Ath. Kehagias and V. Petridis.

**"Predictive Modular Neural Networks for Unsupervised Segmentation
of Switching Time Series: the Data Allocation Problem".**

Predictive Modular Neural Networks for Unsupervised Segmentation of Switching Time Series: the Data Allocation Problem

Ath. Kehagias and V. Petridis

July 24, 2001

Abstract

In this paper we explore some aspects of the problem of on-line, unsupervised learning of a *switching* time series, i.e. a time series which is generated by a combination of several, alternately activated sources. This learning problem can be solved by a two-stage approach: (a) separating of the incoming data to several datasets (one dataset corresponding to each source); (b) developing one model per dataset (i.e. one model per source). We introduce a general *data allocation* methodology which combines the two steps into an iterative scheme: existing models compete for the incoming data; data assigned to each model are used to refine the model. We distinguish between two modes of data allocation: in *parallel* data allocation every incoming datablock is allocated to the model with lowest prediction error; in *serial* data allocation the incoming datablock is allocated to the *first* model with prediction error below a prespecified threshold. We present sufficient conditions for asymptotically correct allocation of the data. We also present numerical experiments to support our theoretical analysis.

1   Introduction

A *switching time series* is a time series generated by a combination of several sources, each source being activated for a particular time interval. The problem which motivates this paper is *learning a switching time series,* and in particular developing input / output models for its constituent sources.

A large number of methods have been developed to solve this problem; we are particularly interested in *on-line, unsupervised* methods using *predictive models*. The term "predictive model" denotes an input / output model of the source behavior. As indicated in the title of this paper, we present our method in the context of *predictive modular neural networks*, a family of models which we have presented in past publications [10, 11, 18, 19, 20]. However, most of our results can be utilized in the context of more general *modular* systems. The main feature of such systems is that each module is *specialized* in describing the input / output behavior of a particular source.

Methods of this type are used extensively in the neural and control literature and are presented under various names such as local experts [7, 8, 17], local controllers [2], multiple models [13, 15, 16] or regime models [3, 4, 5, 9]. The unsupervised problem is considerably harder than the *supervised*

one. In the latter case, labeled data are available for an off-line training phase. I.e. two sequences of data points are available; the first sequence is the actual time series observations and the second sequence indicates which source is active at every time instant. The data can be separated into distinct datasets, one dataset per source, and each dataset is used to train a model of the respective source [10, 11, 18, 19, 20]. In the unsupervised problem, labeled data are *not* available; part of the problem is recognizing which source generated each observation. As soon as this is accomplished, the problem is essentially reduced to the supervised case. An approach to the problem is represented by the Hidden Markov Model methodology (HMM), where each source is mapped to a state of an unobservable Markov chain [6]. However, HMM usually involves relatively simple and *static* source models for every hidden state. There are extensions of HMM which involve dynamic input / output models [12], but usually they operate offline. In this paper we are interested in an online solution to the problem.

In the sequel we will use the term *data allocation* to refer to the problem of assigning data to sources. In this paper we introduce a general *data allocation* iterative scheme. Our scheme can be seen as an generalization of the k-means clustering method [14]. The basic steps of our scheme are the following.

1. A small number of predictive models is randomly initialized.

2. A prediction regarding the next incoming datum is obtained from each model.

3. The incoming datum is allocated to the model which yielded the most accurate prediction.

4. The expanding datasets are used to preiodically retrain the models.

In Section 2 we formally describe the data allocation problem. In Section 3 we present informally two data allocation algorithms and discuss the *emergence of specialization* of predictors. In Section 4 we formally present two data allocation algorithms. In Section 5 we present sufficient conditions for asymptically correct data allocation. In Section 6 we present numerical examples of the use of our algorithms. Section 7 summarizes our conclusions.

2 The Data Allocation Problem

Let us present a more formal description of the data allocation problem. Consider an *observable* time series y_t, $t = 1, 2, ...$ which is generated by a *source* time series z_t; y_t takes values in $\mathbf{R}$ (extensions to vector valued time series are immediate) and z_t takes values in a finite *source* set $\Theta = \{1, 2, ..., K\}$. At time t, y_t depends on the value of z_t, as well as on y_{t-1}, y_{t-2}, ... according to the equation

$$y_t = F_{z_t}(y_{t-1}, y_{t-2}, ..., y_{t-M}).$$

Hence the time series is generated by the combination of K functions: $F_1(.)$, $F_2(.)$, ... , $F_K(.)$. The number K , as well as the functions $F_k(.)$ are unknown. The task is to obtain functions $f_k(.)$ such that

$f_k(.)$ approximates $F_k(.)$ (for $k = 1, 2, ..., K$).

If the source time series z_t were observable, then it would be easy obtain input/output models $f_k(.)$ of the time series y_1, y_2, ... (for $k = 1, 2, ..., K$) as follows: first the observed data y_1, y_2, ... , y_t, ... would be separated into K datasets, one group corresponding to each observed value of z_t (i.e. to each active source); then the k-th dataset would be used to train a sigmoid feedforward neural network $f_k(\cdot)$ to approximate $F_k(\cdot)$ (such an approximation is possible in light of the universal approximation properties of sigmoid feedforward neural networks).

However, since z_t is not observable, it is not obvious how to group the observed data y_1, y_2, ... , y_t, ... The first step in the source identification process must be *data allocation*, i.e. the separation of the observed data into groups, one group corresponding to each active source. Data allocation could be performed by using a time series classification algorithm (for instance see [10, 11, 18, 19, 20]); however, such classification algorithms require knowledge of the models $f_k(.)$. But training the $f_k(.)$'s presupposes separation of the data. It appears that we have entered a vicious circle.

In Section 4 we present algorithms which can break out of this circle by implementing an online, unsupervised process which effect data allocation and predictor training simultaneously. Before formally presenting these data allocation algorithms, let us present an informal description of *emergent specialization*.

3   Emergent Specialization

In this section we discuss the key ideas motivating our algorithms, especially the idea of *emergent specialization*. The discussion is informal and we mostly restrict ourselves to examples involving two sources; a more formal and general presentation is given in Sections 4 and 5.

3.1   Emergent Specialization by Parallel Data Allocation

We now present a very simple example, which sets the stage for our data allocation algorithms. This initial example involves a time series generated by two sources (call them source A and source B) and two predictors (call them predictor no.1 and predictor no.2) of the general form

$$y_t^k = f(y_{t-1}, y_{t-2}, ..., y_{t-M}; w_k) \tag{1}$$

where $k = 1, 2$; $f(\cdot; w)$ is a general sigmoid, feedforward neural networks with weights w; and w_1, w_2 are *randomly* initialized weights. We proceed to informally describe an algorithm which can be reasonably expected to produce w_1 and w_2 values such that each predictor accurately represents the behavior of one source.

Suppose that we observe the time series samples y_t at times $t = 1, 2, ..., M$; at time $t = M + 1$ we use eq.(1) to compute y_t^1 and y_t^2. We compute the prediction errors $e_t^k = |y_{M+1} - y_{M+1}^k|$ ($k = 1, 2$) and allocate y_{M+1} to the dataset of the predictor with smallest e_t^k. We repeat the process for times

$t = M + 2, M + 3, ..., M + L$, randomly forming two datasets; at $t = M + L$ we use the datasets to train the respective predictors. We repeat the process for a large number of steps, retraining the predictors every L steps.

What can we expect regarding the behavior of the predictors? Initially, neither predictor is "specialized" in any particular source and is equally likely to "accept" or "reject" data from either source. However, just before the first training period, one of the two predictors will *happen* to have collected more data from source A than source B. Assume this is predictor no.1 As soon as predictor no.1 is trained on its dataset, it will develop a small bias towards more accurate prediction of source A data. This will have two consequences. First, in the next round of data allocation, predictor no.1 will be likely to predict more accurately source A data than predictor no.2; hence it will collect even more source A data. Secondly, *provided the two sources have sufficiently different input / output behavior,* predictor no.1 will also be likely to predict source B data *less* accurately than predictor no.2 As a result, predictor no.2 will be more likely to get more source B data in its dataset. This behavior will be more accentuated after the second retraining. In short, it appears reasonable that specialization will reinforce itself and that, asymptotically, each predictor will tend to accept data from only one source. At this point correct data allocation has been effected and, in addition, we have obtained one accurate input/output model for each source.

In conclusion, we have presented a data allocation scheme which can be summarized as follows:

$$\text{if } \left| y_t - y_t^1 \right| \leq \left| y_t - y_t^2 \right| \quad \text{then } y_t \text{ is allocated to pred. no.1;}$$
$$\text{if } \left| y_t - y_t^1 \right| > \left| y_t - y_t^2 \right| \quad \text{then } y_t \text{ is allocated to pred. no.2.}$$

This scheme will be called *parallel data allocation* (in contradistinction to *serial data allocation*, to be presented in Section 3.2). This scheme can be expanded to the case of many sources, as will be seen in Section 3.3.

Finally, let us remark that the argument presented in this section *does not guarantee* emergence of specialization, which may be thwarted by several factors. An informal discussion will be presented in Section 3.4 and a rigorous analysis in Section 5, which will establish conditions sufficient for the emergence of asymptotically correct data allocation.

3.2 Emergent Specialization by Serial Data Allocation

We now present an alternative data allocation scheme, the so-called *serial data allocation.* Continuing in the context of the two-sources example of Section 3.1, we can present the serial data allocation scheme concisely as follows:

$$\text{if } \left| y_t - y_t^1 \right| \leq d \quad \text{then } y_t \text{ is allocated to pred. no.1;}$$
$$\text{if } \left| y_t - y_t^1 \right| > d \quad \text{then } y_t \text{ is allocated to pred. no.2.}$$

In other words, in this case each incoming datum is not assigned to the *best* predictor, but to the *first* predictor if it has prediction error below a prespecified threshold d; otherwise to the second predictor.

The argument regarding emergence specialization is very similar to the one regarding parallel data allocation, and hence is given in summary. Initially, one predictor will keep receiving most incoming data (most likely predictor no.2). Assuming a fairly slow switching rate between sources A and B, this predictor will accumulate mostly data from one source and hence will specialize in one source, absorbing all its data and rejecting data from the remaining source. These data will then end up in the second predictor's dataset, so the second predictor will specialize in the remaining source.

We should remark that, in this case, too, the above argument can be extended to more than two sources (see Section 3.3) and can also be made more rigorous (see Section 5).

3.3 The Case of Many Sources

Let us now suppose that more than two sources are involved in the generation of the time series, let us introduce modifications of the data allocation schemes presented in Sections 3.1 and 3.2 and let us discuss informally their behavior.

In the case of parallel data allocation, the modification of the algorithm is as follows: more than two models are randomly initialized and every incoming datum is predicted by each model; the datum is assigned to the model with smallest prediction error (a winner-take-all strategy). In case the number of sources is initially unknown, one can start with a small number of models and keep adding models whenever the prediction error of the existing ones is above a certain threshold. This threshold should be variable, to reflect the fact that predictive performance is expected to improve as more data are allocated to each model.

In the case of serial data allocation, it is perhaps most appropriate to start with a single model and keep adding models in tandem whenever the prediction error of all existing models is above a variable threshold. One can visualize this arrangement of the models as a series of filters, where each filter (model) "absorbs" the data to which it is tuned.

In both the serial and parallel case, specialization can be expected to emerge in much the same manner as in the case of two sources, i.e. each model starts with a random bias towards some source, and this bias is reinforced as more data become available. However, at this point it is appropriate to also consider factors which may have an adverse effect upon specialization; this will be the subject of Section 3.4.

3.4 Additional Considerations

One can easily visualize situations where specialization will not occur. For instance, consider a time series generated by two sources, which have a *very similar input / output behavior*. Then it is conceivable that *one* predictor will be able to model accurately the behavior of both sources. Hence all data will be allocated to this predictor. In other words, we may obtain a predictor which is a satisfactory

input/output model of both sources, but we will never be aware of the fact that two sources have been active. In case we are interested in classification applications, this may be a serious problem. It must be emphasized that similarity of input/output behavior of two or more sources is *relative*, not *absolute*. In particular, it depends strongly on the type of predictors used. A predictor with rich structure and a large number of parameters may be capable of simultaneously capturing the input/output behavior of two fairly distinct sources; conversely a predictor with few parameters may furnish a poor model of even two fairly similar sources. This may be expressed more formally: it can be expected that the data allocation will succeed when the predictor *capacity* is not much higher than the source *complexity*.

Even if the constituent sources have sufficiently different behavior, specialization may fail to emerge in case switching betwene sources takes place at a very rapid rate. In such a case, at the initial stages of the data allocation process, before a predictor has the chance to specialize in a single source, it will collect a mixed dataset. It should be added that *slow switching* is essential in time series modeling and is usually assumed to hold [11, 17]. Of course, this is also a relative issue, i.e. in situations where one may suspect "fast" switching, it will be necessary to increase the sampling rate, so that the time series switches slowly in relation to the learning algorithm.

4   Data Allocation Algorithms

We now present a parallel data allocation algorithm (Section 4.1) and a serial data allocation algorithm (Section 4.2). Both algorithms apply to the case of an unknown but finite number of sources.

The algorithms operate on *data blocks* rather than on isolated data points; for this reason some remarks on notation ar necessary at this point. We symbolize data blocks of length M as follows

$$Y_t = [y_{(t-1) \cdot M+1} ... y_{t \cdot M}], Z_t = [z_{(t-1) \cdot M+1} ... z_{t \cdot M}], E_t = [e_{(t-1) \cdot M+1} ... e_{t \cdot M}].$$

For the prediction error we have $E_t^k = |Y_t - Y_t^k|$; since Y_t and Y_t^k are vectors, $|\cdot|$ will be understoold as a vector norm. Finally, the representation of predictor output will be $Y_t^k = f(Y_t; w_k)$, which is understood in a vectorial sense, i.e. Y_t^k has components

$$y_{(t-1) \cdot M+1}^k = f(y_{(t-1) \cdot M+1}; w_k), ..., y_{t \cdot M}^k = f(y_{t \cdot M}; w_k).$$

Notice also that we assume predictors of first order (but this can be easily extended for predictors of higher order) and that the k-dependence is now introduced through the weight vectors w_k. We assume a sigmoid feedforward form for the $f(\cdot; w)$ functions; the exposition remains unchanged in case of other functional families.

4.1 Parallel Data Allocation Algorithm

This algorithm requires the following inputs: L (retraining period), M (size of data block), J (number of training iterations), N_c (size of training set) and d (prediction error threshold) and, of course, observations of the time series, which appear in the form of datablocks Y_1, Y_2, ... The significance of these parameters has been discussed, with the exception of N_c, which is a dataset growing limit; its significance is discussed in Section 4.3, where the importance of the remaining parameters is also discussed.

PARALLEL DATA ALLOCATION ALGORITHM

Initialization

Set $K = 2$. Initialize randomly 2 predictors described by the functions $f_k(\cdot; w_k^{(0)})$, $k = 1, 2$.

Main Routine

For $t = 1, 2, ...$

    For $k = 1, ..., K$.

        Compute prediction $Y_t^k = f(Y_{t-1}; w_k^{(t-1)})$.

        Compute prediction error $\left|E_t^k\right| = \left|Y_t - Y_t^k\right|$.

    Next k

    Set $k^* = \arg\min_{k=1,2,...,K} \left|E_t^k\right|$.

    Assign the current block Y_t to the predictor k^*.

    If the number of data blocks assigned to predictor k^* is greater than N_c then:

        delete the first data block.

    End If

    If $t = n \cdot L$ (for any n) then:

        For $k = 1, ..., K$.

            Retrain the k-th predictor (using all data assigned to it) for J iterations and so obtain a new $f(\cdot; w_k^{(t)})$ neural network.

        Next k

    End If

    For $k = 1, ..., K$

        If the size of the k-th training set is larger than N_c and $\frac{\sum_{s=1}^{t}\left|E_s^k\right|}{t} > d$.

            Set $K = K + 1$

Replace the k-th predictor by two identical copies of itself.

Allocate all data of the replaced predictor to both new predictors.

End If

If the size of the k-th training set is larger than N_c.

delete the first data block.

End If

Next k

Next t

4.2 Serial Data Allocation Algorithm

The inputs to this algorithm are the same as in Section 4.1. The remarks in Section 3.3 also apply to this algorithm.

SERIAL DATA ALLOCATION ALGORITHM
Initialization

Set $K = 1$. Initialize randomly one predictor, described by the function $f(\cdot; w_1^{(0)})$.

Main Routine

For $t = 1, 2, ...$

For $k = 1, ..., K$.

Compute $Y_t^k = f(Y_{t-1}; w_k^{(t-1)})$.
Compute $|E_t^k| = |Y_t - Y_t^k|$.
If $\left| E_t^k \right| < d$ then

assign the current block Y_t to the k-th predictor.

End If

If the number of data blocks assigned to predictor k is greater than N_c then:

delete the first data block.

End If

Next k.

If Y_t has not been assigned to any predictor,

8

set $K = K + 1$

assign Y_t to the K-th predictor.

End If

If $t = n \cdot L$ (for any n) then:

Loop for $k = 1, ..., K$.

Retrain the k-th predictor (using all data assigned to it) for J iterations and so obtain a new predictor $f(\cdot; w_k^{(t)})$.

Next k

End If

Next t

4.3 Implementation Issues

Several parameters are involved in the data allocation aglorithms. In this section we discuss issues related to their significance and choice of their values.

1. M (size of data block) is related to the switching rate of the constituent sources. Let T_s denote the minimum number of time steps between two successive source switchings. While T_s is unknown, we operate on the assumption of *slow* switching, which means that T_s will be large compared to M. Since the M data points included in a block will all be assigned to the same predictor, it is obviously desirable that they have been generated by the same source. In practice this cannot be guaranteed. In general, a small value of M will increase the likelihood that most blocks contain data from a single source. On the other hand, it has been found that small M leads to an essentially random assignment of data points to sources, especially in the initial stages of segmentation, when the predictors have not specialized sufficiently.

2. The number of training iterations J is relevant in case training is performed by an iterative procedure (e.g. backpropagation). In cases where exact training is possible (e.g. for linear predictors) it is not necessary to perform J training iterations. J should be taken relatively small, since the predictors must not be overspecialized in the early phases of the algorithm, when relatively few data points are available. The choice of J is closely connected to that of L; if L is relatively small (frequent retraining) then J can be small, too; i.e. it may be preferable to retrain the predictors often and by small increments.

3. The parameter N_c (size of training set) is introduced to enable the online use of the algorithm. If there was no limit to the amount of data kept in the data set, retraining would take an increasing amount of time as more data became available.

4. We have no specific recommendations to make regarding the choice of the threshold d; we have found that, within reasonable bounds, its exact values are not crucial to the performance of the algorithms.

5. The algorithm is *competitive:* a data block may be allocated to the k-th predictor even when the error $|E_t^k|$ is large; what matters is that it is smaller than the remaining prediction errors. Note that computation of predictions and training of predictors can be performed in parallel, resulting in significant execution speedup.

6. The algorithm can be modified so as to allow merging of predictors. Merging can occur in case the errors of two predictors are consistently of small and of comparable size for the same observations y_t. For simplicity of presentation we did not include this option in the above description of the algorithm, but it should be obvious how to implement it.

5 Convergence Results

In this section we present rigorous convergence results for both parallel and serial data allocation algorithms. These results pertain to the case of simplified algorithms, which involve two sources and two predictors; for the case of many sources we present a heuristic argument and some indication on how this can be made more rigorous. Even in the simplified cases considered here, the mathematical analysis is rather complex and the proofs of our results quite lengthy. Hence we will only present the setup of the problem and our main convergence theorems; the reader is referred to the book [20] for details.

5.1 Two sources, Parallel Data Allocation

5.1.1 Some Important Processes

We have already introduced the block processes Y_t, Z_t, Y_t^k, E_t^k $(k = 1, 2)$ in Section 4. In the case of two active sources the source process Z_t takes values in $\{1, 2\}$. Recall that time series generation takes place according to the equation

$$Y_t = F_{Z_t}(Y_{t-1}, Y_{t-2}, ...).$$

We assume that every source switching take place independently, according to a fixed probability distribution. In other words, the probability of source i being active at time t is independent of time (the source process is *stationary*) and is denoted (for $i = 1, 2$) by $\pi_i \doteq \Pr(Z_t = i)$. We assume that $\pi_1 > 0$, $\pi_2 > 0$, i.e. that both sources are really active. Finally, since Z_t is a block process, we essentially assume that source switchings can only take place at discrete time values, i.e. that each data block has been generated by a single source. This is in accordance with the slow switching hypothesis.

The *data allocation* processes N_t^{ij} $(i, j = 1, 2)$ are defined by

$$N_t^{11} : \quad \text{no. of source 1 data assigned to predictor 1 at time } t;$$
$$N_t^{12} : \quad \text{no. of source 1 data assigned to predictor 2 at time } t;$$
$$N_t^{21} : \quad \text{no. of source 2 data assigned to predictor 1 at time } t;$$
$$N_t^{22} : \quad \text{no. of source 2 data assigned to predictor 2 at time } t.$$

(Note that $N_0^{ij} = 0$, for $i, j = 1, 2$.) Next, the *specialization* process X_t is defined by

$$X_t \doteq \left(N_t^{11} - N_t^{21} \right) + \left(N_t^{22} - N_t^{12} \right).$$

(So we have $X_0 = 0$.) The significance of the X_t process will be explained in the next section.

5.1.2   Some Important Assumptions

The significance of X_t is best understood by considering the following possibilities.

1. If X_t is positive and large, then either predictor 1 *specializes* in source 1 $(N_t^{11} \gg N_t^{21})$ or predictor 2 *specializes* in source 2 $(N_t^{22} \gg N_t^{12})$, or both.

2. If X_t is negative and large, then either predictor 1 specializes in source 2 $(N_t^{21} \gg N_t^{11})$ or predictor 2 specializes in source 1 $(N_t^{12} \gg N_t^{22})$, or both.

It follows that: *if* the *absolute* value of X_t is large, *then* at least one of the two predictors specializes in one of the two sources. When a predictor specializes in a source, it is expected that it will tend to accept data from this source and reject all other data. It may be expected that, under certain conditions, this process will reinforce itself, resulting, in the limit of infinitely many samples, in "absolute" specialization of both predictors. To test this conjecture mathematically, we need a more precise model of the evolution of X_t. Two components are required for such a model. First, we assume that

A0   For $i = 1, 2$ the following is true:

$$\text{Pr}(\text{Pred. nr.}i \text{ accepts } Y_t | Z_t, Z_{t-1,...}, X_{t-1}, X_{t-2}, ..., Y_{t-1}, Y_{t-2}, ...) = Pr(\text{Pred. nr.}i \text{ accepts } Y_t | Z_t, X_{t-1}).$$

In other words, it is assumed that assignment of Y_t only depends on the currently active source and the current level of specialization. Second, some assumption must be made regarding the *data allocation* probabilities mentioned in **A0**. Let us first define these probabilities more explicitly. For $n = ..., -1, 0, 1, ...$ define

$$a_n \doteq \text{Pr}\left(\text{Pred. nr.1 accepts } Y_t | Z_t = 1, X_{t-1} = n\right), b_n \doteq \text{Pr}\left(\text{Pred. nr.2 accepts } Y_t | Z_t = 2, X_{t-1} = n\right).$$

In other words,

1. a_n is the probability that pred.1 accepts a sample from source 1, given that specialization level is n;

2. b_n is the probability that pred.2 accepts a sample from source 2, given that specialization level is n.

From **A0** follows that X_t is a Markovian process on **Z**. The transition probabilities of X_t are defined by

$$p_{m,n} = \Pr(X_t = n + 1 | X_{t-1} = m)$$

and can be computed explicitly to be (for $n = ..., -1, 0, 1, ...$)

$$p_{n,n+1} = \pi_1 a_n + \pi_2 b_n, \quad p_{n,n-1} = \pi_1 \cdot (1 - a_n) + \pi_2 \cdot (1 - b_n), \quad p_{n,m} = 0 \text{ when } |m - n| \neq 1.$$

Now, regarding the probabilities a_n, b_n, the following assumptions are made.

A1   For all n, $a_n > 0$, and $\lim_{n \to +\infty} a_n = 1$, $\lim_{n \to -\infty} a_n = 0$.

A2   For all n, $b_n > 0$, and $\lim_{n \to +\infty} b_n = 1$, $\lim_{n \to -\infty} b_n = 0$.

In other words the following are assumed.

1. As the specialization level increases to plus infinity (which means that either predictor 1 has received a lot more data from source 1 than from source 2, or that predictor 2 has received a lot more data from source 2 than from source 1, or both):

   (a) predictor 1 is very likely to accept an additional sample from source 1, while it is very unlikely to accept a sample from source 2;

   (b) predictor 2 is very likely to accept an additional sample from source 2, while it is very unlikely to accept a sample from source 1.

2. Similarly, as the specialization level decreases to minus infinity (which means that either predictor 1 has received a lot more data from source 2 than from source 1, or that predictor 2 has received a lot more data from source 1 than from source 2):

   (a) predictor 1 it is very likely to accept an additional sample from source 2, while it is very unlikely to accept a sample from source 1;

   (b) predictor 2 is very likely to accept an additional sample from source 1, while it is very unlikely to accept a sample from source 2.

It must be stressed that whether the above assumptions **A1** and **A2** are satisfied depends on three factors: (a) the input/output behavior of the sources; (b) the type of predictors used; (c) the training algorithm used. *In short, assumptions* **A1**, **A2** *characterize the sources/ predictors/ training combination.*

5.1.3 Convergence Results

Now we are ready to state our results. Two convergence theorems are presented here. The corresponding proofs are quite lengthy and hence are presented in the book [20].

The first theorem regards the behavior of X_t. Here "i.o." means "infinitely often" [1].

Theorem 1 *If conditions **A0, A1, A2** hold, then*

$$\forall m \in \mathbf{Z} \quad \Pr\left(X_t = m \quad i.o.\right) = 0, \tag{2}$$

$$\Pr\left(\lim_{t \to \infty} |X_t| = +\infty\right) = 1, \tag{3}$$

$$\Pr\left(\lim_{t \to \infty} X_t = +\infty\right) + \Pr\left(\lim_{t \to \infty} X_t = -\infty\right) = 1. \tag{4}$$

The most important part is (4): if **A0, A1** and **A2** hold, then "in the long run" two possibilities exist.

$X_t \to +\infty$: Either predictor no.1 will accumulate a lot more source no.1 samples than source no.2 samples, or predictor no.2 will accumulate a lot more source no.2 samples than source no.1 samples, or both.

$X_t \to -\infty$: Either predictor no.1 will accumulate a lot more source no.2 samples than source no.1 samples, or predictor no.2 will accumulate a lot more source no.1 samples than source no.2 samples, or both.

The total probability that one of these two events will take place is one, i.e. in any case at least one predictor will accumulate a lot more data from one source than data from the other source. Notice that Theorem 1 does not say that *both* predictors will specialize; this is stated in the next theorem.

Theorem 2 *If conditions **A0, A1, A2** hold, then*

1. If $\Pr(\lim_{t \to \infty} X_t = +\infty) > 0$ *then*

$$\Pr\left(\lim_{t \to \infty} \frac{N_t^{21}}{N_t^{11}} = 0 \,\bigg|\, \lim_{t \to \infty} X_t = +\infty\right) = 1, \tag{5}$$

$$\Pr\left(\lim_{t \to \infty} \frac{N_t^{12}}{N_t^{22}} = 0 \,\bigg|\, \lim_{t \to \infty} X_t = +\infty\right) = 1. \tag{6}$$

2. If $\Pr(\lim_{t \to \infty} X_t = -\infty) > 0$ *then*

$$\Pr\left(\lim_{t \to \infty} \frac{N_t^{11}}{N_t^{21}} = 0 \,\bigg|\, \lim_{t \to \infty} X_t = -\infty\right) = 1, \tag{7}$$

$$\Pr\left(\lim_{t \to \infty} \frac{N_t^{22}}{N_t^{12}} = 0 \,\bigg|\, \lim_{t \to \infty} X_t = -\infty\right) = 1. \tag{8}$$

Theorem 2 states that, with probability one, *both* predictors will specialize, one in each source and in a "strong" sense . For instance, if $X_t \to +\infty$, then the *proportion* N_t^{21}/N_t^{11} (no. of source 2 samples divided by no. of source 1 samples *assigned to predictor* no.1) goes to zero; this means that "most" of the samples on which predictor 1 was trained come from source 1 and, also, that "most" of the time a sample of source 1 is assigned (classified) to the predictor which is specialized in this source. Hence we can identify source 1 with predictor no.1. Furthermore the proportion N_t^{12}/N_t^{22} (no. of. source 1 samples divided by no. of source 2 samples *assigned to predictor* no.2) also goes to zero ; this means that "most" of the samples on which predictor no.2 was trained come from source 2 and, also, that "most" of the time a sample of source 2 is assigned (classified) to the predictor which is specialized in this source. Hence we can identify source 2 with predictor no.2. A completely symmetric situation holds when $X_t \to -\infty$, with predictor no.1 specializing in source no.2 and predictor no.2 specializing in source no.1. Since, by Theorem 1, X_t goes either to $+\infty$ or to $-\infty$, it follows that specialization of both predictors (one in each source) is guaranteed.

5.2 Two sources, Serial Data Allocation

5.2.1 Some Important Processes

As in Section 5.1, in the case of two active sources the source process Z_t takes values in $\{1,2\}$ and time series generation takes place according to the equation $Y_t = F_{Z_t}(Y_{t-1}, Y_{t-2}, ...)$.The source activation probabilities are again denoted by π_i, $i = 1, 2$. The data allocation processes N_t^{ij} are also defined in exactly the same manner as in Section 5.1.

The specialization process X_t is now defined slightly differently. The variable X_t, denotes the difference between the number of source 1 data and source 2 data *assigned to predictor 1*, i.e.

$$X_t = N_t^{11} - N_t^{21}.$$

5.2.2 Some Important Assumptions

Let us now consider the significance of the variable X_t. Consider the following possibilities.

1. If X_t is positive and large, then predictor no. 1 *specializes* in source no. 1 ($N_t^{11} \gg N_t^{21}$).

2. If X_t is negative and large, then predictor no. 1 specializes in source no.2 ($N_t^{21} \gg N_t^{11}$).

It follows that: if the *absolute* value of X_t is large, then predictor no.1 specializes in one of the two sources and hence it will tend to accept data from this source and reject all other data. We now introduce three assumptions.

B0. For $i = 1, 2$ the following is true:

$\Pr(\text{Pred. nr.}i \text{ accepts } Y_t | Z_t, Z_{t-1,...}, X_{t-1}, X_{t-2}, ..., Y_{t-1}, Y_{t-2}, ...) = Pr(\text{Pred. nr.}i \text{ accepts } Y_t | Z_t, X_{t-1}).$

In other words, it is assumed that assignment of Y_t only depends on the currently active source and the current level of specialization. This is, of course, exactly analogous to assumption **A0.** Second, we define data allocation probabilities, which are however somewhat different than in the parallel data allocation case.

$$a_n \doteq \Pr\left(\text{Pred. nr.1 accepts } Y_t \,|\, Z_t = 1, X_{t-1} = n\right), \qquad b_n \doteq \Pr\left(\text{Pred. nr.1 accepts } Y_t \,|\, Z_t = 2, X_{t-1} = n\right).$$

While a_n is the same as in Section 5.1, b_n is the probability that predictor no.1, *(rather than predictor no.2)* accepts a sample from source 2, given that so far it has accepted n more samples from source no.1 than from source no.2.

From **B0** it follows that X_t is a Markovian process on **Z**. The transition probabilities of X_t are defined by

$$p_{m,n} = \Pr(X_t = n + 1 | X_{t-1} = m)$$

and turn out to be (for $n = ..., -1, 0, 1, ...$)

$$p_{n,n+1} = \pi_1 a_n; \quad p_{n,n-1} = \pi_2 b_n; \quad p_{n,n} = \pi_1 \cdot (1 - a_n) + \pi_2 \cdot (1 - b_n).$$

Probabilities of transition for all other m, such that $|n - m| > 1$ are equal to zero. Regarding the probabilities a_n, b_n, the following assumptions are made.

B1 For all n, $a_n > 0$, and $\lim_{n \to +\infty} a_n = 1,$ $\lim_{n \to -\infty} a_n = 0.$

B2 For all n, $b_n > 0$, and $\lim_{n \to +\infty} b_n = 0,$ $\lim_{n \to -\infty} b_n = 1.$

In other words we have made the following assumptions.

1. As the specialization level increases to plus infinity (which means that predictor no.1 has received a lot more data from source no.1 than from source no.2) predictor no.1 is very likely to accept an additional sample from source no.1, while it is very unlikely to accept a sample from source no.2,

2. Similarly, as the specialization level decreases to minus infinity (which means that predictor no.1 has received a lot more data from source no.2 than from source no.1) predictor no.1 it is very likely to accept an additional sample from source no.2, while it is very unlikely to accept a sample from source no.1.

Note that, once again, the validity of these conditions will depend on the combination of sources/ predictors/ training algorithm/ data allocation threshold. However there is one important difference between **A1,A2** and **B1, B2**. Namely, **B1, B2** depend on both sources, but only on the first predictor, since serial data allocation depends only on the performance of the first predictor.

5.2.3   Convergence Results

Two theorems will be presented. The first theorem regards the behavior of X_t.

Theorem 3 *If conditions **B0, B1, B2** hold, then*

$$\forall m \in \mathbf{Z} \quad \Pr\left(X_t = m \quad i.o.\right) = 0, \tag{9}$$

$$\Pr\left(\lim_{t \to +\infty} |X_t| = +\infty\right) = 1, \tag{10}$$

$$\Pr\left(\lim_{t \to +\infty} X_t = +\infty\right) + \Pr\left(\lim_{t \to +\infty} X_t = -\infty\right) = 1. \tag{11}$$

The most important conclusion is (11): if **B0, B1** and **B2** hold, then there are two possibilities in the long run.

$X_t \to +\infty$: Predictor no.1 will accumulate a lot more source no.1 samples than source no.2 samples.

$X_t \to -\infty$: Predictor no.1 will accumulate a lot more source no.2 samples than source no.1 samples.

The total probability that one of these two events will take place is one, i.e. predictor no. 1 will certainly specialize in one of the two sources; Theorem 3 does not say that *both* predictors will specialize. This is the result of Theorem 3.

Theorem 4 *If conditions **B0, B1, B2** hold, then*

1. *If* $\Pr(\lim_{t \to +\infty} X_t = +\infty) > 0$ *then*

$$\Pr\left(\lim_{t \to +\infty} \frac{N_t^{21}}{N_t^{11}} = 0 \,\middle|\, \lim_{t \to +\infty} X_t = +\infty\right) = 1, \tag{12}$$

$$\Pr\left(\lim_{t \to +\infty} \frac{N_t^{12}}{N_t^{22}} = 0 \,\middle|\, \lim_{t \to +\infty} X_t = +\infty\right) = 1. \tag{13}$$

2. *If* $\Pr(\lim_{t \to +\infty} X_t = -\infty) > 0$ *then*

$$\Pr\left(\lim_{t \to +\infty} \frac{N_t^{11}}{N_t^{21}} = 0 \,\middle|\, \lim_{t \to +\infty} X_t = -\infty\right) = 1, \tag{14}$$

$$\Pr\left(\lim_{t \to +\infty} \frac{N_t^{22}}{N_t^{12}} = 0 \,\middle|\, \lim_{t \to +\infty} X_t = -\infty\right) = 1. \tag{15}$$

Theorem 4 states that with probability one *both* predictors will specialize, one in each source and in the "strong" *ratio* sense, as already discussed in Section 5.1. Since, by Theorem 3, X_t goes either to $+\infty$ or to $-\infty$, it follows that specialization of both predictors (one in each source) is guaranteed.

5.3 Many Sources

Let us now consider the question of convergence when more than two sources are active. We will mostly discuss the case of parallel data allocation; very similar considerations apply to the serial case.

Our analysis in this case is more heuristic than in the previous two sections. The central idea is recursive application of the data allocation scheme. To be specific, suppose that there are K active sources, i.e. the source set is $\Theta = \{1, 2, ..., K\}$. The parallel data allocation scheme starts with two predictors. Suppose that there is a partition of Θ, say $\Theta_1 = \{k_1, k_2, ..., k_{\overline{K}}\}$ and $\Theta_2 = \{k_{\overline{K}+1}, k_{K_1+2}, ..., k_K\}$,where k_1, k_2, ..., k_K is a permutation of 1,2, ..., K and $\overline{K}$ is between 1 and K. Now, we can consider two *composite* sources; the first one has the form

$$Y_t = \sum_{i=1}^{\overline{K}} \mathbf{1}(Z_t = k_i) \cdot F_{k_i}(Y_{t-1}, Y_{t-2}, ..., Y_{t-M}) \tag{16}$$

and the second has the form

$$Y_t = \sum_{i=\overline{K}+1}^{K} \mathbf{1}(Z_t = k_i) \cdot F_{k_i}(Y_{t-1}, Y_{t-2}, ..., Y_{t-M}) \tag{17}$$

(here $\mathbf{1}(A)$ is the indicator function, equal to one when the event A is true, zero otherwise). From eqs.(16), (17) it follows that the generation of the time series can be described by

$$Y_t = \Phi_{\overline{Z}_t}(Y_{t-1}, Y_{t-2}, ..., Y_{t-M}). \tag{18}$$

In other words, we can consider Y_t to be produced by a new ensemble of two successively activated sources, where source activation is denoted by the variable $\overline{Z}_t$, taking values in $\{1, 2\}$ and each of the two new sources is actually a composite of simpler sources. Now the two-sources analysis presented in the previous section also applies to the many sources case, as long as each of the sets Θ_1, Θ_2 is considered as a composite source. In particular, if assumptions **A0, A1, A2** hold, then the parallel data allocation scheme will be convergent in the sense of Theorem 2. Hence the incoming data will be separated into two sets; one set will contain predominantly data generated by the composite source no.1 and the other set will contain predominantly data generated by the composite source no.2. To be more precise, if the variables N_t^{ij} $(i, j = 1, 2)$ have the meaning explained in the previous section, but *with respect to the composite sources no.1 and 2*, then with probability one we will have either

$$\lim_{t\to\infty} \frac{N_t^{21}}{N_t^{11}} = 0 \text{ and } \lim_{t\to\infty} \frac{N_t^{12}}{N_t^{22}} = 0, \quad \text{or} \quad \lim_{t\to\infty} \frac{N_t^{11}}{N_t^{21}} = 0 \text{ and } \lim_{t\to\infty} \frac{N_t^{22}}{N_t^{12}} = 0.$$

Consider, to be specific, the first case. In this case, the proportion of source no.2 generated data that is found in the training data of source no.1 goes to zero. Suppose now that the parallel data allocation scheme is applied once again, only to these data, *using a new combination of predictors/training*

algorithm. Suppose that there is a further partition of the source subset Θ_1 into sets Θ_{11}, Θ_{12}. If conditions **A0, A1, A2** hold true for this new combination of composite sources, predictors and training algorithm, *and given that data from sources belonging to set Θ_2 will be contained in a vanishing proportion in the training data*, it follows from Theorem 2 that the training data will be further separated into two subsets, one corresponding to each source subset Θ_{11}, Θ_{12}. Of course, exactly the same argument applies to source set Θ_2 which will be separated into subsets Θ_{21}, Θ_{22}, each with a corresponding training data subset. This procedure continues until the original source set Θ is hierarchically partitioned into a number of sets for which no further partitions satisfying conditions **A0, A1, A2** are possible. By judicious choice of the predictors and training algorithm it is possible to reduce the sets of the final partition to singletons, i.e. break down the original set Θ to K subsets of the form $\{k_1\}, \{k_2\}, \ldots, \{k_K\}$ where $(k_1, k_2, \ldots, k_K)$ is a permutation of $(1, 2, ..., K)$. In other words, exactly one predictor corresponds to each subset / source and asymptotically correct data allocation is established. Similar arguments can be used to conclude that in the case of serial data allocation asymptotically correct separation of the data will take place.

6   Experiments

In this section we present three groups of data allocation experiments which were used to evaluate the performance of the parallel and serial data allocation schemes.

6.1   Experiment Group A

In the first experiment two sources are used, i.e. z_t takes values in $\{1, 2\}$. The sources are described by the following general form $y_t = f_{z_t}(y_{t-1})$; in other words the time series is generated by functions $f_1(.)$, $f_2(.)$. Specifically, we have

$$f_1(x) = 4x \cdot (1 - x) \quad \text{(a logistic function)};$$

$$f_2(x) = \begin{cases} 2x & \text{if } x \in [0, 0.5) \\ 2 \cdot (1 - x) & \text{if } x \in [0.5, 1] \end{cases} \quad \text{(a tent-map function)}.$$

The two sources are activated consecutively, each for 200 time steps, resulting in a period of 400 time steps. The data allocation task consists in discovering that two sources are active and separating the data y_1, y_2, ... into two groups, one group corresponding to each source. 200 time steps of the composite time series are presented in Figure 1.

A number of experiments are performed using the time series described above, observed at various levels of noise, i.e. at every step y_t is mixed with additive white noise uniformly distributed in the interval $[-A/2, A/2]$. Six values of A are used: 0.00 (noise free case), 0.02, 0.04, 0.10, 0.14, 0.20. The predictors used are 1-4-1 sigmoid neural networks which are trained using a Levenberg-Marquardt

algorithm[1]; the algorithm parameters are taken to be as follows: block length $N=10$, retrain period L $=100$ and $J=5$, $N_c=500$ and $d=0.1$.

Data allocation is performed using both parallel and serial data allocation schemes. In every experiment performed, both schemes succeed in discovering the existence of two sources and proceed in allocating data to the two corresponding predictors. Two quantities are of interest: the time T_c at which the sources are discovered and the classification accuracy c after time T_c. T_c is computed as follows: a running average of prediction errors is calculated for every time t; if for every data allocation the predictor which receives the incoming sample y_t has prediction error less than one half that of the remaining predictors, and if this condition holds for 50 consecutive data allocations (i.e. for 500 observations), then it is assumed that all sources have been discovered and all predictors specialized. Then T_c is set equal to the current t. Then classification accuracy is computed for the 200 data blocks (2000 observations) corresponding to times $T_c+1, T_c+2, ..., T_c+200$; i.e. c is set equal to $\overline{T}/200$ where $\overline{T}$ is the number of correctly classified samples.

For both parallel and serial data allocation, six experiments are performed at every noise level and the resulting c and T_c values are averaged. The average c is plotted as a function of noise level A in Figure 2 and the average T_c is plotted as a function of noise level A in Figure 3.

It can be seen that both schemes perform very well at low to medium noise levels. At high noise levels the parallel data allocation scheme still shows very good performance; the serial scheme achieves a relatively low level of correct classification and, in particular, fails to satisfy the T_c computation criterion (hence the respective part of the graph is missing in Figure 3. On the other hand, at low to middle noise levels, the serial scheme achieves classification faster: the average values of T_c are lower than these of the parallel scheme.

The evolution of the data allocation process in three representative experiments is presented in Figures 4, 5 and 6. Figure 4 corresponds to serial data allocation, at noise level $A=0.00$, Figure 5 corresponds to parallel data allocation, at noise level $A=0.00$ and Figure 6 corresponds to parallel data allocation, at noise level $A=0.14$.

6.2   Experiment Group B

In this group of experiments, three sources are used, i.e. z_t takes values in $\{1, 2, 3\}$. The sources are described by the following general form

$$y_t = f_{z_t}(y_{t-1});$$

in other words the time series is generated by functions $f_1(.)$, $f_2(.)$, $f_3(.)$. The first two functions are as described in the previous section, while $f_3(.) = f_1(f_1(.))$ (i.e. a *double* logistic).

---

[1]Implemented by Magnus Norgaard, Technical University of Denmark, and distributed as part of the Neural Network System Identification Toolbox for Matlab.

The three sources are activated consecutively, each for 200 time steps, resulting in a period of 600 time steps. The data allocation task consists in discovering that three sources are active and separating the data y_1, y_2, ... into three groups, one group corresponding to each source. 200 time steps of the composite time series are presented in Figure 7.

Once again, experiments are performed using the above time series, observed at various levels of noise, i.e. at every step y_t is mixed with additive white noise uniformly distributed in the interval $[-A/2, A/2]$. Six values of A are used: 0.00 (noise free case), 0.02, 0.04, 0.10, 0.14, 0.20. Block length N, predictor types, training algorithm, and algorithm parameters are taken the same as in the previous section.

Again both the parallel and serial data allocation schemes are used. In every experiment performed, (with the exception of serial data allocation and noise level $A \geq 0.14$) both schemes succeed in discovering the existence of three sources and proceed in allocating data to the corresponding predictors. It should be noticed that in the case of parallel data allocation, there is an initial phase where data are allocated to two groups; after specialization in these two groups (composite sources) takes place, two new predictors are introduced for each group and the new incoming data are further allocated to four subgroups, two for each original group. However, it is easily established (using a predictive error comparison criterion) that in one group the two subgroups really correspond to one source, and so these subgroups are merged, resulting in three final groups and predictors. The quantities c and T_c are computed as in the previous section.

For both parallel and serial data allocation, six experiments are performed at every noise level and the resulting c and T_c values are averaged. The average c is plotted as a function of noise level A in Figure 8 and the average T_c is plotted as a function of noise level A in Figure 9.

Again it can be seen that both schemes perform very well at low to medium noise levels (where the serial scheme achieves faster classification) and the parallel data allocation scheme also has very good performance at high noise levels.

6.3   Experiment Group C

In the final experiment group, the time series used is obtained from three sources of the Mackey-Glass type. The original data evolves in continuous time and satisfies the differential equation :

$$\frac{dy}{dt} = -0.1y(t) + \frac{0.2y(t - t_d)}{1 + y(t - t_d)^{10}},$$

For each source a different value of the delay parameter t_d is used, namely t_d=10, 17 and 30. The time series is sampled in discrete time, at a sampling rate $\tau = 6$, with the three sources being activated alternately, for 100 time steps each. The final result is a time series with a switching period of 300. 200 time steps of the composite time series are presented in Figure 10.

Once again, experiments are performed using the above time series, observed at various levels of noise, i.e. at every step y_t is mixed with additive white noise uniformly distributed in the interval

$[-A/2, A/2]$. Six values of A are used: 0.00 (noise free case), 0.02, 0.04, 0.10, 0.14, 0.20. Block length N is equal to 10, the predictors are 5-5-1 sigmoid neural networks. The training algorithm is the same as in the previous sections; the algorithm parameters are taken to be as follows: $L = 100$, $J = 10$, $N_c = 100$ and $d = 0.05$.

Once again both the parallel and serial data allocation schemes are used and in every experiment both schemes succeed in discovering the existence of three sources and allocating data to the corresponding predictors. Similarly to experiment group B, in the case of parallel data allocation there is an initial phase (first level data allocation) where data are allocated to two groups and a second level phase, where two new predictors are introduced for each group. By using the predictive error comparison criterion, it is established that the two subgroups of one group really correspond to one source, and so these subgroups are merged, resulting in three final groups and predictors. The quantities c and T_c are computed as in the previous sections.

For both parallel and serial data allocation, six experiments are performed at every noise level and the resulting c and T_c values are averaged. The average c is plotted as a function of noise level A in Figure 11 and the average T_c is plotted as a function of noise level A in Figure 12. Both data allocation schemes perform very well at all noise levels. The serial scheme achieves considerably faster classification.

6.4   Discussion of the Experiments

The results presented above indicate that both serial and parallel data allocation perform quite well. While serial data allocation reaches a sufficient level of specialization faster, parallel data allocation is more robust to noise. It should be added that the computation requirements are quite modest for both schemes; for instance in experiment group B, allocating 1000 observations takes on the average 3 minutes of processing, for both the serial and parallel data allocation schemes. This processing time corresponds to implementation using MatLab 5, running on a 200 MHz Pentium II computer. Optimized C code would undoubtedly results in much shorter execution times. Hence the above schemes are suitable for online implementation (keep in mind that MATLAB is an interpreted language).

7   Conclusions

Sufficient conditions have been presented in this paper for the convergence of a class of hierarchical, unsupervised, on-line schemes for the classification and identification of switching time series. The schemes considered are characterized by the following features.

1. They use predictive modular neural networks. Identification is performed by training one predictive neural module for every active source; the module is trained on source specific data.

2. Data allocation is unsupervised, i.e. labeled data are not available. Hence observed data are allocated to predictors according to their predictive performance.

21

3. Data allocation is performed in serial or parallel manner.

The analysis concentrates on the data allocation problem, assuming that, once data are correctly labeled for every source, then training one predictor per source is not particularly hard. Hence, the main question discussed here is whether correct data allocation will be asymptotically attained. The answer, as pointed out in Theorems 1, 2, 3 and 4, is affirmative, provided that either conditions **A0, A1, A2** or **B0, B1, B2** are satisfied. These conditions are quite general and do not depend on particular properties of the sources, predictors or training algorithms used. A price paid for the generality of the analysis is that checking whether the conditions hold for a specific identification problem is not easy. However, our analysis indicates that, for a problem which involves sources with fairly distinct behavior and predictors which can model the sources rather accurately the time series, correct data allocation will take place. In this sense, our analysis gives a theoretical justification for common sense expectations. In case the sources are close to each other (in the input-output sense), or measurements are too noisy or the capacity of the predictors is large compared to the complexity of the sources, success of the data allocation scheme is not guaranteed. These conclusions are corroborated from numerical experiments.

Several issues merit further exploration. We are particularly intested in possible connections between capacity / complexity concepts and conditions **A0, A1, A2** and **B0, B1, B2**. It is also worth examining whether introduction of data sharing between the predictors may lead to fundamentally different behavior of data allocation schemes.

References

[1] P. Billingsley, *Probability and Measure*, 1985, Wiley, New York.

[2] T.W. Cacciatore and S.J.Nowlan, "Mixtures of controllers for jump linear and nonlinear plants", in *Advances in Neural Information Processing Systems 6 (NIPS 93)*, 1994, eds. J.D. Cowen, G. Tesauro and J. Alspector, pp. 719-726, San Francisco, CA, Morgan Kaufmann.

[3] M.J. Caputi. "A necessary condition for effective performance of the multiple model adaptive estimator", *IEEE Trans.on Aerospace and Electronic Systems*, Vol.31, pp.1132-1138, 1995.

[4] F. Dufour and P. Bertrand. "The filtering problem for continuous-time linear systems with Markovian switching coefficients", *System and Control Letters*, vol.23, pp.453-461, 1994.

[5] F. Dufour and P. Bertrand. "Stabilizing control for hybrid models", *IEEE Trans. on Automatic Control*, vol.39, pp.2354-2357, 1993.

[6] R.J. Elliot, L. Aggoun and J.B. Moore, *Hidden Markov Models: Estimation and Control*, 1995, Springer.

[7] R.A. Jacobs, M.I. Jordan, S.J. Nowlan and G.E. Hinton. "Adaptive mixtures of local experts", *Neural Computation*, vol.3, pp.79-87, 1991.

[8] M.I. Jordan and R.A. Jacobs. "Hierarchical mixtures of experts and the EM algorithm", *Neural Computation*, vol.6, pp. 181-214, 1994.

[9] T.A. Johansen and B. A. Foss, " Identification of non-linear system structure and parameters using regime decomposition", *Automatica*, vol. 31, pp.321-326, 1995.

[10] A. Kehagias and V. Petridis, "Predictive modular neural networks for time series classification", *Neural Networks*, vol.10, pp.31-49, 1996.

[11] A. Kehagias and V. Petridis, "Time series segmentation using predictive modular neural networks", *Neural Computation*, vol.9, pp.1691-1710, 1997.

[12] P. Kenny, M. Lennig and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol.8, pp.220-225, 1990.

[13] X.R. Li and Yaakov Bar-Shalom, "Multiple-model estimation with variable structure", *IEEE Trans. on Automatic Control*, vol.41, pp.478-493, 1996.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations". In *Proc. of the Berkeley Synposium on Math. Sciences and Probability*, 1965.

[15] K. Narendra. "Adaptation and learning using multiple models, switching and tuning", *IEEE Control Systems*, pp.37-51, 1995.

[16] K. Narendra and J. Balakhrishnan. "Improving transient response of adaptive control systems using multiple models and switching", *IEEE Trans. on Automatic Control*, Vol.39, pp.1861-1866, 1994.

[17] K. Pawelzik, J. Kohlmorgen and K.R. Muller. "Annealed competition of experts for a segmentation and classification of switching dynamics", *Neural Computation*, vol.8, pp.357-372, 1996.

[18] V. Petridis and A. Kehagias, "A recurrent network implementation of time series classification", *Neural Computation*, vol.8, pp.357-372, 1996.

[19] V. Petridis and A. Kehagias. "Modular neural networks for MAP classification of time series and the partition algorithm", *IEEE Trans. on Neural Networks*, vol.7, pp.73-86, 1996.

[20] V. Petridis and A. Kehagias. *Predictive Modular Neural Networks: Applications to Time Series*, 1998, Kluwer Academic Publishers.

[21] H.L. Royden, *Real Analysis*, 1968, Macmillan, New York.
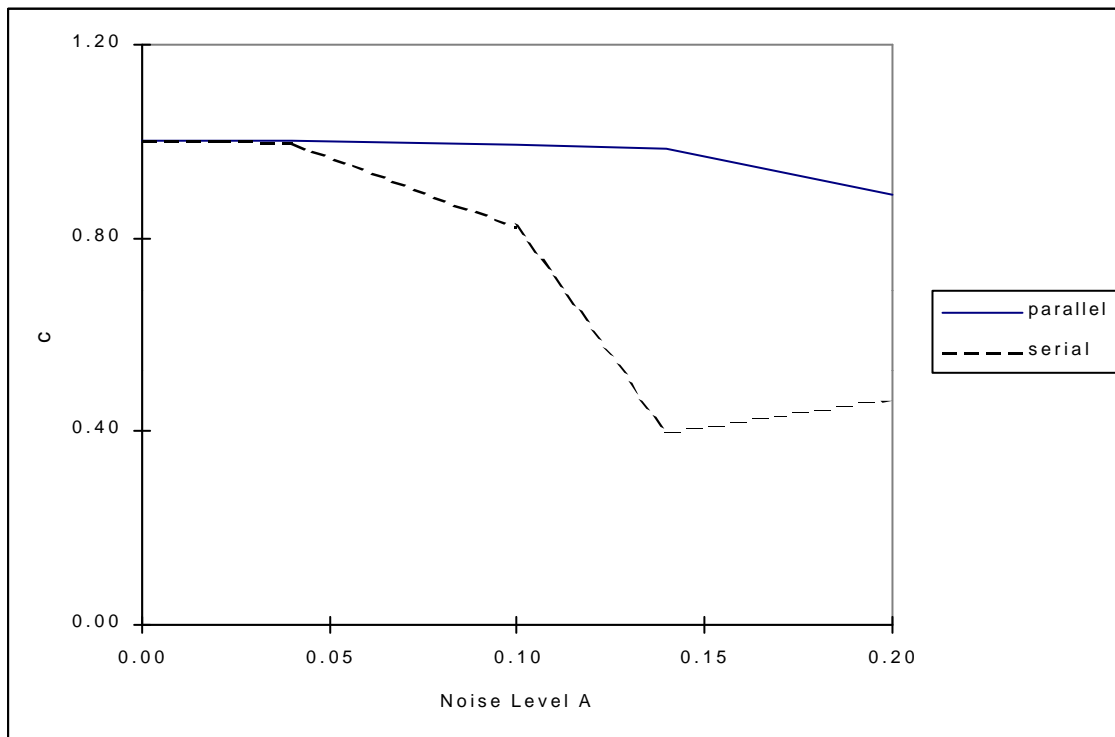
Figure Captions for the paper **"Predictive Modular Neural Networks for Unsupervised Segmentation of Switching Time Series: the Data Allocation Problem"**, by Ath. Kehagias and V. Petridis.
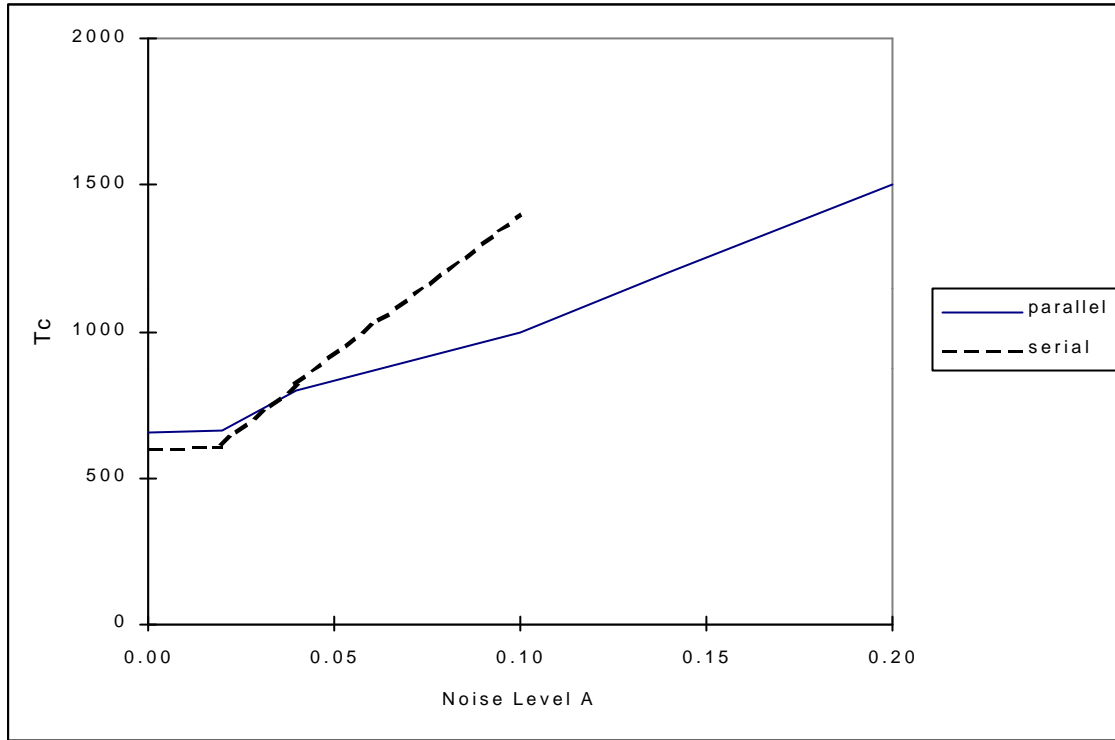
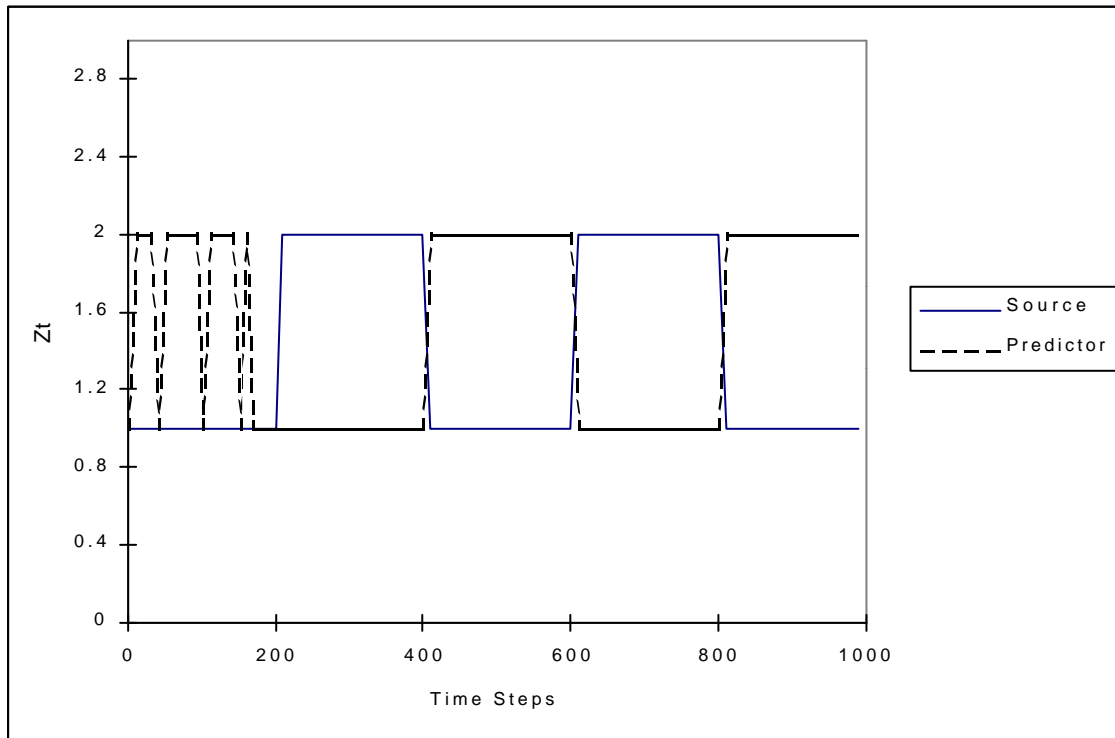**Figure 1.** Composite logistic and tentmap time series.
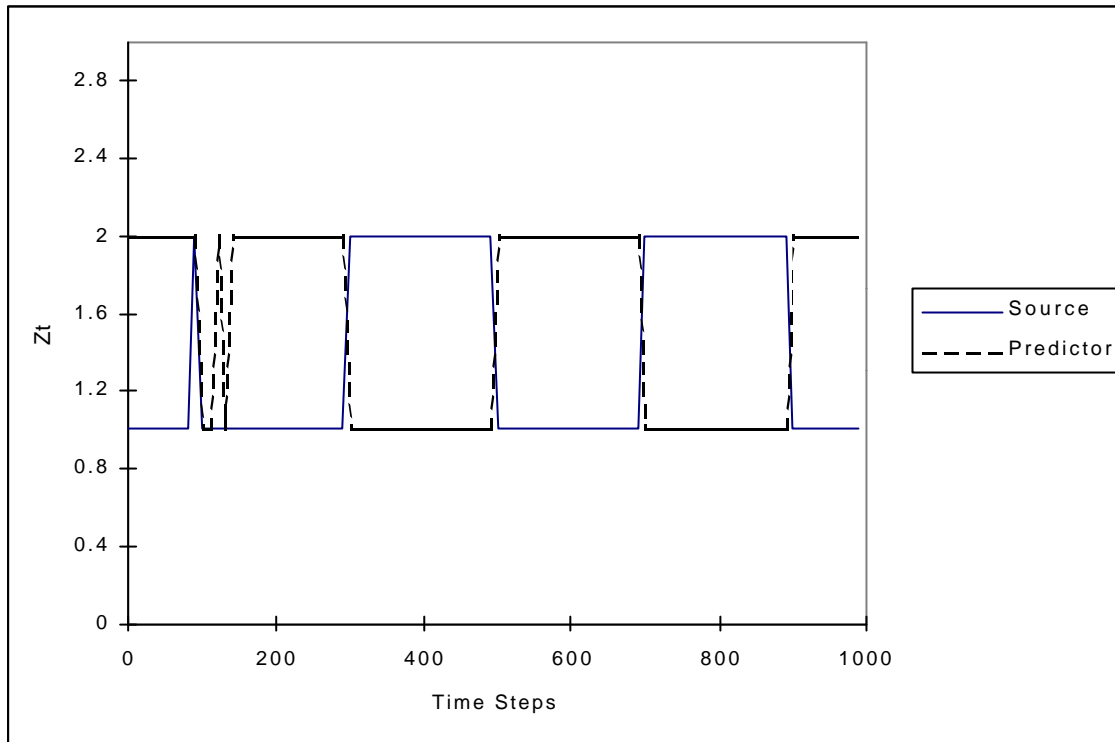**Figure 2.** Classification accuracy c.
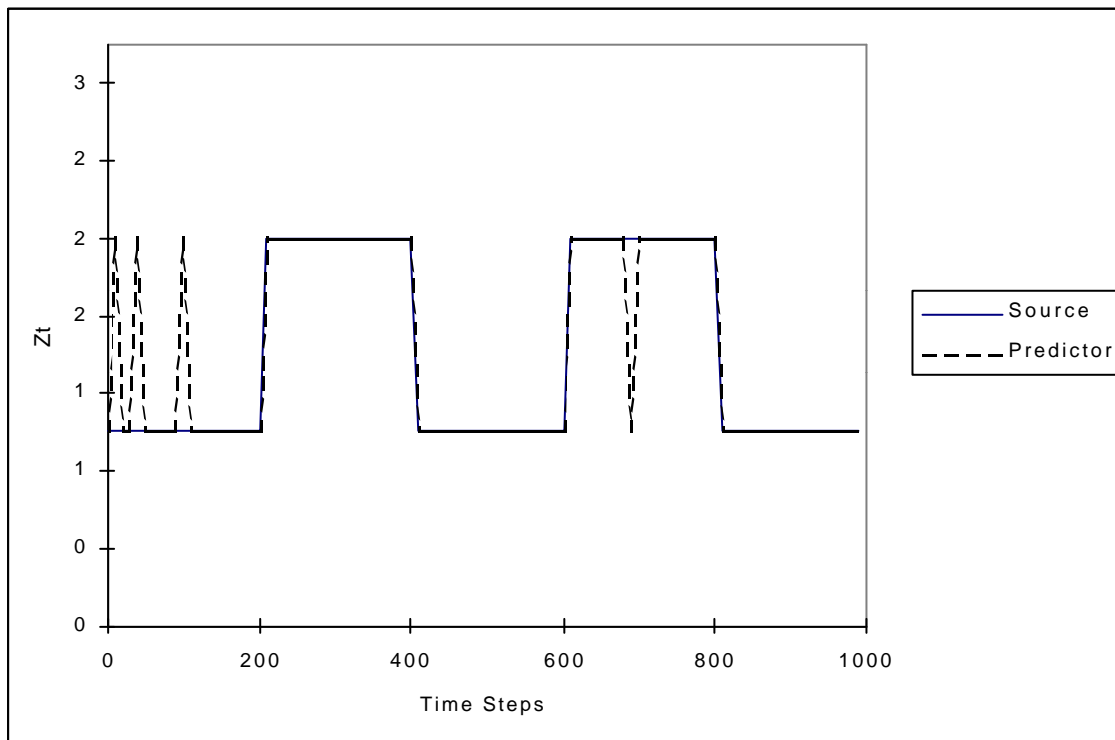**Figure 3.** Classification time T_c.
Figure 4. Evolution of data allocation; serial algorithm, A=0.00.
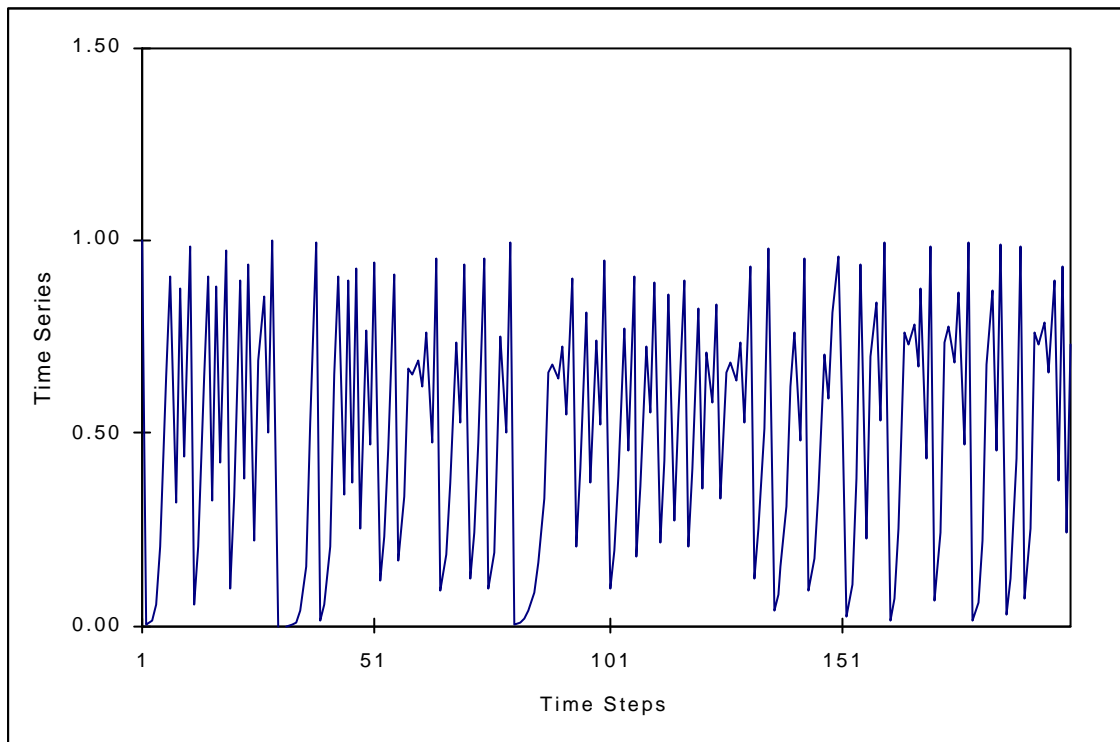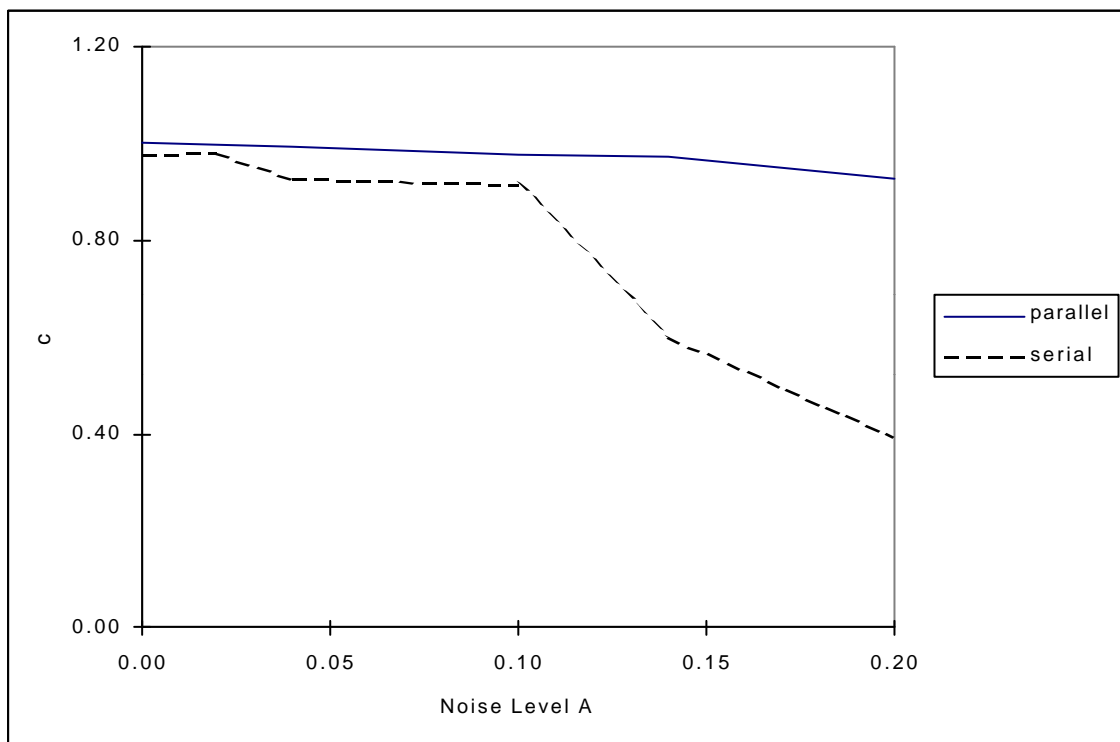**Figure 5.** Evolution of data allocation; parallel algorithm, A=0.00.
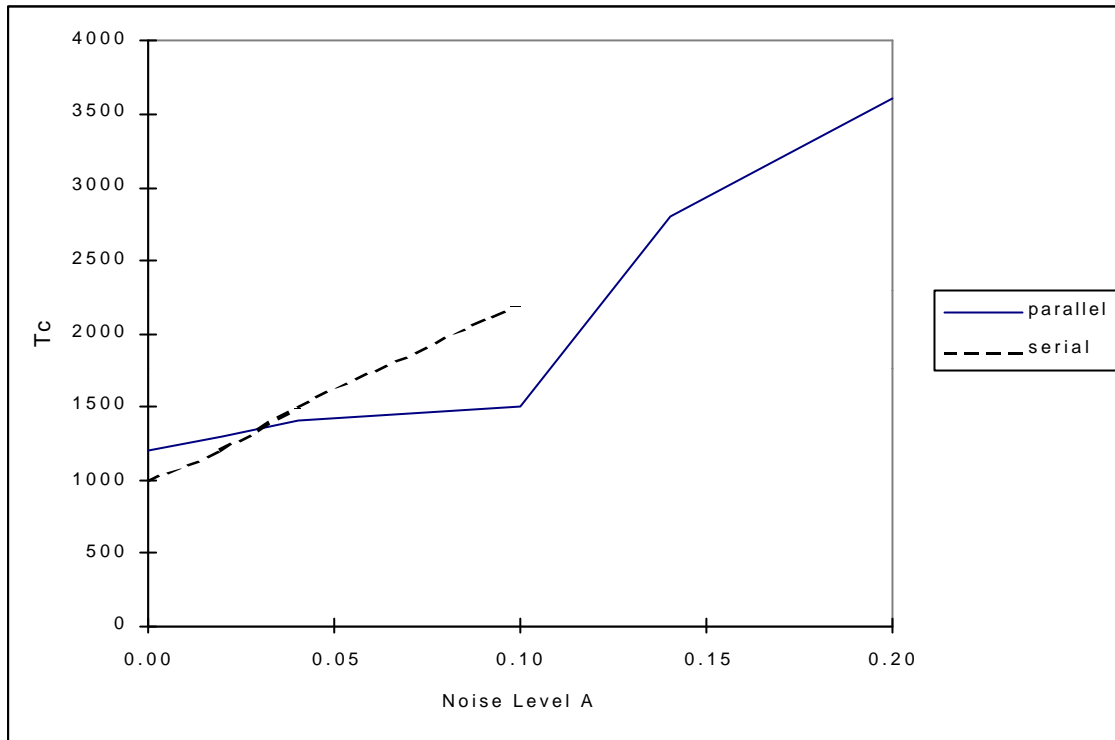**Figure 6.** Evolution of data allocation; parallel algorithm, A=0.14.
Figure 7. Logistic, tent-map, double logistic time series.
Figure 8. Classification accuracy c.
Figure 9. Classification time T_c.
Figure 10. Mackey-Glass time series.
Figure 11. Classification accuracy c.
Figure 12. Classification time T_c.

Figures for the paper **"Predictive Modular Neural Networks for Unsupervised Segmentation of Switching Time Series: the Data Allocation Problem"**, by Ath. Kehagias and V. Petridis.

Figure 1. Composite logistic and tentmap time series.

Figure 2. Classification accuracy c.

Figure 3. Classification time T_c.

Figure 4. Evolution of data allocation; serial algorithm, A=0.00.

Figure 5. Evolution of data allocation; parallel algorithm, *A*=0.00.

Figure 6. Evolution of data allocation; parallel algorithm, *A*=0.14.

28

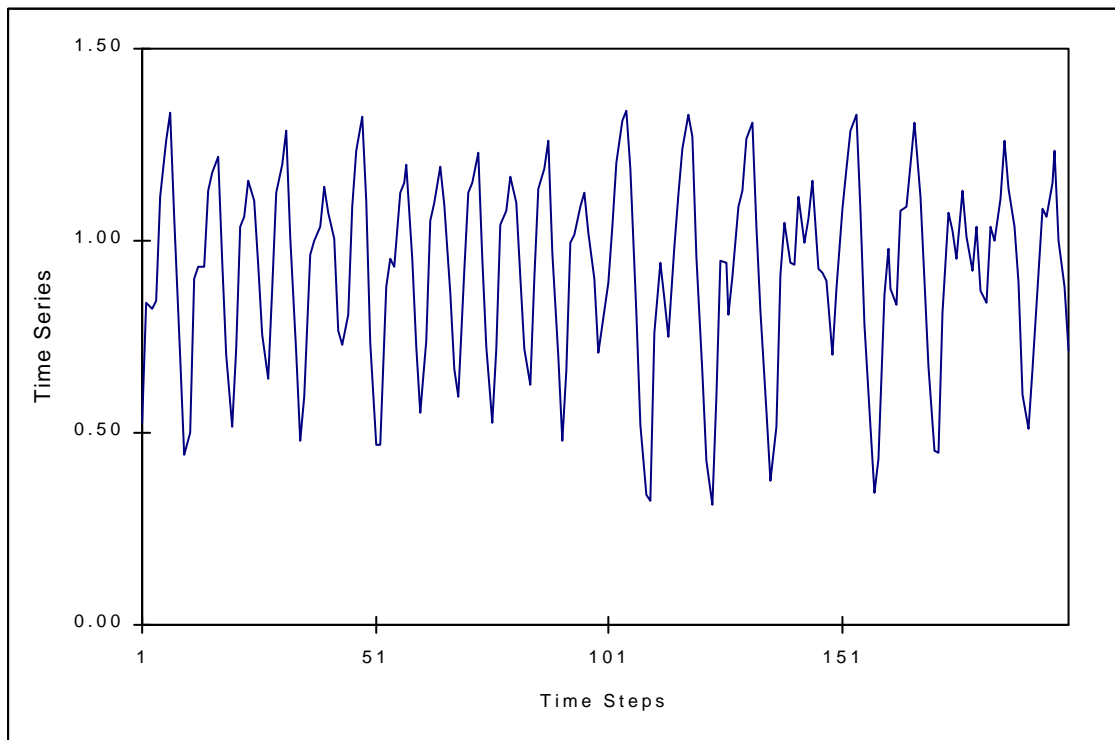**Figure 7.** Logistic, tent-map, double logistic time series.
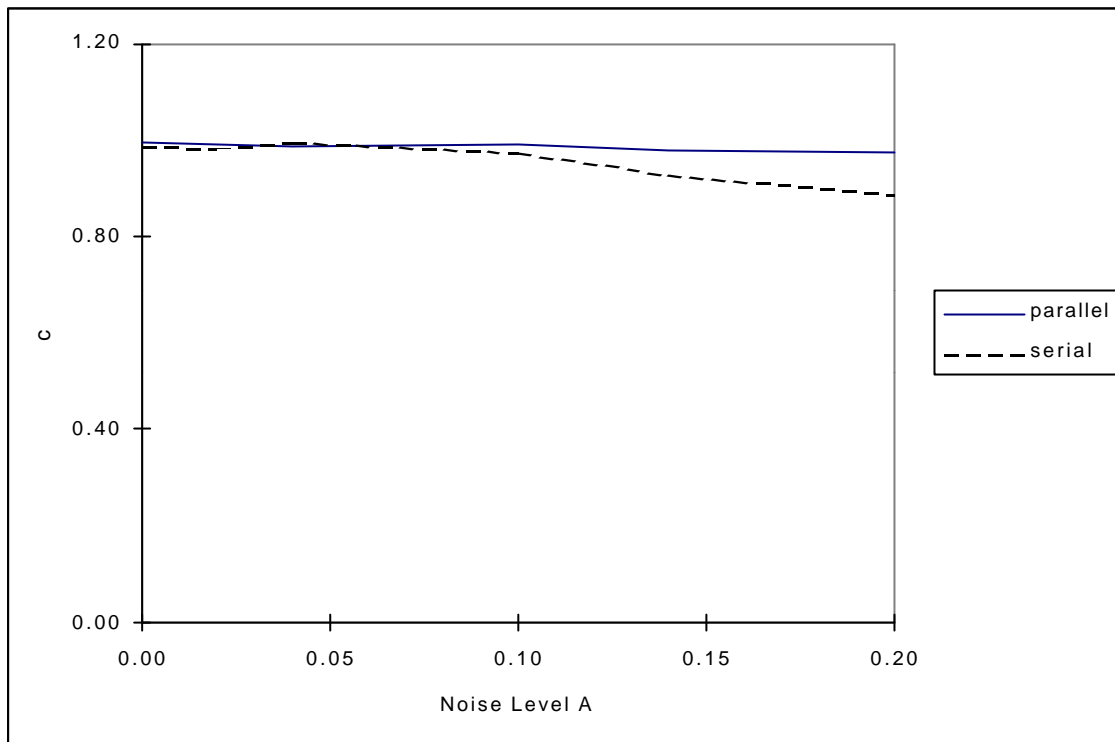

Figure 8. Classification accuracy *c*.

Figure 9. Classification time T_c.

Figure 10. Mackey-Glass time series.

Figure 11. Classification accuracy c.

Figure 12. Classification time T_c.

31