



Ca' Foscari
University
of Venice

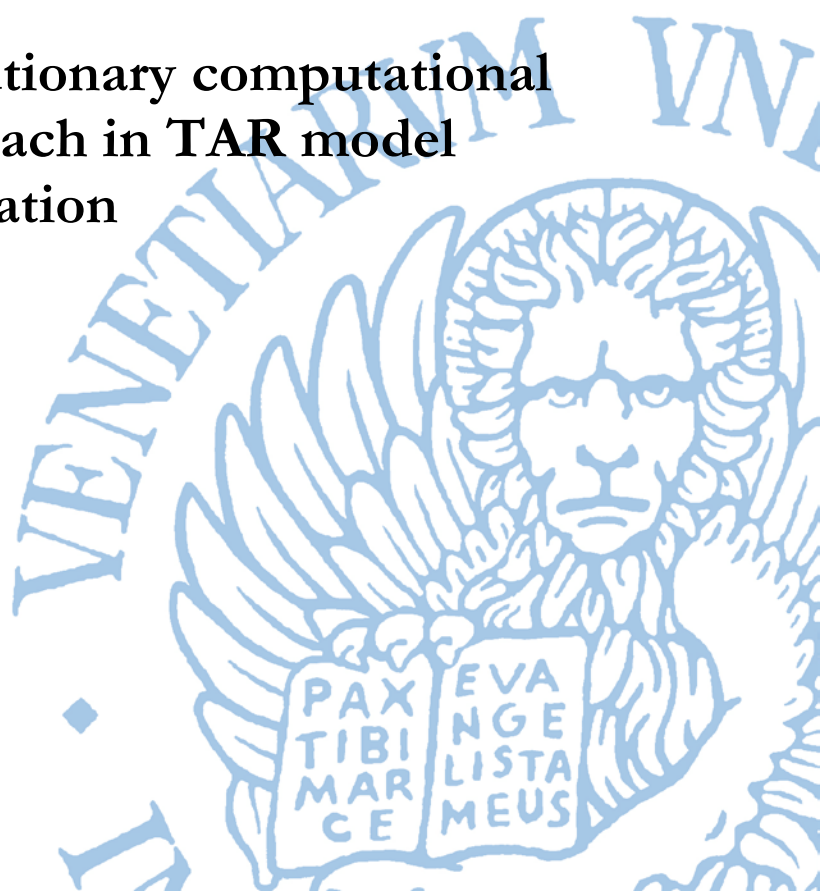
**Department
of Economics**

Working Paper

**Francesca Parpinel and
Claudio Pizzi**

**Evolutionary computational
approach in TAR model
estimation**

ISSN: 1827/3580
No. 26/WP/2011





Evolutionary computational approach in TAR model estimation

Francesca Parpinel **Claudio Pizzi**

*Department of Economics
Ca' Foscari University of Venice*

November 2011

Abstract

The well-known SETAR model introduced by Tong belongs to the wide class of TAR models that may be specified in several different ways. Here we propose to consider the delay parameter as endogenous, that is we make it to depend on both the past value and the specific past regime of the series. In particular, we consider a system switching between two regimes, each of them is a linear autoregressive of order p , with respect to the value assumed by a delayed self-variable compared with an asymmetric threshold; the peculiarity is that the switching rule also depends on the regime in which the system lies at time $t-d$.

In this work we consider two identification procedures: the first one follows the classical estimation for SETAR models, the second one proposes to estimate this model using the Particle Swarm Optimization technique.

Keywords

Parameter Estimation, Threshold Autoregressive Models, Particle Swarm Optimization.

JEL Codes

C13, C32, C51, C63

Address for correspondence:

Claudio Pizzi
Department of Economics
Ca' Foscari University of Venice
Cannaregio 873, Fondamenta S.Giobbe
30121 Venezia - Italy
Phone: (+39) 041 2347420
Fax: (+39) 041 2347444
e-mail: pizzic@unive.it

This Working Paper is published under the auspices of the Department of Economics of the Ca' Foscari University of Venice. Opinions expressed herein are those of the authors and not those of the Department. The Working Paper series is designed to divulge preliminary or incomplete work, circulated to favour discussion and comments. Citation of this paper should consider its provisional character.

1 Introduction

It is well-known that the non nonlinear behaviour of several different data generating process (DGP) may be described by specific models featured by the linearity structure.

In particular, we can model nonlinearity behaviours such as jump resonance, amplitude–frequency dependency, limit cycle, subharmonics and higher harmonics with the class of models proposed by Tong and Lim [17] in the eighties. These last models improve the fitting by means of a piecewise linear model.

After the seminal paper by Tong and Lim, this model captured the attention of many researchers. On one hand many articles tackle the problem to test the threshold hypothesis, as Petrucci and Davies [11], Tsay [18] and Chang and Tong [2]. On the other hand, several authors face the parameters estimation problem applying the procedures to real data. For example, Potter [13] models the nonlinear structure of evolution of US GNP, using the SETAR model to capture the asymmetric responses of output to shocks at different stages of the business cycle. More recently, Battaglia and Protopapas [1], Wu and Chang [19] use genetic algorithms, and Gonzalo and Wolf [5] propose the subsampling techniques.

In this work¹ we want to stretch the asymmetry characterization using more than one threshold even with two regimes. The goal of the proposed model is to better capture the asymmetric nature of some phenomena. In finance for example, the stock market might be *bull* or *bear* and, as it is well known, the behaviour in these two *status* may be different. In particular the threshold at which the bull market begins to operate is different from that at which the bear market begins to operate. This feature wants to expand the idea of the work by Li and Lam [7] in which they showed the possible asymmetric behaviour of stock prices during bear and bull markets may be efficiently modelled by a threshold type non-linear time series model with conditional heteroscedastic variance. They showed, for example, that for *Hong Kong data* the return series could have a conditional mean structure which depends on the rise and fall of the market on a previous day.

Section 2 defines the class of models that we use in this paper, that we label Asymmetric Threshold Autoregressive model, in short AsyTAR, recalling

¹Presented at Conference *ECTS2011-Evolutionary Computation and Time Series*, Villa Mondragone, Rome, June 13-14 2011 <http://ser.sta.uniroma1.it/ects2011>

both the asymmetric behaviour in the two regimes and the different thresholds activating the regimes (introduced by Pizzi [12]). The AsyTAR model allows us to take into account the possible overlapping of the regimes, in its simplest structure.

The aim of this work is to evaluate the ability of an evolutive algorithm to improve some classical estimation procedures that we can use even in this context, namely we wish to estimate simultaneously all the parameters of the model. In Section 2.3 we briefly describe the Particle Swarm Optimization procedure and propose an estimation procedure. Section 3 presents some results obtained on simulated data.

2 The Asymmetric SETAR model

In their paper, Tong and Lim [17] consider among others a very interesting non linear model, the Self-Exciting Threshold AutoRegressive Model (SETAR), that we briefly recall here. Let y_t , with $t = 1, \dots, T$, be a scalar time series, and let $(r_0 = -\infty, r_1, r_2, \dots, r_{l-1}, r_l = \infty)$ a set of ordered real values such that the space \mathbb{R} is partitioned in l intervals $(-\infty, r_1), [r_1, r_2), \dots, [r_{l-1}, \infty)$. Let be $d \in \{1, \dots, T - 1\}$ the delay at which the process reacts to a changing in the system status. So the system is in regime i if $x_{t-d} \in [r_{i-1}, r_i)$.

The simplest SETAR model has only two regimes, and thus a unique threshold r . In each regime an autoregressive model of order p is defined. The switching rule between the two regimes is based on a threshold, that is:

$$y_t = \begin{cases} a_0^{(1)} + \sum_{i=1}^p a_i^{(1)} y_{t-i}^{(1)} + \varepsilon_t^{(1)} & \text{if } y_{t-d} \leq r \\ a_0^{(2)} + \sum_{i=1}^p a_i^{(2)} y_{t-i}^{(2)} + \varepsilon_t^{(2)} & \text{if } y_{t-d} > r \end{cases} \quad (1)$$

One of the main features of the the SETAR model introduced by Tong is that the thresholds define a partition of space \mathbb{R} , in the sense that we cross from the first regime to the second one when the delayed value y_{t-d} increases until the threshold is exceeded whereas the passage from the second to the first regime occurs when the delayed value y_{t-d} decreases until it becomes less than the threshold. In some sense, there is a symmetric behaviour with respect to the threshold. This property is very useful to study some phenomena, but sometimes might be unsuitable to study other ones.

Thus we want to use a new threshold model that allows us to take into account the different behaviours of the process as regard to parameters of the linear models and the threshold, for example the case shown in Section 3.

Let y_t , with $t = 1, \dots, T$, be an observed times series and let $r_1 > r_2$ be two thresholds such that $(-\infty, r_1), (r_2, \infty)$ are two overlapping intervals.

Recalling the structure of model (1) we can define an Asymmetric Self-Exciting Threshold Autoregressive model, in short AsyTAR(p, d), with two regimes each of order p with delay d , as follows

$$y_t = \begin{cases} a_0^{(1)} + \sum_{i=1}^p a_i^{(1)} y_{t-i} + \varepsilon_t^{(1)} & \text{if } y_{t-d} \leq r_1 \text{ and } J_{t-d} = 1 \\ a_0^{(2)} + \sum_{i=1}^p a_i^{(2)} y_{t-i} + \varepsilon_t^{(2)} & \text{if } y_{t-d} > r_1 \text{ and } J_{t-d} = 1 \\ a_0^{(2)} + \sum_{i=1}^p a_i^{(2)} y_{t-i} + \varepsilon_t^{(2)} & \text{if } y_{t-d} > r_2 \text{ and } J_{t-d} = 2 \\ a_0^{(1)} + \sum_{i=1}^p a_i^{(1)} y_{t-i} + \varepsilon_t^{(1)} & \text{if } y_{t-d} \leq r_2 \text{ and } J_{t-d} = 2 \end{cases} \quad (2)$$

where J_{t-d} is an unobservable variable denoting the theoretical regime in which operate the system at time $t - d$, d is the delay with which the system reacts and changes regime, $r_1 > r_2$ are two thresholds depending on the regimes. This fact means that when the system work in regime 1 the activating threshold is different with respect to that working in regime 2. Figure 1 represents the behaviour of a SETAR against an AsyTAR.

2.1 Classical identification and estimation

Another difference in the behaviour of the SETAR and AsyTAR models can be revealed applying, in both cases, the classical estimation procedures for the SETAR. In fact, in order to get the estimates, the classical techniques suggest to use the Standardized Forecast Error (in short SFE).

Let us recall that the identification of the suitable AsyTAR model presents some critical aspects very similar to the analogous SETAR model. Here we consider only some problems that we must face in different ways. In particular the first problem that we tackle is the identification of the thresholds values. The procedure, described in what follows, is the same proposed by Petrucci and Davies [11] and employed also by Tong and Yeung [16] and Hansen [4].

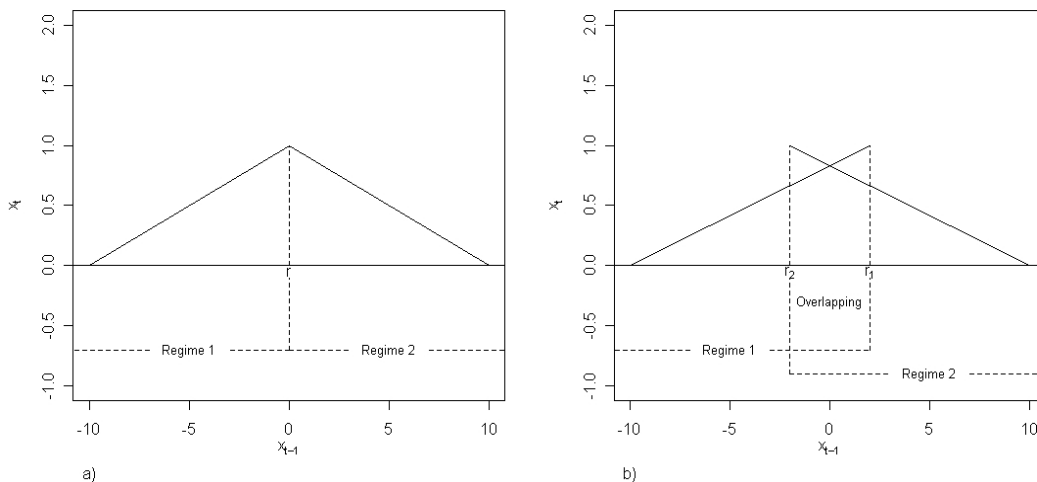


Figure 1: Theoretical Behaviour a) SETAR(2;1,1) with $d = 2$, $r = 0$, $a_1^{(1)} = -0.8$, $a_1^{(2)} = 0.8$, b) AsyTAR with $r_1 = 1$, $r_2 = -1$, $d = 2$, $a_1^{(1)} = -0.8$ and $a_1^{(2)} = .8$;

1. Embed the time serie in an appropriate m -dimensional space using the method of delay (in short MOD).
2. Sort the vectors according to the increasing values of y_{t-d} .
3. Initialize $n = n_1$, where n_1 is the minimum number of observations to regress y_t on delayed observations.
4. Estimate the linear model $y_{(t)} = \alpha + \sum_{i=1}^m \beta_i y_{(t-i)}$ using the first n ordered vectors.
5. Compute the one-step ahead standardized forecast error (SFE).
6. Update $n = n + 1$ and repeat the step 4 and 5 until $n = T$.

The SFE values, calculated in the previous procedure, allow us to detect the presence of a threshold. The plot of the cumulative sum of SFE values against n is informative of the underlying threshold. If the most appropriate model is a SETAR then the value of the cumulative sum of SFE decreases until the value of the variable, with respect to which we have ordered the vectors, is

less than the value of the threshold. When the threshold is exceeded then the cumulative sum of SFE begins to increase. However, if the DGP is an AsyTAR, then the SFE cumulative sum in the lefthand side of the graph decreases until it reaches the first threshold then it remains stable (although with a small growth) until it reaches the second threshold after which it increases again. To show these paths we have simulated two time series.

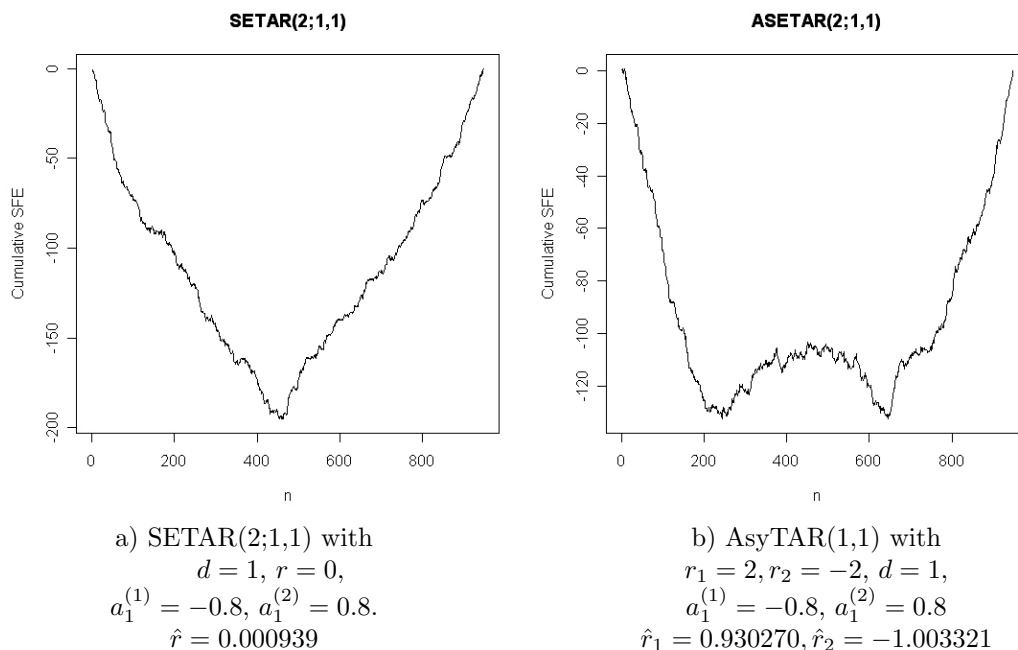


Figure 2: Cumulative sum of standardized forecast errors.

The Figure 2 depicts the results of this simulation: the “V” shape of the curve on the left hand side is obtained by applying the above procedure to a time series generated by a SETAR(2;1,1) with threshold at $r = 0$, delay $d = 2$ and two autoregressive components with parameters $a_1^{(1)} = -.8$ and $a_1^{(2)} = .8$, whereas the “bathtub” shape of the curve on the right han side comes from an AsyTAR(1,1) with thresholds $r_1 = 1, r_2 = -1$ and the same autoregressive components of the SETAR.

Note that the procedure makes us to roughly estimate the value of the threshold. For instance, in the simulated time series of Figure 2, we found that respectively \hat{r} is close to zero for the SETAR model and $\hat{r}_1 = 0.93, \hat{r}_2 = -1.00$

for the AsyTAR model.

After the identification of the thresholds we must estimate the parameters of the AsyTAR but we have to face some problems for the data in the overlapping region. In fact to estimate the parameter of the autoregressive components in the SETAR model the idea is to split the data set in two subsets of observations, one relative to the first regime and the other one concerning the second regime. The estimation of the threshold allows us to split the time series and therefore to estimate the parameters of the two autoregressive components. However, once estimated the thresholds in the AsyTAR there are three subsets of observations: the first one is assigned to the regime 1 ($y_{t-d} < r_2$), the second one is assigned to regime 2 ($y_{t-d} > r_1$), the third one is the overlapping region and it may be assigned to one of the two regimes.

On the one hand therefore we have some observations, which become potentially numerous especially when the threshold are very different, yet we do not know how to use them in the estimation procedure. On the other hand, the reduction of the number of observations used to estimate the parameters has repercussion on the efficiency of the estimates.

If we wish to overcome such a problem following a sort of *classical estimation*, we need to perform the following simple procedure, proposed by Pizzi [12].

After embedding the time series in an appropriate m -dimensional space using MOD, we split the vector time series into three part: the first subset, called S_1 , takes into account all the vectors where $y_{t-d} < r_2$ holds, the second subset, denoted by S_2 , considers all the vectors where $y_{t-d} > r_1$ is valid and the third subset, called S_3 , contains the vectors not included in the previous subsets. At this point, we are able to estimate two different autoregressive models of order p using respectively the observations included in the first and in second subset as defined in the previous step. Given the autoregressive components just estimated, we may derive variable J that defines the regimes active at each t . We have now to re-split the vector according to the variable J in two subsets, eventually update the estimate of the two autoregressive components of the AsyTAR model.

If we apply this technique to the SETAR case we will notice that the cumulative sum of SFE values versus n decreases until the value of the delayed variable is less than the threshold, while increases in the other cases. The behaviour in the AsyTAR case is quite different: there is a subset of ordered n in which the cumulative SFE behaves in a mixed way, as we can see in the

graphs of Figure 2.

We emphasize again that the asymmetry does not refer to the statistical distribution of the variables but to the switching rule. Let's note that for some values of y_{t-d} the system may assume two different configuration. In other words in the interval (r_2, r_1) the process might assume two different structures depending on the system status.

2.2 Application

With the aim of checking the usefulness of the proposed procedure, Pizzi [12] generated one time series by simulation using the model (2) presented in Section 2 with $d = 2$, $r_1 = 1$, $r_2 = -1$, $a_1^{(1)} = -0.8$, and $a_1^{(2)} = 0.8$. Using this data different models can be estimated, such as a linear autoregressive model, a SETAR(2;1,1) and an AsyTAR(1,1). Table 1 summarizes these results.

Model	Estimated Parameters				MSE
AR(1)				$\hat{\phi}_1 = 0.472$	2.125
SETAR(2;1,1)	$\hat{r} = -0.1343$		$\hat{a}_1^1 = -0.022$	$\hat{a}_1^2 = 0.648$	1.698
AsyTAR(1,1)	$\hat{r}_1 = 0.9346$	$\hat{r}_2 = -1.0597$	$\hat{a}_1^1 = -0.661$	$\hat{a}_1^2 = 0.660$	1.539

Table 1: Simulation results.

The proposed procedure, applied to simulated data, shows the ability to correctly detect the thresholds. Moreover the estimated parameters seem quite good although the procedure requires to be improved. Eventually the AsyTAR model, compared with a SETAR model and an AR one, shows a lower value of MSE defined by

$$MSE = \frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2. \quad (3)$$

2.3 Simultaneous parameter estimation

In order to identify a model on the basis of a time series observation, typically in the case of SETAR frameworks, the problem is that all parameters in the model are not simultaneously estimated. Some recent works try to overcome this problem. For instance Battaglia and Protopapas [1] propose the use of a genetic algorithm to estimate a SETAR model, whereas Maringer and Meyer

[8] consider the Simulated Annealing technique in the estimation of a STAR model.

With the same aim we suggest to use another evolutive approach, i.e. the Particle Swarm Optimization (PSO). The PSO, born in the eighties and theorized by Kennedy and Eberhart [6] and Shi and Eberhart [14], replicates the behaviour of natural flocks and swarm of animals having one specific aim or objective, preserving a swarm of particles moving around in the search space and influenced by the improvements discovered by the other particles.

This general optimization procedure is briefly described in Table 2 and may be applied in very different problems.

The procedure is based on the creation of a *population* of agents, also called *particles*, which are uniformly distributed over some space \mathcal{X} . Then it is performed an updating process on the positions of the particles until some stopping criterion is satisfied. Typically each particle's position is evaluated according to an objective function and only if a particles current position is better than its previous best position, it must be updated. For the purpose of our paper, we have now to determine the best particle according to the particle's previous best positions and to update particles' velocities according to

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \varphi_1 \mathbf{U}_1^t(\mathbf{pb}_i^t - \mathbf{x}_i^t) + \varphi_2 \mathbf{U}_2^t(\mathbf{gb}^t - \mathbf{x}_i^t). \quad (4)$$

Eventually the particles are moved to their new positions according to the following rule

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \quad (5)$$

although some variants are proposed in the literature modifying in some way the velocity-update rule.

In particular, Shi and Eberhart [14] defined the *Inertia weight* variant adding in the velocity rule (4) a parameter \mathbf{w} called inertia weight as indicated below

$$\mathbf{v}_i^{t+1} = \mathbf{w}\mathbf{v}_i^t + \varphi_1 \mathbf{U}_1^t(\mathbf{pb}_i^t - \mathbf{x}_i^t) + \varphi_2 \mathbf{U}_2^t(\mathbf{lb}_i^t - \mathbf{x}_i^t) \quad (6)$$

In the *Canonical PSO* proposed by Clerc and Kennedy [3], the inertia weight variant presents the χ constriction factor and

$$\mathbf{v}_i^{t+1} = \chi[\mathbf{v}_i^t + \varphi_1 \mathbf{U}_1^t(\mathbf{pb}_i^t - \mathbf{x}_i^t) + \varphi_2 \mathbf{U}_2^t(\mathbf{lb}^t - \mathbf{x}_i^t)],$$

<p>Creation of a <i>population of particles</i>. Repeating the following 5 steps to gain a stopping criterion.</p> <ol style="list-style-type: none"> 1. Evaluation of each particle's position according to the objective function. 2. Updating a particle's current position. 3. Determine the best particle . 4. Update particles' velocities 5. Move particles to their new positions.
--

Table 2: Classical PSO algorithm

whereas in the *fully Informed PSO* (Mendes [9]) a particle is attracted by every other particle in its neighborhood, as follows

$$\mathbf{v}_i^{t+1} = \chi \left[\mathbf{v}_i^t + \sum_{k \in \mathcal{N}_i} \varphi_k \mathbf{U}_k^t (\mathbf{pb}_k^t - \mathbf{x}_i^t) \right]$$

A lot of other variants exists in the literature, but in this work we decide to employ the *Inertia weight* variant illustrated as follows. We create a *population* of P particles $(\mathbf{x}_i^1, i = 1, \dots, P)$ uniformly distributed over the parametric space and we choose to evaluate each particle's position according to the objective (fitness) function, that in our case is the MSE defined in (3). If a particle's current position is better than its previous best position (\mathbf{pb}_i) , it is updated. Then we determine the best particle (\mathbf{lb}_i) , according to the particle's previous best position and we update particles' velocities according to the following relationship

$$\mathbf{v}_i^{\tau+1} = \omega \mathbf{v}_i^\tau + \varphi_1 \mathbf{U}_1^\tau (\mathbf{pb}_i^\tau - \mathbf{x}_i^\tau) + \varphi_2 \mathbf{U}_2^\tau (\mathbf{lb}_i^\tau - \mathbf{x}_i^\tau)$$

where \mathbf{v}_i^τ is the velocity of particle i at iteration τ , ω is an inertia weight, \mathbf{U}_j^τ are values from uniform random variables, φ_j are weights on the attraction towards the particles own best known position, \mathbf{pb}_i^τ , and the swarms best

known position, \mathbf{lb}^τ . The particles are moved to their new positions according to

$$\mathbf{x}_i^{\tau+1} = \mathbf{x}_i^\tau + \mathbf{v}_i^{\tau+1}.$$

In our case the stopping rule depends on the number of iterations or on the failure to update the particles' positions.

This technique has many advantages if compared to other procedures, first for its simpleness; furthermore it does not require the gradient of the objective functions. Moreover, it starts from a random position of the particles of the swarm and moves the particles in the space of m dimension according to some speed rule with the aim of gaining an optimum. On the other hand, the performance of the procedure is strongly influenced by the tuning of its behavioural parameters (ω , φ_1 , φ_2), and to overcome such a problem Pedersen [10] studies and proposes some optimal combinations for them.

3 Simulation results

In our applications we consider three data generating processes, one in the class of the linear model, one in the SETAR class and the last one in the AsyTAR class and to obtain some Monte Carlo comparison, we simulate $B = 100$ series each of $T = 500$ observations. To perform the algorithm we use 40 particles over 50 replications, obtaining 2000 fitness evaluations. At the first step we generate 40 particles, with dimension equal to the number of model parameters (n_{par} , that in our case belongs to $\{1, 2, 5\}$), by the means of a random vector with independent components each defined as a uniform random variable over $[-1, 1]$. The dimension of the parameters vector depends on the choice of the model that we fit to the data. When we use particles of dimension equal to 1 we are fitting an AR(1) model, whereas when the particle dimension are 2 or 5 we are fitting respectively an AR(2) and an AsyTAR(2;1,1) model.

The optimization function is the mean square error of the simulated series against the fitted values obtained by the model defined by each particle:

$$MSE = \frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2 \quad (7)$$

The parameters procedure are tuned to $\omega = 0.7$, $\varphi_1 = 1.4$, $\varphi_2 = 1.4$, following the directions of Pedersen [10] and some our simulations evidence.

The stopping rule we adopted ends the algorithm when there are no improvements in fitness function or until 100 iteration are reached. In order to arrange an estimate for one series we consider the mean vector of the particles getting the best fitness.

Tables 3, 4 and 5 present some results obtained through the PSO procedure using fitness function (7) where $y_t, t = 1, \dots, T$ is the time series and $\hat{y}_t, t = 1, \dots, T$ is the estimated series according to three different models: simple AsyTAR with 5 parameters and linear autoregressive of order 1 and 2.

In all the Tables the statistics in the last row, defined as *VarWithin*, represent the variability of the particles of the PSO, and in all cases is very low as we expect.

Table 3: PSO estimation over 100 trajectory simulated by AsyTAR: some statistics.

$$\text{model (2) with } a_0^{(1)} = a_0^{(2)} = 0, a_1^{(1)} = 0.8, a_1^{(2)} = -0.8, d = 2, r_1 = 1, r_2 = -1$$

Statistics	PSO with	PSO with		PSO with				
	linear 1 par.	linear 2 par.		AsyTAR 5 parameter				
	a_1	a_1	a_2	$a_1^{(1)}$	$a_1^{(2)}$	r_1	r_2	d
Min.	-0.162	-0.192	-0.180	0.648	-0.843	-1.00	0.33	1.92
1st Qu.	-0.040	-0.041	-0.034	0.762	-0.797	-0.87	0.75	1.99
Median	0.041	0.040	0.034	0.786	-0.769	-0.81	0.86	1.99
Mean	0.062	0.049	0.038	0.784	-0.769	-0.80	0.83	1.99
3rd Qu.	0.138	0.125	0.114	0.814	-0.733	-0.74	0.95	2.00
Max.	0.337	0.264	0.225	0.880	-0.649	-0.48	1.04	2.00
SD	0.129	0.116	0.093	0.040	0.042	0.10	0.15	0.01
VarWithin	4e-06	3e-07	3e-07	7e-3	9e-3	0.39	0.24	0.01

Table 3 reports the statistics of the estimates obtained for series with AsyTAR DGP using three PSO procedures: in the first one the structure of the the fitness uses an AR(1) scheme, and the estimates are close to zero; the second one involves in the estimation process an AR(2) structure, but even in this case, the simulation results show estimates almost close zero; in the last

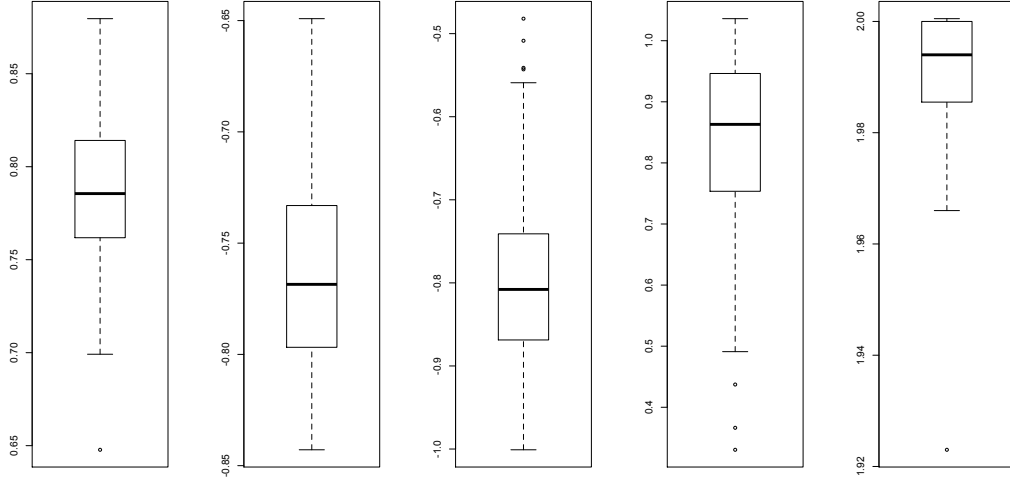


Figure 3: Boxplots of the estimates distributions of 100 simulated trajectories, for case of model (2)

case the procedure considers an AsyTAR scheme and the estimated values are very close to the theoretical ones.

In the case of data drawn from model (2), for each estimates of dimension 5 computed on 100 simulated trajectories, the distributions can be easily depicted. The boxplots corresponding to each dimension are shown in Figure 3, and the performed Shapiro-Wilk test on the Gaussianity hypothesis reveals that we can accept this hypothesis only for the first two dimensions, i.e. for the parameters of linear regime models in which the p -values are respectively 0.50 and 0.13; for the other parameters, namely the two thresholds and the delay, the tests present values lower than 0.001.

In order to control if this procedure well matches even with other DGPs, we considered some linear DGPs. Table 4 shows some statistics for the PSO applied to data simulated by an autoregressive model of order 1, enabling us to notice that the PSO procedure correctly estimates the parameter with either autoregressive or AsyTAR fitness function. We underline that when we use the AR(2) fitness function the ϕ_2 estimates are close to zero, whereas when the fitness function is AsyTAR the threshold estimates are equal to zero and

the autoregressive parameters of the two regimes are both estimated equal to 0.78 in mean. In a nutshell, the PSO algorithm captures the true value of the unknown parameter forcing the redundant parameter to zero, whereas the delay parameter has no sense in this framework.

Table 4: PSO estimation over 100 trajectory simulated by AR: some statistics.

$$\text{model } y_t = 0.8y_{t-1} + \epsilon_t$$

Statistics	PSO with	PSO with		PSO with				
	linear 1 par.	linear 2 par.		AsyTAR 5 parameter				
	a_1	a_1	a_2	$a_1^{(1)}$	$a_1^{(2)}$	r_1	r_2	d
Min.	0.743	0.712	-0.112	0.658	0.652	-0.36	-0.44	1.00
1st Qu.	0.782	0.769	-0.031	0.745	0.751	-0.16	-0.14	1.03
Median	0.799	0.801	-0.003	0.796	0.802	-0.02	0.01	1.07
Mean	0.797	0.799	-0.003	0.781	0.785	-0.03	0.04	1.11
3rd Qu.	0.813	0.829	0.033	0.816	0.818	0.10	0.16	1.17
Max.	0.854	0.910	0.101	0.866	0.861	0.39	0.82	1.37
SD	0.024	0.041	0.045	0.054	0.051	0.18	0.27	0.10
VarWithin	8e-05	6e-07	6e-07	2e-3	2e-3	1.13	1.46	0.17

In Table 5 we present the PSO estimation applied to data simulated with a SETAR model, only with two structure AR(2) and one AsyTAR. Even in this case we can notice that the procedure does not capture a possible linear scheme, while the SETAR scheme is correctly estimated for regime and delay parameters, whereas the threshold is estimated with some problems.

Nevertheless the results we obtained are encouraging, and much more important is to comment the results for AsyTAR data, reported in Table 3. The estimates of autoregressive parameters are very close to their theoretical values, just as the estimated delay parameter. Some problems arise for the estimation of the threshold parameters, but we are positive that these may be overcome if a more appropriate tuning of the PSO procedure is applied. When the fitness function of the PSO is not the correct one, the estimated model is just zero, thus no estimate is provided. This occurs even for SETAR data, as reported in Table 5. This may happen because the estimated model tries

Table 5: PSO estimation over 100 trajectory simulated by SETAR: some statistics.

$$y_t = \begin{cases} 0.8y_{t-1}^{(1)} + \varepsilon_t^{(1)} & \text{if } y_{t-2} \leq 0 \\ -0.8y_{t-1}^{(2)} + \varepsilon_t^{(2)} & \text{if } y_{t-2} > 0 \end{cases}$$

Statistics	PSO with linear 2 par.		PSO with AsyTAR 5 parameter				
	a_1	a_2	$a_1^{(1)}$	$a_1^{(2)}$	r_1	r_2	d
Min.	-0.343	-0.322	0.592	-0.869	0.25	-0.11	1.89
1st Qu.	-0.102	-0.095	0.755	-0.810	0.46	-0.02	1.98
Median	-0.004	-0.014	0.789	-0.787	0.52	0.00	1.99
Mean	-0.018	-0.019	0.783	-0.786	0.52	0.00	1.99
3rd Qu.	0.086	0.067	0.815	-0.765	0.58	0.02	2.00
Max.	0.304	0.217	0.890	-0.606	0.73	0.09	2.00
SD	0.129	0.116	0.051	0.042	0.09	0.03	0.02
VarWithin	3e-07	3e-07	0.019	0.011	0.303	0.02	0.01

to get simultaneously the different behaviours of the DGP both above and below the thresholds. In other words, using an estimation procedure on fewer parameters than those needed, the algorithm leads to a sort of compensation between the estimates.

4 Conclusion

The contribution seems of relevant interests as it provides new perspectives along three different patterns. Firstly, the proposed technique seems to obtain interesting results in the parameters estimation. Secondly, the preliminary results show robustness with respect to model miss-specification. Thirdly, in the cases of threshold model the technique allows to estimate simultaneously thresholds, delay and autoregressive parameters.

In particular, it seems that an appropriate dimension of the particles let us apply the same procedure to time series generated by different kind of processes, although this requires the definition of a fitness function that should cope with different DGPs. In fact, as the algorithm estimates the redundant parameters equal to zero, we improve the identification of the underlying process.

Eventually, we think that this model might be a useful tool to analyze real time series such as finance data that likely show asymmetric features.

References

- [1] BATTAGLIA, F., PROTOPAPAS, M.K., (2011): Multi-regime models for nonlinear nonstationary time series, *Computational Statistics*, 47, 277–295.
- [2] CHAN, K.S., TONG, H., (1990): On Likelihood Ratio Tests for Threshold Autoregression, *Journal of the Royal Statistical Society. Series B (Methodological)*, 52, 469–476.
- [3] CLERC, M., KENNEDY, J. (2002): The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6 (1), 58–73

- [4] HANSEN, B. E. (2000): , Sample Splitting and Threshold Estimation, *Econometrica*, 68, 575–603.
- [5] GONZALO, J., WOLF, M. (2005): Subsampling inference in threshold autoregressive models, *Journal of Econometrics*, 127, 201–224.
- [6] KENNEDY, J., EBERHART, R.C. (1995): Subsampling inference in threshold autoregressive models, in *In Proceedings of IEEE International Conference on Neural Networks*, 1942-1948.
- [7] LI, W.K., LAM, K., (1995), Modelling Asymmetry in Stock Returns by a Threshold Autoregressive Conditional Heteroscedastic Model *The Statistician*, 44, 333–341.
- [8] MARINGER, D.G., MEYER, M., (2008), Smooth Transition Autoregressive Model-New Approaches to the Model Selection Problem, *Non-linear Dynamical Methods and Time Series Analysis*, 12-1.
- [9] MENDES, R. (2004). Population Topologies and Their Influence in Particle Swarm Performance (PhD thesis). *Universidade do Minho*.
- [10] PEDERSEN, M.E.H., (2010), Good Parameters for Particle Swarm Optimization, *Hvass Laboratories*, Technical Report *HL1001*.
- [11] PETRUCCELLI, J.D., DAVIES, N. (1986): A Portmanteau Test for Self-Exciting Threshold Autoregressive-Type Nonlinearity in Time Series, *Biometrika*, 73, 687–694.
- [12] PIZZI, C. (2007): The asymmetric threshold model ASETAR(2;1,1), in *Sco2007*.
- [13] POTTER, N. (1995): A nonlinear approach to US GNP, *Journal of applied econometrics*, 10, 109–125.
- [14] SHI, Y., EBERHART, R. (1998): A modified particle swarm optimizer. in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 69–73.
- [15] TONG, H. (2011): Threshold Models in Time Series Analysis-30 years on, *Statistics and its Interface*, 4:2 107-118.

- [16] TONG, H., YEUNG, I. (1991): Threshold Autoregressive Modelling in Continuous Time. *Statistica Sinica*, 1, 411–430.
- [17] TONG, H., LIM, K.S. (1980): Threshold Autoregression, Limit Cycles and Cyclical Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42, 245–292.
- [18] TSAY, R.S. (1998): Testing and Modeling Multivariate Threshold Models, *Journal of the American Statistical Association*, 93, 1188–1202.
- [19] WU, B., CHANG, C.L., (2002): Using genetic algorithm to parameters (d,r) estimation for threshold autoregressive models. *Comput. Stat. Data Anal.*, 17, 241–264.