

Web Log Analysis Based on Method of Sequence Alignment

Abstract

Consumer behavior is the decision processes and acts of individuals or groups involved in buying and using products. We will propose a method for analyzing consumer behavior data. Firstly the method of sequence alignment is used to calculate the fluctuations, then control charts are introduced to detect possible exceptions. We will verify the results after implementation.

CONTENTS

Abstract	1
CONTENTS	1
1. Introduction	2
2. Principles	2
2.1 Sequence Alignment[1]	2
2.1.1 Overview	2
2.1.2 Needleman-Wunsch Algorithm	3
2.1.3 Smith-Waterman Algorithm	5
2.2 Control Charts	6
2.2.1 Overview	6
2.2.2 EWMA Control Charts [2]	6
2.2.3 CUSUM Charts [3]	7
3. Implementation	7
3.1 Overview and Preprocess	7
3.2 Implementation of Sequence Alignment	9
3.3 Implementation of EWMA Charts	10
3.4 Implementation of CUSUM Charts	10
4. Results and Conclusions	10
References	13
APPENDIX	14
Appendix I: Raw Data	14
Appendix II: Alignment Score	15
Appendix III: EWMA	18
Appendix IV: CUSUM	21

1. Introduction

Consumer behavior is the decision processes and acts of individuals or groups involved in buying and using products, the study of consumer behavior has great significance while making business decisions, therefore a lot of importance is attached to this study. Along with the introduction of modern data analysis methods, the study has been remarkably improved.

In this context, we will propose a method for analyzing consumer behavior.

Specifically, given a web log where customers' purchasing histories are recorded. In order to make appropriate business decisions according to this log, we need to generalize the regularity of the customers' purchasing, and to find out possible unusual behaviors. To do that, firstly the method of sequence alignment is used to calculate the fluctuations of purchasing history in time dimension, then control charts, namely EWMA chart and CUSUM chart, are introduced to detect possible exceptions.

The report consists of four sections: firstly the introduction, including the context and the briefing of the report; in the second section we will represent the principles of algorithms used in the method; in the third section we will discuss how these methods are implemented; finally is the result of these implementation and the conclusion.

2. Principles

2.1 Sequence Alignment[1]

2.1.1 Overview

A sequence is an ordered list of objects or events.

Widely used in bio-informatics, but also in other fields, a sequence alignment is a way of arranging two or several sequences in order to find out the degree of similarity or even homology between these sequences. Generally, such degree is quantified by an alignment score, which is, determined by three factors: a match score, a mismatch score, and a gap penalty. Commonly the match score is positive, and the other two are negative. Thus, a higher score indicates a higher degree of similarity. In practice, these factors are featured in a similarity matrix.

Example 1: Given following two sequences: “ABCD” and “ACD”. We define the similarity

	A	B	C	D
A	1	-1	-2	-3
B	-1	1	-1	-2
C	-2	-1	2	-1
D	-3	-2	-1	3

Fig. 1. Similarity matrix used in example 1

matrix as shown in fig. 1, which indicates, for example, that the match score of pair of D-D is $S(D, D) = 3$, or the mismatch score of pair B-C is $S(B, C) = -1$; we also define that the gap penalty $d = -2$.

In this case, the score for alignment:

Sequence 1: ABCD

Sequence 2: AC -D

will be:

$$S(A, A) + S(B, C) + d + S(D, D) = 1 + (-1) + (-2) + 3 = 1$$

And apparently, the alignment:

Sequence 1: ABCD

Sequence 2: A -CD

will gain the highest score, which will be:

$$S(A, A) + d + S(C, C) + S(D, D) = 1 + (-2) + 2 + 3 = 4$$

Very short or very similar sequences can be aligned manually. While in the majority of alignment problems we ought to deal with long and non-intuitive sequences. Thus computational approaches are introduced. Within the category of pairwise alignment, the global technique Needleman-Wunsch algorithm, which is useful when two sequences are similar and have roughly equal size, and the local technique Smith-Waterman algorithm, useful when both sequences are supposed to contain similar fragment, are widely adopted. Both algorithm are based on dynamic programming, a method for solving complex problems by breaking them down into simpler subproblems.

2.1.2 Needleman-Wunsch Algorithm

Consider the two sequences A and B to be aligned, to align the sequences with the highest score, a matrix M called score matrix, whose number of rows and columns are respectively determined by the length of both sequences, is allocated. Entry located in row i and column j is denoted here by $M(i, j)$, which depends on the three entries corresponding to either or both sequence having one less element, namely $M(i-1, j)$, $M(i, j-1)$ and $M(i-1, j-1)$. It is calculated as follows:

$$M(i, j) = \max(M(i-1, j-1) + S(A_i, B_j), \\ M(i, j-1) + d, \\ M(i-1, j) + d) \quad (1)$$

Where A_i is the i th element in sequence A; $S(A_i, B_j)$ is the similarity value between the i th element in sequence A and the j th element in sequence B.

Once we have computed this score for every entry, we must do a “traceback”, that is to determine the actual alignment that lead to the score. Because when computing the score of an entry we took a max over three other entries, on the traceback, from lower right-hand corner we go to the

location with the highest score among the three: leftwards or upwards corresponds to gaps, and along the diagonal corresponds to a match.

The algorithm could be written in pseudocode as in listing 1.

```
#Assign the scores to the score matrix  $M$ 
# $d$  is the gap penalty
rows = length of sequenceA, columns = length of sequenceB
#Entries in the first row and the first column of  $M$ 
for  $j = 0$  to columns:
     $M(0, j) = j * d$ 
for  $i = 0$  to rows:
     $M(i, 0) = i * d$ 
#Other entries
for  $i = 1$  to rows:
    for  $j = 1$  to columns:
        # Now we have 3 choices
        choice1 =  $M(i-1, j-1) + S(A_i, B_j)$     # If characters are aligned
        choice2 =  $M(i-1, j) + d$                 # Gap in sequence B
        choice3 =  $M(i, j-1) + d$                 # Gap in sequence A
        let  $M(i, j)$  be max(choice1, choice2, choice3)

#Traceback
 $i = \text{rows}, j = \text{columns}$  #lower right-hand corner
while( $i > 0$  or  $j > 0$ ):
    if  $M(i, j) == M(i-1, j-1) + S(A_i, B_j)$ :    #Traceback diagonally
        add  $i$ th character of sequenceA to alignedSequenceA
        add  $j$ th character of sequenceB to alignedSequenceB
        decrement  $i$  by 1
        decrement  $j$  by 1
    else if  $M(i, j) == M(i-1, j) + d$ :            #Traceback upwards
        add  $i$ th character of sequenceA to alignedSequenceA
        add '-' to alignedSequenceB
        decrement  $i$  by 1
    else:                                         #Traceback leftwards
        add '-' to alignedSequenceA
        add  $j$ th character of sequenceB to alignedSequenceB
        decrement  $j$  by 1
```

Listing 1. Pseudocode for Needleman-Wunsch Algorithm

Example 2: Given two sequences: “ACTGATTCA” and “ACGCATCA”. Using -2 as gap penalty, -3 as mismatch penalty, and 2 as a match score.

In this case, since the length of two sequences are 9 and 8, the size of the scoring matrix should be 10×9 . Then assign scores to the matrix using the formula (1). Fig. 2 shows the matrix after

assignment of the scores. Then traceback the optimal path from the lower right-hand corner, it is also shown in fig. 2.

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Fig. 2. Score matrix in example 2

Therefore the alignment result is

Sequence 1: ACTG -ATTCA

Sequence 2: AC -GCAT -CA

2.1.3 Smith-Waterman Algorithm

A local alignment is the best (highest scoring) alignment of a substring of A to a substring of B, as opposed to a global alignment of the entire strings.

The main difference to the Needleman–Wunsch algorithm is that negative scoring matrix entries are set to zero, which renders the (thus positively scoring) local alignments visible. Traceback starts at the highest scoring matrix entry and proceeds until a entry with score zero is encountered.

So that in Smith-Waterman algorithm, formula(1) transforms as follows

$$\begin{aligned}
 M(i, j) = \max(& M(i-1, j-1) + S(A_i, B_j), \\
 & M(i, j-1) + d, \\
 & M(i-1, j) + d, \\
 & 0)
 \end{aligned}
 \tag{2}$$

2.2 Control Charts

2.2.1 Overview

Control charts are tools used to determine if a manufacturing or business process is in a state of statistical control. If a single quality characteristic has been measured or computed from a sample, the control chart shows the value of the quality characteristic versus the sample number or versus time.

If the control chart indicates that the process is currently under control, then no corrections or changes to process control parameters are needed or desired. In addition, data from the process can be used to predict the future performance of the process. If the chart indicates that the monitored process is not in control, analysis of the chart can help determine the sources of variation. A process that is stable but operating outside of desired limits needs to be improved through a deliberate effort to understand the causes of current performance and fundamentally improve the process.

Normally, a control chart consists of:

1. Points representing a certain statistic of the data.
2. The mean of the statistic where a center line will be drawn.
3. Upper and lower control limits that indicate the threshold at which the process output is considered statistically 'unlikely', these two limits are drawn typically at 3 standard errors from the center line.

If the process is in control (and the process statistic is normal), 99.7300% of all the points will fall between the control limits. Any observations outside the limits suggest the introduction of a new (and likely unanticipated) source of variation, known as a special-cause variation.

Different types of control charts are used in various occasions. Particularly while detecting smaller changes, EWMA charts and CUSUM charts are frequently used

2.2.2 EWMA Control Charts [2]

The Exponentially Weighted Moving Average (EWMA) is a statistic for monitoring the process that averages the data in a way that gives less and less weight to data as they are further removed in time.

The observation of EWMA versus time is defined by

$$\begin{aligned} EWMA(t) &= \lambda * Y(t) + (1 - \lambda) * EWMA(t - 1) \\ t &= 1, 2, \dots, n \end{aligned} \quad (3)$$

Where: $EWMA(0)$ is the mean of historical data (target), $Y(t)$ is the observation at time t ; n is the number of observations to be monitored including $EWMA(0)$; $0 < \lambda \leq 1$ is a constant that determines the depth of memory of the EWMA.

Upper limit ucl and lower limit lcl are respectively defined by

$$ucl = EWMA(0) + k * \sqrt{\lambda / (2 - \lambda) * s^2} \quad (4)$$

$$lcl = EWMA(0) - k * \sqrt{\lambda / (2 - \lambda) * s^2} \quad (5)$$

Where: $EWMA(0)$ is the mean of historical data (target); the factor k is either set equal 3 or chosen using the Lucas and Saccucci (1990) tables; s is the standard deviation calculated from historical data; $0 < \lambda \leq 1$ is a constant that determines the depth of memory of the EWMA.

If the $EWMA(t)$ is within the interval between ucl and lcl , the process is in control, and vice versa if the $EWMA(t)$ is outside the interval.

2.2.3 CUSUM Charts [3]

CUSUM stands for cumulative sum. As its name implies, CUSUM involves the calculation of a cumulative sum. The observation values $CUSUMp(i)$ are calculated as follows

$$CUSUMp(i) = \max(0, CUSUMp(i - 1) + x(i) - \bar{x} - k) \quad (6)$$

Where: $CUSUMp(0) = 0$; $x(i)$ is the observation at i ; \bar{x} is the control center (mean of the process);

$$k = \delta * \sigma / 2 \quad (6.1)$$

where δ is the amount of shift that we wish to detect, expressed as a multiple of standard deviation; σ is the standard deviation.

When the value of S exceeds a certain threshold value, a change in value has been found. The above formula only detects changes in the positive direction. When negative changes need to be found as well, the min operation should be used instead of the max operation, and this time a change has been found when the value of S is below the (negative) value of the threshold value.

In practice, we use the formula as follows

$$CUSUMn(i) = \max(0, CUSUMn(i - 1) - x(i) + \bar{x} - k) \quad (7)$$

The certain threshold value is defined as follows

$$h = k * (2 / (\delta * \sigma)) * \ln((1 - \beta) / \alpha) \quad (8)$$

Where δ and σ are explained as above; α is the probability of a false alarm; β is the probability of not detecting that a shift in the process mean has, in fact, occurred.

When either $CUSUMp(i)$ or $CUSUMn(i)$ exceeds h , the process is out of control.

3. Implementation

3.1 Overview and Preprocess

Within this report, consider a web log of a customer's purchase history in a certain period of time in the form as shown in fig.3.

<i>Weeks Expense Items</i>	A	B	C	D	E	F
1	4.84	0.00	0.00	0.00	0.00	0.00
2	1.10	0.00	0.00	3.63	0.00	0.00
3	7.04	0.00	0.00	1.87	0.00	0.00
⋮ ⋮	⋮ ⋮					

Fig. 3 customer's purchase history

Where *Items* A, B, C, D, E, F are six goods that the customer purchased, *Weeks* are the chronological weeks during the period, *Expense* is the expense on each goods during each week.

The reason we use sequence alignment method to reflect fluctuations of the record is that the method can put a specific data in a relatively large scale, make it compare with a range of data, to reduce random errors.

In order to implement sequence alignment method, followed by control charts on this log, instead of a similarity matrix, we need to calculate the similarity of purchase between any two weeks within the log.

To do that, the purchase of each week could be considered as a vector. For example:

$$\text{week}(1): [4.84 \ 0.00 \ 0.00 \ 0.00 \ 0.00 \ 0.00] = 4.84\mathbf{a}$$

$$\text{week}(2): [1.10 \ 0.00 \ 0.00 \ 3.63 \ 0.00 \ 0.00] = 1.10\mathbf{a} + 3.63\mathbf{d}$$

Then we can calculate the similarity between these two vectors.

There are several ways to calculate the similarity between vectors, including Euclidean distance measure, Mahalanobis distance measure, Minkowsk distance measure, etc. Considering that all purchase data share the same unit, and each goods is regarded as independent, therefore the simplest Euclidean distance measure is applicable. In Cartesian coordinates, the Euclidean distance between vector $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_n]$ and $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]$ is given by

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (9)$$

Example 3: The distance between $\text{week}(1)$ and $\text{week}(2)$ is

$$d(\text{week}(1), \text{week}(2)) = \sqrt{(4.84 - 1.10)^2 + (0.00 - 3.63)^2} = 5.21$$

According to definition, two vectors with lower similarity will lead to bigger distance. In sequence alignment, on the contrary, lower similarity need to be featured with smaller similarity values (and it should be always greater than zero).

Hence, we introduce a constant cap , equals to the maximum distance among all the vectors, i.e.

$$cap = \max(d) \quad (10)$$

We define

$$S(\text{week}(i), \text{week}(j)) = \text{cap} - d(\text{week}(i), \text{week}(j)) \quad (11)$$

Where: $S(\text{week}(i), \text{week}(j))$ may be regarded as the similarity value between $\text{week}(i)$ and $\text{week}(j)$.

Apparently, it conforms to the requirements.

3.2 Implementation of Sequence Alignment

Sequence alignment method is applied in the case. Specifically, is to align the data of four consecutive weeks with one following week, then move forward week by week. (i.e., align week 1, 2, 3, 4 with week 5, take the result as first term, then align week 2, 3, 4, 5 with week 6, take the result as second term, and so on)

Considering that the data in each week is equivalent, therefore the gap penalty could be considered as zero. To align the sequences from k th week, i.e., align the sequence A: [$\text{week}(k)$ $\text{week}(k + 1)$ $\text{week}(k + 2)$ $\text{week}(k + 3)$] with sequence B: [$\text{week}(k + 4)$].

In this case pseudocode in listing 1 transforms as in listing 2.

```
#Assign the scores to the score matrix M
rows = 4, columns = 1
#Entries in the first row and the first column of M
for j = 0 to columns:
    M(0, j) = 0
for i = 0 to rows:
    M(i, 0) = 0
#Other entries
for i = 1 to rows:
    for j = 1 to columns:
        # Now we have 3 choices
        choice1 = M(i-1, j-1) + S(week(i + k - 1), week(j + k + 4 - 1))    # If characters are aligned
        choice2 = M(i-1, j)          # Gap in sequence B
        choice3 = M(i, j-1)          # Gap in sequence A
        let M(i, j) be max(choice1, choice2, choice3)

#Traceback
i = rows, j = columns #lower right-hand corner
while(i > 0 or j > 0):
    if M(i, j) == M(i-1, j-1) + S(week(i + k - 1), week(j + k + 4 - 1)): #Traceback diagonally
        add (i + k - 1)th week of sequenceA to alignedSequenceA
        add (j + k + 4 - 1)th week of sequenceB to alignedSequenceB
        decrement i by 1
        decrement j by 1
    else if M(i, j) == M(i-1, j): #Traceback upwards
```

```

    add (i + k - 1)th week of sequenceA to alignedSequenceA
    add '-' to alignedSequenceB
    decrement i by 1
else:
    #Traceback leftwards
    add '-' to alignedSequenceA
    add (j + k + 4 - 1)th week of sequenceB to alignedSequenceB
    decrement j by 1

```

Listing 2. Pseudocode for Needleman-Wunsch Algorithm that implemented in the case

3.3 Implementation of EWMA Charts

In order to calculate the observation of EWMA, considering formula(3). Because the mean of historical data is not available, we take the mean of the current data as $EWMA(0)$. We also let $\lambda = 0.2$. Therefore formula(3) transforms as follows

$$EWMA(t) = 0.2 * Y(t) + (1 - 0.2) * EWMA(t - 1)$$

$$t = 1, 2, \dots, n \quad (12)$$

Where $Y(t)$ is the corresponding alignment score.

In order to calculate the upper and lower limits, considering formula(4) and (5). Similarly, the standard deviation of historical data is not available, we take the standard deviation of the current data as s . We also let $\lambda = 0.2$, $k = 3$. Therefore formula(4) and (5) transforms as follow

$$ucl = EWMA(0) + 3 * \sqrt{0.2 / (2 - 0.2) * s^2} \quad (13)$$

$$lcl = EWMA(0) - 3 * \sqrt{0.2 / (2 - 0.2) * s^2} \quad (14)$$

3.4 Implementation of CUSUM Charts

To calculate the observation of CUSUM, let $\delta = 1$, formula(6) and (7) could be directly applied.

To calculate the threshold, consider formula(8), let $\alpha = 0.0027$ (equivalent to the plus or minus 3 sigma criteria used in a standard Shewhart chart), $\beta = 0.01$, formula(8) becomes

$$h = 10.809 * k \quad (15)$$

4. Results and Conclusions

The complete results are shown in Appendix II(1), III(1), IV(1).

According to the result, we can observe manually from the alignment score(Appendix II (1)) that several values are deviated from the normal ones(marked in red). Although the corresponding EWMA(Appendix III (1)) and CUSUM(Appendix IV (1)) value reveals such deviation(marked

in red too), however, it does not reach the threshold, and therefore the whole process is in control in regards to both chart. Results are shown in fig.4 and fig.5.

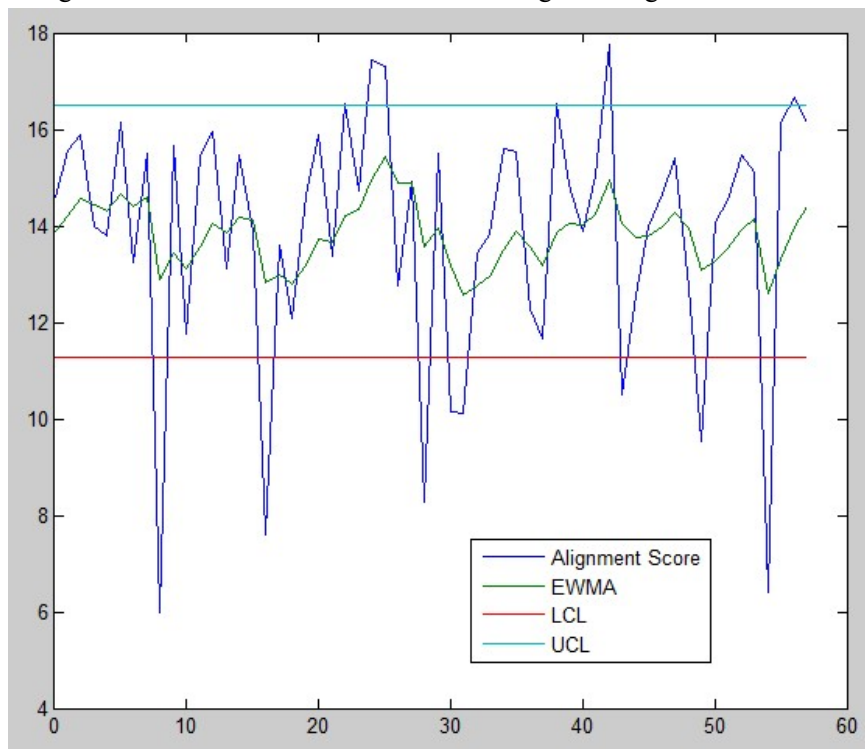


Fig. 4 EWMA(1)

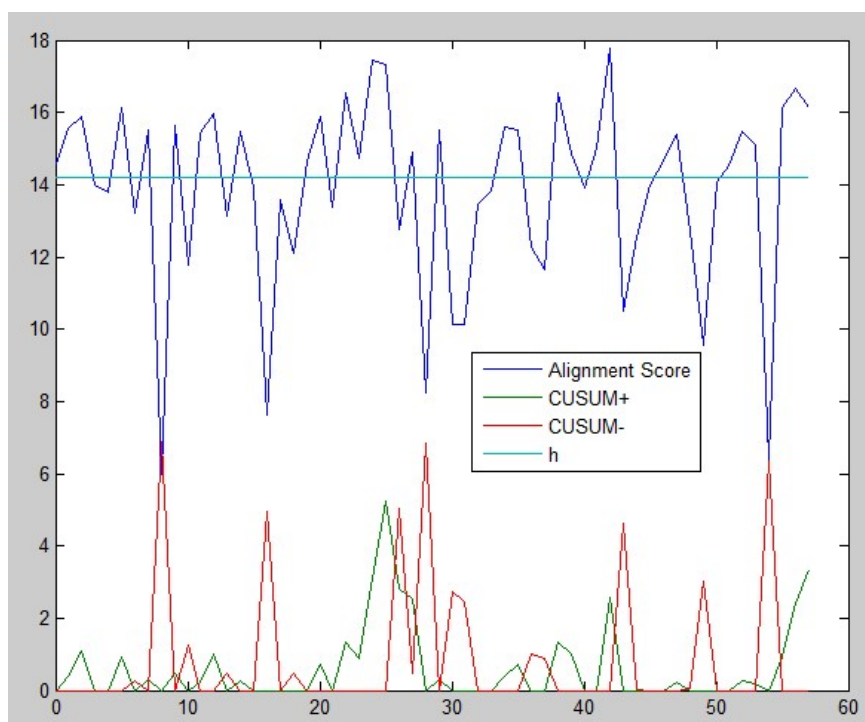


Fig. 5 CUSUM(1)

To figure out how this phenomenon produced, we tried to modify the raw data (in blue brackets) to amplify such deviation. Results are shown in Appendix II(2), III(2), IV(2). We can see that the

deviation becomes more significant (marked in blue). However, threshold, at the same time, increases too, hence no exception prompted. Results are shown in fig.6 and fig.7.

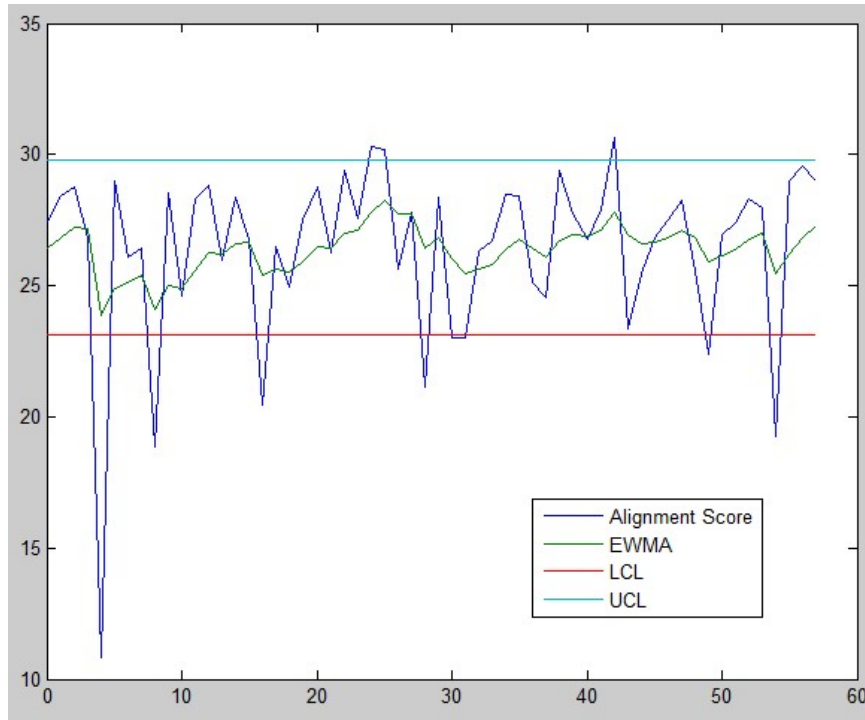


Fig. 6 EWMA(2)

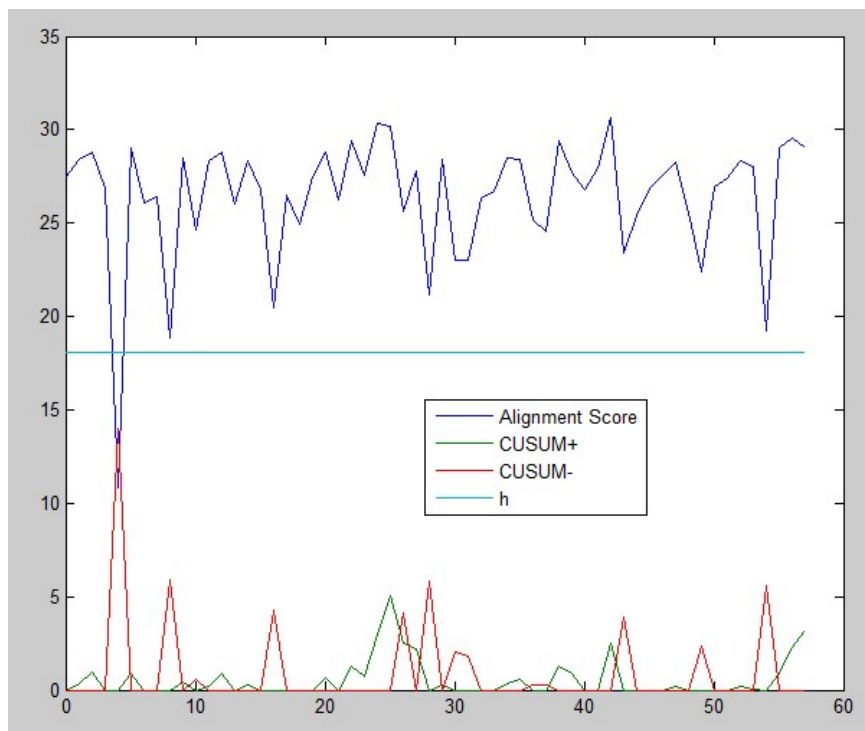


Fig. 7 CUSUM(2)

The reason for such phenomenon is: in definition of both chart, thresholds (formula(4), (5) and formula(8)) are calculated through history mean or history standard deviation data, which,

however, is not available, we have to replace it by current mean or current standard deviation. Such replacement will lead to a result: abnormals in the raw data will in turn affect the threshold.

References

(It's an unexhausted list which only contains the documents that explain the algorithms)

- [1] <http://www.ibm.com/developerworks/library/j-seqalign/#N10385>
- [2] <http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm>
- [3] <http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc323.htm>

APPENDIX

Appendix I: Raw Data

Weeks Expense Item	A	B	C	D	E	F
0	4.84	0	0	0	0	0
1	1.1	0	0	3.63	0	0
2	7.04	0	0	1.87	0	0
3	6.6	0	0	0	3.52	0
4	4.29	0	0	3.63	0	0
5	2.97	0	0	1.87	0	0
6	7.04	0	0	1.87	1.87	0
7	10.34	0	0	1.87	0	0
8	0 (30)	0	1.98	3.63	0	0
9	4.07	0	0	0.66	0	0
10	3.41	0	0	3.63	3.96	0
11	1.1	0	0	3.63	0	0
12	15.84	0	0	0	0	0
13	0	0	0	1.87	0.44	0
14	7.48	0	1.98	3.63	0	0
15	1.32	0	0	0	0	0
16	0	0	0	3.63	0	0
17	9.79	0	0	3.63	3.52	0
18	0	0	0	0	1.87	0
19	10.34	0	1.54	3.63	0	0
20	4.73	8.8	0	0	0	0
21	0	0	0	3.74	0	0
22	6.71	0	1.54	0	2.42	0
23	3.96	0	2.31	0.66	3.52	0
24	0	0	0	1.87	0	0
25	9.46	2.31	0.77	0	0	0
26	1.1	0	0	1.54	0.44	0
27	8.47	0	0.22	1.65	0	0
28	0	0	0	1.54	0	0
29	1.1	0	0	1.65	0	0
30	6.49	2.31	0	1.65	3.96	0
31	0	2.42	0	0	0	0
32	15.07	0	0	0	0.99	0

33	0	0	1.98	1.54	0	0
34	1.1	9.79	0	1.65	0	0
35	7.26	0	0	0	0	0
36	0	0	0	0	3.52	0
37	3.63	0	2.09	0	0	0
38	3.19	0	0	0	0.44	0
39	0.99	0	0	0	0	0
40	7.92	2.97	1.32	1.54	0	0
41	10.23	0	0	1.54	4.62	0
42	2.97	0	0	0	1.65	0
43	0	0	0	0	2.75	0
44	9.35	0	0	0	0	0
45	4.51	0	0	1.54	0	0
46	4.51	0	0	1.54	0	0
47	4.51	0	0	8.8	0	0
48	0	0	2.09	0	0	0
49	1.1	0	0	3.19	0	0
50	2.75	2.42	0	0.66	0	0
51	1.1	0	0	0	0	0
52	6.16	0	1.32	0	2.42	0
53	14.19	0	0	0	1.21	0
54	0	0	0	0	3.52	0
55	2.86	2.42	0	0	1.21	0
56	0	0	0	0	1.21	0
57	1.1	0	2.09	0	0	0
58	13.97	0	0	0	1.21	0
59	1.76	2.42	0	0	0	0
60	1.1	0	0	0	1.21	0
61	0	0	0	0	0	0

Appendix II: Alignment Score

II (1)	
0	14.58172473
1	15.57172473
2	15.90172473
3	13.97871801
4	13.79191318
5	16.13645717
6	13.22831192
7	15.5066861
8	5.983234584
9	15.65012157

10	11. 75783248
11	15. 44086652
12	15. 95755826
13	13. 11910182
14	15. 48277301
15	13. 89042341
16	7. 607581877
17	13. 59027762
18	12. 09632166
19	14. 64106985
20	15. 90172473
21	13. 37309995
22	16. 54188735
23	14. 71538706
24	17. 44172473
25	17. 31818311
26	12. 77307492
27	14. 90327568
28	8. 258791432
29	15. 50401664
30	10. 13959528
31	10. 1190128
32	13. 44941054
33	13. 8270321
34	15. 5910597
35	15. 52815615
36	12. 27722749
37	11. 66895487
38	16. 54188735
39	14. 84895208
40	13. 90135007
41	15. 03938142
42	17. 77172473
43	10. 51172473
44	12. 56789922
45	13. 98350615
46	14. 65268642
47	15. 40992457
48	12. 72010123
49	9. 544489475
50	14. 08385272
51	14. 54401657
52	15. 46172473
53	15. 11801036

54	6. 401214631
55	16. 13645717
56	16. 67172473
57	16. 13645717

II (2)

0	27. 45139194
1	28. 44139194
2	28. 77139194
3	26. 84838522
4	10. 80371062
5	29. 00612438
6	26. 09797913
7	26. 44117766
8	18. 85290179
9	28. 51978877
10	24. 62749969
11	28. 31053373
12	28. 82722546
13	25. 98876902
14	28. 35244021
15	26. 76009061
16	20. 47724908
17	26. 45994482
18	24. 96598887
19	27. 51073706
20	28. 77139194
21	26. 24276715
22	29. 41155455
23	27. 58505426
24	30. 31139194
25	30. 18785032
26	25. 64274212
27	27. 77294288
28	21. 12845864
29	28. 37368385
30	23. 00926249
31	22. 98868
32	26. 31907774
33	26. 6966993
34	28. 4607269
35	28. 39782335

36	25.14689469
37	24.53862208
38	29.41155455
39	27.71861928
40	26.77101728
41	27.90904862
42	30.64139194
43	23.38139194
44	25.43756642
45	26.85317336
46	27.52235363
47	28.27959178
48	25.58976844
49	22.41415668
50	26.95351992
51	27.41368377
52	28.33139194
53	27.98767756
54	19.27088184
55	29.00612438
56	29.54139194
57	29.00612438

Appendix III: EWMA

III (1)	
UCL:	16.50804557176
LCL:	11.26068281698
0	13.88436419
1	14.2218363
2	14.55781399
3	14.44199479
4	14.31197847
5	14.67687421
6	14.38716175
7	14.61106662
8	12.88550021
9	13.43842448
10	13.10230608
11	13.57001817
12	14.04752619
13	13.86184131
14	14.18602765
15	14.1269068

16	12. 82304182
17	12. 97648898
18	12. 80045552
19	13. 16857838
20	13. 71520765
21	13. 64678611
22	14. 22580636
23	14. 3237225
24	14. 94732294
25	15. 42149498
26	14. 89181097
27	14. 89410391
28	13. 56704141
29	13. 95443646
30	13. 19146822
31	12. 57697714
32	12. 75146382
33	12. 96657747
34	13. 49147392
35	13. 89881036
36	13. 57449379
37	13. 193386
38	13. 86308627
39	14. 06025943
40	14. 02847756
41	14. 23065833
42	14. 93887161
43	14. 05344224
44	13. 75633363
45	13. 80176814
46	13. 97195179
47	14. 25954635
48	13. 95165733
49	13. 07022376
50	13. 27294955
51	13. 52716295
52	13. 91407531
53	14. 15486232
54	12. 60413278
55	13. 31059766
56	13. 98282307
57	14. 41354989

III (2)	
UCL:	29.7948979965309
LCL:	23.09961151465511
0	26.44725476
1	26.84608219
2	27.23114414
3	27.15459236
4	23.88441601
5	24.90875768
6	25.14660197
7	25.40551711
8	24.09499405
9	24.97995299
10	24.90946233
11	25.58967661
12	26.23718638
13	26.18750291
14	26.62049037
15	26.64841042
16	25.41417815
17	25.62333149
18	25.49186296
19	25.89563778
20	26.47078861
21	26.42518432
22	27.02245837
23	27.13497755
24	27.77026043
25	28.2537784
26	27.73157115
27	27.73984549
28	26.41756812
29	26.80879127
30	26.04888551
31	25.43684441
32	25.61329108
33	25.82997272
34	26.35612356
35	26.76446352
36	26.44094975
37	26.06048422

38	26.73069828
39	26.92828248
40	26.89682944
41	27.09927328
42	27.80769701
43	26.922436
44	26.62546208
45	26.67100434
46	26.8412742
47	27.12893771
48	26.82110386
49	25.93971442
50	26.14247552
51	26.39671717
52	26.78365213
53	27.02445721
54	25.47374214
55	26.18021859
56	26.85245326
57	27.28318748

Appendix IV: CUSUM

IV(1)

$h = 14.179686004118732$

CUSUM+:		CUSUM-:	
0	0	0	0
1	0.37551985	1	0
2	1.081039701	2	0
3	0	3	0
4	0	4	0
5	0.940252288	5	0
6	0	6	0.284463875
7	0.310481219	7	0
8	0	8	6.899770141
9	0.453916683	9	0
10	0	10	1.268607706
11	0.244661639	11	0
12	1.006015014	12	0
13	0	13	0.459436703
14	0.286568125	14	0
15	0	15	0
16	0	16	4.964941628

17	0	17	0
18	0	18	0.476201842
19	0	19	0
20	0.70551985	20	0
21	0	21	0
22	1.345682463	22	0
23	0.864864635	23	0
24	3.110384486	24	0
25	5.232362717	25	0
26	2.80923275	26	5.031811307
27	2.516303542	27	0.47848058
28	0	28	6.830035616
29	0.307811756	29	0
30	0	30	2.740739982
31	0	31	2.45351071
32	0	32	0
33	0	33	0
34	0.394854814	34	0
35	0.726806079	35	0
36	0	36	1.022102098
37	0	37	0.903568635
38	1.345682463	38	0
39	0.998429657	39	0
40	0	40	0
41	0	41	0
42	2.57551985	42	0
43	0	43	4.636318623
44	0	44	0.004624288
45	0	45	0
46	0	46	0
47	0.213719689	47	0
48	0	48	0.066141963
49	0	49	3.028034031
50	0	50	0
51	0	51	0
52	0.26551985	52	0
53	0.187325323	53	0
54	0	54	6.358634198
55	0.940252288	55	0
56	2.415772138	56	0
57	3.356024427	57	0

IV(2)

h = 18.092337895648846

CUSUM+:		CUSUM-:	
0	0	0	0
1	0.320315564	1	0
2	0.970631127	2	0
3	0	3	0
4	0	4	13.96972251
5	0.885048001	5	0
6	0	6	0
7	0	7	0
8	0	8	5.920531345
9	0.398712396	9	0
10	0	10	0.544645842
11	0.189457353	11	0
12	0.895606441	12	0
13	0	13	0
14	0.231363838	14	0
15	0	15	0
16	0	16	4.296184052
17	0	17	0
18	0	18	0
19	0	19	0
20	0.650315564	20	0
21	0	21	0
22	1.290478176	22	0
23	0.754456062	23	0
24	2.944771625	24	0
25	5.01154557	25	0
26	2.533211316	26	4.142236583
27	2.185077821	27	0
28	0	28	5.830052319
29	0.25260747	29	0
30	0	30	2.016778119
31	0	31	1.784753133
32	0	32	0
33	0	33	0
34	0.339650527	34	0
35	0.616397505	35	0
36	0	36	0.242935948
37	0	37	0.234811058
38	1.290478176	38	0
39	0.888021083	39	0

40	0	40	0
41	0	41	0
42	2. 520315564	42	0
43	0	43	3. 912356759
44	0	44	0
45	0	45	0
46	0	46	0
47	0. 158515403	47	0
48	0	48	0
49	0	49	2. 359276454
50	0	50	0
51	0	51	0
52	0. 210315564	52	0
53	0. 076916749	53	0
54	0	54	5. 579468048
55	0. 885048001	55	0
56	2. 305363565	56	0
57	3. 190411566	57	0