International Journal of Computer Science and Applications,

Vol. 13, No. 1, pp. 70 – 83, 2016

AN ALGORITHM FOR TREND PARTITION DETECTION FROM SEQUENTIAL DATA

GAO XUEDONG

Donlinks School of Economics and Managements, University of Science and Technology Beijing
Beijing, 100083, China
gaoxuedong@manage.ustb.edu.cn

GU KAN*

Donlinks School of Economics and Managements, University of Science and Technology Beijing
Beijing, 100083, China
gukan@xs.ustb.edu.cn

This paper focuses on the detection of trend-based knowledge contained in sequential data. Some key concepts are redefined and an algorithm using the inertia test to dynamically partition a sequence to recognize trend partitions is also proposed. Experiment results prove the effectiveness of the algorithm, and exposing the limitation of traditional primitives. Some suggestions on parameter selection in practical application are given at the same time.

Keywords: Trend; Inertial Test; Trend Partition.

1. Introduction

Data existed in daily life mostly appear in a state of sequence. Time series data whose arrange is relative to the time is one of them. Many researches have been studying time series data widely and deeply for a long time. Different from time series, artificial sequence sorted according to some attributes may contains useful information gets much less attention.

Regardless of which type, the most useful information occurs based on the order of sequential data is trend. Through extracting and analyzing such trends, this process aides in the understanding of objective patterns, further more to detect outliers and forecast. In fact, study on trends are widely applied in fields such as fault diagnosis, investment decision making, military simulations, etc. and get good results.

This paper summarizes the study on trend presenting and extracting, and redefines various important concepts associated with them. An algorithm is used to dynamically partition a sequence data step by step according to the result of inertia test in order to detect and present trend partition. The limitation of traditional primitives is also pointed out.

* Mailbox 739, University of Science and Technology Beijing., No. 30 Xueyuan Road, Haidian District, Beijing

2. Trend Presenting and Extracting

In the field of trend study, at the time when this paper was written, most of the studies are about time series, among which a typical one whose outcomes are mostly about trends are closely related to the representation and extraction of qualitative trends. Qualitative trends primarily focus on the morphological aspects of trends rather than the numerical ones.

2.1. *Definition of Trends*

Prior to studying trend, we have to firstly define and present trends. The general norm of traditional study is to define various collections of fundamental shapes called primitives which demonstrates the most detailed changes in a sequence. By arranging these primitives, one can describe the mega-trend in the whole sequence.

In 1990, Cheung and Stephanopoulos proposed a fundamental language consisting of seven triangular fragments and two trapezoidal fragments [Cheung and Stephanopoulos (1990)]. Janusz and Venkatasubramanian developed a language whose seven primitives are based on the first and second derivatives in 1991 [Janusz and Venkatasubramanian (1991)]. Konstantinov and Yoshida proposed an expandable collection of composite shapes to represent trends in 1992 where they utilized dual-sign strings instead of first and second derivatives [Konstantinov and Yoshida (1992)]. By 2002, Mel´endez and Colomer developed a collection which had trenddefining primitives in thirteen different kinds of shapes [Mel´endez and Colomer (2001)]. And Charbonnier et al. proposed a set of seven trend shapes derived from three fundamental elements which were increasing, decreasing and steady [Char- bonnier et al. (2005)].

These primitives differ from one another due to the difference in research fields or applied targets, but are improvements and changes from the same idea. Generally speaking, most collections of traditional primitives come from a feature of lines which connect three nodes, and who highly resemble the primitives given in publication [Janusz and Venkatasubramanian (1991)].

In the collection, three symbols marked as ‘+’, ‘-’ and ‘0’ were used to present the first and second derivatives, demonstrating the basic shapes of a trend which are portrayed as increasing, decreasing, concaving and convexing. The original set of primitives contained seven shapes, however, the three symbols can be combined to form nine different shapes as in Figure 1.

The majority of primitives in other collections can be obtained from the nine shapes in numerous ways such as distortion and arrangement.

Studies have provided favorable outcomes by presenting trends with countable traditional primitives. However, few studies can explain the reason why countable primitives work well. At the same time, more and more new sets of primitives are being defined because no one could prove the completeness of the selected set of primitives used to present trend characteristics.

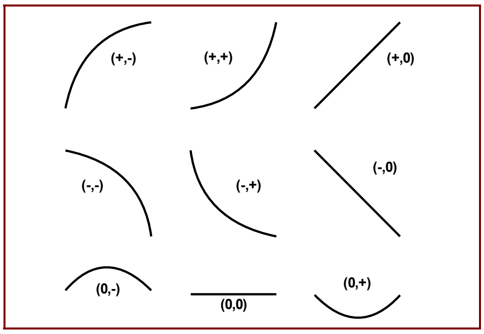

Fig. 1. Typical Traditional Primitives.

2.2. *Trend identification and extraction*

With the primitives, traditional studies tried to divide a sequence into arranged fragments consisted of primitives to describe the shape of the whole sequence automatically.

Konstantinov and Yoshida proposed an algorithm to extract polynomial trends based on a fixed window size [Konstantinov and Yoshida (1992)]. Bakshi and Stephanopoulos then used wavelet decomposition to extract triangular sets from time-series data [Bakshi and Stephanopoulos (1994)]. Vedam and Venkatasubramanian also use wavelet theory to develop an adaptive trend analyzing system [Vedam and Venkatasubramanian (1997)]. Rengasuamy and Venkatasubramanian obtained qualitative trends through neural network [Rengaswamy and Venkatasubramanian (1995)]. By 2003, Dash et al. proposed an algorithm used the fuzzy theory to identify trends [Dash et al. (2003)]. One year later, Dash et al. proposed an interval-halving framework process which automatically identified trends [Dash et al. (2004)]. Chen et al. proposed an improved interval-halving algorithm divides data by data processing [Chen et al. (2011)]. Zhang et al. gave a composed of grey-forecast model and logistic-growth-curve model to improve the accuracy of forecast [Zhang, et al. (2013)].

Considering all the algorithms, a typical way is to divide a sequence into various fixed windows, then fit the data in a polynomial manner in order to identify primitives. Take the algorithm in [Dash et al. (2004)] as an example, if data in the current window is a simple unimodal fragment that fits one of the pre-defined primitives, it is then directly assigned as a primitive. For the composite shapes, primitives can be assigned by splitting them at the point where the first derivative equals to 0 and then testing the separate pieces in their sub-windows respectively.

Curves with high similarity to the original sequence have been successfully obtained with the aforementioned algorithm. Nevertheless, it is still uncertain whether if it is reasonable to divide the sequence into separate pieces prior to the identification of primitives so as to extract only one single trend due to the intervention of human exists in the process.

3. Concepts

In response to the disadvantages mentioned in the previous section, this paper uses real-time analysis for reference, and proposes another way to detect trends. Before explaining the algorithm, several key concepts and definitions will be provided in the following.

Transformation: In a sequence where $P =< p_1, p_2, ..., p_n >$, a ratio used to calculate the difference between the node p_{i-1} and the node p_i is called a transformation from p_{i-1} to p_i.

$$T_i = \frac{p_i - p_{i-1}}{p_{i-1}} \quad (1)$$

Transformation describes the relationship between two nodes such as being increasing, decreasing and in a steady state, and is signed with the symbols of T_i which are +, - and 0. The value of T_i on the other hand, stands for the sharpness of the transformation.

Data in any sequence could be perceived as a single transformation between the first and the last nodes with a large amount of details inevitably ignored. To fully recognize details about the development contained in a sequence, more than two nodes are necessary.

Movement: In a sequence where $P =< p_1, p_2, ..., p_n >$, the evolution path from the node p_{i-1} to p_j $(j > i + 1)$ is called a movement.

The movement function $y(x)$ is a real-valued function about the sequence number of a node which is up to $(j-i)$ order. The basic formula of a movement function is as Eq. (2).

$$\begin{cases} y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{j-i} x^{j-i} \\ x \in \{i, i+1, \ldots, j\} \end{cases} \quad (2)$$

The difference between a transformation and a movement is the amount of nodes taken into consideration. A comparison between two contiguous nodes is a transformation while how it carried over from one node to another is a movement. However, were it not for the smooth demand, any movement can be perceived as a series of transformations.

The movement function transforms the development between two nodes into a function, thereby making a sequence be possible to be fully demonstrated with both the shape and value characteristics kept.

Inertia and inertia test: Inertia occurs when k nodes after node p_j are still contained in the movement function fitted from p_i to p_j within an allowable error range. The inertia test is used to check the existence of inertias.

Fitting nodes and inertial nodes: Fitting nodes are used to fit movement functions. Inertial nodes are the nodes around the movement curve caused by inertia. Inertial nodes are useless for fitting new movement function.

Fitting nodes ultimately determine the movement function while inertial nodes are affected by fitting nodes due to inertia. Inertial nodes are continuation nodes of the original pattern and has little effect on it.

Trend: A movement with underlying inertia from a sequence sampled through fixed steps is called a trend.

The amount of nodes in a trend should be more than the amount of fitting nodes because a trend must pass the inertia test. That is, existing inertial nodes would be automatically included in the movement aside from the fitting nodes. The number of nodes in a trend can be infinite in theory as long as the inertia test is passed.

Adjustment: A movement without underlying inertia from a sequence sampled through fixed steps is called an adjustment.

Adjustments cannot pass the inertia test, as a result, the amount of nodes in an adjustment is equal to the amount used to initialize movement function fitting. When three nodes are used, all possible shapes of adjustments are exactly the same as presented in Figure 1. Therefore, conventional primitives have a higher resemblance to adjustments rather than trends.

Because of the lack of inertia, adjustments cannot be used for prediction purpose.

4. The Problems and Key points in inertia test

4.1. *Movement function fitting*

Trend is a special type of movement, therefore, it is possible to demonstrate trends with a movement function. By definition, a polynomial such as the one in Eq. (2) can be used to represent movement functions. Consequently, fitting a movement function can be converted to a process of solving a high-order polynomial to fitting nodes. Additionally, if the first node in a movement is also the first node in the sequence, then Eq. (2) can be transformed into Eq. (3).

$$\begin{cases} y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{i-1} x^{i-1} \\ x \in \{1, 2, \ldots, i\} \end{cases} \tag{3}$$

Let p be a known data sequence consisting of $i(i > 2)$ fitting nodes. Evidently, data in p satisfies a polynomial up to $(i-1)$ order. Let X be an independent variable matrix constructed with order number of fitting nodes, and mark the coefficient vector of the movement function as w, the aftermath would be Eq. (4) as following.

$$p = X * w \tag{4}$$

And in it,

$$X = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 & \ldots & 1^{i-1} \\ 1 & 2 & 2^2 & 2^3 & \ldots & 2^{i-1} \\ 1 & 3 & 3^2 & 3^3 & \ldots & 3^{i-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & i & i^2 & i^3 & \ldots & i^{i-1} \end{bmatrix}_{i*i}$$

$$w = (a_0, a_1, \ldots, a_{i-1})^T$$

Presumably, w in strict mathematical sense can be attained with a known vector p through Eq. (5) as below. And each w can uniquely determine a movement function.

$$w = X^{-1} * p \tag{5}$$

Determinants for matrix X can then be calculated according to the following.

$$det(X) = 1^{i-1} * 2^{i-2} * \cdots * (i-1)^{1} * i^{0}$$

Given the fact that i is an integer greater than two, therefore, $det(X) \neq 0$, and w is solvable.

4.2. *Inertia Test*

By definition, the existence of inertia can be detected by examining the new nodes fit the movement function or not. In simple terms, it can be transformed into a problem of measure the difference between the theoretical value in the movement function and actual values in the sequence. This paper chose a ratio value to measure discrepancies in order to avoid the unwanted influence caused by absolute value.

Let ε be the allowable error threshold. The difference between actual value p_i and theoretical value y_i at a particular node i is measured as per Eq. (6) below. Any node satisfying Eq. (6) would be considered as passing the inertia test.

$$\left| \frac{p_i - y_i}{y_i} \right| \leq \varepsilon \tag{6}$$

Basic rules applied in performing the inertia test are listed in the follows.

(1) Using three nodes to fit the initial movement function.

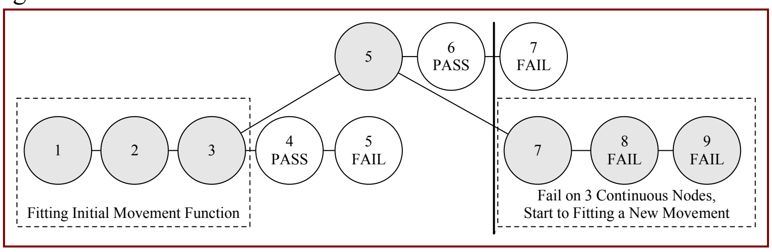

Fig. 2. Basic Process of Inertia Test.

(2) If the inertia test was passed on node p_i, the current movement function is believed to be correct. The inertia test will then be continued on node p_{i+1}.

(3) If the inertia test had failed on node p_i, then this node will be put back into the original set of fitting nodes, and will attempt to fit it into the movement function once again.

(4) If the inertia test has failed all three consecutive nodes being node p_i, p_{i+1} and p_{i+2} for instance, the node p_{i-1} should be the last node of the previous movement. Then a new fitting process will commence beginning from node p_i.

(5) The inertia test stops when there are less than three nodes left in the sequence.

An integral process is shown in Figure 2.

In Figure 2, fitting nodes are in gray while inertia nodes are in white. Node 1, 2 and 3 are used to initialize the fitting procedure. The inertia test of the initial movement function is passed on Node 4 but failed on Node 5. A new movement function fitted from Node 1, 2, 3 and 5 passed the inertia test on Node 6, which means the function is correct. However, Node 7, 8 and 9 all failed to pass the inertia test due to the fact that an old movement had already ended on Node 6, thus, a new movement function should begin from Node 7.

Inertia nodes such as Node 4 in Figure 2 may appear sometimes when fitting a new movement function during the process of inertia testing. Since these nodes have no effect on changes in the movement function, they will be omitted when doing calculations in this paper.

Extend the sequence p to a new sequence contains j fitting nodes, marked as p'. And let X' be the new matrix after adding temporary nodes into X. Eq. (7) from which a new vector w will then arise can ben obtained.

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_i \\ \vdots \\ p_j \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 & \dots & 1^{l-1} \\ 1 & 2 & 2^2 & 2^3 & \dots & 2^{l-1} \\ 1 & 3 & 3^2 & 3^3 & \dots & 3^{l-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & i & i^2 & i^3 & \dots & i^{l-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & j & j^2 & j^3 & \dots & j^{l-1} \end{bmatrix}_{l*l} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_i \\ \vdots \\ a_{l-1} \end{bmatrix} \quad (7)$$

If the error threshold is not controlled, fitting the movement function and performing the inertia test will rapidly accelerate the growth of j and l in X'. Therefore, it is necessary to choose a suitable threshold. Besides, the calculation is done based on the most detailed data. In order to guarantee the algorithm to be practical and avoid overfitting, the selection of step length should also be taken into consideration.

The selection of parameters at the moment has been on a trial-by-examination basis since a sound theoretical technique has not been discovered yet. In this paper, the error threshold is chosen as $\varepsilon = 0.01$, and the step length is kept as same as it is in the data set.

5. Algorithm for trend Partition and trend Detection

A new algorithm is proposed to partition a sequence into fragments and to detect trends based on the concepts and problem solutions provided in previous sections.

5.1. *Algorithm description*

Trends may be acquired from movements and any sequential data who contains more than three nodes could be divided into an ordered list of movements. A movement is considered to have ended with no more trends could be derived when inertia test has failed. A simple movement could be developed from either a trend or an adjustment, which means the length of movements is uncertain.

Taking into account the idea of fitting movement functions and rules to test inertia in prior sections, a basic process of the algorithm is presented as below.

Input: data sequence; error threshold (ε)

Output: sequence of movements

(1) Are there more than 3 nodes remaining in the sequence? If so, move on to (2), otherwise, end the algorithm.
(2) Set matrix X, and acquire vector w.
(3) Read the next node.
(4) Perform the inertia test on current node. If passed, go back to (3), otherwise, proceed to (5).
(5) Is the number of consecutive inertia tests which have failed equals to 3? If yes, extract the current movement, save the result sequence, and go back to (1), otherwise, continue to (6).
(6) Re-define X as X', and acquire a new vector w'.
(7) Perform inertia test on the next node. If passed, replace w with w', go back to (3), otherwise, return to (5).

This algorithm should be continuously iterated until the whole sequence has been divided into an ordered collection of movements.

5.2. *Example*

A simple example is given to help understand the algorithm.

Let $P =< 39, 20, -1, -18, -25, -14, -51, -76, -56, -40, -22, -36, 2, 44, 90 >$, who is actually a segmented function as the following.

$$p = \begin{cases} x^3 - 7x^2 - 5x + 50, 1 \leq x \leq 5 \\ 6x^2 - 115x + 460, 6 \leq x \leq 8 \\ x^2 - 3x - 110, 9 \leq x \leq 11 \\ 2x^2 - 12x - 180, 12 \leq x \leq 15 \end{cases}$$

And the shape is drawn as Figure (3).

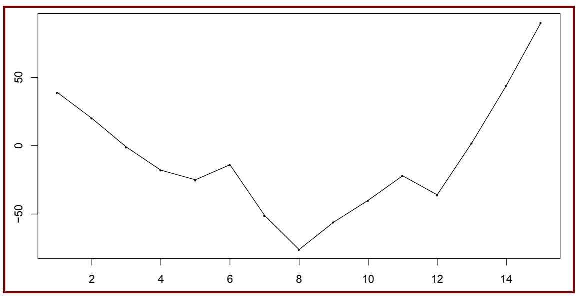

Fig. 3. Shape of example series.

Four movement functions can be acquired automatically using algorithm above in which two are trends and the rest are adjustments. The result is shown as below.

$$\begin{cases} w_1 = [1, -7, -5, 50]^T \\ w_2 = [6, -55, 35]^T \\ w_3 = [1, 13, -70]^T \\ w_4 = [2, 32, -70]^T \end{cases}$$

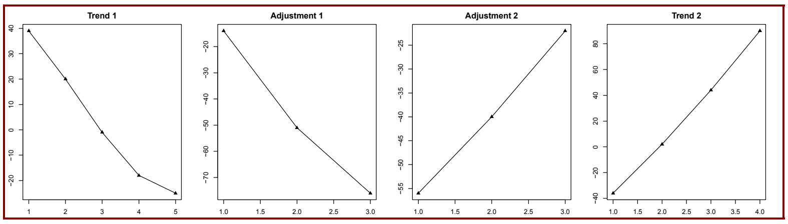

Fig. 4. Result of the simple example.

Besides the first one, notice the movement functions derived from w_2, w_3 and w_4 are shifted leftwards compared to the original functions by 5, 8 and 11 units respectively. The reason why this occurs is because neighboring trends or adjustments have no effect on each other, which then makes each movement brand new in nature, thereby resulting in their assigned serial number to restart at 1.

The example above clearly demonstrated accuracy of the algorithm in terms of sequence partitioning and trend detection.

6. Experimental verification

In order to prove the effectiveness of the algorithm and test the ability of conventional primitives, some data sets from R language and other public repositories are used to test the algorithm programmed in R.

The general idea is to run algorithm on multiple original data sets, and get movement functions. The information about trend contained in the sequences would be acquired from the movement functions. By comparing the movements to the original data and checking the highest order of the movement functions, not only the fitting degree but also the effect of conventional primitives can be verified.

On the basis of previous experiment results, particular data sets are chosen to analyze step length and error threshold.

6.1. *Evaluation criteria*

Root mean square error (RMSE) and Chebyshev distance is used in this paper to evaluate the ability of presenting and fitting degree of movement functions to the original sequential data.

The RMSE is the square root of the sum of squares in predicted values y_i, observed values $\hat{y}_i$, and the observation times n, marked as R^2. Although some papers also use root mean square residual (RMR) to value the difference [Feng et al. (2011)], RMSE is enough for this paper.

R^2 in an n-times observation can be calculated as Eq. (8) below.

$$R^2 = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{8}$$

Generally speaking, the smaller the value of R^2, the better the fitting degree.

This paper modifies the original Chebyshev distance into an improved one by divide it with the distance between the maximum value and the minimum value of two sequences, and the biggest difference between fitting result and original sequence can be gotten from it. The improved Chebyshev distance between two n-dimensional vectors which are P and Q is:

$$D(P,Q) = \frac{\max_i(|p_i - q_i|)}{\max(P,Q) - \min(P,Q)} \tag{9}$$

Because trends can be derived from movement functions, the underlying nodes form a sequence, and the nodes in movement functions and the nodes in original sequences have a one to one correspondence, thereby making it possible to compare between nodes in trends and nodes in the original sequence using RMSE and Chebyshev distance when step is chosen as 1.

6.2. *The result of fitting and partition*

This section selected eight data sets and chose the step length equal to 1 when applying the algorithm. Results are listed in Table 1.

Table 1. Results on 8 Data Sets, for *Step*=0.1.

Data sets	Nodes	Trends/Movements	R^2	Distance	Max(order)
BJsales	150	16/27	0.0605	0.0392	7
USAccDeaths	72	3/22	0.5000	0.0145	3
Skirts	46	4/11	0.0739	0.0150	4
Kings	42	0/14	0.0000	0.0000	2
Fancy	84	2/27	0.1673	0.0001	4
Precip1	100	1/33	0.0021	0.0099	2
NYBirths	168	4/54	0.0011	0.0150	4
WWWusage	100	7/28	0.0400	0.0177	4

The results clearly demonstrate the fitting degree between the fitted sequence and the original sequence. The average value of R^2 is 0.1056, and the average of Chebyshev distance is 0.0139. Respectively, trends and adjustments within the movement function can be detected while minimizing loss and narrowing fluctuation margin of details in the original sequence.

On the other hand, the amount of adjustments detected is far more than the amount of detected trends. The average amount of trends in the movements is 19%. In other words, most movements are in fact adjustments consisting of three nodes. The proportion of trends is much lower than expected.

The results above justify the reason why traditional primitives consisting of three nodes can obtain a superior fitting degree. Moreover, sequences containing more adjustments all result in a smaller R^2 value in the experiment. In some data sets, the value of R^2 can even equal to zero, which means they are perfectly fitted.

As to the maximum order of movement functions, the highest order of the movement functions in the experiment reached a value of 7 while two few-trend-existing data sets, Kings and Precip1, had at most, a power of 2. The others all had a maximum power of 3 or 4. The results proves that the diversity of the movement functions consequently cannot be represented with primitives consisting of only 3 nodes.

In order to further analyze experiment results and draw more conclusions, Skirts is selected to take more experiments.

First step is using Skirts data to draw a line chart. Then, split the curve into pieces with dashes to mark trends out, arriving at Figure 5 below.

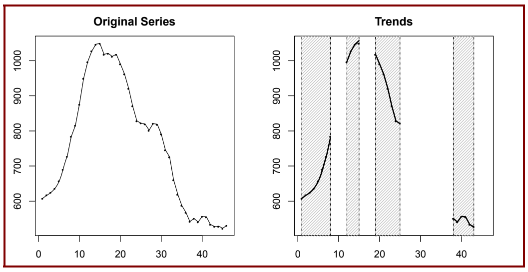

Fig. 5. Trends in Skirts.

The original data sequence is shown in the left figure and the trends are highlighted in right with thick lines. There are four trends in Skirts and each trend covers 6.25 nodes.

Overlap the two figures in Figure (5), and the fitting degree can be checked.

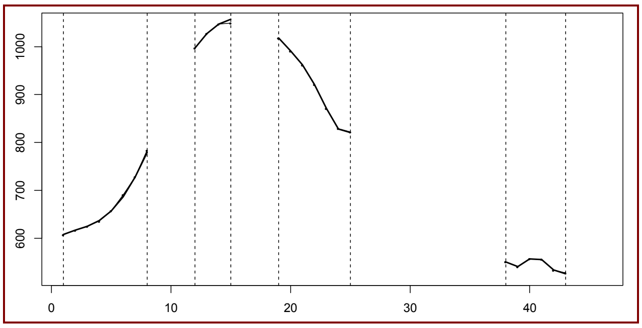

Fig. 6. Fitness between trend and original sequence.

According to the diagrams in Figure (6) above, a favorable fitting degree is achieved in trend areas. Furthermore, movement functions representing trends can demonstrate not only the shapes of the trends but also the value of every node within the trends, making it easy to compare the two sequences side by side.

6.3. *Parameter selection*

The result is influenced by step length and error threshold. All the experiments in previous sections take step as 1 and ε as 0.01. Here gives an experiments on parameter selection.

First take the step length into consideration. By making step length as 1, although details are preserved when fitting the movement functions, both trends and adjustments are too miniscule and practically meaningless in guiding actual work in reality.

According to the definition, additional steps are utilized to perform the algorithm on WWWusage data set. Some results with shade standing for trends are shown in Figure 7.

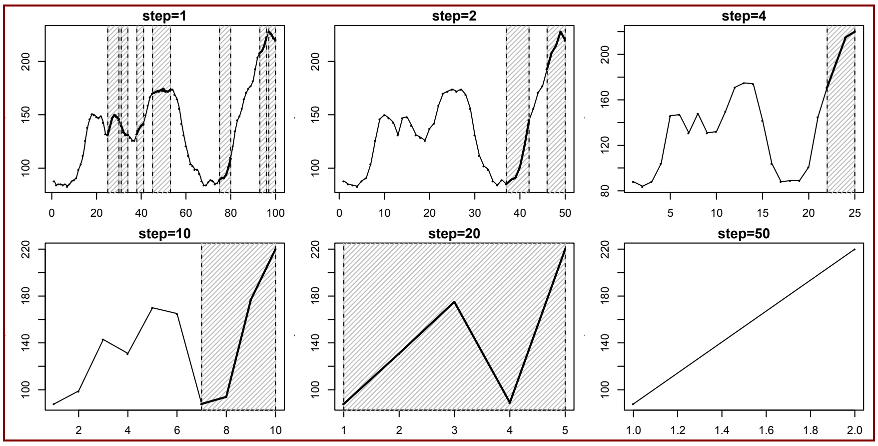

Fig. 7. Trends with different steps in WWWusage.

Based on Figure 7, it can be concluded that variations in step lengths could affect trends at a macroscopic level. Nevertheless, greater the step length does not necessarily translate to detection of more valuable trends.

For example, when the step length gets greater, the shape becomes abstracter, the amount of trends goes fewer and the range of each trend covers is wider. When step is 10, the remaining nodes still keep an obvious shape of the sequence. When step is 20, the sequence leaves four nodes and a rough shape behind. If keep reducing the amount of nodes, there would not be enough nodes to form a single trend, and only an increasing transformation remains.

Interestingly, there are only five nodes in the sequence when the selected step length is 20, and the shape of the five nodes can roughly show the developments process of the original sequential data. At the same time, there is only one trend detected from these five nodes, which means inertia still exists in the current movement. Trends such as the one in this case is very useful for forecasting purposes.

Continue to check the effect of error threshold on WWWusage. When the step length is 4, running the algorithm with different error threshold can draw the Figure 8 as below.

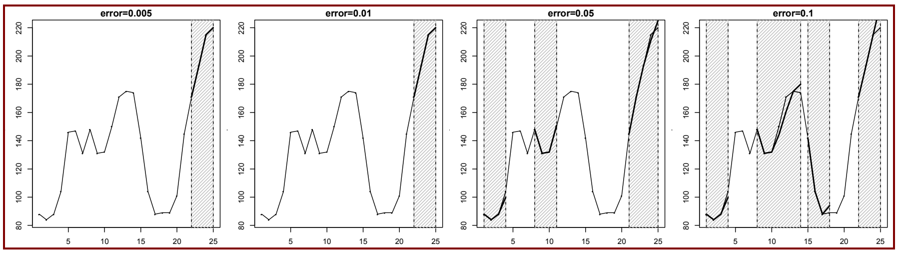

Fig. 8. Trends with different error threshold in WWWUsage.

As we can see, although the difference is small when $\varepsilon = 0.005$ and $\varepsilon = 0.01$, but the fitting degree gets obviously decrease when ε gets greater such as $\varepsilon = 0.05$ and $\varepsilon = 0.1$.

Summing up the above, trend is a relative concept, the amount of trends exist in a sequence is depend on the condition when detecting trends.

7. Conclusion

This paper explained various key concepts, and drew some conclusion as following:

(1) A trend has to be a movement with inertia. Those which does not affect following nodes cannot be considered as a trend.

(2) An algorithm used to divide sequences and detect trends is proposed. The experiment results on the eight data sets came to the conclusion that high-proportion of adjustments in movement functions begets better fitting degree of traditional primitives, and it is difficult to represent real trends with limited amount of nodes.

(3) According to the practical demand, sampling through fixed step length from the sequence and determining an error threshold prior to detecting trends provide better guidance in practical applications.

Acknowledgments

This research is supported by national science fund of China (71272161).

References

Bakshi BR and Stephanopoulos G. (1994). Representation of process trends—III. Multiscale extraction of trends from process data. Computers & Chemical Engineering, 18(4): 267–302.

Charbonnier Sylvie, et al. (2005). Trends extraction and analysis for complex system monitoring and decision support. Engineering Applications of Artificial Intelligence, 18(1): 21–36.

Chen Junping, et al. (2011). An Improved Interval-halving Algorithm for Qualitative Trend Extraction. Instrumentation Technology, (2): 57–60.

Cheung JT-Y and Stephanopoulos George. (1990). Representation of process trends—Part I. A formal representation framework. Computers & Chemical Engineering. 14(4): 495– 510.

Dash Sourabh, et al. (2003). Fuzzy-logic based trend classification for fault diagnosis of chemical processes. Computers & Chemical Engineering. 27(3): 347–362.

Dash Sourabh, et al. (2004). A novel interval-halving framework for automated identification of process trends. AIChE journal. 50(1): 149–162.

Feng Changli, et al. (2011). Factors of knowledge sharing in supply chain-based on perspective of knowledge characteristics. Journal of System and Management Sciences. 1(3): 21–29.

Janusz Margaret E and Venkatasubramanian Venkat. (1991). Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. Engineering Applications of Artificial Intelligence. 4(5): 329–339.

Konstantinov Konstantin B and Yoshida Toshiomi. (1992). Real-time qualitative analysis of the temporal shapes of (bio) process variables. AIChE Journal. 38(11): 1703–1715.

Mel´endez Joaquim and Colomer Joan. (2001). Episodes representation for supervision. Application to diagnosis of a level control system. Workshop on principles of diagnosis DX. 1: 119–126.

Rengaswamy Raghunathan and Venkatasubramanian Venkat. (1995). A syntactic pattern-recognition approach for process monitoring and fault diagnosis. Engineering Applications of Artificial Intelligence. 8(1): 35–51.

Vedam Hiranmayee and Venkatasubramanian Venkat. (1997). A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis. American Control Conference, 1997. Proceedings of the 1997. 1: 309–313.

Zhang Chi, et al. (2013). Research on combination forecast of port cargo throughput based on time series and causality analysis. Journal of Industrial Engineering and Management. 6(1): 124–134.