

# Black-box forecasting and uncertainty

P.J.J.F. Torfs

Wageningen University

## Abstract

In this paper we investigate why it is so difficult to improve the current operational predicting model for Lobith, despite the fact that it is “only” a multi-linear regression model. One of the important reasons for this is the fact that input selection is difficult to generalize to the non-linear case. An example of a more successful approach finishes the article.

## 1 Predicting Lobith

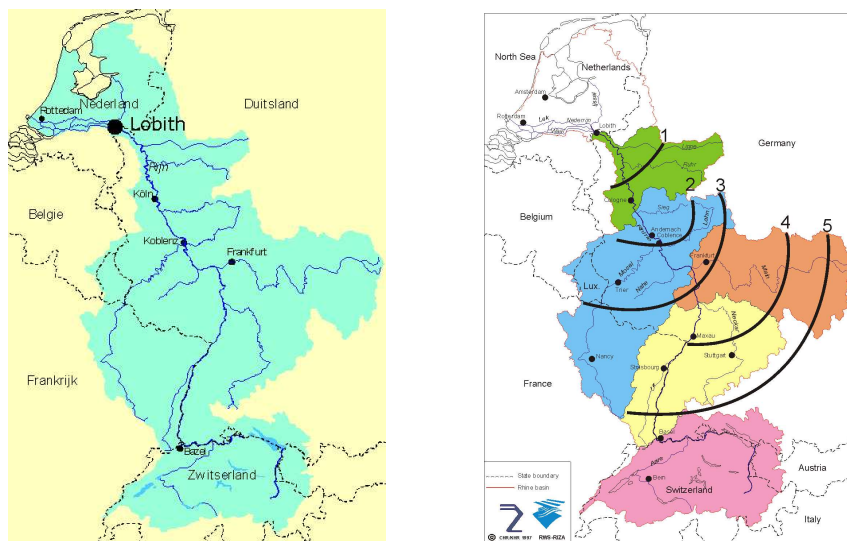


Figure 1: Left: the Rhine catchment and the location of Lobith. Right: traveling times to Lobith.

Figure 1 shows the Rhine catchment, and the location of Lobith at the point where the Rhine enters the Netherlands. This paper discusses only the predictions made by black-box models for the water levels at Lobith for a few (1,2,...) days ahead.

There do exist more physically (both hydrologically and hydraulically) oriented approaches (see e.g. [Spr01]). It is the authors opinion that the future is for hybrid models, where black-box models are used to statistically improve the residuals of physical models.

The decision for mass evacuation (involving more than 250.000 people) during the flood of 1995 was partly based on the prediction model discussed here, and is a prove of the importance of good prediction models in general.

During this flood, the prediction model performed rather well, certainly for the first two days ahead, as illustrated by figure 2.

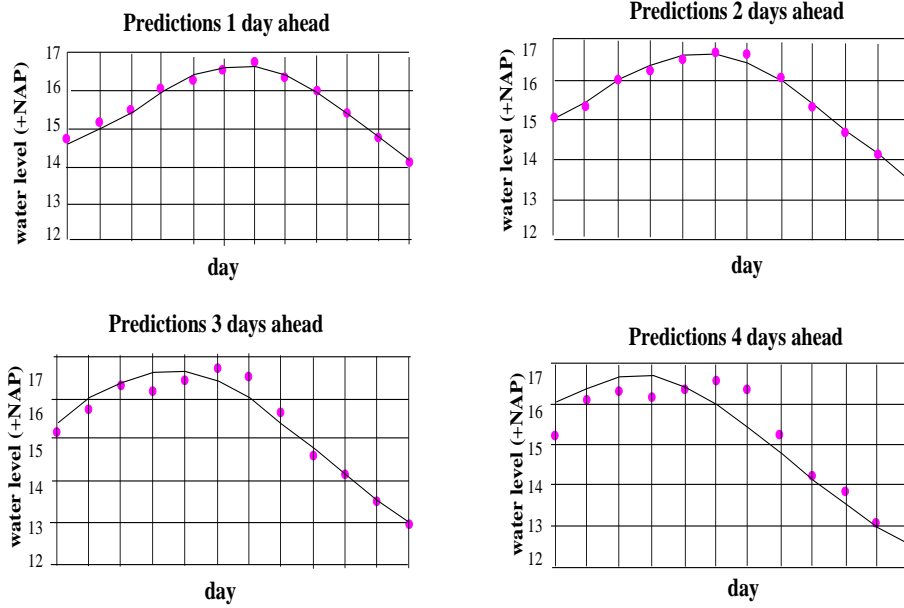


Figure 2: The flood wave of 1995: actual water levels and predictions 1, 2, 3 and 4 days ahead

## 2 The multi-linear regression model

The current operation black-box models are M(ulti) L(inear) R(egression) models. Figure 2 shows some results of these. Formula (1) and formula (2) show two examples<sup>1</sup> (as they are after the last recalibration in 1997, see [PS97] for more details). In these formulas,  $H$  stands for water level,  $P$  for precipitation and  $t$  for time measured in days.

$$\begin{aligned}
 H_{\text{Lobith}}(t+1) = & 0.854 H_{\text{Lobith}}(t) \\
 & + 0.472 H_{\text{Köln}}(t) - 0.333 H_{\text{Köln}}(t-2) \\
 & - 0.031 H_{\text{Plochingen}}(t-2) \\
 & + 0.172 H_{\text{Hattingen}}(t) - 0.146 H_{\text{Hattingen}}(t-1) \\
 & + 0.250 P_{\text{Düsseldorf}}(t-1)
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 H_{\text{Lobith}}(t+2) = & -0.122 H_{\text{Maxau}}(t-3) + 0.157 H_{\text{Worms}}(t) \\
 & + 0.493 H_{\text{Kaub}}(t) - 0.208 H_{\text{Andernach}}(t-3) \\
 & - 0.254 H_{\text{Köln}}(t-1) + 0.810 H_{\text{Lobith}}(t) \\
 & - 0.159 H_{\text{Plochingen}}(t-2) + 0.200 H_{\text{Trier}}(t) \\
 & + 0.243 H_{\text{Kalkofen}}(t) - 0.147 H_{\text{Kalkofen}}(t-1) \\
 & + 0.296 H_{\text{Hattingen}}(t) - 0.250 H_{\text{Hattingen}}(t-1)
 \end{aligned} \tag{2}$$

It is tempting to interpret the lags in the formulas as physical, but the five day difference for instance in formula (2) between Lobith ( $t+2$ ) and Andernach ( $t-3$ ) is out of the physical range (see also figure 1). Also the negative coefficients can not be physically interpreted. The MLR models are indeed black box models.

1. Both equations are meant to be used for cases where  $H_{\text{Lobith}}(t) > 10$  m.

Both formulas use only a selection of the possible inputs: more stations are available (figure 3) and other choices of lags can be made. Making a good selection is an essential step in building a successful MLR model, as we will demonstrate in the next section.

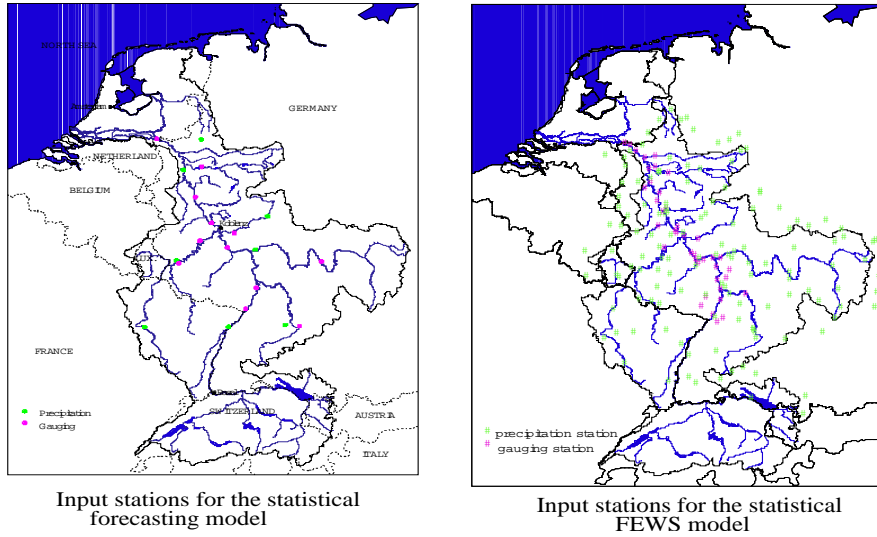


Figure 3: Available data as used in calibrating the MLR model (left) and as available within the FEWS project

### 3 Input selection in MLR modeling: an example

The input selection problem in MLR modeling is well known, and well documented (see [BD59] for a classical reference). Table 1 shows a school book example for MLR modeling. One has to predict the variable  $\mathbf{G}$  based upon a selection of the possible inputs ( $\mathbf{M}$ ,  $\mathbf{H}$  and  $\mathbf{A}$ ).

$\mathbf{G}$	$\mathbf{M}$	$\mathbf{H}$	$\mathbf{A}$
4.4	4.97	39.7	89.72
5	5.01	39.8	92.34
5.5	5.26	40.4	96.56
6.25	5.49	40.5	99.63
7	5.7	40.7	102.97
7.75	6.03	41.2	107.53
8.525	6.55	41.3	112.3
9.1	6.85	40.6	114.9
9.3	7.43	40.7	122.51
9.5	8.02	40.6	129.51
9.7	8.34	39.8	133.73

Table 1: The example data: number of golfers  $\mathbf{G}$  (in millions) in the USA between 1960 and 1970.  $\mathbf{M}$  stands for median income,  $\mathbf{H}$  for average working hours/week and  $\mathbf{A}$  for the average weekly earnings.

I	M	H	A	R <sup>2</sup>
-2.1533 ± 1.1791	1.5178 ± 0.1833			0.8840
-56.268 ± 43.973		1.574 ± 1.086		0.1892
-6.17867 ± 1.42173			0.12481 ± 0.01291	0.9122

Table 2: Results of the MLR models with only one input. The **I** column stands for the intercept. The entries are the coefficients. The standard deviations of each coefficient is given underneath (preceded by the  $\pm$  sign).

Table 2 shows MLR models made by selecting only one input. Using the  $R^2$  as criterion, the model with **A** as input is clearly better than the model build upon **M** and much better than the model with **H**.

Table 3 shows all MLR models with two inputs. The best model (**I+M+H**) is now a combinations of two inputs which were the weakest in the one-input-case. This clearly shows that the best MLR models are not obtained only by extending existing ones with new input.

The reason for this is *multicollinearity*: the variables **M** and **A** are strongly linear dependent. Not only do they measure more or less the same economical concept (median vs. average and income vs earnings), but also the correlation coefficient (a perfect measure for linear dependency) is extremely high:  $\text{COR}[\mathbf{M}, \mathbf{A}] \approx .9963$ . Because the **H** variable is only poorly correlated with the other ( $\text{COR}[\mathbf{H}, \mathbf{A}] \approx 0.213$  and  $\text{COR}[\mathbf{H}, \mathbf{M}] \approx 0.155$ ), its inclusion into a MLR model adds new information and improves the performance.

I	M	H	A	R <sup>2</sup>
-45.0852 ± 9.0547	1.4436 ± 0.1006	1.0721 ± 0.2256		0.9697
-40.974432 ± 9.275522		0.877762 ± 0.232869	0.118059 ± 0.008409	0.9684
-12.3271 ± 4.5712	-2.4797 ± 1.7615		0.3248 ± 0.1426	0.9296

Table 3: Results of MLR models with two inputs.

Table 4 gives the result when one selects all the available inputs to build one single MLR model. Although more complicated than the models of table 3, there is only a negligible improvement in  $R^2$ . The size of the errors on the coefficients is however alarming, showing that the three input model is badly identifiable. This is again due to the multicollinearity as discussed above. A badly identified model quickly leads to over-fitting and the use of it should be avoided.

I	M	H	A	R <sup>2</sup>
-43.85402 ± 10.77331	1.01578 ± 1.67805	1.01394 ± 0.33091	0.03509 ± 0.13735	0.9699

Table 4: Results of the MLR model with all inputs selected.

## 4 Input selection for the predictions of Lobith

For a MLR model for Lobith predictions, observations from many locations are available (see fig 3). From these stations, one can moreover take many lags as possible input (as e.g. the  $H_{\text{Andernach}}(t - 3)$  in equation (2)). In this way the set of possible inputs quickly grows to sizes of 50 and more. The dependency between these inputs is *very high*, as they are all connected by the physical processes of rainfall runoff generation and flood routing. Multicollinearity is certainly present, and the addition of new stations only increases this.

Trying all possible subsets of inputs is virtually impossible. Assuming that fitting a single model takes 1 second,

- trying every possible selection of 25 inputs takes:

$$2^{25} / (60 * 60 * 24 * 365) \approx 1 \text{ year}$$

- trying every possible selection for 50 inputs takes:

$$2^{50} / (60 * 60 * 24 * 365 * 100) \approx 360\,000 \text{ centuries}$$

For this reason, many statistical packages have a sequential selection technique to make MLR models. A typical *forward stepwise selection* step selects one extra input e.g. that one with the highest correlation with the residuals. A typical *backward stepwise selection* steps drops that input that gives the least decrease in  $R^2$ . Most implementations use a combination of both forward and backward steps in combination with a stopping criterion (as was also needed in the example of previous section).

Formulas (1) and (2) were obtained using the sequential stepwise regression technique of the SPSS package.

## 5 Non linear regression techniques

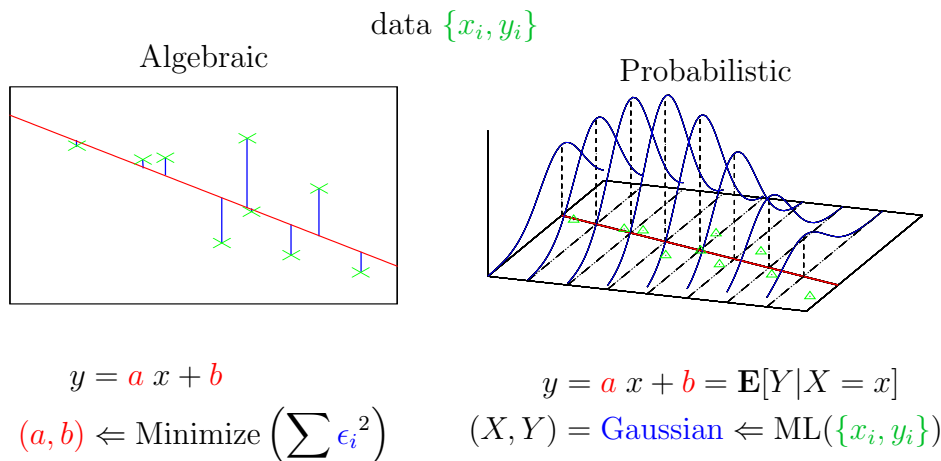


Figure 4: Two views on linear regression: algebraic (left) and probabilistic (right).

It is very tempting to try to improve the prediction for the river Rhine at Lobith by using non linear regression techniques. Nowadays many non-linear black-box models are available (see e.g [HTF01] for some overview), both in theory and in practice:

- Neural Networks
- M(ultivariate)A(daptive)R(egression)S(plines)

- M(ultiple)A(dditive)R(egression)T(rees)
- ...
- Kernel Regression

One way of classifying these models is to start from two fundamental views on linear regression, as illustrated by figure 4. The first view is purely algebraic: one fits the parameters of an algebraic formula –in this case a linear one– such that the data are approximated as good as possible (measured with a least squares criterion). The other view is probabilistic: the data are considered to be a sample from a pair of Gaussian distributed stochastic variables. The prediction for the dependent variable  $y$  is then obtained by taking a conditional expectation.

Neural Networks are an example of a non-linear algebraic regression approach, and will be introduced in the next section. Kernel regression by Mixtures of Gaussians form an example of the generalization of the probabilistic approach, and will be discussed in the section after the next one.

Applying non-linear regression techniques has high potentials, as (see also figure 5):

- **non linear** features may be captured
- new forms of output may give **more information** concerning the uncertainty around the mean predictions.

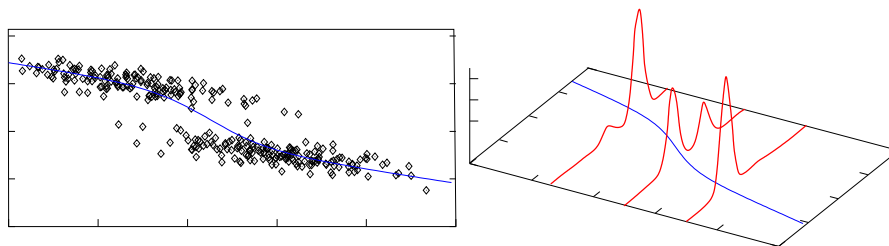


Figure 5: Left: an algebraic non-linear regression result. Right: a probabilistic result (to the same data).

This last point will also be treated in section 9.

## 6 Neural Networks

The particular type of N(eural)N(etworks) discussed in this section are sometimes also called feed-forward NN (see e.g. [Bis98] for an overview of this and other NN).

This simplest NN is that consisting of one neuron. Figure 6 shows an example which has two (can be any number in general) inputs  $x, y$  and two corresponding weights  $(-10.4, 11.4)$ , which are the parameters of the model. There is always a single output called  $z$ . The weighted input is non-linearly transformed into the output by the formula:

$$z = \tanh(-10.4 * x + 11.4 * y) \tag{3}$$

The weights determine the character of the NN. In figure 6, the weights are rather high so that the output is mostly close to 1 or -1, resulting in an almost step function. Figure 7 shows a single neuron with small weights, so that more of the more linear middle part of the transfer function is sampled, resulting in an almost linear relationship.

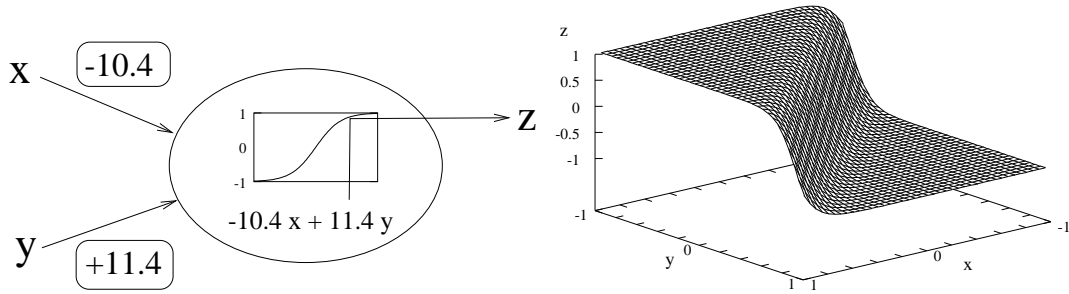


Figure 6: A single neuron with two inputs and one output. Left: the structure, right: the resulting nonlinear function  $z = \text{NN}(x, y)$ .

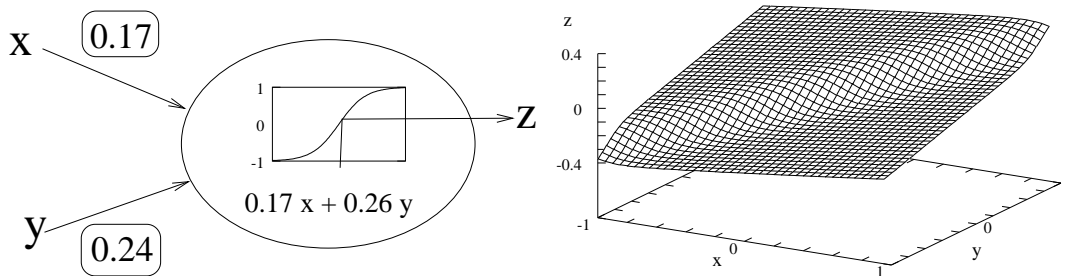


Figure 7: A single neuron resulting in an almost linear relationship

General NN consist of several interconnected neurons as in figure 8. By using the same simple building block of a neuron, NN's can model all kinds of functions, although more complex functions do require more neurons. For this reason, one sometimes calls NN *universal approximators*.

Figure 9 shows a more general NN with two inputs and one output. It is the modelers responsibility to choose the number of hidden layers (usually two or more) and the number of neurons in each layer. Once these are chosen, the weights in the NN should be found by a fitting procedure to the given data<sup>2</sup>. For this many algorithms are available (see e.g. [Bis98]), usually minimizing the squared error. It is important to note that is in general a very difficult minimization problem, as NN usually have many parameters. There are moreover (provable) many local minima, and a fast procedure that finds the first local minimum is usually not the best.

A NN predictor for Lobith should take the form:

$$H_{\text{Lobith}}(t+1) \stackrel{?}{=} \text{NN} (H_{\text{Lobith}}(t), H_{\text{Köln}}(t), H_{\text{Köln}}(t-2), H_{\text{Plochingen}}(t-2), \dots)$$

## 7 Kernel regression

Probabilistic generalization start by fitting non-Gaussian densities to the data (see figure 10). In this section, we discuss the use of so called Mixtures of Gaussians (MG for short) for this purpose. MG's are formally defined by:

$$f(x) = \sum_{i=1}^{N_c} w_i f_{\mathcal{N}(\mu_i, \Sigma_i)}(x) \tag{4}$$

$$f_{\mathcal{N}(\mu, \Sigma)}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp \left[ -\frac{(\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)}{2} \right]$$

2. In the NN literature this is called *supervised learning*.

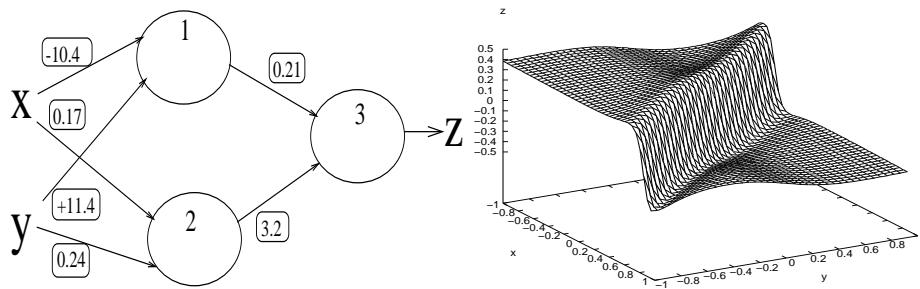


Figure 8: A NN consisting of several neurons combining (by neuron 3) the non-linearity of neuron 1 (see figure 6) and the almost linear relationship of neuron 2 (see figure 7).

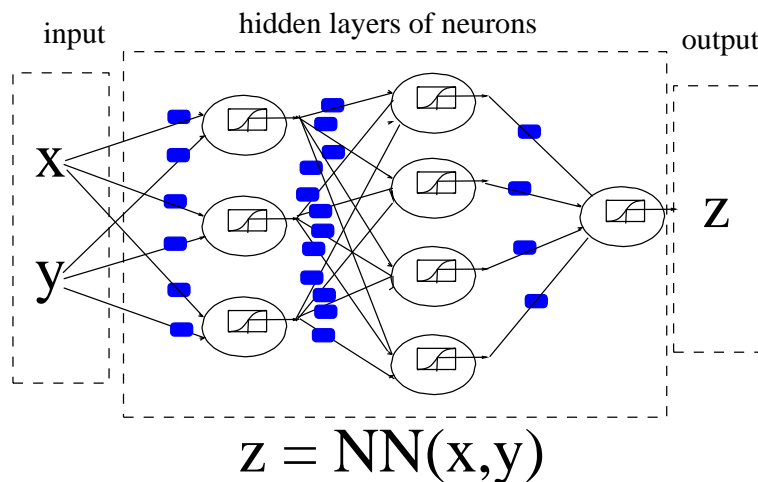


Figure 9: A general NN with two inputs and one output. The internal neurons are organized in so called hidden layers. A weight (blue rectangle in the figure) is associated with each connection. These weights are the parameters of the model and should be found by fitting.

Figure 11 illustrates this definition.

Fitting a MG to an ensemble means fitting weights  $w$ , means  $\mu$  and covariances  $\Sigma$  of the components. For this, many algorithms are available (see e.g. [FJ02]), most based on maximum likelihood resulting in an E(xpectation)M(aximization)-type of algorithm (see [MK97]).

Densities of the MG type inherit a lot of good properties from their Gaussian components: calculation is easy and fast, marginal and conditional densities are again MG's and can be calculated analytically, (Monte Carlo) simulation is extremely easy and fast.

Conditional densities and moments are the key to kernel regression, as illustrated by figure 12.

There is however a price attached to the use of MG's:

- They usually require a large calibration time, as there are many parameters involved (the  $w$ , means  $\mu$  and covariances  $\Sigma$  of formula (4))
- There rests a so called "curse of dimensionality" on fitting probabilities in high dimensions (see section 2.5 in [HTF01] for a discussion). The result is that kernel regression in cases with very many inputs is difficult.

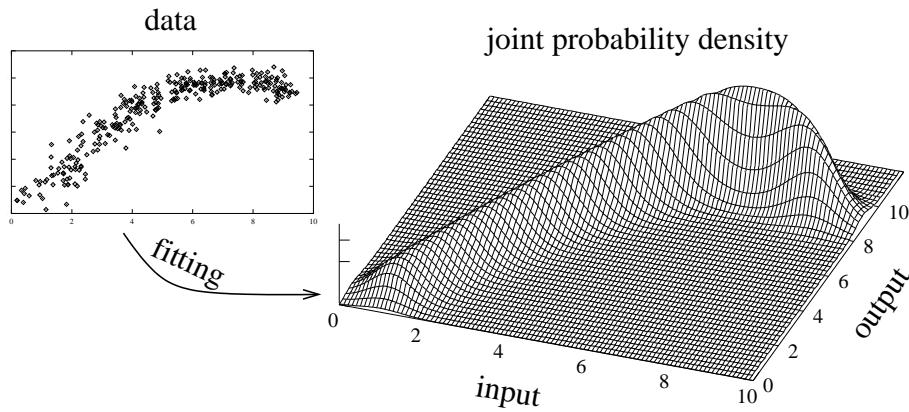


Figure 10: Data (top left) and a MG density fitted to these data (bottom left).

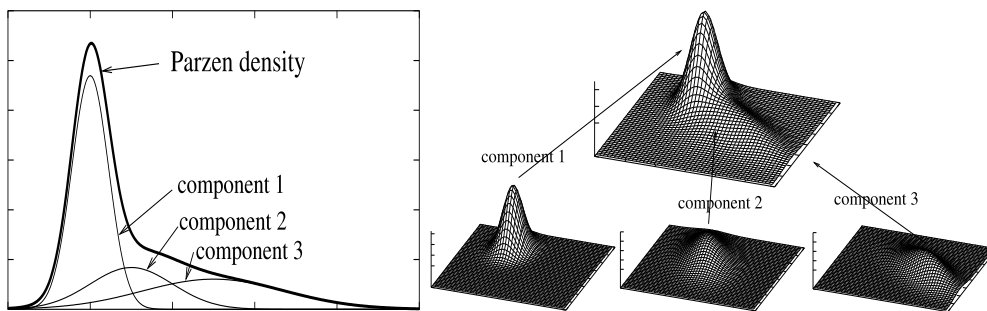


Figure 11: Mixtures of Gaussians. Left a one dimensional example, right a two dimensional.

- There is (certainly in high dimensions) a danger of over-fitting.
- New parameters emerge, as e.g. the number of components of the MG.

Despite this price, it is still promising to try to use MG. One of the possible benefits could be that, even if the “mean” prediction could not be improved, the conditional densities offer an interpretation of the uncertainty around this mean (see figure 13).

## 8 Non-linear regression techniques for the prediction of Lobith

It is the experience of the author that **for the prediction of Lobith** the new techniques do not improve significantly the linear results.

One of the reasons for this failure is certainly that the Rhine is a large river, resulting in a near to linear behavior, as illustrated by figure 14.

One of the major problems however in trying to apply these non-linear techniques is that of the input selection. Input selection is even more important than for the linear case as –due to the many parameters involved– the danger of over-fitting is greater.

As fitting one non-linear model takes considerable more time than fitting a linear one, a brute force approach as trying every possible selection is even more out of order than in the linear case.

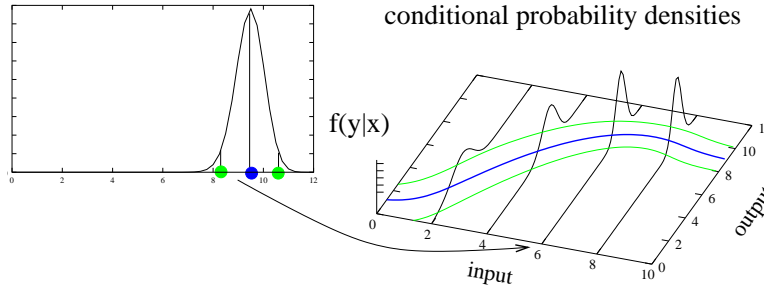


Figure 12: Characteristic results of kernel regression: conditional densities and conditional moments (the MG of figure 10 was used).

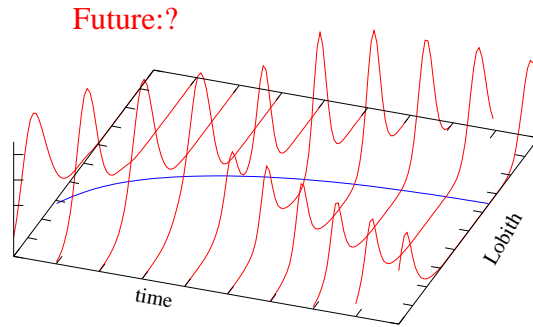


Figure 13: Possible benefit of using kernel regression by MG's: interpretation of uncertainty by means of the conditional densities (in red) around the mean (blue line).

Generally accepted techniques for non-linear input selection are not yet available (see e.g. [DK96] for a discussion of the technique of principal components in the framework of NN's) or take even more computation time than the brute force approach.

Using the inputs selected by linear techniques is frustrating, as one may expect that these are not selected for their “non-linear” information content.

For the probabilistic approach, one of the key issues seems to be the generalization of “correlation”. Probably the most promising is mutual information, defined by (see [Mac03]):

$$I(X; Y) = \int dx \int dy f_{X,Y}(x, y) \log \left( \frac{f_{X,Y}(x, y)}{f_X(x)f_Y(y)} \right) \quad (5)$$

where  $f_{X,Y}$  is the joint probability density of the stochastic variables  $X$  and  $Y$  and  $f_X$  and  $f_Y$  their marginals. The calculation of this measure of non-linear dependency is however rather difficult, and requires careful density fitting.

## 9 Improving residual analysis

A more promising approach seems to be residual analysis by non-linear techniques. We illustrate this by the analysis of the residuals of the MLR model of equation(1), although residuals of other, e.g. more physical based prediction methods, could be also be investigated.

Figure 15 shows these residuals plotted vs the water level of Lobith at the day of prediction. As the values on the x-axis where part of the regressors, there is no linear information

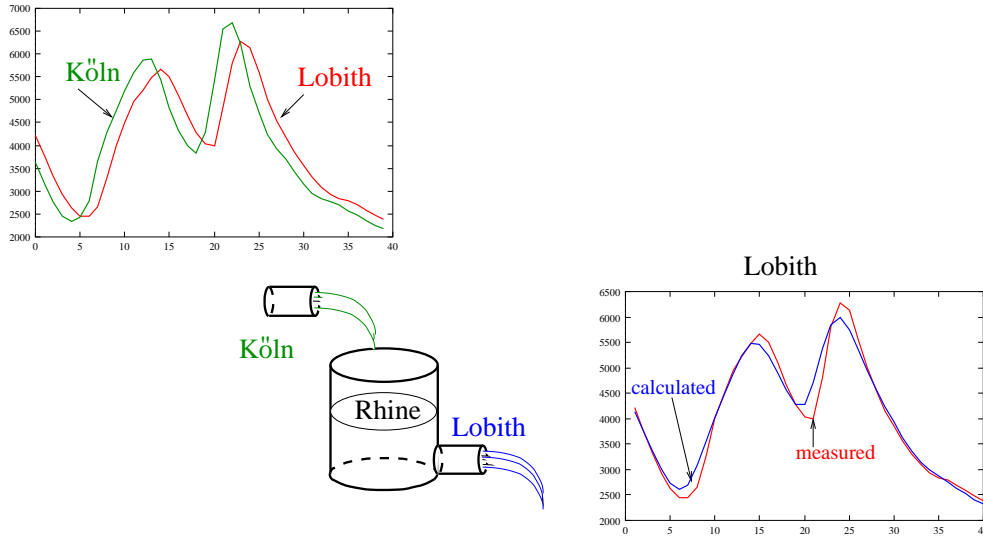


Figure 14: A linear reservoir fitted to the Rhine trajectory Köln-Lobith for the flood wave of 1976. Top left: the (discharge) data. Bottom right: the calculated and measured discharges for Lobith.

left in the x-axis values that may improve the prediction (the correlation is 0). To analyze possible non-linear information content, we start by fitting a MG to the data cloud, as illustrated by figure 16.

This density may be used to assess prediction uncertainty as follows. Assume that at a certain moment  $t$  a water level of 14.10 m is measured at Lobith. We use (as before) formula (1) to predict the mean water level for the next day. To describe the uncertainty of the prediction error, the conditional density –conditioned on the actual value of 14.10 m– of the previously fitted MG is used, as illustrated in figure 17. This density, being e.g. skewed, offers more information than just the standard deviation, which would be the classical MLR way of describing the uncertainty.

Figure 18 shows how these conditional densities do depend on the current conditioning value at Lobith. The bimodality at the far extreme (for  $\text{Lobith}(t)=17\text{m}$ , a value that has never been observed) may be a questionable extrapolation artifact, but the skewness for smaller levels may prove very informative.

## 10 Conclusions

The current MLR prediction models for Lobith are very efficient. The effectiveness of a new non-linear technique (NN, Kernel Regression by MG's) depends heavily on the embedding framework of auxiliary tools (analysis of variance, selection procedures, ...).

Brute force (i.e. try all) does not solve the selection problem for the non-linear models. Universally applicable new techniques to tackle this are not yet available. As distributed modeling and remote sensing use is growing, the selection problem will become even more important in the future.

Subtle non-linear analysis of residuals of existing models may prove to be more valuable than attempts to improve the mean predictions.

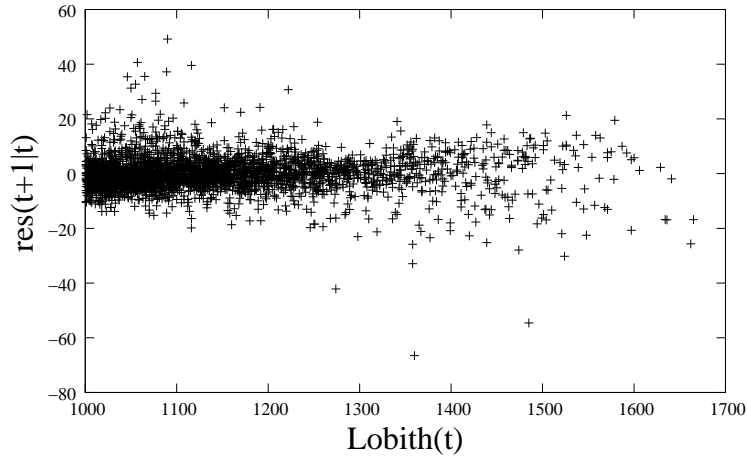


Figure 15: The one day ahead residuals of equation 1 plotted vs the water level of Lobith at the day of prediction

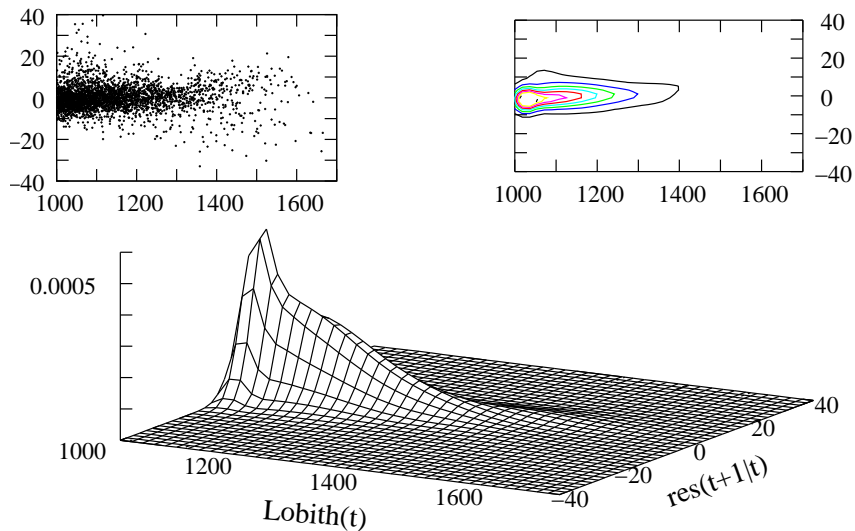


Figure 16: A MG fitted to the data of figure 15. Top left: the data cloud, bottom: the fitted MG, top right: a contour plot of the MG.

## References

- [BD59] G. Box and N. Draper. A basis for the selection of a response surface design. *Journal of the American Statistical Association*, 54:622–654, 1959.
- [Bis98] C. Bishop. *Neural Networks for Pattern Recognition*. Claridon Press, Oxford, 1998.
- [DK96] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks*. John Wiley & Sons, 1996.
- [FJ02] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *IEEE T. Pattern Ana*, 23(3):381–396, 2002.
- [HTF01] T. Hastie, R. Tibshiriani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [Mac03] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- [MK97] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. New York: Wiley, 1997.

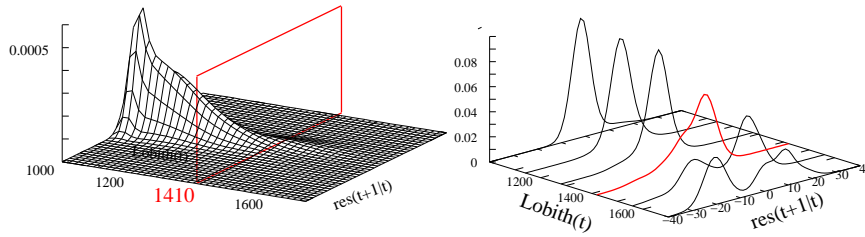


Figure 17: Taking the conditional density of the MG of figure 16 to improve the uncertainty analysis of the prediction

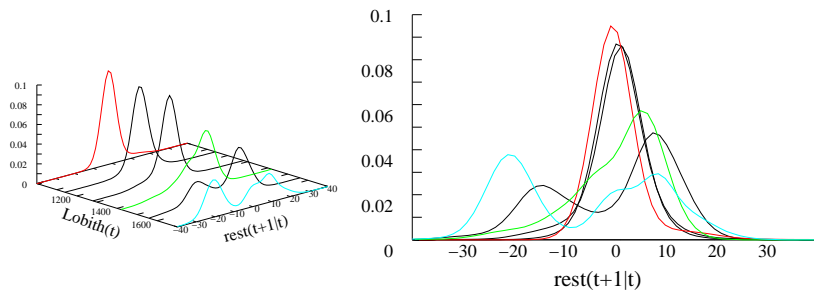


Figure 18: An overview of conditinal densities of the MG of figure 16. The differences in these conditinal densities proves that the water level of Lobith at time  $t$  contains valuable information about the uncertainty of  $\text{res}(t + 1|t)$ .

- [PS97] B. Parmet and E Sprokkereef. Heralibratie Model Lobith. Technical report, RIZA rapport 97.061, 1997.
- [Spr01] E. Sprokkereef. Extension of the Flood Forecasting Model FLORIJN. Technical report, NCR publication No 12-2001, 2001.