Research Project

Comparison of Conformal Prediction and Bootstrapping on a Regression Problem with Random Forests

Main Sources:

[1]: Johansson2014\_Article\_RegressionConformalPredictionW\_edit

[2]: Evaluation of a Variance-Based Nonconformity

[3]: manuscript\_RFIntervals\_FinalVersion

Abstract

Introduction

Related Work

Methods

Random Forest

Conformal Prediction

Bootstrap

Empirical Evaluation

Experimental setup

Data Simulation

Data Splits

Computer Specifications

Chosen Parameters

Results

Conclusion

Future Improvements

Experimental Setup

Data Simulation

Empirical experiments are often conducted on real world datasets, but because this research focuses on the parametric version of bootstrapping, we opted for an artificially simulated dataset instead. This means that we can be certain about the distribution of the response variable, which is necessary for parametric bootstrap.

The implementation was based on [3] that uses an additive error model: Y = m(x) + e, where the predictor x is 50-dimensional simulated from a multivariate normal distribution with covariance matrix Sigma , where Sigma p is an AR(1) covariance matrix with rho = 0.6 and diagonal values equal to 1. The error term e is Gaussian noise and the m(x) mean function is nonlinear with interaction between the predictors to make the data more complex. 100000 instances were simulated from this model.

Data Splits

The experiments use the holdout method for evaluation with 70%-30% split between training and test sets. The first 70000 instances were used as the pool for training and the last 30000 for testing. For each different data size, the appropriate number of observations were sampled from these pools randomly to ensure that no test instance has been used for training or for validation while optimizing the parameters. It is important to note that while bootstrapping uses all training examples for training the random forest, conformal prediction utilizes only a part of it for proper training and the rest for calibration. A possible approach would have been to make the number of instances used for training equal for both methods, but since calibration is an important step in obtaining prediction intervals, I made the decision to use a subset of the training data for calibration. The number of calibration instances was set to Eq 13. from [1] according to [1], where Z is the full training set. This means that around 20-25% of the full training set was used for calibration.

Computer Specifications

The experiments were run on a notebook with an Intel Core i5-8250U processor (4-core 1.6 GHz base frequency) and 12GB of RAM. The conformal predictions were run on a single core, but the bootstrapping utilized all cores in parallel, which is straightforward to implement with the boot() function on a Linux system.

Chosen Parameters

Three parameters were optimized using grid search: the number of trees in a random forest (ntree), the sensitivity of the nonconformity measure in conformal prediction (beta) and the number of bootstrap replicates (R).

10000 instances were used from the training pool for the grid search, where ntree was chosen to be 125, because this value resulted in a “low enough” mean squared error. A higher number of trees would make the predictions slightly more accurate, but only at the a significantly larger computation cost.

Beta was chosen to be 0.01, because the mean of conformal prediction region sizes was the smallest with this value, and the runtime was slightly shorter than with others. The confidence level set for the interval was barely affected by beta.

The number of bootstrap replicates should be as large as possible, but increasing this parameter increases the runtime linearly, so the optimal value was set to a low value (R = 200) that still results in reasonably good coverage rates and mean region sizes.