COMISEF
Computational Optimization Methods in Statistics, Econometrics and Finance

- Marie Curie Research and Training Network funded by the EU Commission through MRTN-CT-2006-034270 -

COMISEF WORKING PAPERS SERIES

WPS-009 20/02/2009

Time-varying Multi-regime Models Fitting by Genetic Algorithms

F. Battaglia
M. Protopapas

Time-varying Multi-regime Models Fitting by Genetic Algorithms

Francesco Battaglia and Mattheos K. Protopapas

February 20, 2009

Abstract

Many time series exhibit both nonlinearity and nonstationarity. Though both features have often been taken into account separately, few attempts have been proposed to model them simultaneously. We consider threshold models, and present a general model allowing for different regimes both in time and in levels, where regime transitions may happen according to self-exciting, or smoothly varying, or piecewise linear threshold modeling. Since fitting such a model involves the choice of a large number of structural parameters, we propose a procedure based on genetic algorithms, evaluating models by means of a generalized identification criterion. The performance of the proposed procedure is illustrated with a simulation study and applications to some real data.

KEYWORDS. Nonlinear time series; Nonstationary time series; Threshold model.

1 INTRODUCTION

Linear models are not enough flexible to fit many time series conveniently. Features like nonlinearity, structural change, and slow modifications in time of the dynamic structure, cannot be accounted for by the widely used autoregressive integrated moving average models, and suggested more complicated models (see, e.g., Priestley, 1988). Among them, most popular are the models which alternate in regime (see, e.g., Tong, 1990), by

[0]Francesco Battaglia is Professor (E-mail: *francesco.battaglia@uniroma1.it*) and Mattheos K. Protopapas (E-mail: *mantzos.protopapas@hotmail.com*) is Marie Curie Fellow, Department of Statistics, Sapienza University of Rome, Italy. This work was supported by European Commission through Marie Curie Research and Training Network COMISEF Computational Methods in Statistics, Econometrics and Finance, and by Italian Ministry of Education through a national research grant PRIN2007

explaining the current data through two (or more) alternative linear models, according to the value of an indicator (the *driving variable*). On the other hand, for taking into account structural modifications to which the series may be subject in time, several authors consider simple linear models (usually pure autoregressive) where the parameters are subject to change in time (e.g., Lin and Teräsvirta, 1994) .

Since often both nonlinearity and nonstationarity are found in the same series, it would be important, though difficult, to allow simultaneously regime alternating and time-varying coefficients, but this perspective was not considered by many authors since recently. Lundberg, Teräsvirta, and van Dijk (2003) proposed a model based on an autoregressive parameter which changes both according to time and to a driving variable, alternating between two regimes with smooth change (smoothly varying autoregressive, STAR). This formulation essentially includes the self-exciting threshold models (SETAR) of Tong (1990); we extend it to consider the piecewise linear threshold models (Baragona, Battaglia, and Cucina, 2004).

Though very general and flexible, the Lundberg et al. (2003) formulation has received theoretical attention and several extensions (for example, recently Kapetanios, Shin, and Snell, 2003; McAleer and Medeiros, 2008; Amado and Teräsvirta, 2008), but perhaps not so many applications as expected. A reason is that for building a time-varying smooth transition autoregressive model, many parameters have to be selected (e.g., order. thresholds) for which, on one hand, no analytical optimization method is available, and, on the other hand, substantive reasons for motivating this choice are rarely found.

Similar problems in statistical inference and time series analysis, where a huge number of solutions have to be compared, were successfully addressed using meta-heuristic methods. We propose here the use of genetic algorithms for building the time series models taken into account.

The plan of the paper is as follows. Section two addresses nonlinear threshold models,

and time-varying versions of them, introducing a general formulation and discussing identification and estimation issues, and motivating use of meta-heuristic methods. Section three introduces genetic algorithms and discusses how to set up a genetic algorithm for time-varying multi-regime model building. Section four reports results of a simulation study for evaluating the proposed method, and Section five presents some applications to real data. Conclusions and guidelines for future research are drawn in the last Section.

2 MULTI-REGIME MODELS

Many models have been proposed for reproducing time series which alternate between two (or more) different behaviors. Threshold models are useful when the working regime depends on the value of a variable (the driving variable): if it is smaller than a *threshold*, the series follows the first regime, and the second regime if it is larger. Usually, the driving variable is taken equal to the delayed values of the series itself (*self-exciting threshold model SETAR*, see, e. g., Tong, 1990), and in the two regimes two different linear stationary autoregressive models are defined. A SETAR model may be expressed as follows:

$$X_t = \begin{cases} \phi_0^{(1)} + \phi_1^{(1)} X_{t-1} + \ldots + \phi_p^{(1)} X_{t-p} + \varepsilon_t, & X_{t-d} \leq r \\ \phi_0^{(2)} + \phi_1^{(2)} X_{t-1} + \ldots + \phi_p^{(2)} X_{t-p} + \varepsilon_t, & X_{t-d} > r \end{cases} \tag{1}$$

where r is the threshold, d is the delay and $\{\varepsilon_t\}$ a gaussian purely white noise with mean zero and variance σ^2.

In a SETAR process, switching between regimes is sudden and the autoregressive coefficients ϕ_j are discontinuous as functions of the driving variable. An alternative, preserving continuity of the autoregressive coefficients, was proposed by Teräsvirta (1994) and is known as *smoothly varying threshold model* (STAR). The transition from the first

regime to the second is driven by a logistic function:

$$G(z) = [1 + \exp\{-\gamma(z - r)\}]^{-1} \tag{2}$$

where r is the threshold, and the model equation may be written as:

$$X_t = \{\phi_0^{(1)} + \sum_j \phi_j^{(1)} X_{t-j}\}\{1 - G(X_{t-d})\} + \{\phi_0^{(2)} + \sum_j \phi_j^{(2)} X_{t-j}\}G(X_{t-d}). \tag{3}$$

The parameter γ in (2) controls the speed of transition from the first to the second regime.

A further alternative is to suppose that the autoregressive coefficients are still continuous in X_{t-d}, but vary linearly with it, with different slopes in the two regimes. Such a model is called a *piecewise linear threshold autoregressive model* (PLTAR) (see Baragona et al., 2004) and may be defined by:

$$X_t = \begin{cases} \phi_0^{(1)} + \sum_j \left(\alpha_j^{(1)} + \beta_j^{(1)} X_{t-d}\right) X_{t-j} + \varepsilon_t, & X_{t-d} \leq r \\ \phi_0^{(2)} + \sum_j \left(\alpha_j^{(2)} + \beta_j^{(2)} X_{t-d}\right) X_{t-j} + \varepsilon_t, & X_{t-d} > r \end{cases} \tag{4}$$

where, for continuity, $\alpha_j^{(2)} = \alpha_j^{(1)} + \beta_j^{(1)} r$. The intercept terms ϕ_0 do not depend linearly on X_{t-d}, since they would be undistinguishable from α_d.

Using a spline function $S(z) = (z - r)I[z > r]$, where $I[.]$ is indicator function, the PLTAR model may also be written:

$$X_t = \phi_0^{(1)} + \{\phi_0^{(1)} - \phi_0^{(2)}\}I[X_{t-d} > r] + \sum_j \{\lambda_j + \mu_j X_{t-d} + \nu_j S(X_{t-d})\}X_{t-j} + \varepsilon_t \tag{5}$$

where $\lambda_j = \alpha_j^{(1)}, \mu_j = \beta_j^{(1)}, \nu_j = \beta_j^{(2)} - \beta_j^{(1)}$.

All these models may be seen as particular cases of the general state dependent model

of Priestley (1988):

$$X_t = \phi_0(X_{t-d}) + \sum_j \phi_j(X_{t-d}) X_{t-j} + \varepsilon_t \tag{6}$$

and are also particularizations of the so-called functional autoregressive models (see, e. g., Chen and Tsay, 1993). However, their essential feature is that, given the threshold, order and delay, the SETAR, the PLTAR and the STAR (except for γ) are linear in the unknown coefficients, which may therefore be simply maximum-likelihood estimated by least squares. The three threshold models may be expressed in form (6) using a general formulation for the autoregressive coefficients:

$$\phi_j(z) = \lambda_j + \mu_j\, z + \nu_j\, H(z). \tag{7}$$

Equation (7) gives a SETAR model for $\lambda_j = \phi_j^{(1)}, \mu_j = 0, \nu_j = \phi_j^{(2)} - \phi_j^{(1)}$ and $H(z) = I[z > r]$; it gives a STAR model if λ_j, μ_j, ν_j are as above and $H(z) = G(z)$ is a logistic; and gives a PLTAR if the notation introduced in (5) is used for λ_j, μ_j, ν_j, and $H(z) = S(z) = (z - r)I[z > r]$. However, there is no need of considering the SETAR model separately, because a smoothly varying model with sufficiently large value of the γ parameter is essentially equivalent to a SETAR model.

An important extension to nonstationarity was proposed by Lundberg et al. (2003), who proposed to allow the autoregressive coefficients to alternate between two regimes according to a smooth transition, both depending on delayed level X_{t-d}, and on time. The resulting time-varying smooth transition autoregressive model has a functional structure of type (6) where the autoregressive coefficients depend both on levels and on time through

two different logistic functions $G(X_{t-d})$ and $G'(t)$

$$\phi_j(X_{t-d}, t) = \{\alpha_{1j}[1 - G(X_{t-d})] + \alpha_{2j}G(X_{t-d})\}\{1 - G'(t)\} + \{\alpha_{3j}[1 - G(X_{t-d})] + \alpha_{4j}G(X_{t-d})\}G'(t).$$

A similar time-varying generalization may be proposed also for the general threshold model (7), by allowing the coefficients λ_j, μ_j and ν_j also vary in time, according to a threshold model driven by time:

$$\begin{aligned} \lambda_j(t) &= a_j + b_j t + c_j H'(t) \\ \mu_j(t) &= d_j + f_j t + g_j H'(t) \\ \nu_j(t) &= h_j + k_j t + m_j H'(t) \end{aligned} \tag{8}$$

which again may be particularized to a PLTAR or STAR by choosing the appropriate form for the function $H'(t)$. On combining (7) and (8) we obtain a general time-varying threshold model:

$$X_t = \phi_0(t, X_{t-d}) + \sum_j \phi_j(t, X_{t-d})X_{t-j} + \varepsilon_t$$

with

$$\phi_j(t, X_{t-d}) = a_j + b_j t + c_j H'(t) + d_j X_{t-d} + f_j t X_{t-d} + g_j X_{t-d} H'(t) + h_j H(X_{t-d}) + k_j t H(X_{t-d}) + m_j H'(t) H(X_{t-d}). \tag{9}$$

Model (9) is unidentifiable since some of its parameters are redundant, but they are constrained to zero according to the type of dependence upon time and/or levels. For

Table 1: Time-varying threshold model types and constrained coefficients

Type in time	Type in levels	Coefficients constrained to zero
Stationary	Linear	$b_j, c_j, d_j, f_j, g_j, h_j, k_j, m_j, j = 0, \dots, p$
Stationary	STAR	$b_j, c_j, d_j, f_j, g_j, k_j, m_j, j = 0, \dots, p$
Stationary	PLTAR	$b_j, c_j, f_j, g_j, k_j, m_j, j = 0, \dots, p; d_0$
STAR	Linear	$b_j, d_j, f_j, g_j, h_j, k_j, m_j, j = 0, \dots, p$
PLTAR	Linear	$d_j, f_j, g_j, h_j, k_j, m_j, j = 0, \dots, p$
STAR	STAR	$b_j, d_j, f_j, g_j, k_j, j = 0, \dots, p$
STAR	PLTAR	$b_j, f_j, k_j, j = 0, \dots, p; d_0, g_0$
PLTAR	STAR	$d_j f_j, g_j, j = 0, \dots, p$
PLTAR	PLTAR	d_0, f_0, g_0

example, if the dependence on levels is STAR, then, as observed above, $\mu_j = 0$, therefore $d_j = f_j = 0$; if dependence on time is of PLTAR type, then the intercept autoregressive coefficient ϕ_0 is not allowed to vary linearly with t, which implies $d_0 = f_0 = g_0 = 0$, and so on. Table 1 considers the nine different model types and describes, for each of them, what coefficients are constrained to zero.

Note that we have assumed the same maximum lag p for any dependence type: this is an unnecessary simplification, and one further extension is advisable, allowing the maximum order of the coefficients depending on time be different from that related to the dependence on levels. We call p_L the maximum lag at which the autoregressive coefficients ϕ_j vary with level X_{t-d}, and denote by p_T the maximum lag at which they vary with time. Thus, $\lambda_j(t)$, $\mu_j(t)$ and $\nu_j(t)$ are all zero for $j > p_L$ and any t, while they are constant with respect to t when $j > p_T$. On continuing to denote by p the overall order, obviously with $p \geq p_L$ and $p \geq p_T$, and assuming $q = \min(p_L, p_T)$, the maximum lags of the coefficients in the general notation (9) modify as follows:

$$a_j,\ j \leq p\,;\ b_j, c_j,\ j \leq p_T\,; d_j, h_j,\ j \leq p_L\,;\ f_j, g_j, k_j, m_j,\ j \leq q.$$

Though introducing different orders may appear more complex, it achieves a reduction of the number of coefficients to be estimated; also, if $p_L = 0$ the model is linear, and if

$p_T = 0$ it is stationary.

Summarizing, a time-varying threshold model of the proposed type is characterized by the following parameters:

a) Structural parameters. Model type in time: stationary, PLTAR or STAR (including self-exciting threshold). Model type in levels: linear, PLTAR or STAR. Delay of the driving variable, d. Orders: in levels p_L, in time p_T and overall p. Threshold in time r_T. Threshold in levels r_L. Logistic speed parameters in levels, γ_L and in time, γ_T.

b) Autoregressive coefficients $a_j, b_j, c_j, d_j, f_j, g_j, h_j, k_j, m_j$.

We have separated the parameters into two categories because they have a completely different behavior in model fitting. Conditional on values of the structural parameters sub a, the model is linear in the autoregressive coefficients sub b, thus they may be easily estimated by least squares on equating to zero the derivatives of the sum of squared residuals; if the innovations $\{\varepsilon_t\}$ are gaussian, these are maximum likelihood estimates. On the other hand, no analytical optimization method is available for estimating the structural parameters. In principle we should consider all possible combinations of values of the model types, delay, orders, thresholds (and gammas), estimate the autoregressive coefficients and the residual sum of squares, and select the best. It may be argued that this does not apply to thresholds and gammas since they are continuous variables, and their values could be estimated together with the autoregressive coefficients employing a nonlinear optimization algorithm. However, this idea proved impractical because of computational difficulty, and all published applications of threshold models (except those using genetic algorithms) are based upon on a discrete grid search for threshold and gammas.

Therefore we are faced with a large, discrete space of solutions to be scanned for the

best value, without analytical tools. Problems of this kind are frequently encountered in statistical applications and have been recently addressed by means of meta-heuristic methods, in particular genetic algorithms. Genetic algorithms have been proposed in the last decade for selecting variables in regression (Chatterjee, Laudato, and Lynch, 1996), for autoregressive moving average model fitting (Gaetan, 2000), for outlier detection (Crawford and Wainwright, 1995), for cluster analysis (e.g. Bandyopadhyay and Maulik, 2002), to cite but a few. The use of genetic algorithms was suggested specifically for building threshold models (Wu and Chang, 2002; Davis, Lee, and Rodriguez-Yam, 2006) and other nonlinear dynamics (e. g. for bilinear models, see Chen, Cherng, and Wu, 2001).

3 GENETIC ALGORITHMS

3.1 The Structure of a GA

A genetic algorithm (Holland, 1975), is a heuristic algorithm, inspired by evolutionary processes of ecological systems, that finds optimal (or near-optimal) solutions to complex optimization problems. In a genetic algorithm, the potential solutions of the problem at hand are coded in chromosomes. A chromosome is a string of binary digits or other symbols, that corresponds to a given solution of the problem (e.g. if our goal is to find the x that minimizes a function $f(x)$ in a given interval $[a, b]$, chromosomes correspond to real numbers that belong in the interval $[a, b]$). The usual genetic algorithm (also called "canonical" genetic algorithm, Goldberg, 1989) use sets of these chromosomes, called "populations" that evolve through "generations" (usually, the initial population is chosen at random), by the operators of "crossover" and "mutation", in order to discover the chromosome-solution that corresponds to the optimal solution to the problem.

A "fitness function" measures the eligibility of a chromosome to become a "parent" for

the chromosomes of the next generation. The fitness function is linked to the value of the objective function (the value of the $f(x)$ in the previous example) by either a mathematical transformation that ensures that the fitness of the chromosomes is an increasing function of their performance in the problem at hand (i.e. in our minimization example, lower values of $f(x)$, should imply higher values for the fitness of x), or simply by ordering the chromosomes of the current population in terms of their performance ("ordered fitness" – if we had a population of 50 chromosomes in the minimization example, the chromosome that corresponds to the x that leads to the lower value of $f(x)$ should have fitness equal to 50, the chromosome with the second lower $f(x)$, fitness equal to 49, and so on).

The usual rule for selecting parents is the "roulette wheel selection" (Goldberg, 1989), meaning that the probability of a chromosome to be selected as a parent is proportional to its fitness. So for a total of half the number of chromosomes in the populations, two parents are selected, using sampling with replacement, and probability proportional to fitness, that are used to form two children.

The children chromosomes may be formed by operating the "crossover" operator to the two parents, with a given probability (probability of crossover). In the usual form of crossover (single point crossover, Holland, 1975), the (usually) binary strings of the two parents are cut at a specific point and the two substrings are exchanged. So assuming that the first chromosome is '01110001', the second is '11001110' and the cutting point is the fifth binary digit (bit), the first child should have '0111' in the first four bits (from the second parent) and '1110' in the last four bits. So it should be '01111110'. The second child is formed by the other two sub-strings: '1100' from the second parent and '0001' taken from the first, and is '11000001'. The cutting point may be selected randomly (random point crossover, Holland, 1975), two or more cutting points may be used (multi-point crossover, Holland, 1975), or for any given bit, a parent is selected at random and its corresponding bit is assigned to the corresponding bit of the first child, while the other

parent's bit is assigned to the corresponding bit of the second child (uniform crossover, Syswerda, 1989).

Finally, the "mutation" operator is performed upon the children chromosomes, to create the final chromosomes of the new generation. Each binary digit of the chromosome is subject to inversion (from 1 to 0 and vice versa) under a given (small) probability. That resembles the mutation of animal chromosomes, which can change due to chance only, and allows the introduction of new genetic material into the "gene pool". If all the chromosomes in a population have for example, "0" at their first position-gene, there is no way that a "1" at the first position of any of the children chromosomes, is introduced by the sole use of the crossover operator. So, mutation gives the opportunity for entirely new genes to be introduced in the population and be tested. Genetic algorithms are stochastic procedures and their properties have been studied mainly using Markov Chain Theory or by means of Statistical Mechanics (see e.g. Reeves and Rowe, 2003, for a review), obtaining several convergence results for the best solution obtained as the number of generations increases. In the case of a canonical algorithm with elitist strategy, like ours, Rudolph (1997) has shown that the difference between the best fitness reached in the n-th generation and the optimal fitness value is a non-negative supermartingale converging to zero almost surely as $n \to \infty$.

A pseudo code for "canonical" genetic algorithm is as follows:

1. Set the values of the parameters regarding population size (pop), probability of crossover (p_{cr}), probability of mutation (p_{mut}), number of generations (gen) and all the other parameters relevant to this application.

2. Generate random initial population of chromosomes.

3. Calculate the fitness values of the chromosomes.

4. Select two of the chromosomes as parents, with probability proportional to their

fitness.

5. If crossover is used (depending on the probability of crossover mentioned above), combine the genes of these chromosomes using the crossover operator to form two children chromosomes. In the case no crossover is applied, the children chromosomes will be initially, just copies of the parent chromosomes.

6. Then apply the mutation operator to the children chromosomes, so that some (if any) random bits of the children chromosomes are inverted.

7. Repeat steps 4-6, until *pop* children chromosomes have been formed.

8. Repeat steps 3-7 until the specified number of generations (*gen*) have passed.

3.2 GA Implementation

3.2.1. Chromosome Encoding.

In the model selection problem analyzed here, the following decision variables are relevant: the type of model in time and in levels, the order in time and in levels as well as the overall order, the delay of the driving variable in STAR and PLTAR models as well as the thresholds (in time and levels) and, finally, in the case of a STAR model, the γ coefficients (in time and/or in levels). Therefore the chromosome (a string of binary digits) consists of 10 sub-strings, each one corresponding to one of these decision variables. Furthermore, crossover is performed in such a way, that these bit-strings remain intact when inherited. In all fields where there is a "1-1" correspondence between the binary value of the string and the value of the corresponding decision variable, the following equation is used

$$x = \sum_{n=1}^{L} e_n 2^{n-1} + d \tag{10}$$

where x is the value of the decision variable, L is the length of the corresponding bit-field in the chromosome, e_n, the n_{th} bit of the bit-field, and d the minimum value of the decision variable (so for the delay that assumes values $1, 2, ..., 8$, $L = 3$ and $d = 1$).

The first two bits of a chromosome encode the type of model in levels. Two bits are required, because there are three different choices for the type of the model: stationary, STAR and PLTAR. A value of one corresponds to a STAR model, two to a PLTAR model and zero and three to a stationary model. The same encoding is used for the next two bits of the chromosome that denote the type of the model in time.

The order of the model in time ranges from 0 to 7, and needs 3 bits. The same holds true for the next bit-string, which denotes the order of the model in levels, and resides on bits 8-10. The overall order of the model is greater or equal to the orders in time and in levels, with a maximum value of 14. We use a 3-bit string for the corresponding field in the chromosome (bits 11-13), to encode the overall order. The value of that field (between 0 and 7, because 3 bits are used) is added to the maximum value of the fields denoting the order in time and in delays. That results to the desired value between max{Order in Time, Order in Levels} and 14.

Bits 14-16 encode the delay of the driving variable in a STAR or PLTAR models. Possible values are between 1 and 8, so three bits are required.

The fields denoting the thresholds in time and in levels follow, and each one consists of 12 bits. From the models comes the conclusion that instead of choosing a threshold ϵ in-between two consecutive (in terms of magnitude) observations, $Y_k \leq \tau < Y_{k+1}$, one can choose a threshold value that equals the maximum observed value Y_k that is not larger than the initial threshold τ and have the same essential effect. Therefore we constrain the threshold value to be one of the observed values of the time series. We further constrain the number of observations per regime, to at least $1/m$ of the total. So, the threshold in time can be any integer from time N/m to time $(m-1)N/m$, where N is the total

number of observations, and $m > 1$. For the threshold in levels, we use the sorted set of the observed values, and therefore, the threshold in levels equals the $2_{nd}, 3_{rd}, \ldots (m-1)_{th}$ (N/m)-quantile of the set of the observed values. To achieve the previous result, another equation is used after the calculation of x from equation (10). The actual threshold in time is given by

$$T_t = \left[\frac{(m-2)Nx}{m(2^L-1)}\right] + \frac{N}{m} \tag{11}$$

where $[z]$ is the integer part of z, x is calculated from (10). For calculating the threshold in levels we have to do an additional step. First of all we use the sequence of the observed values of the time series, sorted in ascending order $Y_1, Y_2, \ldots, Y_N$; then we decode the corresponding sub-field using equations (10) and (11) as before, to calculate the index of the sorted series that will determine the threshold in levels by

$$T_L = Y_{T_t} \tag{12}$$

The index T_t calculated from (11) is not used on itself, but it is considered to be the index of the observation of the sorted series that, finally, constitutes the threshold in levels.

3.2.2. Encoding smooth transition parameters.

The final sub-strings of the chromosome encode the γ parameter of the STAR model in time and/or in levels. This may be done by mapping a L-bit binary number into the real interval (γ_1, γ_2), as usual:

$$\gamma = \gamma_1 + \frac{(\gamma_2 - \gamma_1)x}{2^L - 1} \tag{13}$$

after having chosen a minimum value γ_1 and a maximum value γ_2 for the parameter γ. Note that the γ parameter controls the speed of change from 0 to 1 in the logistic

function. As a maximum value, we can select one that makes the STAR model eventually indistinguishable from a SETAR, i.e. such that, for a sufficiently small value ϵ the logistic function $G(\cdot)$ has value ϵ immediately before the threshold, and value $(1-\epsilon)$ immediately after. Let us consider first the time-varying case, and suppose that t_0 is the threshold in time. Then we may put

$$G(t_0-1) = \{1+e^{\gamma}\}^{-1} \leq \epsilon \; ; \; G(t_0+1) = \{1+e^{-\gamma}\}^{-1} \geq 1-\epsilon$$

which amounts to

$$\gamma \geq \log\left(\frac{1-\epsilon}{\epsilon}\right) \tag{14}$$

For the gamma parameter of the logistic function in level, a similar argument may be applied to the ordered sequence of the observed values. If Y_T denotes the threshold in levels, then

$$G(Y_{T-1}) = \left[1+e^{-\gamma(Y_{T-1}-Y_T)}\right]^{-1} \leq \epsilon \; ; \; G(Y_{T+1}) = \left[1+e^{-\gamma(Y_{T+1}-Y_T)}\right]^{-1} \geq 1-\epsilon$$

For evaluating approximately the constraints we put

$$Y_{T+1} - Y_T \simeq Y_T - Y_{T-1} \simeq \frac{Y_N - Y_1}{N} = s$$

so that they are solved by

$$\gamma \geq \frac{1}{s}\log\left(\frac{1-\epsilon}{\epsilon}\right).$$

In order to select the minimum gamma values, we shall assume that the change from ϵ to $1-\epsilon$ in the logistic function requires an interval not longer than $\frac{1}{q}$ of the full observation

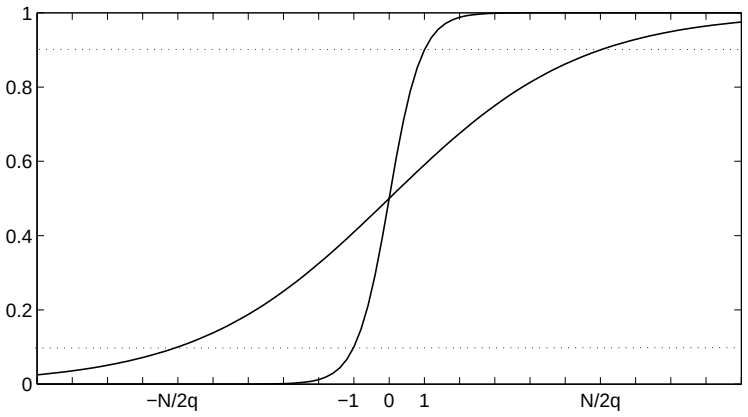

Figure 1: Logistic function with γ equal to minimum and maximum selected values. Threshold=0 and $\epsilon = 0.1$

interval. Consequently in a time-varying STAR we have

$$G\left(t_0 - \frac{N}{2q}\right)\left[1 + \exp\left(\frac{\gamma N}{2q}\right)\right]^{-1} \geq \epsilon \; ; \; G\left(t_0 + \frac{N}{2q}\right)\left[1 + \exp\left(\frac{-\gamma N}{2q}\right)\right]^{-1} \leq 1 - \epsilon$$

leading to

$$\gamma \leq \frac{2q}{N} \log\left(\frac{1-\epsilon}{\epsilon}\right).$$

Thus the search interval for gamma in time will be $[\log(1-\epsilon)/\epsilon, (2q/N)\log(1-\epsilon)/\epsilon]$, and for gamma in levels $[N/(Y_N - Y_1)\log(1-\epsilon)\epsilon, 2q/(Y_N - Y_1)\log(1-\epsilon)/\epsilon]$, see Figure 1 where the logistic function is drawn with the two extremal values of the parameters γ.

3.2.3. Fitness Evaluation.

Each chromosome specifies model type and all relevant structure parameters. Conditional on these choices, the model is linear in the autoregressive coefficients, that may be estimated by gaussian maximum likelihood, and the residual variance estimate $\hat{\sigma}^2$ may also be computed. For evaluating the fitness of each chromosome, we suggest to use an iden-

tification criterion for time series modeling. A generalized AIC criterion (Bhansali and Downham, 1977) assigns to chromosome M with p parameters the score

$$GAIC_c(M) = N \log \hat{\sigma}^2 + cp$$

where c is a tuning constant. The original Akaike's criterion is obtained for $c = 2$, the Schwartz's criterion for $c = \log N$. Obviously the fitness value should be inversely related to the identification criterion value, therefore we use a simple negative exponential transformation

$$fitness(M) = \exp\left[\frac{-GAIC_c(M)}{N}\right] = \hat{\sigma}^{-2} \exp\left(\frac{-cp}{N}\right)$$

The strategy of minimizing an identification criterion stems from the assumption that there is not an unique exact model suitable for our data, rather we only wish to select inside a set of models, that ensuring the best fit. This would appear in contradiction with the strategy of Lundberg et al. (2003) based on testing the null hypothesis of a stationary linear autoregressive process against nonlinearity and nonstationarity. However we can show that with a suitable choice of the tuning constant our strategy is essentially equivalent to the testing procedure of Lundberg et al. (2003). They use as a test statistic either

$$S_1 = N\frac{(SSR_0 - SSR_1)}{SSR_0}$$

or

$$S_2 = \frac{(SSR_0 - SSR_1)/(3p+1)}{SSR_1/(N-4p-2)}$$

where SSR_0 is the sum of squared residuals when fitting a stationary linear autoregressive model (Model M0), and SSR_1 is the sum of squared residuals of a time-varying STAR model (model M1). The null hypothesis is rejected if S_1 exceeds the $(1-\alpha)$-quantile of a chi-square with $3p+1$ degrees of freedom, or if S_2 exceeds the $(1-\alpha)$-quantile of a F distribution with $3p+1$ and $N-4p-2$ degrees of freedom, which are essentially equivalent for N large. Thus we shall select a time-varying STAR model if $S_1 > \bar{\chi}^2_{3p+1}(1-\alpha)$, or

$$N \log \hat{\sigma}_1^2 + \bar{\chi}^2_{3p+1}(1-\alpha) < N \log \hat{\sigma}_0^2 \tag{15}$$

Now, since model M1 has $4p+2$ parameters and model M0 has only $p+1$, the values of the identification criteria are $GAIC_c(M0) = N \log \hat{\sigma}_0^2 + c(4p+2)$; $GAIC_c(M1) = N \log \hat{\sigma}_1^2 + c(p+1)$, and (15) is equivalent to

$$GAIC_c(M1) - GAIC_c(M0) < \bar{\chi}^2_{3p+1}(1-\alpha) + c(3p+1)$$

Therefore the same decision (rejecting M0) is taken using the generalized identification criterion if $GAIC_c(M1) < GAIC_c(M0)$ or

$$c > \frac{1}{3p+1}\bar{\chi}^2_{3p+1}(1-\alpha) \tag{16}$$

Note that the right hand side of (16) is decreasing as p increases. When $p=1$ its value is 2.37 for $\alpha = 0.05$, 2.78 for $\alpha = 0.025$ and 3.32 for $\alpha = 0.01$.

3.2.4. Genetic Operators and Parameters.

As stated in the previous section, the initial population is chosen at random. The size of the population (number of chromosomes) is set to 50, after considering suggestions by Alander (1992), who suggests that the number of chromosomes in the population should be between the length of the chromosome and twice that number (between 55 and 110 in

our case), and Reeves (1993) who suggests that the size of the population should be such, that the probability p that any allele (bit value) is present at each locus (bit position) is high enough, and proposes the formula

$$n \geq \left[1 + \log\left(-\frac{l}{\log p}\right)\frac{1}{\log 2}\right]$$

where $l \equiv$ number of genes, $p \equiv$ probability and $n \equiv$ population size. For $p > 0.999$ a population size of 20 chromosomes would be enough in our case.

We use a canonical genetic algorithm, so the roulette-wheel rule is used for parents' selection, as defined in the previous section. After selecting the two parent chromosomes, random-point crossover is used to form the children chromosomes. However we restrict the viable cutting points to the boundaries of the sub-strings defining the type of models, the lags etc. in order to ensure that these sub-strings are not "cut in half" during the crossover operation. Thus, a child chromosome inherits exactly one of its parents' sub-strings. Finally, mutation is operated to the two children chromosomes to acquire their final forms. As stated in the previous section, mutation is bit-wise independent, with a fixed probability p_{mut}. In order to decide this probability we postulate that the probability that at least one bit is mutated in every chromosomes is Q, yielding that

$$(1 - p_{mut})^{55} = 1 - Q = 0.5 \Rightarrow p_{mut} = 1 - (1 - Q)^{1/55}$$

For $Q = 0.25$, p_{mut} equals 0.005, for $Q = 0.5$ we obtain $p_m = 0.0125$, and for $Q = 0.95$, $p_{mut} = 0.05$. Finally, an elitist strategy was used, meaning that the chromosome that has the highest fitness, is inherited (as it is) in the population of the next generation, replacing a - randomly selected - child chromosome.

4 A SIMULATION STUDY

The performance of the proposed procedure was checked on a set of simulated series. We selected the following time series models:

Model 1 First-order stationary autoregressive model $X_t = \phi X_{t-1} + \varepsilon_t$ with $\{\varepsilon_t\}$ independent gaussian $N(0,1)$ variables, and values for the autoregressive coefficients -9., -.7, -.5, -.3, 0.0, o.3, 0.5, 0.7, 0.9.

Model 2 Smooth transition STAR models with order one, not time-varying:

$$X_t = \{\phi_1[1 - G(X_{t-d})] + \phi_2 G(X_{t-d})\}X_{t-1} + \varepsilon_t$$

with $\phi_1 = 0.5, \phi_2 = -9., -.7, -.5, -3., 0.0, 0.3, 0.5, 0.7, 0.9$, assuming delay $d = 1$, threshold equal to zero and parameter γ equal to 15, so that

$$G(z) = [1 + \exp\{-15z\}]^{-1}.$$

Using notation of model (9) the coefficient values are $a_1 = 0.5, h_1 = -a_1 + [0, \pm.3, \pm.5, \pm.7, \pm.9]$.

Model 3 Stationary in levels and time-varying according to a smooth transition, first order:

$$X_t = \{\phi_1[1 - G'(t)] + \phi_2 G'(t)\}X_{t-1} + \varepsilon_t$$

with same coefficient values as Model 2, threshold in time equal to N/2 and parameter γ equal to 0.1:

$$G'(t) = [1 + \exp\{-0.1(t - N/2)\}]^{-1}$$

Table 2: Coefficient values for simulated series according to Model 4

series	1	2	3	4	5	6	7	8	9	10
a_1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
c_1	-.9	-.6	-.4	-.2	0.0	0.4	0.1	-.1	-.3	-.5
h_1	0.4	0.4	0.4	0.4	0.4	-.9	-.9	-.9	-.9	-.9

Model 4 Time-varying threshold with smooth transition both in time and in levels, first order:

$$X_t = \big(\{\phi_1[1-G(X_{t-d})] + \phi_2 G(X_{t-d})\}[1-G'(t)] + \{\phi_3[1-G(X_{t-d})] + \phi_4 G(X_{t-d})\}G'(t)\big)X_{t-1} + \varepsilon_t$$

with delay one and threshold in levels zero, and threshold in time $N/2$, gamma in levels five and gamma in time 0.1. The coefficient values are chosen so that there is no interaction between time and levels regimes: $\phi_4 - \phi_3 = \phi_2 - \phi_1$. It may be easily seen that the correspondence between these coefficients and those in (9) is as follows:

$$a_1 = \phi_1 \ ; \ c_1 = \phi_3 - \phi_1 \ ; \ h_1 = \phi_2 - \phi_1 \ ; \ m_1 = \phi_2 - \phi_1 - \phi_3 + \phi_4 = 0.$$

We adopted 10 combinations of values as shown in Table 2.

Model 5 Time-varying threshold with SETAR transition both in time and in levels, first order:

$$X_t = \big(\{\phi_1 I[X_{t-1} \le 0] + \phi_2 I[X_{t-1} > 0]\}I[t \le t_0] + \{\phi_3 I[X_{t-1} \le 0] + \phi_4 I[X_{t-1} > 0]\}I[t > t_0]\big)X_{t-1} + \varepsilon_t$$

threshold in levels zero and threshold in time $t_0 = N/2$. Nine different coefficient values were simulated according to Table 3.

Table 3: Autoregressive coefficient values selected for Model 5 simulation, $t_0 = N/2$

series number	1	2	3	4	5	6	7	8	9
$x \leq 0, t \leq t_0$	0.80	0.75	0.70	0.65	0.60	0.55	0.50	0.45	0.40
$x \leq 0, t > t_0$	-.10	-.15	-.20	-.25	-.30	-.35	-.40	-.45	-.50
$x > 0, t \leq t_0$	-.40	-.45	-.50	-.55	-.60	-.65	-.70	-.75	-.80
$x > 0, t > t_0$	0.60	0.55	0.50	0.45	0.40	0.35	0.30	0.25	0.20

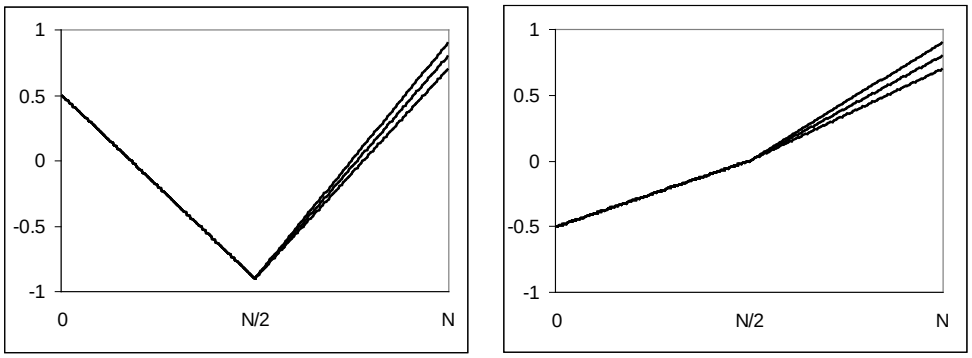

Figure 2: Dependence of the autoregressive coefficient on time for the six series simulated according to Model 6. Left panel: series 1–3, right panel: series 4–6

Model 6 Self-exciting threshold SETAR in levels, with time-varying autoregressive coefficients following a PLTAR model in time, first order without interaction. The autoregressive coefficient changes with time, while its sign changes according to the sign of X_{t-1}:

$$X_t = \{\lambda + \mu t + \nu S'(t)\}\text{sign}(X_{t-1}) + \varepsilon_t$$

where $S'(t) = (t - t_0)I[t > t_0]$ and threshold $t_0 = N/2$. Thus, the autoregression depends on the absolute value of the delayed observation, and varies with time. Model 6 corresponds to notation (9) with only a_1, b_1 and c_1 different from zero. Six different coefficient value combinations were used, the evolution in time of the autoregressive coefficients is shown in figure 2.

For each parameter set of each model, 100 series with $N = 500$ observations were

simulated, and on each of them the proposed canonical genetic algorithm with elitist strategy was run, with mutation probability 0.05 and cross-over probability equal to one. For coding thresholds we assumed the minimum number of observations per regime equal to $N/5 = 100$. Finally, for encoding the γ parameters we assumed $\epsilon = 0.01$ leading to the intervals $(46/N\,,\ 4.6)$ in time and $(46/(Y_N - Y_1)\,,\ 4.6N/(Y_N - Y_1))$ in levels. We used the fitness function obtained by a negative exponential transformation of the identification criterion, selecting the value $c = 3$ for the tuning constant. Thus the fitness value of model M with p coefficients is

$$f(M) = \exp\{-3p/N\}\hat{\sigma}^{-2}(M)$$

where $\hat{\sigma}^2(M)$ is the residual variance of the estimated model.

What we are mainly concerned with is the frequency of selection of the correct model by the genetic algorithm. In the analysis of each single series, the success in selecting the true data generating model depends on two main factors:

i) the genetic algorithm discovers the solution with the largest fitness value

ii) the correct model actually gives, on the analyzed series, a larger fitness than any other competing model.

While *i)* depends on the genetic algorithm features, and the convergence results ensure that an arbitrary approximation may be obtained by increasing the number of generations, achievement of *ii)* depends on the bias in estimating the model coefficients, and even on the choice of the tuning constant, therefore it may be improved by increasing the number of observations, but cannot be controlled when analyzing a single series. In order to separate the effects of the two factors we computed for each analyzed series what we call the optimal model, i.e., the model that attains the best fitness conditional on the knowledge of the actual values of the orders, thresholds, delays and gammas. In this way,

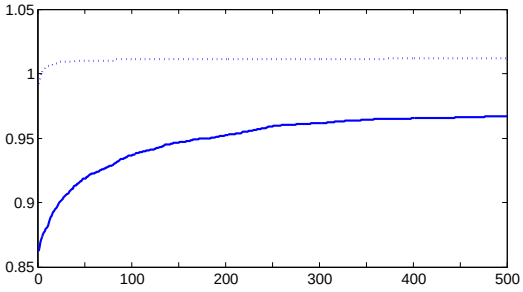

Figure 3: Behavior of the best fitness reached at a given generation, average for the 100 replications of series 1 of Model 1 (dotted line) and series 1 of Model 6 (continuous line)

any difference between the optimal model, and that selected by the genetic algorithm, arises because the algorithm did not discover the right chromosome (the one coding the true orders, delays, thresholds and gammas).

The choice of the number of generations to be run is in general problem-dependent, and few theoretical guidelines may be invoked. One possible popular solution is continuing the algorithm until no new better solution is found for a sufficiently large number of generations. As an alternative, for the present simulations we performed a pilot study to describe the typical behavior of the best solution found as generations flow. Some results are shown in figure 3 where the average of the best fitness achieved on the 100 replications of the first series of Models 1 and 6 are plotted against the generation number. As it may be expected, when the data generating process is more complicate, a larger number of generations is needed to approach the optimum. Basing on these results, we decided to run 500 generations when analyzing nonlinear or time-varying series, while we stopped at generation 100 when analyzing the stationary series of Model 1.

First of all we consider the size of our procedure. Results of the application to the series simulated according to the stationary autoregressive Model 1 are shown in table 4, where the frequency of stationary linear model selection for 500 replications each of the nine series are reported. The first row relates to the optimal choice as defined above, i.e.,

Table 4: Frequency of linear stationary model selection, 500 replications of series generated according to Model 1 and various first-order autoregressive coefficients

ϕ_1	-9.	-.7	-.5	-.3	0.0	0.3	0.5	0.7	0.9
optimal	.978	.976	.966	.960	.972	.976	.976	.978	.976
GA	.938	.928	.898	.914	.932	.928	.936	.946	.950

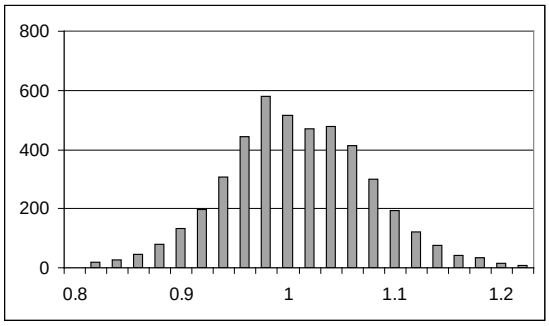

Figure 4: Distribution of the best fitness obtained in genetic algorithm, series from Model 1

computation of fitness based on knowledge of the structural parameters, and the second row refers to the genetic algorithm best fitness solution found after 500 generations.

We selected a value of 3 for the tuning constant, that corresponds in the Lundberg et al. (2003) testing strategy to a size between 0.025 and 0.01, and the observed size with the optimal choice is very similar, ranging from 2.2 to 4% with an average of 2.7. Results are slightly worse when looking at the actual application of genetic algorithm (therefore, without any information about orders), where the frequency of selection of an uncorrect model ranges from 0.05 to 0.1 with an average of 0.07.

To get an idea of the results variability, Figure 4 shows the distribution of the best fitness reached by the genetic algorithm for the whole set of 4500 analyzed series of Model 1.

Let us turn now to examine series which exhibit only one feature, nonlinearity (Model 2) or nonstationarity (Model 3). Figures 5 for Model 2, and 6 for Model 3, show the frequencies of selection of different models, both according to the optimal choice (dotted

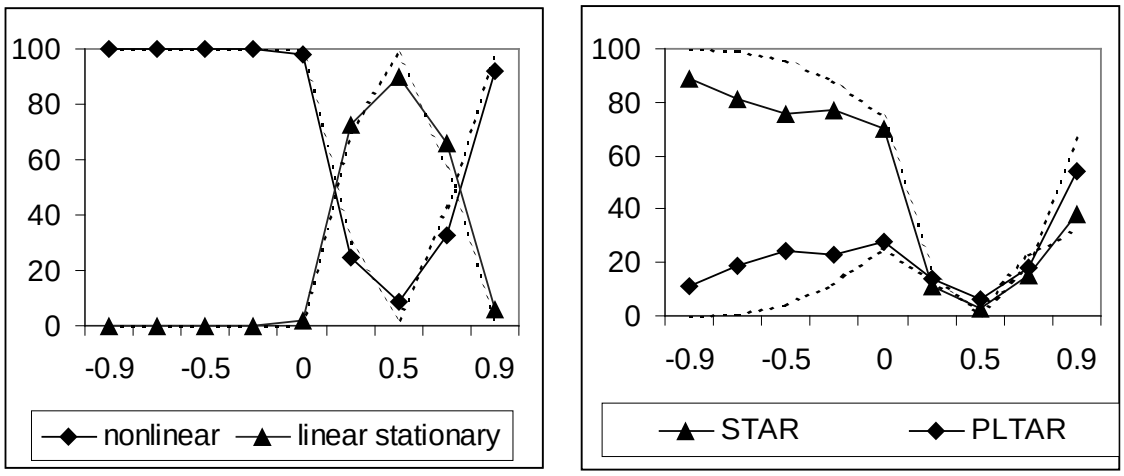

Figure 5: Frequency of selection of different models in series generated according to Model 2 (STAR). In the first regime $\phi_1 = 0.5$, ϕ_2 is shown in abscissa. Continuous line: results of the genetic algorithm application; dotted line: results according to optimal fitness

line) and the results of genetic algorithm (continuous line). Abscissas relate to the values of the autoregressive coefficient above (or after) the threshold, while the AR coefficient in the first regime equals, in both models, 0.5; thus, when $\phi_2 = 0.5$ the series are linear stationary. The left panel compares the selection frequency of nonlinear or time-varying model against a linear stationary autoregression, and the right panel describes which type of threshold model was selected. Results are based on 500 generations, and an overall evaluation suggests that the genetic algorithm is successful in optimization, because the continuous and dotted lines are always very near. Concerning the efficiency of method, conclusions are similar to those of Lundberg et al. (2003). When the difference from the simple AR case is clear cut, since ϕ_2 is far from ϕ_1, the frequency of correct selection is very large, while it decreases when the difference $\phi_2 - \phi_1$ becomes small. In addition, since in our framework two different types of dependence are taken into account, smooth transition and piecewise linear, an uncorrect choice between the two types may happen, though with small frequency, and generally only when the series is nearly stationary or linear.

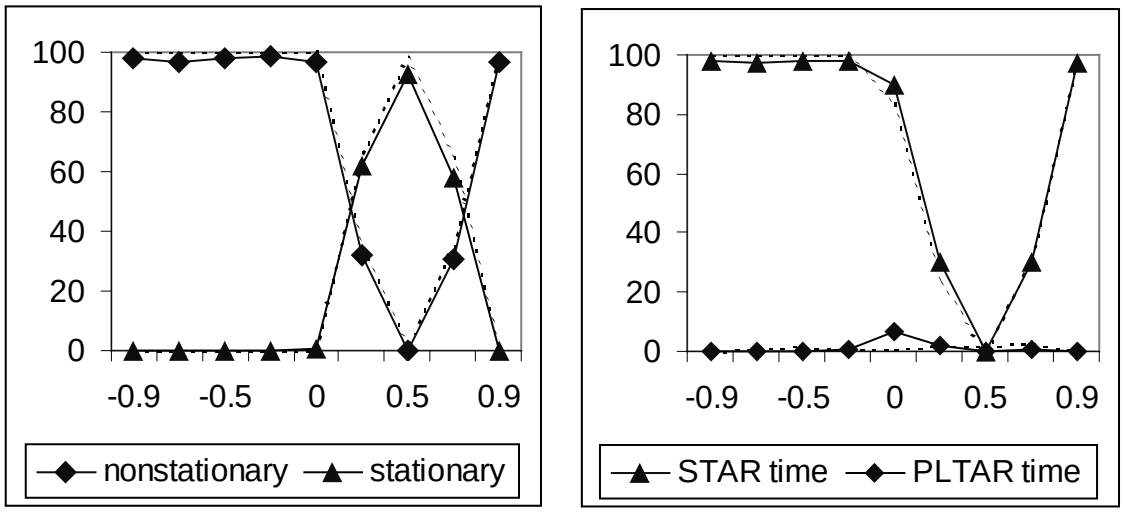

Figure 6: Frequency of selection of different models in series generated according to Model 3 (Time-varying). In the first regime $\phi_1 = 0.5$, ϕ_2 is shown in abscissa. Continuous line: results of the genetic algorithm application; dotted line: results according to optimal fitness

In Model 4 the series are generated according to a first order autoregressive process whose coefficient is smoothly varying both in levels and in time, results are shown in Figure 7 for the ten different coefficient choices detailed in table 2 above. For this series the linear stationary model is almost never chosen. The left panel shows the frequency of choice of nonstationary in time and nonlinear in levels models, that of time-varying linear models and that of nonlinear but stationary models. Continuous lines refer to genetic algorithm, dotted lines to the choice based on optimal fitness. As may be seen from table 2, series 5 is not time-varying (since the coefficient multplying the logistic in time, c_1, is zero) and series 7 and 8 are nearly so, and this is well reflected in the results. The nonlinearity in levels is stronger for series 6 to 10 than for series 1 to 4, and this implies a larger frequency of selecting the nonlinear models for those series. The right panel shows the frequency of selection of the correct model type (STAR-STAR) in contrast with type STAR in time and PLTAR in levels: this last choice is sometimes indicated by the genetic algorithm when nonlinearity is more evident, while the model selection based on optimal

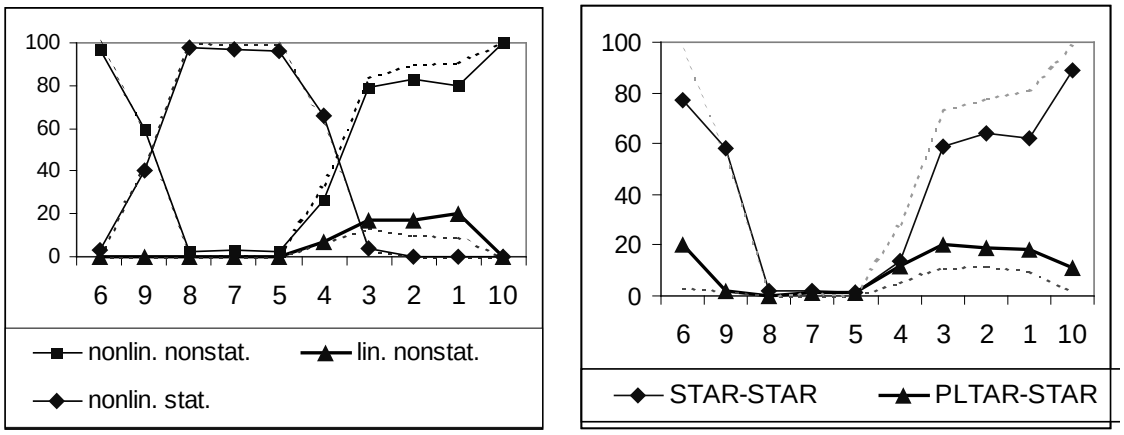

Figure 7: Frequency of selection of different models for the 10 series generated according to Model 4 (STAR in levels, STAR in time). Continuous line: results of the genetic algorithm application; dotted line: results according to optimal fitness

fitness is slightly less influenced by this drawback.

Let us consider now results for simulations according to Model 5. Since this is a self-exciting threshold model, both in levels and in time, the correct selection in our procedure is STAR-STAR with large estimated γ values. The results are displayed in Figure 8, and show that the genetic algorithm suggests the correct choice in most cases, with a small frequency (about five percent) of selection of the nonlinear stationary model, and a similar frequency of choice of the STAR in time and PLTAR in levels model. These results are nearly uniform for the nine different series, and are not far from those based on the optimal fitness.

Results for Model 6 are reported in Figure 9. The simulated model is STAR in levels and PLTAR in time, and the Figure shows the frequency of selection of the correct model, the frequency of selection of a nonlinear nonstationary model (difference between two frequencies being due therefore to confusion between STAR and PLTAR models), and the frequency of correct model choice based on optimal fitness. Here also the procedure based on genetic algorithm is satisfying since simultaneous nonlinearity and nonstationarity is detected in almost 95% cases, and the correct STAR-PLTAR model is chosen more than

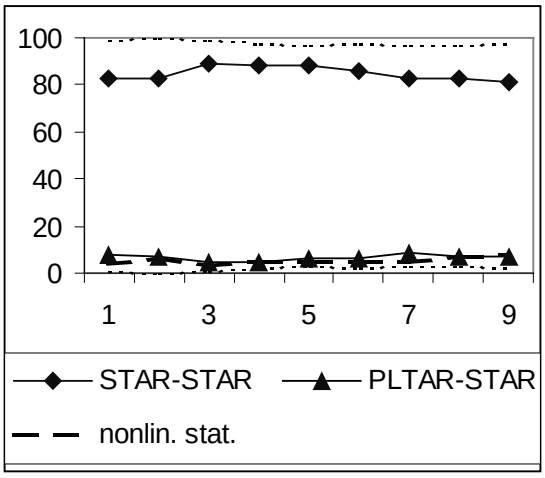

Figure 8: Frequency of selection of different models for the 9 series generated according to Model 5 (SETAR in levels, SETAR in time). Continuous line: results of the genetic algorithm application; dotted line: results according to optimal fitness

90% times, only exception being series 6, whose coefficients are the nearest to zero, where in 9 simulations out of 100 a linear stationary autoregressive model is suggested.

To conclude, we address briefly the quality of estimates of the remaining model parameters. An example of the distributions of estimated γ appears in Figure 10. The distributions are generally asymmetric with a single mode in the actual value, and are somewhat dispersed. This may be due to the large number of bits we employed for encoding the values of γ, which would require a considerably larger number of generations to obtain more precise results. However, it has been noted that an inaccuracy of the estimate of the smoothness parameter does not necessarily invalidates results (see Teräsvirta, 1994, 1998).

For threshold estimation, some empirical distributions are shown in Figure 11. These distributions appear symmetric, with the mode in the actual value. We tried also different values for the threshold, and conclude that the results do not vary much, provided that there is a sufficient number of observations in each regime.

The number of observations is indeed critical for good estimation results: to under-

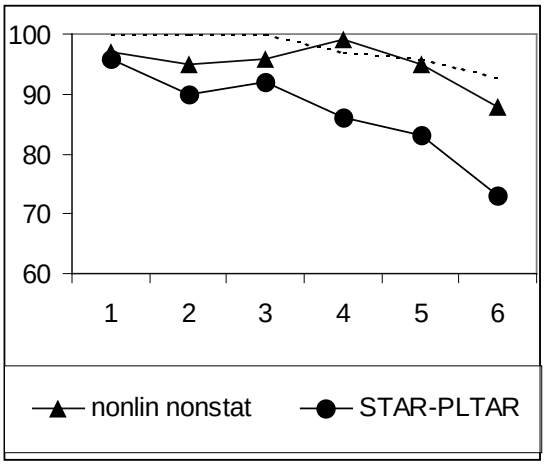

Figure 9: Frequency of selection of different models for the 6 series generated according to Model 6 (STAR in levels, PLTAR in time). Continuous line: results of the genetic algorithm application; dotted line: results according to optimal fitness

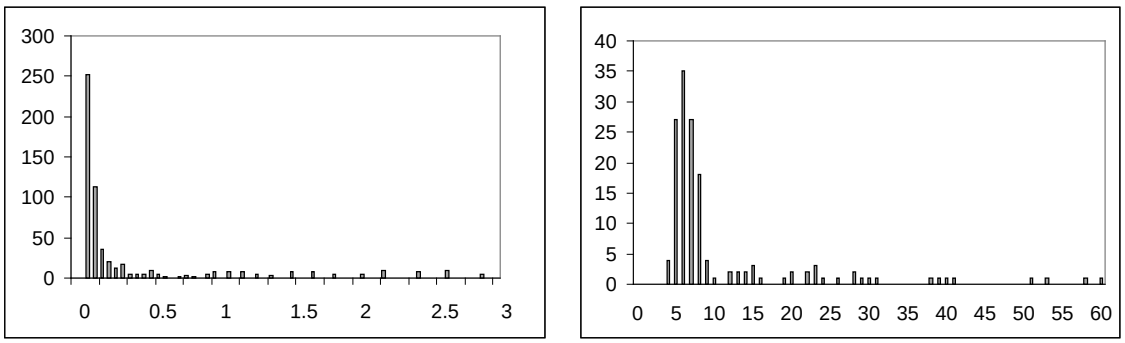

Figure 10: Distribution of the estimated γ parameter. Left panel: γ for time, estimates from Model 3, actual value is 0.1. Right panel: γ for levels, estimates from Model 4, actual value is 5

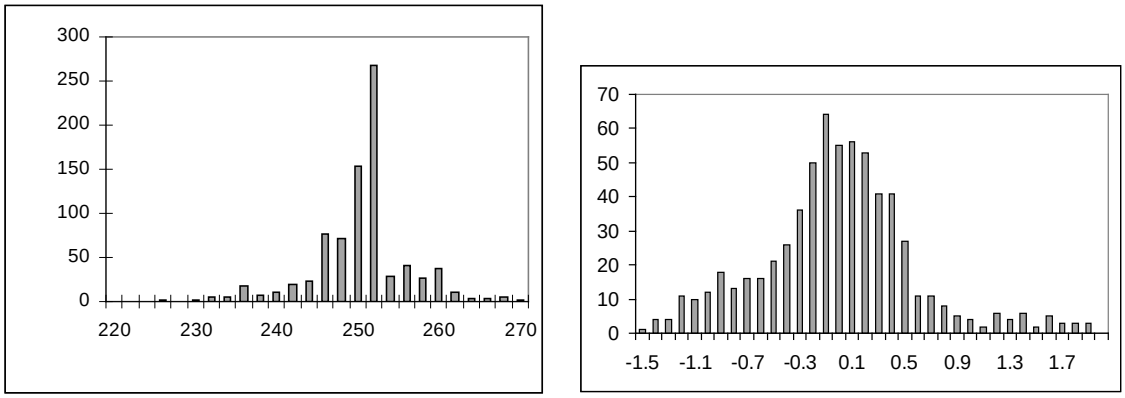

Figure 11: Distribution of the estimated thresholds. Left panel: threshold for time, estimates from Model 6, actual value is 250. Right panel: threshold for levels, estimates from Model 2, actual value is 0.0

stand this effect better, we have run the genetic algorithm procedure on series generated by the same model but on increasing the number of observations N (and maintaining the threshold in time at $N/2$), results for two specific series are displayed in Figure 12. The first series is generated according to Model 5 and is self-exciting threshold both in levels and in time, and the results are nearly stable as soon as N reaches 350. The second series is a STAR generated according to Model 2 with $\phi_1 = 0.5$ and $\phi_2 = 0.7$, its nonlinear behavior is less evident and it may be confused with a linear autoregressive process (see Figure 5). Here the frequency of correct detection increases uniformly with the series length N for all considered values, while the frequency of a linear stationary choice constantly decreases . We conclude, as expected, that the ability to distinguish between nearly similar models is greatly influenced by the bias and variability of the estimates of the autoregressive coefficients, which decrease as the series length increases.

5 APPLICATIONS TO REAL DATA

We apply our genetic algorithm procedure to some real data. To achieve more parsimonious representations, we extended it to incomplete models, devoting some genes of

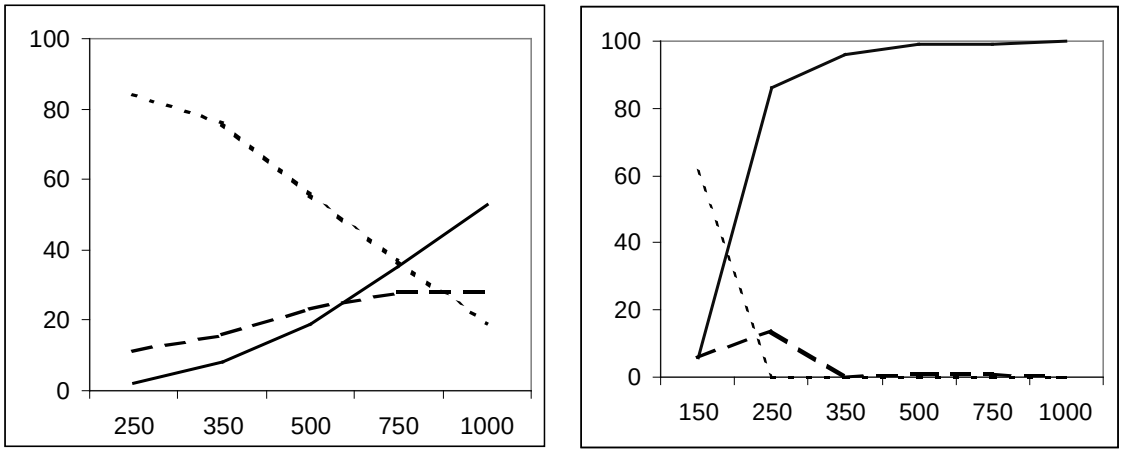

Figure 12: Selection frequencies of different models on series with increasing N, 100 replications. Left panel: simulations according to Model 2 (STAR), $\phi_1 = 0.5, \phi_2 = 0.7$. Right panel: simulations according to series 1 of model 5 (SETAR both in levels and in time). Continuous line: frequency of correct model selection; dotted line: frequency of linear stationary model selection; dashed line: frequency of selection of competitor models

the chromosome to select the lags at which autoregressive coefficients are present. The first series is the help-wanted advertising index, as analyzed by Lundberg et al. (2003), to which we refer for details on the data, shown in Figure 13, left panel. To compare with their results, we also chose as driving variable the twelfth differences. The selected model was a STAR in levels and STAR in time, with a behavior not so different from that proposed in Lundberg et al. (2003), but with the main difference that the threshold in time was found in April 1969 rather than in December 1979. Our model has a slightly smaller sum of squared residuals (0.646 instead of 0.691) and, containing 12 coefficients rather than 15, a considerably larger fitness. We tried also to build a model using directly the data as driving variable, and obtained a linear structure with STAR time-varying coefficients (with a slightly larger residual sum of squares 0.699, but only 10 coefficients): their values were relatively close to those of the previous model, and the threshold in time was exactly the same. It seems that this time splitting reflects better the two different behaviors that may be perceived in Figure 13 (left panel), the first one with an increasing

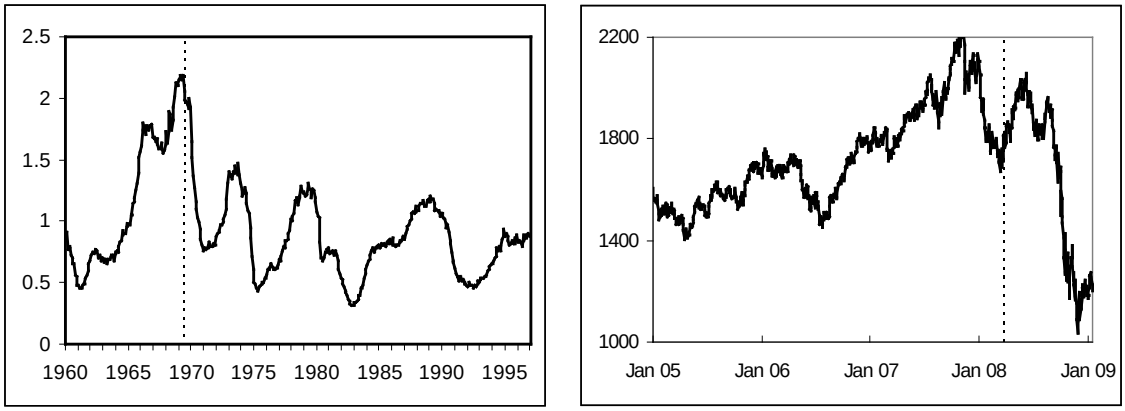

Figure 13: Left panel: Help-wanted advertising index monthly data 1990-96. Right Panel: Nasdaq-100 (∧NDX) index, daily 1995-2008 (dotted line: identified time threshold)

trend and the second one essentially cyclical.

We have also considered three stock exchange indexes, Nasdaq-100, Dow-Jones and Standard & Poor's 500. The proposed procedure was applied to the first differences of the daily close index data, years 2005-08, resulting for each series in STAR-STAR models with similar features. The estimated threshold in time is at the beginning of April, 2008, and the thresholds in levels are negative but small in absolute values. The most relevant differences between regimes are essentially in the estimated intercept coefficients , and the R^2 values of the models range, for the three indexes, from 0.12 to 0.135, measuring the advantage of the time-varying smooth transition explanation with respect to the random walk hypothesis. Though it may seem not so large, such an advantage increases when considering forecasts: the relative efficiency of lead-2 forecasts of the STAR-STAR model with respect to the random walk (as measured by the reciprocal of the mean square forecast errors ratio) was 2.37 for the entire data set, and 4.45 for the last six month of 2008. The corresponding figures for lead-3 forecasts were 1.97 and 3.71.

The STAR-STAR models residuals exhibit considerable heteroskedasticity, with a larger variability in the second time regime, and might be further analyzed by means of conditional heteroskedastic models. Figure 13 (right panel) shows the Nasdaq data

together with the estimated time threshold.

6 CONCLUSIONS

We have seen that the time-varying threshold models may account for nonlinear and non-stationary behavior of a time series, and how such complicate models may be successfully, and relatively easily, fitted using genetic algorithms.

Several extensions may be proposed: first, as we noted when analyzing stock indexes, the volatility may also be subject to different regimes, and double-threshold models have been introduced for describing levels and volatility simultaneously (Li and Li, 1996); an extension to time-varying structures could be interesting, though complicate, and therefore suitable to be addressed by means of a genetic algorithm. Second, an extensions to models with more than two regimes is also interesting and may be appropriate for repeated structural changes; the generalization of the proposed procedure to more regimes, though conceptually simple, requires a completely different chromosome architecture, and deserves further research.

REFERENCES

Alander, J. T. (1992), "On Optimal Population Size of Genetic Algorithms," in *Proc. CompEuro92*, IEEE Computer Society Press, pp. 65–70.

Amado, C. and Teräsvirta, T. (2008), "Modelling Conditional and Unconditional Heteroskedasticity with Smoothly Time-Varying Structure," SSE/EFI Working Paper in Economics and Finance 691, Stockholm School of Economics, Stockholm.

Bandyopadhyay, S. and Maulik, U. (2002), "Genetic Clustering for Automatic Evolution

of Clusters and Application to Image Classification," *Pattern Recognition*, 35, 1197–1208.

Baragona, R., Battaglia, F., and Cucina, D. (2004), "Fitting Piecewise Linear Threshold Autoregressive Models by Means of Genetic Algorithms," *Computational Statistics & Data Analysis*, 47, 277–295.

Bhansali, R. J. and Downham, D. Y. (1977), "Some Properties of the Order of an Autoregressive Model Selected by a Generalization of Akaike's EPF Criterion," *Biometrika*, 64, 547–551.

Chatterjee, S., Laudato, M., and Lynch, L. A. (1996), "Genetic Algorithms and Their Statistical Applications: an Introduction," *Computational Statistics & Data Analysis*, 22, 633–651.

Chen, C. W. S., Cherng, T.-H., and Wu, B. (2001), "On the Selection of Subset Bilinear Time Series Models: a Genetic Algorithm Approach," *Computational Statistics*, 16, 505–517.

Chen, R. and Tsay, R. S. (1993), "Functional Autoregressive Models," *Journal of the American Statistical Association*, 88, 298–308.

Crawford, K. D. and Wainwright, R. L. (1995), "Applying Genetic Algorithms to Outlier Detection," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, ed. Eshelman, L. J., San Mateo, CA: Morgan Kaufmann, pp. 546–550.

Davis, R., Lee, T., and Rodriguez-Yam, G. (2006), "Structural Break Estimation for Nonstationary Time Series Models," *Journal of the American Statistical Association*, 101, 223–239.

Gaetan, C. (2000), "Subset ARMA Model Identification Using Genetic Algorithms," *Journal of Time Series Analysis*, 21, 559–570.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley.

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.

Kapetanios, G., Shin, Y., and Snell, A. (2003), "Testing for a Unit Root in the Nonlinear STAR Framework," *Journal of Econometrics*, 112, 359–379.

Li, C. W. and Li, W. K. (1996), "On a Double-Threshold Autoregressive Heteroscedastic Time Series Model," *Journal of Applied Econometrics*, 11, 253–274.

Lin, C. and Teräsvirta, T. (1994), "Testing the Constancy of Regression Parameters Against Continuous Structural Change," *Journal of Econometrics*, 62, 211–228.

Lundberg, S., Teräsvirta, T., and van Dijk, D. (2003), "Time-Varying Smooth Transition Autoregressive Models," *Journal of Business and Economic Statistics*, 21, 104–121.

McAleer, M. and Medeiros, M. C. (2008), "A Multiple Regime Smooth Transition Heterogeneous Autoregressive Model for Long Memory and Asymmetries," *Journal of Econometrics*, 147, 104–119.

Priestley, M. B. (1988), *Non-linear and Non-stationary Time Series Analysis*, London: Academic Press.

Reeves, C. R. (1993), *Modern Heuristic Techniques for Combinatorial Problems*, New York: Wiley.

Reeves, C. R. and Rowe, J. E. (2003), *Genetic Algorithms - Principles and Perspective: a Guide to GA Theory*, London: Kluwer.

Rudolph, G. (1997), *Convergence Properties of Evolutionary Algorithms*, Hamburg: Verlag Dr. Kovač.

Syswerda, G. (1989), "Uniform Crossover in Genetic Algorithms," in *Proceedings 3rd International Conference on Genetic Algorithms*, ed. Schaffer, J. D., Los Altos: Morgan Kaufmann, pp. 2–9.

Teräsvirta, T. (1994), "Specification, Estimation and Evaluation of Smooth Transition Autoregressive Models," *Journal of the American Statistical Association*, 89, 208–218.

— (1998), "Modeling Economic Relationships with Smooth Transition Regressions," in *Handbook of Applied Economic Statistics*, eds. Ullah, A. and Giles, D. E. A., New York: Marcel Dekker, pp. 507–552.

Tong, H. (1990), *Non Linear Time Series: A Dynamical System Approach*, Oxford: Oxford University Press.

Wu, B. and Chang, C.-L. (2002), "Using Genetic Algorithms to Parameters (d,r) Estimation for Threshold Autoregressive Models," *Computational Statistics & Data Analysis*, 38, 315–330.