Advanced Science and Technology Letters
Vol.120 (HSDI 2015), pp.830-834
http://dx.doi.org/10.14257/astl.2015.120.163

Variable Window Operation Method to Respond to Context Reasoning

Donghyeok Seoh, Younghwan Oh[1]

[1]Korea Nazarene University, 48, Wolbong-Ro, Seobuk-Ku
Cheonan, Korea. 31172
{absuh@nsu.ac.kr}, [1]{yhoh@kornu.ac.kr}

Abstract. This study proposes a method of determining a window size for data stream flexibly and variably. In the proposed method, a sliding window size is controlled variably depending on the evaluation of the event detected and reported by a sensor. Therefore, it is necessary to make criteria to judge rapid changes in sensing values and critical events, and to come up with a reasonable interoperability scheme in consideration of the lapse of time, and the development and declination of event context. This study tries to find a reasonable sliding window size control method interacting with context and thereby to contribute to the research on flexible and efficient context reasoning and context awareness.

Keywords: Sliding Window, Data Stream Analysis, Context Reasoning.

1 Introduction

These days, there have been studies on how to determine a window size in the sliding window technique. How to determine a window size, or a time interval, greatly influences efficient analysis and process of data. If a window size becomes too large, it is possible to obtain enough data and the materials necessary for analysis in a given time, but it is impossible to deal with continuously flowing data efficiently. If a window size becomes too small, it is required to make calculates often and thus it is impossible to obtain enough data to analyze in calculation. Accordingly, it is essential to determine a proper window size to process data stream efficiently.

This study proposes a method to determine a window size for data stream flexibly and variably. The purpose of sensor and network operation is to obtain real-world context information through sensors. Therefore, this study tries to find a method to determine a window size on the basis of context. In the proposed method, after noticeable events are detected, a window size turns small, and when such events decreased, the window size gradually gets back to its original size. For the method, it is necessary to make criteria to judge rapid changes in sensing values and critical events, and to come up with a reasonable interoperability scheme in consideration of

---

[1] Corresponding author.

ISSN: 2287-1233 ASTL

the lapse of time, and the development and declination of event context. Therefore, this study tries to find a reasonable sliding window size control method interacting with context and thereby to contribute to the research on flexible and efficient context reasoning and context awareness.

2 Variable window operation method to respond to context reasoning

It is assumed that each sensor data comes in with the lapse of time, and its risk value calculated also comes in. If the risk value increases, a window size needs to become smaller to check the context change. If the risk value decreases or stands still, a window size needs to become larger to process data efficiently.
However, an excessive rise in a window size causes mass data to be saved in a memory continuously. Therefore, it is necessary to have important data of a stream left in the memory and remove other data from the memory.

A window size is defined as follows:
In this study, a data stream as a raw data is considered to be a sequence of sensor data, and to process them, windows are considered to come in at a proper time interval. In addition, it is assumed that a data is already processed and has the calculated risk value. Therefore, sensor data are not taken into account, but risk values in windows are handled. In other words, a data stream *DS* is $DS = < W_{1,} W_{2,} \cdots, W_{n,} >$, where W means a window.

The window of the unit time length b-a+1 is $W_{n,}$ (n^{th} window)
For $W_{n,} = < D_{a,} D_{a+1,} \cdots, D_{b-1,} D_b >$, the size of window function S is defined as the difference between the first data time and the last data time when the data in a window are sorted out in the time sequence.

$$S(W_n) = a - b$$

Now, a window size can be determined with the use of the change in the risk values calculated in real time.
At the time of 1 second, data start to be analyzed. Therefore, the initial window size is set to 1.
The risk value coming in real time is already normalized, and thus it is assumed to be set to a value between 0 and 3.
If the risk value increases, the window size is updated quickly. Risk factors need to be checked more frequently.
The risk change is defined as

$$D'_k = D_k - D_{k-1} (D_1 = 0)$$

It means the difference between the risk in the previous time slot and the risk at the present time.

$$\text{if } D'_k > 0 \vee D_k > \delta \rightarrow S(W_{n+1}) = S(W_n) - 1$$

If the risk change tends to increase, or if the risk value exceeds a certain criterium, a size of the next window is decreased.
Here, defined is $\delta = 1.5$. The maximum risk value is 3. Therefore, if it exceeds 1.5, half the maximum value, risk factors are considered to influence context greatly.

$$\text{if } D'_k \leq 0 \rightarrow S(W_{n+1}) = S(W_n) + 1$$

Here, a window size is increased or decreased in real time. In other words, a size of the next window is changed by the change in the risk detected in the current window, and therefore the size continues to be changed while the current windows are in process. If one risk is reported at one second, the limit of a window size is presented as follows:

$$1 \leq S(W) \leq 30$$

Holding a data of more than 30 seconds in one window causes the burden of memory use. Therefore, a window size should not be set to more than 30 seconds. Meanwhile, if it is assumed that one risk is reported at one second, a window size should not be set to less than 1 second. If a window size is set to less than 1 second, there might be a window that fails to read data. To prevent that, a window size should not be set to less than 1 second.

3 Experiment

The input was the value calculated in the way of multiplying the risk value by 3.

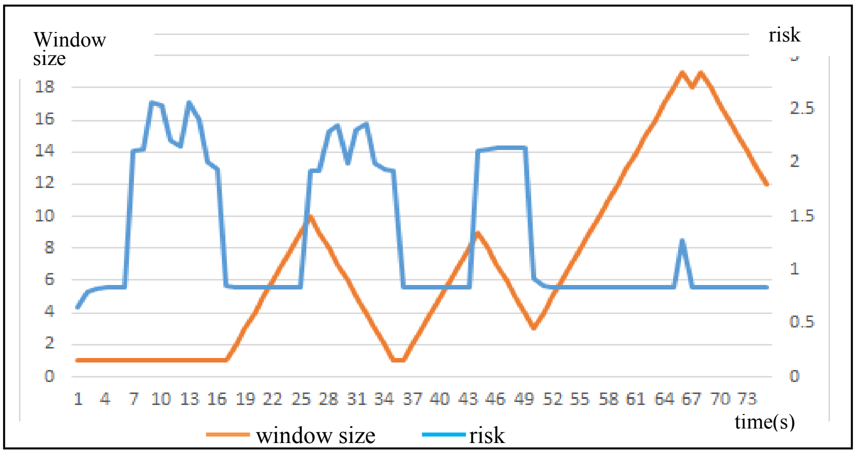

Fig.1 window size by risk value

The results are presented in [Fig. 1].

As shown in the graph, when the risk value increases or becomes more than 1.5, the window size decreases, and when the risk value decreases or stands still, the window size increases.

The input risk values and the output window sizes are presented in [Table 1]

Table 1. window size by input risk values

risk	window size	risk	window size	risk	window size
0.6588	1	0.8351	9	2.1331	4
0.7960	1	1.9229	10	0.9226	3
0.8264	1	1.9229	9	0.8544	4
0.8332	1	2.2877	8	0.8395	5
0.8347	1	2.3566	7	0.8361	6
0.8351	1	1.9989	6	0.8354	7
2.1130	1	2.3082	5	0.8352	8
2.1315	1	2.3592	4	0.8352	9
2.5618	1	1.9993	3	0.8352	10
2.5377	1	1.9386	2	0.8352	11
2.2025	1	1.9263	1	0.8352	12
2.1472	1	0.8369	1	0.8352	13
2.5663	1	0.8356	2	0.8352	14
2.4045	1	0.8353	3	0.8352	15
2.0106	1	0.8352	4	0.8352	16
1.9411	1	0.8352	5	0.8352	17
0.8444	1	0.8352	6	0.8351	18
0.8372	2	0.8352	7	1.2750	19
0.8356	3	0.8351	8	0.8352	18
0.8353	4	2.1130	9	0.8352	19
0.8352	5	2.1315	8	0.8352	18
0.8352	6	2.1330	7	0.8352	17
0.8352	7	2.1331	6	0.8352	16
0.8352	8	2.1331	5	0.8352	15

4 Conclusion

This study proposed a method to determine a window size for data stream flexibly and variably. The purpose of sensor and network operation is to obtain real-world context information through sensors. Therefore, this study found a method to determine a window size on the basis of context. In the proposed method, after noticeable events are detected, a window size turns small, and when such events decreased, the window size gradually gets back to its original size. For the method, this study set forth the criteria to judge rapid changes in sensing values and critical events, and suggested a reasonable interoperability scheme in consideration of the lapse of time, and the development and declination of event context. Therefore, this study tried to contribute to the research on flexible and efficient context reasoning and context awareness.

Acknowledgments. This research was supported by Korea Nazarene University Research Grants 2015.

References

1. Tao, Yufei, and Dimitris P.,: Maintaining sliding window skylines on data streams." Knowledge and Data Engineering, IEEE Transactions on 18.3, 377-391 (2006).
2. Lee C.H., Lin C.R., and Chen M.S.,:Sliding-window filtering: an efficient algorithm for incremental mining.: Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001.
3. Arasu, Arvind, and Jennifer W.,:Resource sharing in continuous sliding-window aggregates.: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004.
4. Wyner, Aaron D., and Jacob Z.,:The sliding-window Lempel-Ziv algorithm is asymptotically optimal.: Proceedings of the IEEE 82.6 ,872-877(1994).
5. Gu, Irene Y.H., and Math HJ B.,:Estimating interharmonics by using sliding-window ESPRIT.: Power Delivery, IEEE Transactions on 23.1, 13-23 (2008)
6. Krämer, Jürgen, and Bernhard S.,:Semantics and implementation of continuous sliding window queries over data streams.: ACM Transactions on Database Systems (TODS) 34.1 (2009)
7. Fares, Mario A.,:A sliding window-based method to detect selective constraints in protein-coding genes and its application to RNA viruses.: Journal of molecular evolution 55.5, 509-521 (2002)

