ROBERTO BARAGONA – FRANCESCO BATTAGLIA
DOMENICO CUCINA

Estimating threshold subset autoregressive moving-average models by genetic algorithms

Summary - A genetic algorithm is proposed to estimate the parameters of a self-exciting threshold subset autoregressive moving-average model. The threshold model is composed of several linear autoregressive moving-average models. Each one of these models applies according to a “switch mechanism” that is based on the comparison between the delayed observation and some “threshold” values. Our procedure incorporates the identification in each “regime” of a “subset” model. Subset models are useful as they allow the number of parameters to be reduced so that only those really needed are included in the model. The proposed procedure is used for modeling the well-known Canadian lynx data.

Key Words - Time series; Self-exciting threshold subset autoregressive moving-average models; Delay parameter; AIC criterion; Genetic algorithms; Canadian lynx data.

1. INTRODUCTION

The estimation of parameters of self-exciting threshold autoregressive moving-average (SETARMA) model (see Tong (1990)) requires that simultaneously the delay parameter, the number of regimes, the threshold parameters, and the autoregressive (AR) and moving-average (MA) orders and lags be determined according to some optimization criterion. This is a combinatoric-like problem where the solution has to be searched in a discrete large space. Only if the problem size is smallest all possible models may be tried. Then, parameters may be estimated for each one, and the best model selected according to a given criterion, usually derived from some approximation to the Kullback-Leibler information. For problems of moderate size, such complete enumeration and evaluation is practically infeasible.

The characteristic features of the SETARMA identification problem suggest that a local search technique will be useful to efficiently examine the space of
the solutions and select the combination of parameters that corresponds to the best model, for the given criterion. A popular stochastic search technique uses the optimization heuristic procedure derived from the genetic algorithms (GA), early introduced by Holland (1975) to describe, by means of a computer model, the evolution of a biological population towards the adaptation to the environment. This concept has been applied to “artificial” populations composed of potential solutions of an optimization problem as well (see, for instance, De Jong (1975), Goldberg (1989), Whitley (1994).) Properties of the GAs, viewed as function optimizers, were studied by Rudolph (1994), Jennison and Sheehan (1995), Vose (1999), amongst others. In addition, applications in many fields were envisaged and particular procedures proposed, for instance digital filter design, pattern recognition systems, planning and scheduling problems, and many others (see Haupt and Haupt (1998) and Man, Tang and Kwong (1999).) Applications in statistics were discussed by Chatterjee, Laudato and Lynch (1996) and Chatterjee and Laudato (1997). Besides, in time series analysis applications focused on outlier identification (Baragona, Battaglia and Calzini (2001a),) cluster analysis of time series (Baragona, Battaglia and Calzini (2001b),) subset VAR modeling and detecting influential observations in vector time series (Bozdogan and Bearse (2003),) selection of subset bilinear time series models (Chen, Cherng and Wu (2001),) subset autoregressive moving-average (ARMA) models identification (Gaetan (2000), Minerva and Poli (2001)) and self-exciting autoregressive (SETAR) models identification (Wu and Chang (2002).)

In this paper we bring together the results from these three latter papers and generalize and widen their application to cover the SETARMA model identification problem when subset ARMA models are to be specified for each regime and the size of the problem is rather large. Usually, in the SETAR framework much attention is devoted to the choice of delay parameter, threshold parameters and AR orders, and, for any linear model in each regime, all parameters up to the order for that regime are estimated. Then, further refinement is needed for excluding from each model unnecessary and/or noisy parameters corresponding to certain lags (Thanoon (1990).) In a Bayesian framework, So and Chen (2003) proposed simulating the unknown parameters and identifying the best subset SETAR model simultaneously by adopting the Markov chain Monte Carlo techniques. Many examples of SETAR model building in practice may be found in Tong (1990, Chapter 7.) A common method for finding the appropriate threshold parameters consists essentially in the investigation of the percentiles of the time series empirical distribution. On the other hand, the identification of the subset AR models was extensively studied. An algorithm to evaluate efficiently all possible models has been developed by Haggan and Oyetunji (1984). However, even this method may hardly cope with the identification task of a subset AR unless the maximum allowed lag is taken small enough. Efficient algorithms that do not consider all possible models were
Estimating threshold subset autoregressive moving-average models by genetic algorithms

proposed by Yu and Lin (1991), Zhang and Terrel (1997), and Chen (1999), for instance. These procedures are anyway requiring an increasingly large computation time for even small increase of the problem size, so that the proposals concerned with procedures based on the GAs seem well motivated.

The paper is organized as follows. In Section 2 the subset SETARMA model is introduced. A simplified version of the Hannan and Rissanen (1982) “innovation regression” method is adopted for ARMA parameter estimation. In Section 3 the identification procedure based on the GAs is presented. In Section 4 the performance of the procedure for modeling some artificial time series is investigated. Application to the well-known Canadian lynx data is discussed in Section 5. In Section 6 conclusions are drawn.

2. The subset SETARMA model

Let \( k-1 \) finite real values \( \{ r_1 < \ldots < r_{k-1} \} \) be given (the “threshold parameters,”) and let \( r_0 = -\infty \) and \( r_k = +\infty \). The \( k \) disjoint intervals \( R_i = (r_{i-1}, r_i], \) \( i = 1, \ldots, k, \) partition the real axis \( \mathbb{R} \). Let an integer \( d \) (the “delay parameter”) be also given in an interval \( (1, D) \). For the real valued time series \( \{ y_t, \text{ integer } t \} \) each interval \( R_i \) defines a “regime,” in the sense that, for any \( t \), the time series value \( y_t \) is said to follow the regime \( i \) if \( y_{t-d} \in R_i \). Let \( p \) and \( q \) denote the maximum number of AR and MA parameters respectively. Then, let us assume that in each regime two positive integers \( p^{(i)} \) and \( q^{(i)} \) be specified, \( p^{(i)} \leq p \leq P \) and \( q^{(i)} \leq q \leq Q \), any \( i \), must hold. Then a subset SETARMA model takes the form

\[
y_t = c^{(i)} + \sum_{u=1}^{p^{(i)}} \phi_{j_u^{(i)}}^{(i)} y_{t-j_u^{(i)}} + \sum_{v=1}^{q^{(i)}} \theta_{h_v^{(i)}}^{(i)} e_{t-h_v^{(i)}} + e_t, \tag{1}
\]

where the regime \( i, i = 1, \ldots, k, \) depends on the delayed output \( y_{t-d} \), and \( \{ e_t \} \) is a sequence of independently identically distributed random variables with mean zero and variance \( \sigma^2_e \). Either sum in (1) vanishes if \( p^{(i)} = 0 \) or \( q^{(i)} = 0 \).

The delay parameter \( d \), the number of regimes \( k \), the threshold parameters \( r_i \) and the AR and MA lags in each regime are called the “structural parameters.” Once such parameters are determined, the coefficients \( c^{(i)}, \phi_{j_u^{(i)}}^{(i)} \) and \( \theta_{h_v^{(i)}}^{(i)} \) may be estimated by using either a non linear least-squares algorithm or, under Gaussianity assumption, the maximum likelihood method. However, for ARMA
models, these methods are expensive, while we are compelled to resort to some fast algorithm for the GA to perform efficiently in a reasonable time. Following Gaetan (2000), we use the innovation regression method proposed by Hannan and Rissanen (1982) with some slight modifications.

Let \( n \) observations \( \{y_1, \ldots, y_n\} \) be available. By assuming that the structural parameters are known, the steps of the estimation procedure are outlined below.

1) Fit a “long” autoregression to the time series data for each regime \( i, i = 1, \ldots, k \). This gives the estimated innovations \( \hat{e}_t \). In practice, we choose the AR order, in a given interval \((1, P_{\text{max}})\), which minimizes the AIC criterion (Akaike (1973), Akaike (1977)). Other criteria may be used, for instance the Akaike’s BIC criterion (Akaike (1979)), the Schwarz’s BIC criterion (Schwarz (1978)) and the AICC (Wong and Li (1998)). However, in the present context, better results were not gained by resorting to criteria other than the Akaike’s AIC.

2) Estimate the ARMA parameters in (1) by least squares after the innovations \( \{e_{t-h_1}, \ldots, e_{t-h_q}\} \) are substituted by their estimates \( \{\hat{e}_{t-h_1}, \ldots, \hat{e}_{t-h_q}\} \) computed in the previous step.

3) Compute the NAIC criterion described by Tong (1990, p. 379)

\[
\text{NAIC}(x^*) = \left\{ \sum_{i=1}^{k} n_i \log(S_i/n_i) + 2 \sum_{i=1}^{k} (p^{(i)} + q^{(i)} + 1) \right\} / (\text{Effective sample size}), \tag{2}
\]

where \( x^* \) in (2) denotes the vector of the structural parameters

\[
x^* = \left( d; r_1, \ldots, r_{k-1}; \left\{ p^{(i)}; j_1^{(i)}, \ldots, j_{p(i)}^{(i)}; q^{(i)}; h_1^{(i)}, \ldots, h_q^{(i)} \right\} \right), \tag{3}
\]

\( n_i \) is the number of observations that belong to the regime \( i \) and \( S_i \) is the residual sum of squares for the \( i \)-th subset ARMA model estimated in the previous step. The criterion (2) modifies the Akaike’s AIC criterion so that it can be applied to threshold models.

Steps 1), 2) and 3) are repeated for each \( d \in (1, D) \), where \( D \) is a pre-specified integer.

Computations are to be done for \( t \geq n_0 \), where the integer \( n_0 \) is chosen \( n_0 \geq P_{\text{max}} + \max(P, Q) \). A minimum number of observations \( m \) is required in each regime. The number \( m \) has to be chosen so as to ensure that in all linear regressions the number of parameters is less than the number of normal equations.
3. The Ga procedure

It is clear from the discussion in the previous Section that we hope that the GA will be able to provide us with the structural parameters of the subset SETARMA model (1) but the delay parameter $d$. In addition, the function to be optimized is the NAIC criterion as defined in (2). Note that, at this stage, we view the criterion as a function of the structural parameters only. In practice we are establishing a mapping between the criterion values and a mixed integer and real vector $x^*$, this latter being the set of the structural parameters chosen by the searching procedure. This explains well our need for heuristic search methods, as the function to be optimized has no continuous argument, and we cannot base, not even approximately, on properties such as continuity, differentiability and convexity. In what follows, our GA procedure will be outlined in some detail.

3.1. Encoding

Any vector $x^*$ as defined in (3) represents a tentative solution to the problem of specifying the structural parameters that lead to the optimal subset SETARMA model for the data set. The vector $x^*$ may be encoded as a binary string (the “chromosome”) $x$. It is apparent that the chromosome $x$ is composed of “fragments” each one with its own meaning. According to the GA practice, the binary encoding is preferred because the GA theory was developed mainly by assuming binary chromosomes (see Holland (1975) and Jennison and Sheehan (1995).) In addition, the binary encoding simplifies the “genetic operators.” Note that in the chromosome $x$ there are only the specifications about the ARMA coefficients, as a constant in each regime is always included in our SETARMA model. Moreover, we do not encode the delay parameter $d$ in the chromosome $x$ because all admissible values are considered exhaustively.

The first fragment of the chromosome is concerned with the threshold parameters. These latter are searched amongst the time series values. This does not imply any loss of generality since the inequality $y_{t-d} \leq r_i$, say, may well be replaced by $y_{t-d} \leq y_{tj}$, where $y_{tj}$ is the greatest observation less than or equal to $r_i$. Such simple device avoids the need for binary representation of real values. Though this is certainly feasible (see, for instance, Chatterjee, Laudato and Lynch (1996),) it requires a number of bits rather large if we want a good approximation. Let us consider the observations from 1 to $n$ arranged in ascending order, and let $\{y_{r_1} \ldots , y_{r_{n-2m}}\}$ denote the observations from the place $m+1$ to the place $n-m$. Then, we define a binary string with length $n-2m$, where each bit corresponds to a time series observation according to the sequence as previously defined. An observation is assumed as a threshold parameter if the bit is 1, while the bit is zero otherwise. Thus
$y_{ij}$ is a threshold if $x_j = 1$. The number of 1’s in this fragment determines the number of regimes as $k = x_1 + \ldots + x_{n-2m} + 1$. The remaining fragments encode the identification parameters of $k$ subset ARMA models. Let us choose $\nu$ the number of bits for the binary representation of integer values $p^{(i)}$ and $q^{(i)}$, any $i$. We use the usual binary form, so that the integers we may encode vary from 0 to $2^\nu - 1$. Let us choose $\mu$ the number of bits for the lag values binary representation, $\mu \geq \nu$. The lag values are constrained in the interval $(1, 2^\mu - 1)$. The length $\ell$ of the chromosome $x = \{x_1, \ldots, x_\ell\}$ may be easily computed as

$$\ell = (n - 2m) + 2k\nu + \mu \left\{ \sum_{i=1}^{k} (p^{(i)} + q^{(i)}) \right\}.$$ 

It is apparent that the chromosome $x$ has variable length, and its length depends both on the number of regimes $k$ and on the number $p^{(i)} + q^{(i)}$ of “active” ARMA lags. In addition, the “fragments” that compose the chromosome $x$ are not independent. For example, the number of 1’s in the fragment that includes the bits from 1 to $n - 2m$ determines how many subset ARMA models are to be encoded in the remaining bits. In turn, the number of lag values is controlled by the first $\nu$ bits of either AR or MA parts. So, the chromosome has a “hierarchical structure” (see Man, Tang and Kwong (1999, Chapter 4).) The number of regimes $k$ occupies the first place in the hierarchy. The number of ARMA models depends obviously on the number of regimes. The second hierarchy level includes the number $p^{(i)}$ of coefficients AR and the number $q^{(i)}$ of coefficients MA in each model, $i = 1, \ldots, k$. At the lowest level there are the “active” lags that correspond to the model coefficients, that is the AR lags $j_{1}^{(i)}, \ldots, j_{p^{(i)}}^{(i)}$ and the MA lags $h_{1}^{(i)}, \ldots, h_{q^{(i)}}^{(i)}$, $i = 1, \ldots, k$. This hierarchical structure has to be taken into account for the design of the crossover and mutation operators. The delay parameter $d$, however, is not included in such hierarchical structure, as all allowable values from 1 to $D$ are tried.

Some comments are worth noting. Upper values have to be specified for most of the structural parameters. The delay parameter $d$ may vary from 1 to $D$. A maximum number of regimes is specified, $K$ say. Upper bounds for the number of AR lags, $p$, say, and MA lags, $q$, say, must be specified as well. Obviously, they cannot exceed $2^\nu - 1$. Also, values $P$ and $Q$, say, must be assumed as maximum AR lag and MA lag respectively. They have to be greater than zero and cannot exceed $2^\mu - 1$. In addition, an upper bound $P_{\text{max}}$ has to be pre-specified as well for the order of the “auxiliary” autoregression. The largest chromosome length is

$$\max \ \ell = (n - 2m) + 2K\nu + (p + q)K\mu.$$
For example, \( d \) given, a tentative solution may be the proposal subset SETARMA model

\[
y_t = \begin{cases} 
    c^{(1)} + \phi_1^{(1)} y_{t-1} + \theta_2^{(1)} e_{t-2} + e_t & \text{if } y_{t-d} \leq y_{t0} \\
    c^{(2)} + \phi_3^{(2)} y_{t-3} + \theta_1^{(2)} e_{t-1} + e_t & \text{if } y_{t-d} > y_{t0}.
\end{cases}
\]

If \( n = 50, m = 15, \nu = 2 \) and \( \mu = 3 \), then such tentative solution will be encoded as the following binary string \( x \)

\[
0000000001000000000001010010100101101101
\]

where the symbols \( \| \) and \( | \) are inserted to highlight the “fragments” but they are not really included in the chromosome. We assign 1 to the bit that corresponds to the observation that defines the threshold parameter. Recall that we consider the observations in ascending order, but we include in the chromosome only those which are between the 16-th (index \( t_1 \)) and the 35-th place (index \( t_{20} \)).

The first subset ARMA model follows, which corresponds to the first regime. The number of AR parameters is 1, with order 1, and the number of MA parameters is 1 as well, but its order is 2. Last, the second regime model is encoded. There is a single AR parameter (order 3) and a single MA parameter (order 1.)

### 3.2. Initial population and iterative cycle

Each GA procedure is repeated for given \( d, d = 1, \ldots, D \). Then, any tentative solution is represented by a binary string of length \( \ell \). A pre-specified number of admissible strings, \( s \) say, are generated at random, and form the initial set of tentative solutions. The set of initial solutions is manipulated by means of the so-called “evolutionary operators.” Many have been proposed, but, according to the simple GA procedure, we consider only selection, crossover and mutation. Each one of these three operators accomplish its own task. Selection makes the best fit individuals spread in the next generations. Crossover combines promising solutions, as they come from the selection step, to put together blocks that are themselves parts of “high quality” solutions. Mutation maintains diversity in the population, and possibly recovers bits that would be impossible to create by means of the other two operators. These three operators modify the solutions and produce a new set that is designed to possess greater average fitness function than the initial one. The procedure is iterative, and in each step a new set of solutions is generated from the previous one. Any set of solutions is called “population,” and the iterative procedure mimics the evolution of the population through a sequence of generations in such a way that more and more tentative solutions (the “individuals” in the populations) approach the global optimum. Details about the effectiveness of the GA as
optimization tool may be found, for instance, in Rudolph (1994) and Jennison and Sheehan (1995). In each iteration the new generation replaces the past one completely. This device is often referred to as taking the “generational gap” equal to unity. The iterative procedure stops as soon as the maximum pre-specified number of generations, $N$ say, is attained, or some stopping criterion is met.

3.3. Selection

At any stage of the iterative GA procedure, given the time series $y$, we may compute the NAIC criterion (2) for each given $x$. The evaluation of a string within the GA framework, however, is done by means of a positive real-valued function called “fitness function.” As greater is the fitness function of a string, as closer to the optimal solution this latter has to be considered. Since (2) may be negative, let us define our fitness function

$$f(x) = \exp(-\text{NAIC}(x)/C),$$

where $C$ is a problem-dependent constant which is introduced to prevent the occurrence of overflow in the computation and to scale the fitness function suitably.

The selection is performed by means of the “roulette wheel rule.” Let the population consist of the binary strings $\{x^{(1)}, x^{(2)}, \ldots, x^{(s)}\}$. Then, by using (4), we may compute

$$F = \sum_{i=1}^{s} f(x^{(i)}).$$

We select $s$ times a chromosome from the population. Such sampling procedure is done with replacement, and each chromosome $i$ in the population has probability $f(x^{(i)})/F$ to be chosen. We obtain a new population, sometimes called “new generation,” that replaces the initial population. In other words, for $s$ times a string is drawn from the current population with probability proportional to its fitness. The selected strings replace the population. In such new population we may find a number of copies of some chromosomes while others are missing. If the largest fitness function that may be computed from the chromosomes in the new population is less than that computed for the past population, we adopt the “elitist strategy” (see Rudolph (1994), Rudolph (1997) and Jennison and Sheehan (1995) for motivation.) The chromosome for which the best fitness was computed in the past population is recovered and replaces the chromosome that, in the new population, has the smallest fitness function value.
3.4. Crossover

The probability of crossover $p_c$ has to be pre-specified. It means that, in each generation, $\lfloor p_c n/2 \rfloor$ pairs are selected at random and the crossover operator is applied to each of them. An individual may be chosen more than once to enter a pair and its chromosome is updated each time. Let $x^{(1)}$ and $x^{(2)}$ denote the two individuals in the pair. The bits from the location 1 to $n - 2m$ are examined and the locations where the first chromosome, or the second, or both, have bit value equal to one are recorded. If no ones are found, then crossover does not take place. Otherwise, let $\{i_1, \ldots, i_\gamma\}$ denote such locations. We select in this set a single location at random, and let us denote $i_\alpha$ such location. Then, two cases are to be considered:

1. Either $x_{i_\alpha}^{(1)} = 1$ or $x_{i_\alpha}^{(2)} = 1$. If, for instance, $x_{i_\alpha}^{(1)} = 0$ and $x_{i_\alpha}^{(2)} = 1$, then we set $x_{i_\alpha}^{(1)} = 1$ and $x_{i_\alpha}^{(2)} = 0$. This means that the number of regimes encoded in the second chromosome decreases by one, whilst a new regime adds to the number of regimes encoded in the first chromosome. Let us denote $i_\beta$ the first location such that $x_{i_\beta}^{(1)} = 1$ and $i_\alpha < i_\beta$. The model specification encoded in the first chromosome changes, in the sense that the observations that followed the regime $i_\beta$ (defined by $y_t \leq y_{i_\beta}$) now follow two different regimes, that is $i_\alpha$ ($y_t \leq y_{i_\alpha}$) and $i_\beta$ ($y_{i_\alpha} < y_t \leq y_{i_\beta}$).

The model corresponding, in the second chromosome, to the regime $i_\alpha$ is transferred to the first chromosome in front of the model corresponding to the regime $i_\beta$. The model encoded in the second chromosome changes as well, but the observations that followed the regime $i_\alpha$ now follow the same regime as the observations that “belong” to the regime $i_\delta$, where $i_\delta$ is the first location such that $x_{i_\delta}^{(2)} = 1$ and $i_\alpha < i_\delta$.

2. Both $x_{i_\alpha}^{(1)} = 1$ and $x_{i_\alpha}^{(2)} = 1$. In this case, the model encoded in the second chromosome for the regime $i_\alpha$ exchanges with that encoded in the first chromosome for the same regime $i_\alpha$.

Note that the last regime never exchanges between chromosomes. Furthermore, the crossover may produce only a single change of regime in each of the two chromosome, either adding one, or deleting one, or exchanging models. A model is treated as a block, and crossover does not exchange “fragments” of models between chromosomes. Other implementations have been tried, but the one that we propose has been found the most appropriate, as it does not disrupt the chromosome structure too heavily. The main guidelines about this choice have been that offsprings from crossover should not have to differ completely from their “parents chromosomes.”
For instance, let the two individuals in the pair have chromosomes that encode the proposal subset SETARMA model

\[ y_t = \begin{cases} 
  c^{(1)} + \phi^{(1)}_1 y_{t-1} + \theta^{(1)}_2 e_{t-2} + e_t & \text{if } y_{t-d} \leq y_{i10} \\
  c^{(2)} + \phi^{(2)}_3 y_{t-3} + \theta^{(2)}_1 e_{t-1} + e_t & \text{if } y_{t-d} > y_{i10}
\end{cases} \]

and

\[ y_t = \begin{cases} 
  c^{(1)} + \phi^{(1)}_1 y_{t-1} + \theta^{(1)}_1 e_{t-1} + e_t & \text{if } y_{t-d} \leq y_{i15} \\
  c^{(2)} + \phi^{(2)}_3 y_{t-3} + e_t & \text{if } y_{i15} < y_{t-d} \leq y_{i21} \\
  c^{(3)} + \phi^{(3)}_1 y_{t-1} + \theta^{(3)}_1 e_{t-1} + e_t & \text{if } y_{t-d} > y_{i21}
\end{cases} \]

respectively. Then, if we assume \( n = 50, m = 10, \nu = 2 \) and \( \mu = 3 \) these two tentative solutions will be encoded as the following binary strings \( x^{(1)} \) and \( x^{(2)} \):

\[
\begin{align*}
000000000100000000000000000000 & \parallel 01|001|01|010|01|011|01|001 \\
000100000000000000000000000000 & \parallel 01|001|01|001|01|011|00|01|001|01|001
\end{align*}
\]

The bits in places 5, 10 and 21 correspond to candidate locations for crossover. Let the random choice yield the bit 21. Then, the result from the crossover operator will be the two new chromosomes

\[
\begin{align*}
000000000100000000000000000000 & \parallel 01|001|01|010|01|011|00|01|011|01|001 \\
000100000000000000000000000000 & \parallel 01|001|01|001|01|011|00|01|001|01|001
\end{align*}
\]

that is the new proposal subset SETARMA models

\[ y_t = \begin{cases} 
  c^{(1)} + \phi^{(1)}_1 y_{t-1} + \theta^{(1)}_2 e_{t-2} + e_t & \text{if } y_{t-d} \leq y_{i10} \\
  c^{(2)} + \phi^{(2)}_3 y_{t-3} + e_t & \text{if } y_{i10} < y_{t-d} \leq y_{i21} \\
  c^{(3)} + \phi^{(3)}_1 y_{t-1} + \theta^{(3)}_1 e_{t-1} + e_t & \text{if } y_{t-d} > y_{i21}
\end{cases} \]

and

\[ y_t = \begin{cases} 
  c^{(1)} + \phi^{(1)}_1 y_{t-1} + \theta^{(1)}_1 e_{t-1} + e_t & \text{if } y_{t-d} \leq y_{i15} \\
  c^{(2)} + \phi^{(2)}_1 y_{t-1} + \theta^{(2)}_1 e_{t-1} + e_t & \text{if } y_{t-d} > y_{i15}
\end{cases} \]

3.5. Mutation

Any bit of any string is allowed to flip with probability \( p_m \), usually quite small. The hierarchical structure of our chromosomes, however, implies that mutation in some fragment may impose that fragments at lower levels change as well. We proceed by illustrating our implementation of the mutation operator, separately for each fragment of the chromosome.
Threshold parameters

Any bit from 1 to \( n - 2m \) may flip with probability \( p_m \). Obviously, for each bit, there are only two possible cases, that is either the bit changes from 1 to 0 or the bit changes from 0 to 1. Let \( \alpha, 1 \leq \alpha \leq n - 2m \), denote the location where a mutation occurs. Two distinct courses of action have to be adopted in the two cases:

1. The bit value in \( \alpha \) is one. After the mutation, \( x_\alpha = 0 \) and a regime is deleted. The corresponding model is canceled, and the observations that followed the regime \( \alpha \) have to follow the next regime.
2. The bit value in \( \alpha \) is zero. After the mutation, \( x_\alpha = 1 \) and a new regime is created. A model is generated at random for such new regime, and some observations now are to be modeled accordingly. The new model is encoded in the chromosome.

Number of AR parameters (for each regime)

Any of the \( \nu \) bits that encode in binary form the number of AR coefficients may mutate. As a result, the number of AR parameters \( p \) may turn into \( p^* \). Then, the following three cases may occur:

1. \( p = p^* \). The chromosome is left unchanged.
2. \( p > p^* \). In this case, \( p - p^* \) lags are deleted at random in the AR part of the model.
3. \( p < p^* \). In this case, \( p^* - p \) lags are generated at random in the interval \([1, P]\). If some generated lags equal any existing lag, then new lags are generated at random until the newly generated lags all differ from the \( p \) existing ones.

The new number \( p^* \) of AR coefficients replaces the previous one, \( p \), and it is encoded in the chromosome.

AR lags (for each regime)

Let \( p^{(i)} \) be the number of AR parameters, that is the number of AR lags. Any binary string that represents the encoded lags, \( j_{1}^{(i)}, \ldots, j_{p^{(i)}}^{(i)} \), may mutate to produce \( p^{(i)} \) new lags. However, the chromosome fragment has to be “legalized,” in the sense that all zero lags have to be canceled, and, if two or more lags happen to be equal, only one has to be retained. This means that the number of AR coefficients may possibly decrease and it has to be adjusted according to the result from the mutation of the AR lags. The new number of AR parameters is encoded along with the mutated AR lags.

Number of MA parameters (for each regime)

The procedure is identical as for the number of AR parameters, \( q, q^* \) substituting \( p, p^* \) and \( Q \) substituting \( P \).
**MA lags (for each regime)**

The same procedure as for the AR lags applies.

The mutation of the AR number of coefficients and lags, and MA number of coefficients and lags, is to be repeated for each regime, that is for $i = 1, \ldots, k$.

For example, under the assumptions $n = 50$, $m = 15$, $\nu = 2$ and $\mu = 3$, let the following chromosome

\[000000000100000000000000000010101000101\]

encode the subset SETARMA model

\[
y_t = \begin{cases} 
  c^{(1)} + \phi_1^{(1)} y_{t-1} + \theta_2^{(1)} e_{t-2} + e_t & \text{if } y_{t-d} \leq y_{t_{10}} \\
  c^{(2)} + \phi_3^{(2)} y_{t-3} + \theta_1^{(2)} e_{t-1} + e_t & \text{if } y_{t-d} > y_{t_{10}}. 
\end{cases}
\]

The chromosome is scanned and, for each bit, a uniform random number in the interval $(0, 1)$, $u$, say, is generated. If $u \leq p_m$, then the bit flips, otherwise it remains unchanged. Let the bit at the location 8, say, mutate. Then, a new regime has to be added to the model. The threshold parameter is assumed $y_{t_8}$, that is $r_1$ is equal to the observation that occupies the 8-th place in the ascending orders of the time series observations from $t = m + 1$ to $t = n - 2m$. An ARMA model is generated at random for this new regime. Let such model be an ARMA(3,2) with its second AR coefficient missing. This model is encoded in front of the two models already encoded in the chromosome. Let the next mutation occur at location 10. In this case, a regime is deleted along with the corresponding ARMA(1,2) model. Let mutation involve now the second level in the chromosome fragments hierarchy, that is the number of AR and MA parameters. Let the bit at location 37 in the present chromosome mutate (i.e. the tenth bit from the right.) The AR number of coefficients turns into 3, and let lags 1 and 2 be added. If there are no further mutations as far as the number of AR and MA parameters are concerned, it remains to take mutation at the lowest level, the lag values, into account. Let the second bit from the right mutate. This implies that the first lag of the MA part of the last model turns into 3. Summing up, after mutation the chromosome is

\[000000010000000000000000001010101100100010|0010011|1001010|11001010011|01|011\]

which encodes the subset SETARMA model

\[
y_t = \begin{cases} 
  c^{(1)} + \phi_1^{(1)} y_{t-1} + \phi_3^{(1)} y_{t-3} + \theta_1^{(1)} e_{t-1} + \theta_2^{(1)} e_{t-2} + e_t & \text{if } y_{t-d} \leq y_{t_8} \\
  c^{(2)} + \phi_1^{(2)} y_{t-1} + \phi_2^{(2)} y_{t-2} + \phi_3^{(2)} y_{t-3} + \theta_3^{(2)} e_{t-3} + e_t & \text{if } y_{t-d} > y_{t_8}. 
\end{cases}
\]
4. GA checking by simulation

A simulation experiment was performed. A thousand artificial time series were generated from each of the two regimes subset SETARMA models referenced 1, 2, 3 and 4.

\[
\text{model 1} \quad y_t = \begin{cases} 
0.8y_{t-2} + e_t & \text{if } y_{t-1} \leq 0.0 \\
-0.4y_{t-1} + e_t - 0.4e_{t-2} & \text{if } y_{t-1} > 0.0,
\end{cases}
\]

\[
\text{model 2} \quad y_t = \begin{cases} 
e_t - 0.8e_{t-2} & \text{if } y_{t-1} \leq 0.0 \\
0.4y_{t-2} + e_t + 0.4e_{t-1} & \text{if } y_{t-1} > 0.0,
\end{cases}
\]

\[
\text{model 3} \quad y_t = \begin{cases} 
0.75y_{t-1} - 0.5y_{t-3} + e_t & \text{if } y_{t-1} \leq 0.0 \\
-0.6y_{t-2} + e_t & \text{if } y_{t-1} > 0.0,
\end{cases}
\]

\[
\text{model 4} \quad y_t = \begin{cases} 
-1.2y_{t-1} - 0.7y_{t-2} + e_t & \text{if } y_{t-1} \leq 0.0 \\
0.8y_{t-3} + e_t & \text{if } y_{t-1} > 0.0.
\end{cases}
\]

For each time series a stretch of 1000 independent standard normal variates were obtained using an algorithm for generating pseudo casual uniform random numbers (L'Ecuyer (1988)) and an algorithm for computing the corresponding normal numbers (Wichura (1988)). Both routines were downloaded from the web site http://lib.stat.cmu.edu/apstat/. Each white noise sequence was then transformed into time series by the threshold model structure. The first 500 artificial observations were discarded to eliminate the transient effect of initial values. So, for each time series 500 observations were available. The procedure expounded in the previous Section was applied to each time series. Results were then averaged, for each model, over the replications where the correct number of regimes \( k \) and delay parameter \( d \) were identified. We chose the size of the GA population equal to 30, the crossover probability \( p_c = 0.9 \), the mutation probability \( p_m = 0.1 \) and the maximum allowed number of generations equal to 200. The constant \( C \) that is to be included in the fitness function (4) computation was set to unity. In Table 1 the number of correct identifications of number of regimes \( k \) and delay parameter \( d \), the mean absolute error (MAE) of the estimated threshold parameter, the average pooled residuals variance, the Ljung and Box's (1978) test \( Q \) rejection percentages for 25, 30 and 35 lags and the number of generations needed for the GA to reach the best solution are reported.
Table 1: Results from the simulation experiment.

<table>
<thead>
<tr>
<th>Model</th>
<th>Correctly Id.</th>
<th>Threshold Par.</th>
<th>MAE of Estim.</th>
<th>Average Pooled Resid's Variance</th>
<th>Ljung-Box’s Test Q% Rejections</th>
<th>GA Generations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.3%</td>
<td>0.1751</td>
<td>1.0542</td>
<td>6.8% 6.6% 7.0%</td>
<td>25 lags 30 lags 35 lags</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>99.6%</td>
<td>0.1431</td>
<td>1.0552</td>
<td>11.4% 9.7% 8.7%</td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>3</td>
<td>95.5%</td>
<td>0.4189</td>
<td>1.0030</td>
<td>7.0% 6.4% 6.2%</td>
<td></td>
<td>119</td>
</tr>
<tr>
<td>4</td>
<td>100.0%</td>
<td>0.4732</td>
<td>1.0940</td>
<td>9.0% 8.5% 8.3%</td>
<td></td>
<td>63</td>
</tr>
</tbody>
</table>

We may see that the number of regimes $k$ and the delay parameter $d$ are correctly identified almost always. Correct identification is only slightly less frequent for model 3, which is a subset SETAR model. The subset SETAR models 3 and 4 exhibit the MAE of the threshold parameter estimates larger than for the subset SETARMA models 1 and 2. The average pooled residuals variance, however, is close to unity for all four models. Very similar too are the rejection percentages for the test Q. These are slightly larger than the nominal 5% value.

In Figures 1, 2, 3 and 4 the histograms of the test Q values, computed for 35 lags, are displayed along with the theoretical chi square probability density function. Curves are quite close each other. This circumstance allows us to conclude that the procedure was a valid one for identification and estimation of models 1-4. It seems that the maximum number of generations set for the GA is appropriate, because the average number of generations needed to reach

![Figure 1. Model 1 Ljung and Box’s Q statistic: empirical and theoretical probability density function.](image-url)
the largest fitness function is much lower than the maximum allowed number of generations.

Figure 2. Model 2 Ljung and Box’s Q statistic: empirical and theoretical probability density function.

Figure 3. Model 3 Ljung and Box’s Q statistic: empirical and theoretical probability density function.
Methods for assessing the threshold parameter were proposed that search the empirical time series percentiles or some other selected values (see, for instance, Lim and Tong (1980), Moeanaddin and Tong (1988) and Tong (1990).) The results that can be obtained by using such methods are likely to be less accurate than those obtained by using the GA. This latter algorithm is able indeed to perform the search in a larger set of candidate values and jointly with the other model parameters. Moreover, if the number of regimes is not known in advance, the same search has to be performed several times. The GA performs instead the search for the number of regimes and threshold parameters simultaneously. This circumstance is specially valuable when the number of regimes is likely to exceed two. We attempted some comparisons concerned with some simple two regimes SETAR models. As a matter of fact, most existing methods developed for identification purpose are supposed to deal with only two regimes models in practice. Let, for instance,

\[
\text{model 5} \quad y_t = \begin{cases} 
-0.8y_{t-1} + e_t & \text{if } y_{t-1} \leq 0.0 \\
-0.2y_{t-1} + e_t & \text{if } y_{t-1} > 0.0.
\end{cases}
\]

A thousand artificial time series were generated from model 5. The same method was used as explained for models 1-4 above. By trying each of the empirical percentiles, and retaining the one that gave the least NAIC, we obtained the average estimated threshold parameter equal to 0.086. The GA procedure
yielded 0.073 instead. The average residual variance was 0.9857 in the former case, 0.9822 in the latter one. Test Q rejection percentages were 8.3% and 5.8% respectively. On the whole, better estimates were obtained by using the GA procedure. We tried other models that led to the same conclusions.

5. Testing the GA procedure for modeling the Canadian lynx data

The annual records of the number of lynx trapped in the McKenzie River district of North-west Canada from 1821 to 1934 are known as Canadian lynx data, and include 114 observations. This time series has been extensively studied (see Tong (1990).) Data are usually transformed as $\log_{10}(\text{number recorded as trapped in year } 1820+t)$, $t = 1, \ldots, 114$. We downloaded the data set from the web site http://www-personal.buseco.monash.edu.au/~hyndman/TSDL.

For using our procedure, we had to set first the largest allowed values for model structural parameters. We chose the maximum delay parameter $D = 3$. The maximum number of regimes was set equal to $K = 3$. Then, we chose $\nu = 2$, so that the maximum allowed number of parameters was 6, and $\mu = 4$, so that the maximum allowed lag was 15. A constant term was included for each regime. We required that at least $m = 30$ observations had to belong to each regime. The maximum length of the chromosome resulted $\ell = 138$. Then, as far as the GA parameters are concerned, we set the size of the population $s = 50$ and the maximum number $N$ of generations equal to 50000. The constant $C$ which is included in the definition (4) was set to unity. We tried several choices for the crossover and mutation probability. The results yielded by the GA procedure for each one of such choices are reported in Table 2. The figures in Table 2 are the minimum NAIC, the best choice for the delay parameter $d$, the residuals mean square error (MSE) and the number of generations to reach the best solution within the maximum allowed number of iterations.

<table>
<thead>
<tr>
<th>$p_c = 0.8$</th>
<th>$p_m = 0.001$</th>
<th>$p_m = 0.01$</th>
<th>$p_m = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MSE</td>
<td>0.02457</td>
<td>0.02841</td>
<td>0.02582</td>
</tr>
<tr>
<td>iter.</td>
<td>36367</td>
<td>12007</td>
<td>35140</td>
</tr>
<tr>
<td>$p_c = 0.9$</td>
<td>$p_m = 0.001$</td>
<td>$p_m = 0.01$</td>
<td>$p_m = 0.1$</td>
</tr>
<tr>
<td>$d$</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MSE</td>
<td>0.02643</td>
<td>0.02749</td>
<td>0.02499</td>
</tr>
<tr>
<td>iter.</td>
<td>27891</td>
<td>11265</td>
<td>30639</td>
</tr>
</tbody>
</table>

Table 2: Performance of the GA procedure varying crossover and mutation probabilities.
We may note that the results depend on the GA parameters $p_c$ and $p_m$. The number of generations needed to attain the largest fitness function value is greater if $p_c = 0.8$ than if $p_c = 0.9$. It seems that higher crossover rate may speed the convergence of the GA. As far as the mutation rate is concerned, the best choice is not $p_m = 0.01$ where it is likely that premature convergence has occurred. Either large or small mutation rates ensure instead better results. The increase of the fitness function values is sharpest within the first $100-200$ generations. Afterwards there are only some upwards steps, which imply some refinement of the estimates. In Figure 5 the behavior of the largest fitness as a function of the generation is displayed for $p_c = 0.9$ and $p_m = 0.1$. The sharp increase of the fitness function ends after about 150 generations, then the improvement of the fitness function largest value may hardly be noticed. Plotting the fitness function was limited to 1000 generations because after generation 653 we had only the single improvement at generation 30639.

![Figure 5](image-url)

Figure 5. Largest fitness function, generations 1-1000 ($p_c = 0.9, p_m = 0.1$.)

The model corresponding to the minimum AIC (obtained for $p_c = 0.9$ and $p_m = 0.1$) is a subset SETARMA(2:(1,4,12;0),(5,9,12;1,2)) with delay parameter $d = 3$, $k = 2$ regimes, and threshold parameter equal to $3.0141$. The pooled residual variance is $0.025$, and NAIC$=-3.5528$. The actual number of observations is 43 for the first regime and 36 for the second one. The model
was as follows

\[
y_t = \begin{cases} 
1.1798 + 0.9339y_{t-1} - 0.1971y_{t-4} - 0.0951y_{t-12} + e_t & \text{if } y_{t-3} \leq 3.0141 \\
4.8977 - 0.7386y_{t-5} + 0.3368y_{t-9} - 0.2154y_{t-12} + 1.3183e_{t-1} + 0.7781e_{t-2} + e_t & \text{if } y_{t-3} > 3.0141.
\end{cases}
\]

For comparison, we estimated a SETAR model by using the GA and for the same choices of \(p_c\) and \(p_m\) and 50000 generations. In this case, our results did not depend on crossover and mutation probabilities, and the model yielded by the procedure was always the same (but \(p_c = 0.8\) and \(p_m = 0.1\) where the model resulted slightly worse,) with threshold parameter equal to 3, pooled residual variance 0.035 and NAIC value equal to \(-3.1922\). The number of observations is 60 in the first regime and 50 in the second one. The resulting SETAR(2;4,3) model is

\[
y_t = \begin{cases} 
1.0763 + 0.8949y_{t-1} - 0.0145y_{t-2} + 0.0083y_{t-3} - 0.2084y_{t-4} + e_t & \text{if } y_{t-3} \leq 3.0 \\
1.0484 + 1.5517y_{t-1} - 1.2751y_{t-2} + 0.3557y_{t-3} + e_t & \text{if } y_{t-3} > 3.0.
\end{cases}
\]

In this latter case we could save one parameter, but both the residual variance and the NAIC are larger than for SETARMA model. Several threshold models for the Canadian lynx data were reported by Tong (1990, pp. 380–381, Table 7.5.) The NAIC values for these models range from \(-3.3438\) to \(-2.061\), all greater than the NAIC obtained by fitting the SETARMA model. Let us consider for comparison the model proposed by Ghaddar and Tong (1981) as reported in Tong (1990, p. 387.) Such SETAR(2;7,2) model is as follows

\[
y_t = \begin{cases} 
0.546 + 1.032y_{t-1} - 0.173y_{t-2} + 0.171y_{t-3} - 0.431y_{t-4} + 0.332y_{t-5} - 0.284y_{t-6} + 0.210y_{t-7} + e_t & \text{if } y_{t-2} \leq 3.116 \\
2.632 + 1.492y_{t-1} - 1.324y_{t-2} + e_t & \text{if } y_{t-2} > 3.116.
\end{cases}
\]

The residual variance is 0.0358. In Figure 6 the original time series is plotted along with the one-step-ahead forecasts from the estimated SETARMA and SETAR models and from the Ghaddar and Tong’s model. The last 34 observations are considered from 1901 to 1934. In Figure 7 the residuals from SETARMA and both SETAR models are plotted for the same years. The SETARMA structure improves the forecasts as the mean squared error is equal to 0.0369 whilst the mean squared error is equal to 0.0422 for the SETAR model estimated by the GA and 0.0416 for the Ghaddar and Tong’s model.
Figure 6. Canadian lynx data: original time series (circles) and one-step-ahead forecasts by SETARMA model (solid line) SETAR (GA) (dot) SETAR (Ghaddar and Tong) (dash).

Figure 7. Canadian lynx data: residuals plot after fitting a SETARMA (solid line) and two SETAR models (estimated by the GA: dot, Ghaddar and Tong model: dash).
6. Conclusions

Self-exciting threshold subset autoregressive moving average models may allow parsimonious structure to be fitted to the data. However, finding the structural parameters is a formidable task, which requires combinatorial optimization techniques. The “space of the solutions” is discrete and large, so that resorting to genetic algorithms for searching such space may be viewed as a promising device. In fact, the objective function, the AIC criterion or one of its many variants, is not to be considered as a function of real parameters only. As a function of the structural parameters, the objective function does not meet the usual requirements for gradient based optimization algorithms be applied. We proposed a procedure based on genetic algorithms, and defined in detail the encoding, the genetic operators and the fitness function (that is, the objective function in the genetic algorithms context) computation. We showed that any self-exciting threshold subset autoregressive moving average model may be put in a form which is convenient for the practical effectiveness of genetic operators. A simulation experiment and application to the well-known Canadian lynx data are presented. The results show that some genetic algorithm parameters, that is the crossover and mutation probabilities, have to be chosen carefully, because the results depend, to a rather large extent, on their choice. It seems that both crossover and mutation rates have better to be taken rather large to yield the best results. Then, adding the moving average structure to the autoregressive one seems very useful for improving the adherence of the threshold model to the data. Moreover, the fitted model does not require too many parameters, that is it may be considered a parsimonious model.

Acknowledgments

The financial support through grants from MIUR, Italy, Research Group PRIN-2000 “Stochastic models and simulation methods for dependent data,” 2000, and from CNR, Italy, “Non linear models and new computational methods in time series analysis and forecasting,” (grant C003399) 2000 is gratefully acknowledged. The present article is based on a paper given at the Conference on Stochastic Models and Simulation Methods for Dependent Data organized by Dipartimento di Scienze Economiche, Gestionali e Sociali, Facoltà di Economia, Università degli Studi del Molise in Campobasso, Italy, 28-29 April 2003. The computer program is written in Fortran and the source code is available from the authors on request. Comments from an anonymous referee are acknowledged with thanks.

References

Akademia Kiado, Budapest, 267–281.


