Evolutionary Based ARIMA Models for Stock Price Forecasting

Tomas Vantuch and Ivan Zelinka

VSB-Technical University of Ostrava, 17. listopadu 15 708 33,
Ostrava-Poruba, Czech Republic
{tomas.vantuch.st,ivan.zelinka}@vsb.cz

Abstract. Time series prediction is mostly based on computing future values by the time set past behavior. If the prediction like this is met with a reality, we can say that the time set has a memory, otherwise the new values of time set are not affected by its past values. In the second case we can say, there is no memory in the time set and it is pure randomness. In a faith of "market memory", the stock prices are often studied, analyzed and forecasted by a statistic, an econometric, a computer science... In this article the econometric ARIMA model is taken for previously mentioned purpose and its constructing and estimation is modified by evolution algorithms. The algorithms are genetic algorithm (GA) and particle swarm optimization PSO.

Keywords: ARIMA, GA, PSO, AIC, BIC, forecasting.

1 Introduction

Nowadays, there is quite often used the econometric ARIMA model [1] proposed by Box and Jenkins for an analysis and a forecast of time series because of its complexity and variability. The main part of the model is the combination of auto-regression (AR) and moving-average (MA) polynomials into one complex polynomial:

$$y_t = \mu + \sum_{i=1}^{p}(\gamma_i y_{t-i}) + \sum_{i=1}^{q}(\theta_i \epsilon_{t-i}) + \epsilon_t \tag{1}$$

This model is based on statistic analysis of a time set. At first, it has to be fulfilled the condition of a stationarity of a time set. A time set is stationary if it does not contain any trends or seasonal behavior and its mean and variance does not change over time. The condition of stationary behavior is crucial for input time set.

The next step in ARIMA modeling is an estimating the model [4–9]. It means to estimate p and q parameters for AR and MA polynomials and the level of differentiation.

An auto-regression is the polynomial that contains variables of the time set moved q-periods back in time and multiplied by AR coefficients γ. This sum is later increased by model parameter μ and white noise ϵ.

A. Sanayei et al. (eds.), *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, Emergence, Complexity and Computation 14, DOI: 10.1007/978-3-319-10759-2_25

$$y_t = \mu + \sum_{i=1}^{p}(\gamma_i y_{t-i}) + \epsilon_t \tag{2}$$

MA polynomial does not contain any variable from a time set and it has nothing to do with familiar known moving-average function. MA contains its own set of MA coefficients multiplied by model residuals and at the end the whole sum is increased by model parameter and white noise.

$$y_t = \mu + \sum_{i=1}^{q}(\theta_i \epsilon_{t-i}) + \epsilon_t \tag{3}$$

The letter "I" in ARIMA model stands for "integrated", which means the level of differentiation of the time set.

The values of p and q parameters for AR and MA part can be obtained from the behavior of auto-correlation (ACF) and partial auto-correlation (PACF) function in the nth level of differentiation [1, 9]. There are alternative methods, that use patterns evaluation by symbols "X" and "O" in a matrix, like SCAN [6], ESACF [7] to determine p and q values.

By all this tasks we can say that the finding of the suitable ARIMA model can be harder and not very solid job and it could be the point of other improvements.

This opportunity is taken in this article. We will try to use an evolution algorithm approach [10] to find the suitable model for the time set. This approach requires to have some clear evaluation function to know if the reproduced model is good enough. The evaluation can be made of AIC [4] and BIC [5] criteria. Both of them are designed like a likelihood penalization criteria

$$[2logL + kp] \tag{4}$$

where L stands for the maximal value of the likelihood function of the model, p is count of model parameters, k has for AIC the value of 2 and for BIC the value is $log(n)$.

The next step it to estimate AR and MA coefficients for our previously obtained model. This coefficients are normally gained from maximum likelihood function [2] that is based on maximization of searched parameters' probability:

$$L(\Theta, x) = f_{\Theta}(x) = f(x_2|x_1) * f(x_3|x_2)... * f(x_n|x_{n-1}) \tag{5}$$

This maximum likelihood function can be replaced by other approaches too. For this experiment there is used the particle swarm optimization algorithm [13].

With estimated coefficients we will be able to create forecasting and evaluate it to find out if it is good enough or what can be improved.

2 Experiment Design

In this experiment, as it was mentioned before, the main idea was to work with ARIMA model. The first part of the experiment is to create ARIMA(p,d,q) model by the genetic algorithm [10]. For the GA we will use Java framework ECJ [15], that is known solution for evolutionary based computing methods. The creating of the model by individuals' genotype and its evaluation by AIC and BIC criteria is provided by Matlab econometric toolbox.

This evolved model will have the AR and MA coefficients computed by MLE method, so in the next step we will try to breed the new parameters by PSO algorithm. As an evaluation function for this process will be the quality of the forecasting of the time set evaluated by MSE.

In the end we will compare gained results by forecasting made by classic approach with PSO and we will see whether it is worth.

2.1 Data Set

For this experiment historical data of a stock market title as Microsoft(MSFT) were used. This historical data consist of records known as candles, that contains information of some period of time, in this case the period of time was one day. Each candle contains information about the highest price of the period (HIGH), the lowest price of the period (LOW), the first price of the period (OPEN), the last price (CLOSE), the quantity of traded instruments (VOLUME) and of course the time stamp of the period. There are more than ten years of observations but in this experiment, there was not used more than four years of its length, mostly because of longer execution time of all the algorithms. All the historical data were taken from yahoo finance [https://finance.yahoo.com].

3 Analysis of Historical Data

As it was said before, at first we analyze our time set for stationarity, autocorrelation and we create the suitable ARIMA model by ACF and PACF approach.

Because we worked with the time set of the stock prices, we can be sure, that this time set will contain up-trends or down-trends, which indicates its non-stationarity. Proceeded Dickey-Fuller test [16] rejected our hypothesis about stationary behavior by resulted value $\gamma* = 0$ as well.

Afterwards, we compute the first differentiation and draw the process of ACF and PACF function. It is expected the "tailing" of the ACF, that will indicate the value of q parameter and "cutting off" of the PACF, that will indicate the value of p parameter.

The chart of ACF shows, that there is no confirmed correlation between time set variables and it indicates stationary behavior of the time set. We check it by new Dickey-Fuller test and the result $\gamma* = 1$ confirms the hypothesis that the differentiated time set is no more non-stationary.

These charts' progresses show some possible combinations of the future ARIMA models. The p and q values will be lower by this method, because there was no present of correlations between variables.

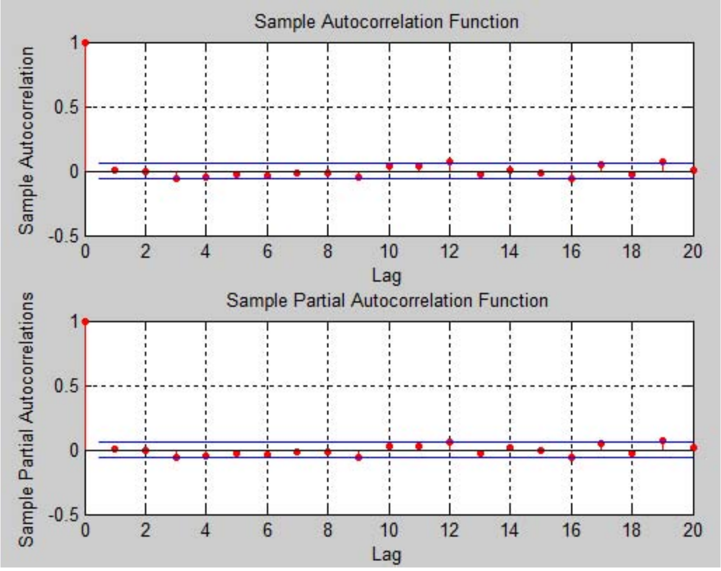

Fig. 1. Progress of ACF and PACF function after first differentiation

4 Genetic Algorithm Approach for Model Building

The suitable ARIMA model for later testing is evolved by the genetic algorithm. The Genetic algorithm (GA) belongs to evolution algorithms [10]. This set of algorithms works like an imitation of the evolution theory described by Darwin. The GA purpose is to evolve the best combination of variables from the scanning set.

As a first step of the GA there is a population of random individuals. Each of the individuals is represented by the binary vector with the length of 12. The transfer from genotype to phenotype is made by the splitting of whole vector into three parts by four bits and each part belongs to some ARIMA variable like ARIMA(p = 1011,d = 0010,q = 1100).

There is a fitness function for an evaluation of individuals. By the theory of evolution, the fitness function simulates the natural selection. It has a purpose that better individuals according to the fitness will live longer and create the new generation by its crossover. The one-point crossover [12] was used for crossbreeding in this experiment.

The breeded individuals are effected by an added mutation. It means that in the random time, there can be changed one bit of an individual genotype. This process brings to this algorithm some added randomness.

The fitness function of this GA covers the requirement of the minimal value of AIC and BIC and some added advantage for individuals with greater values of p and q parameters. This advantage is added because the longer ARIMA polynomial we will have as result, the more we can optimize by the PSO.

$$fitness = (10^5/AIC + BIC) + p + q \tag{6}$$

The GA runs in this case by this steps:

1. Initialization - creating the first generation from one prototype individual
2. Repeat in limited count of cycles
   (a) Evaluate individuals:
      i. Create the ARIMA model by the binary vector
      ii. Estimate parameters by the default MLE function
      iii. Return AIC and BIC by the minimal likelihood function value
   (b) Proceed crossover on the selected individuals to create new generation
   (c) Add random mutation
3. End of loop
4. Return the best individual

The GA returned in this experiment the ARIMA model with parameters $p = 12, d = 2$ and $q = 8$. This result was compared to the previously chosen models.

Table 1. ARIMA models evaluation

	BIC	AIC
ARIMA(12,2,8)	**485.6266**	**400.4396**
ARIMA(1,1,0)	415.0880	403.9767
ARIMA(0,1,1)	415.0647	403.9534
ARIMA(2,1,1)	423.6821	405.1632
ARIMA(1,1,2)	428.7655	406.5429
ARIMA(2,1,3)	434.0470	408.1205

The model from the GA has softly greater BIC value but because of lower AIC and much greater count of coefficients, is was chosen as the model for the later PSO optimization and prediction.

5 PSO Approach for Parameters Estimation

The evolutionary algorithm used for the breeding of the model coefficients is the particle swarm optimization [13].

The PSO is known, very powerful algorithm inspired by swarm intelligence. Each individual is described by its position, velocity and memory of the latest best position. This algorithm is not divided into generations, because individuals are not dying and creating again, they are just moving during iterations in N-dimensional space. Their moves are affected by its previous best position or the best position in its neighborhood [14]. The quality of the position is evaluated by the objective function, and during every iteration, all the current positions of all individuals are confronted with its best positions.

Obviously the count of dimensions of the searched space influences the count of individuals, which are adjusted. In this case we have 22 dimensional space, We need to breed the AR coefficients, which are the vector of size 12. Than we need to breed MA coefficients, which are the another vector of size 8 and two added variables - one for model constant μ and another for variance.

In the case of PSO, it was used the item of neighborhood [14] which means that individual is not affected by the best position of all individuals, but only by the best position of smaller amount of closer individuals, its neighbors. The neighborhood is not defined by the distance between individuals, but randomly. The present of the neighborhood brings to this system the opportunity find the best global maximum from more observed local maximums.

The objective function in our case is to decrease the value of MSE of forecasted values to minimum.

The algorithm for parameters breeding:

1. Initialization - place all particles in n-dimensional hyper space and adjust to them the velocity and the neighborhood
2. Repeat in limited count of cycles
   - (a) Evaluate the position of individual:
     - i. Create the ARIMA model from obtained parameters
     - ii. Create the forecasting for next n values
     - iii. Compare the forecasting values to real values and compute MSE
   - (b) Actualize particles position
   - (c) Actualize the previous best position
3. End of the loop
4. Return the particle with the best position

Concretely, each particles' evaluation consists of the creating of the ARIMA model and splitting time set. Time set has to be split into two parts. At first there is the input time set for ARIMA model and the second part will be compared to this ARIMA model prediction. The predicted values are compared to the real values (second part of the time set) and the best fit is saved as the best position.

The best position will do the forecast of next n values and this will be compared to forecast of the same ARIMA model but with coefficients made by the MLE. Arima model has to be estimated and it was provided in Matlab. The same subset from time set was used in case of "estimate" Matlab command and also in the PSO approach.

6 Conclusions

This article covers the simple idea of working with ARIMA models by the approach of evolution algorithms. In the section of modeling, there was described very simple kind of obtaining quite suitable ARIMA model without knowledge of stationarity, ACF or PACF. The resulted model had sufficient low values of likelihood penalization criteria like AIC and BIC.

In the second part, there was a task to estimate AR and MA coefficients in ”training part” of time set by the PSO algorithm and the best set of coefficients (PSO particle) was used to create prediction compared to classic ARIMA prediction.

There were created two tests for this conclusion. Their difference is the size of time set, adjusted for ”training phase”. The first has ”training phase” of size of ten and the second has this size increased to twenty. The count of particles and iterations was in both test adjusted to same values (particles = 100, iterations = 40).

There are some charts to provide our results.

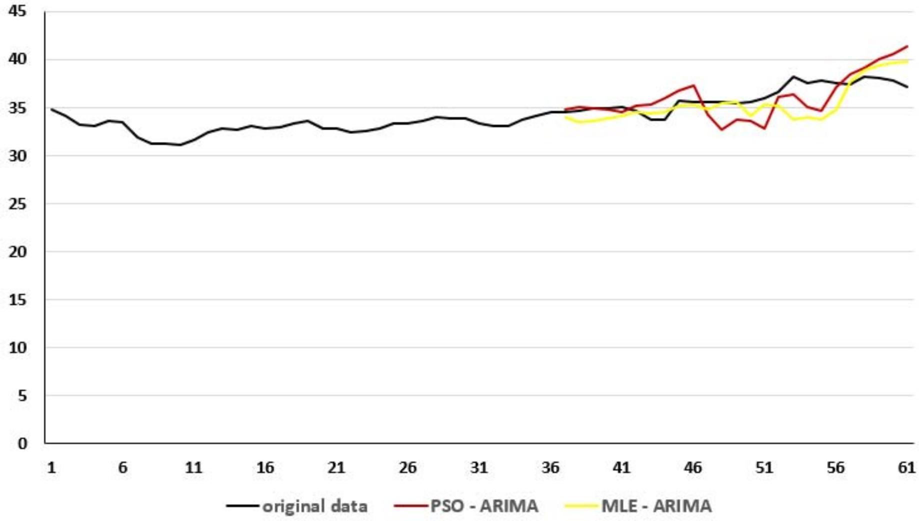

Fig. 2. Chart of prediction when size of the ”learning phase” was 10

This chart describes the value of MSE between forecasting and real values of time set.

As we can see, these tests do not prove that the estimation of the coefficient by PSO has significantly improved the ARIMA results, but we can say that this combination is at least comparable to standard ARIMA computing.

There are some areas for improvement of this approach. One of the weaknesses of these tests was very low number of PSO iterations. PSO can obtain better results in parallel computing, where it can provide more iterations in a shorter time.

The other improvement can be gained by adjusting bigger time set for learning phase and to work with bigger values of p and q of ARIMA model. Both of these tasks require more computing resources.

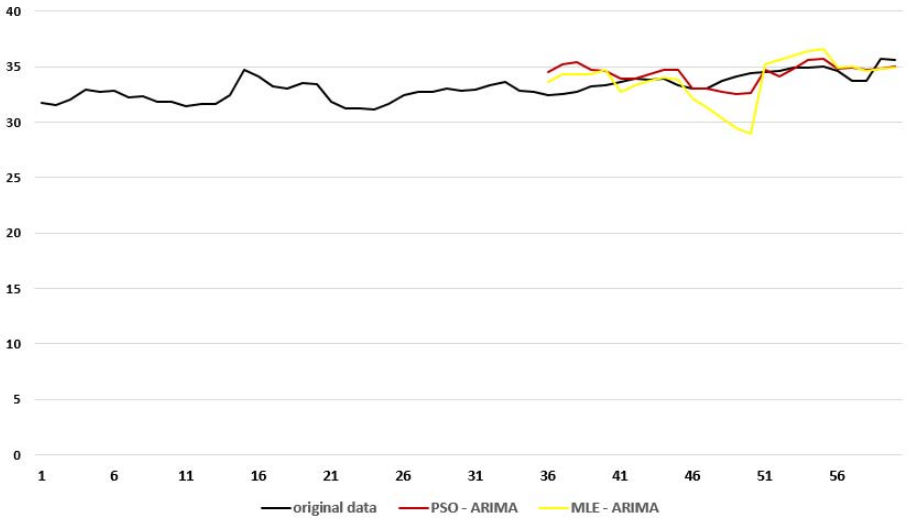

Fig. 3. Chart of prediction when size of the "learning phase" was 20

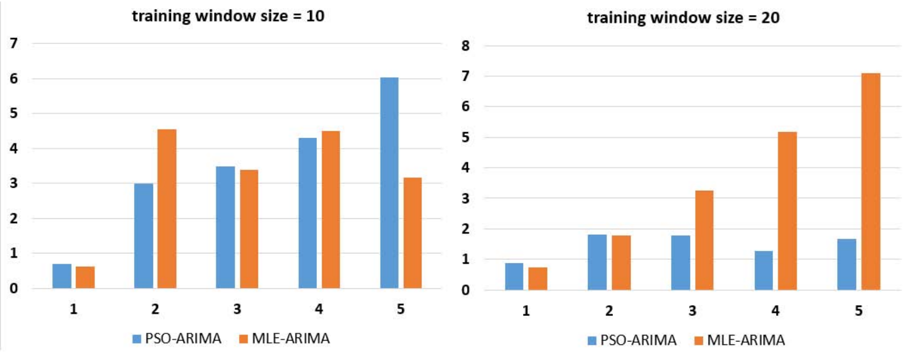

Fig. 4. Chart of the MSE on five predicted values during both tests

Finally, there are many other evolutionary algorithms like symbolic regression, differential evolution, neural networks to try to combine them with ARIMA.

Acknowledgment. The following grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, is partially supported by Grant of SGS No. SP2014/159, VB - Technical University of Ostrava, Czech Republic, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness.

References

1. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1976)
2. Ljung, G.M., Box, G.E.P.: The likelihood function of stationary autoregressive-moving average models. Biometrika 66(2), 265–270 (1979)
3. Morf, M., Sidhu, G.S., Kailath, T.: Some new algorithms for recursive estimation on constant linear discrete time systems. IEEE Transactions on Automatic Control 19(4), 315–323 (1974)
4. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
5. Weakliem, D.L.: A Critique of the Bayesian Information Criterion for Model Selection. Sociological Methods and Research 27(3), 359–397 (1999)
6. Tasy, R.S., Tiao, G.C.: Use of canonical analysis in time series model identification. Biometrika 72(2), 299–315 (1985)
7. Tasy, R.S., Tiao, G.C.: Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary ARMA model. Journal of the American Statistical Association 79(1), 84–96 (1984)
8. Hannan, E.J., Rissanen, J.: Recursive estimation of mixed autogressive-moving average order. Biometrika 69(1), 81–94 (1982)
9. Hannan, E.J., Quinn, B.G.: The determination of the order of an autoregression. Journal of the Royal Statistical Society B 41(2), 190–195 (1979)
10. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
11. Kinnear Jr., K.E. (ed.): Advances in Genetic Programming. MIT Press, Cambridge (1994)
12. Poli, R., Langdon, W.B.: Genetic Programming with One-Point Crossover
13. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks, Perth, Australia, pp. 1942–1948 (November 1995)
14. Kennedy, J.: The particle swarm: Social adaptation of knowledge. In: Proc. 1997 Int. Conf. Evolutionary Computation, Indianapolis, IN, pp. 303–308 (April 1997)
15. White, D.R.: Software review: the ECJ toolkit (March 2012)
16. Dickey, D.G.: Dickey-Fuller Tests. In: International Encyclopedia of Statistical Science, pp. 385–388 (2011)