BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 6
Special issue with selection of extended papers from 6th International Conference on Logistic, Informatics and Service Science LISS'2016

Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0080

Determination of the Starting Point in Time Series for Trend Detection Based on Overlapping Trend

Gao Xuedong, Gu Kan

Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing, China
Emails: gaoxuedong@manage.ustb.edu.cn gukan@xs.ustb.edu.cn

***Abstract*: *The traditional time series studies consider the time series as a whole while carrying on the trend detection; therefore not enough attention is paid to the stage characteristic. On the other hand, the piecewise linear fitting type methods for trend detection are lacking consideration of the possibility that the same node belongs to multiple trends. The above two methods are affected by the start position of the sequence. In this paper, the concept of overlapping trend is proposed, and the definition of milestone nodes is given on its base; these way not only the recognition of overlapping trend is realized, but also the negative influence of the starting point of sequence is effectively reduced. The experimental results show that the computational accuracy is not affected by the improved algorithm and the time cost is greatly reduced when dealing with the processing tasks on dynamic growing data sequence.*

***Keywords*: *Overlapping trend, milestone nodes, trend detection, calculate cost.*

1. Introduction

A typical task of time-series data analysing is to obtain trend information from it. Classical statistical methods can obtain coarse-grained trend from a complete time series by decomposition. The complexity of calculation is high because of the high dimensions, noise and complexity of time series data. Therefore, it is customary to cut the time series data into some sub-sequences as trend primitives mostly like the one proposed in [1]. Fragment analysis based on the primitives would be taken then. This idea and the characteristics of time series representation are very close.

The time series feature representation method can reduce the dimension of time series and denoise it. There are plenty of time series feature representation methods such as piecewise aggregate approximation, domain transform, piecewise

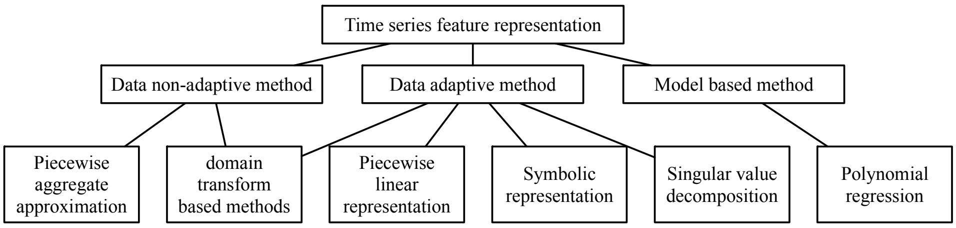

Time series feature representation
Data non-adaptive method
Data adaptive method
Model based method
Piecewise aggregate approximation
domain transform based methods
Piecewise linear representation
Symbolic representation
Singular value decomposition
Polynomial regression

Fig. 1. Time series feature representation methods

Piecewise aggregate approximation methods split the sequence into equal length fragments, and construct a new sequence with the average value of each subsequence, the example is the algorithm in [3]. Two most typical domain transform methods are discrete Fourier transform in [4] and discrete wavelet transform as the algorithm in [5]. Piecewise linear representation methods are the simplest and easiest to use. They linear fit the sequence piecewise based on some strategies, for example, sliding window was chosen in [6]. Symbolic representation methods transform the sequence into a string sequence consisted of basic units predefined, the most familiar one of which is some algorithms based on primitives proposed in [1]. Singular value decomposition reduces the dimension of the sequence with the help of principal component analysis as in [7]. Polynomial regression methods assume that the series data come from a particular mathematical model; hence a polynomial regression can be used to represent the data. Polynomial regression is very intuitive, and it is easy to piecewise represent a time series data with a set of equations, as the work in [8]. Whichever of the methods finally comes to the idea of segmenting a sequence into sub-sequences and then processing further transaction? The idea is simple, easy to use and suitable for processing any kind of time series data. However, it is difficult to discover overlapping trends and verify the correctness of the result when detection trends from series data.

Li and Zheng [8] proposed a data representation method based on PPR to segment time series data. Fuchs et al. [9] used the least squares method combined with orthogonal polynomials to fit the time series. Xuedong and Kan [10] redefine the concept of trend in sequence data and point out that sequences with inertia only can be called trend, and then extract the trend by means of piecewise polynomial representation and inertia test. The process of inertial test is shown in Fig. 4a.

The dimensionality of the Vandermonde matrix use in [10] increases rapidly with inertia testing process, which makes it complex for calculating. An assumption is proposed to solve the problem, that is, if a node passed inertia test, it is believed that the node has no influence to the general development of the sequence and can be represented with all the current fitting nodes. According to the assumption, a line escapement Vandermonde matrix would be used to fitting movement function when the former one is proven to be out of place. The new Vandermonde matrix is

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_k \\ \vdots \\ p_j \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 & \dots & 1^{k-1} \\ 1 & 2 & 2^2 & 2^3 & \dots & 2^{k-1} \\ 1 & 3 & 3^2 & 3^3 & \dots & 3^{k-1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & k & k^2 & k^3 & \dots & k^{k-1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & j & j^2 & j^3 & \dots & j^{k-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{k-1} \\ \vdots \\ a_{j-1} \end{bmatrix}, \tag{1}$$

where the number from k to j is discontinuous.

All of the above methods can identify trend information from sequence to some extent, and provide theoretical guidance for some practical requirements. For example, some scholars use time series analysis method to analyse the port cargo throughput, like [11], and some others try to use time analysis method to establish a global logistics system, as in [12]. However, the results of these methods are mostly dependent on the starting node of the sequence, and are one-time calculations; it is difficult to deal with the growing sequence.

In this paper, the trend detection algorithm based on inertia test is improved; the existence of overlapping trend is defined. The definition of milestone node is also given, which solves the problem that the trend detection starting point problem and reduce the calculate cost when new nodes are added to the sequence.

2. Trend detection starting point problem

2.1. Emergence of trend detection starting point problem

Any sequence of variables in chronological order can be called a time series according to the definition in [13]. In condition of dealing with practical problems, the limited observation samples of time series are obtained by observing and recording, so as to carry out various kinds of analysis work.

Traditional statistical methods, such as [13], define the trend as a real-valued function of time t over the entire sequence. A time-series data is considered as a whole, so naturally the first data node in the series is assumed to be the starting point of the sequence. At this time, in order to ensure the accuracy and persuasiveness of the analysis results, any analysis carried out in fact implied the next

Hypothesis. The data before the current initial node have no effect for the time series trend detection results or its influence can be expressed by the analysis results of the post-nodes in sequence and ignored.

This presupposes is necessary and unambiguously useful for time series analysis work to be carried out over a sufficiently long time series or for some study on clearly defined time period, that can help the researcher focus on what is needed concerned about without the interference of unnecessary information.

So, the initial node of the time series data and the real starting point in actual may be different for these following reasons:

- It is not possible to obtain all real data nodes due to technical capability or historical reasons;
- Analysts specialize in the analysis of data for a specific time period. True starting point is not needed;
- There are too many historical data; analysts only need the most recent.

2.2. Influence on the trend detection result

In order to deal with the high dimensionality and sparsity, time series data dimensionality reduction and feature discovery are often used. These methods will produce a new sequence for the approximate replacement of the original time series. The basic idea is to retain the main form of time series, ignore the small details to achieve compression. One of the most typical algorithms proposed in [14] is the polynomial fitting which uses a window of a certain length to segment the sequence data, or obtains several sub-sequences according to a number of key points in the sequence and then fitted.

AirPassengers data set in *R* is the classic Box & Jenkins airline data contains 144 nodes, which records monthly totals of international airline passenger from 1949 to 1960. When step is 1 and error threshold is 0.01, the calculation results of algorithm proposed in [10] can be approximated as a piecewise linear representation of the original sequence and can therefore be used to demonstrate the effect of the different starting points of the sequence. The results are shown in Fig. 2, in which the shadow parts represent for trends.

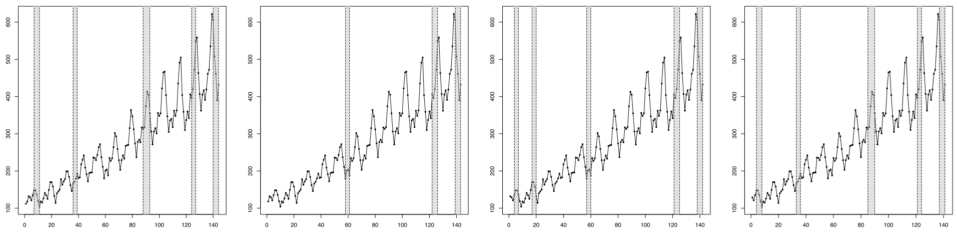


Fig. 2. Influence of different start nodes (1-4) for trend detection in AirPassengers

As is shown in the figure, difference in starting nodes not only causes different number of trends but also leads to different position in sequence where trends are detected. All these trends satisfy the definition of a trend thus a fixed start node cannot make all trends be found.

Consider the case of sequence data prediction, if the sequence data is long enough, the impact of the different beginning node of the series will be reduced, but the short ones are not so lucky. The BJsale dataset is a time-series data about sales. The last 55 nodes in the sequence are used to construct the sub-sequences, and then the first three nodes of the subsequence are used as three different starting points for the algorithm to perform from. The classical ARIMA algorithm is used to predict the 24 node in future. The result is shown in Fig. 3.

It can be seen from Fig. 3 that different starting points do have some effect on the predicted results in a short time series.

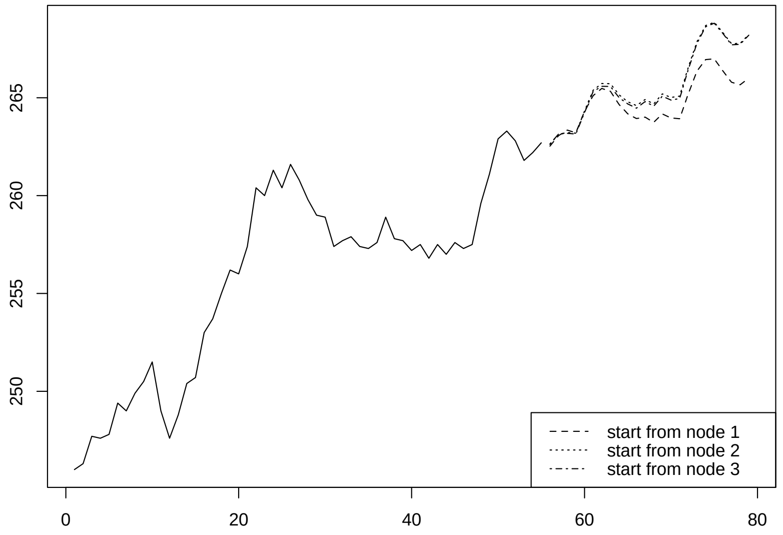


Fig. 3. Influence of different start nodes (1-3) for prediction in BJsales

2.3. Influence on the computational complexity

For a time-series with a fixed length, trend identification operations will cost fixed time. However, when the length of the sequence is not fixed and the nodes are fewer, the time-consuming for calculation is less.

Traditional time series analysis treats the sequence as a static data object, and the work is carried out in a scope of clear start and end time. In recent years, a number of work, such as streaming data mining firstly proposed in [15], is to find a new form of time series data, where the length of time series will continue to grow with the passage of time.

Traditional time series analysis method is difficult to analyse the dynamic growing time series. Once the new data is added into the sequence, it needs to traverse the sequence from the beginning, resulting in a large computational cost. As to the streaming data mining methods, little research for the trend type knowledge is proposed.

2.4. Overlapping trend

Xuedong and Kan [10] re-define the trend in their paper. Taking the definition of the trend in that paper, combined with the impact analysis of the starting point of the trend, a node belongs to a trend still has possibility to lead to another new trend sequence.

This situation, referred to as the overlap of trends, corresponds to a number of trends that called overlapping trend. Specific definitions is

Definition. Overlapping trend. For two trend sequences $t_1 = \langle p_a, p_b, \ldots, p_f, p_g, \ldots, p_k\rangle$ and $t_2 = \langle p_f, p_g, \ldots, p_k, \ldots, p_j\rangle$ in a time series data $p = \langle p_1, p_2, \ldots, p_n\rangle$, if $\exists p_i$, makes $p_i \in t_1$ and $p_i \in t_2$ at the same time, then $o_t = \langle p_f, p_g, \ldots, p_i, \ldots, p_k\rangle$ is called *an overlapping trend.*

Overlapping trend is ubiquitous. When a trend comes to its end, a period of adjustment or another trend is both possible to be. In case of a new trend, there is a high possibility that it prepared to occur when the previous one is still going on. The most extreme case is one trend contains another, that is, all nodes make up a trend is in another longer trend. In mathematical, overlapping trend can be described as a range in a polynomial can fit out another lower order polynomial.

The traditional studies cannot detect overlapping trends directly, much less deal with them.

3. Improvement on Trend Detection Algorithm

3.1. Shortcomings of the Original Algorithm and the improvements

When a node does not pass the inertia test, there are two cases, one is the inertia has disappeared from the current sequence, a movement comes to its end in the form of a trend or an adjustment; another is the current movement function is not enough to describe the current trend, a new movement function is needed. At this point, the original fitting nodes have been unable to completely express the inertia nodes.

Therefore, Trend Detection Algorithm based on inertia test can quickly and effectively identify the trend, but there are some shortcomings, specifically, include the following:

Step 1. Using Vandermonde matrix directly for movement function fitting, which causes high computational burden, some nodes cannot be taken into consideration when fitting movement functions.

Step 2. A new movement starts next to the previous one, and lose sight of overlapping trends.

Step 3. The algorithm must take all nodes in the sequence into consideration; it is difficult to deal with the growing number of data nodes.

Step 4. The result may be affected because of the difference of starting point.

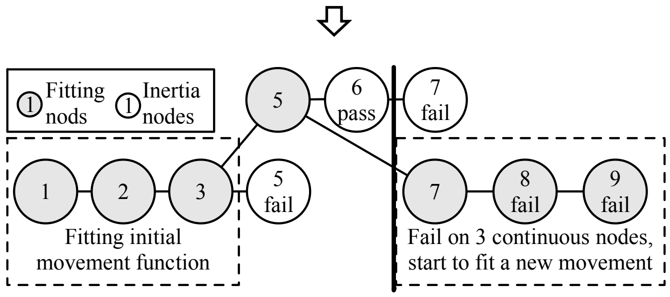
In this paper, these deficiencies are improved.

To fit a k-th-order polynomial in a sequence P contains n nodes, in particular, if $k = n - 1$, the rank of Vandermonde matrix is n, and the only least-square solution is $A = X^{-1}P$, which is the same as Vandermonde matrix solution. Besides the same result, least square equation set can be solved by iteration which leads to a smaller memory overhead. So this paper uses least square method to replace Vandermonde matrix method but leave the selection strategy for the order of polynomials. At this point, the first shortcoming is no longer a problem.

For the shortcoming of overlapping trend detection, we introduce sliding window to solve it. A new round of fitting starts from the node next the first node in previous movement with a fixed-width window, for example, three nodes as in [10], and no upper limit. The fitting finishes when inertia test fails for three times.

The remaining two issues will be discussed and solve in the next sections.

The improved inertia test process is shown in Fig. 4b.


1 Fitting nods
1 Inertia nodes
5
6 pass
7 fail
1
2
3
5 fail
7
8 fail
9 fail
Fitting initial movement function
Fail on 3 continuous nodes, start to fit a new movement

(b) Improved inertia test process

Fig. 4. Fitting node selection and inertia test

And here come the basic steps.

Begin

Input: Sequence, error threshold ε

Output: Set of trends

Step 1. $p_i = p_1$;

Step 2. Do fit movement function and inertia test on $\langle p_i, p_{i+1}, ..., p_n\rangle$, save the result into a list as *w*;

Step 3. $i = n$? If yes, go to Step 4, else make $i = i+1$ and go to Step 2;

Step 4. Remove duplicate values in *w*;

Step 5. Merge overlapping trends and output the result.

End

To merge overlapping trends, the following two rules need to be followed:

1. If one trend contains another, remove the shorter one.
2. If some but not all of the nodes in a trend is in another trend, keep both of them.

3.2. Starting point problem and dynamic computing problem solving

The difference of the starting point of the time-series will affect the analysis result only when the length of the sequence is short. However, if the length is too long, it will lead to recalculation of the original sequence every time a new node joins, resulting in high computational cost. After the introduction of the overlapping trend, due to the idea of near enumeration, considering the full possibilities of the existing data, the influence of the different starting points of the sequence will be compressed to the minimum, but the burden of calculation will not fall, which is very unfavourable for the increasing sequence processing.

However, in the case of overlapping trends, the choice of starting point for trend recognition becomes less important in the alternative. When a complete trend recognition operation has been done on a time series, the original trend information contained in can be considered been totally extracted. At this point, if a new node joins, the trend detection operation will be done since the last node of the original sequence with little information loss.

Nonetheless, we still hope to minimize this loss of information, thus presenting the concept of milestone nodes for sequence data trend detection.

Definition. Milestone nodes. There is a node in a time series, and the trend detection result on nodes after it is not influenced by the nodes before it. The particular node is called a *milestone node in the time series*.

Thus, the first node in sequence when doing traditional trend detection operation is a typical milestone node in theory, and the hypothesis in traditional study is able to be interpreted. In order to solve the problem of identifying new data nodes, we only need to find the last milestone node in the original sequence data, and start a new trend detection operation from this node. The number of data nodes that need re-fitting calculation will be reduced, greatly enhancing the computational efficiency.

When applied to the actual operation, the number of trends corresponding to each node is recorded with the process of algorithm. When the trend detection operation is completed, a reverse traversal will begin from the last node. The node which is nearest to the end of the sequence and corresponding minimum number of trends is the required milestone node.

In terms of node location and the number of corresponding trends, priority is given to the number of corresponding trends, i.e., the nodes closest to the end of the sequence that do not belong to any one trend or the starting point (end point) of a trend is preferred.

Therefore, if we want to use the algorithm to solve dynamically increase sequence problems, there are three steps in fact:

Step 1. Detect trend from a time series by trend detection algorithm based on inertia test.

Step 2. Extract the last milestone node in the sequence according.

Step 3. Start a new loop of trend fitting from the particular milestone node when new nodes join the sequence.

3.3. A simple example

A simple example is given to help understand the algorithm. First a time-series data contains 14 nodes segmented from the classical data set BJsales is given:

$P = \langle 198.6, 200.0, 200.3, 201.2, 201.6, 201.5, 201.5, 203.5, 204.9, 207.1, 210.5, 210.5, 209.8, 208.8 \rangle$.

Apply the algorithm improved in this paper and the result is shown in Fig. 5.

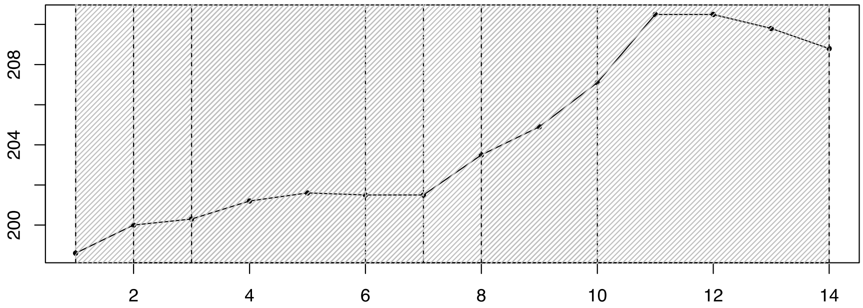


Fig. 5. Trend detection result on example data

A total of four trends were identified, namely: [p_1: p_6], [p_2: p_7], [p_3: p_{10}], [p_8: p_{14}], and the coefficients of the four equations are, respectively:

$$\begin{bmatrix} 197.400000 & 1.376587 & -0.097619 & -0.002778 & 0.000000 \\ 199.53333 & 0.28228 & 0.14484 & -0.02315 & 0.000000 \\ 196.80357 & 5.16297 & -2.00095 & 0.30922 & -0.01496 \\ 208.2143 & -9.1289 & 5.3833 & -0.9682 & 0.0553 \end{bmatrix}.$$

The whole sequence is completely covered by trends. And there are two 4th order polynomials and two 3rd order polynomials among them. Specifically, the nodes from the 3rd node to the 6th node belong to three trends at the same time, which is a typical phenomenon of overlapping trend.

As to the milestone node, the 8th node and the 10th node are two of the best candidates, are both belong to two trends. But because of the 10th node is closer to the end of the sequence, we use it as the chosen one for next process of trend detection.

4. Experiments

In order to examine the validity and practicability of the improved algorithm, we use *R* to program it and take some experiments on the program. The data sets used are all chosen from native data sets in *R*.

4.1. Detection effect

The first thing to be checked is the detection effect of the improved algorithm compared to the original one.

In [10], eight data sets are used to check the algorithm and good results are gotten. This experiment selects five of them which are lack of trends and adds another five to get 10 data sets. The experiment results are shown in Table 1.

Two data sets got error while detecting trends using the original algorithm because of the high memory cost. The results in table show that the improved algorithm detects more trends. The reason is obviously the overlapping trends. The highest order of the movement functions changes little, which means a pretty high-order polynomial is not necessary for representing a movement. One exception is the BJsales, on which the number of trends rises significantly and the highest order has a great variation, too. A further analysis is taken.

Table 1. The experiment results (step=1)

No	Data set	Sequence length	Trends amount		Highest order	
			Original	Improved	Original	Improved
1	uspop	19	3	3	4	5
2	USAccDeaths	72	3	5	3	3
3	LakeHuron	98	NA	5	NA	18
4	WWWusage	100	7	24	4	6
5	AirPassengers	144	5	10	4	4
6	BJsales	150	16	28	7	12
7	UKDriverDeaths	192	5	9	3	4
8	Nottem	240	7	20	4	4
9	sunspot.year	289	3	6	3	3
10	co2	468	NA	97	3	3

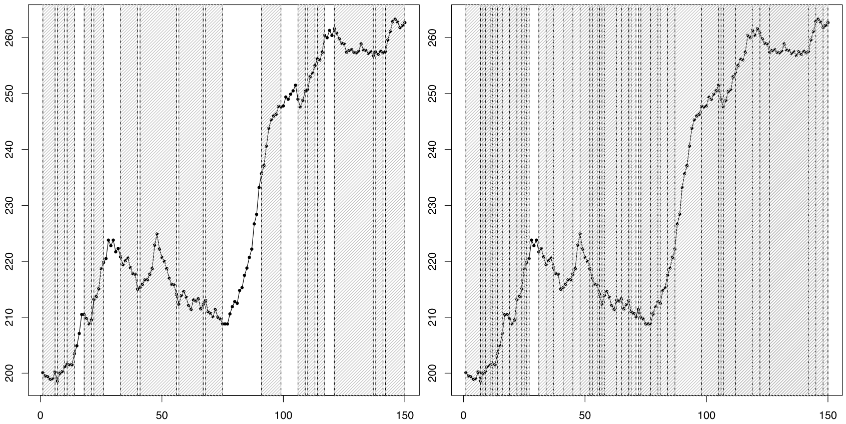


Fig. 6. Difference between two algorithms on BJsales

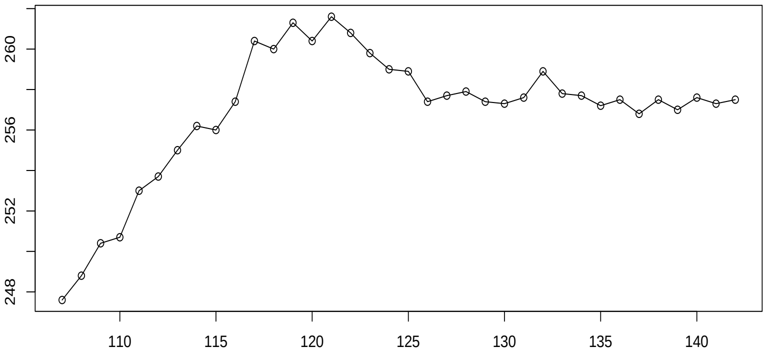


Fig. 7. Subsequence with highest order in BJsales

Surprisingly, the trend almost completely covers the entire sequence, a large number of nodes simultaneously belong to multiple trend range. This shows that BJsales data has a very strong continuity, and the vast majority of nodes in the sequence are subject to trends. At the same time, the error threshold is set to 0.01 to ensure a high degree of polynomial fitting, and the resulting polynomial sequence is highly expressive to the original sequence.

And then examine the highest-order related information in detail. The highest order appears between the 107th node and the 142nd node, as shown in Fig. 7.

It can be seen that this subsequence has a long duration and many changes, which leads to the need to use a higher order polynomial to fit and express; but even 12th order which is used, compared with 36 nodes long sequence, still achieves a fact that a lower order polynomial is able to characterize a high-order sequence.

4.2. Predictive ability

Time series study results are often used for prediction. This paper compares the improved algorithm with ARIMA.

According to the previous experiment, BJsales is suitable for prediction. In order to measure the predictive effect, the first 2/3 part of BJsales is chosen for this experiment. BJsales has 150 nodes and a trend is found from node 84 to node 106, thus we use first 102 nodes to predict the next 4 nodes from 103th node to 106th node.

We firstly investigate the ARIMA. BJsales is a non-stationary series, and needs first difference on the first 102 nodes. The predicted result is

$$[248.8398, 248.7757, 248.7500, 248.7397]^{\mathrm{T}}.$$

The improved algorithm based on sliding window does not need all nodes but only the trend part which is from 84th node to 102nd node. The result is

$$[249.7798, 250.0806, 249.7211, 248.0292]^{\mathrm{T}}.$$

The root means squared errors between the algorithm results and the original values are, respectively, 1.448717 and 0.822345. The improved algorithm is better as shown on Fig. 8.

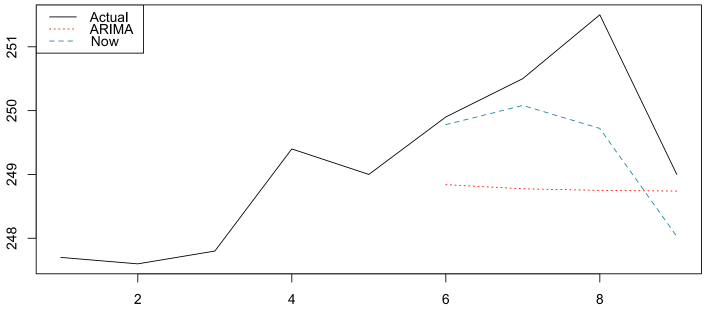


Fig. 8. Prediction effect comparison between the improved algorithm and ARIMA

The improved algorithm not only gets closer values, but also keeps the general shape of the sequence. More than that, the improved algorithm needs fewer nodes than ARIMA to achieve such results, which means it is suitable for streaming data processing.

4.3. Usage of milestone node

Further experiments are taken to verify the role of milestone nodes. The BJsales data are averaged into two segments, the first segment being the pre-sequence (pre) and the second segment being the newly added subsequence (post).

Firstly, 19 trends were obtained on (pre). The milestone node closest to the end of the sequence appeared in the 72nd position, and recorded. After the (pos) is added to (pre), a new sequence is formed which is the same as the original one. According to the milestone node concept, a new trend recognition process is make start since the last milestone node in (pre)-node 72, and 10 trends are obtained then. The two detection results are concatenated for comparing with trend detection results when BJsales is a complete data sequence. The comparison results are identical, while the cost of computation has to be greatly reduced because the nodes before node 72 in (pre) have no need to recalculate.

The experiment results are show in Fig. 9.

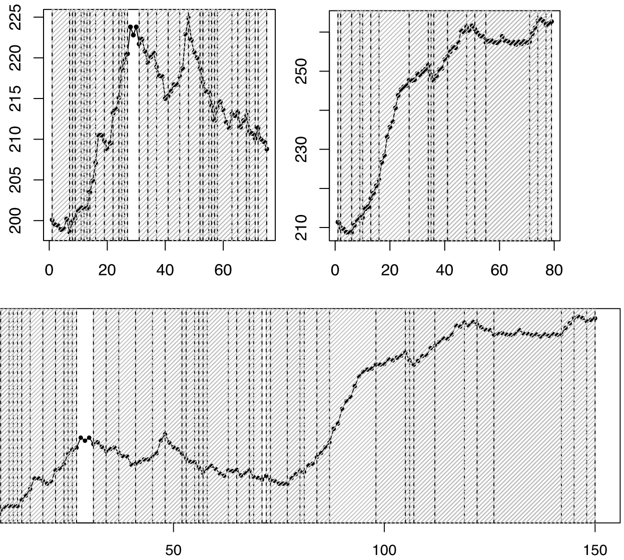


Fig. 9. Trend detection result on two sub-sequences of BJsales compared to the original result

Among them, the last trend in (pre) segment is [72:75], and the first and second trends of the (post) segment are [72:80] and [73:81] respectively. Compared to the original results, it can be seen that the trends in the original sequence around node 72 are the same as that obtained in (post). This is due to the fact that the sequence is artificially cut into two segments, where the trend from the beginning of node 72 has not been truly ended.

Through this part of the time, it can be proved that the start point location is identified by the trend recognition of the milestone node, the trend information which should be existed is not lost, and the repetitive calculation amount can be controlled at a low level.

5. Conclusion

This paper makes some improvements both theoretically and operationally for the traditional trend detection algorithms which have some shortcomings. The following results are obtained:

1. The definition of overlapping trend is given, and it is pointed out that the simple linear segmentation method for trend detection will lose important information.

2. Aiming at the disadvantage of different starting points and the difficulty of recognizing the overlapping trend, sliding window is introduced, and the polynomial fitting ability is improved by using the least squares fitting method.

3. The definition and significance of milestone nodes are given, which greatly reduces the computational complexity when detecting trend from dynamically growing time series data.

Experimental results verify the above conclusions.

***Acknowledgments*:** This research is supported by National Science Fund of China (Grant No 71272161).

References

1. Janusz, M. E., V. Venkatasubramanian. Automatic Generation of Qualitative Descriptions of Process Trends for Fault Detection and Diagnosis. – Eng. Appl. Artif. Intell., Vol. **4**, 1991, No 5, pp. 329-339.
2. Li, H., C. Guo. Survey of Feature Representations Andsimilarity Measurements in Time Series Data Mining. – Appl. Res. Comput., Vol. **30**, 2013, No 5, pp. 1285-1291.
3. Keogh, E., K. Chakrabarti, M. Pazzani, S. Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. – Knowl. Inf. Syst., Vol. **3**, 2001, No 3, pp. 263-286.
4. Agrawal, R., C. Faloutsos, A. Swami. Efficient Similarity Search in Sequence Databases. – Springer, 1993.
5. Struzik, Z. R., A. Siebes. Wavelet Transform in Similarity Paradigm. – Springer, 1998.
6. Keogh, E., S. Chu, D. Hart, M. Pazzani. An Online Algorithm for Segmenting Time Series. – In: Proc. of IEEE International Conference on Data Mining 2001 (ICDM'01), IEEE, 2001, pp. 289-296.
7. Korn, F., H. V. Jagadish, C. Faloutsos. Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. – ACM SIGMOD Rec., Vol. **26**, 1997, No 2, pp. 289-300.
8. Li, A., Q. Zheng. Dimensionality Reduction and Similarity Search in Large Time Series Databases. – Chin. J. Comput., Vol. **28**, 2005, No 9, pp. 1467-1475.
9. Fuchs, E., T. Gruber, J. Nitschke, B. Sick. Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations. – Pattern. Anal. Mach. Intell. IEEE Trans. On, Vol. **32**, 2010, No 12, pp. 2232-2245.
10. Xuedong, G., G. Kan. The Inertia Test and Trend Partition for Trend Detection in Sequential Data. – In: 2015 International Conference on Logistics, Informatics and Service Sciences (LISS'15). –IEEE, 2015, pp. 1-6.
11. Zhang, C., L. Huang, Z. Zhao. Research on Combination Forecast of Port Cargo Throughput Based on Time Series and Causality Analysis. – J. Ind. Eng. Manag., Vol. **6**, 2013, No 1, pp. 124-134.
12. Kuhn, J. Time Analysis in International Logistics Systems with the 6-Sigma Approach Towards an International JIT-System. – J. Syst. Manag. Sci., Vol. **1**, 2011, No 1, pp. 73-81.
13. He, S. Applied Time Series Analysis. Peking University Press, 2004.
14. Fang, J. Time Series Piecewise Linear Representation and Qualitative Trend Analysis. Mastersthesis, Lanzhou University of Technology, 2013.
15. Babcock, B., S. Babu, M. Datar, R. Motwani, J. Widom. Models and Issues in Data Stream Systems. – In: Proc. of 21 ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, 2002, pp. 1-16.