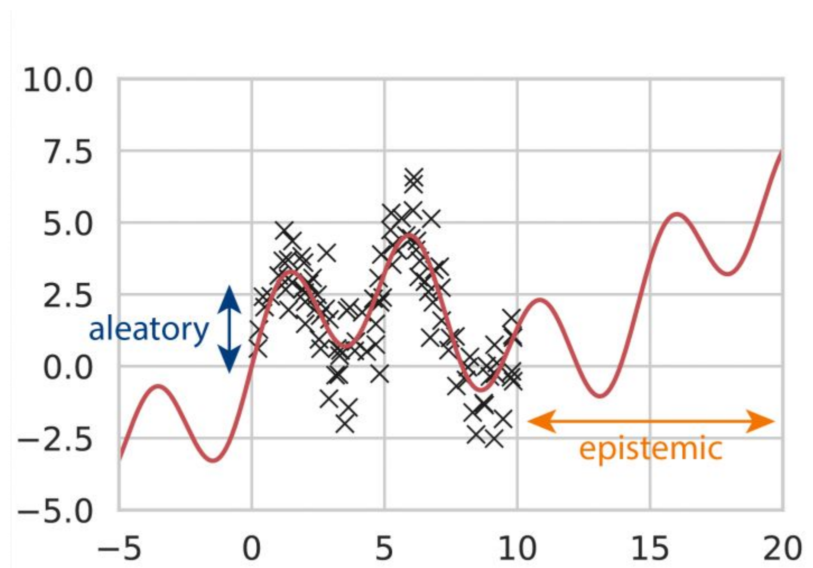
Uncertainty Quantification for Regression

Krishnan Jothi Ramalingam

Computational Sciences in Engineering, Technische Universität Braunschweig

Introduction

- In many safety critical applications and business investments, uncertainty needs to be quantified beforehand to make well informed decisions.
- Epistemic uncertainty describes what the model does not know due to lack of training data in that domain. With enough training samples, epistemic uncertainty decreases.
- Aleatoric uncertainty arises from the intrinsic randomness of the observations. Cannot be reduced with more data



- **Objectives:** Compare different uncertainty quantification methods for regression, interpret their results, provide insights into their drawbacks and discuss few other quantification techniques.

Experiments

The dataset is provided by IAV GmbH. It consists of a target variable, which is dependent on 10 other input variables. Training and test set consists of 23 and 5 time series data respectively.

Vanilla Regression:

Classical Mean Squared Error (MSE) is used as loss function to train the model.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,$$

Probabilistic Regression [1] :

- In order to account for the distributional variance of the prediction, the mean of the negative log-likelihood function of the normal distribution is used as a loss function. This model predicts the mean $\hat{y}$ and an additional output v, such that $\sigma = e^{v}$ and confidence interval is $\pm 2\sigma$.

$$NLL = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{(y_i - \hat{y}_i)^2}{2\sigma_i^2} + 2ln(\sigma_i)\right),$$

- Intuitively, the numerator of the first term in the negative log-likelihood function encourages the mean prediction $\hat{y}$ to be close to the true target y, while the denominator makes sure the variance σ^2 is large when the deviation from the mean $(y - \hat{y})^2$ is large. The second term acts as a counterweight for the variance not to grow indefinitely.

Regression with intervals [2]:

- This model directly predicts the upper bound $\overline{y}$ and lower bound $\underline{y}$ of the confidence interval
- The following loss function defines the interval loss to be zero if a target lies inside the interval and the squared distance to the interval boundary if it lies outside the interval. An additional linear penalty is employed to control the interval size. The tightness parameter $\beta > 0$ determines the outlier-sensitivity of the intervals during training. We chose $\beta = 0.002$.

$$\frac{1}{N}\sum_{i=1}^{N}\left[max(y_i - \overline{y}_i, 0)^2 + max(\underline{y}_i - y_i, 0)^2 + \beta \cdot (\overline{y}_i - \underline{y}_i)\right]$$

Monte Carlo dropout [3]:

- In Monte Carlo dropout, the network nodes on the dropout layer are not only randomly turned off during training, but also during inference. This results in a different $\hat{y}$ every time an input value is passed to the network.
- By obtaining a set of $\hat{y}$ for a given input, the parameter estimates can be computed which quantifies the distributional variance.
- In areas where there was no training data to learn from, the network behaviour is not controllable so we expect a high variance among the different network subsets and vice versa.

Performance measure:

In order to reasonably compare all the methods, we define accuracy as follows:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$f(x) = \begin{cases} 1 & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

Where,

x is the true target value in the test set

a and b are the predicted/derived upper and lower bounds respectively

Drawbacks

Probabilistic Regression :

This method captures the aleatory uncertainty well in areas of sufficient training data. The drawback is the prior assumption of the data distribution.

Regression with intervals :

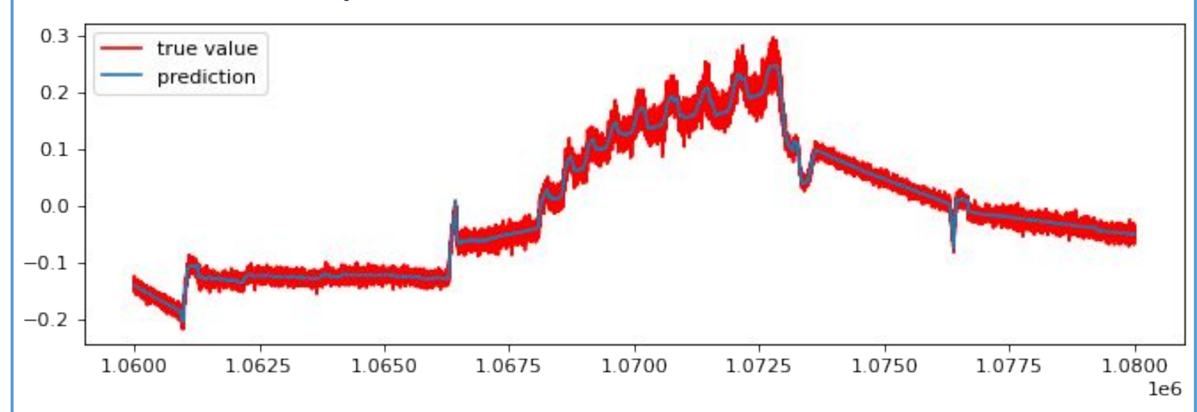
Only $\overline{y}$ and $\underline{y}$ are predicted, but the model does not tell any information about $\hat{y}$. Obtaining good confidence interval also depends on the choice of tightness parameter β, which includes an additional hyperparameter.

Monte Carlo dropout :

Obtaining a good parameter estimate depends on the large number of forward passes for a given input. Hence this method is time consuming and computationally intensive.

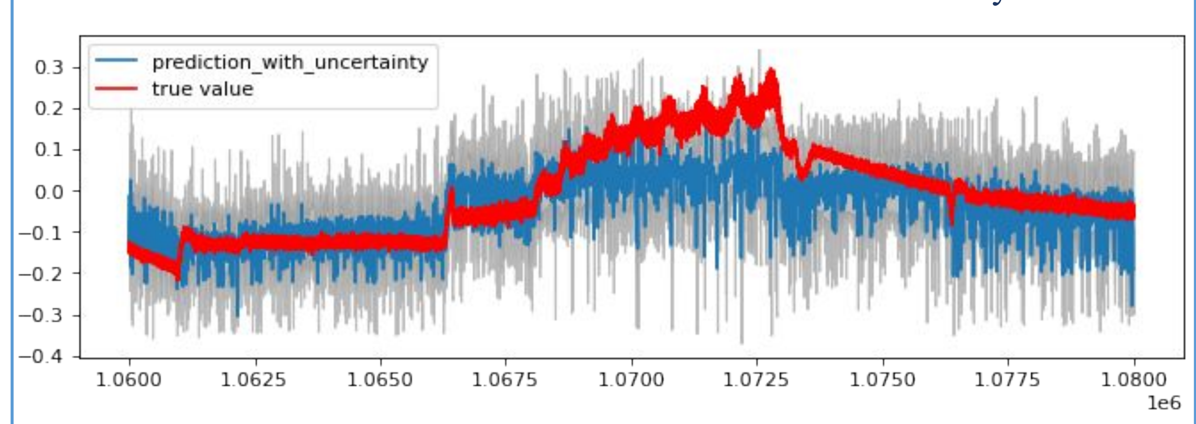
Results

Without uncertainty measure:



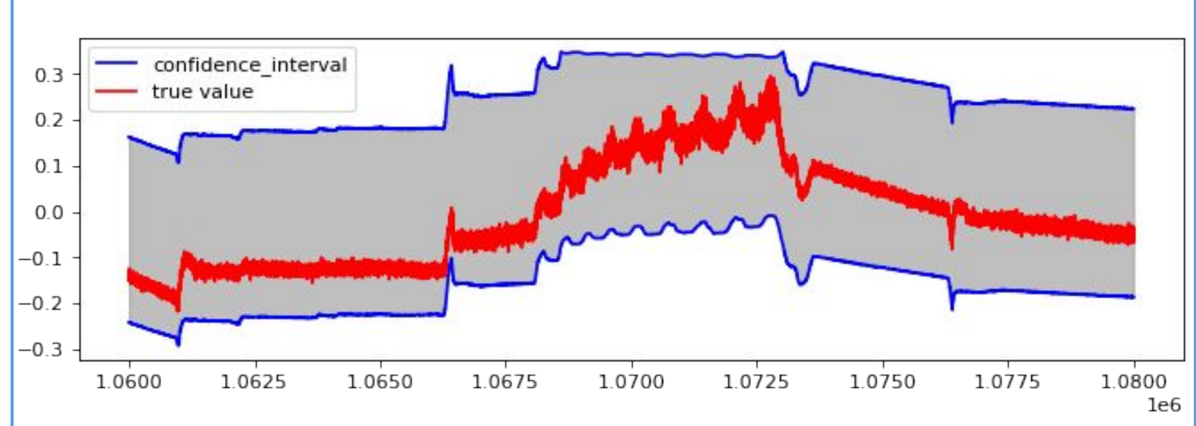
Probabilistic regression:

Accuracy = 79.12%



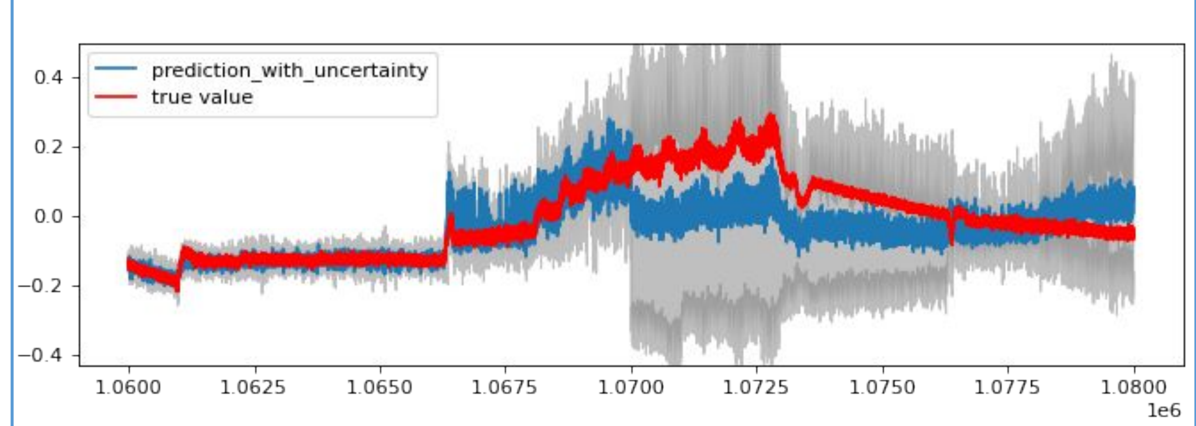
Regression with intervals:

Accuracy = 100%

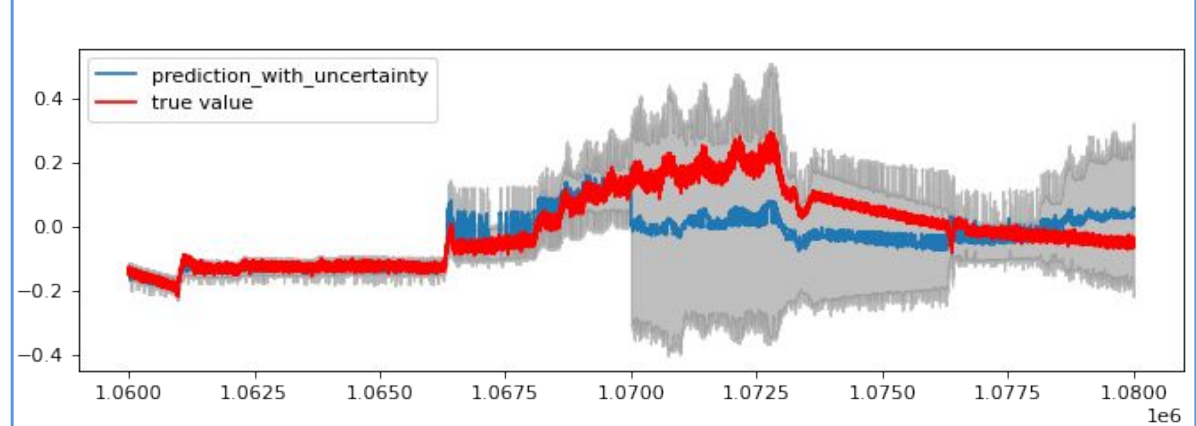


Monte Carlo dropout:

Samples = 2 Accuracy = 77.10%



Samples = 500 Accuracy = 98.62%



Conclusion

- Although the accuracy of the interval neural network is 100%, it is observed from the results that the confidence interval is relatively larger in some regions when compared to other methods. This can be modified by adjusting the tightness parameter β.
- In some regions, the confidence interval predicted by MC dropout method is quite small, but the model evaluation is very costly.
- Even if the prior data distribution is not Gaussian, the concept of probabilistic regression can still be used. If the data follows other parametric distributions, a similar approach can be used by taking NLL of complex distributions like GMM.

Other approaches:

Ensemble Averaging: Instead of training a single network, we can train an ensemble of M networks with different random initializations. For a final prediction, we can take all networks and combine their results into a Gaussian mixture distribution from which we can, again, extract single mean and variance estimations [4].

Modified MC dropout: Instead of only predicting $\hat{y}$, we can let the model predict both $\hat{y}$ and the additional output v as in probabilistic regression. Finally, the outputs of multiple forward passes are averaged.

Modified INN: Instead of only predicting the upper and lower bounds, we can also include the $\hat{y}$ as the model output. This is achieved by minimizing two different loss functions during training.

References

[1] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. CoRR, abs/1805.11327, 2018.

[2] Luis Oala, Cosmas Heiß, Jan Macdonald, Maximilian März, Wojciech Samek, and Gitta Kutyniok. Interval neural networks: Uncertainty scores, 2020.

[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

[4] Balaji Lakshminarayanan, Alexander Pritzel and Charles Blundell Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, NIPS, 2017