Variable Scaling for Time Series Prediction: Application to the ESTSP'07 and the NN3 Forecasting Competitions

Amaury Lendasse and Elia Liitiainen

Abstract— **In this paper, variable selection and variable scaling are used in order to select the best regressor for the problem of time series prediction. Direct prediction methodology is used instead of the classic recursive methodology. Least Squares Support Vector Machines (LS-SVM) and K-NN approximator are used in order to avoid local minimal in the training phase of the model. The global methodology is applied to the ESTSP'07 competition dataset [1] and the dataset B of the NN3 Forecasting Competition [2].**

I. Introduction

Time series forecasting is a challenge in many fields. In finance, experts forecast stock exchange courses or stock market indices; data processing specialists forecast the flow of information on their networks; producers of electricity forecast the load of the following day. The common point to their problems is the following: how can one analyse and use the past to predict the future?

Many techniques exist for the approximation of the underlying process of a time series: linear methods such as ARX, ARMA, etc. [3], and nonlinear ones such as artificial neural networks [4]. In general, these methods try to build a model of the process. The model is then used on the last values of the series to predict the future values. The common difficulty to all the methods is the determination of sufficient and necessary information for an accurate prediction.

A new challenge in the field of time series prediction is the Long-Term Prediction: several steps ahead have to be predicted. Long-Term Prediction has to face growing uncertainties arising from various sources, for instance, accumulation of errors and the lack of information [4].

In this paper, a global methodology to perform direct prediction is presented. It includes variable selection and variable scaling. The variable selection criterion is based on a Nonparametric Noise Estimation (NNE) performed by Delta Test.

In this paper, Least Squares Support Vector Machines (LS-SVM) and K-NN approximator are used as nonlinear models in order to avoid local minima problems [5].

Section 2 presents the prediction strategy for the Long-Term Prediction of Time Series. In Section 3 Delta Test is introduced. Section 4 introduces the variable selection and scaling selection. The LS-SVM model is briefly summarized in Section 5 and K-NN in section 6. Experimental results are shown in Section 7 using the competition datasets.

II. Time Series Prediction

The time series prediction problem is the prediction of future values based on the previous values and the current value of the time series (see Equation 1). The previous values and the current value of the time series are used as inputs for the prediction model. One-step ahead prediction is needed in general and is referred to as Short-Term Prediction. But when multi-step ahead predictions are needed, it is called a Long-Term Prediction problem.

Unlike the Short-Term time series prediction, the Long-Term Prediction is typically faced with growing uncertainties arising from various sources. For instance, the accumulation of errors and the lack of information make the prediction more difficult. In Long-Term Prediction, performing multiple step ahead prediction, there are several alternatives to build models. In the following sections, two variants of prediction strategies are introduced and compared: the Direct and the Recursive Prediction Strategies.

A. Recursive Prediction Strategy

To predict several steps ahead values of a time series, Recursive Strategy seems to be the most intuitive and simple method. It uses the predicted values as known data to predict the next ones. In more detail, the model can be constructed by first making one-step ahead prediction:

$$\hat{y}_{t+1} = f_1(y_t, y_{t-1}, ..., y_{t-M+1}), \tag{1}$$

where M denotes the number inputs. The regressor of the model is defined as the vector of inputs: $y_t, y_{t-1}, ..., y_{t-M+1}$. It is possible to use also exogenous variables as inputs in the regressor, but they are not considered here in order to simplify the notation. Nevertheless, the presented global methodology can also be used with exogenous variables.

To predict the next value, the same model is used:

$$\hat{y}_{t+2} = f_1(\hat{y}_{t+1}, y_t, y_{t-1}, ..., y_{t-M+2}). \tag{2}$$

In Equation 2, the predicted value of $\hat{y}_{t+1}$ is used instead of the true value, which is unknown. Then, for the H-steps ahead prediction, $\hat{y}_{t+2}$ to $\hat{y}_{t+H}$ are predicted iteratively. So, when the regressor length M is larger than H, there are $M - H$ real data in the regressor to predict the H^{th} step. But when H exceeds M, all the inputs are the predicted values. The use of the predicted values as inputs deteriorates the accuracy of the prediction.

Amaury Lendasse and Elia Liitiainen are with the Helsinki University of Technology - Laboratory of Computer and Information Science, P.O. Box 5400, 02015 HUT - Finland (email: lendasse@hut.fi).

B. Direct Prediction Strategy

Another strategy for the Long-Term Prediction is the Direct Strategy. For the H-steps ahead prediction, the model is

$$\hat{y}_{t+h} = f_h(y_t, y_{t-1}, ..., y_{t-M+1}) \text{ with } 1 \leq h \leq H. \tag{3}$$

This strategy estimates H direct models between the regressor (which does not contain any predicted values) and the H outputs. The errors in the predicted values are not accumulated in the next prediction. When all the values, from $\hat{y}_{t+1}$ to $\hat{y}_{t+H}$, need to be predicted, H different models must be built. The direct strategy increases the complexity of the prediction, but more accurate results are achieved.

III. Nonparametric Noise Estimator using the Delta Test

Delta Test (DT) is a technique for estimating the variance of the noise, or the mean square error (MSE), that can be achieved without overfitting [6]. The evaluation of the NNE is done using the DT estimation introduced by Stefansson in [7].

Given N input-output pairs: $(x_i, y_i) \in \mathbb{R}^M \times \mathbb{R}$, the relationship between x_i and y_i can be expressed as:

$$y_i = f(x_i) + r_i, \tag{4}$$

where f is the unknown function and r is the noise. The Delta Test estimates the variance of the noise r.

The DT is useful for evaluating the nonlinear correlation between two random variables, namely, input and output pairs. The DT has been introduced for model selection but also for variable selection: the set of inputs that minimizes the DT is the one that is selected. Indeed, according to the DT, the selected set of variables is the one that represents the relationship between variables and output in the most deterministic way.

DT is based on hypotheses coming from the continuity of the regression function. If two points x and x' are close in the input space, the continuity of regression function implies the outputs $f(x)$ and $f(x')$ will be close enough in the output space. Alternatively, if the corresponding output values are not close in the output space, this is due to the influence of the noise.

Let us denote the first nearest neighbor of the point x_i in the set $\{x_1, \ldots, x_N\}$ by x_{NN}. Then the delta test, δ is defined as:

$$\delta \quad = \quad \frac{1}{2N} \sum_{i=1}^{N} \left| y_{NN(i)} - y_i \right|^2, \tag{5}$$

where $y_{NN(i)}$ is the output of $x_{NN(i)}$. For the proof of the convergence of the Delta Test, see [6].

IV. Variable and Scaling Selection

Variable scaling is a usual preprocessing step in both function approximation and time series analysis. In scaling, weights are used to reflect the relevance of the input variables to the output to be estimated. That is, scaling seeks for redundant inputs and assigns them low weights to reduce the corresponding influence on the learning process. In such a context, it is clear that variable selection is a particular case of scaling: by weighting irrelevant variables by zero we are, indeed, enforcing selection. For the sake of brevity, only the main concepts referring to the regression problem are presented here. Nevertheless, the extension to time series analysis is trivial.

A. Projecting the Input Space with Mahalanobis Matrices

The Mahalanobis distance $d_M(x_i, x_j)$ of two d-dimensional observations x_i, x_j is a proximity (or 'similarity') measure defined on the dependencies between the embedding dimensions. Formally, $d_M(x_i, x_j)$ extends the traditional Euclidean distance $d(x_i, x_j) = [(x_i - x_j)^T(x_i - x_j)]^{1/2}$ transforming the observations subspace by means of a (dd) full-rank matrix M:

$$d(x_i, x_j) = [(x_i - x_j)^T M(x_i - x_j)]^{1/2}, \tag{6}$$

From the previous equation, it is evident that: i) if $M = I$ then the original Euclidean metric is retained, and ii) if M is a (dd) diagonal matrix then the original space is simply rescaled according to the diagonal elements. Matrix M is also symmetric and semi-definite positive, by definition. Moreover, the Mahalanobis matrix M can be factorized as:

$$M = S^T S, \tag{7}$$

with a matrix S that can linearly map the observations into the subspace spanned by the eigenvectors of the transformation. The learned metric in the projection subspace is still the Euclidean distance, that is:

$$\begin{aligned} d(x_i, x_j) = \quad & [(x_i - x_j)^T M(x_i - x_j)]^{1/2} \\ & = [(Sx_i - Sx_j)^T(Sx_i - Sx_j)]^{1/2}, \end{aligned} \tag{8}$$

where, by restricting S to be a non-square ($s * d$, with $s < d$) matrix, the transformation performs both a reduction of the dimensionality and the scaling of the original input subspace. The resulting subspace has an induced global metric of lower rank suitable for reducing the 'curse of dimensionality'. A particular case of Mahalanobis matrix selection is the scaling selection. It is presented in detail in the next section.

B. Scaling

Variable scaling can be seen as a generalization of variable selection; in variable selection the scalars are restricted to attain either values 0 or 1, while in scaling all the values from the range $[0, 1]$ are accepted. In this paper, we use Delta Test (DT) as a critirion for selecting the scaling weights. The scalars are optimized by iterative Forward-Backward Selection (FBS) (see [8], for example). FBS is usually used for variable selection, but it can be extended to scaling as well; Instead of turning scalars from 0 to 1 or vice versa, increases by $1/h$ (in the case of forward selection) or decreases by $1/h$ (in the case of backward selection) are allowed. DT is useful in evaluation of correlation of random variables and therefore it can be used for scaling: The weights that give the smallest δ are selected.

V. LS-SVM

LS-SVM is a least square modification of the Support Vector Machine (SVM) [5]. SVM is a powerful adaptive method mainly because of its good generalization performance and robustness to high dimensional data [9]. Another attractive property is that training of a SVM leads into a quadratic programming task which guarantees that the optimum, once it has been found, is a global one.

The optimization problem of LS-SVM is simplified so that it reduces into a linear set of equations. Thus the problem is much faster to solve and at the same time the absence of local minima is guaranteed.

Consider a set of N training examples $(\mathbf{x}_i, y_i)_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the i-th input and $y_i \in \mathbb{R}$ is the corresponding output pattern. The LS-SVM model becomes

$$\hat{y} = \boldsymbol{\omega}^T \boldsymbol{\psi}(\mathbf{x}) + b, \tag{9}$$

where $\boldsymbol{\psi} : \mathbb{R}^n \longmapsto \mathbb{R}^{n_h}$ is a mapping from the input space onto a higher dimensional hidden space, $\boldsymbol{\omega} \in \mathbb{R}^{n_h}$ is a weight vector and b is a bias term. The optimization problem is formulated as

$$\min_{\boldsymbol{\omega}, b} \; J(\boldsymbol{\omega}, e) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{1}{2}\gamma \sum_{i=1}^{N} e_i^2, \tag{10}$$
$$\texttt{s.t.} \quad y_i = \boldsymbol{\omega}^T \boldsymbol{\psi}(\mathbf{x_i}) + b + e_i,$$

where e_i is the prediction error and $\gamma \geq 0$ is a regularization parameter that controls the trade-off between flatness of the function and accuracy of the function. The dual problem can be obtained using Lagrangian multipliers which leads into a linear KKT system that is easy to solve [5]. Using the dual solution, the original model (9) can be reformatted as

$$\hat{y} = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x_i}) + b,$$

where the kernel $K(\mathbf{x}, \mathbf{x_i}) = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\psi}(\mathbf{x_i})$ is a continuous and symmetric mapping from $\mathbb{R}^n \times \mathbb{R}^n$ to $\mathbb{R}$ and α_i are the Lagrange multipliers. It should be emphasized that although we formally define the high dimensional hidden space $\mathbb{R}^{n_h}$ and the mapping $\boldsymbol{\psi}(\mathbf{x})$, there is no need to compute anything in the hidden space; the knowledge of the kernel K is enough. A widely-used choice for is the standard Gaussian kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \texttt{exp}\{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2/\theta^2\}$.

VI. *k*-Nearest Neighbors

The *k*-Nearest Neighbors (*k*-NN) approximation method is a very simple, but powerful method. It has been used in many different applications and particularly in classification tasks [10]. The key idea behind the *k*-NN is that similar training samples have similar output values. One has to look for a certain number of nearest neighbors, according to the Euclidean distance [10], and their corresponding output values to get the approximation of the desired output.

We calculate the estimation of the output simply by using the average of the outputs of the neighbors in the neighborhood as

$$\hat{y}_i = \frac{\sum_{j=1}^{k} y_{P(j)}}{k}, \tag{11}$$

where $\hat{y}_i$ represents the output estimation, $P(j)$ is the index number of the j^{th} nearest neighbor of sample $\mathbf{x}_i$ and k is the number of neighbors used.

It is possible to use some weighting of the neighbors in the neighborhood or more sophisticated neighbor selection methods, but these aspects are not considered here.

We use the same neighborhood size for every data point, so we use a global k, which must be determined beforehand. The *k*-NN is a method with no parameters whatsoever: only the structural aspects, the number of neighbors and the inputs, need to be determined. After that, the *k*-NN is ready to be applied to the problem at hand.

VII. Experimental Results

In this paper, the global methodology is applied to the ESTSP'07 competition dataset [1] and the dataset B of the NN3 Forecasting Competition [2]. To illustrate the results, the ESTSP'07, the 3th and the 4th time series of NN3 are used. The results related to the other NN3 time series will also be submitted to the competition but are not presented here.

A. ESTSP'07 Results

The dataset is shown in Figure 1.

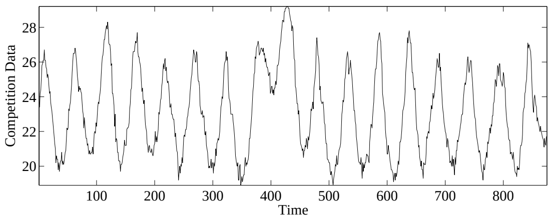

Fig. 1. Competition dataset.

In order to test the methodology, the dataset is divided into two sets, a small learning set and the global learning set. The small learning set consists of 465 first values and the global learning set consists in the 875 values. The regressor size is set to 10. The small learning set is used in order to evaluate the performances of the methodology.

The variable scaling is selected in order to minimize the Delta Test estimation. Because the DT is not continuous with respect to the scaling factors, a forward-backward optimization is used. The variable scaling coefficients are selected between a set of discrete values: [0 0.1 0.2 ... 0.9 1].

This discretization provides satisfactory results and reduces computational time.

The variable scaling is performed for each of the 50 prediction models from equation 3 used in direct prediction methodology. The estimation of the NNE (using Delta Test) are shown in Figure 2.

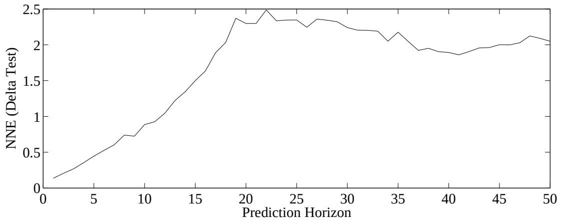

Fig. 2. Estimation of the NNE (using Delta Test) with respect to the horizon of prediction.

LS-SVM models are used to build the predictions. The result of the 50 step-ahead prediction and is represented in figure 3.

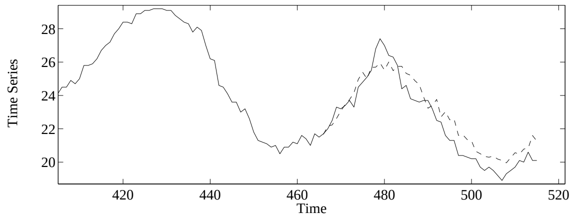

Fig. 3. Comparison between the time series (solid line) and the prediction (dashed line)

Then, the same methodology is used with the global learning set in order to predict the competition values. The result of the 50 step-ahead prediction is represented in figure 4.

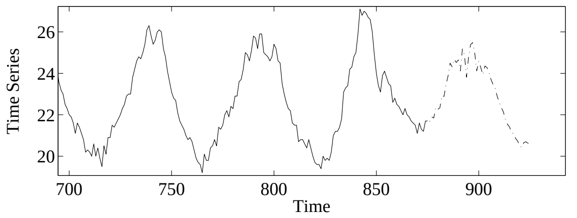

Fig. 4. Prediction of 50 next values of the competition dataset. The real values are presented by the solid line and the dashed one presents the prediction.

B. NN3 Results

In this competition the goal is the prediction of the 18 next values of the time series.

1) NN3 Results: 4th Time Series: The 4th dataset is shown in Figure 5.

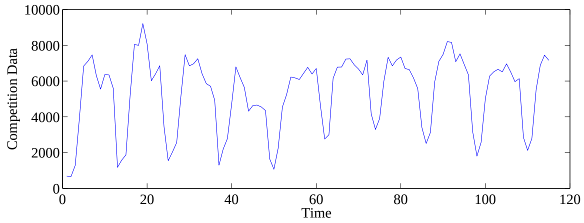

Fig. 5. Competition dataset.

The same methodology is applied. The regressor size is set to 12. The variable scaling is performed for each of the 18 prediction models from equation 3 used in direct prediction methodology. The estimation of the NNE (using Delta Test) are shown in Figure 6.

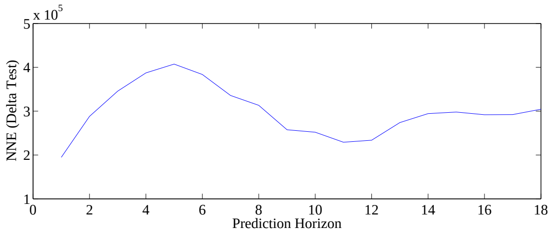

Fig. 6. Estimation of the NNE (using Delta Test) with respect to the horizon of prediction.

K-NN models are used to build the predictions. The result of the 18 step-ahead prediction is represented in figure 7.

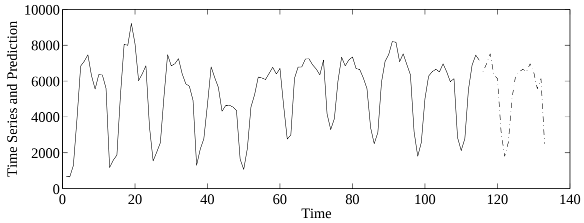

Fig. 7. Prediction of 18 next values of the competition dataset. The real values are presented by the solid line and the dashed one presents the prediction.

2) NN3 Results: 3rd Time Series: The 3rd dataset is shown in Figure 8.

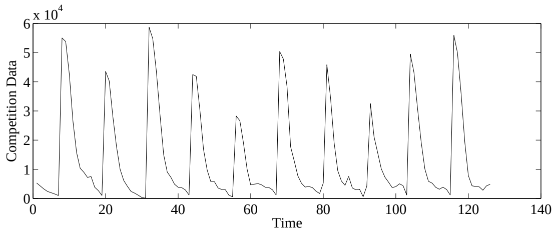

Fig. 8. Competition dataset.

The same methodology is applied. The regressor size is set to 12. The variable scaling is performed for each of the 18 prediction models from equation 3 used in direct prediction methodology. The estimation of the NNE (using Delta Test) are shown in Figure 9.

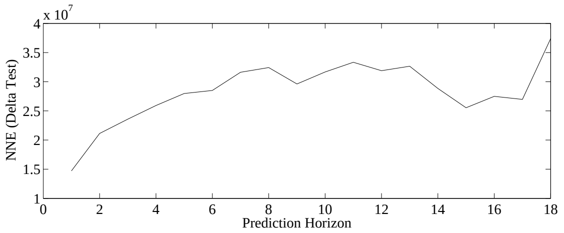

Fig. 9. Estimation of the NNE (using Delta Test) with respect to the horizon of prediction.

K-NN models are used to build the predictions. The result of the 18 step-ahead prediction is represented in figure 10.

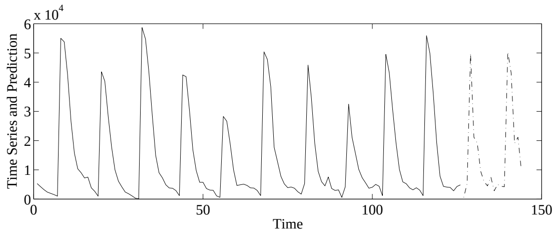

Fig. 10. Prediction of 18 next values of the competition dataset. The real values are presented by the solid line and the dashed one presents the prediction.

VIII. Conclusion

In this paper, we have presented a totally automatic methodology for the long-term prediction of time series.

This automatic methodology uses direct prediction strategy. This increases the computational time but improves the quality of the results.

In order to perform the variable scaling, Delta Test estimation is used. The scaling that minimized the NNE is selected. To reduce the computational time, a discrete scaling is used and a forward-backward optimization is selected.

Further research will be done to improve the minimization of the NNE estimation. Other experiments will be performed in the fields of time series prediction and function approximation.

Acknowledgment

Part the work of A. Lendasse and E. Liitiainen is supported by the project of New Information Processing Principles, 44886, of the Academy of Finland. The work of A. Lendasse is supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] Http://www.estsp2007.org/.

[2] Http://www.neural-forecasting-competition.com/.

[3] L. Ljung, *System identification theory for User*. Prentice-Hall, Englewood CliPs, NJ, 1987.

[4] A. Weigend and N. Gershenfeld, *Times Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley Publishing Company, 1994.

[5] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific Publishing Co., Pte, Ltd. (Singapore), Nov. 2002.

[6] A. J. Jones, "New tools in non-linear modeling and prediction," *Computational Management Science*, vol. 1, pp. 109–149, 2004.

[7] A. Stefansson, N. Koncar, and A. J. Jones, "A note on the gamma test," *Neural Computing & Applications*, no. 5(3), pp. 131–133, 1997.

[8] N. Benoudjit, E. Cools, M. Meurens, and M. Verleysen, "Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models." *Chemometrics and Intelligent Laboratory Systems*, vol. 70, pp. 47–53, 2004.

[9] H. S., *Neural Networks: A Comprehensive Foundation, 2nd ed.* New York: Prentice Hall Inc., 1999.

[10] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.