Algebraic Level-Set Approach for the Segmentation of Financial Time Series

Rita Palivonaite(✉), Kristina Lukoseviciute, and Minvydas Ragulskis

Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-222, LT-51368 Kaunas, Lithuania
{rita.palivonaite,kristina.lukoseviciute,minvydas.ragulskis}@ktu.lt
http://www.personalas.ktu.lt/~mragul

Abstract. Adaptive algebraic level-set segmentation algorithm of financial time series is presented in this paper. The proposed algorithm is based on the algebraic one step-forward predictor with internal smoothing, which is used to identify a near optimal algebraic model. Particle swarm optimization algorithm is exploited for the detection of a base algebraic fragment of the time series. A combinatorial algorithm is used to detect intervals where predictions are lower than a predefined level. Moreover, the combinatorial algorithm does assess the simplicity of the identified near optimal algebraic model. Automatic adaptive identification of quasi-stationary segments can be employed for complex financial time series.

Keywords: Segmentation · Financial time series · Particle swarm optimization

1 Introduction

Financial time series prediction is a challenging task for researchers and practitioners in different fields of science and engineering. Many different techniques are used to analyze time series. Data mining community consider such major tasks as indexing, clustering, classification, segmentation, prediction and summarization [1]. Discovering information from massive data becomes a challenge, which leads to the necessity to present data in reduced form. The dimensionality reduction of the data is the first step to efficiently deal with data mining tasks [2]. Segmentation in time series analysis is often referred to as a dimensionality reduction algorithm. Most time series segmentation algorithms can be grouped into one of the following three categories: sliding windows [3], top-down [4], bottom-up [5]. Segments are usually used for representing financial time series. One frequently used segmentation method is Piecewise Linear Approximation (PLA) [6,7]. PLA has been applied for pattern matching [8] and predicting the trading points [9] in the stock market. In predicting stock movement, financial analysts not only consider the trend identified by the curve but also take into account certain points on the time series data. Segments extracted from financial

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2014, LNCS 8602, pp. 239–250, 2014.
DOI: 10.1007/978-3-662-45523-4_20

time series are widely used in trend analysis as well as in predicting future tendency of the price movement. Time series segmentation method based on turning points, which are extracted from the maximum or minimum points of the time series is proposed in [10]. This method generates segments at different levels of details and achieves satisfactory results in preserving higher number of trends compared to other segmentation approach.

A novel time series segmentation algorithm based on algebraic predictor with internal smoothing and an adaptive level-set method for the assessment of prediction errors is presented in this paper. The algebraic prediction method with internal smoothing (APIS) is used to forecast time series, build a model of the process and then use this model for the segmentation of financial time series [11].

2 Preliminaries

Time series prediction is a challenging task in many fields of economics and finance. Despite of plenty forecasting techniques, there is no single method outperforming all others in all situations. In this paper we will use an algebraic one point forward prediction technique. The main difference of the algebraic prediction technique first introduced in [12] and developed in [11] from other alternative time series predictors is in the fact that the algebraic predictor identifies the algebraic complexity of the time series by means of the Hankel rank (or H-rank) of a sequence. The identification of the H-rank will serve as key computational tool in the segmentation procedure of the analyzed time series.

The definition of the H-rank is presented in [12], but we will give a brief overview of the computational techniques used for the identification of H-ranks.

Let S be a sequence of real numbers:

$$S := (x_0, x_1, x_2, \ldots) \tag{1}$$

The Hankel matrix $H^{(n)}$ constructed from the elements of the sequence S is defined as follows:

$$H^{(n)} := \begin{vmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ x_1 & x_2 & \cdots & x_n \\ \cdots & \cdots & \cdots & \cdots \\ x_n & x_{n+1} & \cdots & x_{2n-2} \end{vmatrix} \tag{2}$$

where n denotes the order of the square matrix. The determinant of the Hankel matrix is denoted by $d^{(n)} = \det H^{(n)}$; $n \geq 1$. The H-rank of the sequence is such natural number $m = Hr\,(x_k; k \in Z_0)$ that satisfies the following condition[12]:

$$d^{m+k} = 0 \tag{3}$$

for all $k \in 1, 2, \ldots$, but $d^m \neq 0$.

Let us assume that the H-rank of the sequence is m, $m \leq +\infty$. Then the elements of the deterministic algebraic sequence S are expressed in the form[12]:

$$x_n = \sum_{k=1}^{r} \sum_{l=0}^{n_k-1} \mu_{kl} \binom{n}{l} \rho_k^{n-l};\ n = 0, 1, 2, \ldots \tag{4}$$

where the characteristic roots $\rho_k \in C$; $k = 1, 2, \ldots, r$ can be determined from the Hankel characteristic equation

$$\begin{vmatrix} x_0 & x_1 & \cdots & x_m \\ x_1 & x_2 & \cdots & x_{m+1} \\ & & \cdots & \\ x_{m-1} & x_m & \cdots & x_{2m-1} \\ 1 & \rho & \cdots & \rho^m \end{vmatrix} = 0; \tag{5}$$

the recurrence indexes of these roots n_k ($n_k \in N$) satisfy the equality $n_1 + n_2 + \ldots + n_r = m$; coefficients $\mu_{kl} \in C$; $k = 1, 2, \ldots, r$; $l = 0, 1, \ldots, n_k - 1$ can be determined from a system of linear algebraic equations which can be formed from the systems of equalities in Eq. (4), and this system has a unique solution [13].

The algebraic prediction technique in [12] exploits the concept of the H-rank and performs the extrapolation of the reconstructed algebraic model into the future. But a random sequence does not have a rank simply due to an inevitable contamination by noise. Thus special evolutionary computational strategies are developed in [12] for the identification of a closest skeleton algebraic sequence to the real-world time series, but due to variability of forecasted values the algebraic predictor in [12] is applicable for estimation of local minimums and maximums in day-ahead forecasting applications. In this paper for financial time series segmentation task we employ enhanced algebraic method with procedure of internal smoothing (APIS) [11]. Internal smoothing procedure enable reaching a healthy balance between excellent variability of skeleton algebraic sequences and valuable properties of predictors based on the moving averaging method which is widely used in financial time series primary analysis and forecasting.

The forecasting idea is based on the assumption that the sequence $\tilde{S}$ is produced by adding noise to some unknown algebraic sequence:

$$\tilde{S} := (x_0 - \varepsilon_0, x_1 - \varepsilon_1, x_2 - \varepsilon_2, \ldots) = (\tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \ldots) \tag{6}$$

We will try to indentify algebraic relationships in the available observation data and to smooth the forecast before the prediction is done. In order to remove inherent random variation we employ simple moving average smoothing technique:

$$\bar{x}_k = \frac{1}{s} \sum_{s=1}^{s-1} x_{k-i-1} \tag{7}$$

where $\bar{x}_k$ is a smoothed value at the moment k; s is the averaging window. The width of averaging window s should be preselected for each time series, though some general recommendations are given in [11]. It is common that for financial time series forecasting the best result is achieved with $s = 1$, it is so

called the naïve method [14]. In this paper according to these recommendations we set $s = 1$.

The schematic diagram of the algebraic prediction with internal smoothing process is illustrated in Fig. 1.

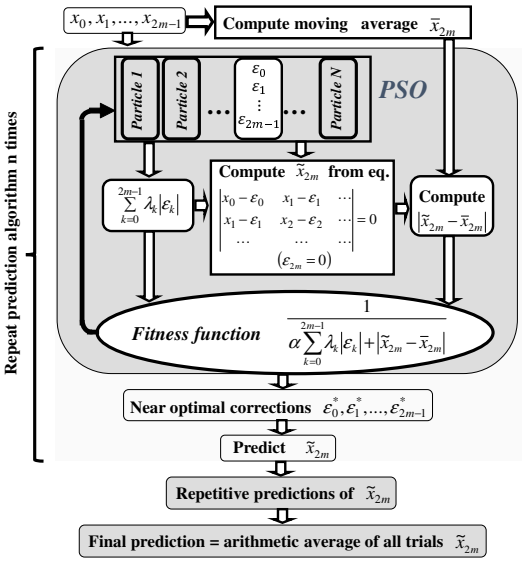

Fig. 1. The illustration of the algebraic prediction with internal smoothing (APIS) forecasting procedure

Let the H-rank of that unknown algebraic sequence is assumed to be equal to m; $2m$ observations x_0, x_1, x_2, …, x_{2m-1} are available for the building the algebraic model of the process; x_{2m-1} is the value of the observation at the present moment. The first task is to compute the moving average forecast value $\bar{x}_{2m}$ with parameter $s = 1$. Secondly, the set of corrections $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2m-1}\}$ must be identified before any algebraic predictions could be made.

Evolutionary algorithms will be used to identify the near-optimal set of corrections. Particle swarm optimization (PSO) techniques have been successfully employed for the identification of the skeleton algebraic sequence in [11]. We will also use PSO for the selection of a near-optimal set of corrections. Though the selection of the parameters of PSO remains mostly empirical and depends on the structure of the fitness function [15], we fix $w = 0.6$ and $c_1 = c_2 = 1.7$ as recommended by Trelea [16] (c_1 and c_2 are two positive constants, called acceleration constants, representing weightings of the stochastic acceleration terms that pull each particle toward the *particle's best* and the *global best*; w is the inertia weight balancing the global and the local search). Due to the indication that the effect of the population size on the performance of the PSO method is

of minimum significance [17] and most researchers use a swarm size of 10 to 60 [18], we set the swarm size for PSO to 30 particles.

As the black thick arrow in Fig. 1 illustrates, a new set of near-optimal corrections $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2m-1}\}$ is generated every time when the PSO algorithm is executed. The next step is to determine the element $\tilde{x}_{2m}$ from the following equality based on the eq. (3), when the H-rank is assumed to be m:

$$\begin{vmatrix} x_0 - \varepsilon_0 & x_1 - \varepsilon_1 & \cdots & x_m - \varepsilon_m \\ x_1 - \varepsilon_1 & x_2 - \varepsilon_2 & \cdots & x_{m+1} - \varepsilon_{m+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_m - \varepsilon_m & x_{m+1} - \varepsilon_{m+1} & \cdots & \tilde{x}_{2m} \end{vmatrix} = 0; \tag{8}$$

The goal of selecting the set of corrections $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2m-1}\}$ is to minimize any distortions from original time series. Therefore, the fitness function for the set of corrections $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2m-1}\}$ has to be maximized [11]:

$$F(\varepsilon_0, , \varepsilon_1, \ldots, \varepsilon_{2m-1}) = \frac{1}{\alpha \sum_{k=0}^{2m-1} \lambda_k |\varepsilon_k| + |\tilde{x}_{2m} - \bar{x}_{2m}|}; \ \alpha > 0; \tag{9}$$

where α is the penalty proportion between the sum of weighted corrections and the difference of forecasts based on skeleton algebraic sequences and moving averages; coefficients λ_k determine the tolerance corridor for the corrections (all corrections would be the same if $\lambda_k = 1/(2m)$; $k = 0, 1, \ldots, 2m-1$) [12].

3 The Construction of the Segmentation Algorithm

3.1 Time Series Prediction Procedure

As mentioned previously, we will use the time series prediction algorithm based on algebraic prediction with internal smoothing (APIS) [11] for the segmentation of the time series. But instead of trying to identify the most appropriate H-rank of the time series at the beginning of the prediction process, we will perform the prediction at different preset values of the H-rank.

In general, the selection of the effective range of H-ranks can be free, though too wide range of H-ranks would raise the computational costs required by the proposed technique. We preselect $2 \leq Hr \leq 10$ for the artificial time series with additive noise. We started with $Hr = 2$ for an elementary arithmetic progression, which is quite acceptable for financial time series segmentation task. On the other hand, the length of the vector of corrections $\{\varepsilon_k\}$ is equal to 20 already at $Hr = 10$ (what raises computational costs of the prediction algorithm and increases the complexity of the identified algebraic model).

The prediction algorithm extrapolates the skeleton sequence by one element into the future: $\tilde{x}_{2m}$ is the algebraic prediction of the sequence $(x_0, x_1, \ldots, x_{2m-1})$ (Fig. 1). Next, we shift the observation window by one element and predict $\tilde{x}_{2m+1}$. The process is continued until the last element of the original data sequence is predicted.

The next step is the selection of the tolerable error level L for the algebraic prediction of the analyzed time series. The basic idea of the proposed technique is straightforward: the preselected algebraic model is sufficiently accurate if extrapolation errors of the prediction are lower than predefined error level L. All continuous time series prediction intervals with extrapolation errors lower than the level L are considered as the segments. Recommendations for selection of tolerable error level for time series segmentation are developed in [19].

3.2 Combinatorial Aspects of the Segmentation Algorithm

The proposed segmentation algorithm is based on two important concepts:

a) The algorithm must automatically identify the longest time interval where the current algebraic model does not produce forecasting errors higher than the predetermined level L;

b) The segmentation algorithm must also evaluate the simplicity of algebraic model in each identified segment.

In other words, the segmentation algorithm should find a conciliation between two extremities which are graphically represented in Fig. 2. The x-axis represents the order of the algebraic model (H-rank), the vertical axis illustrates the adaptive preference of the segmentation algorithm.

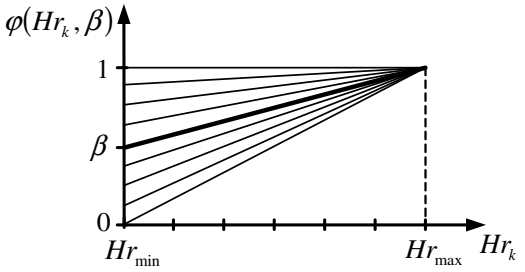

Fig. 2. The schematic diagram of the preference function $\varphi\,(Hr_k, \beta)$ illustrated by the thick solid line going through points (Hr_{min};β) and (Hr_{max};1)

The preference function $\varphi\,(Hr_k, \beta)$ is defined as linear function with parameter β, $0 \leq \beta \leq 1$:

$$\varphi\,(Hr_k, \beta) = \beta + \frac{Hr_k - Hr_{min}}{Hr_{max} - Hr_{min}}\,(1 - \beta) \tag{10}$$

The preference function $\varphi\,(Hr_k, \beta)$ does not assess the simplicity of the algebraic model at $\beta = 1$. On the contrary, the preference function $\varphi\,(Hr_k, \beta)$ gives the highest priority to the simplest algebraic model at $\beta = 0$. We will use the arithmetic average between these two cases by set at $\beta = 0.5$ (Fig. 2). Then

the segmentation procedure can be illustrated by the following example. The longest segment $l_n^{(i)}$ is identified in the first step of the algorithm – note that before taking the decision which segment is given a highest priority we divide the length of each segment by the value of the preference function $\varphi(Hr_k, \beta)$:

$$l_k^{(i)} = \frac{l_k^{(i)}}{\varphi(Hr_k, \beta)} \tag{11}$$

where $l_n^{(i)}$ is the length of the segment, Hr_k – the k-th H-rank, i – an iteration of combinatorial algorithm.

Thus a shorter segment can be given a higher priority than a longer segment if only the algebraic model of the shorter segment is sufficiently simpler compared to the algebraic model of the longer model.

A schematic diagram of combinatorial segmentation algorithm for the identification of longest continuous intervals of successful predictions in the effective range of H-ranks is illustrated in Fig. 3.

The main idea of combinatorial segmentation algorithm is characterized by this illustrative example.

Step 1. Set the level L and perform the algebraic forecasting of the given data at different preselected H-ranks. Mark intervals of the time series where forecasting errors are lower than L. Such marking is schematically illustrated in Fig. 3 (a). Horizontal lines denote continuous intervals for each discrete H-rank (the vertical axis stands for the H-rank). The length of intervals is denoted as $l_k^{(i)}$, where the index k stands for the k-th H-rank and i is the iteration of the segmentation process (at the beginning of the process it is set to zero). As we introduced previously the evaluation of the length of the interval depends on the H-rank: the smaller is the H-rank – the prior is the interval of this H-rank. In this schematic example we set the parameter $\beta = 0.5$ of function $\varphi(Hr_k, \beta)$ (eq. 10). It means that the interval of the lowest H-rank is twice important as the interval of the highest H-rank.

Step 2. Identify the longest continuous interval. Though in our schematic example the interval $(t_2; t_7)$ of the highest H-rank is the longest with length $l_5^{(0)}$, but after evaluation of all segments by the preference function $\varphi(Hr_k, \beta)$, the longest picked interval is $l_1^{(0)}$ (with the lowest H-rank). We considered that the length $l_1^{(0)} = 2 \cdot l_5^{(0)}$ with $\beta = 0.5$. We marked the longest evaluated continuous interval by gray shaded box in Fig. 3 (a).

Step 3. Denote the marked interval as the segment associated to the according H-rank; erase all the information about the other H-ranks above and below the marked segment. The selected segment $(t_0; t_3)$ is marked by a thick solid horizontal line in Fig. 3 (b).

Step 4. With preference function $\varphi(Hr_k, \beta)$ evaluate the longest continuous interval in the zones not occupied by the marked segments (return to the Step 2). Though the longest interval is with the highest H-rank $(t_3; t_7)$ (note that it was truncated after the Step 2), but after evaluation by function $\varphi(Hr_k, \beta)$ it is not considered as the segment. Besides, though the lengths of $l_2^{(1)}$ and $l_3^{(1)}$

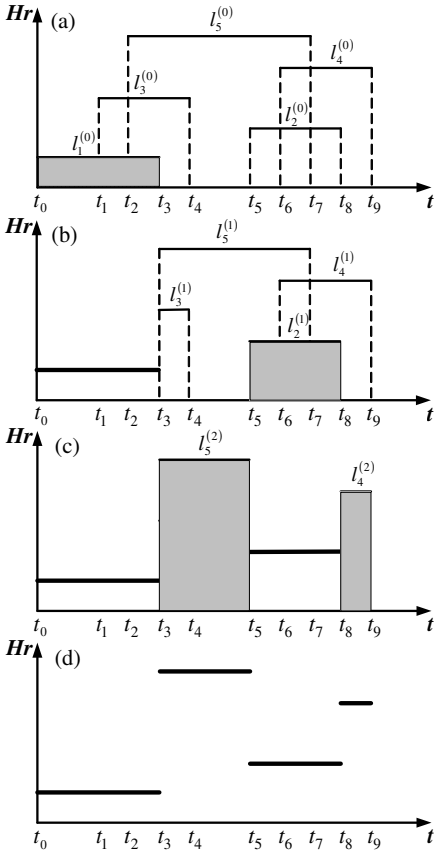

Fig. 3. The illustration of the combinatorial segmentation algorithm. Horizontal lines in part (a) show intervals where algebraic prediction errors are lower than the pre-set level L (the height of a line stands for the appropriate H-rank). The length of the longest weighted interval above the level L for all H-ranks is denoted as $l_k^{(i)}$, where indexes k indicate the k-th H-rank Hr and i indicates the iteration of combinatorial segmentation algorithm. The parameter β is set to 0.5. The gray-shaded area in part (a) illustrates the longest preferred continuous line interval which is associated to a separate segment in part (b). The process is continued through parts (b - d) until the whole sequence is split into separate segments. Thick solid lines represent the result of the segmentation algorithm.

are the same size, the interval $(t_5; t_8)$ is selected due to the lowest H-rank. It is marked as gray shaded box.

Step 5. Erase all the information below and above the selected segment and evaluate the rest segments. At the final step we select two not overlapping intervals $(t_3; t_5)$ and $(t_8; t_9)$ as the last segments.

Finally, the segmentation algorithm identifies four distinct segments $(t_0; t_3)$, $(t_3; t_5)$, $(t_5; t_8)$ and $(t_8; t_9)$ (Fig. 3 (d)). Note that in the real world time series

it might be that in some intervals any particular segment could not be selected due to high forecasting errors exceeding the predefined level L.

4 Computational Experiments with Financial Time Series

We test the functionality of the proposed segmentation algorithm using real-world finance time series. We select a standard STLFSI (St. Louis Fed Financial Stress Index) time series describing 230 consecutive measures of the degree of financial stress in the markets (we selected mounthly data range from 1993-Dec-01 to 2013-Mar-31) [20]. Financial STFSI time series is constructed from 18 weekly data series: seven interest rate series, six yield spreads and five other indicators. Each of these variables captures some aspect of financial stress. Accordingly, as the level of financial stress in the economy changes, the data series are likely to move together. Note that vertical axis is transformed to interval $[0; 1]$. Due to specific properties of APIS forecasting method, this transformation ensures lower time series prediction errors[11]. We set parameter $\beta = 0.5$ and the tolerable error level $L = 0.05$ (we consider that algebraic model is sufficiently accurate if extrapolation errors of the prediction are lower than 5 % length of all range of time series data values). The segmentation result of STLFSI time series is presented in Fig. 4. The PSO algorithm is iterated 30 times and the averaged result of APIS is presented.

Our segmentation method has singled out the following mounthly intervals: 1994-May-31 – 1998-Mar-31, 1998-Jul-31 – 2001-Apr-30, 2001-May-31 – 2002-Jan-31, 2002-Mar-31 – 2007-Mar-31, 2008-Nov-30 – 2009-Dec-31, 2010-Jan-31 – 2011-Mar-31, 2011-Apr-30 - 2013-Jan-31.

A comparative assessment of the functionality of the proposed technique with other typical segmentation methods is required in order to understand if our methodology does outperform other methods or not. One of the most commonly used representations is piecewise linear approximation. In the context of data mining, it supports change point detection.

The first comparison is performed with sliding-windows segmentation method. This method is based on growing, usually linear, segment until it exceeds some user-specified criteria. The process repeats with the next data point [3]. Due to comparison we have chosen the same tolerable error level $L = 0.05$. The results of the change point detection are presented in Fig. 4(c). It is clear that linear approximation is not as precise as APIS and approximation errors exceed the preset tolerable error level L more common. Naturally, this leads to the higher number of segments. Such approximation becomes greatly over-fragmented for real-world datasets with inevitable additive noise [5].

The second comparison is performed with the bottom-up segmentation method. Starting from the finest approximation, segments are merged until some stopping criterion is met [5]. In our case, the stopping criterion is the maximum error per segment exceeding the preset error level $L = 0.05$. The segmentation results are presented in Fig. 4(d). The bottom-up algorithm is the natural complement to the

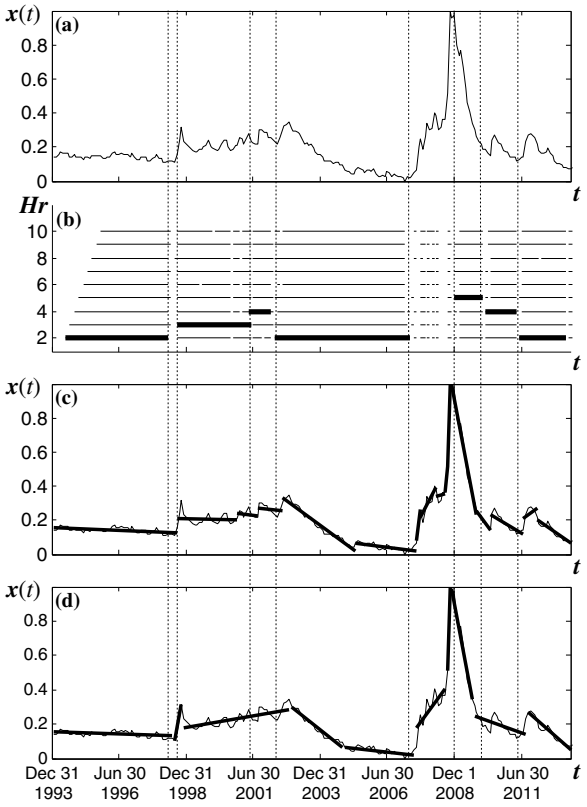

Fig. 4. The segmentation of STLFSI time series. The time series is illustrated in part (a); The result of the segmentation with APIS at tolerable error level $L = 0.05$ is presented in part (b). The result of the segmentation with sliding-windows method is presented in part (c); The result of the segmentation with bottom-up method is presented in part (d).

top-down algorithm. Top-down: the time series is recursively partitioned until some stopping criterion is met. Empirical comparison of the major segmentation algorithms on a very diverse collection of datasets does show that top-down and bottom-up algorithms produce similar results [5] - and thus (due to the space limitations) the results of top-down algorithm are excluded.

Generally, we are able to locate different algebraic relationships while other segmentation methods are approximated only by one type of function, usually by a low order polynomial.

5 Conclusions

Any time series segmentation algorithm must comply with two major requirements: the time series must be approximated by the simplest possible mathematical model

(the model identification is a task of the data mining process) and finding change points, which are used as markers between appropriate segments. The main advantage of our proposed method is based on the fact that we do identify algebraic models of the process – but the order of this algebraic model is detected completely automatically. In other words, one does not have to analyze the time series employing different methods and techniques.

The proposed segmentation algorithm reveals that the hidden structure of the time series is able to identify potential changes in the evolution of the process and exploits predictability as a tool for the characterization of complexity. Such predactability can be directly used for the decision-making analysis in financial time series analysis.

Acknowledgments. This research was funded by a grant (No. MIP-100/2012) from the Research Council of Lithuania.

References

1. Esling, P., Agon, C.: Time-series data mining. ACM Computing Surveys (CSUR) **45**(1), 12 (2012)
2. Ralanamahatana, C.A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M., Das, G.: Mining Time Series Data, Data Mining and Knowledge Discovery Handbook. pp. 1069–1103 Springer, (2005)
3. Qu, Y., Wang, C., Wang, S.: Supporting fast search in time series for movement patterns in multiples scales. In: Proceedings of the 7th International Conference on Information and Knowledge Management, CIKM 1998, pp. 251–258 (1998)
4. Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., Allan, J.: Mining of Concurent Text and Time Series. In: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, pp. 37–44 (2000)
5. Keogh, E.J., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: a survey and novel approach. Data Mining in Time Series Databasis 57, 1–22 (2003)
6. Keogh, E.J., Pazzani, M.J.: Relevance feedback retrieval of time series data. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, United States, pp. 183–190 (1999)
7. Li, C.S., Yu, P.S., Castelli, V.: MALM: a framework for mining sequence database at multiple abstraction levels. In: Proceedings of the 7th International Conference on Information and Knowledge Management, Bethesda, Maryland, United States, pp. 267–272 (1998)
8. Zhang, Z., Jiang, J., Liu, X., Lau, W.C., Wang, H., Wang, S.-S., Song, X., Xu, D.: Pattern Recognition in Stock Data Based on a New Segmentation Algorithm. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 520–525. Springer, Heidelberg (2007)
9. Chang, P.C., Fan, C.Y., Liu, C.H.: Integrating a piecewise linear representation method and a neural network model for stock trading points prediction. IEEE Transactions on Systems, Man, and Cybernetics Part C 39(1), 80–92 (2009)
10. Yin, J., Si, Y.W., Gong, Z.: Financial Time Series segmentation Based On Turning Points. In: Proceedings of 2011 International Conference on System Science and Engineering, Macau, China, pp. 394–399 (2011)

11. Palivonaite, R., Ragulskis, M.: Short-term time series algebraic forecasting with internal smoothing. Neurocomputing (article in press) (2014)
12. Ragulskis, M., Lukoseviciute, K., Navickas, Z., Palivonaite, R.: Short-term time series forecasting based on the identification of skeleton algebraic sequences. Neurocomputing **74**, 1735–1747 (2011)
13. Navickas, Z., Bikulciene, L.: Expressions of solutions of ordinary differential equations by standard functions. Mathematical Modeling and Analysis **11**, 399–412 (2006)
14. Lacina, M., Lee, B.B., Xu, R.Z.: An evaluation of financial analysts and naïve methods in forecasting long-term earnings. Advances in Business and Management Forecasting 8, 77–101 (2011)
15. Eberhart, R.C., Kennedy, J.: Particle swarm optimization: developments, applications and resources. In: Proceedings of IEEE Congress on Evolutionary Computation, Seoul, Korea. pp. 81–86. IEEE Service Center, Piscataway, (2000)
16. Trelea, I.: C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters **85**(6), 317–325 (2003)
17. Shi, Y.H., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, CA, pp. 591–600. Springer, New York (1998)
18. Kennedy, J., Eberhart, R.C., Shi, Y.H.: Swarm Intelligence. Morgan Kaufman (2001)
19. Palivonaite, R., Lukoseviciute, K., Ragulskis, M.: Algebraic segmentation of short nonstationary time series based on evolutionary prediction algorithms. Neurocomputing **121**, 1354–364 (2013)
20. St. Louis Fed Financial Stress Index News Releases (2013) (accessed November 1, 2013). http://www.stlouisfed.org/newsroom/financial-stress-index