

# A NEW TIME SERIES CLASSIFICATION APPROACH

Konstantinos N. Plataniotis

Dimitrios Androutsos

Anastasios N. Venetsanopoulos<sup>1</sup>

<sup>1</sup> Department of Electrical and Computer Engineering  
University of Toronto, Toronto, ON, M5S 3G4, CANADA  
E-mail: anv@dsp.toronto.edu  
URL: <http://www.comm.toronto.edu/~dsp/dsp.html>

## ABSTRACT

A new approach to the problem of time series classification is discussed in this paper. A new adaptive classification scheme is introduced and compared with existing approaches, such as the Bayesian approach and the Incremental Credit Assignment approach. Simulation results are included to demonstrate the effectiveness of the new methodology.

## 1. INTRODUCTION

Adaptive classification of time series is a problem which is encountered in many real word applications, such as non-linear systems with unknown parameters, target tracking, reconfigurable systems, EEG diagnosis and speech identification [1]- [2]. The usual formulation of such a problem is that it is assumed that the observable time series  $y(k)$ ,  $k = 1, 2, \dots$  is generated by an unknown source model  $S(\theta_i)$  where  $\theta_i$  is a parameter which completely characterizes the model taking values in a finite set  $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$ . The objective of an adaptive classification scheme is to recursively identify the source model which generates the time series by selecting the optimal value for the parameter  $\theta$  [1].

Such schemes achieve their objective by utilizing a bank of predictors each matched to an assumed source model labeled by the appropriate parameter  $S(\theta_i)$ . In the sequence, each predictor is used to generate an on-line estimate of the next available observation of the series  $y(k)$  utilizing the past values  $(y(k-1), y(k-2), \dots, y(1))$ . Based on the corresponding prediction errors the model which best fits the data is selected by the classification procedure. In this paper, we utilize neural network based predictors, each one trained off-line with labeled data from a particular source model, to generate the corresponding prediction errors used by the classification scheme.

The first approach utilized for the solution of the above problem is the Bayesian one. In the context of the Bayesian solution, the random variable  $\theta$  is assumed taking values in the set  $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$ . The Bayesian classification scheme assumes that at time index  $k$  the sequence  $(y(k), y(k-1), y(k-2), \dots, y(2), y(1))$  has been generated by the source model  $S(\hat{\theta}(k))$  (as it is approximated by the corresponding neural predictor), with  $\hat{\theta}(k) = \theta_j$  which has

the maximum a-posteriori probability

$$p_j(k) = Prob(\theta = \theta_j | (y(k), y(k-1), \dots, y(1))) \quad (1)$$

with a-priori probabilities  $p_j(0) = Prob(\theta = \theta_j | k = 0) = \frac{1}{n}$  considering all models to be equiprobable. The maximum a-posteriori probability can be calculated recursively using Baye's rule, presented in [3] and applied initially within the context of neural networks in [1], [4] and later used in [2], [5].

Following the approach in [1] the a-posteriori probability for the  $j^{th}$  neural predictor, with  $j = 1, 2, \dots, n$  is calculated recursively as follows:

$$p_j(k+1) = \frac{f(y(k+1) | ((y(k), y(k-1), \dots), \theta_j) p_j(k))}{\sum_{i=1}^n f(y(k+1) | ((y(k), y(k-1), \dots), \theta_i) p_i(k))} \quad (2)$$

The calculation of the a-posteriori probability  $p_j(k)$  depends on the form of the neural network predictor  $\hat{y}_j(k) = \hat{f}((y(k), y(k-1), \dots, y(1)), \theta_j)$  used to approximate the actual time series  $y(k)$  under the assumption that the time series is generated by the source model  $S(\theta_i)$  for  $j = 1, 2, \dots, n$ . Multilayer perceptrons with sigmoidal neurons trained with the backpropagation rule are used as predictors in this paper.

In the Bayesian context the time series is attributed to the source that maximizes the a-posteriori probability. However, there are some problems associated with the Bayesian methodology. Most notably, these are due to the computational complexity of its implementation and its sensitivity to outliers. To overcome these limitations, a new classification methodology, the so-called Incremental Credit Assignment (ICRA) was introduced recently [5]. Starting from the Bayesian calculation of the a-posteriori probabilities, this classification schemes classifies the time series to the  $i^{th}$  source which maximizes the following recursively calculated quantity:

$$q_i(k) = q_i(k-1)[1 + \gamma(g(e(k)_i) - \sum_{j=1}^n q_j(k-1)g(e(k)_j))] \quad (3)$$

where  $q(k)_i$  is the quantity associated with the  $i^{th}$  source at time instant  $k$  and  $g(e(k))$  is the error in prediction defined as in the case of Bayesian learning.

Although this approach seems more general than the Bayesian one, it is still sensitive to outliers, since the same methodology is used to calculate the errors between the actual source and the  $i^{th}$  assumed model. In addition, it is

obvious through inspection that the ICRA credit function, is as computationally complex as the Bayesian approach of (2). Therefore, a new classification scheme which is robust to non-Gaussian noise and computationally attractive should be introduced. Such an approach is discussed in the next section.

## 2. NEAREST NEIGHBOR CLASSIFICATION

In this section we consider an improved and robust classification scheme. The new scheme is simple, robust to possible outliers, and is shown to result in classification results comparable to that of the Bayesian rule with a simpler implementation.

As for the Bayesian approach, a bank of neural network-based predictors are trained off-line using labeled data. In the sequence, during the on-line phase we define the instantaneous prediction error  $\hat{e}_j = \Phi(y(k) - \hat{y}(k))$ ,  $k = 1, 2, \dots$ ,  $j = 1, 2, \dots, n$ , where  $\Phi(\cdot)$  is a robust function of the prediction error selected to reduce the effects of any present outlier. The Bayesian classification scheme discussed in the introduction utilizes the square norm (Euclidean distance)  $\hat{e}_j = (y(k) - \hat{y}(k))^2$  to calculate the instantaneous prediction error [5]. However, if outliers are assumed present other error measures more robust to outliers, such as the  $L_1$  (*City Block distance*) can be used instead. In this case, the instantaneous prediction error is defined as:

$$\hat{e}_j = |(y(k) - \hat{y}(k))| \quad (4)$$

The instantaneous prediction error  $\hat{e}_j(k)$  depends not only on the current measurement  $y(k)$  but also on past measurements which have been used to form each predictor. Therefore, an aggregate error is utilized in the proposed classification scheme to predict the best signal source. For the  $j^{th}$  predictor, such an error can be defined as follows:

$$d_j(k) = \gamma d_j(k-1) + \hat{e}_j(k) \quad (5)$$

where  $\gamma$  is a decay factor slightly less than 1 for discounting past data, and  $j = 1, 2, \dots, n$ ,  $k = 1, 2, \dots$ .

Based on the corresponding aggregate prediction error a credibility weight  $w_j$  is then assigned to the  $j^{th}$  assumed model.

Each one of the weights is a function of the distance between the model under consideration and all other models under consideration. In this paper, a neighbor weighting function is utilized to assign weights to each one of the competing models. Thus, the credibility weights are calculated as follows:

$$w_i = 1 \quad \text{if } d_{(i)} = d_{(1)}, \quad w_i = 0 \quad \text{if } d_{(i)} = d_{(s)}$$

and

$$w_i = \frac{(d_{(s)} - d_{(i)}) + \alpha(d_{(s)} - d_{(1)})}{(1 + \alpha)(d_{(s)} - d_{(1)})}, \quad (6)$$

otherwise

where  $j = 1, 2, \dots, n$  and  $d_{max}(k)$ ,  $d_{min}(k)$  are the corresponding maximum/minimum aggregated prediction errors.

The regulating parameter  $\alpha$  takes values in the interval  $[0, 1)$ . These values are either fixed reflecting the designers

confidence about the distances among the image vectors or can be calculated adaptively using the equation:

$$\alpha = \frac{1}{d_{(s)} - d_{(i)}} \quad (7)$$

The following points should be made regarding the proposed classification methodology:

- The value of the weight expresses the degree to which the model specified by the parameter  $\theta_i$  is close to the actual model which generates the sequence, and far away from the worst model. From the definition of the weights it is obvious that the model with the minimum prediction error will be assigned the highest credibility weight.
- It is evident that the outcome of the proposed classification scheme depends on the choice of the error criterion selected to calculate the instantaneous error between the assumed model and the observed actual value. Error measures robust to outliers, such as the  $L_1$  norm used here can reduce the effect of any impulses present, resulting in a decision scheme immune to high levels on non-Gaussian noise.
- Finally, the new scheme is recursive, thus it is appropriate for on-line classification of dynamic series and can be implemented in parallel using only adders and ranking elements.

## 3. APPLICATIONS

To demonstrate the effectiveness of the proposed methodology, the problem of logistic time series detection is considered. A logistic time series is generated by the following difference equation [4], [5]:

$$x(k) = a(x(k-1)(1-x(k-1))) \quad (8)$$

$$y(k) = x(k) + w(k) \quad (9)$$

with  $k = 1, 2, \dots$  and  $w(k)$  is zero-mean white noise, uniformly distributed in the interval  $[-0.25, 0.25]$ . During the actual implementation phase a source switching takes place. The actual time series is generated again using the above model. For the first half of the data (500 steps) the actual value of the parameter is  $a = 4.0$ . For the remaining 500 steps the parameter value used to generate the actual series is  $a = 3.75$ . During the off-line training phase three neural predictors have been trained using the following values; Predictor I: ( $a=3.5$ ), Predictor II: ( $a=4.0$ ), Predictor III: ( $a=3.75$ ). Three time series classification schemes are compared in this paper. The Bayesian approach (BSC) discussed in [4], [1] and [2], the Incremental Credit Assignment (ICRA) approach of [5] and the Nearest neighbor approach (NNSC) of (6) with  $\alpha = 0$ . The classification performance for the different schemes is plotted in Figs. 1-3 and the prediction performance of the proposed NNSC approach in terms of the Normalized Mean Square Error (NMSE) for 1000 steps averaged over 1000 Monte Carlo runs is depicted in Fig. 4. From the graphs listed here, it easily can be seen that the new scheme immediately identifies the switch in the active source model. On the contrary, both the Bayesian approach as well as the ICRA approach of [5] took more than 150 steps to identify the correct source/model.

#### 4. CONCLUSION

In conclusion, the proposed nearest neighbor classification scheme outperforms other decision schemes used for the problem of time series classification, offers superior performance in the case of switching sources without complicate calculations and expensive implementation requirements.

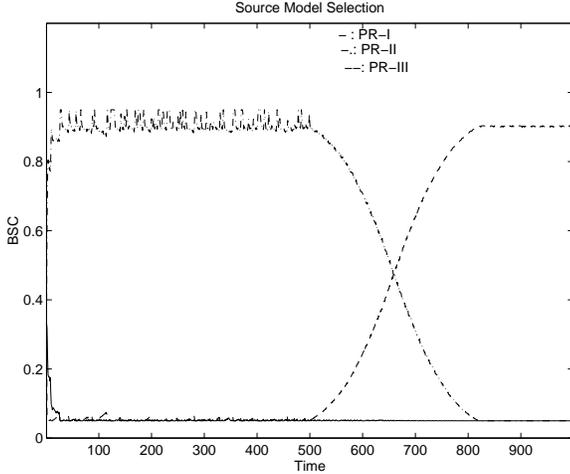


Figure 1. Model Selection, Bayesian Approach

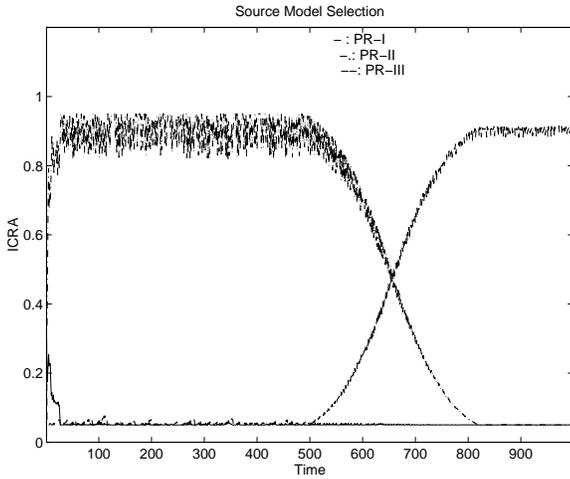


Figure 2. Model Selection, ICRA Approach

#### REFERENCES

- [1] K.N. Plataniotis, *Distributed Parallel Processing State Estimation Algorithms*, Ph.D Dissertation, Florida Institute of Technology, Melbourne, FL., USA, 1994.
- [2] V. Petridis, A. Kehagias, Modular Neural Networks for MAP Classification of Time Series and the Partition A lgorithm, *IEEE Trans. on Neural Networks*, vol. 7, no. 1, pp. 73-85, 1996.
- [3] C.G. Hilborn, D.G. Lainiotis, Optimal estimation in the presence of unknown parameters, *IEEE Trans. on System Science and Cybernetics*, vol. 5, pp. 38-43, 1969.

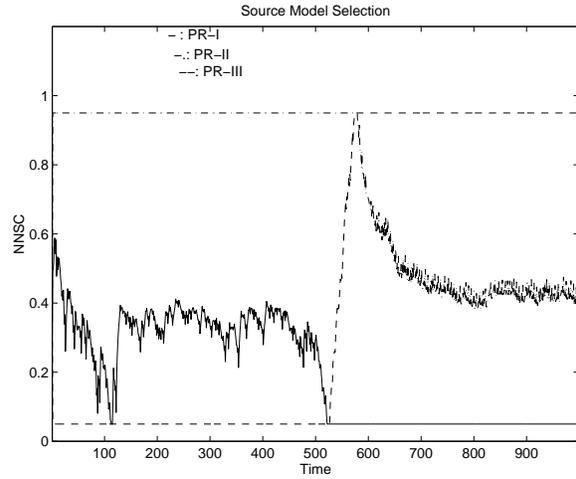


Figure 3. Model Selection, Nearest Neighbor Approach

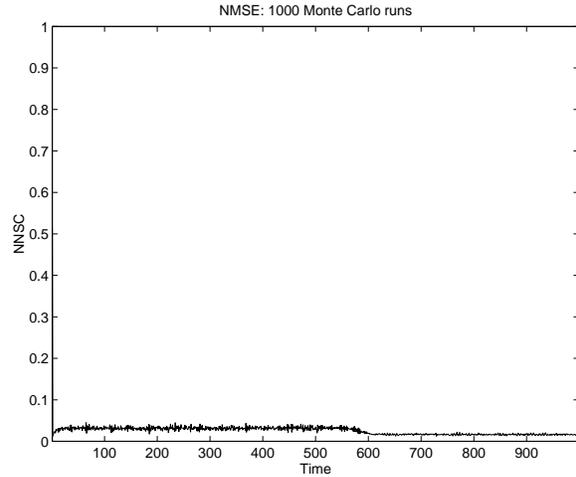


Figure 4. Nearest Neighbor Approach, NMSE, 1000 Monte Carlo

- [4] D.G. Lainiotis, K.N. Plataniotis, Adaptive Dynamic Neural Network Estimation, *Proceedings of the IEEE International Conference on Neural Networks, ICNN-94*, vol. III, pp. 4710-4717, 1994.
- [5] V. Petridis, A. Kehagias, A Recurrent Network Implementation of Time Series Classification, *Neural Computation*, vol. 8, pp. 357-372, 1996.