. Dysthe, K., Krogstad, H.E., Müller, P.: Oceanic rogue waves. Annu. Rev. Fluid Mech. **40**, 287–310 (2008)
15. Haario, H., Laine, M., Mira, A., Saksman, E.: Dram: efficient adaptive mcmc. Statistics and computing **16**(4), 339–354 (2006)
16. Kahn, H., Harris, T.E.: Estimation of particle transmission by random sampling. National Bureau of Standards applied mathematics series **12**, 27–30 (1951)
17. Kahn, H., Marshall, A.W.: Methods of reducing sample size in monte carlo computations. Journal of the Operations Research Society of America **1**(5), 263–278 (1953)
18. Katafygiotis, L., Cheung, S.H.: A two-stage subset simulation-based approach for calculating the reliability of inelastic structural systems subjected to gaussian random excitations. Computer methods in applied mechanics and engineering **194**(12-16), 1581–1595 (2005)

19. Katafygiotis, L.S., Zuev, K.M.: Geometric insight into the challenges of solving high-dimensional reliability problems. Probabilistic Engineering Mechanics **23**(2-3), 208–218 (2008)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer Science & Business Media (2008)
22. Metropolis, N., Ulam, S.: The monte carlo method. Journal of the American statistical association **44**(247), 335–341 (1949)
23. Mohamad, M.A., Sapsis, T.P.: A sequential sampling strategy for extreme event statistics in nonlinear dynamical systems. arXiv preprint arXiv:1804.07240 (2018)
24. Mohamad, M.A., Sapsis, T.P.: Sequential sampling strategy for extreme event statistics in nonlinear dynamical systems. Proceedings of the National Academy of Sciences **115**(44), 11138–11143 (2018)
25. Rao, V., Anitescu, M.: Efficient computation of extreme excursion probabilities for dynamical systems (2020)
26. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA (1 2006)
27. Rice, S.O.: Mathematical analysis of random noise. Bell Labs Technical Journal **23**(3), 282–332 (1944)
28. Robert, C.P., Casella, G.: Monte carlo statistical methods (springer texts in statistics) (2005)
29. Ross, T., Lott, N.: A climatology of 1980-2003 extreme weather and climate events. US Department of Commerece, National Ocanic and Atmospheric Administration, National Environmental Satellite Data and Information Service, National Climatic Data Center (2003)
30. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)
31. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
32. "The North American Electricity Reliability Corporation": 2017 long-term reliability assessment. online (2017), https://www.nerc.com/pa/RAPA/ra/Reliability
33. Zuev, K.M., Beck, J.L., Au, S.K., Katafygiotis, L.S.: Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. Computers & Structures **92**, 283–296 (2012)