Mutual Information and *k*-Nearest Neighbors Approximator for Time Series Prediction

Antti Sorjamaa, Jin Hao, and Amaury Lendasse[†]

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

Abstract. This paper presents a method that combines Mutual Information and *k*-Nearest Neighbors approximator for time series prediction. Mutual Information is used for input selection. *K*-Nearest Neighbors approximator is used to improve the input selection and to provide a simple but accurate prediction method. Due to its simplicity the method is repeated to build a large number of models that are used for long-term prediction of time series. The Santa Fe A time series is used as an example.

Keywords: Time Series, Input Selection, Mutual Information, *k*-NN.

1 Introduction

In any function approximation, system identification, classification or prediction task one usually wants to find the best possible model and the best possible parameters to have a good performance. Selected model must be generalizing enough still preserving accuracy and reliability without unnecessary complexity, which increases computational load and thus calculation time. Optimal parameters must be determined for every model to be able to rank the models according to their performances.

In this paper we use Mutual Information (MI), described in Section 2, to select the inputs for direct long-term prediction of a time series. Leave-one-out (LOO) method, described in Section 3, is used to select the correct parameter for MI. Both MI and LOO rely on the *k*-Nearest Neighbors (*k*-NN) method, which is described in Section 4. Section 5 gives information about the time series prediction problem and finally the obtained experimental results, conclusions and further work are presented in Sections 6 and 7.

2 Mutual Information for Input Selection

Input selection is one of the most important issues in machine learning, especially when the number of observations is relatively small compared to the number of inputs. In practice, the necessary size of the dataset increases dramatically with the

[†] Part the work of A. Sorjamaa, J. Hao and A. Lendasse is supported by the project of New Information Processing Principles, 44886, of the Academy of Finland.

W. Duch et al. (Eds.): ICANN 2005, LNCS 3697, pp. 553 – 558, 2005.

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 approaches have been proposed [1-3]. Some of them deal with the problem of feature selection as a generalization error estimation problem. These approaches are very time consuming and may take several weeks. However, there are other approaches [4-5], which select a priori inputs based only on the dataset, as presented in this paper.

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

The MI between two variables, let say X and Y, is the amount of information obtained from X in the 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 [6].

Let's 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. Equation (1) leads to complicated integrations, so some approaches have been proposed to evaluate them numerically. In this paper, a recent estimator based on l-NN statistics is used [7] (l is used instead of k here to avoid confusion with the k appearing in section 4). 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 l nearest neighbors. MI is derived from equation (1) and is estimated as

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

where N is the size of the dataset, l is the number of nearest neighbors and $\psi(x)$ is the digamma function,

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

$\psi(1) \approx -0.5772156$ and $\langle \ldots \rangle$ denotes averages of n_x and n_y over all $1 \le i \le N$ and over all realizations of the random samples. $n_x(i)$ and $n_y(i)$ are the number of points in the region $\|x_i - x_j\| \le \varepsilon_x(i)/2$ and $\|y_i - y_j\| \le \varepsilon_y(i)/2$, $\varepsilon_x(i)$ and $\varepsilon_y(i)$ are the edge lengths of the smallest rectangle around point i containing l nearest neighbors. Software for calculating the MI based on this method can be downloaded from [8].

3 Leave-One-Out

Leave-one-out [4] is a special case of k-fold cross-validation resampling method. In k-fold cross-validation the training data is divided into k approximately equal sized sets. LOO procedure is the same as k-fold cross-validation with k equal to the size of the training set N. For each model to be tested, LOO procedure is used to calculate the generalization error estimate by removing each data point at a time from the training set, building a model with the rest of the training data and calculating the validation error with the one taken out. This procedure is done for every data point in the train-

ing set and the estimate of the generalization error is calculated as a mean of all *k*, or *N*, validation errors (4).

$$\hat{E}_{gen}(q) = \frac{\sum_{i=1}^{N} \left(h^q(x_i, \theta_i^*(q)) - y_i \right)^2}{N}, \tag{4}$$

where x_i is the i^{th} input vector from the training set, y_i is the corresponding output, h^q denotes the q^{th} tested model and $\theta_i^*(q)$ includes the model parameters without using (x_i, y_i) in the training. Finally, as a result from the LOO procedure, we select the model that gives us the smallest generalization error estimate.

4 *k*-Nearest-Neighbors Approximator

K-Nearest Neighbors approximation method is a very simple, but powerful method. It has been used in many different applications and particularly in classification tasks [9]. The key idea behind the *k*-NN is that similar input data vectors have similar output values. One has to look for a certain number of nearest neighbors, according to Euclidean distance [9], and their corresponding output values to get the output approximation. We can calculate the estimation of the outputs by using the average of the outputs of the neighbors in the neighborhood. If the pairs (x_i, y_i) represent the data with x_i as an *n*-dimensional input and y_i as a scalar output value, *k*-NN approximation is

$$\hat{y}_i = \frac{\sum_{j=1}^{k} y_{P(j)}}{k}, \tag{5}$$

where $\hat{y}_i$ represents the output estimation, $P(j)$ is the index number of the j^{th} nearest neighbor of the input x_i and k is the number of neighbors that are used. We use the same neighborhood size for every data point, so we use a global k, which must be determined. In our experiments, different k values are tested and the one which gives the minimum LOO error is selected.

5 Time Series Prediction

Time series prediction can be considered as a modeling problem [10]: a model is built between the inputs and the outputs. Then, it is used to predict the future values based on previous values. In this paper we use direct forecast to perform the long-term prediction. In order to predict the values of a time series, *M* different models are built,

$$\hat{y}(t+m) = f_m(y(t-1), y(t-2), ..., y(t-n)) \ , \tag{6}$$

with $m = 0,1,\ldots M\text{-}1$, M is the maximum horizon of prediction and f_m is the model related to time step m The input variables on the right-hand part of (6) form the regressor, where n is the regressor size.

6 Experimental Results

The dataset used in the experiments is the Santa Fe A Laser Data. To test the influence of the size of dataset on the selection of parameter l in MI calculation, two steps are followed.

The first experiment is done with 10 000 data, 9900 from which is used for training and the rest 100 for testing. In order to apply the prediction model from equation (6), we set the maximum time horizon M to 100 and the regressor size n to 10.

For time step one ($m = 0$ in equation (6)), MI is used to select the best inputs. The estimation of MI based on equation (2) is calculated with different number of neighbors, with $l = 1,\ldots 10$. All the 2^n-1 combinations of inputs are tested; the one that gives maximum MI is selected.

Finally, k-NN and LOO are used to select the l, which minimizes the LOO error, presented in Fig. 1.a.

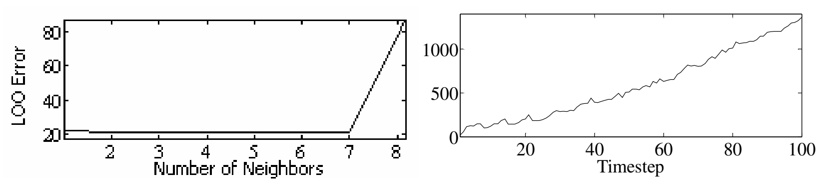

Fig. 1a. The LOO error according to different l in the MI method. **Fig. 1b.** The LOO errors according to 100 time steps.

Based on this result, $l = 5$ is chosen for the MI estimation. The input selection results for the first 50 time steps are listed in Table 1. After the input selection k-NN and LOO are used with the selected inputs for each time step and the graph of the resulting learning LOO errors are shown in Fig. 1.b.

In Fig. 2, the inputs selected with MI are used to predict 100 time steps. For each time step, f_m in equation (6) is performed using the k-NN method, and the k in equation (5) is determined by the LOO method. The real prediction error (MSE) is then calculated. For this experiment, the MSE is 2.24.

Table 1. Selected Inputs for each prediction with 10 000 data. The rows of the table represents $y(t+m)$, $m = 0,\ldots,49$, from left to right; the columns represents $y(t-n)$, $n = 1,\ldots,10$, from top to bottom. The cross mark means for one $y(t+m)$, the related input $y(t-n)$ is selected.

X	X	X				X	X	X	X		X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		X	X	X	X				X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X					X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X				X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X					X	X	X	X			X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X						X	X	X	X				X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X						X	X	X	X				X	X	X	X				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
					X	X	X	X			X	X	X	X	X	X	X	X	X		X	X	X		X		X	X	X				X	X		X	X		X	X	X	X	X	X	X		X		
			X	X	X				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

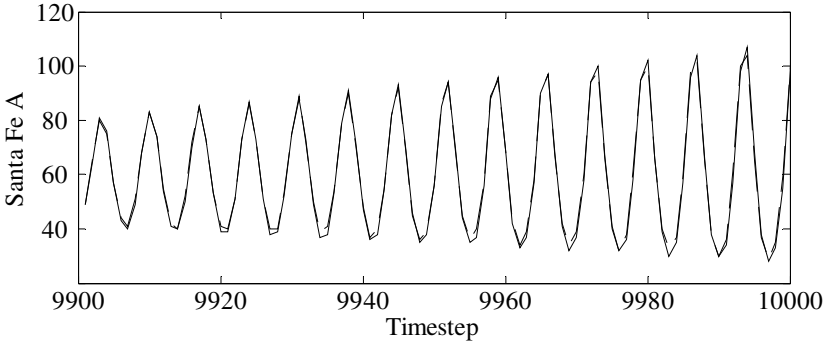

Fig. 2. 100 predictions (solid line) and the real values (dashed line)

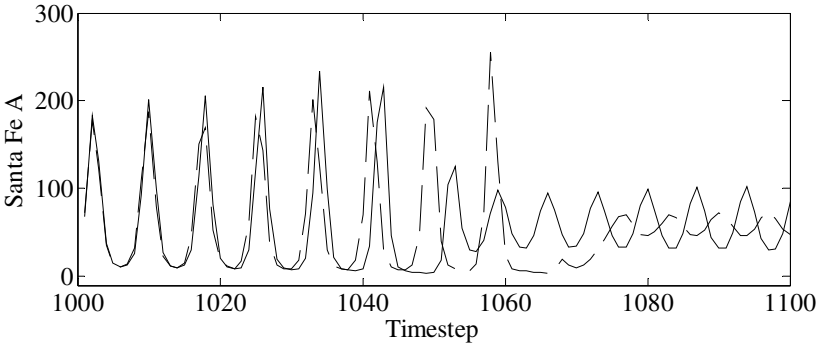

Fig. 3. 100 predictions (solid line) and the real values (dashed line)

In the second experiment, first 1000 data points are used for training and the next 100 points for testing. The procedure follows the first experiment. Based on the LOO error according to different l in the estimation of MI, $l = 2$ is chosen.

The prediction using *k*-NN and LOO based on the selected inputs by MI is plotted in Fig. 3.

7 Conclusions and Further Work

In this paper, MI is used to select the inputs for time series prediction problem. It has been illustrated with the experiments that the *k*-NN approximator and LOO method can be used to tune the main parameter of the MI estimator.

k-NN has also been used as an approximation model itself. Although Fig. 3 shows that after step 50, the jump of Santa Fe A Laser Data is not predicted correctly, the results are accurate in other parts. It is also possible to use another regression model to improve the quality of the predictions (Multilayer Perceptrons, Radial Basis Function Networks, Support Vector Machines, etc.). However, the advantage of the *k*-NN

approximators is that it is possible to build a large number of models to perform a direct prediction of a time series in a quite reasonable time.

In the future, we will study different algorithms for estimating the MI and their possible implementations to input selection problems. On the other hand, the implementation of input selection methods directly to *k*-NN approach will also be studied.

References

1. Kwak, N., Chong-Ho, Ch.: Input feature selection for classification problems. Neural Networks, IEEE Transactions, Vol. 13, Issue 1 (2002) 143–159.
2. 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.
3. 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).
4. 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).
5. Jones, A., J.: New Tools in Non-linear Modeling and Prediction. Computational Management Science, Vol. 1, Issue 2 (2004) 109-149.
6. 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.
7. Alexander, K., Harald, S., Peter, G.: Estimating Mutual Information. John-von-Neumann Institute for Computing, Germany, D-52425. (2004).
8. URL: http://tinyurl.com/bj73w.
9. Bishop C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995).
10. 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.