Methodology for long-term prediction of time series

Article *in* Neurocomputing · October 2007

DOI: 10.1016/j.neucom.2006.06.015 · Source: DBLP

Amaury Lendasse
University of Iowa

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Methodology for long-term prediction of time series

Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, Amaury Lendasse*

Helsinki University of Technology, Adaptive Informatics Research Centre, P.O. Box 5400, 02015 Espoo, Finland

Abstract

In this paper, a global methodology for the long-term prediction of time series is proposed. This methodology combines direct prediction strategy and sophisticated input selection criteria: *k*-nearest neighbors approximation method (*k*-NN), mutual information (MI) and nonparametric noise estimation (NNE). A global input selection strategy that combines forward selection, backward elimination (or pruning) and forward–backward selection is introduced. This methodology is used to optimize the three input selection criteria (*k*-NN, MI and NNE). The methodology is successfully applied to a real life benchmark: the Poland Electricity Load dataset.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Time series prediction; Input selection; *k*-Nearest neighbors; Mutual information; Nonparametric noise estimation; Recursive prediction; Direct prediction; Least squares support vector machines

1. 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 analyze 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. [11], and nonlinear ones such as artificial neural networks [21]. 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 [21]. In this paper, two variants of prediction strategies, namely, direct and recursive predictions are studied and compared. This paper illustrates that the direct prediction strategy gives better results than the recursive one.

In this paper, a global methodology to perform direct prediction is presented. It includes input selection strategies and input selection criteria. A global input selection strategy that combines forward selection, backward elimination and forward–backward selection is introduced. It is shown that this selection strategy is a good alternative to exhaustive search, which suffers from too large computational load.

Three different input selection criteria are presented for the comparison of the input sets: *k*-nearest neighbors based input selection criteria (*k*-NN), mutual information (MI) and nonparametric noise estimation (NNE). The optimal set of inputs is the one that optimizes one of the three criteria; for example, the optimal set of inputs can be defined as the one that maximizes the MI between the inputs and the output.

This paper shows that all the presented criteria (*k*-NN, MI and NNE) provide good selections of inputs. It is also shown experimentally that the introduced global methodology provides accurate predictions with all three presented criteria.

*Corresponding author.
E-mail addresses: antti.sorjamaa@hut.fi (A. Sorjamaa), jhao@cis.hut.fi (J. Hao), nreyhani@cis.hut.fi (N. Reyhani), yji@cis.hut.fi (Y. Ji), lendasse@cis.hut.fi, amaury.lendasse@hut.fi (A. Lendasse).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.06.015

In this paper, least squares support vector machines (LS-SVM) are used as nonlinear models in order to avoid local minima problems [19].

Section 2 presents the prediction strategies for the long-term prediction of time series. In Section 3 the global methodology is introduced. Section 3.1 presents the input selection strategies and Section 3.2 the input selection criteria. Finally, the prediction model LS-SVM is briefly summarized in Section 4 and experimental results are shown in Section 5 using a real life benchmark: the Poland electricity load dataset.

2. 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 Eq. (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.

2.1. 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:

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

where M denotes the number inputs. The regressor of the model is defined as the vector of inputs: $y_t, y_{t-1}, \ldots, y_{t-M+1}$. 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:

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

In Eq. (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 Hth 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.

2.2. Direct prediction strategy

Another strategy for the long-term prediction is the direct strategy. For the H-steps ahead prediction, the model is

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

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 as illustrated in Section 5.

3. Methodology

In the experiments, the direct prediction strategy is used. H models have to be built as shown in Eq. (3). For each model, three different input selection criteria are presented:

- minimization of the k-NN leave-one-out generalization error estimate,
- maximization of the MI between the inputs and the output,
- minimization of the NNE.

In order to optimize one of the criteria, a global input selection strategy combining the forward selection, the backward elimination and the forward–backward selection is presented in Section 3.1.

The estimation of MI and NNE demands the choice of hyperparameters. The definitions and the significance of the hyperparameters are more deeply explained in Sections 3.2.2 and 3.2.3. In this paper, the most adequate hyperparameter values are selected by minimizing the LOO error provided by k-NN approximators presented in Section 3.2.

In order to avoid local minima in the training phase of the nonlinear models (f_k in Eq. (3)), the LS-SVM are used. The LS-SVM are presented in Section 4.

3.1. Input selection strategies

Input selection is an essential pre-processing stage to guarantee high accuracy, efficiency and scalability [7] in problems such as machine learning, especially when the number of observations is relatively small compared to the number of inputs. It has been the subject in many application domains like pattern recognition [14], process identification [15], time series modeling [20] and

Please cite this article as: A. Sorjamaa, et al., Methodology for long-term prediction of time series, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015

econometrics [13]. Problems that occur due to poor selection of input variables are:

- If the input dimensionality is too large, the 'curse of dimensionality' problem [20] may happen. Moreover, the computational complexity and memory requirements of the learning model increase. Additional unrelated inputs lead to poor models (lack of generalization).
- Understanding complex models (too many inputs) is more difficult than simple models (less inputs), which can provide comparable good performances.

Usually, the input selection methods can be divided into two broad classes: *filter* methods and *wrapper* methods, see Fig. 1.

In the case of the filter methods, the best subset of inputs is selected *a priori* based only on the dataset. The input subset is chosen by an evaluation criterion, which measures the relationship of each subset of input variables with the output. In the literature, plenty of filter measure methods of different natures [2] exist: distance metrics, dependence measures, scores based on the information theory, ... , etc. In the case of the wrapper methods, the best input subset is selected according to the criterion, which is directly defined from the learning algorithm. The wrapper methods search for a good subset of inputs using the learning model itself as a part of the evaluation function. This evaluation function is also employed to induce the final learning model.

Comparing these two types of input selection strategies, the wrapper methods solve the real problem. But it is potentially very time consuming, as the ultimate algorithm has to be included in the cost function. Therefore, thousands of evaluations are performed when searching for the best subset. For example, if 15 input variables are considered and if the forward selection strategy (introduced in Section 3.1.2) is used, then $15(15+1)/2=120$ different subsets have to be tested. In practice, more than 15 inputs are realistic for time series prediction problems and the computational time is thus increased dramatically.

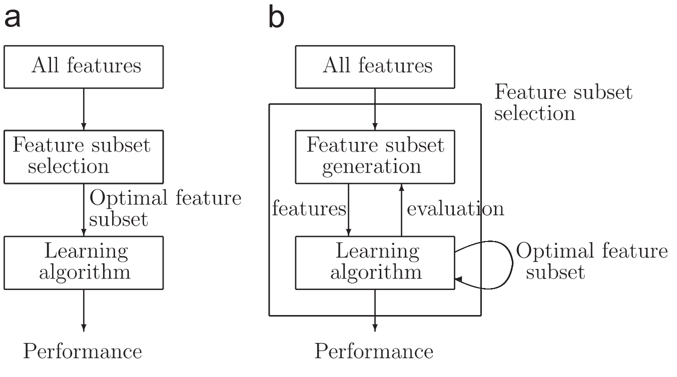


Fig. 1. Two approaches of input variable subset selection. (a) Filter method, (b) wrapper method.

On the contrary, the filter method is much faster because the procedure is simpler. In this paper, due to the long computational time of the wrapper method, it is unrealistic to compare the wrapper and filter methods for the input selection problem that is studied.

In the following sections, the discussion is focused on the filter methods. The filter method selects a set of inputs by optimizing a criterion over different combinations of inputs. The criterion computes the dependencies between each combination of inputs and the output using predictability, correlation, mutual information or other statistics. Various alternatives of these criteria exist.

This paper uses three methods based on different criteria: k-NN, MI and NNE. In the following, MI is taken as an example to explain the global input selection strategy. For the other two input selection criteria, the procedures are similar.

3.1.1. Exhaustive search

The optimal algorithm is to compute MI between all the possible combinations of inputs and the output, e.g. 2^M-1 inputs combinations are tested (M is the number of input variables). Then, the one that gives maximum MI is selected. In the case of long-term prediction of time series, M is usually larger than 15, so the exhaustive search procedure becomes too time consuming. Therefore, a global input selection strategy that combines forward selection, backward elimination and forward–backward selection is used. Forward selection, backward elimination and forward–backward selection are summarized in the following sections.

3.1.2. Forward selection

In this method, starting from the empty set S of selected input variables, the best available input is added to the set S one by one, until the size of S is M. Suppose we have a set of inputs $X^i, i=1,2,\dots,M$ and output Y, the algorithm is summarized in Fig. 2.

In forward selection, only $M(M+1)/2$ different input sets are evaluated. This is much less than the number of input sets evaluated with exhaustive search. On the other hand, optimality is not guaranteed. The selected set may not be the global optimal one.

```
(1) (Initialization)
    Set $F$ to be the initial set of original $M$ inputs, and $S$ to be the empty
    set which will contain the selected inputs.
(2) (Selection of the variables)
    Find:
        $X^s = \arg\max_{X^i} \mathrm{MI}(\{S, X^i\}, Y), \qquad X^i \in F.$
    Save $\mathrm{MI}(\{S, X^s\}, Y)$ and move $X^s$ from $F$ to $S$.
    Continue with the same way, until the size of $S$ is $M$.
    At the same time, save the MI value for each size of set $S$.
(3) (Result)
    Compare the MI values for all sizes of sets $S$, the selection result is the
    set $S$ that maximizes the MI.
```

Fig. 2. Forward selection strategy.

Please cite this article as: A. Sorjamaa, et al., Methodology for long-term prediction of time series, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015

3.1.3. Backward elimination or pruning

Backward elimination, also called pruning [12] procedure, is the opposite of forward selection process. In this strategy, the selected inputs set S is initialized to contain all the input variables. Then, the input variable for which the elimination maximizes MI is removed from set S one by one, until the size of S is 1.

Basically, backward elimination is the same procedure as forward selection presented in the previous section, but reversed. It evaluates the same amount of input sets as forward selection, $M(M+1)/2$. Also, the same restriction exists, optimality is not guaranteed.

3.1.4. Forward–backward selection

Both forward selection and backward elimination methods suffer from an incomplete search. Forward–backward selection algorithm combines both methods. It offers the flexibility to reconsider input variables previously discarded and *vice versa*, to discard input variables previously selected. It can start from any initial input set, including empty, full or randomly initialized input set.

Let us suppose a set of inputs X^i, $i = 1, 2, \ldots, M$ and output Y, the procedure of the forward–backward Selection is summarized in Fig. 3.

It is noted that the selection result depends on the initialization of the input set. In this paper, two options are considered. One is to begin from the empty set and the other is to begin from the full set.

The number of input sets to be evaluated varies and is dependent on the initialization of the input set, the stopping criteria and the nature of the problem. Still, it is not guaranteed that in all cases this selection method finds the global optimal input set.

3.1.5. Global selection strategy

In order to select the best input set, we propose to use all four selection methods: forward selection, backward elimination, forward–backward selection initialized with an empty set of inputs and forward–backward selection initialized with a full set of inputs. All four selection methods are fast to perform, but do not always converge to the same input set, because of the local minima. Therefore, it is necessary to use all of them to get more optimal selection. From the candidate input sets of all four selection methods, the one that optimizes the chosen criteria (k-NN, MI or NNE) is selected.

(1) (Initialization)
Let set S to be the selected input set, which can contain any input variables, and set F to be the unselected input set containing the inputs, which are not in set S. Compute $\mathrm{MI}(S, Y)$.

(2) (Forward-backward selection)
Find:

$$X^s = \arg\max_{X^i, X^j} \{\mathrm{MI}(\{S, X^j\}, Y)\} \cup \{\mathrm{MI}(S \backslash X^i, Y)\}, X^i \in S, X^j \in F.$$

If the previous $\mathrm{MI}(S, Y)$ is larger than the new MI, stop; otherwise, update set S and save the new MI, repeat step 2 until S is equal to any former selected S.

(3) (Result)
The selection result is in set S.

Fig. 3. Forward–backward selection strategy.

This combined strategy does not guarantee the selection of the best input set that would be obtained with the exhaustive search strategy. Nevertheless, the input selection is improved and the number of tested subsets is considerably reduced compared to the exhaustive search strategy.

3.2. Input selection criteria

3.2.1. k-Nearest neighbors

The k-NN approximation method is a very simple and powerful method. It has been used in many different applications, particularly for classification tasks [3]. The key idea behind the k-NN is that samples with similar inputs have similar output values. Nearest neighbors are selected, according to Euclidean distance, and their corresponding output values are used to obtain the approximation of the desired output. In this paper, the estimation of the output is calculated simply by averaging the outputs of the nearest neighbors:

$$\hat{y}_i = \frac{\sum_{j=1}^{k} y_{j(i)}}{k}, \tag{4}$$

where $\hat{y}_i$ represents the estimate (approximation) of the output, $y_{j(i)}$ is the output of the jth nearest neighbor of sample x_i and k denotes the number of neighbors used.

The distances between samples are influenced by the input selection. Then, the nearest neighbors and the approximation of the outputs depend on the input selection.

The k-NN is a nonparametric method and only k, the number of neighbors, has to be determined. The selection of k can be performed by many different model structure selection techniques, for example k-fold cross-validation [9], leave-one-out [9], Bootstrap [5] and Bootstrap 632 [6]. These methods estimate the generalization error obtained for each value of k. The selected k is the one that minimizes the generalization error.

In [16] all methods, the leave-one-out and Bootstraps, select the same input sets. Moreover, the number of neighbors is more efficiently selected by the Bootstraps [16]. It has also been shown that the k-NN itself is a good approximator for time series [16]. In this paper, however, the k-NN is not used as an approximator, but as a tool to select the input set.

3.2.2. Mutual information

The MI can be used to evaluate the dependencies between random variables. The MI between two variables, let, say, X and Y be the amount of information obtained from X in the presence of Y and *vice versa*. In time series prediction problem, if Y is the output and X is a subset of the input variables, the MI between X and Y is one criterion for measuring the dependence between inputs (regressor) and output. Thus, the inputs subset X, which gives maximum MI, is chosen to predict the output Y.

The definition of MI originates from the entropy in the information theory. For continuous random variables (scalar or vector), let $\mu^{X,Y}, \mu^X$ and μ^Y represent the joint probability density function and the two marginal density functions of the variables. The entropy of X is defined by Shannon [1] as

$$H(X) = -\int_{-\infty}^{\infty} \mu^X(x) \log \mu^X(x)\,\mathrm{d}x, \tag{5}$$

where log is the natural logarithm and then, the information is measured in natural units.

The remaining uncertainty of X is measured by the conditional entropy as

$$H(X|Y) = -\int_{-\infty}^{\infty} \mu^Y(y) \times \int_{-\infty}^{\infty} \mu^X(x|Y=y) \log \mu^X(x|Y=y)\,\mathrm{d}x\,\mathrm{d}y. \tag{6}$$

The joint entropy is defined as

$$H(X,Y) = -\int_{-\infty}^{\infty} \mu^{X,Y}(x,y) \log \mu^{X,Y}(x,y)\,\mathrm{d}x\,\mathrm{d}y. \tag{7}$$

The MI between variables X and Y is defined as [4]

$$\begin{aligned}\mathrm{MI}(X,Y) &= H(Y) - H(Y|X)\\ &= H(X) + H(Y) - H(X,Y).\end{aligned} \tag{8}$$

From Eqs. (5) to (8), MI is computed as

$$\mathrm{MI}(X,Y) = \int_{-\infty}^{\infty} \mu^{X,Y}(x,y) \log \frac{\mu^{X,Y}(x,y)}{\mu^X(x)\mu^Y(y)}\,\mathrm{d}x\,\mathrm{d}y. \tag{9}$$

For computing the MI, only the estimations of the probability density functions $\mu^{X,Y}, \mu^X$ and μ^Y are required.

In this paper, $\mathrm{MI}(X,Y)$ is estimated by a k-NN approach presented in [10]. In order to distinguish the number of neighbors that used in the MI and the one used in the k-NN, the number of neighbors is denoted by l for the estimation of MI.

The novelty of this l-NN based MI estimator consists in its ability to estimate the MI between two variables of any dimensional space. Then, the estimation of MI depends on the predefined value l.

In [10], it is suggested to use a mid-range value $l=6$. But it has been shown that when applied to time series prediction problems, l needs to be tuned for different datasets and different data dimensions in order to obtain better performance. As explained in Section 3, to select the inputs based on the l-NN MI estimator, the optimal l is selected using k-NN and leave-one-out.

3.2.3. Nonparametric noise estimator using the gamma test

Gamma test (GT) is a technique for estimating the variance of the noise, or the mean square error (MSE), that can be achieved without overfitting [8]. The evaluation of the NNE is done using the GT estimation introduced by Stefansson in [17].

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

$$y_i = f(x_i) + r_i, \tag{10}$$

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

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

GT 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.

Two versions for evaluating the GT are suggested. The first one evaluates the value of γ, σ in increasing sized sets of data. Then the result for a particular parameter pair is obtained by averaging the results from all set sizes. The new or refined version establishes the estimation based on the k-NN differences instead of increasing the number of data points gradually. In order to distinguish the k used in the NNE context from the conventional k in k-NN, the number of nearest neighbors is denoted by p.

Let us denote the pth nearest neighbor of the point x_i in the set $\{x_1, \ldots, x_N\}$ by $x_{p(i)}$. Then the following variables, γ_N and σ_N are defined as

$$\gamma_N(p) = \frac{1}{2N}\sum_{i=1}^{N} |y_{p(i)} - y_i|^2, \tag{11}$$

$$\sigma_N(p) = \frac{1}{2N}\sum_{i=1}^{N} |x_{p(i)} - x_i|^2, \tag{12}$$

where $|.|$ denotes the Euclidean metric and $y_{p(i)}$ is the output of $x_{p(i)}$. For correctly selected p [8], the constant term of the linear regression model between the pairs $(\gamma_N(p), \sigma_N(p))$ determines the noise variance estimate. For the proof of the convergence of the GT, see [8].

The GT assumes the existence of the first and second derivatives of the regression function. Let us denote

$$\nabla f(x) = \left(\frac{\partial f}{\partial x(i)}\right)_{i=1}^{M}, \quad Hf(x) = \left(\frac{\partial^2 f}{\partial x(i)\partial x(j)}\right)_{i,j=1}^{M}, \tag{13}$$

where x_i and x_j are the ith and jth components of x, respectively. M is the number of variables. The GT requires both $|Hf(x)|$ and $|\nabla f(x)|$ are bounded.

These two conditions are general and are usually satisfied in practical problems. The GT requires no other assumption on the smoothness property of the regression

Please cite this article as: A. Sorjamaa, et al., Methodology for long-term prediction of time series, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015

function. Consequently, the method is able to deal with the regression functions of any degree of roughness.

The second assumption is about the noise distribution:

$$E_{\Phi}\{r\} = 0 \quad \text{and} \quad E_{\Phi}\{r^2\} = \text{var}\{\varepsilon\} < \infty, \tag{14}$$

$$E_{\Phi}\{r^3\} < \infty \quad \text{and} \quad E_{\Phi}\{r^4\} < \infty, \tag{15}$$

where $E_{\phi}\{r\}$ is the noise density function. Furthermore, it is required that the noisy variable should be independent and identically distributed. In the case of heterogeneous noise, the GT provides the average of noise variance extracted from the whole dataset.

As discussed above (see Eq. (11)), the GT depends on the number of p used to evaluate the regression. It is suggested to use a mid-range value $p = 10$ [8]. But, when applied to time series prediction problems, p needs to be tuned for each dataset and for each set of variables to obtain better performance. As explained in Section 3, to select the inputs the optimal p is selected using k-NN and leave-one-out.

4. Nonlinear models

In this paper, LS-SVM are used as nonlinear models [19], which are defined in their primal weight space by [22,18]

$$\hat{y} = \omega^{\mathrm{T}}\varphi(\mathbf{x}) + b, \tag{16}$$

where $\varphi(\mathbf{x})$ is a function, which maps the input space into a higher-dimensional feature space, $\mathbf{x}$ is the vector of inputs. ω and b are the parameters of the model. The optimization problem can be formulated as

$$\min_{\omega,b,e} J(\omega, e) = \frac{1}{2}\omega^{\mathrm{T}}\omega + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2, \tag{17}$$

$$\text{subject to } y_i = \boldsymbol{\omega}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b + e_i, \quad i = 1, \ldots, N, \tag{18}$$

and the solution is

$$h(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{19}$$

In the above equations, i refers to the index of a sample and $K(\mathbf{x}, \mathbf{x}_i)$ is the kernel function defined as the dot product between the $\varphi(\mathbf{x})^{\mathrm{T}}$ and $\varphi(\mathbf{x})$. Training methods for the estimation of the ω and b parameters can be found in [22].

5. Experiments

5.1. Dataset

One time series is used as an example. The dataset is called Poland Electricity Load, and it represents two periods of the daily electricity load of Poland during around 1500 days in the 1990s [23]. The quasi-sinusoidal seasonal variation is clearly visible from the dataset.

The first 1000 values are used for training, and the remaining data for testing. The learning part of the dataset is shown in Fig. 4.

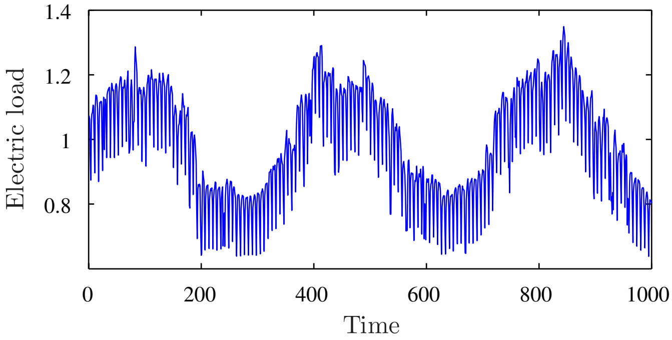


Fig. 4. Learning set of the Poland Electricity Load dataset.

5.2. Results

The maximum regressor size is set to 15 according to [16]. Two-weeks regressor is large enough to catch the main dynamics of the electricity load time series. The selected inputs based on the three methods are shown in Table 1. For example, the inputs selected by MI for the one-step ahead prediction are t, $t-6$ and $t-7$. Then, the prediction model is

$$y(t+1) = f_1(y(t), y(t-6), y(t-7)). \tag{20}$$

From Table 1, it can also be seen that the time distance between the target time and some selected inputs is constant over the whole prediction horizon. For example, input $t-6$ is used to predict $t+1$, input $t-5$ is used to predict $t+2$, input $t-4$ is used to predict $t+3$, etc. This fact is due to the weekly dynamics of the time series.

The number of inputs selected by the k-NN varies from 2 to 9 and on average is 7 (from the maximum of 15 inputs). It shows that the models are sparse and the curse of dimensionality is reduced.

The sparsity also enables a physical interpretation of the selected inputs. For example, for one-step ahead prediction, the inputs selected by the k-NN are t, $t-5$, $t-6$, $t-7$ and $t-13$. This means, that in order to predict the load of the next day, let us say Tuesday, we need to use the load of Monday (current day); Wednesday, Tuesday, Monday of the previous week and Tuesday 2 weeks before. The load of the current day is needed, because it is the most up-to-date measurement. Monday, Tuesday and Wednesday of the previous week are needed to estimate the trend of the electricity load over Tuesday. Tuesday 2 weeks before is needed to handle the day specific changes in the electricity load.

The LS-SVM are used to compare the performances. A 10-fold cross-validation [9] procedure for model structure selection purposes has been applied.

The MSE of direct prediction on the test set, based on the three input selection methods are drawn in Fig. 5.

All input selection criteria (k-NN, MI and GT) provide good and quite similar inputs. The selected inputs also provide predictions with similar errors. From the three

Table 1
Selected inputs for the Poland electricity Load dataset

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	x ○ △	x △	x ○ △	x ○ △	○ △	x ○ △	x ○ △	x ○ △	x ○ △	x ○ △	○ △	○ △	x ○ △	x ○ △	x ○ △
-1		x ○	△	x ○	x ○ △	x ○ △		○		○ △	△	x ○ △	x ○ △		△
-2		x ○ △	x ○ △	x △	x ○ △		○		△	○	○ △	x ○ △	△		
-3		x	x △	x ○ △	△	△				○ △	x ○ △	○			△
-4		○ △	x ○ △	△	△				△	x ○ △	△	△			
-5	○ △	x ○ △	△	○			○		x ○ △	△	△		△		△
-6	x ○ △	○ △	△	○				x ○ △	△	○ △	○ △	△			x ○ △
-7	x △	○ △			○	x ○	○			○			○	○ △	○
-8					x ○	x ○ △		○		○		x ○	x ○ △	○	△
-9		○			x ○ △	○				○	△	x ○ △	△		
-10			○ △	x ○ △							x ○ △	○			
-11	○	○ △	x ○ △	△					○ △	x △	x				
-12	○	○ △		x				△	x ○ △						
-13	○ △		x ○	x				x ○ △						x	x ○ △
-14	○		x ○				x ○ △	x ○ △	x ○ △	x △	○		x ○	x ○ △	x △

The numbers in the first row and first column represent time steps and regressor index, respectively. Symbol X is for MI selected inputs, O represents NNE selection results, Δ is for k-NN selected results.

Please cite this article as: A. Sorjamaa, et al., Methodology for long-term prediction of time series, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015

methods, the k-NN is the fastest and therefore should be preferred.

The MSE of direct and recursive predictions on the test set based on the k-NN input selection criteria are shown in Fig. 6.

From Fig. 6 it can be seen, that the direct prediction strategy gives smaller error than the recursive one. The error difference increases as the horizon of prediction increases. The error of the direct strategy is linear with respect to the horizon of prediction. This is not the case for the recursive strategy.

Fifteen time step predictions of the direct prediction method based on the k-NN input selection method are given in Fig. 7.

In Fig. 7 it can be seen that the long-term prediction has captured the intrinsic behavior of the time series.

Our results agree with the intuition and with the models used by the real life electricity companies in their electricity consumption estimation.

Similar results have been obtained on other time series benchmarks. The direct prediction strategy always provides accurate predictions. Furthermore, the global methodology introduced in this paper provides sparse and accurate long-term prediction models that can be easily interpreted.

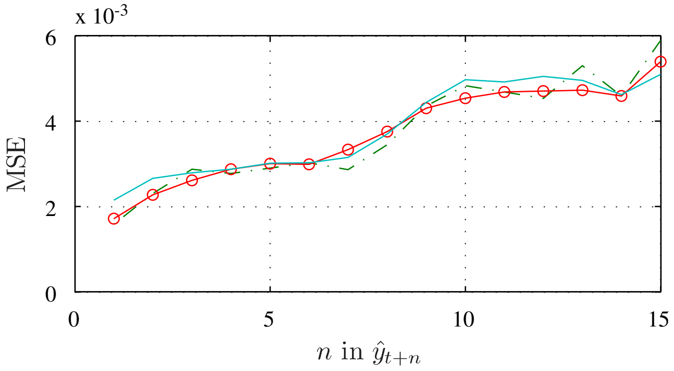


Fig. 5. The MSE of different methods on the test set of the Poland Electricity Load data: dashed line with · mark corresponds to MI selected inputs, solid line is for the NNE selected inputs, and solid line with ○ mark corresponds to the k-NN selected inputs.

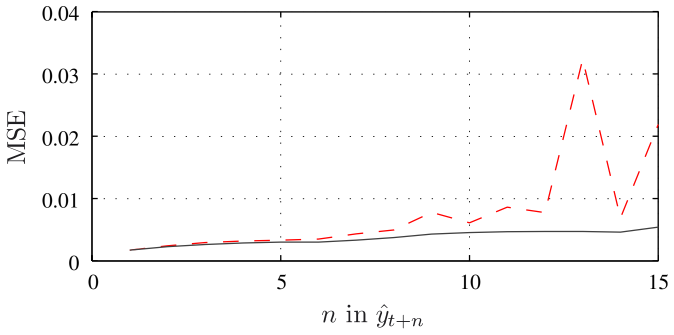


Fig. 6. The MSE of the direct and recursive predictions for the test set of Poland Electricity Load data: solid line represents the direct prediction error and dashed line is for the recursive prediction error.

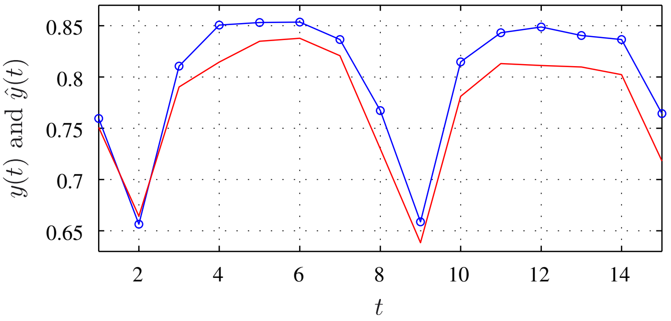


Fig. 7. Prediction results of the k-NN method for the Poland Electricity Load data: solid line is for the true values and solid line with ○ mark represents the prediction results.

6. Conclusion

This paper presents a global methodology for the long-term prediction of the time series. It illustrates that the direct prediction strategy gives better results than the recursive one. On the other hand, the direct prediction strategy multiplies the computational load by the number of prediction steps needed. In order to deal with the increase of the computational load, a fast and reliable global input selection strategy has been introduced.

It has been shown that the k-NN, the MI and the NNE criteria provide good selections of inputs. It is also shown that global input selection strategy combining the forward selection, the backward elimination and the forward–backward selection is a good alternative to the exhaustive search, which suffers from a too large computational load.

The k-NN selection criterion is the fastest, because the selection of hyperparameters is not needed. This makes k-NN roughly 10 times faster than MI and 20 times faster than NNE.

The use of LS-SVM, which do not suffer from the problems of local minima, allows reliable comparison. The methodology has been applied successfully to a real life benchmark. The sparseness of the selected models allows straightforward physical interpretations.

In further works, efforts have to be done to reduce the computational load of the input selection criteria. Alternatives to the forward, the backward and the forward–backward selection strategies have to be explored as well.

References

[1] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Network 50 (1994) 537–550.

[2] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, in: Handbook of Statistics, vol. II, 1982, pp. 773–910.

[3] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.

[4] T. Cover, J. Thomas, Elements of Information Theory, Wiley, New York, 1991.

[5] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, London, 1993.

[6] B. Efron, R.J. Tibshirani, Improvements on cross-validation: the .632+ bootstrap method, J. Am. Statist. Assoc. 92 (1997) 548–560.

[7] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2001.

[8] A.J. Jones, New tools in non-linear modeling and prediction, Comput. Manage. Sci. 1 (2004) 109–149.

[9] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, 1995.

[10] A. Kraskov, H. Stgbauer, P. Grassberger, Estimating mutual information, Phys. Rev. 69 (2004) 066138.

[11] L. Ljung, System Identification Theory for User, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[12] G. Manzini, Perimeter search in restricted memory, Comput. Math. Appl. 32 (1996) 37–45.

[13] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, Eur. J. Oper. Res. 171 (2006) 842–858.

[14] E. Rasek, A contribution to the problem of feature selection with similarity functionals in pattern recognition, Pattern Recognition 3 (1971) 31–36.

[15] Q. Shen, R. Jensen, Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring, Pattern Recognition 37 (2004) 1351–1363.

[16] A. Sorjamaa, N. Reyhani, A. Lendasse, Input and structure selection for *k*-nn approximator, in: J. Cabestany, A. Prieto, F.S. Hernandez (Eds.), Lecture Notes in Computer Science, vol. 3512, pp. 985–991. Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Barcelona, Spain, Springer, Berlin/Heidelberg, 2005.

[17] A. Stefansson, N. Koncar, A.J. Jones, A note on the gamma test, Neural Comput. Appl. 5 (3) (1997) 131–133.

[18] J.A.K. Suykens, J.D. Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (2002) 85–105.

[19] J.A.K. Suykens, T.V. Gestel, J.D. Brabanter, B.D. Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.

[20] M. Verleysen, D. Francois, The curse of dimensionality in data mining and time series prediction, in: J. Cabestany, A. Prieto, F.S. Hernandez (Eds.), Lecture Notes in Computer Science, vol. 3512, pp. 758–770. Invited Talk in Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Barcelona, Spain, Springer, Berlin/Heidelberg, 2005.

[21] A. Weigend, N. Gershenfeld, Times Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, MA, 1994.

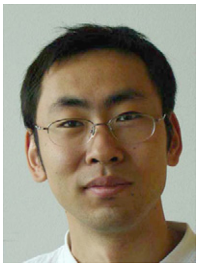
[22] Available from ⟨http://www.esat.kuleuven.ac.be/sista/lssvmlab/⟩.

[23] ⟨http://www.cis.hut.fi/projects/tsp/?page = Timeseries⟩.

Antti Sorjamaa was born in 1980 in a small city in northern Finland. He received his master degree from Helsinki University of Technology in 2005. His Master's thesis is entitled ''Strategies for the Long-Term Prediction of Time Series using Local Models''. Currently, he is continuing as a Ph.D. student in HUT. He is the author or the coauthor of six scientific papers in international journals, books or communications to conferences with reviewing committee. The topic of his research is missing value problems in temporal databases.

Jin Hao was born in China in 1980. She received her bachelor degree from Beijing University of Technology in 2003 and her master degree from Helsinki University of Technology in 2005. Her master thesis title is "Input Selection using Mutual Information—Applications to Time Series Prediction". She is the author or the coauthor of five scientific papers in international journals, books or communications to conferences with reviewing committee. She is now working in the Telecommunications and Networking Department of Samsung Electronics in Korea.

Nima Reyhani was born in the northern part of Iran (Persia) in 1979. He received his bachelor degree from University of Isfahan. During his bachelor studies, he was working in Iran Telecom Research Center. He received his master degree from Helsinki University of Technology, Finland. He is the author or the coauthor of seven scientific papers in international journals, books or communications to conferences with reviewing committee. Now, he is a Ph.D. student in HUT and his field of research is Noise Estimation.



Yongnan Ji was born in 1981 in Daqing, in northern part of China. He received his bachelor degree from Harbin Institute of Technology in 2003, China. In 2005, he received his master degree from Helsinki University of Technology, Finland. The title of his master thesis is "Least Squares Support Vector Machines for Time Series Prediction". He is the author or the coauthor of four scientific papers in international journals, books or communications to conferences with reviewing committee. He is currently a Ph.D. student in HUT, working on machine learning algorithms for chemometrics data.

Amaury Lendasse was born in 1972 in Belgium. He received the M.S. degree in mechanical engineering from the Université catholique de Louvain (Belgium) in 1996, M.S. in control in 1997 and Ph.D. in 2003 from the same University. In 2003, he has been a postdoctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. Since 2004, he is a senior researcher in the Adaptive Informatics Research Centre in the Helsinki University of Technology in Finland. He is leading the Time Series Prediction Group. He is the author or the coauthor of 64 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, nonlinear approximation in financial problems, functional neural networks and classification.

Please cite this article as: A. Sorjamaa, et al., Methodology for long-term prediction of time series, Neurocomputing (2007), doi:10.1016/j.neucom.2006.06.015