Interval and High Accuracy Results for Parameter Estimation in Time Series Models

M. A. Campos and M. de B. Correia

***Abstract*— The goal of this paper is to apply interval and high accuracy methods [7, 8, 6] to estimating parameters for the AR(1), AR(2), and ARMA(1,1) models [5]. The proposed interval algorithms are extensions of the Modified Moments Method (MMM) [4] to interval values. The results of applying interval methods are compared with the results of applying traditional (floating-point) computations.**

I. Introduction

In this paper, by a *statistical estimation problem* (or simply an *estimation problem*, for short) we mean the following problem:

- *We know:*
  - a statistical model that describes a certain real-life phenomenon;
  - the results of experiments and/or observations of this phenomenon.
- *We do not know* the exact values of the parameters that characterize this particular phenomenon.
- *We want to estimate* the (unknown) parameters of the model (these parameters are also called *population parameters*).

Two kinds of errors are associated with the estimation problem.

- The first type of error is a *statistical* error. The results of observations form a *random sample* from the total population. So, even if all the observations are absolutely precise, instead of the actual value μ_p of the population parameter, we get an estimate μ_s that is computed from the random sample values. For example, an estimator for the population mean μ_p is $\mu_s = (\sum_{i=1}^{n} X_i)/n$, where $(X_1, X_2, \ldots, X_n)$ is the random sample.
- The second type or error is *numerical*. This error is caused by the following two sources:
  - First, the observations and measurements cannot be absolutely accurate. As a result, the observed values x_i are (slightly) different from the actual values X_i of the random sample.
  - Second, the computations are not absolutely precise.

  After we find an estimator μ_s for the unknown parameter value μ_p, we will apply this estimator not to the actual values X_i from the random sample, but to the observed values x_i. The resulting estimate μ_e can thus be different from μ_s. For example, if $(x_1, x_2, \ldots, x_n)$ are the observed values of the random sample, then the estimate for the mean μ_p is $\mu_e = (\sum_{i=1}^{n} x_i)/n$. In general, the estimate μ_s is a function of the random sample, while the estimate μ_e is an *evaluation* of the value of that function μ_s at a given point.

In short:

- Finding the function μ_s that estimates the desired parameter μ_p is a *statistical* problem.
- Estimating the value of this function μ_s is a *numerical* problem.

So, in parameters estimation, there are random and numerical errors.

We want to perform our computations in such a way that we will be able to guarantee the precision of the resulting estimate μ_e. The fact that all operations on a computer have a finite precision means that after each operation, we do not have the precise value; if we know the precision Δ and the computation result $\bar{x}$, then we can guarantee that the actual value of the result belongs to the interval $[\bar{x}-\Delta, \bar{x}+\Delta]$. So, to get operations with guaranteed accuracy, we must perform operations on intervals.

A number of methods to define operations on intervals that produce guaranteed precision (in particular, least significant bit accuracy) have been proposed by Moore [7, 8], Kulisch and Miranker [6], Alefeld and Herzberger [2] among others. Implementations of these methods have been developed in extensions of the languages Pascal, Fortran, C

Departamento de Informática, Universidade Federal de Pernambuco, Brazil, e-mail:{mac,mbc}@di.ufpe.br. The authors are grateful to the referee for his comments.

and C++ [1].

The importance of interval methods for scientific computation is emphasized, for example, by Böhm in [3], where the values of the polynomial $f(x) = 23616x^5 - 161522x^4 + 401773x^3 - 406754x^2 + 87511x + 66576$ are computed for $x = 1.7800, 1.7801, ..., 1.7810$. At first, computations are performed using traditional floating-point arithmetic; the results are:

$$f(1.7800) = 0, \quad f(1.7801) < 0, \quad f(1.7802) > 0,$$
$$f(1.7803) < 0, \quad f(1.7804) > 0, \quad f(1.7805) < 0,$$
$$f(1.7806) > 0, \quad f(1.7807) < 0, \quad f(1.7808) < 0,$$
$$f(1.7809) > 0, \quad f(1.7810) < 0.$$

These results give a misleading information about the location of the real zeroes of this polynomial, because they show that a polynomial of degree 5 has ... 8 zeros. Then, the same values are evaluated by an algorithm based on interval methods and high accuracy arithmetic. The results are:

$$f(1.7800) < 0, \quad f(1.7801) < 0, \quad f(1.7802) < 0,$$
$$f(1.7803) < 0, \quad f(1.7804) < 0, \quad f(1.7805) > 0,$$
$$f(1.7806) > 0, \quad f(1.7807) > 0, \quad f(1.7808) < 0,$$
$$f(1.7809) < 0, \quad f(1.7810) < 0.$$

The goal of this paper is to apply interval and high accuracy methods to estimating the parameters of the AR(1), AR(2), and ARMA(1,1) time series models. The proposed interval algorithms are extensions of the Modified Moments Method [4] to interval values. In the conclusion, we present a comparison between interval and floating-point results.

Box and Jenkins [5] define the *autoregressive* model of order p (denoted AR(p)) and the mixed *autoregressive-moving average model* of orders p and q (denoted ARMA(p,q)), respectively, by the formulas

$$\phi(B)z_t = a_t,$$
$$\phi(B)z_t = \theta(B)a_t,$$

where:

- $\{z_t\}$, $t \in T = \{0, \pm1, \pm2, \ldots\}$ is a sequence that this model describes;
- $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ is the *autoregressive stationary operator*;
- $\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$ is the *moving average invertible operator*;
- θ_i and ϕ_i are real numbers;
- B is the *backward shift operator* defined by $B^m a_t = a_{t-m}$, and
- $\{a_t\}$, $t \in T = \{0, \pm1, \pm2, \ldots\}$ is a sequence of uncorrelated random variables with mean zero and constant variance σ_a^2.

The modified moments method for autoregressive models was proposed by [4]. Its algorithm is based on expressions that use both autocovariance generating function and autocorrelation function (consequently, this method has different versions for AR(p) and ARMA(p,q) models). In this paper, we propose an interval version of this method: *M*odified *M*oments *M*ethod based on *I*ntervals, MMMI for short.

Denotations. In the following text:

- N will denote the number of observed values z_t;
- M will denote the number of iterations;
- $m = 0, \ldots, M$ will denote the number of the current iteration.

Comment. The detailed description of the MMM method can be found in [4]. For the readers who are familiar with statistical methods and denotations, we can present the basic iterative formula for MMM:

$$\hat{\rho}_{k,m} =$$
$$(A/\gamma_0)_m(1/N) + (N/(N-k))r_k(1-(A/\gamma_0)_m(1/N)),$$

where:

- $k = 1, \ldots, K$ is the number of parameters to estimate;
- $A = \gamma(1)$, i.e., A is the autocovariance generating function for $B = 1$, and
- r_k is the sample autocorrelation.

II. Results

We have designed three interval algorithms and implemented them in Pascal-SC [9]. These algorithms compute interval estimates for the parameters of the AR(1), AR(2) and ARMA(1,1) models. In these algorithms, we will distinguish between:

- The *actual* (unknown) values μ_p of the parameters; these actual values will be denoted by ϕ_i and θ_i.
- The estimates μ_s of these parameters that are based on the finite sample. These estimates will be denoted by $\tilde{\phi}_i$ and $\tilde{\theta}_i$. Since we cannot perform all computations with absolute accuracy, we do not know these values either.
- Instead, we will compute *intervals* $I\phi_i$ and $I\theta_i$ that contain the (unknown) values of the estimates $\tilde{\phi}_i$ and $\tilde{\theta}_i$. These estimates were previously denoted by μ_e.

Our three algorithms compute the intervals for the three models:

Series	N	Model	Parameters	Parameters Value	
				MMMI	MLFP
A	197	ARMA(1,1)	ϕ_1	0.8818451^3_2	0.92 ($\pm$.04)
			θ_1	0.5143846^2_1	0.58 ($\pm$.08)
C	226	AR(1)	ϕ_1	0.81198850^8_7	0.82 ($\pm$.04)
D	310	AR(1)	ϕ_1	0.86910036^8_7	0.87 ($\pm$.03)
E	100	AR(2)	ϕ_1	1.392147^8_7	1.42 ($\pm$.07)
			ϕ_2	-0.69458869^8_9	-0.73 ($\pm$.07)
F	70	AR(2)	ϕ_1	-0.30655610^0_1	-0.34 ($\pm$.12)
			ϕ_2	0.19820084^3_2	0.19 ($\pm$.12)

Table 1: Parameters value of the models AR(1), AR(2) and ARMA(1,1)

- Algorithm 1 computes the interval $I\phi_1$ that contains the (unknown) precise estimate $\tilde{\phi}_1$ of the parameter ϕ_1 of the AR(1) model $z_t - \phi_1 z_{t-1} = a_t$.
- Algorithm 2 computes the intervals $I\phi_1$ and $I\phi_2$ that contain the (unknown) precise estimates $\tilde{\phi}_1$ and $\tilde{\phi}_2$ of the parameters ϕ_1 and ϕ_2 of the AR(2) model $z_t - \phi_1 z_{t-1} - \phi_2 z_{t-2} = a_t$.
- Algorithm 3 computes the intervals $I\phi_1$ and $I\theta_1$ that contain the (unknown) precise estimates $\tilde{\phi}_1$ and $\tilde{\theta}_1$ of the parameters ϕ_1 and θ_1 of the ARMA(1,1) model $z_t - \phi_1 z_{t-1} = a_t - \theta_1 a_{t-1}$.

All three algorithms are iterative: on each iteration m, we compute the intervals $I\phi_{1,m}$, $I\phi_{2,m}$, and/or $I\theta_{1,m}$ that contain the precise estimates of the corresponding parameters. The intervals that we get on the last iteration form the desired output ($I\phi_1$, $I\phi_2$, or $I\theta_1$). All three algorithms use interval operations.

The algorithms were run until one of the following conditions was satisfied:

- On the current iteration, we already know the values of the estimates with the desired accuracy $\alpha > 0$, i.e., if the width of all the intervals computed on a given iteration does not exceed 2α (this number α must be given from the very beginning).
- The number of iterations (m) exceeded a given limit M (we chose $M = 100$).
- The intervals obtained on two consequent iterations coincided.

Table 1 contains the results of applying our method MMMI to the data from Box and Jenkins [5]. For comparison, the same table also contains the results of applying maximum likelihood method with floating point arithmetic (MLFP) to the same time series (these results also come from [5]).

Estimates from the column MLFP represent the *statistical* error component: namely, they represent the estimates obtained by the maximum likelihood method, and the standard deviations of these estimates. By combining the estimates e and their standard deviations σ, we can compute the *confidence intervals* $(e - k\sigma, e + k\sigma)$ within which the (unknown) actual values ϕ_1, ϕ_2, and/or θ_1 lie with a given confidence level (i.e., crudely speaking, with a given probability). For example, if we take a confidence level that correspond to "one sigma" deviation, then the numbers corresponding to the series A mean that the actual (unknown) values ϕ_1 and θ_1 are inside the confidence intervals (0.88,0.96) and (0.50,0.66) (with this confidence level).

Estimates from the column MMMI represent the *numerical error* component: for example, the numbers corresponding to series A mean that the (unknown) precise statistical estimates $\tilde{\phi}_1$ and $\tilde{\theta}_1$ for the parameters ϕ_1 and θ_1 are inside the intervals $[0.88184512, 0.88184513]$ and $[0.51438461, 0.51438462]$. In particular, if we use a midpoint of each interval as the estimate, then, as a fit to series A, we get an AR(1,1) model $z_t - 0.88184512 z_{t-1} = a_t - 0.51438461 a_{t-1}$.

III. Conclusions

Numerical and confidence intervals are *different*; their definitions are different, so it is important not to confuse them and not to use the numerical intervals instead the confidence ones.

Statistical errors are usually much larger than numerical ones, so, in many applications (especially if we are only interested in confidence intervals), we can safely neglect the numerical errors and consider only statistical ones. However, for some applications, we need the use the statistical estimates themselves; in this case, it is necessary to estimate them as accurately as possible.

Our experiments has shown if we use $\alpha \leq 10^{-k}$ in the stopping rule, then we get a k digit accuracy in the resulting estimate.

For $k = 11$ and $k = 12$, we have noticed that the larger the size N of the sample, the fewer iterations are needed for our algorithms to convergence, and the smaller are the widths of the resulting intervals.

As a reasonable direction of future research in this area, we propose to apply interval and other high accuracy algorithms to improve the accuracy with which the (endpoints of the) confidence intervals are computed.

References

[1] E. Adams and U. W. Kulisch, **Scientific Computing with Automatic Result Verification**, Number 189 in Mathematics in Science and Engineering, Academic Press, 1993.

[2] G. Alefeld and J. Herzberger, **Introduction to Interval Computation**, Academic Press, N. Y., 1983.

[3] H. Böhm,"Evaluation of Arithmetic Expressions with Maximum Accuracy", *A New Approach to Scientific Computation*, 1983, pp. 121–137.

[4] E. Bora-Senta and S. Kounias, "Parameter Estimation and Order Determination of Autoregressive Models", *Analysing Time Series*, 1980, pp. 93–108.

[5] G. E. Box and G. M. Jenkins, **Time Series Analysis Forecasting and Control**, Holden-Day, 1976.

[6] U. W. Kulisch and W. L. Miranker, **Computer Arithmetic in Theory and Practice**, Academic Press, 1981.

[7] R. E. Moore, **Interval Analysis**, Prentice Hall, Englewood Cliffs, NJ, 1966.

[8] R. E. Moore, **Methods and Applications of Interval Analysis**, SIAM, Philadelphia, Pennsylvania, 1979.

[9] S. M. Rump, "Computer Demonstration Packages for Standard Problems of Numerical Mathematics", *A New Approach to Scientific Computation*, 1983, pp. 28–49.