Quantifying Uncertainty in Neural Network Ensembles using U-Statistics

Jordan Schupbach
Department of Mathematical Sciences
Montana State University
Bozeman, MT, USA
jordan.schupbach@montana.edu

John W. Sheppard
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
john.sheppard@montana.edu

Tyler Forrester
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
tyler.forrester@gmail.com

***Abstract*—Quantifying uncertainty is critically important to many applications of predictive modeling. In this paper we apply a recently developed method that uses U-statistics as a basis for estimating uncertainty in ensemble regressors to the case of neural network ensembles. U-statistics generalize the notion of a sample mean and provide distributional properties to estimates obtained by ensembles of estimators. With this method, we train neural networks on subsamples of the data and use the resulting ensemble to estimate the variance of the point estimates from the ensemble. We demonstrate that neural networks predicting a regression function exhibit the required theoretical properties for use in this ensemble method, and we then perform a coverage probability study of three simulated data sets to show that the empirical coverage probabilities match the theoretical values.**

I. Introduction

One major limitation of neural networks is the reliable and computationally efficient quantification of model uncertainty. This limitation arises from the fact the trained model does not capture probability estimates directly. By considering ensembles of models, however, we can use the behaviors of these ensembles as a basis for estimating confidence in predictions. This paper uses the U-statistic framework developed by Mentch and Hooker [1] for estimating uncertainty in ensembles, to build confidence intervals of mean point estimates obtained by ensembles of neural networks. We perform a coverage probability analysis of these methods for predicted means in linear and nonlinear regression settings as well as apply the method to a real-world data set. Our contributions are the demonstration that neural networks fit within this U-Statistics framework for estimating uncertainty as well as the empirical evaluation of these estimates of uncertainty through coverage analysis in linear and nonlinear regression settings.

A. Background

Before presenting our experiments, we begin by introducing some preliminary material. We first give some basic background knowledge on feedforward neural networks. We then introduce U-statistics and the related infinite order U-statistics that can be used to estimate model uncertainty. We show that neural networks can be used as an estimator in a U-statistic framework. Finally, we describe others' efforts in quantification of uncertainty in neural networks and neural network ensembles.

1) Feedforward Neural Networks: In this paper, we study ensembles of feedforward neural networks [2] as estimators in regression problems. For our study, we divide our neural networks into three layers: an input layer, a hidden layer, and an output layer. The activation function in the hidden nodes is represented by

$$f(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b)$$

where σ is a sigmoid function (e.g., logistic or hyperbolic tangent), $\mathbf{w}$ is a vector of weights, $\mathbf{x}$ is vector of inputs, and b is the bias or threshold. In the training of the neural networks, the loss function is squared error for regression and logistic loss (i.e., cross entropy) for classification. For the gradient descent algorithm, we use the L-BFGS update [3], a limited memory extension of the Broyden-Fletcher-Goldfarb-Shanno algorithm, which has a positive definite secant update [4]. The L-BFGS method has been shown to converge faster and perform better on small datasets [5].

2) Neural Network Ensembles: Neural network ensembles[1] have a long history of being used to improve predictive power. It is standard to train many networks with different hyperparameter combinations, while only using the network that performs the best on the validation set. Researchers have looked for ways to use the discarded neural networks to improve performance, thus creating such an ensemble [6].

Several good surveys on ensembles have been published, most recently by Li [7] with earlier work done by Wozniak *et al.* [8]. An introductory textbook into ensemble methods has been written by Rokach [9].

There are many earlier references to ensemble learning in the machine learning literature, such as [10], [11]. Salamon and Hansen were the first to propose using neural network ensembles [12], where their paper proved, with some restrictions[2], that if each network performed slightly better than random chance on a prediction, a large enough ensemble would guarantee a correct prediction, giving a solid theoretical foundation for ensemble use. This work has been extended in regression to show that the estimated mean of a neural network ensemble is at least as good as any member's estimate [13].

[1]Neural network ensembles are sometimes also called committees.

[2]The primary restriction was the assumption that each member's prediction was independent of the other members' prediction in the ensemble.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

The two most common approaches to training ensembles (in general) are bagging [14] and boosting [15]. Bagging is the process of training members of a neural network ensemble on bootstrapped samples and then combining those members using an aggregated average in regression or plurality voting in classification. Bootstrapping is a sampling method where N samples are chosen randomly with replacement from a training set with N observations. This method is robust to model misspecification and overfitting [16].

Another popular resampling technique, boosting, focuses training on the examples on which the algorithm performs the worst [15]. AdaBoost.M1, the most popular boosting algorithm, iteratively fits classifiers on weighted versions of the dataset [17]. Many extensions of AdaBoost exist. For example, Peerlinck et. al. recently combined ideas from Adaboost.R2, Adaboost.RΔ and Adaboost.RT to create an "approximate AdaBoost" (or AdaBoost.App) [18].

The resampling technique used in this paper is **sub**sample **agg**regat**ing** (subbagging). The technique is best described through U-statistics and the related resampling techniques discussed in the next section.

3) U-Statistics: U-statistics (where the "U" refers to being unbiased) are an important, broad class of minimum variance unbiased estimators. They were considered early on by Kendall [19] and Wilcoxon [20] in the estimation of rank correlation. They were then formalized, being found to be of minimum variance by Halmos [21] and asymptotically normal by Hoeffding [22]. A good introduction to the subject can be found in Lehmann [23] and more thorough treatment of the subject can be found in Lee [24]. More relevant to the purpose of this paper is that predictions from ensemble methods can be shown to be U-statistics given some regularity conditions. Here, we give an introduction to U-statistics in this context, originally described by Mentch and Hooker [1].

Consider an i.i.d random sample $X_1, \ldots, X_n$ from some population with cdf F. Here, we assume that F is unknown rather than belonging to some parametric class. Next, consider the expectation functional $\theta(F) = E[\phi(X_1, \ldots, X_a)]$, where kernel ϕ is assumed to be permutation symmetric. Then $\phi(X_1, \ldots, X_a)$ is an unbiased estimator for θ with sample of size n, as is $\phi(X_{i_1}, \ldots, X_{i_n})$ for any a-tuple, where $1 \leqslant i_1 \leqslant \cdots \leqslant i_n \leqslant n$. The uniform minimum variance unbiased estimator (UMVUE) of θ is given by

$$U = \frac{1}{\binom{n}{a}} \sum_{i_1} \cdots \sum_{i_a} \phi(X_{i_1}, \ldots, X_{i_a}).$$

Since ϕ is symmetric, so is U. Note that we can view U-statistics as a generalization of the sample mean, where we average the kernel ϕ over all $\binom{n}{a}$ subsamples of size a. Then U converges asymptotically to a normal distribution with variance $\frac{k^2}{n}\zeta_{1,k}$ where

$$\zeta_{1,k} = cov[\phi(X_1, \ldots, X_a); \phi(X_1, X_2', \ldots, X_a')]$$

and $X_2', \ldots, X_a' \sim F$ [22]. That is, asymptotic variance is proportional to the covariance of two subsamples having only one sample in common.

B. Related Work

1) Neural Network Ensembles: Negative correlation learning has been proposed as an ensemble neural network method [25]. In negative correlation learning, an ensemble of neural networks is trained simultaneously with a loss function that contains a penalty for the correlation between the networks. The loss function [26] is

$$\mathcal{L}(F_i) = \frac{1}{N}\left(\sum_{n=1}^{N}\left(F_i(n) - d(n)\right)^2 + \lambda \sum_{n=1}^{N} p_i(n)\right)$$

$$p_i(n) = \left(F_i(n) - F(n)\right) \sum_{i \neq j}^{N} \left(F_j(n) - F(n)\right)$$

where $F_i(n)$ is the output of network i on the nth training pattern, $F(n)$ is the average output of the ensemble on the nth training pattern, $d(n)$ is the target value, and $0 < \lambda < 1$.

The mean squared error (MSE) of an ensemble can be decomposed into variance, covariance, and bias components. The larger λ is in the loss function, the larger the decrease in the covariance component of the MSE [27]. This type of loss function causes individual network members to decompose tasks into subtasks [26].

More recently, Pearce et. al [28] have used ensembles of Bayesian neural networks for estimating uncertainty by showing an extension of the usual ensemble approach results in approximate Bayesian inference.

2) Uncertainty Estimation for Neural Networks: Geman *et al.* [29] showed consistency in neural networks by increasing the number of hidden layers asymptotically and demonstrating that, as the number of nodes increase, the amount of prediction bias decreases and prediction variance increases, thus matching the bias-variance trade off.

Tibshirani [16] compared (among other methods) estimated uncertainty via maximum likelihood using the delta method and assuming that the errors of a neural network are Gaussian. Estimation involves calculating a Hessian matrix. Since calculating the Hessian in large networks is impractical, this methodology has limitations. Tibshirani also compared the sandwich estimator to estimate parameter variance in neural networks [16]. This method produces an asymptotically consistent covariance matrix without distributional assumptions or even an assurance that the correct model generated the parameter as long as that parameter is consistent. It is also robust to heteroscedasticity [30]. These relaxed conditions make it appealing for neural networks, though some researchers have raised concerns about its accuracy in practice [30].

Bootstrapping has also been proposed to quantify uncertainty. This approach to quantification uses the bootstrap resampling process to generate error estimates. Tibshirani found that this method produced the most accurate estimates of prediction standard errors [16] among the methods he compared.

More recently, a method called MC (Monte Carlo) dropout has been used to quantify uncertainty in neural networks [31]. The method averages outputs over ensembles formed from

Table I: Notation Used

Term	Definition
n	Size of random sample
k_n	Size of subsample
m_n	Ensemble size
$\tilde{\mathbf{z}}^{(i)}$	Fixed sample
n_{MC}	Ensemble size trained with $\tilde{\mathbf{z}}^{(i)}$
$n_{\tilde{z}}$	Number of fixed samples

subsamples of nodes of one neural network using dropout to select the active nodes in the network, turning a single network into an ensemble of networks. Another recent method uses samples from the training set, augments those samples with synthetic adversarial observations, and then assumes the average of the ensembles trained on the resulting sample follows the Gaussian distribution. [32].

Variance in neural networks can be divided into two categories, the accumulation of small random noise from unknown features in the data and the neural network's optimum approximation. It has been shown that the variance in neural networks can be reduced by training multiple neural networks while varying the initial conditions of the network, fixing both training set and architecture, and then averaging their results [33]. As previously stated, Gaussian confidence intervals have been constructed for neural networks [16]. However, these intervals typically overstate the true variability of neural network ensembles because the ensemble methods have a dampening effect on variance [34]. Bagging has also been used to estimate both confidence intervals and prediction intervals in neural network ensembles. Some research supports that these methods produce better coverage probabilities than the previously proposed Gaussian confidence intervals [34].

Bayesian Neural Network models allow for a natural estimation of uncertainty. However, MCMC algorithms can be computationally demanding for fitting such models. An equivalence of Gaussian process models to network models in the limit of infinite width has allowed the construction of a kernel for Bayesian neural network models, allowing for the quantification of uncertainty [35]. More recently, this result has been extended to deep neural networks [36] and deep convolutional and residual neural networks [37].

Recently, many approaches to estimating uncertainty have focused on prediction intervals (*e.g.* [38] and [39]). Although our approach currently does not allow for estimating prediction intervals, they do have the advantage of being able to construct hypothesis tests. Although not explored in this paper, Mentch and Hooker showed that differences in model estimates are also a U-statistic and hence allow for the hypothesis testing of covariates.

II. Neural Network Ensemble-based U-Statistics

In the following, we describe the theoretical motivation for our work. We follow this with an explanation of the algorithms implemented in our experiments. For this discussion, the main pieces of notation are explained in Table I.

A. Theoretical Motivation

Consider a random sample $(\boldsymbol{X}, \boldsymbol{Y}) \overset{\text{iid}}{\sim} F$ of size n. Suppose we build a neural network N from a subbagged sample of size a taken from our dataset. Suppose further that we do this for all $\binom{n}{a}$ subsamples. We can then take the average of the predictions for some $\boldsymbol{x}^*$ from these neural networks as the estimate of our predicted value. Let us write this average as

$$b(\boldsymbol{x}^*) = \frac{1}{\binom{n}{a}} \sum_{(i)} N_{\boldsymbol{x}^*}((X_{i_1}, Y_{i_1}), \ldots, (X_{i_a}, Y_{i_a})).$$

Given some regularity conditions—unbiased and permutation symmetric—we have a procedure that results in a U-statistic for these predicted values [1]. N has been shown to be an unbiased estimator [6] that is asymptotically consistent [40] and if trained by batch update is permutation symmetric. N can be used as estimator that results in a U-statistic for the predicted values.

Unfortunately, it is generally computationally infeasible to build neural networks for all $\binom{n}{a}$ subsamples of the data. It has been shown that taking $m \leqslant \binom{n}{a}$ subsamples of size a results in

$$b_m(\boldsymbol{x}^*) = \frac{1}{m} \sum_{(i)} N_{\boldsymbol{x}^*}((X_{i_1}, Y_{i_1}), \ldots, (X_{i_a}, Y_{i_a})),$$

which is an incomplete U-statistic. Even so, this has been shown to be asymptotically normal and unbiased by Janson [41], assuming the variance of the estimator converges to zero at a rate faster than $\sqrt{n}$. Neural Network N has been shown to be mean integrated squared error (MISE) consistent[3] such that $MISE = O\left(\frac{C_f^2}{s}\right) + O\left(\frac{sd}{n} \log n\right)$ where C_f is a constant related to smoothing from the training process, s is the number of nodes in the network, d is the number of covariates, and n is the sample size [40]. It may also make sense for us to scale m with n. Specifically, we could consider taking subsamples of size $m_n = \binom{n}{k_n}$ giving us

$$b_{m_n,k_n}(\boldsymbol{x}^*) = \frac{1}{m_n} \sum_{(i)} N_{\boldsymbol{x}^*}((X_{i_1}, Y_{i_1}), \ldots, (X_{i_a}, Y_{i_a})),$$

which is an infinite order U-statistic or a resampled statistic when $m_n \neq \binom{n}{k_n}$.

Frees developed necessary and sufficient conditions for asymptotic normality when m_n grows faster than n [42], and Mentch and Hooker developed conditions for individual means $E[b(\boldsymbol{x}^*)]$ for all growth rates of m_n with respect to n. So long as the estimates for a bounded regression function are bounded, the variance of the kernel function ϕ is bounded, $\lim \frac{n}{m_n} = \alpha$, $\lim \frac{k_n}{\sqrt{n}} = 0$, and $\lim \sigma_{1,k_n} \neq 0$, then the infinite order U-statistic will be asymptotically normal with the following distributions [1]:

$$\text{if } \alpha = 0, \text{ then } \frac{\sqrt{n}(U_{n,k_n,m_n} - \theta_{k_n})}{\sqrt{k_n^2 \zeta_{1,k_n}}} \xrightarrow{d} \mathcal{N}(0,1)$$

[3] $\mathrm{E} \|f_n - f\|_2^2 = \mathrm{E} \int (f_n(x) - f(x))^2 \, dx$ (variance and squared bias)

Algorithm 1 Neural Network Subbagging

Select number and size of subsamples m_n and k_n
for i in 1 to m_n **do**
    Take subsample of size k_n from training set
    Train network N_i using subsample
    N_i estimates $\mathbf{x}^*$
    Store N_i estimation
Average m_n predictions estimate $b_{n,k_n,m_n}(\boldsymbol{x}^*)$

$$\text{if } 0 < \alpha < \infty, \text{ then } \frac{\sqrt{m_n}(U_{n,k_n,m_n} - \theta_{k_n})}{\sqrt{\frac{k_n^2}{\alpha}\zeta_{1,k_n} + \zeta_{k_n,k_n}}} \xrightarrow{d} \mathcal{N}(0,1)$$

$$\text{if } \alpha = \infty, \text{ then } \frac{\sqrt{m_n}(U_{n,k_n,m_n} - \theta_{k_n})}{\sqrt{\zeta_{k_n,k_n}}} \xrightarrow{d} \mathcal{N}(0,1)$$

We show that the necessary conditions to justify using neural networks for this ensemble approach hold in Appendix A.

We will choose k_n approximately on the order of $\sqrt{n}$. This choice of k_n replaces the requirement of exponential tails on the error distribution with the requirement that $nP(|\epsilon| > \sqrt{n}) \to 0$. It also assures that the $\lim \frac{k_n}{\sqrt{n}} = 0$. Finally by choosing a small k_n, the time complexity is similar to a bootstrap method, while generating large ensembles. Note that it is not a requirement to choose k_n on the order of $\sqrt{n}$.

The subbagging method assumes that the estimator is built using the same procedure. The distributional results above do not rely on the method of building the neural network outside the weak regularity conditions; however, the subbagging procedure does require that each neural network be built using the same method. This would preclude using dropout since each estimator would be built using randomly selected samples of nodes on each training interval. We would then need a different justification for the estimator, similar to the extension of U-statistics to random forests [1].

B. Algorithms

Mentch and Hooker propose two routes to estimating variance using infinite order U-statistics. The first method is described as an external method because the estimation of ζ_{1,k_n} and ζ_{k_n,k_n} are done in a separate step outside of the point estimation. The second method is called internal since ζ_{1,k_n} and ζ_{k_n,k_n} are both estimated in the same procedure as the point estimate. The external algorithms are described here.

The neural network subbagging method (Algorithm 1) trains a neural network on a subsample of the random sample, suggested to be on the order of $\sqrt{n}$. Once the network is trained, the point estimate is saved. Another neural network is then trained on a new subsample whose point estimates are saved, and this process is repeated until a collection of k_n point estimates is obtained. The average of the point estimates is the subbagging method's estimate.

The estimation of ζ_{1,k_n} (Algorithm 2) is done by keeping one sample constant between subsamples. The size of the subsample is again suggested to be on the order of $\sqrt{n}$. The neural network is trained on this subsample, and its prediction is stored. For n_{MC} trials, a new sample is selected with the same fixed sample point $\tilde{\mathbf{z}}^{(i)}$ in each sample, then the average of the n_{MC} predictions is recorded. Another $\tilde{\mathbf{z}}^{(i)}$ is selected and the process repeated $n_{\tilde{z}}$ times, then the variance of the collection of ensemble predictions is computed.

Algorithm 2 Neural Network ζ_{1,k_n} Estimation

for i in 1 to $n_{\tilde{z}}$ **do**
    Select initial fixed point $\tilde{\mathbf{z}}^{(i)}$
    for i in 1 to n_{MC} **do**
        Select subsample $S_{\tilde{\mathbf{z}}^{(i)},j}$ of size k_n
        Train network N_i using subsample $S_{\tilde{\mathbf{z}}^{(i)},j}$
        Store N_i prediction at $\mathbf{x}^*$
    Record average of the n_{MC} predictions
Compute variance of the $n_{\tilde{z}}$ averages

Algorithm 3 Neural Network ζ_{k_n,k_n} Estimation

for i in 1 to $n_{\tilde{z}}$ **do**
    Select subsample, S_{k_n}, of size k_n
    Train network N_i using subsample S_{k_n}
    Store N_i prediction at $\mathbf{x}^*$
Compute variance of the $n_{\tilde{z}}$ predictions

Algorithm 4 Internal Variance Estimation ζ_{1,k_n} and ζ_{k_n,k_n}

for i in 1 to $n_{\tilde{z}}$ **do**
    Select initial fixed point $\tilde{z}^{(i)}$
    for j in 1 to n_{mc} **do**
        Select subsample $S_{i,j}$ of size $(k_n - 1)$
        Append $\tilde{z}^{(i)}$ to $S_{i,j}$
        Train network N_i
        N_i estimates $N_{\boldsymbol{x}*}((X_{k_1}, Y_{k_1}), \ldots, (X_{k_n}, Y_{k_n}))$
        Store N_i estimation
    Record average prediction n_{mc} networks
Compute variance for $n_{\tilde{z}}$ averages to estimate ζ_{1,k_n}
Compute variance of all predictions to estimate ζ_{k_n,k_n}
Compute the mean of all predictions to estimate θ_{k_n}

To estimate ζ_{k_n,k_n} (Algorithm 3), we repeat the previously mentioned subbagging procedure for $n_{\tilde{z}}$ rather than m_n times, and instead of calculating the mean of the ensemble point estimate, we calculate the variance of the point estimates. We can combine the three algorithms into the internal variance estimation (Algorithm 4) [1].

Rather than estimating θ_k outside the variance estimation, it becomes the average of the fixed point ensemble used to estimate ζ_{1,k_n}. This also addresses the primary bottleneck of the external method. The internal and external estimates of ζ_{1,k_n} and ζ_{k_n,k_n} were found to be comparable by Mentch and Hooker [1].

III. Experiments

Our experiments assess the confidence interval estimates using resampled statistics of neural network point estimates

empirically through a coverage study of three simulated regression functions. The three functions are the simple linear regression (SLR) model (Equation 1), the Weibull model as a simple nonlinear regression (SNLR) model (Equation 2), and the MARS model originally described in [43] (Equation 3). The SNLR model is an example of a model for pasture regrowth given in a text by Huet *et al.* [44].

$$\begin{aligned} f(x) &= 2x_1 + \epsilon; \\ \mathcal{X} &= [0, 20] \\ \epsilon &\sim N(0, 2) \end{aligned} \tag{1}$$

$$\begin{aligned} f(x, \theta) &= \theta_1 - \theta_2 \exp(-\exp(\theta_3 + \theta_4 log\ x)) + \epsilon \\ &= 70 - 61 \exp(-\exp(-10 + 2.4 \log\ x)) + \epsilon; \\ \mathcal{X} &= [0, 80]; \\ \epsilon &\sim N(0, 2) \end{aligned} \tag{2}$$

$$\begin{aligned} f(x) &= 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.05)^2 \\ &+ 10x_4 + 5x_5 + \epsilon; \\ \mathcal{X} &= [0, 1]^5; \\ \epsilon &\sim N(0, 2) \end{aligned} \tag{3}$$

To assess the reliability of point and variance estimates of estimated means generated from the subbagging method, we generate 500 confidence intervals to estimate the coverage probabilities of the ensemble method for each simulated model across a range of points in each domain. Confidence intervals of point estimates are constructed at the $\alpha = .05$ confidence level.

We built a feed forward network with one hidden layer consisting of 500 sigmoid activation nodes, squared error loss, and using the L-BFGS optimization method implemented in the scikit-learn python toolkit [45] for each of the aforementioned models. Parameters of the neural networks were kept the same to assess the effect on performance varying the problem can have on fixed network parameters. These parameters are summarized in Table II.

With parameter values of $n_z = 50$ and $n_{mc} = 1000$, a total of 50,000 neural networks are constructed from subsample sizes of $k_n = 31$ to get predicted confidence intervals for a given model. Five hundred confidence intervals are constructed for each model to estimate empirical coverage rates. These computational efforts were performed on the Hyalite High Performance Computing System, operated and supported by University Information Technology Research Cyberinfrastructure at Montana State University.

In addition to our simulated experiments, we also highlight the usefulness of uncertainty quantification of neural network ensembles by applying the method to the real-world problem of predicting power plant output from a set of defined features. The dataset considered is the Combined Cycle Power Plant data set available from the UCI machine learning data repository [46]. This dataset consists of 47,850 observations split into 5 subsets for conducting 5×2 cross-validation (CV).

Table II: Experimental Parameters

Parameter	Parameter Value
Coverage Study Size	500
n	1000
k_n	31
m_n	51
n_{MC}	1000
$n_{\tilde{z}}$	50
LR-Method	Inverse Scaling
LR-Init	0.01
Hidden Layers	1
Number of Nodes	500
Activation	tanh
Max Iter	1×10^4
Tolerance	1×10^{-6}

The predictor, average hourly full load electrical power output (MW), is modeled as a function of ambient temperature (AT), ambient pressure (AP), relative humidity (RH) and exhaust vacuum (V). For a full description of the data see work by Tüfekci [47]. The same method for the learning the synthetic datasets was used for modeling this dataset. In the work by Tüfekci, several classifiers were compared using RMSE estimated via 5×2 CV, including a radial basis function feed forward neural network.

In choosing this dataset, we highlight the usefulness of obtaining estimates of uncertainty for neural network ensemble predictions. The goal of a power plant is to minimize costs while providing enough power so as to not cause power outages. The implication of this is that power is naturally overproduced, so obtaining good estimates for power demand and power supply is crucial for profit maximization. Thus, if a power plant can obtain good estimates of uncertainty in predicted output, they can minimize overproduction, potentially saving vast amounts of resources in the process.

IV. Results

For each of the three simulated datasets, coverage rates are obtained for point estimates across the range of each of their domains. Coverage probability plots are obtained for each and summarized in Figure 1. We see that theoretical rates of coverage are obtained in both the SLR and SNLR case. However, coverage rates are lower in the MARS case. We hypothesize that this is due to the increase in number of covariates and sample size being held constant. Indeed, the number of covariates shows up in the expression for MISE in neural networks. We also see edge effects present as indicated by the drop in coverage rates near the boundary of the domains. We provide plots for bias, standard error (SE), and t-statistics (bias divided by SE) for point-wise estimates across the sample space for each dataset. We observe that the deviations away from the theoretical coverage rates are due to the large amount of bias near the edges of the sample space. This observed edge bias is consistent with other nonparametric regression estimators (*e.g.* [48]).

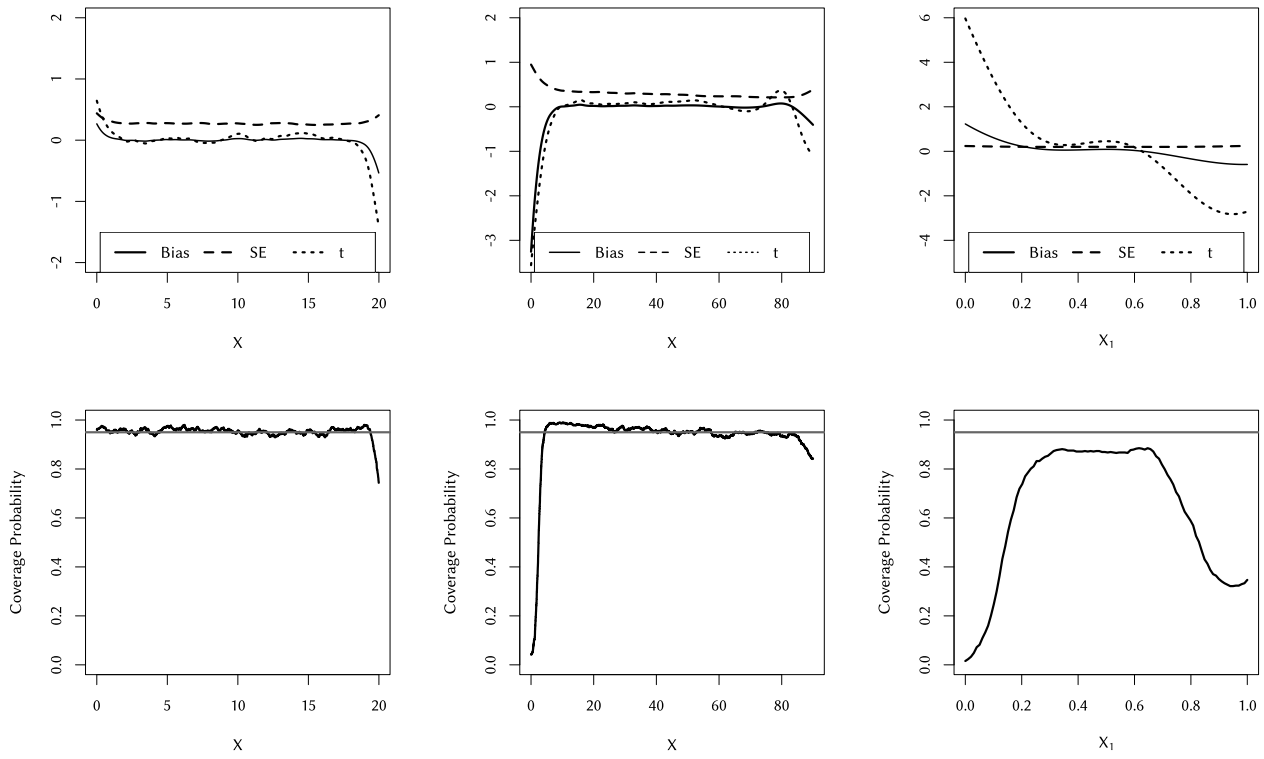


Figure 1: Bias and coverage results for the simulated data sets. Plots on the top row give bias, standard error, and t-statistics for the point estimates across the sample space and are relative to the true mean of the function. Plots on the bottom row give coverage probabilities across the sample space for each of the three simulated data sets. For the MARS data set, we give figures for X_1 holding all other variables to the middle of their domain at 0.5.

In the power plant data set, we observed an RMSE of 6.65 using the ensemble approach. Note that the goal of our approach is not to obtain the best in class results, but rather to obtain reliable estimates of model uncertainty in our predictions. With this in mind, there are some properties that would be desirable for our estimates of uncertainty. Namely, we would expect that uncertainty would be greater in regions where the density of samples is lower in the training set as well as regions of the sample space where there is larger variability in the response. Along these lines, we provide a plot of SE as a function of average 10 nearest neighbor distance (as an estimate of density) given in Figure 2a. We use the magnitude of the residuals as a proxy for variability in the response. Thus, we also plot SE as a function of absolute residuals given in Figure 2b. As expected, we can see an increasing relationship in both figures.

V. Discussion

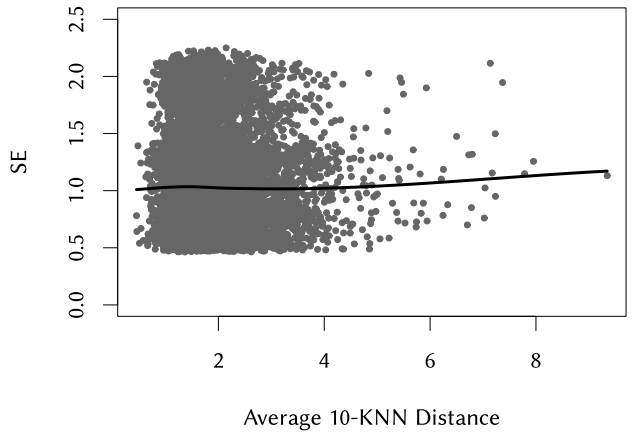
The neural network ensemble method presented in this paper has a number of practical advantages. In particular, our method may have computational advantages on datasets with a very large number of observations. It may be impractical to train a network using the full training set; however, it may be feasible to train many networks on fewer observations and then use the resulting ensemble prediction with the ancillary effect of generating accurate uncertainty estimates.

An observation that was made in the course of experimentation was that, consistently, the confidence intervals on the edges of the sample space were wider than for point estimates in the interior, but that the bias tended to be so great that coverage probabilities were far from their theoretical quantities. This is an observation generally consistent with nonparametric regressors. That is, there tend to be biases near the boundary of the sample space for these estimators.

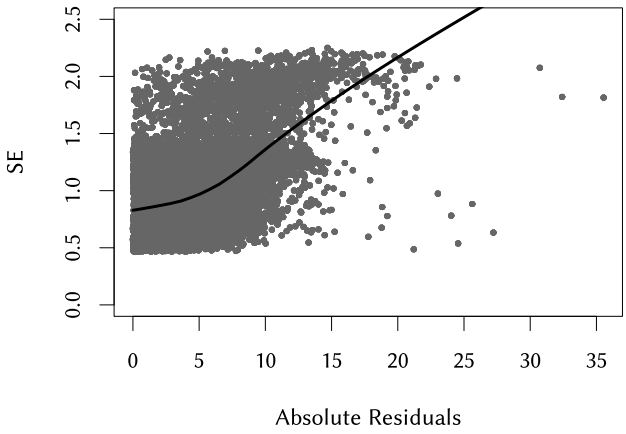
In the original paper by Mentch and Hooker, the authors noted their ensemble method can serve as a bridge between the machine learning and statistical communities by enabling estimation of uncertainty for machine learning methods and tying the theoretical justification to U-statistics. In applying their method to a real world dataset where uncertainty estimates have practical value, we hope to highlight this point.

VI. Future Work

Many exciting research questions remain open. For example, it is evident that bias corrections for the boundary are needed to have confidence in predictions near the edge of the



(a) Standard error versus 10-KNN average distance



(b) Standard error versus residuals

sample space. One possible avenue is to use a data reflection approach by first reflecting data across the boundary and then building neural networks on this augmented data. This is an approach that has been taken in the kernel estimation of probability densities (*e.g.* [49] p. 29). We think this approach could be applied to this scenario in a straightforward way in order to mitigate the effects of boundary bias.

Additionally, neural networks have high performance in the large sample case: however, performance can suffer if the model does not have enough examples to include. Because the ensemble method uses subsamples on the order of $\sqrt{n}$, these models can experience a great deal of bias in the small sample case. For this reason, it would be worth exploring this method for neural networks that tend to perform better in the small sample case.

Finally, the distributional results in Mentch and Hooker based on U-statistics are for point estimates of the mean and hence allow for the construction of confidence intervals. However, in application it is of practical importance to compute the uncertainty of a new response (*i.e.*, a prediction interval). For this reason, future work includes the extension of these asymptotic results to estimating uncertainty of a new response to enable the construction of prediction intervals.

ACKNOWLEDGMENTS

We would like to thank members of the Numerical Intelligent Systems Laboratory at Montana State University for many useful discussions in the completion of this project. We also thank members of the 2018 CSCI 547 Machine Learning class following presentation of an earlier version of this work. Finally, we would like to thank the reviewers for their thoughtful suggestions that improved the quality of this paper.

APPENDIX A

The general condition that needs to be satisfied to be an incomplete U-statistic is given in the theorem below.

Theorem. *Let* $Z_1, Z_2, \ldots \overset{iid}{\sim} F_Z$ *with* $\theta_{k_n} = \mathbb{E}_{h_{k_n}}(Z_1, \ldots, Z_{k_n})$ *and define* $h_{1,k_n}(z) = \mathbb{E}_{h_{k_n}}(z, z_2, \ldots, z_{k_n}) - \theta_{k_n}$*, then for all* $\delta > 0$

$$\lim_{n\to\infty} \frac{1}{\zeta_{1,k_n}} \int_{|h_{1,k_n}(Z_1)| \geqslant \delta\sqrt{n\zeta_{1,k_n}}} h^2_{1,k_n}(Z_1) dP = 0$$

For our purposes, we make use of a stricter but more intuitive condition, which we state here.

Theorem. *Consider a bounded regression function* F*. If there exists a constant* c *such that for all* $k_n \geqslant 1$*,*

$$|h((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{k_{n+1}}, Y_{k_{n+1}})) - h((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{k_{n+1}}, Y^*_{k_{n+1}}))| \leqslant c|Y_{k_{n+1}} - Y^*_{k_{n+1}}|$$

where $Y_{k_n+1} = F(\mathbf{X}_{k_n}) + \epsilon_{k_n+1}, Y^*_{k_n+1} = F(\mathbf{X}_{k_n}) + \epsilon^*_{k_n+1}$ *and where* ϵ_{k_n+1} *and* $\epsilon^*_{k_n+1}$ *are i.i.d. with either exponential tails or simply that* $nP(|\epsilon| > \sqrt{n}) \to 0$ *so long as* $k_n = o(\sqrt{n})$*, then the prior condition for the limiting results holds.*

This condition means that if we can bound the difference between the prediction of one neural network with another where we change a single response ($Y_{k_{n+1}}$), then the predicted values will follow an incomplete U-statistic. We now show that this condition holds for a neural network predictor N.

Proof. Let Y_1 and Y_2 be two sets of data differing by a single response. Now, denote predictions from N as $\hat{f}_1$ and $\hat{f}_2$. We can write $f = \hat{f}_1 + \epsilon_1$ and $f = \hat{f}_2 + \epsilon_2$ where we assume that for each ϵ_i we either have exponential tails or $k_n = o(\sqrt{n})$ and $nP(|\epsilon| > \sqrt{n}) \to 0$. Now, consider the previously stated

result by [40] showing neural network predictors to have Mean Integrated Square Error of $O\left(\frac{C_f^2}{s}\right) + O\left(\frac{sd}{n}\log n\right)$. Thus,

$$\begin{aligned}||\hat{f}_1 - \hat{f}_2||_2 &= ||(f-\epsilon_1) - (f-\epsilon_2)||_2 \\ &= ||\epsilon_2 - \epsilon_1||_2 \\ &\leqslant ||\epsilon_1||_2 + ||\epsilon_2||_2 \\ &= O\left(\frac{C_f^2}{n}\right) + O\left(\frac{nd}{N}\log N\right),\end{aligned}$$

resulting in $|\hat{f}_1 - \hat{f}_2| \leqslant c|Y_{k_{n+1}} - Y^*_{k_{n+1}}|$ as desired.

□

REFERENCES

[1] L. Mentch and G. Hooker, "Quantifying uncertainty in random forests via confidence intervals and hypothesis tests," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 841–881, 2016.

[2] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[3] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[4] J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996, vol. 16.

[5] Z. H. Fu, "Comparison of gradient descent, stochastic gradient descent and L-BFGS," http://www.fuzihao.org/blog/archives/page/3/, accessed: 2020-01-08.

[6] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[7] H. Li, X. Wang, and S. Ding, "Research and development of neural network ensembles: a survey," *Artificial Intelligence Review*, vol. 49, no. 4, pp. 455–479, 2018.

[8] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[9] L. Rokach, *Pattern classification using ensemble methods*. World Scientific, 2010, vol. 75.

[10] N. J. Nilsson, *Learning machines.* New York: McGraw-Hill, 1965.

[11] L. Kanal, "Patterns in pattern recognition: 1968-1974," *IEEE Transactions on information theory*, vol. 20, no. 6, pp. 697–722, 1974.

[12] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[13] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," in *How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems*, L. N. Cooper, Ed., 1995, pp. 342–358.

[14] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[15] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[16] R. Tibshirani, "A comparison of some error estimates for neural network models," *Neural Computation*, vol. 8, no. 1, pp. 152–163, 1996.

[17] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[18] A. Peerlinck, J. Sheppard, and J. Senecal, "Adaboost with neural networks for yield and protein prediction in precision agriculture," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[19] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[20] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[21] P. R. Halmos *et al.*, "The theory of unbiased estimation," *The Annals of Mathematical Statistics*, vol. 17, no. 1, pp. 34–43, 1946.

[22] W. Hoeffding, "A class of statistics with asymptotically normal distribution," *The annals of mathematical statistics*, pp. 293–325, 1948.

[23] E. L. Lehmann, *Elements of large-sample theory*. Springer Science & Business Media, 2004.

[24] A. Lee, *U-statistics: Theory and Practice*. New York: Marcel Dekker, Inc., 1990.

[25] Y. Liu and X. Yao, "Negatively correlated neural networks can produce best ensembles," *Australian journal of intelligent information processing systems*, vol. 4, no. 3/4, pp. 176–185, 1997.

[26] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 380–387, 2000.

[27] G. Brown and J. Wyatt, "Negative correlation learning and the ambiguity family of ensemble methods," in *International Workshop on Multiple Classifier Systems*. Springer, 2003, pp. 266–275.

[28] T. Pearce, M. Zaki, A. Brintrup, N. Anastassacos, and A. Neely, "Uncertainty in neural networks: Bayesian ensembling," *arXiv preprint arXiv:1810.05546*, 2018.

[29] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[30] G. Kauermann and R. J. Carroll, "A note on the efficiency of sandwich covariance matrix estimation," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1387–1396, 2001.

[31] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016, pp. 1050–1059.

[32] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[33] U. Naftaly, N. Intrator, and D. Horn, "Optimal ensemble averaging of neural networks," *Network: Computation in Neural Systems*, vol. 8, no. 3, pp. 283–296, 1997.

[34] J. G. Carney, P. Cunningham, and U. Bhagwan, "Confidence and prediction intervals for neural network ensembles," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 1999, pp. 1215–1218.

[35] R. M. Neal, "Priors for infinite networks," in *Bayesian Learning for Neural Networks*. Springer, 1996, pp. 29–53.

[36] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," *arXiv preprint arXiv:1711.00165*, 2017.

[37] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep convolutional networks as shallow gaussian processes," *arXiv preprint arXiv:1808.05587*, 2018.

[38] J. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 748–757, 1997.

[39] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE access*, vol. 6, pp. 36 218–36 234, 2018.

[40] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Machine learning*, vol. 14, no. 1, pp. 115–133, 1994.

[41] S. Janson, "The asymptotic distributions of incomplete u-statistics," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 66, no. 4, pp. 495–505, 1984.

[42] E. W. Frees, "Infinite order u-statistics," *Scandinavian Journal of Statistics*, pp. 29–45, 1989.

[43] J. H. Friedman *et al.*, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[44] S. Huet, A. Bouvier, M.-A. Poursat, and E. Jolivet, *Statistical tools for nonlinear regression: a practical guide with S-PLUS and R examples*. Springer Science & Business Media, 2006.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[46] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[47] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, 2014.

[48] R. Eubank and P. Speckman, "A bias reduction theorem with applications in nonparametric regression," *Scandinavian Journal of Statistics*, pp. 211–222, 1991.

[49] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.