Direct and Recursive Prediction of Time Series Using Mutual Information Selection

Yongnan Ji, Jin Hao, Nima Reyhani, and Amaury Lendasse

Neural Network Research Centre,
Helsinki University of Technology, P.O. Box 5400,
02150 Espoo, Finland
{yji, jhao, nreyhani, lendasse}@cis.hut.fi

Abstract. This paper presents a comparison between direct and recursive prediction strategies. In order to perform the input selection, an approach based on mutual information is used. The mutual information is computed between all the possible input sets and the outputs. Least Squares Support Vector Machines are used as non-linear models to avoid local minima problems. Results are illustrated on the Poland electricity load benchmark and they show the superiority of the direct prediction strategy.

Keywords: Time Series Prediction, Mutual Information, Direct Prediction, Recursive Prediction, Least Squares Support Vector Machines and Prediction Strategy.

1 Introduction

Prediction is an important part of decision making and planning process in engineering, business, medicine and many other application domains. Long-term prediction is typically faced with growing uncertainties arising from various sources, for instance, accumulation of errors and lack of information [1]. In long-term prediction, when predicting multiple steps ahead, we have several choices. In this work, two variants of prediction approaches, namely, direct and recursive prediction, using Least Squares Support Vector Machines (LS-SVM) [17], are studied and compared. Meanwhile, to improve the efficiency of prediction, mutual information (MI) is used to select the inputs [12]. Based on the experimental results, a combination of input selection and forecast strategy which can give comparatively accurate long-term time series prediction will be presented.

The paper is organized as follows: in section 2, mutual information is introduced. Time series prediction is explained in section 3. In section 4, LS-SVM is defined. In section 5 we present the experimental results and in section 6 conclusions and further works are presented.

J. Cabestany, A. Prieto, and D.F. Sandoval (Eds.): IWANN 2005, LNCS 3512, pp. 1010 – 1017, 2005.

2 Mutual Information for Input Selection

2.1 Input Selection

Input selection is one of the most important issues in machines learning especially when the number of observations is relatively small compared to the number of inputs. In practice, there is no dataset with infinite number of data points and furthermore, the necessary size of the dataset increases dramatically with the number of observations (curse of dimensionality). To circumvent this, one should first select the best inputs or regressors in the sense that they contain the necessary information. Then it would be possible to capture and reconstruct the underlying relationship between input-output data pairs. Within this respect, some model dependent approaches have been proposed [2-6].

Some of them deal with the problem of feature selection as a generalization error estimation problem. In this methodology, the set of inputs that minimize the generalization error is selected using Leave-one-out, Bootstrap or other resampling techniques. These approaches are very time consuming and may take several weeks. However, there are model independent approaches [7-11] which select a priori inputs based only on the dataset, as presented in this paper. So the computational load would be less than in model dependent cases. Model independent approaches select 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 corresponding output using predictability, correlation, mutual information or other statistics.

In this paper, the mutual information is used as a criterion to select the best input variables (from a set of possible variables) for long-term prediction purpose.

2.2 Mutual Information

The mutual information (MI) between two variables, let say X and Y, is the amount of information obtained from X in presence of Y, and vice versa. MI can be used for evaluating the dependencies between random variables, and has been applied for Feature Selection and Blind Source Separation [12].

Let us consider two random variables: the MI between them would be

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \ , \tag{1}$$

where $H(.)$ computes the Shannon's entropy. In the continuous entropy case, equation (1) leads to complicated integrations, so some approaches have been proposed to evaluate them numerically. In this paper, a recent estimator based on k-Nearest Neighbors statistics is used [13]. The novelty of this approach consists in its ability to estimate the MI between two variables of any dimensional spaces. The basic idea is to estimate $H(.)$ from the average distance to the k-Nearest Neighbors (over all x_i). MI is derived from equation (1) and is estimated as

$$I(X,Y) = \psi(k) - 1/k - \left\langle \psi(n_x) + \psi(n_y) \right\rangle + \psi(N) \ , \tag{2}$$

with N the size of dataset and $\psi(x)$ the digamma function,

$$\psi(x) = \Gamma(x) - 1 - d\Gamma(x)/dx \ , \tag{3}$$

$\psi(1) \approx -0.5772156$ and

$$< \ldots >= N^{-1} \sum_{i=1}^{N} E[\ldots(i)] \ . \tag{4}$$

$n_x(i)$, $n_y(i)$ are the numbers of points in the region $\|x_i - x_j\| \leq \varepsilon_x(i)/2$ and $\|y_i - y_j\| \leq \varepsilon_y(i)/2$. $\varepsilon(i)/2$ is the distance from z_i to its k-Nearest Neighbors. $\varepsilon_x(i)/2$ and $\varepsilon_y(i)/2$ are the projections of $\varepsilon(i)/2$ [14]. k is set to be 6, as suggested in [14]. Software for calculating the MI based on this method can be downloaded from [15].

3 Time Series Prediction

Basically, time series prediction can be considered as a modeling problem [16]: a model is built between the input and the output. Then, it is used to predict the future values based on the previous values. In this paper we use two different strategies to perform the long-term prediction, which are direct and recursive forecasts.

3.1 Direct Forecast

In order to predict the values of a time series, M +1 different models are built,

$$\hat{y}(t+m) = f_m(y(t-1), y(t-2), \ldots, y(t-n)) \ , \tag{5}$$

with m = 0,1,…M, M is the maximum horizon of prediction. The input variables on the right-hand part of (5) form the regressor, where n is the regressor size.

3.2 Recursive Forecast

Alternatively, model can be constructed by first making one step ahead prediction,

$$\hat{y}(t) = f(y(t-1), y(t-2), \ldots, y(t-n)) \ , \tag{6}$$

and then predict the next value using the same model,

$$\hat{y}(t+1) = f(\hat{y}(t), y(t-1), y(t-2), \ldots, y(t-n+1)) \ . \tag{7}$$

In equation (7), the predicted value of $y(t)$ is used instead of the value itself, which is unknown. Then, $\hat{y}(t+1)$ to $\hat{y}(t+M)$ are predicted recursively.

4 Least Squares Support Vector Machines

LS-SVM are regularized supervised approximators. Comparing with simple SVM, Only linear equation is needed to solve the results, which avoids the local minima in SVM. A short summary of the LS-SVM is given here; more details are given in [17].

The LS-SVM model [18-20] is defined in its primal weight space by,

$$\hat{y} = \omega^T \varphi(\mathbf{x}) + b \ , \tag{8}$$

where $\varphi(x)$ is a function which maps the input space into a higher dimensional feature space, $\boldsymbol{x}$ is the N-dimensional vector of inputs x_i, and ω and b the parameters of the model. In Least Squares Support Vector Machines for function estimation, the following optimization problem is formulated,

$$\min_{\omega,b,e} J(\omega, e) = \frac{1}{2}\omega^T \omega + \gamma \frac{1}{2}\sum_{i=1}^{N} e_i^2 \ , \tag{9}$$

subject to the equality constraints,

$$y^i = \omega^T \varphi(\mathbf{x}^i) + b + e^i, i = 1, \ldots, N \ . \tag{10}$$

In equation (10), the superscript i refers to the number of a sample. Solving this optimization problem in dual space leads to finding the α_i and b coefficients in the following solution,

$$h(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \ . \tag{11}$$

Function $K(\mathbf{x}, \mathbf{x}_i)$ is the kernel defined as the dot product between the $\varphi(\mathbf{x})^{\mathrm{T}}$ and $\varphi(\mathbf{x})$ mappings. The meta-parameters of the LS-SVM model are σ, width of the Gaussian kernels (taken to be identical for all kernels), and γ, regularization factor. LS-SVM can be viewed as a form of parametric ridge regression in the primal space. Training methods for the estimation of the ω and b parameters can be found in [17].

5 Experimental Results

The dataset used in this experiment is a benchmark in the field of time series prediction: the Poland Electricity Dataset [21]. It represents the daily electricity load of Poland during 2500 days in the 90s.

The first two thirds of the whole dataset is used for training, and the remaining data for testing. To apply the prediction model in equation (5), we set the maximum time horizon $M = 6$ and the regressor size $n = 8$.

First, MI presented in section 2.2 is used to select the best input variables. All the 2^n-1 combinations of inputs are tested. Then, the one that gives the maximum MI is selected. The selection results for direct forecast are:

Table 1. Input selection results of MI

	$y(t)$	$y(t+1)$	$y(t+2)$	$y(t+3)$	$y(t+4)$	$y(t+5)$	$y(t+6)$
$y(t-1)$	X	X	X	X	X	X	X
$y(t-2)$	X	X	X	X	X	X	X
$y(t-3)$			X		X	X	
$y(t-4)$		X	X	X			
$y(t-5)$		X	X				
$y(t-6)$	X			X			
$y(t-7)$				X			
$y(t-8)$							X

For example, the 4th column means that,

$$\hat{y}(t+3) = f_3(y(t-1), y(t-2), y(t-4), y(t-6), y(t-7)) \ . \tag{12}$$

Then the LS-SVM is used to make the prediction. To select the optimal parameters model selection method should be used here, in the experiment, leave-one-out is uses. The errors for the leave-one-out procedure of every pairs of γ and σ are listed. Then the area around the minima is zoomed and searched until the hyper parameters are found. For recursive prediction, only one function is used, so one pair of γ and σ is needed, which is (33, 0.1). For direct prediction, seven pairs of parameters are required. They are (33, 0.1), (40, 0.1), (27, 0.1), (27, 0.1), (27, 0.1), (22, 0.1) and (27, 0.1). The mean square error values of the results are listed in the table below:

Table 2. MSE values of direct and recursive prediction

	$y(t)$	$y(t+1)$	$y(t+2)$	$y(t+3)$	$y(t+4)$	$y(t+5)$	$y(t+6)$
direct	0,00154	0,00186	0,00178	0,00195	0,00276	0,00260	0,00260
recursive	0,00154	0,00362	0,00486	0,00644	0,00715	0,00708	0,00713

As illustration, the MSE values are presented also in Fig. 1:

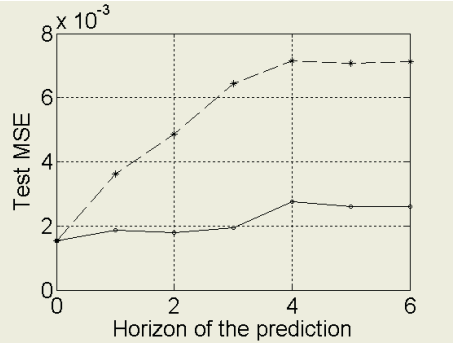

Fig. 1. Prediction results comparison: dashed line corresponds to recursive prediction and solid line corresponds to direct prediction

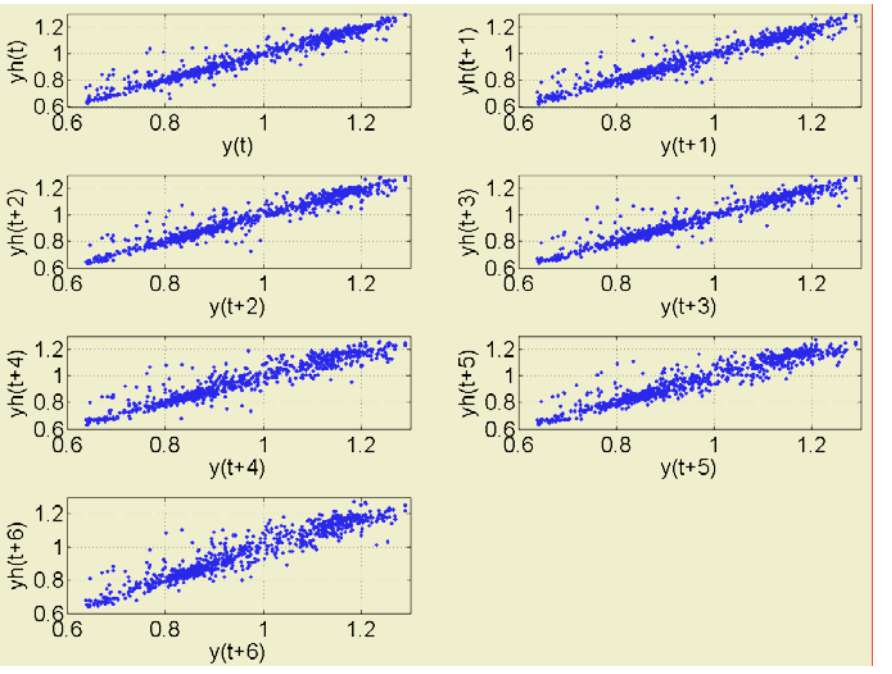

Fig. 2. $\hat{y}(t)$ (represented as *yh*) and $y(t)$ for each horizon of prediction

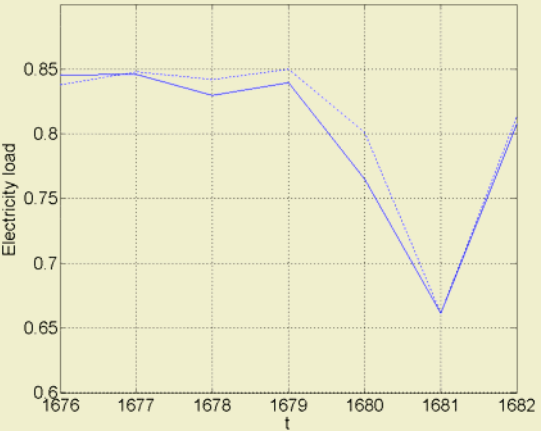

Fig. 3. An example of prediction: $\hat{y}(t)$ is represented in dotted line and $y(t)$ is represented in solid line

In Fig. 1, the horizontal axis represents i in $y(t+i)$, which varies from 0 to 6. The vertical axis represents the corresponding MSE values. The dashed line shows MSE values for recursive prediction and the solid line shows MSE values for direct predic-

tion. From this figure, it can be seen that as i increases, the performances of the direct predictions are better than that of the recursive ones.

To illustrate the prediction results, the predicted values by direct prediction are plotted against the real data in Fig. 2. The more the points are concentrated around a line, the better the predictions are. It can be seen that when i is large, the distribution of the points diverts from a line, because the prediction becomes more difficult.

In Fig. 3, one example of the prediction results is given. The dashed line represents seven real values from the Poland dataset. The solid line is the estimation using direct prediction. The figure shows that the predicted values and the real values are very close.

The same methodology has been applied to other benchmark and similar results have been obtained.

6 Conclusion

In this paper, we compared two long-term prediction strategies: direct forecast and recursive forecast. MI is used to perform the input selection for both strategies: MI works as a criterion to estimate the dependencies between each combination of inputs and the corresponding output. Though 2^n - 1 combinations must be calculated, it is fast compared to other input selection methods. The results show that this MI based method can provide a good input selection.

Comparing both long-term prediction strategies, direct prediction gives better performances than recursive prediction. The former strategy requires multiple models. Nevertheless, due to the simplicity of the MI input selection method, direct prediction strategy can be used in practice. Thus, the combination of direct prediction and MI input selection can be considered as an efficient approach for a long-term time series prediction.

Acknowledgements

Part of work of Y. Ji, J. Hao, N. Reyhani and A. Lendasse is supported by the project of New Information Processing Pinciples, 44886, of the Academy of Finland.

References

1. Weigend A.S., Gershenfeld N.A.: Times Series Prediction: Forecasting the future and Understanding the Past. Addison-Wesley, Reading MA (1994).
2. Kwak, N., Chong-Ho, Ch.: Input feature selection for classification problems. Neural Networks, IEEE Transactions, Vol. 13, Issue 1 (2002) 143–159.
3. Zongker, D., Jain, A.: Algorithms for feature selection: An evaluation Pattern Recognition. Proceedings of the 13th International Conference, Vol. 2, 25-29 (1996) 18-22.
4. Ng., A Y.: On feature selection: learning with exponentially many irrelevant features as training examples. In Proc. 15th Intl. Conf. on Machines Learning (1998) 404-412.

5. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature Selection for High-Dimensional Genomic Microarray Data. Proc. of the Eighteenth International Conference in Machine Learning, ICML2001 (2001).
6. Law, M., Figueiredo, M., Jain, A.: Feature Saliency in Unsupervised Learning. Tech. Rep., Computer Science and Eng.., Michigan State Univ (2002).
7. Efron, B., Tibshirani, R., J.: Improvements on cross-validation: The .632+ bootstrap method. Amer., Statist. Assoc. 92 (1997) 548–560.
8. Stone, M., J.: An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. Royal, Statist. Soc. B39 (1977) 44–7.
9. Kohavi, R.: A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proc. of the 14th Int. Joint Conf. on A.I., Vol. 2, Canada (1995).
10. Efron, B.: Estimating the error rate of a prediction rule: improvements on cross-validation. Journal of American Statistical Association, Vol. 78 Issue. 382 (1993) 316–331.
11. Jones, A., J.: New Tools in Non-linear Modeling and Prediction. Computational Management Science, Vol. 1, Issue 2 (2004) 109-149.
12. Yang, H., H., Amari, S.: Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information, Neural Comput., vol. 9 (1997) 1457-1482.
13. A. Kraskov, H. Stögbauer, and P. Grassberger, Phys. Rev. E, in press http://arxiv.org/abs/cond-mat/0305641
14. Harald, S., Alexander, K., Sergey, A., A., Peter, G.: Least Dependent Component Analysis Based on Mutual Information. Phys. Rev. E 70, 066123 September 28 (2004).
15. URL: http://www.fz-juelich.de/nic/Forschungsgruppen/Komplexe_Systeme/software/milca-home.html.
16. Xiaoyu, L., Bing, W., K., Simon, Y., F.: Time Series Prediction Based on Fuzzy Principles. Department of Electrical & Computer Engineering. FAMU-FSU College of Engineering, Florida State University. Tallahassee, FL 32310.
17. Suykens, J., A., Van Gestel, K., T., De Brabanter, J., De.Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore, ISBN 981-238-151-1 (2002).
18. Suykens, J., A., De brabanter, K., J., Lukas, L., Vandewalle, J.: Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing, Volume 48, Issues 1-4, October 2002, Pages 85-105.
19. Suykens, J., A., K., Lukas, L., Vandewalle, J.: Sparse Least Squares Support Vector Machine Classifiers, in: Proc. of the European Symposium on Artificial Neural Networks ESANN'2000 Bruges (2000) 37-42.
20. Suykens, J., A., K., Vandewalle, J.: Training multilayer perceptron classifiers based on modified support vector method. IEEE Transactions on Neural Networks, vol. 10, no. 4 Jul. (1999) 907-911.
21. Cottrell, M., Girard, B., Rousset, P.: Forecasting of curves using a Kohonen classification. Joumal of Forecasting, 17: (5-6) (SEP-NOV) (1998) 429-439.