LAG SPACE ESTIMATION IN TIME SERIES MODELLING

Cyril Goutte

Department for Mathematical Modelling
Technical University of Denmark, building 321 – DK-2800 Lyngby, Denmark
cg@imm.dtu.dk

ABSTRACT

The purpose of this contribution is to investigate some techniques for finding the relevant lag-space, i.e. input information, for time series modelling. This is an important aspect of time series modelling, as it conditions the design of the model through the regressor vector a.k.a. input layer in a neural network. We give a rough description of the problem, insist on the concept of generalisation, and propose a generalisation-based method. We compare it to a non-parametric test, and carry out experiments, both on the well-known Hénon map, and on a real data set.

1. INTRODUCTION

Let us assume that a time series is obtained from a mapping $X_t = f(X_{t-u_1}, X_{t-u_2}, \ldots, X_{t-u_m})$. The m delays can include long term dependencies, in order to take into account e.g. some seasonality. The (u_i) are the *primary dependencies*, the smallest set of sufficient, not necessarily consecutive delays. All other dependencies are obtained through a combination of mappings and are dubbed *higher order* dependencies.

The use of higher order dependencies in the modelling process leads to possibly over-parameterised, and thus less efficient, models. It is therefore of importance to try and estimate the optimal lag-space, i.e. find the primary dependencies. This allows to minimise the number of parameters and optimises the predictive abilities.

In the following, we recall the concept of generalisation, and introduce a generalisation-based method for estimating the lag-space of time series. We also evoke a non-parametric method for finding the embedding dimension of time series, to which our results will be compared. Experiments are carried out on two problems: a small artificial design inspired from the Hénon map, and a real data set. The results are discussed and future prospects are singled out.

This work was partially supported by a research fellowship from Danmarks Tekniske Universitet

2. GENERALISATION ESTIMATES

Let us consider a model f_w of the time series mapping. x will denote the set of input delays X_{t-u} while y is the output $\widehat{X}_t$. The training set contains N input-output examples sampled from the system. An estimate of the optimal parameters is usually obtained by minimising the average residuals (empirical risk), possibly augmented with a regularisation term:

$$S(w) = \frac{1}{N}\sum_{k=1}^{N}\left(f_w\left(x^{(k)}\right) - y^{(k)}\right)^2 \qquad (1)$$

The performance of the model is the ability to generalise to previously unseen cases, measured by the average risk or generalisation error:

$$G(w) = \int \left(f_w(x) - y\right)^2 p(y,x)\, dx\, dy \qquad (2)$$

In terms of generalisation error, the optimal lag-space is one that minimises $G(w)$. Note that for different sets of inputs, the model, and thus w will differ. The generalisation error is usually impossible to calculate. A crucial issue is thus to estimate G. Common such estimates are provided by cross-validation methods. With some assumptions on the problem, algebraic estimates of the generalisation error [6, 5] offer a handy and computationally efficient alternative [2]. Let us e.g. consider a generalisation of the FPE [1]:

$$\widehat{G}_{\mathrm{FPE}}(\widehat{w}) = \left(\frac{N+\widehat{P}}{N-\widehat{P}}\right) S(\widehat{w}) \qquad (3)$$

where $\widehat{P}$ is the *effective number of parameters*, the calculation of which depends on the regularisation method.

In that context, let us introduce a generalisation-based method for estimating the relevant lag-space: the *naïve generalisation* (NG) method. It consists in selecting the delays that lead to a decrease in estimated generalisation error. In order to avoid delays corresponding to a marginal decrease in error, we introduce a selection parameter α. E.g. if $\alpha = 0.99$, a candidate delay has to outperform the current selection by at least 1%. The algorithm is the following:

1. Initialise: $d = 0$; $G_{min} = \sigma_x^2$; no input selected.
2. Model: $d = d + 1$; add delay $t - d$ to selected inputs; calculate $\widehat{G}$ for resulting model.
3. Test: if $\frac{\widehat{G}}{G_{min}} < \alpha$, select delay $t - d$; $G_{min} = \widehat{G}$. Otherwise discard delay.
4. Goto step 2 until stop condition is reached.

The selection terminates when a *stop condition* is reached. It can relate to e.g. $\widehat{G}$ or the maximum admissible delay.

3. EMBEDDING DIMENSION

The δ-test was introduced by [7] to determine the embedding dimension of time series. It relies on a continuity argument, with a smoothness assumption. It roughly considers that for a well determined input space, close inputs should correspond to close outputs. When the lag-space is lacking some information, close inputs can lead to arbitrarily[1] far outputs due to the effect of the missing delay(s). On the other hand, inclusion of an irrelevant delay can make arbitrarily far inputs (along the dimension corresponding to that delay) correspond to close outputs.

It is an entirely non-parametric test, relying only on the data. It does not need the specification of a model (step 2 above). A detailed presentation of the δ-test is not possible in the space alloted here, and the reader is referred to [7] for a thorough presentation.

4. HÉNON MAP

First we compare both lag-space selection techniques on a modified version of the well-known Hénon map. We generate 1000 data on this chaotic time series for training, as well as a validation set containing 10000 examples, using the expression:

$$X_t = 1 - a.X_{t-2}^2 + b.X_{t-4} \quad (4)$$

with $a = 1,4$ and $b = 0,3$. Delays 2 and 4 have been used here instead of 1 and 2 to check the methods' ability to detect "gaps" in the lag-space. Performing the δ-test on the training set leads to the choice of 2 and 4 as relevant delays (as expected).

[1]in the limit of the data variation.

We have used the naïve generalisation method with two different kinds of estimation: a linear model, using the FPE as a generalisation estimate, and a non-parametric kernel smoother, together with the *leave-one-out* (LOO) cross-validation estimate of generalisation error. The NG method selects all even delays from 2 to 12 in the linear case, and delays 2 and 4 for the kernel smoother, leading to the following performance:

Non-noisy map	#inp	$S(\widehat{w})$	$\widehat{G}(\widehat{w})$	Valid.
Linear (NG)	6	0,361	0,380	0,365
Linear (δ)	2	0,429	0,431	0,454
Kernel (NG)	2	0,000	0,000	0,000
Kernel (δ)	2	0,000	0,000	0,000

The $\widehat{G}(\widehat{w})$ column displays the generalisation estimate (either FPE or LOO), and the last column contains the mean squared error measured on the validation set of 10000 elements.

We also perform some experiments adding Gaussian noise with $\sigma = 0.2$ on the training data. In that case, NG selects one additional delay for the linear model $(t - 20)$ as well as for the kernel smoother $(t - 6)$. Performance is shown in the following table, where the validation error is calculated on non-noisy data:

Noisy data	#inp	$S(\widehat{w})$	$\widehat{G}(\widehat{w})$	Valid.
Linear (NG)	7	0,408	0,414	0,383
Linear (δ)	2	0,475	0,477	0,453
Kernel (NG)	3	0,095	0,117	0,025
Kernel (δ)	2	0,145	0,158	0,027

These experiments suggest a useful feature of NG: when the model is insufficient (linear model), it selects additional delays. It thus yields significantly better generalisation performance than the "optimal" set of delays.

5. REAL DATA EXPERIMENTS

We will now attempt to estimate the lag-space of a real time series for which this information is unknown. The data contains the mean monthly flow of the Fraser river at Hope (British Columbia), from March 1913 to December 1990, amounting to 946 measurements[2]. Figure 1 displays the time series between June 1921 and October 1929. Predictably, it exhibits a roughly periodic feature, reaching maxima every 11 to 13 month (mostly in June). Otherwise, the behaviour seems chaotic.

We use only 315 (one third) of the available data to estimate the lag-space, leaving the rest (631) as validation set to provide an empirical estimate of the generalisation error. Furthermore, the estimation will be

[2]Data set available from statlib in the `datasets` directory.

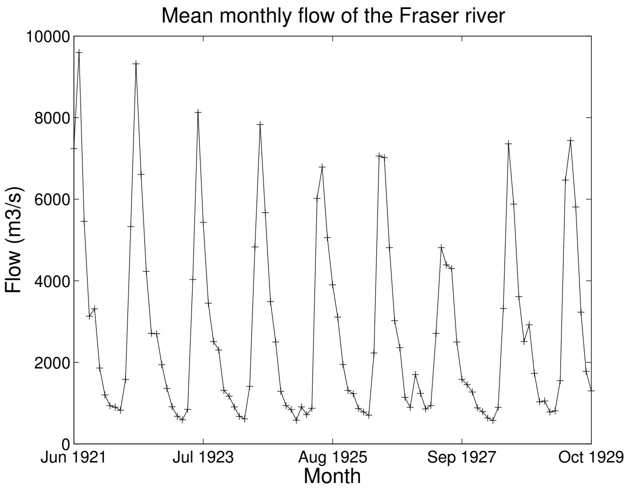

Figure 1: Mean monthly flow of the Fraser River at Hope (B.C.). June 1921 is data no. 100.

performed on log-values of the data, which possess a better distribution than the raw data.

Performing a δ-test on the Fraser river data leads us to select delays 1, 2, 4, 7, 8 and 11 as relevant inputs.

Linear modelling

First we consider a linear model. The following delays lead to a significant decrease in generalisation error: 1, 2, 4–7, 10–13, 16, 23–26, 35, 43, 48 and 49. The NG method thus produces a model with 19 parameters.

Please note that as the maximum delay d increases, the number of available examples decreases. With our 315-long sequence, we produce 314 training examples with $d = 1$, but only 285 when $d = 30$. The number of examples in (3) is $N = 315 - d$, so that the ratio $\frac{N+P}{N-P}$ grows as d increases.

Refer to section 6 for detailed results.

Non-linear modelling

We address the issue of non-linear modelling through the use of neural networks models. These models have been applied extensively in the past few years, in many fields including signal processing. We will here consider a standard multi-layered perceptrons model with n inputs, one hidden layer containing N_h cells and one output:

$$f_w(x) = \sum_{j=1}^{N_h} W_j.h\left(\sum_{i=1}^{n} w_{ji}.x_i + w_{j0}\right) + W_0 \qquad (5)$$

where $h(\cdot)$ is the—usually sigmoïd—transfer function. w_{ij} is the weight of the connection from input i to hidden cell j and W_j the weight of the connection from hidden cell j to the output. W_0 and the w_{j0} are the *biases* of the model.

The model is identified by minimising (1) added with a weight decay for regularisation purposes. The count of the *number of effective parameters* is done as in [6]. It is extremely important to get an accurate estimate of $\widehat{P}$ as the total number of parameters in multi-layered perceptrons grows rapidly with the size of the input: the total number of parameter is $P = (n+2)\,N_h + 1$. If we limit the number of hidden cells to 5, a neural network with 12 inputs contains 71 parameters, which compares unfavourably with the 303 training patterns available.

Using $N_h = 5$ hidden units, the NG method estimates that the relevant inputs are delays 1-4, 7, 10, 11 and 23, resulting in a 56 parameters network. On the other hand, using the embedding dimension information, the network is limited to 6 inputs and 41 parameters.

Non-parametric modelling

All results are compared to a non-parametric modelling technique: a kernel smoother using a Gaussian kernel shape[3] [3]. Generalisation is assessed by the leave-one-out cross-validation estimate, which is also used to tune the kernel size. The NG method selects a total of 14 relevant delays: 1–4, 6–8, 10, 11, 13, 19, 24, 26 and 27.

6. RESULTS AND DISCUSSION

Results for all models and both lag-space estimation schemes are gathered in the following table for comparison. The predictions provided by the linear and non-linear models (using the NG method) are displayed on figure 2. Notice that the first peak is poorly predicted by the linear model, and almost perfectly by the neural network. Otherwise, both predictions are very close, as indicated by the similar generalisation error scores.

Model	#inp.	$S(\widehat{w})$	$\widehat{G}(\widehat{w})$	Valid.
Linear (NG)	19	0,0449	0,0518	0,0444
Linear (δ)	6	0,0696	0,0724	0,0618
Kernel (NG)	14	0,0220	0,0635	0,0566
Kernel (δ)	6	0,0432	0,0693	0,0537
Neur.net (NG)	9	0,0381	0,0562	0,0439
Neur.net (δ)	6	0,0490	0,0643	0,0487

From these results we see that though NG is a rather coarse method, it outperforms the δ-test for both linear and non-linear modelling. On the other hand, NG

[3] simulations carried out with other kernel shapes lead to similar performance.

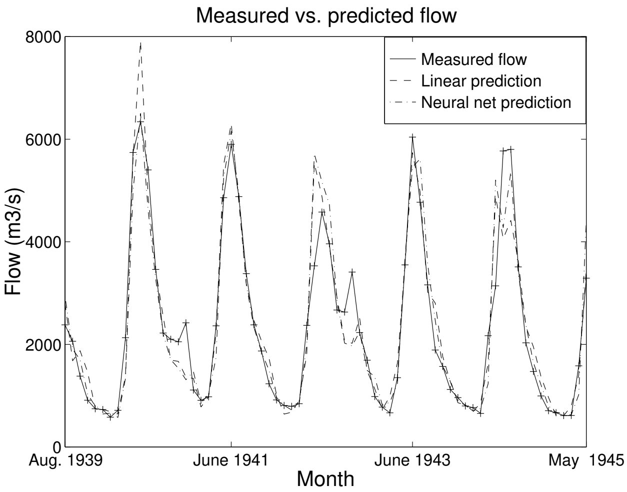

Figure 2: Time series and prediction for two models.

selects more inputs, leading to a higher number of parameters. For comparison, let us mention that the first 6 parameters selected by NG with the linear model yield a performance (both in training and generalisation) around 0, 1.

Neural networks are an interesting alternative, providing good results, i.e. low generalisation error, for a reduced number of inputs. This is indeed expected, given their non-linear nature, and universal approximation properties.

In all these experiments, we notice that the generalisation estimates are often over-estimated, but they manage to "keep the information" i.e. low estimates correspond to low generalisation error, as long as we restrict the comparison to the same kind of estimates and the same class of models.

Furthermore, the experiments give several insights into lag-space estimation for time series modelling:

1. The δ-test yields homogeneous results and depends only on the data. The specification of a model is actually not even necessary.

2. On the other hand, the NG method is model dependent. It has to be applied for each model and can prove really time-consuming.

3. In our experiments, the NG method tends to select more delays, further in the past, as long as the (estimated) generalisation error decreases.

4. The non-parametric δ-test needs a large amount of data to provide reliable results. On the other hand, the rougher NG uses the available data to probe further into the lag-space (e.g. delay 49 for the linear model).

5. The *naïve generalisation* method is a typical *forward selection* procedure [4]. Performing a *backward elimination* step along the same lines on the set of inputs selected for the Neural Networks, we realise that deleting inputs 4, 7, 10 and 23 actually leads to a decrease in (estimated) generalisation error. The resulting neural network has only 5 inputs, and 0, 0423 / 0, 0542 / 0, 0425 as training, estimated generalisation and validation performance (respectively).

The main prospect for future work is linked to the treatment of relevance in the NG method. Here we check this relevance by simply comparing the generalisation estimates, using α as a "level of significance". The use of statistical tests for checking this relevance is an obvious improvement to this method. Work along this line is in progress and will be the object of a future communication.

Acknowledgments

We wish to acknowledge valuable discussions on model order and lag space estimation with Patrick Gallinari, Magnus Nørgaard, and the learning group at IMM.

7. REFERENCES

[1] H. Akaike. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21:243–247, 1969.

[2] C. Goutte. Some computational complexity aspects of neural network training. Unpublished manuscript, February 1996.

[3] W. Härdle. *Applied nonparametric regression*. Number 19 in Econometric Society Monographs. Cambridge University Press, 1990.

[4] R. R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49, March 1976.

[5] J. Larsen and L. K. Hansen. Generalized performance of regularized neural networks models. In J. Vlontzos, J. N. Hwang, and E. Wilson, editors, *Neural Networks for Signal Processing – Proceedings of the 1994 IEEE Workshop*, number IV in NNSP, pages 42–51, Piscataway, New Jersey, 1994. IEEE.

[6] J. Moody. Note on generalization, regularization and architecture selection in nonlinear learning systems. In B. H. Juang, S. Y. Kung, and C. A. Kamm, editors, *Proceedings of the first IEEE Workshop on Neural Networks for Signal Processing*, number I in NNSP, pages 1–10, Piscataway, New Jersey, 1991. IEEE.

[7] H. Pi and C. Peterson. Finding the embedding dimension and variable dependences in time series. *Neural Computation*, 6(3):509–520, May 1994.