Jul 1st, 12:00 AM

A Novel Method to Estimate the Model Uncertainty Based on the Model Errors

Durga Lal Shrestha

Dimitri P. Solomatine

A Novel Method to Estimate the Model Uncertainty Based on the Model Errors

[1]Durga Lal Shrestha and <u>Dimitri P. Solomatine</u>

[1]UNESCO-IHE Institute for Water Education, Delft, The Netherlands

E-mail: d.solomatine@unesco-ihe.org

Abstract: This paper presents a novel method for estimating "total" predictive uncertainty using machine learning techniques. By the term "total" we mean that all sources of uncertainty are taken into account, including that of the input and observed data, model parameters and structure, without attempting to separate the contribution given by these different sources. We assume that the model error, which is mismatch between the observed and modelled value reflects all sources of uncertainty. Fuzzy *c*-means clustering was employed to cluster the input space into different zones or clusters assuming that the all the examples those belong to the particular cluster have similar model errors. The prediction interval is constructed for each cluster on the basis of empirical distributions of the historical model errors associated with all examples of the particular cluster. Prediction interval for the individual example is derived from cluster based prediction interval according to their membership grades in each cluster. Linear or non-linear regression model is then built in calibration data that approximates an underlying functional relationship between an input vector and the computed prediction intervals. Finally, this model is applied to estimate the prediction intervals in verification data. The method was tested on hydrologic datasets using various machine learning techniques. Preliminary results show that the method has certain advantage if compared to other methods.

Keywords: Model uncertainty; prediction interval; fuzzy clustering.

1. INTRODUCTION

In forecasting environmental variables the decision makers often require not only point forecasts but also the associated uncertainty estimates. In weather prediction such practice is common but in other areas, e.g., in water and environmental management, the prevailing format of forecasting was for a long time deterministic and rarely took into account the various sources of uncertainties - input data, observed data, parameter, and model structure. Lately there is an increased interest to developing methods to quantify model uncertainty. This can be done using several approaches:

- forecasting the model outputs probabilistically as it is often used in hydrological modeling [Krzysztofowicz, 2000];
- estimating uncertainty by analyzing the statistical properties of the model errors that occurred in reproducing the observed historical data. This approach has been used for time series forecasting [Wonnacott and Wonnacott, 1996, Nix and Weigend, 1994].

- simulation and re-sampling based techniques, generally referred to as ensemble, or Monte Carlo methods (one of the versions of such approach used in hydrologic modeling is a *generalized likelihood uncertainty estimator*, GLUE [Beven and Binley, 1992]).
- fuzzy theory based methods [Abebe et al., 2000; Maskey et al., 2004].

The first and the third approaches require the prior distributions of the uncertainty of the input parameters to be propagated through the model to the outputs. The second approach requires certain assumptions about the data and the errors, and, obviously, the relevance and accuracy of such approach depends on the validity of these assumptions. The last approach requires knowledge of the membership function of the quantity subject to the uncertainty. It should be noted that most of the researches are considering the individual sources of uncertainty (for example parameter or input data uncertainty) rather than the combined effect of all sources of uncertainty.

This paper presents a novel approach using machine learning techniques to estimate the total model uncertainty that takes into account all sources of uncertainty without attempting to separate the contributions given by the different sources of uncertainty. We assume that the model error, which is mismatch between the observed and modelled value, reflects all the sources of uncertainty. In this paper, uncertainty is quantified in the form of two quantiles of the underlying distribution of model errors. Training (calibration) set is partitioned into different clusters having similar model errors; machine learning models are built for prediction intervals (PI) for each cluster and for each example. The proposed method is employed to estimate the PIs by several machine learning techniques for a number of environmental datasets, and is compared to other methods.

2. PREDICTION INTERVAL

An interval forecast is usually comprised of the upper and lower limits between which a future unknown value (e.g. a point forecast) is expected to lie with a prescribed probability. This limit is called prediction limit (PL) or bound, while the interval is called the *prediction interval* (PI) (Figure 1). The prescribed probability is called *confidence level*. The following sub-sections briefly present the methods for constructing PI for the model outputs.
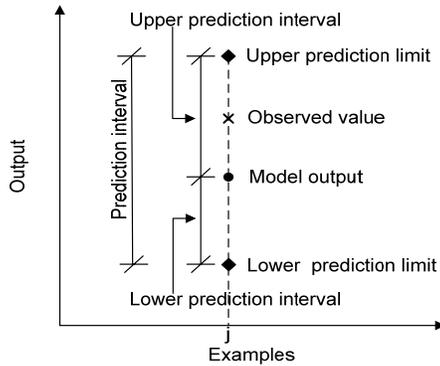

Figure 1. Terminology used in the paper.

2.1 Prediction Interval for Linear Regression

We assume to have the regression model $y = f(\mathbf{x})$ to predict a set of the observed (target) values t_i, (i=1,..., n) associated with the real-valued input vectors $\mathbf{x}_i$, $\mathbf{x}_i \in \Re^m$. Most of the methods to construct $100(1-\alpha)$% prediction limit (PL) for the model output typically assume that the error has Gaussian distribution with zero mean (so model bias is zero) and the standard

deviation σ and for one dimensional input (m=1) are expressed as:

$$PL^U = y + z_{\alpha/2}\sigma \quad , PL^L = y - z_{\alpha/2}\sigma \tag{1}$$

where PL^U and PL^L are the upper and lower PLs respectively, $z_{\alpha/2}$ is the value of the standard normal variate $N(0,1)$ with cumulative probability level of $\alpha/2$. Since prediction is assumed unbiased, PLs in (1) are symmetric about y. Generally error variance σ^2 is not known in practice and is estimated from the data. An unbiased estimate of σ^2 with n–p degrees of freedom, denoted by s^2, is given by the formula:

$$s^2 = SSE / (n\text{-}p) = \frac{1}{n-p}\sum_{i=1}^{n}(t_i - y_i)^2 \tag{2}$$

where p is number of parameters in the model and *SSE* is the *sum squared error.*

If the error variance s^2 is not constant in the output space (i.e. y is heteroscedastic), then (2) can be modified to give an estimate for model output y_i for each observation i as follows [Wonnacott and Wonnacott, 1996]:

$$s_{y_i}^2 = s^2(1 + 1/n + (x_i - \overline{x})^2 /(n-1)s_x^2) \tag{3}$$

where x_i; s_x^2 and $\overline{x}$ are the input, input sample variance and input mean respectively; $i = 1,...,n$. It can be seen that the error variance for the output y_i is always larger than s^2 and it depends on how far x_i is from $\overline{x}$. For multivariate linear regression, (3) can be modified as:

$$s_{y_i}^2 = s^2(1 + \mathbf{x}_i^T (X^T X)^{-1}\mathbf{x}_i) \tag{4}$$

where X is a matrix of input space, appended to a column of 1's as the leftmost column, and $\mathbf{x}_i$ is i[th] row of matrix X. This method will be referred to as *linear regression variance estimator* (LRVE) method.

2.2 Prediction Interval for Non-Linear Regression

For non-linear models especially for large number of variables with complex relationship and black box type models (for example, artificial neural networks) derivation of the error variance s^2, and hence computation of the PI is not so easy. However, resampling based techniques have been reported in the literature to estimate s^2 and thus to compute the PI for an artificial neural network (ANN), one of the non-linear regression models (see for example, Nix and Weigend [1994]. Typically, these techniques are

based on the premise that the error variance s^2 can be decomposed into three terms: model bias, model variance, and target noise. The model variance can be estimated by building an ensemble of ANNs using data resampling. Target noise is estimated by training yet another ANN on the residuals of this ensemble's predictions. The mentioned methods assume the zero mean of the error distribution (zero model bias) and this assumption is very often not justified. The necessity to generate many model ensembles to ensure a reliable estimate leads to high computation times.

3. MACHINE LEARNING TECHNIQUES

A machine learning (ML) technique is an algorithm that estimates an unknown mapping between a system's inputs and its outputs from the available data [Mitchell, 1998]. As such a dependency is discovered, it can be used to predict the future system's output from the known input values. In this paper we used artificial neural networks (ANN), locally weighted regression (LWR) and M5 model trees (MT) as such techniques.

An ANN is the most widely used ML technique and regarded as universal function approximation due to its ability to represent both linear and non-linear relationships Haykin [1999]. ANNs consist of a large number of simple processing elements called *neurons* or *nodes*. Each neuron is connected to other neurons by means of direct links, each being associated with a weight that represents information being used by the network in its effort to solve the problem. The weights are determined by training the networks based on pairs of input-output dataset.

A model tree (MT) is hierarchical (tree-like) modular model consisting of splitting rules in non-terminal nodes and the multivariate linear regression models at the leaves, so it is analogous to a piecewise linear function. MT learning is fast and the results are interpretable. See Quinlan [1992] for more details.

Locally weighted regression (LWR) (see Atkeson *et al.* [1997]) is an instance-based method; it predicts the given input instance by querying entire instance space to find those instances which are local (similar) to the given input instance and predicting based on those local instances. LWR generates local models by giving a higher weight to the instances in the neighbourhood of new input vector. It weights the training instances according to their distance to the test instance and builds a linear regression on the weighted data. Training instances close to the test instance receive a higher weight and those far away – a lower one.

Clustering involves the task of partitioning a dataset into a number of homogenous clusters with respect to a suitable similarity measure. In the traditional hard clustering (e.g., using *k*-means method), each data point is assumed to be in exactly one cluster. This condition can be relaxed and allow for each instance to belong to a cluster with some degree, interpreted as a "fuzzy" membership in a cluster. A point may belong to several clusters with some degree (membership grade) in the range [0, 1]. The most known method of fuzzy clustering is the *fuzzy c-means* (FCM) [Bezdek, 1981].

4. METHODOLOGY

Due to the various sources of uncertainty mentioned in the section 1, it is not surprising that the model outputs do not match the observed values well. The proposed method is based on an idea that the historical residuals (errors) between the model outputs and the observed data are the best available quantitative indicators of the discrepancy between the model and the real-world system or process, and give the valuable information that can be used to assess the model uncertainty. These residuals are often functions of the model input's values and can be modelled. Note that in contrast to the methods considered above, we do not assume any distribution of model errors; as a consequence, the model bias can be non-zero.
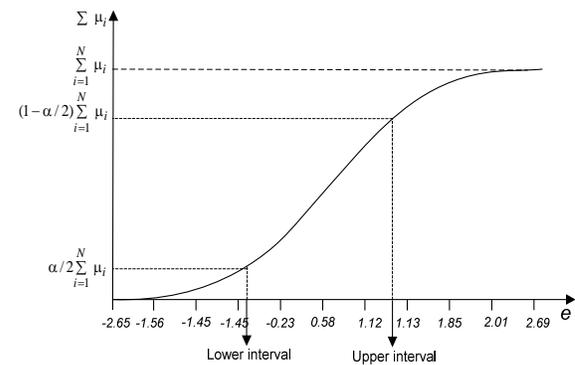
Figure 2. Computation of the prediction interval in case of using fuzzy c-means clustering.

The input dataset can be partitioned into several clusters corresponding to different values of historical residuals. It can be assumed that the region in the input space that is associated with any particular cluster has residuals with similar values. Having identified the clusters, the PIs for each cluster are computed from empirical distributions of the corresponding historical residuals. For instance, in order to construct $100(1-\alpha)$% PI, the $\alpha/2*100$ and

(1-α/2)*100 percentile values are taken from empirical distribution of residuals for lower and upper PI respectively. Typical value for α is 0.05, which corresponds to 95% confidence interval. If the input space is divided into crisp clusters, e.g., by k-means clustering, and each instance belongs to exactly one cluster, this computation is straightforward. However, in the case of fuzzy clustering where each instance belongs to more than one cluster and is associated with several membership grades, the computation of the above percentiles should take this into account. To calculate PI, the instances should first be sorted with respect to the corresponding errors in ascending order. The following expression gives the lower prediction interval (PIC, see Figure 1 for terminology) for cluster i:

$$PIC_i^L = e_j \qquad j: \sum_{k=1}^{j} \mu_{i,k} < \alpha/2 \sum_{k=1}^{n} \mu_{i,j} \qquad (5)$$

where j is the maximum value of it that satisfies the above inequality, e_j is the error associated with the instance j (instances are sorted), $\mu_{i,j}$ is the membership grade of the jth instance to cluster i. Similar type of expression can be obtained for the upper PI (PIC^U). This is illustrated in Figure 2.

Once the PI is computed for each cluster, the PI for each instance in input space can be computed; note that this computation also depends upon the clustering technique employed. If crisp clustering is employed, then the PI for each instance in the particular cluster is the same as that of the cluster. In case of fuzzy clustering, the so called "fuzzy committee" approach is used and the PI is computed using the weighted mean of the PI of each cluster as:

$$PI_j^L = \sum_{i=1}^{c} \mu_{i,j} \, PIC_i^L \quad , PI_j^U = \sum_{i=1}^{c} \mu_{i,j} \, PIC_i^U \qquad (6)$$

where PI_j^L and PI_j^U are the lower and upper PI for j^{th} instance respectively. Once the lower and upper PI for each input instance is obtained, PLs are computed by simply adding model output to them:

$$PL_j^L = y_j + PI_j^L \quad , PL_j^U = y_j + PI_j^U \qquad (7)$$

where PL_j^L and PL_j^U are the lower and upper PLs for j^{th} instance respectively. Having these, two independent mapping functions are constructed that estimate an underlying functional relationship between an input **x** and the computed PLs limits as:

$$PL^L = f_U^L(\mathrm{x};\ \theta^L), \quad PL^U = f_U^U(\mathrm{x};\ \theta^U) \qquad (8)$$

where the mapping functions $f_U^L(.)$ and $f_U^U(.)$ estimate the lower and upper PLs respectively, θ^L and θ^U are their parameters. Note that mapping functions can take any form, from linear regression to non-linear functions such as ANN. The target variable of these functions might be either the PI or PL. The mapping function f_U will be referred to as the *local uncertainty estimation model* (LUEM).

We assess the performance of the LUEM by evaluating the *prediction interval coverage probability* (PICP). The PICP is the probability that the target of an input pattern lies within the estimated PLs and is computed by the corresponding frequency as follows:

$$\mathrm{PICP} = \frac{1}{n}\mathrm{count}\, j, \; j: PL_j^L \le t_j \le PL_j^U \qquad (9)$$

If the clustering technique and the LUEM are optimal, then the PICP value will be consistently close to the (1-α)%. Another performance measure for the PL was used as well. This is the *mean prediction interval* (MPI) calculated across all points in the test dataset. It is estimated by

$$\mathrm{MPI} = \frac{1}{n} \sum_{j=1}^{n} [PL_j^U - PL_j^L] \qquad (10)$$

5. EXPERIMENTAL DESIGN

5.1 Datasets

The method was tested on a number of datasets; here the results for hydrologic datasets are reported. They related to the river flows prediction in the Sieve catchment in Italy [e.g., Solomatine and Dulal, 2003]. Prediction of river flows Q_{t+i} several hours ahead (i=1, 3 or 6) is based on using the previous values of flow ($Q_{t-\tau q}$) and previous values of rainfall ($RE_{t-\tau r}$), where τq is between 0 and 2 hours and τr is between 0 and 5 hours. The regression models were based on 1854 examples. Test data consisted of 300 instances. Note that the input variables were the same for the prediction model, clustering, and LUEM.

5.2 Procedure

A LUEM model was constructed to estimate the PI on the test dataset as follows. The Fuzzy c-means

clustering technique was first employed to construct the PI for each cluster and then each instance in the training dataset. Note that the input to the LUEM may constitute part or all of input variables, which are used in the prediction model. The targets for the LUEM are the upper and lower PLs which are computed from the PIs by adding model outputs. The PLs were constructed for 95% confidence level unless specified otherwise.

First, the LUEM using bivariate linear regression was employed for dataset consisting the two most influencing input variables (variables with the highest correlation with the output). Then the input variables set was extended (SieveQ1, SieveQ3 and SieveQ6 datasets). To estimate the effect of models' complexity on the PLs, experiments were also conducted using LWR, MT and ANNs.

6. RESULTS AND DISCUSSION

The number of clusters was optimized (more on that see [Shrestha and Solomatine, 2006]) using the Xie-Beni separation index. The optimal numbers of clusters are between 4 and 6. Figure 3 shows clustering of input examples in Sieve catchment for 1 hour ahead prediction of runoff (SieveQ1 dataset). The results show that the input examples with very high runoff have maximum membership grades to Cluster 1 (denoted by C1). The input examples with very low values of runoff have maximum membership grades to cluster 5 (C5).
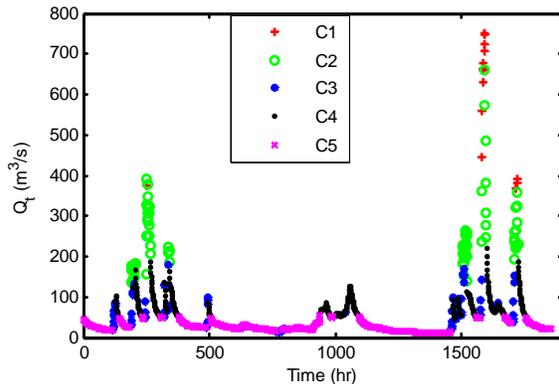

Figure 3. Clustering of input examples in SieveQ1 training dataset using fuzzy c-means clustering.

The performance of the LUEM is compared to that of the LRVE approach on the dataset with two input variables. It is observed that the LUEM shows superior performance with respect to both the PICP and the MPI. The performance of the models with more lagged input variables was compared to that of other machine learning techniques. We employed

multiple linear regression, LWR, ANN and MT to predict runoff 1, 3 and 6 hours ahead (SieveQi datasets). We also used these methods to estimate the PLs. The results show that the performance of MT is better than that of the other methods; performances of linear regression and LWR are comparable.

Figure 4 shows the computed PLs for 95% confidence level in SieveQ1 test dataset using MT. It can be seen that 96.67% of the observed data are enclosed within the PLs. This value is very close to the desired value of 95%. We compared the results with the *uniform interval method* (UIM) that constructs single PI from the empirical distribution of errors on the whole training data and is applied uniformly to the test dataset. The LUEM performs consistently better than the UIM as PICPs of LUEM are closer to the desired confidence level.
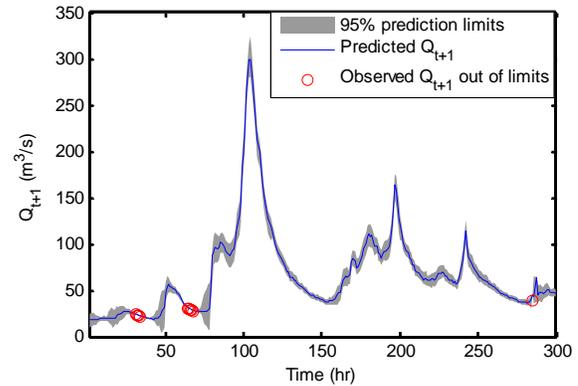
Figure 4. Computed prediction limits for SieveQ1 test dataset.
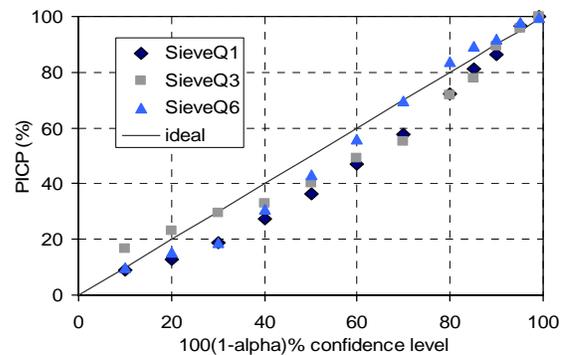
**Figure 5.** The PICP for different values of confidence level.

Figure 5 shows the deviation of the PICPs from the desired confidence level (MT model was used). The PIs were constructed for various confidence levels ranging from 10% to 99%. It is to be noticed that the PICPs are very close to the desired confidence levels

at values higher than 80% and in practice the PI are constructed around this value. Furthermore, it can be noted that the PIs are too narrow in most of the cases as the PICPs are below the straight line. Such evidence was also reported by Chatfield [2000]. In these cases the LUEM underestimates uncertainty of the model outputs. Table 1 shows the computed PIs (SieveQ1 dataset) using 95% degree confidence level.

Table 1. Results on test dataset using M5 model tree.

Experi-ment	RMSE	LUEM		UIM	
		PICP	MPI	PICP	MPI
SieveQ1	3.61	96.67	15.25	91.33	11.80
SieveQ3	13.67	95.67	43.27	89.33	40.58
SieveQ6	22.89	97.67	96.34	91.33	81.6

Figure 6 presents a fan chart showing the MPI with the different forecast lead times and the different confidence levels. It is evident that the width of the PI increases with the increase of the confidence level. Moreover it is also illustrated that the width of PI increases as forecast lead time increases
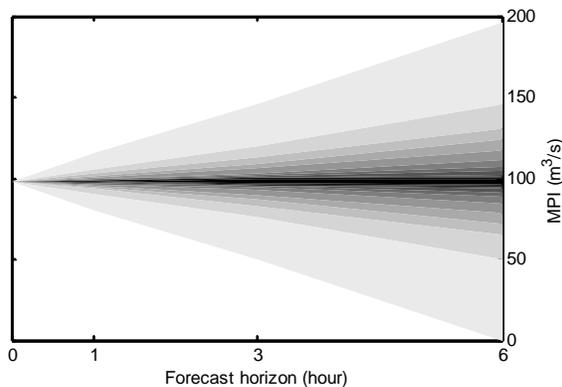
Figure 6. Fan chart showing the model uncertainty for various forecast horizons. The darkest strip covers 10% probability and the lightest - 99%.

7. CONCLUSIONS

A novel method to estimate the total uncertainty of the model outputs using machine learning techniques is presented. It explicitly takes into account all sources of uncertainty of the model outputs and is independent of the prediction model structure as it requires only the model outputs. Unlike the existing techniques the methodology does not require the knowledge of prior distribution of parameters or errors. The upper and lower prediction intervals are calculated independently.

The methodology was applied to the data-driven prediction (regression) models based on both artificial and real hydrologic datasets, and was compared to LRVE approach typically used with the linear regression models. The advantages of the new method were demonstrated.

8. REFERENCES

Abebe A.J., Guinot, V., and Solomatine, D.P., Fuzzy alpha-cut vs. Monte Carlo techniques in assessing uncertainty in model parameters. *Proc. 4th Int. Conf. on Hydroinformatics*. Cedar-Rapids, 2000.

Atkeson, C.G., Moore, A.W., and Schaal, S., Locally weighted learning, *Artificial Intelligence Review*, *11*(1-5), 11-73, 1997.

Beven, K.J., and Binley, J., The future of distributed models: model calibration and uncertainty prediction, *Hydrological Processes*, *6*, 279-298, 1992.

Bezdek, J.C., *Pattern recognition with fuzzy objective function algorithms,* Norwell, MA, USA: Kluwer Academic Publishers, 1981.

Chatfield, C., *Time series forecasting,* Chapman & Hall/CRC, 2000.

Haykin, S., *Neural Networks: a comprehensive foundation*, Prentice Hall, NJ, 1999.

Krzysztofowicz, R., The case for probabilistic forecasting in hydrology, *J. of Hydrology*, 249, 2-9, 2000.

Maskey, S., Guinot, V., and Price, R.K., Treatment of precipitation uncertainty in rainfall-runoff modelling, *Adv. in Water Resources*, 27, 889-898, 2004.

Mitchell, T.M., *Machine Learning*, McGraw-Hill, 1998.

Nix, D. and Weigend, A., Estimating the mean and variance of the target probability distribution, *Proc. of the Int. Joint Conference in Neural Networks, IEEE*, 55-60, 1994.

Quinlan, J.R., Learning with continuous classes, Proc. of the 5th Australian Joint Conference on AI, World Scientific, Singapore, 343-348, 1992.

Solomatine, D.P., and Dulal, K.N., Model tree as an alternative to neural network in rainfall-runoff modelling, *Hydrol. Sci. J.*, 48(3), 399–411, 2003.

Shrestha, D.L., and D.P. Solomatine, Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks*, 19(2), 225-235, 2006.

Wonnacott, T.H. and Wonnacott, R.J., *Introductory Statistics*, John Wiley & Sons, Inc., 1996.