Model Identification and Parameter Estimation of ARMA Models by Means of Evolutionary Algorithms

Susanne Rolf*
Dept. of Statistics
University of Dortmund
Germany

Joachim Sprave†
Dept. of Computer Science
University of Dortmund
Germany

Wolfgang Urfer‡
Dept. of Statistics
University of Dortmund
Germany

1 Introduction

The field of time series analysis and forecasting methods has significantly changed in the last decade due to the influence of new knowledge in non-linear dynamics. New methods such as artificial neural networks replaced traditional approaches which usually were appropriate for linear models only.
Nevertheless, there are still applications where accurate estimations of linear processes, such as ARMA models, are sufficient. However, the methods for this class of models were developed more than 20 years ago, with the restrictions of those days' computers in mind. The work described in this paper is an attempt to combine the ideas of the widely used Box-Jenkins method [2] with new approaches to model identification and parameter estimation based on Evolutionary Algorithms, a class of probabilistic parameter optimization methods.

2 Traditional ARMA Process Estimation

The ARMA$[p,q]$- or Box-Jenkins-model is one of the most traditional techniques in statistical time series analysis. The assumed model is of the form

$$x_t = \alpha_1 x_{t-1} + \ldots \alpha_p x_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \ldots - \beta_q \varepsilon_{t-q},$$

where p is the order of the autoregressive part, q is the order of the moving-average part, $\alpha_1, \ldots, \alpha_p$ are the autoregressive parameters and $\beta_1, \ldots, \beta_q$ are the moving-average parameters. ε_t denotes the series of errors. The time series x_t is supposed to be stationary.
The statistician's way of approaching the modeling of such an ARMA$[p,q]$ process is to first determine the model orders p and q. Therefore, it is necessary to calculate some basic quantities of the time series, namely the empirical autocorrelation function and the empirical partial autocorrelation function. The plots of these two functions give hints with respect to the model orders. Nevertheless, this method is only useful for low model orders and does not provide a reliable tool for model identification.
Once the orders of the process are determined, it is necessary to estimate the model parameters. The two methods used for this problem are least-squares and maximum-likelihood estimation. For both possibilities a corresponding function has to be established and optimized. The optimization itself is usually made by means of numerical methods which -

especially in problems of such complexity - do not guarantee a convergence to the global optimum.
The necessary third step in the process of ARMA modeling is the verification of the found model's plausibility, called the model diagnosis. It covers - among others - the following diagnosis tools:
The Ljung-Box statistic provides a useful tool to investigate the independence of the residuals, which is an important assumption for the model. The statistic essentially contains the sum of the squared residual correlations up to a specified lag. This sum being too large indicates that the model is not sufficient for the data.
To be sure that the fitted model is stationary, the zeroes of the characteristic polynomial of the model's AR-part have to be evaluated. The model is stationary if all these zeroes lie outside of the complex unit circle.
Furthermore, to avoid parameter redundancy the zeroes of the characteristic AR-polynomial have to be compared to the zeroes of the characteristic MA-polynomial. Common zeroes indicate parameter redundancy, which means that the model can be shortened by two parameters.
Since there are various reasons to keep the model order as low as possible, information criteria can be introduced to combine the need for a good fit with the "principle of parsimony". These criteria (e.g. Bayesian Information Criterion, Akaike's Information Criterion) join the residual variance on the one hand and the model orders on the other hand. The analyst's aim is then to minimize such a criterion.
If all these restrictions and conditions can be fulfilled, the found model can be accepted. On the other hand, if one of them is violated, the whole process of model identification, parameter estimation and model diagnosis has to be repeated until an appropriate model has been found.

3 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are adaptation and optimization methods based on the Darwinian model of natural evolution. Although developed independently, all variants of EAs (Genetic Algorithms[5, 4], Evolution Strategies[7, 9], Evolutionary Programming[3]) can be seen as particular realizations of the same idea. In fact, EAs are not algorithms in the usual meaning, but form a class of methods based on similar principles.
By far the most applications of EAs are parameter optimization tasks: For a given objective function

$$f : D \mapsto \mathbb{R}, \quad D \subseteq \mathbb{R}^n$$

where n is the number of problem parameters, find the point

$$x^* \in D : f(x^*) \leq f(x) \quad \forall x \in D.$$

In EAs, points in the search space are modeled as *individuals* in an artificial environment. In contrast to most other optimization methods, a set of points (*population*) is handled in each iteration rather than just a single point. The chance of being selected as a parent of the next generation depends on the *fitness* of an individual, which is usually a simple transformation of the objective function. New search points are generated by

- Reproduction:
  Create offspring from one parent (asexual reproduction), or from two or more parents (sexual reproduction, recombination, crossing over).

- Mutation:
  Apply small random changes to new search points.

- Selection:
  Individuals with a high fitness are more likely to be selected for reproduction.

The major advantage of EAs is that they do not require any knowledge of the characteristics of a given optimization problem. They do not depend on certain properties of the objective function, such as convexity, differentiability, or smoothness. EAs are especially useful for global optimization tasks, where deterministic methods suffer from being attracted by local minima. For a comprehensive introduction to EAs, the reader is referred to [1]. In the following, we focus on the differences between the EA used for the work described here and the traditional implementations.

3.1 Internal Representation

An important aspect in the definition of a fitness function for an EA is the so-called *genotype to phenotype mapping*. Following the model of natural evolution, the genotype is the internal representation of an individual, like a blueprint. The phenotype is a particular realization of this blueprint, so two individuals with identical genes may look different due to environmental influences. Likewise, the internal representation used in EAs must be transformed in a problem specific form to evaluate the fitness. E.g., the traditional encoding of an individual in Genetic Algorithms is a binary string. In order to evaluate such a string as a set of fixed point numbers, bits must be grouped together, interpreted as integer numbers, and scaled to the application-dependent parameter range.
For almost every successful real-world application, the homogeneous internal representations used in the basic EAs is not well suited. Instead, a problem specific internal representation must be defined in order to avoid domain restrictions and repair operators. For a discussion of non-standard representations, see [6].

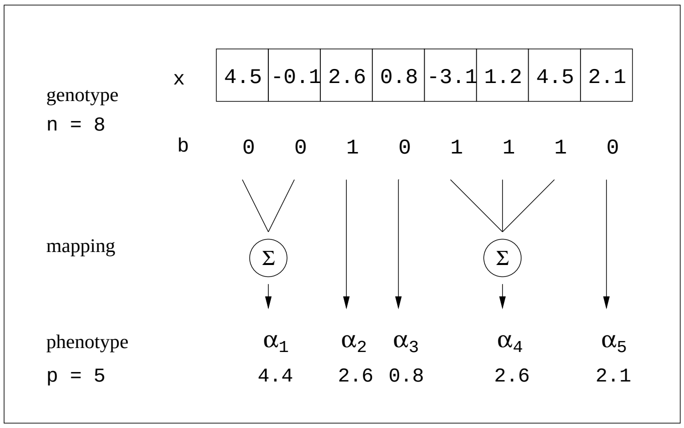

Fig. 1: Mapping from a fixed dimensional internal representation to a variable length AR parameter vector. The resulting order is $p = 5$

Since the AR and MA models orders are variable, the number of real values required for an evaluation is not fixed. A mapping that transforms a fixed dimensional internal representation to variable dimensional problem instances was introduced in [8]. The genome in the EA consists of a vector of real values $x_i \in \mathbb{R}, i = 1, \ldots, n$, and a bit string $b, b_i \in \{0, 1\}, i = 1, \ldots, n$. Real variables are summed up as long as the corresponding bits are equal. Obviously, n is an upper bound for the dimension of the resulting parameter vector.

3.2 Genetic Operators

Mutation and recombination of the real variables are taken from the Evolution Strategy as described in [10]. For the binary strings, the standard Genetic Algorithms operators, i.e. bit flip mutation and two point crossover, are used.
The selection operators for the experiments described here are local variants of the (μ, λ) Evolution Strategy selection, as suggested in [11].

4 Estimation of the Error Series

The main problem of the error series estimation lies in the fact that the quality of the parameter estimation depends very much on the accuracy of the estimated error series. The simple traditional way of obtaining the errors is to fit a very long AR-process to the given time series by means of the Levinson-Durbin recursion and to use the residual series of this process as an estimate for the error series.
In this approach, another idea of error estimation is used: Instead of using the suggested AR-process to gain the residuals, the residuals of the currently best fitting ARMA-model are used as error series for the next iteration:

$$\hat{\hat{\varepsilon}}_t = x_t - (\alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} - \beta_1 \hat{\varepsilon}_{t-1} - \ldots - \beta_q \hat{\varepsilon}_{t-q}).$$

As a preliminary estimate for the first step, all elements are set to their expected value zero. This technique leads to a permanent improvement of the fit of the error series as well as of the model itself. This means that the same set of parameters can result in different fitness values depending on the current error estimation.
The approach takes advantage of one of the features of Evolutionary Algorithms: They are robust against noise and therefore enable the whole method to estimate the error series on-the-fly.

5 Definition of the Fitness Function

The aim of combining traditional ARMA modeling with Evolutionary Algorithms is to provide a tool that offers model identification as well as parameter estimation and model diagnosis without needing the help of a user in the first place. Therefore, the Evolutionary Algorithm has to optimize the model order and the parameters simultaneously with respect to the restrictions made available by the different model diagnosis tests.
As a fitness function both least squares and likelihood function can be taken into consideration. For the least squares estimation at first the series of errors has to be estimated, which is done as described above. After this the least squares function

$$\sum [x_t - (\alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} - \beta_1 \hat{\varepsilon}_{t-1} - \ldots - \beta_q \hat{\varepsilon}_{t-q})]^2$$

can be calculated for each individual to evaluate its fitness.
It is now possible to add the different statistical model diagnosis tools to improve the estimation's quality:
To decide, whether or not a parameter is significantly different from zero, the statistical t-test with hypothesis "The parameter equals zero" can be applied to each estimated parameter. Alternatively, confidence-intervals for the parameters can be constructed to make sure that all estimated parameters are needed for the model. If the test did not reject the hypothesis for a parameter or a parameter's confidence-interval contained zero, this parameter would be removed from the parameter set.
To make sure that there is no parameter redundancy and that the suggested model is stationary, the zeroes of the characteristic functions of the AR- and the MA-part have to be

evaluated and compared as described above. All these criteria can be used as restrictions to the parameter space.
The Ljung-Box statistic on the other hand could be used both as restriction or as part of the fitness function: For a restriction the statistic would be compared with the 95%-quantile of the corresponding χ^2-distribution. The test would reject the hypothesis of independent residuals, if the value of the statistic was larger than the quantile. On the other hand the statistic's value could be looked at as another criterion to minimize, since a low correlation of the residuals is desirable.
The information criteria naturally can not be used as restrictions, but since they contain the model orders as well as the fit, they could be used as fitness functions.
After all, there is more than one function that would be desirable to optimize and so it is necessary to take multi-criteria optimization into consideration.

6 First Results

Although the practical work is still in the beginning, a few experiments have been made to show the feasibility of the approach. In order to be able to proof the correctness of the results, synthetic ARMA$[p,q]$ time series up to $p,q=3$ were used for this study.
As a first positive result it can be said that the estimation of the error series is very successful, the estimates approach the true simulated series rapidly, as can be seen in figure 2.

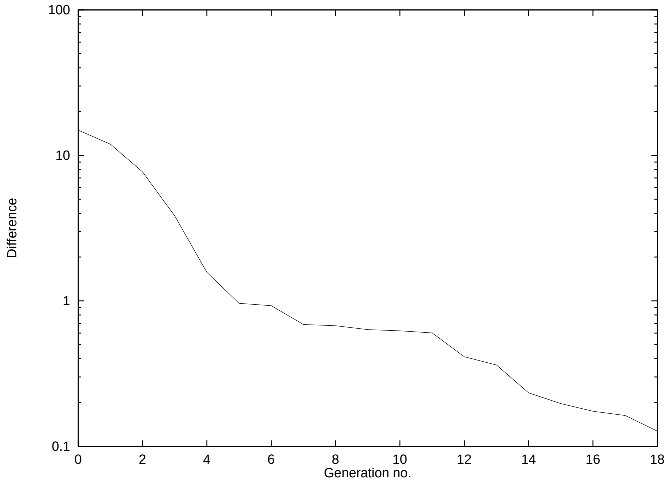

Fig. 2: Typical decrease of the mean absolute difference between the real errors from the synthetically generated time series and the errors estimated by the EA over the first 18 generations. The standard deviation of the real errors is 1.0.

As far as the achieved fit is concerned, the figures are also convincing. Though the model orders are not always correctly identified, the least squares distance reaches values comparable to those reached by the Statistical Analysis System (SAS) [12]. For the latter one, of course, the correct model orders must be given.
As already mentioned, the identification of the model orders is not reliable yet. The frequency of correct identifications depends very strongly on the given time series. As an average over all tested models it can be said that a quota of about 20% of the models have been identified correctly.
To improve the frequency of correct identifications, different diagnosis tools have been introduced and tested: The t-test on significant difference of the parameters from zero was thought to be useful to reduce the estimated model orders, but turned out to be unable

to improve the former results since it eliminated not many parameters. Better results are expected from the introduction of confidence-intervals instead of the test, since they do not depend on an α-level which is almost impossible to control for such a large number of tests. As well as the t-test, the test for stationarity turned out to be not necessary, the algorithm automatically estimates stationary models. This result is not surprising since the algorithm would be expected to produce stationary models when there is no trend or season in the simulated time series.
The χ^2-test on uncorrelated residuals behaved contrary to the t-test: Introduced as a restriction to the model, the Evolutionary Algorithm turned out to be unable to find solutions in the feasible region. The approach to introduce the statistic as part of the fitness function or as a further optimization criterion therefore seems to be more promising.

7 Conclusions

A new approach to parameter estimation in statistical applications has been introduced. From the first results it can be seen that EAs are suitable tools for the given problem, although they cannot compete with 20 years of experience which is integrated in the traditional methods. However, the ARMA estimation problem is only one example to show the feasibility of the use of EAs in this field. As soon as the given problem is either not well analyzed or too complex to develop a corresponding heuristic, EAs may be the last resort.
It can also be concluded from the first experiments with various statistical tests that multi-criteria optimization should be preferred to penalty functions when the quality of the solution depends on additional statistical quantities.

Acknowledgments

Susanne Rolf and Joachim Sprave gratefully acknowledge support by grant 01 JR 509 A of the German *Federal Ministry of Education, Science, Research and Technology* (BMBF).

References

[1] Th. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, 1996.

[2] G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, San Francisco, 1970.

[3] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution.* Wiley, New York, 1966.

[4] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, Reading, MA, 1989.

[5] J. H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[6] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, Berlin, 1994.

[7] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Dissertation, Technical University of Berlin, Berlin, Germany, 1971.

[8] M. Schütz and J. Sprave. Application of parallel mixed-integer evolution strategies with mutation rate pooling. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 345–354. The MIT Press, Cambridge, MA, 1996.

[9] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume 26 of *Interdisciplinary Systems Research.* Birkhäuser, Basel, 1977.

[10] H.-P. Schwefel. *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology Series. Wiley, New York, 1995.

[11] J. Sprave. Linear neighborhood evolution strategy. In A. V. Sebald and L. J. Fogel, editors, *Proceedings of the Third Annual Conference on Evolutionary Programming*, pages 42–51. World Scientific, Singapore, 1994.

[12] Statistical Analysis System Institute, Cary, NC. *SAS ETS user's guide*, 2. edition, 1993.