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

NEURAL PREDICTIVE INTERVALS FOR SIMULATION METAMODELING

Henry Lam
Haofeng Zhang

Department of Industrial Engineering and Operations Research
Columbia University
500 W. 120th Street
New York, NY 10027

ABSTRACT

Simulation metamodeling refers to the use of lower-fidelity models to represent input-output relations with few simulation runs. Stochastic kriging, which uses Gaussian process and captures both aleatoric and epistemic uncertainties, is a versatile and predominant technique for such a task. However, this approach relies on specific model assumptions and could encounter scalability challenges. In this paper, we study an alternate metamodeling approach using neural-network-based input-output prediction intervals. We cast the metamodeling into an empirical constrained optimization framework to train the neural network that attains accurate prediction in terms of coverage and prediction interval width. We present a validation machinery and show how our method can enjoy a distribution-free finite-sample guarantee on the prediction performance. We demonstrate the superior performance of our method compared with other methods including stochastic kriging through some numerical examples.

1 INTRODUCTION

Stochastic simulation aims to compute output summaries from complex stochastic models that could not be handled analytically. Often the random output quantity, say $Y(x)$, depends on a certain input factor or design parameter x, and $y(x) = E[Y(x)]$ is called the (mean) response surface. Building a response surface serves a range of usages across sensitivity analysis and optimization, or simply for visualization or an understanding of the relation landscape. It is especially useful when the simulation run is computationally expensive. In this situation, we may not be able to run enough simulations to estimate the value of $y(x)$ at each prediction point x whenever the need comes up. It is thus useful to get an approximate response surface by running a number of simulation runs at possibly various points of x in advance, and building this surface using regression tools. This task is often known as simulation metamodeling in the literature (Barton and Meckesheimer 2006; Staum 2009).

A benchmark approach in simulation metamodeling is stochastic kriging (SK) (Ankenman et al. 2008; Ankenman et al. 2010), which is a versatile technique applicable to general nonlinear input-output relations based on Gaussian process (GP) regression. SK can be viewed as a generalization of kriging (Stein 1999; Kleijnen 2009), which captures only epistemic or extrinsic uncertainty (i.e., error coming from model assumption or fitting error), to include aleatory or intrinsic uncertainty (i.e., the stochasticity of the system itself that cannot be washed away with enough fitting data), by including extra variances in the Gaussian process. In the literature, the estimator produced by SK typically focuses on mean response surface estimation (i.e., the $y(x)$ above) or quantile-based response measures (Chen and Kim 2013; Bekki et al. 2014; Chen and Kim 2016), but enhanced estimators that measure aleatory uncertainty, such as the estimated variance or prediction intervals, could be naturally produced as a by-product. In this paper, we will primarily focus on techniques that account for and produce measures to quantify both uncertainties.

Despite the strengths and popularity of SK, it incurs some potential limitations. First is that it relies on the normality of aleatory uncertainties. This assumption can potentially be relaxed by running a large number of simulation runs per design point and invoking the central limit theorem. However, this approach cannot capture the distributional shape of the aleatory uncertainty, and also enforces the use of many runs per point that could add computational demand. Moreover, to estimate the aleatory variance, one typically plugs in the sample variance at each design point, which again requires multiple responses at each design point. Second, the mean estimation or prediction involves matrix inversion that could lead to two computational issues: a) calculating matrix inversion is computationally demanding when the matrix size is large; b) near-singularity of the matrix could arise if simulation outputs of some of design points are too highly correlated, e.g., when design points are too close to each other (Staum 2009). In general, when a large number of design points are available, we should expect a method to perform well because of sufficient data. Unfortunately, in this ideal case both computational issues of SK arise and make it less applicable. Third, while SK enjoys attractive Bayesian interpretation, little is known about the finite-sample frequentist guarantee on its prediction performance, which could depend heavily on the prior correlation structure imposed on the GP.

Motivated by the above, in this paper we propose an alternate simulation metamodeling method built on neural-network-based prediction intervals. This method outputs, for each design point x, a lower bound $L(x)$ and upper bound $U(x)$ that cover the random output $Y(x)$ with high probability in some sense. To be more precise, we utilize the notion of so-called *expected coverage rate*, a recent criterion for constructing high-quality prediction intervals (Khosravi et al. 2010; Pearce et al. 2018; Rosenfeld et al. 2018; Chen et al. 2021). The expected coverage rate measures the probability of $[L(X),U(X)]$ covering $Y(X)$, when X and Y are both viewed as random. Here, the randomness of Y is simply the aleatory uncertainty or stochasticity of simulation, and the randomness of X is chosen by the user, which is often uniform over the input domain (but can be a more general distribution). Because in simulation metamodeling researchers are interested in fitting globally and equally for each X, one common objective is to minimize the *integrated mean squared error* (IMSE) (Sacks et al. 1989; Ankenman et al. 2010), which implies that a uniform distribution of X is assigned. In addition, we also allow the possibility of a more general distribution on X. This will enable users to assign different weights on different input points and similarly develop a "weighted"-integrated MSE as a criterion.

Let us explain the high-quality criterion mentioned above. The expected coverage rate itself cannot well characterize the performance of prediction intervals since a sufficiently wide prediction interval can cover any random output $Y(x)$. Thus interval width has been a common metric to measure the conservativeness of prediction intervals (Barber et al. 2019; Zhang et al. 2019). From this view, constructing a high-quality prediction interval can be formalized as a constrained stochastic optimization problem that optimizes the expected interval width while maintaining the expected coverage rate (Rosenfeld et al. 2018; Chen et al. 2021).

In this work, we propose a neural-network-based method to train a high-quality prediction interval based on this constrained optimization problem. While this problem is difficult to solve directly, it provides a framework for empirical training with the following details. First we consider its empirical formulation to approximate this problem. We derive an empirical loss function from its Lagrangian function with the Lagrangian multiplier as a hyper-parameter. With the lower and upper bounds given by a neural network, we optimize this neural network via gradient descent. Finally, the Lagrangian multiplier needs to be properly calibrated, and we propose an easy-to-implement validation strategy to select the best model that can guarantee the coverage attainment with prefixed confidence. This work extends our previous work (Chen et al. 2021) to simulation metamodeling, with a generalization to handle possibly multiple simulation runs per design point.

Our method has the following advantages: (1) Our method is arguably computationally scalable. The high efficiency of our method stems from recent advances in training a neural network such as Adam gradient descent (Kingma and Ba 2014) and mini-batch training manner (Li et al. 2014). (2) Our method is distribution-free in the sense that we do not impose distribution assumption and structure on the data. This avoids the common mismatches between data and models, for example, if the GP structure assumption is violated, or if the estimations of the mean or variance of GPs are unreliable. (3) Our method does not require multiple simulation runs per input point, which is essential for SK to make reliable estimation of aleatory variance. (4) Our method enjoys some finite-sample guarantees on the overall coverage of the simulation output.

We close this introduction by reviewing several mainstream approaches for constructing prediction intervals. The first is to estimate quantiles and convert them into intervals. This approach includes classical quantile regression (Koenker and Hallock 2001), quantile regression forests (Meinshausen 2006) and quantile-based stochastic kriging (Chen and Kim 2013; Bekki et al. 2014). However, little is known about their finite-sample performance. The second is conformal prediction. Conformal prediction uses past training data to determine precise levels of confidence in future predictions (Shafer and Vovk 2008; Lei et al. 2018). It can generate distribution-free prediction intervals, some of which enjoy finite-sample coverage guarantees. The third is using deep learning. Neural networks have achieved impressive performance in constructing high-quality prediction intervals (Khosravi et al. 2010; Pearce et al. 2018; Chen et al. 2021), and training a neural network now has become efficient and computationally cheap. Moreover, a neural network provides a more general class of functions rather than linear functions or quadratic functions that are widely used as the "trend" functions in metamodel (Law et al. 2000) and thus it can handle more sophisticated data structure. However, most of these works are empirical. Our work follows this stream but enjoys a stronger statistical guarantee about coverage than conformal prediction and empirically produces less conservative intervals. Lastly, while our work focuses primarily on constructing prediction intervals to quantify uncertainty, in situations where a mean response prediction is also desired, we can integrate our approach with other works such as Thiagarajan et al. (2020) to estimate means and prediction intervals simultaneously.

2 STOCHASTIC KRIGING

Before introducing our method, we first briefly review the predominant approach in simulation metamodeling, SK. Although the literature has primarily focused on mean response prediction, SK can also be used naturally to generate prediction intervals by leveraging the posterior predictive distribution.

We consider the following setting. We have a design variable $X \in \mathscr{X} \subset \mathbb{R}^d$. Suppose for each input value $x_i \in \mathscr{X}$ in $\{x_1, x_2, ..., x_n\}$, we run simulation replications of size r_i at x_i and obtain the simulation output $y_{i,j} \in \mathbb{R}$ where $j = 1, .., r_i$. Let the sample mean of responses at x_i be $\bar{y}_i = \frac{1}{r_i} \sum_{j=1}^{r_i} y_{i,j}$ and $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, ..., \bar{y}_n)^T$ in short.

SK makes the following assumption where a stochastic simulation's output is the summation of the extrinsic Gaussian process M, whose realization is the response surface, and an independent intrinsic noise ε (Ankenman et al. 2010):

$$Y_j(x) = M(x) + \varepsilon_j(x), \; M \sim GP(\mu, \sigma^2), \; \varepsilon_j \sim GP(0, c) \tag{1}$$

where $GP(\mu, \sigma^2)$ is a Gaussian process (GP) with mean $\mu(x)$ and covariance between any two points x and x' given by $\sigma^2(x, x')$. The two GPs M and ε_j are assumed to be independent. The intrinsic noise $\varepsilon_1(x), \varepsilon_2(x), ...$ at the same input point x is naturally independent and identically distributed across replications. For different x's, it follows the $GP(0, c)$ structure. In this model, μ represents the input-output trend, σ^2 represents the epistemic uncertainty and c represents the aleatoric uncertainty. In particular, if $\varepsilon_j \equiv 0$, then this reduces to kriging (Stein 1999). If $c(x, x) \equiv c_0$ and $c(x, x') = 0$ if $x \neq x'$ (i.e., i.i.d. Guassian noise at every input point), then this is kriging with measurement error (Cressie 1993). Note that the normality of the intrinsic noise $\varepsilon_j(x)$ is an additional assumption proposed by the standard SK model to show the rationality of the SK prediction (Staum 2009; Ankenman et al. 2010; Chen and Kim 2014). Consequently, standard SK does not directly estimate the distributional shape of aleatory uncertainty.

To implement the above estimation in practice, we need to assume some structure on the mean and covariance functions. The original work of SK (Ankenman et al. 2010) suggests that one could use, for example, $\mu(x) = \boldsymbol{f}(x)^T\beta$ where $\boldsymbol{f}$ is a vector of known functions of x and β is a vector of unknown parameters of compatible dimension, and $\sigma^2(x,x') = \tau^2 r_\theta(|x-x'|)$ where r_θ is chosen from a set of functions parameterized by some unknown parameters θ. These parameters can be calibrated via maximum likelihood estimation.

The SK prediction at a new point $\tilde{x}$ is defined as follows (Ankenman et al. 2010):

$$\tilde{\mu}(\tilde{x}) = \mu(\tilde{x}) + \sigma^2(\tilde{x},\mathbf{x})(\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x}))^{-1}(\bar{\mathbf{y}}-\mu(\mathbf{x})).$$

Here we use the shorthand

$$\sigma^2(\tilde{x},\mathbf{x}) = (\sigma^2(\tilde{x},x_1),...,\sigma^2(\tilde{x},x_n)),\ \ \sigma^2(\mathbf{x},\tilde{x}) = (\sigma^2(x_1,\tilde{x}),...,\sigma^2(x_n,\tilde{x}))^T = \sigma^2(\tilde{x},\mathbf{x})^T,$$

$$\sigma^2(\mathbf{x},\mathbf{x}) = (\sigma^2(x_i,x_j))_{i=1,...n;j=1,...,n},\ \ \mu(\mathbf{x}) = (\mu(x_1),\mu(x_2),...,\mu(x_n))^T$$

$$c(\mathbf{x},\mathbf{x}) = \Big(c(x_i,x_j)\big(\mathbf{1}_{(i\neq j)} + \frac{1}{r_i}\mathbf{1}_{(i=j)}\big)\Big)_{i=1,...n;j=1,...,n},$$

where $\mathbf{x} = (x_1,x_2,...,x_n)^T$ and $\bar{\mathbf{y}} = (\bar{y}_1,\bar{y}_2,...,\bar{y}_n)^T$.

We consider the linear predictors of the form $\lambda_0 + \boldsymbol{\lambda}^T\bar{\mathbf{y}}$ to predict the true mean response. Then Appendix EC.1 of Ankenman et al. (2010) shows that the SK prediction is the MSE-optimal predictor. In addition, in general (if we drop out the form of linear predictors) the SK prediction can be alternatively interpreted as the posterior mean with GP assumptions on the aleatory and epistemic uncertainty in (1). To see this, we view the model (1) as the prior belief and then observe the simulation data. Provided the validity of the specified form of uncertainty, SK naturally leads to generation of a prediction interval (in a Bayesian sense) under the posterior predictive distribution.

Lemma 1 Suppose the SK model (1) is a prior belief. The simulation data we observe is (x_i, y_{ij}) where $j = 1,...,r_i$ and $i = 1,...n$. Then the posterior distribution of simulation output $\tilde{y}$ at an unobserved point $\tilde{x}$ is a Gaussian distribution with mean

$$\tilde{\mu}(\tilde{x}) = \mu(\tilde{x}) + \sigma^2(\tilde{x},\mathbf{x})(\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x}))^{-1}(\bar{\mathbf{y}}-\mu(\mathbf{x}))$$

and variance

$$\tilde{\sigma}^2(\tilde{x}) = \sigma^2(\tilde{x},\tilde{x}) + \sigma^2(\tilde{x},\mathbf{x})(\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x}))^{-1}\sigma^2(\mathbf{x},\tilde{x}).$$

It is then easy to see that the $1-\alpha$ prediction interval based on the SK model (1) is

$$[\tilde{\mu}(\tilde{x}) - \lambda\tilde{\sigma}(\tilde{x}), \tilde{\mu}(\tilde{x}) + \lambda\tilde{\sigma}(\tilde{x})]$$

where λ is the $1-\alpha/2$ quantile of the standard univariate Gaussian distribution.

The SK prediction requires us to evaluate the matrix inversion $(\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x}))^{-1}$ as in Lemma 1. To slightly reduce the computational complexity, Staum (2009) suggests to apply cholesky factorization on the matrix $\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x})$ since it is non-negative, which takes complexity of $O(n^3)$ in the number n of design points. This step is computationally demanding especially when the matrix size is large. Another computational issue is that numerical problems about near-singularity of $\sigma^2(\mathbf{x},\mathbf{x})+c(\mathbf{x},\mathbf{x})$ arise if simulation outputs of some of design points are too highly correlated (Staum 2009).

Lemma 1 assumes that the structures of μ, σ^2 and c are known to get the prediction. In practice, we should also estimate μ, σ^2 and c based on the data. Ankenman et al. (2010) provide some guidance. For example, they suggest to plug in the sample variance at each design point to estimate the aleatoric variance $c(\mathbf{x},\mathbf{x})$. Since one or very few responses at each design point will make the variance estimation unreliable, this requires us to have access to multiple responses at the design point, which could add simulation demand. We also point out that the posterior predictive distribution from Lemma 1 is based on the normality assumption of the intrinsic noise in (1). It is beyond the scope of standard SK if the normality assumption is violated by the actual data.

In general, SK performance depends on the appropriate choice of these structures in the model and little is known about its finite-sample guarantee on the prediction performance. In the following, we consider a prediction interval approach to capture epistemic and aleatory uncertainties in metamodeling which enjoys performance guarantees.

3 PREDICTION INTERVALS AND CONFORMAL PREDICTION

We introduce another way to communicate uncertainty, which is called a prediction interval. To be rigorous, we assume that the simulation data (x_i, y_i) follow a general joint distribution π. Note that unlike Equation (1) in Section 2, we do not make any assumptions about the distribution π throughout this section.

Definition 1 An interval $[L(x), U(x)]$, where both $L, U : \mathscr{X} \to \mathbb{R}$, is called a strong prediction interval with prediction level $1-\alpha$ $(0 \leq \alpha \leq 1)$ if

$$\mathbb{P}_{\pi(Y|X)}(Y \in [L(X), U(X)] | L, U, X) \geq 1-\alpha, \quad \forall \text{ a.e. } X \tag{2}$$

where $\mathbb{P}_{\pi(Y|X)}$ denotes the probability with respect to the conditional distribution $\pi(Y|X)$ with L, U, X fixed.

Although this definition of prediction interval is appealing and desirable, it is intangible to measure and difficult to achieve in general without assuming some simple structure on the joint distribution π. Quantile regression forests (Meinshausen 2006), quantile-based stochastic kriging (Chen and Kim 2013), and random forest prediction intervals (Zhang et al. 2019) could be applied for this target but no finite-sample guarantee is known. Therefore, we focus on the following relaxation of prediction intervals (Rosenfeld et al. 2018; Chen et al. 2021).

Definition 2 An interval $[L(x), U(x)]$, where both $L, U : \mathscr{X} \to \mathbb{R}$, is called a prediction interval with prediction level $1-\alpha$ $(0 \leq \alpha \leq 1)$ if

$$\mathbb{P}_{\pi}(Y \in [L(X), U(X)] | L, U) \geq 1-\alpha \tag{3}$$

where $\mathbb{P}_{\pi}$ denotes the probability with respect to the joint distribution π on (X, Y) with L, U fixed.

For example, a prediction interval with 95% prediction level is expected to cover at least 95 percent of the simulation outputs. This definition is in parallel with the IMSE criterion (Ankenman et al. 2010) since the expected coverage rate is defined globally for all X. If the prediction interval at x has a very high width $U(x) - L(x)$, we can interpret that it has high uncertainty at point x. In addition, there is another type of definition of a prediction interval, which we name a weak prediction interval. This definition is widely-used in the area of conformal prediction (Lei et al. 2018).

Definition 3 An interval $[L(x), U(x)]$, where both $L, U : \mathscr{X} \to \mathbb{R}$, is called a weak prediction interval with prediction level $1-\alpha$ $(0 \leq \alpha \leq 1)$ if

$$\mathbb{P}_{\pi^{n+1}}(Y \in [L(X), U(X)]) \geq 1-\alpha, \tag{4}$$

where $\mathbb{P}_{\pi^{n+1}}$ denotes the probability with respect to the joint distribution π^{n+1} on both future point (X, Y) and training data of size n (for training L and U).

Unlike Equation (3), Equation (4) takes into account the randomness in L and U since L and U are constructed by the (random) training data. It is easy to see that a prediction interval with prediction level $1-\alpha$ must be a weak prediction interval with prediction level $1-\alpha$ by taking probability with respect to L, U in Equation (3).

Conformal prediction is a class of generic approaches to construct distribution-free weak prediction intervals. The original conformal prediction (Vovk et al. 2005) incurs a high computational cost since it requires retraining for each new observed x. The split conformal prediction (Lei et al. 2015) improves the computational efficiency and offers an assumption-free guarantee but comes at the cost of higher variance. More recently, split conformal quantile regression combines the quantile regression and split conformal prediction to output intervals that are adaptive to heteroscedasticity whereas the standard split conformal prediction cannot (Romano et al. 2019). Some of recent conformal prediction approaches do not have finite-sample coverage guarantee while they are more efficient and can generate narrower intervals. These methods include the Jackknife conformal prediction (Lei et al. 2018), K-fold conformal prediction (Vovk 2015), Jackknife+ conformal prediction and CV+ conformal prediction (Barber et al. 2019).

For the rest of the section, we briefly review two widely-used conformal prediction approaches: split conformal prediction and split conformal quantile regression. To begin with, suppose we have observed i.i.d. training data $\{(x_i.y_i) : i = 1,...,n\}$ and we aim to construct a weak prediction interval at any point $\tilde{x}$ with prediction level $1-\alpha$. First, we randomly split $\{1,...,n\}$ into two disjoint sets $\mathscr{I}_1$ and $\mathscr{I}_2$.

In split conformal prediction, given any regression algorithm $\mathscr{A}$, a regression model is fit to $\{(x_i.y_i) : i \in \mathscr{I}_1\}$:

$$\mathscr{A}(\{(x_i.y_i) : i \in \mathscr{I}_1\}) \to \hat{\mu}(x).$$

Define $R_i = |y_i - \hat{\mu}(x_i)|$, $i \in \mathscr{I}_2$, the residuals on $\mathscr{I}_2$. Then compute $Q_{1-\alpha}(R,\mathscr{I}_2) := (1-\alpha)(1+1/|\mathscr{I}_2|)$-th empirical quantile of the set $\{R_i : i \in \mathscr{I}_2\}$. Finally, the prediction interval at a new point $\tilde{x}$ is given by

$$[\hat{\mu}(\tilde{x}) - Q_{1-\alpha}(R,\mathscr{I}_2), \hat{\mu}(\tilde{x}) + Q_{1-\alpha}(R,\mathscr{I}_2)].$$

Lei et al. (2018) have shown that this interval satisfies Equation (4).

In split conformal quantile regression, given any quantile regression algorithm $\mathscr{A}'$, a quantile regression model is fit to $\{(x_i.y_i) : i \in \mathscr{I}_1\}$:

$$\mathscr{A}'(\{(x_i.y_i) : i \in \mathscr{I}_1\}) \to (\hat{q}_{\alpha/2}(x), \hat{q}_{1-\alpha/2}(x)).$$

Define $E_i = \max\{\hat{q}_{\alpha/2}(x_i) - y_i, y_i - \hat{q}_{1-\alpha/2}(x_i)\}$, $i \in \mathscr{I}_2$, the residuals on $\mathscr{I}_2$. Then compute $Q'_{1-\alpha}(E,\mathscr{I}_2) := (1-\alpha)(1+1/|\mathscr{I}_2|)$-th empirical quantile of the set $\{E_i : i \in \mathscr{I}_2\}$. Finally, the prediction interval at a new point $\tilde{x}$ is given by

$$[\hat{q}_{\alpha/2}(\tilde{x}) - Q'_{1-\alpha}(E,\mathscr{I}_2), \hat{q}_{1-\alpha/2}(\tilde{x}) + Q'_{1-\alpha}(E,\mathscr{I}_2)].$$

Romano et al. (2019) have shown that this interval satisfies Equation (4).

Despite its generality, conformal prediction mainly focuses on weak prediction intervals (4). We will propose a deep-learning-based method which can provide a stronger guarantee (3). Moreover, our approach explicitly optimizes the interval width, and thus typically generates less conservative prediction intervals than conformal prediction.

4 DEEP LEARNING FOR PREDICTION INTERVALS

Constructing a high-quality prediction interval requires to balance a tradeoff between the expected interval width and the expected coverage rate maintenance. This viewpoint can be formalized as a constrained stochastic optimization problem, which has been used in previews work such as Khosravi et al. (2010), Pearce et al. (2018), Rosenfeld et al. (2018), Chen et al. (2021). Although it is difficult to solve this problem directly, it provides a framework for developing our following training procedure.

To be more concrete, we aim to find two functions L and U, both in a class of functions $\mathscr{H}$ given by a neural network, so that $[L(x),U(x)]$ is the "optimal-quality" prediction interval with prediction level $1-\alpha$, which is formulated as the following constrained stochstic optimization problem:

$$\begin{aligned} \min_{L,U\in\mathscr{H} \text{ and } L\le U} \quad & \mathbb{E}_{\pi_X}[U(X)-L(X)] \\ \text{subject to } & \mathbb{P}_{\pi}(Y \in [L(X),U(X)]|L,U) \ge 1-\alpha \end{aligned} \tag{5}$$

where $\mathbb{E}_{\pi_X}$ denotes the expectation with respect to the marginal distribution of X. Given a set of data $\mathscr{D} = \{(x_i, y_i) : i = 1, ..., n\}$, we approximate (5) with the following empirical constrained optimization problem:

$$\begin{aligned} \widehat{\text{opt}}(t) : \min_{L,U \in \mathscr{H} \text{ and } L \le U} \quad & \mathbb{E}_{\hat{\pi}_X}[U(X) - L(X)] \\ \text{subject to } & \mathbb{P}_{\hat{\pi}}(Y \in [L(X), U(X)] | L, U) \ge 1 - \alpha + t \end{aligned} \tag{6}$$

parameterized by $t \in [0, \alpha]$, where $\mathbb{E}_{\hat{\pi}_X}$, $\mathbb{P}_{\hat{\pi}}$ are expectation and probability with respect to the empirical distribution constructed from the data $\mathscr{D}$, i.e., $\mathbb{E}_{\hat{\pi}_X}[U(X) - L(X)] = \frac{1}{n}\sum_{i=1}^n (U(x_i) - L(x_i))$, $\mathbb{P}_{\hat{\pi}}(Y \in [L(X), U(X)] | L, U) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{y_i \in [L(x_i), U(x_i)]}$. The "safety margin" t in (6) is introduced to boost the performance guarantee between (5) and (6). Note that if t is too small, say, $t = 0$, then because of finite-sample errors, the true coverage can be smaller than the empirical coverage with substantial probability (in fact, central limit theorem tells us that this happens with probability $\approx 1/2$), so that even when the empirical coverage reaches the target level $1 - \alpha$, the true coverage can be (much) lower. On the other hand, if t is too large, it will greatly reduce the feasible set of (6), leading to a deterioration in the objective interval width. Therefore the margin t needs to be properly calibrated.

In theory, the learning guarantee of feasibility and optimality between (5) and (6) has been extensively studies in Chen et al. (2021). For instance, if the class of function $\mathscr{H}$ has finite VC dimension $\text{vc}(\mathscr{H})$ such as a neural network, then one of their results reveals that, after ignoring logarithmic factors, the dataset size n needed to learn a good prediction interval with guaranteed coverage from $\widehat{\text{opt}}(t)$ is of order $\Omega(\text{vc}(\mathscr{H}))$, if t of order $O\left(\sqrt{\text{vc}(\mathscr{H})/n}\right)$ is adopted. The corresponding optimality gap in width is $O\left(\sqrt{\text{vc}(\mathscr{H})/n}\right)$.

It is almost impossible to directly solve (6) with a general function set $\mathscr{H}$. Therefore we consider an empirical approach using deep learning. First we derive a Lagrangian formulation of (6). This formulation introduces the dual multiplier λ which can be viewed as a tunable hyper-parameter to balance the tradeoff between the objective and the constraint in (6). Specifically, we consider

$$L(\lambda) = \mathbb{E}_{\hat{\pi}_X}[U(X) - L(X)] + \lambda(1 - \alpha + t - \mathbb{P}_{\hat{\pi}}(Y \in [L(X), U(X)] | L, U))$$

or

$$L(\lambda) = \frac{1}{n}\sum_{i=1}^n (U(x_i) - L(x_i)) + \frac{\lambda}{n}\sum_{i=1}^n \mathbf{1}_{y_i \notin [L(x_i), U(x_i)]} + \text{constant}$$

where λ is the multiplier. Intuitively, if λ is large, $(U(x_i) - L(x_i))$ contributes less to the overall loss function, and hence the resulting interval tends to be wide but has a high coverage rate. On the contrary, a small λ entails a narrow interval with a low coverage rate. We treat λ as a hyper-parameter in the loss function and use deep learning to approximately solve this problem. To achieve this, we assume that L and U are two output neurons given by a neural network model with network parameters θ to be optimized based on the loss function:

$$L(\theta; \lambda) = \frac{1}{n}\sum_{i=1}^n (U_\theta(x_i) - L_\theta(x_i)) + \frac{\lambda}{n}\sum_{i=1}^n I_{y_i \notin [L_\theta(x_i), U_\theta(x_i)]}$$

Then we run gradient descent on $L(\theta; \lambda)$ with respect to θ to find the optimal network parameters θ^*. Note that this L is non-smooth with respect to θ. Therefore to implement gradient descent, we use a "soft" version of the Lagrangian function. For instance, we can adopt the following form of "soft" loss:

$$l(\theta) := \frac{1}{n}\sum_{i=1}^n (U_\theta(x_i) - L_\theta(x_i))^2 + \frac{\lambda}{n}\sum_{i=1}^n (\max\{L_\theta(x_i) - y_i, 0\} + \max\{y_i - U_\theta(x_i), 0\})^2 \tag{7}$$

In practice, a mini-batch gradient descent (Li et al. 2014) is adopted to accelerate the training procedure and thus the above n is interpreted as the mini-batch size in each iteration. Finally, we note that λ is a hyper-parameter during the training procedure which is directly related to the coverage constraint (5). As in the standard learning routine, hyper-parameters are usually chosen via a validation procedure. This will be our main task in Section 5.

5 VALIDATION PROCEDURE

In this section, we aim to provide a validation algorithm to select the prediction interval model. In brief, our proposed method selects the margin t in (6) in a data-driven manner to guarantee that the selected prediction interval will attain the target prediction level with high confidence. This validation procedure is inspired from Chen et al. (2021), but with a generalization to handle possibly multiple simulation runs per design point.

In the standard learning routine, the best hyper-parameter or model is chosen via a validation procedure. To do this, we first train multiple "candidate" models, each with different hyper-parameters. Then we evaluate these trained models on a validation set to select the optimal one. It is in general a non-trivial task how to properly select the optimal prediction interval model. A naive, but natural approach is that we choose the model with the shortest average interval width, among candidate models whose empirical coverage rate on the validation set reaches the target prediction level (i.e., we use the criterion (6) with $t = 0$ to choose the best model). However, as we have discussed in Section 4, this natural approach cannot guarantee the feasibility of this model. Therefore a more elaborate validation procedure is needed.

Suppose the observed simulation data are $\{(x_i, y_{i1}, ..., y_{ir}) : i = 1, ..., n\}$ where we assume the number of simulation replications at each design point x_i is the same, r. We randomly divide the data into two disjoint subsets: training set and validation set. The training set is used to train multiple candidate prediction interval models, say $\{\mathrm{PI}_j(x) = [L_j(x), U_j(x)] : j = 1, \dots, m\}$. These models can be obtained by setting different values at λ in the loss formulation (7) in Section 4, but can also be a more general collection of models. Then we use the validation data, say $\mathscr{D}_v = \{(x'_i, y'_{i1}, ..., y'_{ir}) : i = 1, ..., n_v\}$, independent of the training, for the validation procedure. Our goal is to output K models, each with given prediction level $1-\alpha_k$ $(k = 1, ..., K)$.

```
Algorithm 1: Normalized Prediction Interval Validation
  Input: Candidate models $\{\mathrm{PI}_j = [L_j, U_j] : j = 1, \dots, m\}$, target prediction levels
    $\{1-\alpha_k \in (0,1) : k = 1, \dots, K\}$, validation data $\mathscr{D}_v = \{(x'_i, y'_{i1}, ..., y'_{ir}) : i = 1, ..., n_v\}$, and
    confidence level $1-\beta \in (0,1)$.
  Procedure:
  1. For each $\mathrm{PI}_j$, $j = 1, \dots, m$, compute its empirical coverage rate on $\mathscr{D}_v$,
     $\hat{\mathrm{CR}}(\mathrm{PI}_j) := \frac{1}{n_v}\sum_{i=1}^{n_v}\frac{1}{r}\sum_{l=1}^{r} I_{y'_{il}\in \mathrm{PI}_j(x'_i)}$. Compute the sample covariance matrix $\hat{\Sigma} \in \mathbb{R}^{m\times m}$ with
     $\hat{\Sigma}_{j_1,j_2} = \frac{1}{n_v}\sum_{i=1}^{n_v}\left(\frac{1}{r}\sum_{l=1}^{r} I_{y'_{il}\in \mathrm{PI}_{j_1}(x'_i)} - \hat{\mathrm{CR}}(\mathrm{PI}_{j_1})\right)\left(\frac{1}{r}\sum_{l=1}^{r} I_{y'_{il}\in \mathrm{PI}_{j_2}(x'_i)} - \hat{\mathrm{CR}}(\mathrm{PI}_{j_2})\right)$.
  2. Let $\hat{\sigma}_j^2 = \hat{\Sigma}_{j,j}$, and compute $q_{1-\beta}$, the $(1-\beta)$-quantile of $\max_{1\le j\le m}\{Z_j/\hat{\sigma}_j : \hat{\sigma}_j > 0\}$ where
     $(Z_1, \dots, Z_m)$ is a multivariate Gaussian with mean zero and covariance $\hat{\Sigma}$.
  3. For each target level $1-\alpha_k$, $k = 1, \dots, K$, compute
       $j^*_{1-\alpha_k} = \arg\min_{1\le j\le m}\left\{\frac{1}{n_v}\left(\sum_{i=1}^{n_v}|\mathrm{PI}_j(x'_i)|\right) : \hat{\mathrm{CR}}(\mathrm{PI}_j) \ge 1-\alpha_k + \frac{q_{1-\beta}\hat{\sigma}_j}{\sqrt{n_v}}\right\}$
     where $|\mathrm{PI}_j(\cdot)| := U_j(\cdot) - L_j(\cdot)$ is the width.
  Output: $\mathrm{PI}_{j^*_{1-\alpha_k}}$ for $k = 1, \dots, K$.
```

Algorithm 1 describes our proposed validation procedure. In this algorithm, we check the feasibility of each candidate model on the validation set, using the criterion $\hat{\mathrm{CR}}(\mathrm{PI}_j) := \frac{1}{n_v}\sum_{i=1}^{n_v}\frac{1}{r}\sum_{j=1}^{r} I_{y'_{ij}\in \mathrm{PI}_j(x'_i)} \ge 1-\alpha_k+\varepsilon_j$ for some selected margins ε_j. Then we choose the one among them with the smallest average interval width. The choice of ε_j is based on the quantile of the supremum of a multivariate Gaussian distribution. We give some intuitive explanations of this algorithm here. Let $\mathrm{CR}(\mathrm{PI}_j) := \mathbb{P}_\pi(Y \in \mathrm{PI}_j(X)|\mathrm{PI}_j)$

```
Algorithm 2: Unnormalized Prediction Interval Validation
  Input: Same as in Algorithm 1.
  Procedure:
  1. Same as in Algorithm 1.
  2. Compute $q'_{1-\beta}$, the $(1-\beta)$-quantile of $\max\{Z_j : 1 \le j \le m\}$ where $(Z_1,\dots,Z_m)$ is a
     multivariate Gaussian with mean zero and covariance $\hat{\Sigma}$.
  3. For each target level $1-\alpha_k$, $k=1,\dots,K$, compute
       $$j^*_{1-\alpha_k} = \arg\min_{1\le j\le m} \left\{ \frac{1}{n_v}\Big(\sum_{i=1}^{n_v} |\mathrm{PI}_j(x'_i)|\Big) : \hat{\mathrm{CR}}(\mathrm{PI}_j) \ge 1-\alpha_k + \frac{q'_{1-\beta}}{\sqrt{n_v}} \right\}$$
     where $|\mathrm{PI}_j(\cdot)| := U_j(\cdot) - L_j(\cdot)$ is the width.
  Output: $\mathrm{PI}_{j^*_{1-\alpha_k}}$ for $k=1,\dots,K$.
```

denote the expected coverage rate of PI_j. Then the multivariate central limit theorem implies that

$$\sqrt{n_v}\big(\hat{\mathrm{CR}}(\mathrm{PI}_1) - \mathrm{CR}(\mathrm{PI}_1), \dots, \hat{\mathrm{CR}}(\mathrm{PI}_m) - \mathrm{CR}(\mathrm{PI}_m)\big) \xrightarrow{d} N(0,\Sigma)$$

where Σ is the covariance matrix. Approximating Σ with the sample covariance $\hat{\Sigma}$ from Step 1 of Algorithm 1 and applying the continuous mapping theorem, we have

$$\sqrt{n_v}\max_j(\hat{\mathrm{CR}}(\mathrm{PI}_j) - \mathrm{CR}(\mathrm{PI}_j))/\hat{\sigma}_j \xrightarrow{d} \max_j Z_j/\hat{\sigma}_j$$

where $(Z_j)_{j=1,\dots,m}$ follows $N(0,\hat{\Sigma})$. By using the $1-\beta$ quantile of $\max_j Z_j/\hat{\sigma}_j$ in the margins, we should expect that $\mathrm{CR}(\mathrm{PI}_j) \ge \hat{\mathrm{CR}}(\mathrm{PI}_j) - q_{1-\beta}\hat{\sigma}_j/\sqrt{n_v}$ for all $j=1,\dots,m$ uniformly with probability $\approx 1-\beta$.

In fact, we have the following theorem to precisely describe the error bound based on recent high-dimensional Berry-Esseen theorems (Chernozhukov et al. 2017). This theorem is a generalization of the result in Chen et al. (2021) to handle multiple simulation runs r per design point.

Theorem 2 Let $1-\underline{\alpha} := \max_{j=1,\dots,m}\mathrm{CR}(\mathrm{PI}_j)$, $1-\alpha_{\min} := 1-\min_{k=1,\dots,K}\alpha_k$, and $\tilde{\alpha} := \min\{\alpha_{\min}, 1-\max_{k=1,\dots,K}\alpha_k\}$. For every collection of interval models $\{\mathrm{PI}_j : 1\le j\le m\}$, every n_v, and $\beta\in(0,\frac{1}{2})$, the prediction intervals output by Algorithm 1 satisfy

$$\begin{aligned}&\mathbb{P}_{\mathscr{D}_v}(\mathrm{CR}(\mathrm{PI}_{j^*_{1-\alpha_k}}) \ge 1-\alpha_k \text{ for all } k=1,\dots,K)\\ \ge\ & 1-\beta - C_1\bigg(\Big(\frac{r\log^7(mn_v)}{n_v\tilde{\alpha}}\Big)^{\frac{1}{6}} + \exp\big(-C_2 n_v \min\big\{\varepsilon, \frac{r\varepsilon^2}{\underline{\alpha}(1-\underline{\alpha})}\big\}\big)\bigg)\end{aligned}$$

with $\varepsilon = \max\big\{\alpha_{\min} - \underline{\alpha} - C_1\big((\frac{\underline{\alpha}(1-\underline{\alpha})}{rn_v} + \frac{\log(n_v\alpha_{\min})}{n_v^2})\log(m/\beta)\big)^{1/2}, 0\big\}$, where $\mathrm{CR}(\mathrm{PI}_j) := \mathbb{P}_\pi(Y\in\mathrm{PI}_j(X)|\mathrm{PI}_j)$, $\mathbb{P}_{\mathscr{D}_v}$ denotes the probability with respect to the validation data $\mathscr{D}_v$, and C_1, C_2 are universal constants.

Proof. We essentially follow the proof of Theorem 5.1 in Chen et al. (2021) with some modifications. To see this, we consider to apply Berry-Esseen theorems (Lemma J.11 in Chen et al. (2021)) to the random vectors $W_i := \Big(\frac{1}{r}\sum_{l=1}^r I_{y'_{il}\in\mathrm{PI}_1(x'_i)}, \dots, \frac{1}{r}\sum_{l=1}^r I_{y'_{il}\in\mathrm{PI}_m(x'_i)}\Big)$ where $i=1,\dots,n_v$. It is easy to see that W_i are i.i.d. for $i=1,\dots,n_v$ and

$$W_i^{(t)} = \frac{1}{r}\sum_{l=1}^r I_{y'_{il}\in\mathrm{PI}_t(x'_i)} \in [0,1], \quad \mathrm{Var}[W_i^{(t)}] = \frac{1}{r}\mathrm{Var}[I_{y'_{i1}\in\mathrm{PI}_t(x'_i)}] = \frac{1}{r}\Big(\mathrm{CR}(\mathrm{PI}_t)\big(1-\mathrm{CR}(\mathrm{PI}_t)\big)\Big).$$

This implies that Assumption 3 and 4 in Chen et al. (2021) for Berry-Esseen theorems hold with $D = \frac{C\sqrt{r}}{\sqrt{\tilde{\alpha}}}$. The rest of the proof is similar to theirs. □

Theorem 2 shows that the prediction interval output by our validation procedure, Algorithm 1, has finite-sample guarantee on the coverage rate attainment. In general any prediction intervals that satisfy the constraint in Step 3 have such a feasibility guarantee. Meanwhile, in order to have better performance on the objective, we select the one among them with the smallest average interval width on the validation set. In addition to Theorem 2, our validation procedure also possesses guaranteed performance regarding the optimality of the expected interval width among candidate models. This guarantee can be found in Chen et al. (2021). A similar validation strategy, viewed as an "unnormalized" (as opposed to "normalized") version of Algorithm 1 when we handle the variance term $\hat{\sigma}_j$, is described in Algorithm 2. Its performance guarantee can be established similarly as Theorem 2.

6 EXPERIMENTS

We conduct numerical experiments with focus on the metamodeling, i.e., we use the given simulation data to build a prediction interval for future simulation outputs. We do not focus on the experiment design such as how to optimally choose the design points and simulation replications. Instead, we assume these data are supplied in a single batch in a random or arbitrary manner.

Consider the same example as in Ankenman et al. (2010). Let $Y(x)$ be the steady-state number of customers in an M/M/1 queue with service rate 1 and arrival rate x. We aim to build prediction intervals with 95% prediction level for the simulation output $Y(x)$ over the domain $0.2 < x < 1$. We consider two experiment designs: (1) Sparse and deep design: Training data consist of 7 design points, $x = 0.3, 0.4, 0.5, ..., 0.9$, making 50 simulation replications at each of them. (2) Dense and shallow design: Training data consist of 40 design points, $x = 0.22, 0.24, 0.26, ..., 0.98$, making 5 simulation replications at each of them. In both cases, we implement different methods (described below) on these training data to construct prediction intervals and then evaluate the performance of each method on new test data, which are obtained at 15 input points i.i.d. drawn from Uniform$(0.2, 1)$, making 100 simulation replications at each of them. A high-quality prediction interval should cover at least 95% test data while maintaining a small interval width.

For SK, we use Lemma 1 to generate prediction intervals under the posterior predictive distribution. Our estimation of the GP structure follows the instruction in Section 2.2 and Section 4 in Ankenman et al. (2010). For split conformal prediction (SCP), our implementation is based on the description in Section 3 where the base regression algorithm is a neural network consisting of 1 hidden layer with 20 neurons using mean square loss. For quantile regression forests (QRF), this is the well-known algorithm proposed in Meinshausen (2006). For split conformalized quantile regression (SCQR), our implementation is based on the description in Section 3 where the base quantile regression algorithm is exactly the QRF. For neural-network-based prediction intervals, we adopt the loss function and method described in Section 4. The base neural network consists of 1 hidden layer with 20 neurons and the output layer with 2 neurons representing L, U. By adjusting λ in (7), prediction interval models with different coverage rates can be trained, which are then used in our validation algorithms to obtain the final model. We implement three validation strategies: Algorithm 1 (NNGN), Algorithm 2 (NNGU), and the natural validation scheme (NNVA). In NNVA, we directly compares the empirical coverage rates on the validation set to the target levels without the Gaussian margins, as we mentioned in Section 5. The validation data are split from the training data: 60% data for training networks and 40% data for validation procedure.

We conduct numerical experiments by using the above methods to construct prediction intervals with prediction level $1 - \alpha = 95\%$. Each experiment is repeated for $N = 50$ times to estimate the confidence of coverage attainment. The performance of each method is evaluated based on two metrics: exceedance probability (EP) and interval width (IW). EP captures the success ratio in achieving the target prediction level among $N = 50$ experimental repetitions, while IW indicates the average interval width among $N = 50$

experimental repetitions. Formally:

$$EP = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{CR_i \geq 1-\alpha\}, \quad IW = \frac{1}{Nn_{te}}\sum_{i=1}^{N}\sum_{j=1}^{n_{te}}(U_i(x_j) - L_i(x_j))$$

where $n_{te} = 15$ is the size of test points, CR_i is the empirical coverage rate on test data of the i-th prediction interval from the i-th repetition and the target prediction level $1-\alpha = 95\%$. Throughout our experiments, the confidence level $1-\beta$ is set to 0.9 in the validation procedure. The best result is achieved by the model with the smallest IW value among those with $EP \geq 0.9$. If no model achieves $EP \geq 0.9$, then the one with highest EP is the best.

Experimental results are shown in Table 1 which reports the values of EP and IW for each method under 2 experimental designs. It is shown that prediction intervals given by SK tend to be very conservative in terms of interval width although they usually achieve the desired prediction level. Moreover, in the dense and shallow design (Design 2), we observe that the computational issue about the near-singularity of the matrix sometimes arises, making SK not applicable. If this happens, we restart the SK experiment. Among the methods with high EP ($EP \geq 0.9$), the IW values of our NNGN are the smallest in both cases. In addition, our NNGN and NNGU generate similar competitive results. In contrast, the natural validation scheme NNVA performs badly on test data in terms of achieving the desired prediction level ($EP \leq 0.7$) but our NNGN and NNGU enjoy high confidence for that, which demonstrates the effectiveness of our validation algorithms. This observation coincides with our discussion in Section 4: The margin parameter $t > 0$ in (6) needs to be properly calibrated. Our method is also computationally cheap: Its running time is less than SK in the dense and shallow design where the matrix inversion in SK is slow and moreover, that matrix is sometimes near-singular. Therefore, we can expect that in the high-dimensional setting where a lot of design points should be placed, the computational cost of our method will be much less than SK.

Table 1: Prediction intervals with 95% prediction level. The best results are in **bold**.

	SK		SCP		QRF		SCQR		NNVA		Ours-NNGN		Ours-NNGU	
Data	EP	IW	EP	IW	EP	IW	EP	IW	EP	IW	EP	IW	EP	IW
Experiment Design 1	0.96	9.97	0.60	5.85	0.92	4.51	0.94	4.72	0.48	3.82	**0.90**	**4.47**	0.98	4.66
Experiment Design 2	0.98	10.37	0.80	7.52	0.98	4.97	0.96	4.91	0.70	4.07	**0.90**	**4.26**	0.86	4.22

ACKNOWLEDGMENTS

We gratefully acknowledge support from the National Science Foundation under grants CAREER CMMI-1834710 and IIS-1849280 and the JP Morgan Chase Faculty Research Award.

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2008. “Stochastic kriging for simulation metamodeling”. In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 362–370. Miami, Florida: Institute of Electrical and Electronics Engineers, Inc.

Ankenman, B., B. L. Nelson, and J. Staum. 2010. “Stochastic Kriging for Simulation Metamodeling”. *Operations Research* 58(2):371–382.

Barber, R. F., E. J. Candes, A. Ramdas, and R. J. Tibshirani. 2019. “Predictive inference with the jackknife+”. *arXiv preprint arXiv:1905.02928*.

Barton, R. R., and M. Meckesheimer. 2006. “Metamodel-based simulation optimization”. *Handbooks in operations research and management science* 13:535–574.

Bekki, J. M., X. Chen, and D. Batur. 2014. “Steady-state quantile parameter estimation: an empirical comparison of stochastic kriging and quantile regression”. In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 3880–3891. Savannah, Georgia: Institute of Electrical and Electronics Engineers, Inc.

Chen, H., Z. Huang, H. Lam, H. Qian, and H. Zhang. 2021. “Learning Prediction Intervals for Regression: Generalization and Calibration”. In *International Conference on Artificial Intelligence and Statistics*. April 13th-15th, held virtually, 820–828.

Chen, X., and K.-K. Kim. 2013. “Building metamodels for quantile-based measures using sectioning”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 521–532. Washington, D.C.: Institute of Electrical and Electronics Engineers, Inc.

Chen, X., and K.-K. Kim. 2014. “Stochastic kriging with biased sample estimates”. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 24(2):1–23.

Chen, X., and K.-K. Kim. 2016. “Efficient VaR and CVaR measurement via stochastic kriging”. *INFORMS Journal on Computing* 28(4):629–644.

Chernozhukov, V., D. Chetverikov, K. Kato et al. 2017. “Central limit theorems and bootstrap in high dimensions”. *The Annals of Probability* 45(4):2309–2352.

Cressie, N. A. 1993. “Statistics for spatial data”. Technical report, John Wiley & Sons, Inc.

Khosravi, A., S. Nahavandi, D. Creighton, and A. F. Atiya. 2010. “Lower upper bound estimation method for construction of neural network-based prediction intervals”. *IEEE Transactions on Neural Networks* 22(3):337–346.

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

Kleijnen, J. P. 2009. “Kriging metamodeling in simulation: A review”. *European journal of operational research* 192(3):707–716.

Koenker, R., and K. F. Hallock. 2001. “Quantile regression”. *Journal of Economic Perspectives* 15(4):143–156.

Law, A. M., W. D. Kelton, and W. D. Kelton. 2000. *Simulation modeling and analysis*, Volume 3. McGraw-Hill New York.

Lei, J., M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. 2018. “Distribution-free predictive inference for regression”. *Journal of the American Statistical Association* 113(523):1094–1111.

Lei, J., A. Rinaldo, and L. Wasserman. 2015. “A conformal prediction approach to explore functional data”. *Annals of Mathematics and Artificial Intelligence* 74(1-2):29–43.

Li, M., T. Zhang, Y. Chen, and A. J. Smola. 2014. “Efficient mini-batch training for stochastic optimization”. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. August 24nd-27th, New York, USA, 661–670.

Meinshausen, N. 2006. “Quantile Regression Forests”. *Journal of Machine Learning Research* 7(35):983–999.

Pearce, T., M. Zaki, A. Brintrup, and A. Neely. 2018. “High-quality prediction intervals for deep learning: A distribution-free, ensembled approach”. In *International Conference on Machine Learning*. July 10th-15th, Stockholm, Sweden, 4075-4084.

Romano, Y., E. Patterson, and E. Candes. 2019. “Conformalized quantile regression”. In *Advances in Neural Information Processing Systems*. December 8th-14th, Vancouver, Canada, 3543–3553.

Rosenfeld, N., Y. Mansour, and E. Yom-Tov. 2018. “Discriminative Learning of Prediction Intervals”. In *International Conference on Artificial Intelligence and Statistics*. April 9th-11th, Lanzarote, Canary Islands, 347–355.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. “Design and analysis of computer experiments”. *Statistical Science*:409–423.

Shafer, G., and V. Vovk. 2008. “A Tutorial on Conformal Prediction.”. *Journal of Machine Learning Research* 9(3).

Staum, J. 2009. “Better simulation metamodeling: The why, what, and how of stochastic kriging”. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 119–133. Austin, Texas: Institute of Electrical and Electronics Engineers, Inc.

Stein, M. L. 1999. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.

Thiagarajan, J. J., B. Venkatesh, P. Sattigeri, and P.-T. Bremer. 2020. “Building calibrated deep models via uncertainty matching with auxiliary interval predictors”. In *Proceedings of the AAAI Conference on Artificial Intelligence*. February 2nd-9th, held virtually, 6005–6012.

Vovk, V. 2015. “Cross-conformal predictors”. *Annals of Mathematics and Artificial Intelligence* 74(1-2):9–28.

Vovk, V., A. Gammerman, and G. Shafer. 2005. *Algorithmic learning in a random world*. Springer Science & Business Media.

Zhang, H., J. Zimmerman, D. Nettleton, and D. J. Nordman. 2019. “Random forest prediction intervals”. *The American Statistician*:1–15.

AUTHOR BIOGRAPHIES

HENRY LAM is an Associate Professor in the Department of Industrial Engineering and Operations Research at Columbia University. His research focuses on Monte Carlo simulation, uncertainty quantification, risk analysis, and stochastic and robust optimization. His email address is khl2114@columbia.edu.

HAOFENG ZHANG is a PhD student in the Department of Industrial Engineering and Operations Research at Columbia University. His primary research interests lie in applying Monte Carlo simulation, uncertainty quantification and robust optimization to developing data-driven predictive methodologies. His email address is hz2553@columbia.edu.