AUTOMATIC IDENTIFICATION OF ARIMA TIME SERIES BY EXPERT SYSTEMS USING PARADIGMS OF ARTIFICIAL INTELLIGENCE

O. Valenzuela, L. Márquez, M. Pasadas and I. Rojas

Abstract. In this study we seek to resolve one of the most important problems in time series, the identification of the model, using the Box-Jenkins method. Our goal is to obtain an expert system based on paradigms of artificial intelligence, such as fuzzy logic and genetic algorithms, so that the model can be identified automatically, without the necessity for a human expert to intervene. A set of rules based on fuzzy logic is constructed, using as the main source of information the evolution and behaviour of the coefficients of autocorrelation and partial autocorrelation obtained from the time series. Each rule of the expert system is assigned a weight that determines the importance of this rule in the phase of model identification. A priori, the relevance of the rules is unknown, and so the rule system constructed is optimised by means of genetic algorithms.

Keywords: Time series, ARIMA, fuzzy logic, expert systems, genetic algorithms.

AMS classification:

§1. INTRODUCTION

The prediction of time series is addressed not just as a question of academic or mathematical interest, but also because it is of unarguable use, for example, in economics [2], [3]. Perhaps the best, and doubtless the most accurate, way to make a prediction is to apply a model, i.e. a mathematical equation, that reflects all the terms contained within a series. The models used for this are termed ARIMA models.

Box and Jenkins [1] proposed a three-stage practical procedure to obtain an appropriate model (Fig. 1). In the identification phase, two tools are used to measure the correlation between the observations within a single series of data. These tools are the estimated autocorrelation function (ACF) and the estimated partial autocorrelation function (PACF). The following step is to group the statistical relations within a data series by means of a linear mathematical model. Box and Jenkins proposed a wide variety of ARIMA models to choose from; for the present study, the estimated ACF and PACF were used to select one or more appropriate ARIMA models. The basic idea is that each ARIMA model should be associated with theoretical ACF and PACF. During the identification stage, we obtained the estimated ACF and PACF, calculated from the data derived from various theoretical ACF and PACF. Consequently,

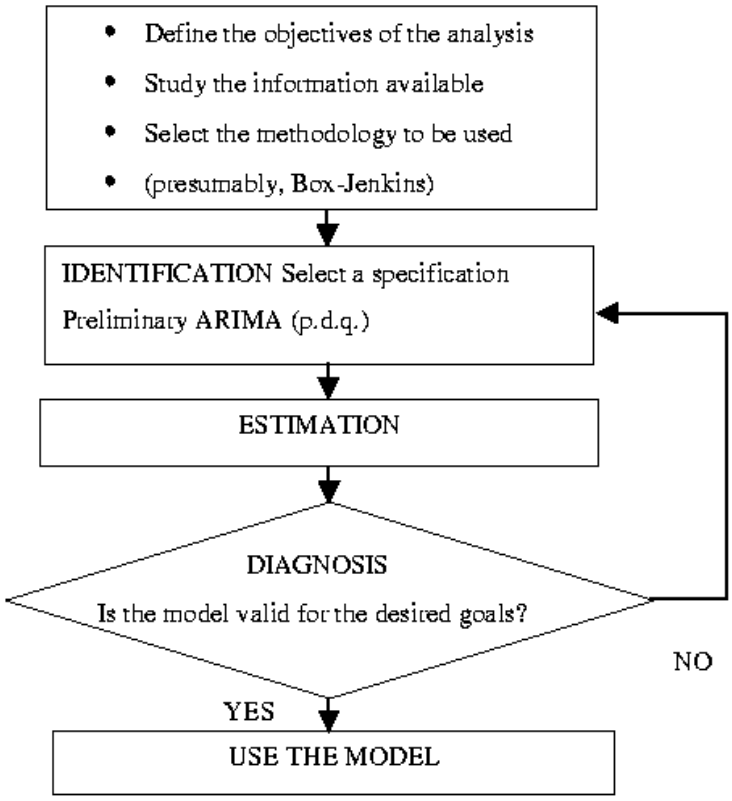

Figure 1: Iterative process of the Box-Jenkins method to create an ARIMA model

we chose the model that most resembled the theoretical one. It should be remembered that any model selected during this stage is merely considered a candidate for the final model. In order to choose a suitable model, the following two procedures are employed. Firstly, in an estimation stage, we obtain accurate estimates of the coefficients for the model selected in the identification stage. In other words, we choose a model and then fit it to the available data, calculating the necessary coefficients. If these estimated coefficients do not meet certain mathematical conditions, the model is rejected [4], [5].

Subsequently, in a checking stage, we make sure that the model is statistically valid. This can be done in various ways, and we may even be led to choose a model that is better than the present one. If the selected model does not fulfil certain requirements, it is rejected and we must return and repeat the identification stage. If all the conditions are met, the model is deemed to be correct and may be used to predict the time series. During the first, identification, stage, we require the intervention of a human expert to identify the model and, by means of a complex calculation process, obtain its parameters. In the commercially-available methods for time series analysis, the user must also propose a time series model. In the present study, we seek to use paradigms based on artificial intelligence (fuzzy logic and genetic algorithms) for the most relevant parts of the Box-Jenkins method, thus eliminating the need for a human expert to identify the model.

§2. DESCRIPTION OF AN ARIMA MODEL

In practice, it is sometimes necessary to include autoregressive terms and moving averages in a single model [6], and the series is then described by the following equation:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - ... - \theta_q \epsilon_{t-q_t} \quad (1)$$

and if we use the backwards displacement or lag operator:

$$\Phi(B) Y_t = \Theta(B) \epsilon_t. \quad (2)$$

This type of process is called autoregressive mixed with a (p,q)-order moving average, sometimes abbreviated to ARMA(p,q). For example, an ARMA(1,1) process would be:

$$Y_t - \phi_1 Y_{t-1} = \epsilon_t - \theta_1 \epsilon_{t-1}. \quad (3)$$

Linear processes must satisfy two main conditions in order to be treated as ARIMA processes: the stationarity condition and the reversibility condition.

2.1. Stationarity condition

As commented above, a process is stationary when it presents a constant mean and variance. The autocovariance and autocorrelations must satisfy a series of conditions for stationarity to be said to exist. For a linear process, these can be grouped into the single condition that the roots of the polynomial equation

$$\Phi(B) = (1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p) = 0 \quad (4)$$

lie outside the unit circle. This can be seen more clearly if we apply this general condition to an AR(1) series in which all the parameters ϕ_i except ϕ_1 are null. The parameter ϕ_1 of an AR(1) series must then satisfy the condition $|\phi_1| < 1$ to ensure that the series is stationary. As the root of $1 - \phi_1 B = 0$ is $B = \phi_1 - 1$, this condition is equivalent to stating that the root of $1 - \phi_1 B = 0$ must lie outside the unit circle. In the general case, for a process AR(p) in which $Yt = \Phi(B)\epsilon_t$, we find that

$$\Phi(B) = (1 - G_1 B)(1 - G_2 B)...(1 - G_p B). \quad (5)$$

And the roots of $\Phi(B) = 0$ must be found outside the unit circle. The equation $\Phi(B) = 0$ is known as the characteristic equation of the process.

2.2. Reversibility condition

The reversibility condition is independent of that of stationarity, and is also applicable to non-stationary linear models. To demonstrate the basic idea of reversibility, let us consider the following model:

$$Y_t = (1 - \theta_1 B)\epsilon_t \quad (6)$$

eliminating ϵ_t in terms of Y:

$$\epsilon_t = Y_t + \theta_1 (Y_t - \theta_1 \epsilon_{t-1}) = ... = Y_t + \theta_1 Y_{t-1} + \theta_1^2 Y_{t-2} + ... + \theta_1^N Y_{t-N} + \theta_1^{N+1} \epsilon_{t-N-1} \quad (7)$$

and if $|\theta_1| < 1$ then the final term in the previous equation becomes less and less important as N increases, and furthermore the weight of each lagged Yt is reduced as the lag value rises. However, if $|\theta_1| \geq 1$, the actual deviation Yt in the above equation depends on the previous Y, the weight of which increases with the value of N. This situation is avoided by requiring $|\theta_1| < 1$. The series is then said to be reversible, and thus the MA(1) model is transformed into an AR(∞) one. These two conditions are taken into account in the rules for the expert system.

§3. IDENTIFICATION OF ARIMA MODELS

On many occasions, a series is not exclusively fitted to one model, but rather various models may equally well fit the series [7]. If we follow the norms of Box and Jenkins, the model chosen is nearly always the simplest one, i.e. that involving fewest terms [1]. The project was carried out in the following way: the time series of a model must fulfil a set of rules. Given such a rule set, and a time series to be analysed, according to the number of rules fulfilled by this series, it will be of one or another type.

A large number of rules can be defined for the simplest AR(1) and MA(1) models, but when these become more complicated, the number of clearly utilizable rules is much lower. This can give rise to problems. For example, let us assume that a single rule is defined for an AR(2) model and that five are defined for an AR(1) model. It is then possible that an AR(2) time series may fulfil 2 of the 5 rules defined for an AR(1) model, as in most cases these rules are not unequivocal. Of course, the given AR(2) series also fulfils the rule corresponding to its own model. Thus, we will eventually have an AR(2) series that fulfils one rule corresponding to the AR(2) model and two that correspond to the AR(1) model. Therefore, it will be identified by the program as AR(1) and an erroneous result will be obtained.

To overcome this kind of problem, we propose the following: firstly, a weight should be assigned to each rule, so that what is finally measured is not the number of rules fulfilled by the series, but rather the sum of the weights of these rules. This procedure provides more accurate results. Secondly, the number of rules defined for each model is approximately the same, and so there is not such a great difference as in the above example.

In summary, what we have to achieve is a program that, on the basis of a given number of rules, is capable of assigning weights to each so that the series that are analysed can be correctly identified.

For this reason genetic algorithms are used to assign weights to the various rules, and this is the novel aspect of the current project. To date, and as observed above, trial-and-error methods have been applied. Fundamentally, a visual examination has been made of the ACF and the PACF, from which relevant conclusions are drawn. This technique, naturally, requires a great deal of skill and long practice. On the basis of this visual examination, the various models possible are identified. An estimate is made of the F and q coefficients, and a decision is taken regarding which of the estimated models best fits the series, using mathematical tests. If there are two models fitted equally well by the series, the simpler one is chosen.

3.1. Initial rules utilized

Before the learning program starts to test the rules fulfilled by each series, a number of simple tasks must be performed:

1. *Obtain the estimated ACF and PACF coefficients and the error criterion.* For this purpose, the above-described formulae are applied.

2. *Check that the series is not white noise*. If 95% of the samples fall within the error criterion, the series is white noise, the model is classified (0,d,0) (where d is the differentiation) and no analysis may be undertaken. This check might not be of any use for the learning program, as what is received by the program is a simulated series and not white noise.

3. *Exponential fit of the first terms of the ACF and of the PACF*. The program attempts to determine whether the shape of the ACF and of the PACF is similar to that of any of the theoretical shapes of the various models described above. To do this, an exponential fit is carried out on the first terms of the ACF and of the PACF, fitting them to an $exp(-\beta x)$ curve. By these means, values of β are obtained for the ACF and the PACF, and these values will be used, together with those of the correlation coefficient, to help identify the model.

4. *Determine the spikes in the coefficients of autocorrelation and of partial autocorrelation*. In fact, if the series is (for example) of the AR(1) type, it will present a spike in the PACF. This is what is determined in this stage of the procedure. As the program is unaware of the type of series presented, it checks the number of ACF and PACF coefficients that are 70% above the error criterion.

5. *Estimate Φ and θ*. When this step is reached, we still do not know the series type, but it is possible to estimate Φ and θ for various known types of series, such that we perform an estimation of coefficients for the following models:
   - AR(1) $\rightarrow$ Estimating the value of Φ_1
   - AR(2) $\rightarrow$ Estimating the values of Φ_1 and Φ_2.
   - MA(1) $\rightarrow$ Estimating the value of θ_1.
   - MA(2) $\rightarrow$ Estimating the values of θ_1 and θ_2.
   - ARMA(1,1) $\rightarrow$ Estimating the values of Φ_1 and θ_1.

   In this way, if (for example) the series is MA(2), the coefficient that is estimated for the Φ_1 case will not fulfil the corresponding mathematical rules and this model will be rejected. As a special case, we might consider that of estimating the θ_1 and θ_2 coefficients for the MA(2) models. Remember that these coefficients are estimated by applying the Yule-Walker equations, which in this case are non-linear second-order equations. They are resolved by means of the non-linear least-squares method.

6. Test changes of sign in the ACF and the PACF. This is performed in order to determine which model is best fitted.

3.2. Rules of the expert system

Once all the above has been accomplished, we can begin to apply the rules for identification. Two types of rule are used, the first of which is somewhat subjective. Our aim is to determine whether the shapes of the ACF and of the PACF are similar to any of the theoretical shapes of the above-described models. The second rule type is purely mathematical, and tests whether the estimated Φ and θ coefficients fulfill all the requirements. Finally, we have a set of negative rules; if these are fulfilled, then the series does not correspond to the model. The last step is to subtract the weights of these rules, rather than summing them, as is done for the other rules.

The expert system that is created consists of a total of 35 fuzzy rules. Due to space limitations imposed on the present paper, we can only discuss a few of these, together with the justification for their inclusion.

- **Rule 1.** *If the PACF fall more abruptly than the ACF, then the model is AR(p), where p is the PACF number immediately above the error criterion.* This rule is suggested by the shape of the AR models. In such a model, the ACF fall smoothly, while the PACF fall abruptly. The number of PACF above the error criterion will be the order of the AR model. To determine this, the program uses the previously-obtained β values of the exponential fit. The exponential presenting the most abrupt change is the one with the highest absolute β value, such that if the β calculated for the PACF is greater than that for the ACF, the model will be AR. To determine the order, we examine the number of spikes in the PACF. As the series are not ideal, but have a component of white noise, we only take into consideration the PACF that are 70% above the error criterion.

- **Rule 2.** *If the ACF fall more abruptly than the PACF, then the model is MA(q), where q is the number of ACF above the error criterion.* This rule is the inverse of the previous one, and the explanation is analogous. As before, the series is not an ideal one, and so to determine the ACF we take into consideration the ACF that are 70% above the error criterion.

- **Rule 21.** *If the estimated Φ_1 and Φ_2 coefficients fulfil the stationarity rules, then the series corresponds to the AR(2) model.* The stationarity conditions for this type of series are:

$$\begin{aligned} \Phi_1 + \Phi_2 &< 1 \\ \Phi_2 - \Phi_1 &< 1 \\ -1 < \Phi_2 &< 1 \end{aligned}$$

  The coefficients must fulfill the three requirements simultaneously; otherwise, the series will not be valid.

As stated above, the system thus created is made up of 35 rules, each of which is assigned a value or relative weight.

§4. FUZZY INFERENCE: ANALYSIS OF THE JOINT ACTIVATION OF ALL THE RULES

After having tested which rules are fulfilled by a series, the following step is to identify the model, taking into account the weights assigned. This is done by summing the weights of the rules fulfilled by the series, taking into account the model to which they refer. In other words, if 5 rules indicate that the series is of the type AR(1), while 3 say it is of type AR(2), we must sum the weights of the 5 rules, on the one hand, and those of the 3 rules, on the other. The larger of these sums then corresponds to the model that is sought.

The above procedure must be carried out bearing in mind that some rules present negative weights. Thus, the last of the rules described are restrictive, that is, if they are fulfilled then the series is not of a given type. The weights corresponding to these restrictive rules must be subtracted.

§5. OPTIMIZATION OF THE WEIGHTS FOR EACH RULE BY MEANS OF GENETIC ALGORITHMS

A genetic algorithm is used to determine the weight assigned to each rule. The limits of the weights range from 0 to 1. The most complex task is the creation of the fitness function, which must perform the following tasks:

1. Using the learning program, evaluate various series of known types, obtaining one or more results for each.

2. Calculate the distances of the possible models obtained for each series from the real model, and store a distance for each series. If various models are possible for a given series, test whether the one assigned the highest weight is at the lowest distance from the real result. If so, there is no penalisation, but otherwise what is stored is the average of the distances of the possible models from reality. This is one way of penalising a surplus of possibilities.

3. Sum the distances thus obtained. The fitness function seeks to minimise the distance. As a maximisation algorithm must be applied, the function to be maximised is then:

$$F = \frac{1}{\sum_i d_i + \epsilon}, \tag{8}$$

where ϵ is a constant required to avoid an infinite result with zero distances, and where d_i is the distance from the model obtained for the series i to the real model. The distance used is the Euclidean distance, i.e. given two vectors $v_1 = (x, y, z)$ and $v_2 = (a, b, c)$, the distance between them is:

$$d(v_1, v_2) = \sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}. \tag{9}$$

§6. RESULTS OBTAINED

The project was successfully concluded, and very promising results were obtained. We analysed a large quantity of real series used as benchmarks in the prediction of time series (http://ubmail.ubalt.edu/ harsham/stat-data/ opre330Forecast. htm). Below, we present a bar chart showing the weights assigned to the different rules, ranging from 0 to 1.

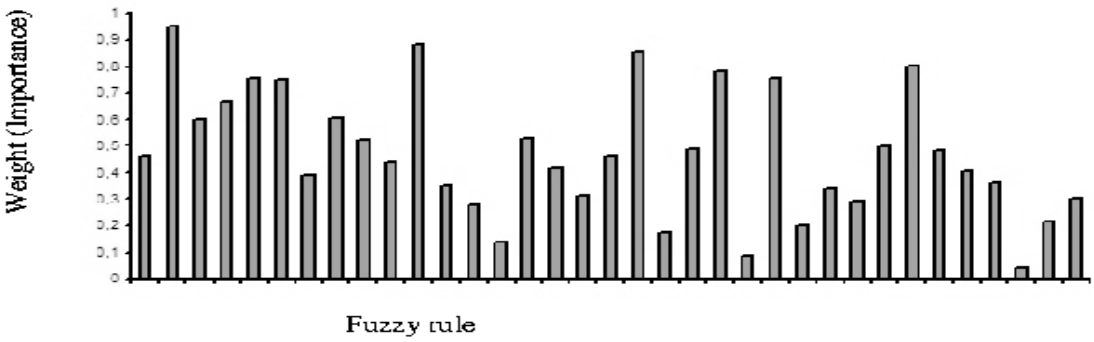

Figure 2 : Weights assigned by the learning program to the different rules

The very highest weights correspond to cases producing "visual" impressions such as "the correlation coefficients diminish smoothly" or "the PACF coefficients have two spikes". These rules mean the program can be applied not just to the series that have been trained, but also to other types. The rest of the larger weights correspond to the more "mathematical " rules, which impose values for the estimated F and q.

Example of a real series catalogued as AR(2)
We analyse a series of the type AR(2), taken from the bibliography, corresponding to personal saving as a percentage of personal income. If people save a higher proportion of their incomes, then less will be spent on goods and services. This reduction in demand may lead to a fall in the national product and to an increase in unemployment. The series is question is shown in Figure 3.

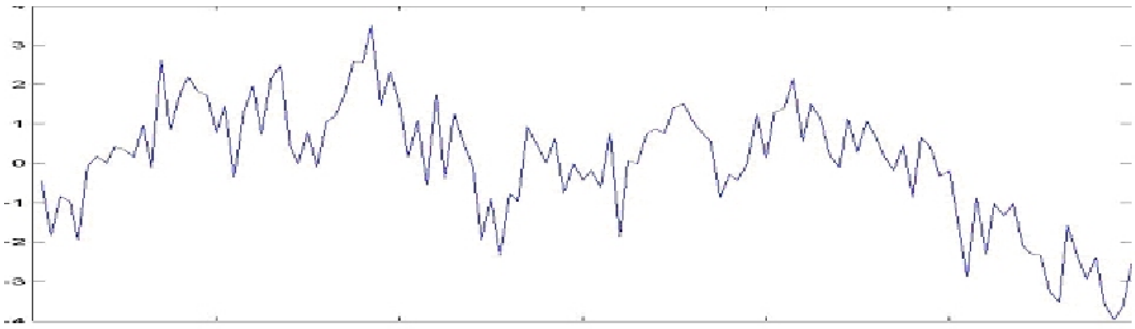

Figure 3: Real AR(2) series

We calculated 48 autocorrelation coefficients and partial autocorrelation coefficients, obtaining the following results:

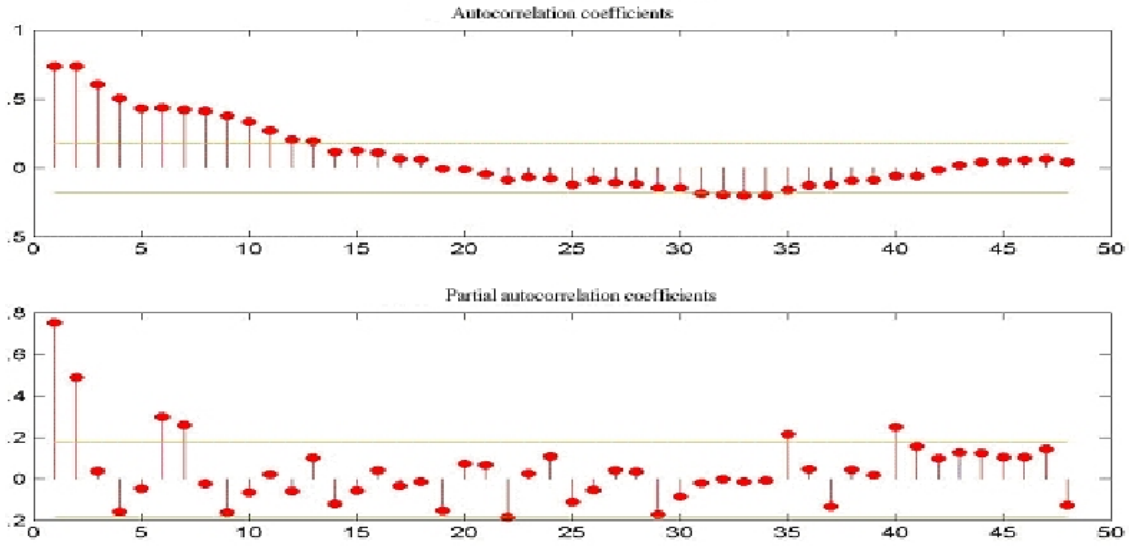

Figure 4 : ACF and PACF for a real AR(2) series

It was found that the series is not constituted of white noise. Analysis of the periodicity also reveals a non-periodic component. The exponential fit of the first terms gives the following results:

ACF: $y = exp(-0.1471x)$ with a correlation coefficient of 0.947

PACF: $y = exp(-0.6643x)$ with a correlation coefficient of 0.5994

In this case, the series could be said to correspond to a pure AR model, because the correlation coefficient of the exponential fit shows that the ACF decrease exponentially, but the PACF do not. This fact is subsequently taken into account by the program. The next step required is to calculate the correlation coefficients that are significantly above the error criterion. We found there to be 5 significant autocorrelation coefficients, but only 2 partial autocorrelation coefficients. This, too, is an indication that the case in question is that of a pure AR model. Calculation of the Φ and θ coefficients gave the following results:

Model	Coefficients	Equation
AR(1)	$\Phi_1 = 0.7361$	$z_t = 0.7361 z_{t-1} + e_t$
AR(2)	$\Phi_1 = 0.4281$ $\Phi_2 = 0.4184$	$z_t = 0.4 z_{t-1} - 0.4184 z_{t-2} + e_t$
MA(1)	$\theta_1 = 5$	$z_t = e_t - 5 e_{t-1}$
MA(2)	$\theta_1 = -0.7208$ $\theta_2 = -1$	$z_t = e_t + 0.7208 e_{t-1} + 1 e_{t-2}$
ARMA(1,1)	$\Phi_1 = 0.9965$ $\theta_1 = 0.8688$	$z_t - 0.9965 z_{t-1} = e_t - 0.8688 e_{t-1}$

Application of the rules produced the following results:

- Rules 10, 11 and 13 suggested an AR(1) model
- Rules 1, 7, 21, 22, 23 and 24 suggested an AR(2) model

- Rules 18 and 28 suggested an MA(1) model

- Rule 29 suggested an ARMA(1,1) model

Taking the weights corresponding to each rule, the model suggested in each case is as follows:

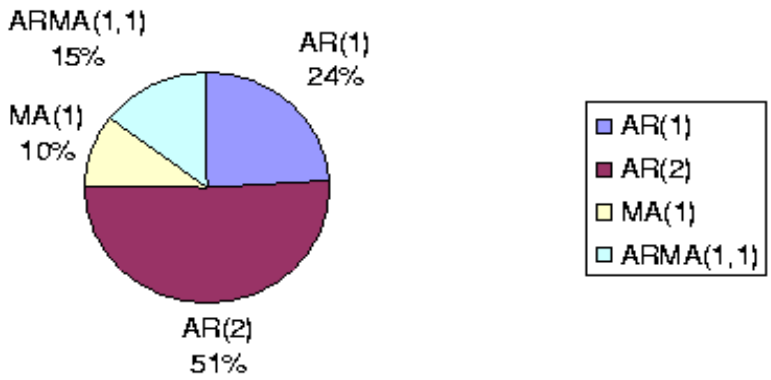

Figure 5 : Weights obtained for the different types of series within a real AR(2) series

This series is clearly of the AR(2) type, with the equation

$$z_t = 0.4z_{t-1} - 0.4184z_{t-2} + e_t.$$

We now present a summary of the results obtained. The percentages were obtained from the simulated series and from the series described in the bibliography.

- AR(1): The programme correctly identified the models for all the series, which is equivalent to 100% of the series of this type identified.

- AR(2): The programme correctly identified the models for all the series, which is equivalent to 100% of the series of this type identified.

- MA(1): The programme correctly identified the models for 90% of the series of this type identified. The series that were not correctly identified were usually confused with the AR(1)model.

- MA(2): The programme correctly identified the models for 90% of the series of this type identified. The series that were not correctly identified were usually confused with the AR(1) and with the MA(1) models.

- ARMA(1,1): The programme correctly identified the models for 70% of the series of this type identified. The series that were not correctly identified were usually confused with the AR(1) and with the MA(1) models.

§7. CONCLUSIONS

The results obtained for the simulations, except in the case of the mixed series, were highly promising. It can be concluded that this program represents an advance in methods of recognising ARIMA models; it obtains good results, and is fast and reliable. It should be noted that currently available commercial programs such as SPSS, Statgraphcs and Matlab require the intervention of a human expert to decide the identification of a model.

Acknowledgements

Part of this study was assisted by the Spanish government project DPI2001-3219 and by Beatriz Aparicio del Moral...

References

[1] BOX, G.E.P., JENKINS, G.M. AND REINSEL, G.C. : *Time Series Analysis: Forecasting and Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1994

[2] HAMILTON, J. D.: *Time Series Analysis*. Princeton University Press, Princeton, 1994.

[3] I.ROJAS, H.POMARES, J.L.BERNIER, J.ORTEGA, B.PINO, F.J.PELAYO, A.PRIETO, "Time series analysis using normalized PG-RBF network with regression weights", *Neurocomputing, vol. 42*, 2002, pp. 267-285

[4] NELSON, C.R.: *Applied Time Series Analysis for Managerial Forecasting*, Holden Day, San Francisco, 1973.

[5] PEÑA, DANIEL: *Estadística Modelos y Métodos. 2 Modelos Lineales y Series Temporales*, Alianza Universidad Textos, Madrid, 1994.

[6] VANDAELE, W. : *Applied Time Series and Box-Jenkins Models*. Academic Press, New York, 1983.

[7] WEI, W.S. : *Time Series Analysis. Univariate and Multivariate Methods*. Addison Wesley, Redwood City, California, 1990.

I. Rojas
Departamento de Arquitectura y Tecnología de Computadores
Universidad de Granada
Granada, Spain

O. Valenzuela, L. Márquez, M. Pasadas
Departamento de Matemáticas Aplicadas
Universidad de Granada,
Granada, Spain