# Input Selection for Long-Term Prediction of Time Series

Jarkko Tikka, Jaakko Hollmén, and Amaury Lendasse

Helsinki University of Technology, Laboratory of Computer and Information Science, P.O. Box 5400, FI-02015 HUT, Finland
tikka@mail.cis.hut.fi
http://www.cis.hut.fi/tikka

**Abstract.** Prediction of time series is an important problem in many areas of science and engineering. Extending the horizon of predictions further to the future is the challenging and difficult task of long-term prediction. In this paper, we investigate the problem of selecting non-contiguous input variables for an autoregressive prediction model in order to improve the prediction ability. We present an algorithm in the spirit of backward selection which removes variables sequentially from the prediction models based on the significance of the individual regressors. We successfully test the algorithm with a non-linear system by selecting inputs with a linear model and finally train a non-linear predictor with the selected variables on Santa Fe laser data set.

## 1 Introduction

Time series prediction plays an important role in analysis of many problems encountered in science and engineering, for instance in climatology [1], ecology [2], electricity production [3], and economics [4].

Long-term prediction is difficult and time-consuming, since it extends the horizon of prediction further to the future, adding significant uncertainty in the prediction task. We address the problem of input selection for the purpose of long-term prediction. We present an algorithm for input variable selection in the spirit of backward selection, in which variables are progressively removed from the autoregressive prediction model. The removal of variables is based on a median and a standard deviation of parameter distributions sampled with a bootstrap resampling procedure [5]. These statistics reflect the uncertainty for a variable to play an important role in the prediction task. We apply the algorithm in a long-term prediction setting, where input selection is performed for different $k$-step-ahead prediction models. Whereas the input selection is accomplished with the linear autoregressive models, the final model is a non-linear predictor used in the prediction of a prominent non-linear benchmark (Santa Fe laser data set [6]).

## 2 Resampling Using Bootstrap

The bootstrap is a statistical resampling method represented by Efron *et al.* in [5]. The idea of bootstrap is to sample original data and to estimate some statistics from these resampled sets. No assumptions are made about the forms of probability distributions in the bootstrap method. The statistic of interest and its distribution are computed by resampling the original data with replacement.

In this article the bootstrap is applied in context of the linear autoregressive process AR($l$). It can be represented by the following equation

$$y_t = \sum_{i=1}^{l} \alpha_i y_{t-i} + \epsilon_t \; . \tag{1}$$

It is assumed that $\epsilon_t$ is a sequence of independently normally distributed random noise with zero mean and common finite variance. Let us assume that $N + l$ observations are available. Equation (1) can be written in the matrix form as follows

$$\boldsymbol{y} = \boldsymbol{Y}\boldsymbol{\alpha} + \boldsymbol{\epsilon} \; , \tag{2}$$

where $\boldsymbol{y} = (y_{l+1}, \ldots, y_{l+N})^T$, $\boldsymbol{\epsilon} = (\epsilon_{l+1}, \ldots, \epsilon_{l+N})^T$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_l)^T$ and

$$\boldsymbol{Y}_{(N \times l)} = \begin{bmatrix} y_l & y_{l-1} & \cdots & y_1 \\ y_{l+1} & y_l & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{l+N-1} & y_{l+N-2} & \cdots & y_N \end{bmatrix} . \tag{3}$$

The parameters $\boldsymbol{\alpha}$ are estimated by minimizing the mean square error (MSE), i.e., $(1/N)\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 = (1/N)\|\boldsymbol{y} - \boldsymbol{Y}\hat{\boldsymbol{\alpha}}\|^2$ where $\|.\|$ is the $L_2$-norm. The least squares estimates of parameters are $\hat{\boldsymbol{\alpha}} = (\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{y}$. Here we assume that the data are normalized to zero mean and unit variance and, thus there is no need for the constant term in Equations (1) and (2). Due to the normalization the upper limit for the MSE is 1, which is obtained by the mean predictor.

Bootstrapping the regression model can be done in two different ways and both could also be applied in this case. The methods are bootstrapping residuals and bootstrapping pairs [5]. The regressors i.e. the columns of the design matrix $\boldsymbol{Y}$ are treated as fixed quantities in the bootstrapping residuals approach. That assumption is strong and it can fail even if Equation (2) for the AR($l$) process is correct. In the bootstrapping pairs approach weaker assumptions about validity of Equation (2) are made.

In the bootstrapping pairs, $\hat{P}$ is assumed to be an empirical distribution of the observed data pairs $(y_{l+t}; y_{l+t-1}, y_{l+t-2}, \ldots, y_t) = (y_{l+t}; \boldsymbol{Y}_{l+t})$, where $t = 1, \ldots, N$ and $y_{l+t}$ and $\boldsymbol{Y}_{l+t} = [y_{l+t-1}, y_{l+t-2}, \ldots, y_t]$ are values of the target and the regressors, respectively. In other words, $y_{l+t}$ is one element of the vector $\boldsymbol{y}$ and $\boldsymbol{Y}_{l+t}$ is the corresponding row from the matrix $\boldsymbol{Y}$. $\hat{P}$ puts probability mass of $1/N$ on each pair $(y_{l+t}; \boldsymbol{Y}_{l+t})$. A bootstrap sample is a random sample of size $N$ drawn with replacement from the population of $N$ pairs $(y_{l+t}; \boldsymbol{Y}_{l+t})$. $B$ independent bootstrap samples $(\boldsymbol{y}^{*j}, \boldsymbol{Y}^{*j}), j = 1, \ldots, B$, of size $N$ are drawn with replacement from the distribution $\hat{P}$. Therefore, some data pairs from the original data $(\boldsymbol{y}, \boldsymbol{Y})$ can appear zero times or once or twice etc. in the bootstrap sample $(\boldsymbol{y}^{*j}, \boldsymbol{Y}^{*j})$.

The bootstrap replications $\hat{\boldsymbol{\alpha}}^{*j}$ of the estimates of parameters $\hat{\boldsymbol{\alpha}}$ are $\hat{\boldsymbol{\alpha}}^{*j} = (\boldsymbol{Y}^{*j^T}\boldsymbol{Y}^{*j})^{-1}\boldsymbol{Y}^{*j^T}\boldsymbol{y}^{*j}$. $B$ bootstrap replications $\hat{\alpha}_i^{*j}$ form a distribution of the estimates of parameters $\hat{\alpha}_i$. This distribution of the bootstrap replications can be used in an evaluation of the significance of parameters. In this paper, the evaluation is based on medians and standard deviations of the distributions.

## 3 Input Selection

The usual task is to define the order $l$ in Equation (1) and use all the inputs $y_{t-i}, i = 1, \ldots, l$ in the prediction of $y_t$. This kind of solution is referred to ordinary least squares (OLS) solution later on. The OLS solution is not always satisfactory. The OLS estimates have low bias but a large variance, which may diminish the prediction accuracy. The prediction accuracy can sometimes be improved by shrinking some estimates of parameters toward zero or setting them exactly to zero [7].

In this paper, we present a methodology to select a subset of inputs that have the strongest explanatory power in the AR process. This also makes the interpretation of the dependencies in time series easier.

First, the maximum number of inputs have to be set, i.e., to define the value of $l$. It can be selected to be a relatively large to ensure that the final order of the estimated AR process can be large enough. The procedure continues by estimating AR process using all the inputs $y_{t-i}, i = 1, \ldots, l$ that is calculating the OLS estimates of parameters $\hat{\alpha}_i$. The distributions of parameters $\hat{\alpha}_i$ and the standard deviation $s$ of the training MSE are estimated using the bootstrapping pairs method. In addition, the validation error is calculated using the parameters $\hat{\alpha}_i$. The validation error is the MSE for the validation set, i.e., for data that are not used in the training phase.

The next step is to delete the least significant regressor. The median value $m_i$ and the standard deviation $\sigma_i$ of parameter $\hat{\alpha}_i$ are calculated from the bootstrap replications $\hat{\alpha}_i^{*j}$. The ratio $|m_i|/\sigma_i$ is used as a measure of significance of the corresponding parameter. The parameter or regressor with the smallest ratio is pruned from the set of possible inputs. After that, the bootstrapping using the remaining inputs and pruning is repeated as long as there are variables left in the set of inputs.

The quantity $d_i = (\sum_{j=1}^{B}(m_i - \hat{\alpha}_i^{*j})^2)^{\frac{1}{2}}/(B-1)$ could be used instead of the standard deviation as a measure of the width of the distribution of the estimates of parameters. Another alternative is to estimate the width of the distribution using the difference $\Delta_i = \hat{\alpha}_{i,high}^{j*} - \hat{\alpha}_{i,low}^{j*}$ , where $\hat{\alpha}_{i,high}^{j*}$ is $B \cdot (1-q)$th and $\hat{\alpha}_{i,low}^{j*}$ is $B \cdot q$th value in the ordered list of the $B$ replications of $\hat{\alpha}_i^{*j}$ and $q$ could be for instance 0.165.

The Algorithm 1 produces sequentially $l$ different models. The computational complexity of the proposed algorithm is $\mathcal{O}(l)$ whereas a backward selection type of algorithm would be of complexity $\mathcal{O}(l^2)$ and the full, exhaustive (brute-force) search of all possible variable configurations would take $\mathcal{O}(2^l)$. The purpose is to select a model which is as sparse as possible in the sense that it includes only a

**Algorithm 1** Algorithm for input selection

```
1: Let $\mathcal{L}$ be the set of indices of the inputs. In the beginning, $\mathcal{L}$ includes all the indices
   $i$, $y_{t-i}, i = 1, \dots, l$.
2: if $i \in \mathcal{L}$ estimate $B$ bootstrap replications $\hat{\alpha}_i^{j*}, j = 1, \dots B$ of the parameters $\hat{\alpha}_i$ by
   minimizing MSE using the training data, otherwise set $\hat{\alpha}_i = 0, i \notin \mathcal{L}$
3: Compute the mean and the standard deviation $s_{\mathcal{L}}$ of the bootstrap replications of
   the training MSEs.
4: Estimate parameters $\hat{\alpha}_i$ using the training data and compute the validation error
   $E_v$ using the estimated model.
5: Compute the median $m_i$ and the standard deviation $\sigma_i$ for all the estimates $\hat{\alpha}_i, i \in$
   $\mathcal{L}$ from the bootstrap replications $\hat{\alpha}_i^{j*}$.
6: Compute the ratio $|m_i|/\sigma_i, \forall i \in \mathcal{L}$ and delete the index with the smallest ratio
   from the set $\mathcal{L}$.
7: if $\mathcal{L} \neq \varnothing$ go to step 2, otherwise go to step 8
8: Find the minimum validation error $E_v^{min}$. Sum $E_v^{min}$ and corresponding $s_{\mathcal{L}}^{min}$. Find
   the smallest set of indices $\mathcal{L}_f$ i.e. the least complex model whose validation error
   is under the previous sum. The inputs corresponding to the indices in the set $\mathcal{L}_f$
   are included to the final model.
```

few input variables. Goals of this approach are to avoid the curse of dimensionality, over-parameterization, and overfitting for the non-linear modeling phase. On the other hand, the prediction accuracy should not decrease significantly. The initial model is selected based on the minimum validation error $E_v^{min}$. The corresponding standard deviation $s^{min}$ of training error is also used as the estimate of standard deviation of $E_v^{min}$. The final model is the least complex model whose validation error is under the sum $E_v^{min} + s^{min}$. The procedure is listed step by step in detail in Algorithm 1.

The ratio $|m_i|/\sigma_i$ can be seen as a $Z$-score test statistic to test the hypothesis [8] that a parameter $\hat{\alpha}_i = 0$. The statistics are estimated using bootstrap, so there are not made any assumptions about the probability distributions of parameters. This applies especially when the median and the difference $\Delta_i$ would be used. We use the median instead of the mean as the location parameter of the distribution, since it is a reasonable estimate for skewed distributions, and distributions with outliers. Furthermore, median is the same as the mean, if the distribution is symmetric. The standard deviation and deviation based on the median are reasonable measures for the symmetric distributions, whereas the difference $\Delta_i$ describes better the width of an asymmetric distribution. In the case of a symmetric distribution, for instance a Gaussian distribution, $\sigma_i$, $d_i$, and $\Delta_i$ produce equal estimates of the width of the distributions.

In addition, the ratio can be considered as a signal-to-noise ratio. If the ratio is small then the bootstrap replications of parameters vary strongly around the median and the importance of the corresponding input in the prediction is insignificant. The idea of proposed algorithm is similar to the traditional backward stepwise selection, which sequentially deletes inputs based on the $F$-statistics [8].

Algorithm 1 can also be applied to long-term prediction. In that case the target is $y_{t+k}, k > 0$ in the Equation (1).

## 4 Long-Term Prediction

In the literature, two techniques are used for long-term prediction [6]. In recurrent prediction — the first of the two approaches — only one model is built, that is, $\hat{y}_t = f(y_{t-1}, \ldots, y_{t-l})$. The same model is used repeatedly to obtain the long-term prediction. For instance, for the two-step-prediction the model $\hat{y}_{t+1} = f(\hat{y}_t, y_{t-1}, \ldots, y_{t-l+1})$ is used [9]. The main advantage of this method is that only one model has to be built, the disadvantage being that the quality of predictions is rapidly decreasing with respect to the increasing horizon of prediction. Therefore, in the context of long-term prediction, we focus on the models of the form $\hat{y}_{t+k} = f_k(y_{t-1}, \ldots, y_{t-l})$, which is known as the direct predictor. Because the computational load of this method, i.e., building different models up to the maximum prediction horizon $K$, we need a fast method for input selection. Moreover, we want to select inputs in a non-contiguous manner, for instance, allowing for the inputs such as $\hat{y}_{t+k} = f_k(y_{t-1}, y_{t-3}, y_{t-8}, y_{t-12})$.

## 5 Experiments on the Santa Fe Laser Data Set

The data set used in the experiments is the Santa Fe laser data set described in [6]. The training set size $N_{tr} = 1000$ is selected to be the same as in the Santa Fe time-series competition [6] (see Figure 1). The validation set used for the selection of the model-orders is chosen to be large ($N_{val} = 4546$) in order to perform well on structure selection, and a large testing set ($N_{test} = 4547$) should ensure realistic statistical estimate of the performances of the results.

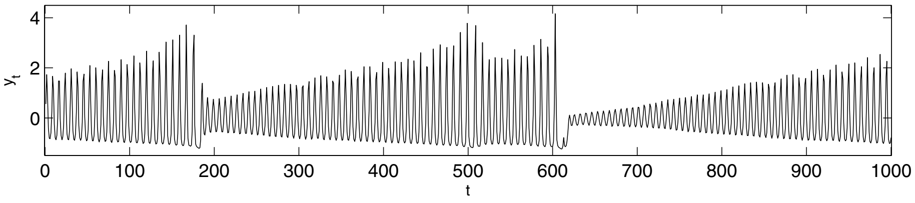

**Fig. 1.** The first thousand samples $N_{tr}$ from Santa Fe laser time series

### 5.1 Linear Models and Input Selection

Input selection Algorithm 1 has been used to yield input variables with a maximum number of inputs $l = 20$ and the maximum prediction horizon being $K = 19$ corresponding to 20 models. The number of bootstrap resamples has been chosen to be $B = 1000$, which is considered to be large enough for reliably estimating the distribution of the parameters in the linear model.

Figure 2 (a) presents an example of input selection in the case of two-step-ahead prediction ($k = +1$). In this case, Algorithm 1 selected the model with 6 inputs. Compared to the model with 18 inputs with the minimum validation

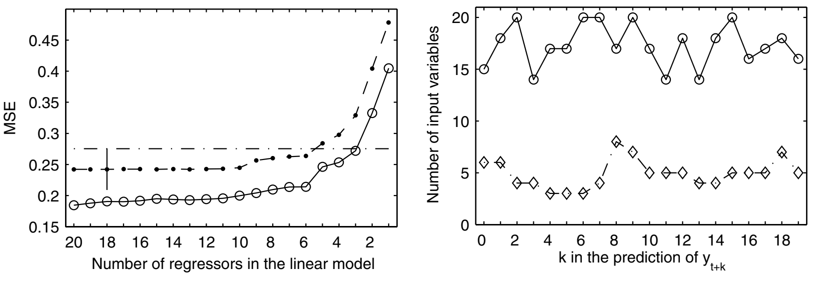

**Fig. 2.** (*left*) An example of input selection, training error (*solid line*), validation error (*dashed line*), vertical line marks the minimum validation error (18 inputs) and a standard deviation of training error, the final model is chosen to be the least complex model (6 inputs) within the standard error (*dash-dotted line*). (*right*) number of selected inputs for each 20 models according to the minimum validation error (*solid line*) and the standard error (*dash-dotted line*)

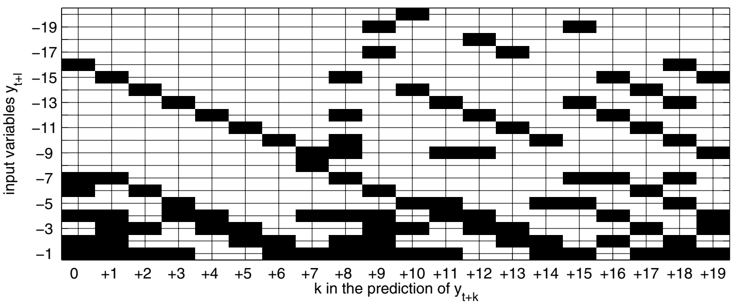

**Fig. 3.** The final models. The targets $y_{t+k}, k = 0, +1, \ldots, +19$ are in the horizontal axis and the possible regressors $y_{t+l}, l = -1, -2, \ldots, -20$ are in the vertical axis. The selected inputs are denoted by black rectangles in each column

error the number of parameters has decreased considerably (by 12). At the same time the validation error has only increased less than one standard deviation of the minimum validation error. In Figure 2 (b), the number of inputs for each 20 models according to the minimum validation error and the final number of selected inputs are shown. The reduction in the number of parameters is on average 12. In Figure 3, the selected inputs for all long-term models are visualized. Parsimonious models are selected based on the proposed algorithm, on average 5 inputs are selected. For instance, the 12-step-ahead model has 5 inputs, marked by 5 black rectangles on the vertical axis at column $k = +11$ ($\hat{y}_{t+11} = f(y_{t-1}, y_{t-4}, y_{t-5}, y_{t-9}, y_{t-13})$).

The final models were obtained using the median $m_i$ and the standard deviation $\sigma_i$ in the input selection. The final models were exactly same as in Figure 3, when the input selection was repeated using $d_i$ and $\Delta_i$ instead of $\sigma_i$. This might indicate that the distributions are approximately normal distributions.

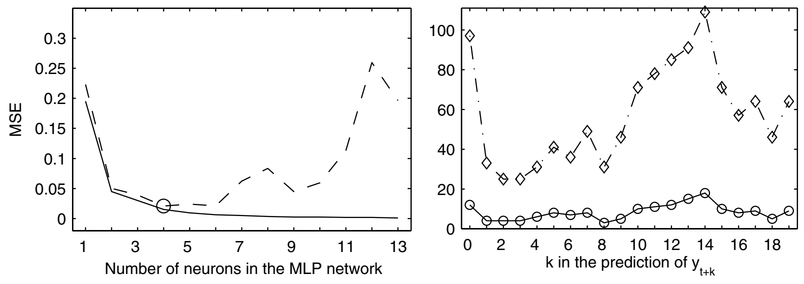

**Fig. 4.** (*left*) An example of the selection of number of neurons $p$ (the target is $y_{t+1}$), training error (*solid line*), validation error (*dashed line*), circle represents minimum validation error. (*right*) the number of neurons $p$ (*solid line*) and the number of parameters (*dash-dotted line*) in each 20 MLP network

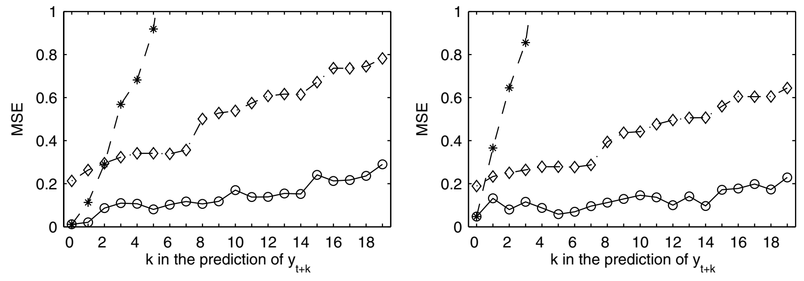

**Fig. 5.** MSE for MLP networks (*solid line*), linear models (*dash-dotted line*) and recurrent prediction (*dashed line*) for both validation (*left panel*) and test sets (*right panel*), respectively

## 5.2 Non-linear, Non-contiguous Regressor Model

Based on the results from the previous section, we train a non-linear model. Here, we have used a multi-layer perceptron (MLP) network [10]. The output of the network is $\hat{y}_t = \mu + \sum_{j=1}^{p} \beta_j \tanh\left(\sum_{i=1}^{l} w_{ji} y_{t-i} + b_j\right)$, where $p$ is the number of neurons in the hidden layer. The network is trained using the Levenberg-Marquardt optimization method by back-propagating the error gradients [10]. Hundred different initializations are used in the training of the models in order to avoid local minima of the training error, and the best model is chosen (for each model-order). The minimum training error (among different initializations) is a monotonically decreasing function with respect to $p$. Validation error is evaluated for the best non-linear model to choose the final number of neurons $p$. The maximum number of tested neurons was 20. An example is shown in Figure 4 (a). The selected number of neurons $p$, which is less than 20 in each network, and the corresponding number of parameters in each 20 MLP networks are shown in Figure 4 (b). The maximum number of parameters in the non-linear models is 109. Because of the parsimony of the selected input set, it is possible to train the model from a small training sample such as ours ($N_{tr} = 1000$).

Figure 5 illustrates the differences between the recurrent and direct prediction for the validation (left panel) and the test (right panel) set up to the maximum prediction horizon $K$. The direct non-linear predictions (solid line) clearly outperform the recurrent approach (dashed line). The direct predictions calculated using the linear models (dash-dotted line) are also better than the recurrent non-linear predictions in the long-term.

## 6 Conclusions

The proposed algorithm selected parsimonious sets of inputs for all long-term prediction models. The non-linear models built using the corresponding inputs led to good prediction performance. Direct long-term prediction is superior to recurrent prediction for the whole prediction horizon. The main advantage of the proposed approach is that it combines fast input selection with accurate but computationally demanding non-linear prediction. Linear complexity of the input selection makes this approach viable for input selection in large-scale problems. Further research include experiments with different data sets and comparisons to different input selection methodologies, e.g. the use of mutual information, to achieve accurate comprehension of the performance of the proposed algorithm.

## References

1. Dal Cin, C., Moens, L., Dierickx, P., Bastin, G., Zech, Y.: An Integrated Approach for Real-time Flood-map Forecasting on the Belgian Meuse River. (Natural Hazards), In press.
2. Sulkava, M., Tikka, J., Hollmén, J.: Sparse Regression for Analyzing the Development of Foliar Nutrient Concentrations in Coniferous Trees. In: Proceedings of the Fourth International Workshop on Environmental Applications of Machine Learning (EAML 2004). (2004) 57-58
3. Lendasse, A., Lee, J., Wertz, V., Verleysen, M.: Forecasting Electricity Consumption Using Nonlinear Projection and Self-organizing Maps. Neurocomputing **48** (2002) 299-311
4. Hamilton, J.D.: Analysis of Time Series Subject to Changes in Regime. Journal of Econometrics **45** (1990) 39-70
5. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC (1993)
6. Weigend, A.S., Gershenfeld, N.A. (eds.): Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley (1994)
7. Copas, J.: Regression, Prediction and Shrinkage. Journal of the Royal Statistical Society (Series B) **45** (1983) 311-354
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning – Data Mining, Inference and Prediction. Springer Series in Statistics. Springer (2001)
9. Tresp, V., Hofmann, R.: Nonlinear Time-series Prediction with Missing and Noisy Data. Neural Computation **10** (1998) 731-747
10. Bishop, C.: Neural Networks in Pattern Recognition. Oxford Press (1996)