Find the Right Transaction Length for Stream Mining : A Distance Approach

Jie Deng†, Zhiguo Qu*, Yongxu Zhu†, Gabriel-Miro Muntean*and Xiaojun Wang*

The Rince Institute,
Dublin City University,
Ireland

E-mail: †{jie.deng3,zhu.zhuyonz2}@mail.dcu.ie
*{zhiguo.qu,munteang,xiaojun.wang}@dcu.ie

Abstract — **Stream data mining has drawn people's attention for the last decade. Different algorithms have been proposed and applied in different areas. Most of the stream data mining algorithms are use a sliding window to cache the stream during mining. Most research have been focused on statically or dynamically generate the sliding window, yet the proper selection of the transaction length have not been addressed. Transaction length decides the length the pattern found in a stream and affect the mining processing time as well. This paper proposed a distance method to evaluate the proper transaction length value in mining process. Experiment demonstrated that this method could successfully find the pattern length in emulated telecommunication stream data. By using this method in data pre-processing, it could find a suitable transaction length value for the mining process which could make mining more efficient therefore improve the performance.**

Keywords — **Stream data mining; Sequential pattern mining; transaction length; data mining parameter**

I INTRODUCTION

Data stream mining is becoming more and more common these days as data mining technology is deeply used in various industries. Web analysis, fraud detection, network traffic analysis and other applications require a dynamic processing ability which makes stream mining so important. Finding patterns from data is not the only target any more, how to finding patterns in limited time has becoming the most difficult problem.

What makes stream different from traditional dataset is that it is unrealistic to keep all the data in the memory or even storage. Traditional data mining algorithm like Apriori or PrifixSpan always need to load the whole dataset into memory, and count frequency from the snapshot of the dataset. In the situation that data elements arrival online[1] and data stream are potentially unbound in size[2], the mining process will never has control to the whole dataset[3].

To deal with these attributes of stream, different approach have been proposed including sampling of data, histogram and wavelet transform[4]. The most common stream mining algorithm is a sliding window approach[5, 6], and using a tree structure to represent the item[7]. A sliding window method could take a snapshot of the stream and using mining process in memory[8, 9, 10]. The dynamic adjustment of window has also been proposed[11, 12, 13] for a more efficient use of memory.

Recently, integrate stream mining algorithm with different applications are become common, such as Twitter analysis[14], web access analysis[15], and even stock market analysis[16]. Though few paper proposed a parameter free[17, 18] or context aware[19, 20] approach to improve stream mining process, the transaction length setting problem

still have not been addressed.
In this paper, we first show how important transaction length affects the result and the mining time of stream mining process in section II. Further more, we propose a distance measurement method to predict the most suitable transaction length setting for the mining process, and show how this distance measurement method works with emulated telecommunication data in section III.

II IMPACT OF TRANSACTION LENGTH

Transaction is the basic processing unit in data mining algorithm. The length of transaction indicates how many items contained in each transaction. In traditional dataset, transaction length is often defined by the inner attributes of each item inside a transaction, that is to say, all the items in a transaction definitely share the same attribute. For example, in traditional retail market mining, the transaction is defined by costumer shopping recodes.
In stream data mining however, the transaction boundary does not come alone with the dataset. This is because the items come in a stream do not contains extra attribute for classification, and mining process do not have time to inspect the attributes as well. The most common processing unit algorithms focus on deciding the most appropriate sliding window size instead, with fixed or flexible window size[21].
To show that transaction length is also important to mining process, we focus on the follow three aspects to evaluate how transaction length affect the mining result and the length of time taken to get the mining result.

a) Transaction length and patterns

It is obvious that the longer patterns can be found in a transaction with more items contained in it. When more items are contained in a transaction, more kinds of combination of itemsets could be formed. With same window size, a longer transaction length will always have the possibility to find more candidates. For the algorithm like PrefixSpan which uses a projection method to reduce the search space, a large transaction could lead to a larger projected dataset which eventually will get longer patterns.
An example datasets are defined to illustrate the impact of transaction length on the mining result.The pattern in the dataset is of length 7. By running PrefixSpan with an increasing transaction length, we can see that the number of meaningful patterns found always come to peak when transaction length equals to the pattern length or integer multiples the pattern length, as shown in figure 1. The same trend can be observed from the ratio of meaningful pattern to the number of whole result set, as shown in figure2.

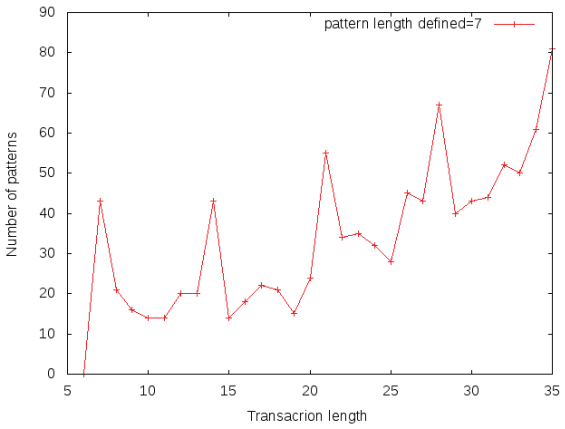

Fig. 1: The number of patterns found in dataset (pattern length 7).

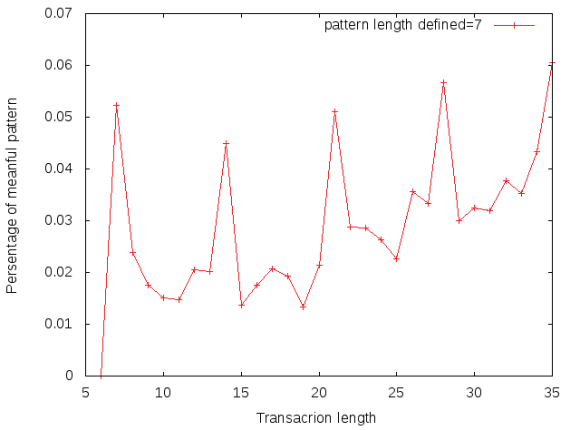

Fig. 2: The ratio of meaningful patterns found in dataset.

b) Transaction length and processing time

However, with the benefit of finding more patterns and longer patterns, the larger transaction have to pay for it with longer mining time. Using the PrefixSpan algorithm for example, the projected database contains more dataset which means more rounds of mining are needed. Each mining round is a recursion of finding frequent items and projected dataset. So it is easy to understand that a larger transaction will cost PrefixSpan longer to finish.
As stream mining algorithms have a requirements of processing time[13], we also need to discover how transaction length affects the processing time. To illustrate the relationship between transaction length and processing time, we compare the processing time from transaction length of 6 to 15 as shown in figure 3. The processing time increasing exponentially with increase transaction length.

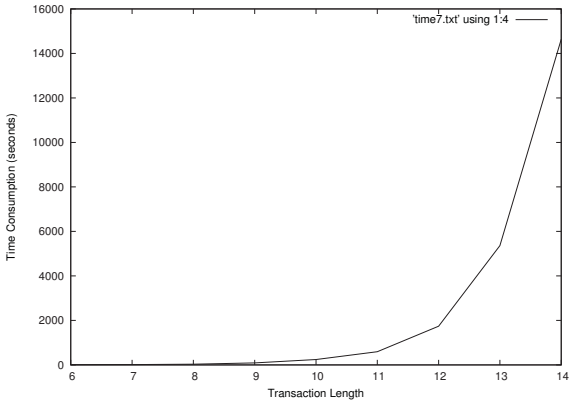

Fig. 3: The processing time of example dataset.

III DISTANCE APPROACH

For a mining algorithm to find patterns in a dataset, the frequency of items or itemsets are counted. An itemset must appear repeatedly in the stream to be regarded as a pattern. That is to say, as long as we understand how the itemsets repeat in the dataset, we can vaguely predict how patterns will appear.

Statistically, the distribution of items in dataset could indicate some features about the data. But the distribution of items could only shows how each item behaviours in dataset instead of correlation ration between each items. Usually, distance is used to measure how items are connected with each other[22].

The concept of distance here is the number of items between the two appearance of the same item.

$$\text{distance} = Position_{a2} - Position_{a1} \tag{1}$$

By measuring the appearance distance of the item, the value will show how intensively the item appears in the dataset. More importantly, we found that if the items belongs to a pattern, then the mean distance value (formula 2) tends to be the same. That gives us an hint that by cluster the mean distance of each items, we can get the members which belongs to a pattern.

$$\text{mean distance} = \frac{\sum distance_a}{\text{total number of item a}} \tag{2}$$

To verify this finding, we take different group of experiments to show how exactly the patterns match the distance attribute. The dataset is generated by OpenMSC[23], an open source mscgen-based(Message Sequence Charts) mobile network trace file generator. After defining a telecommunication protocol in msc file, this generator can emulate the communication process, and output event streams represented by integers. The use-case defined in this experiment is a 7-step communication process between UE(user equipment) and BS(base station). Each BS could handle multiple UEs, so there are $NumberofUE \times NumberofBS \times StepofProcess$ patterns in total.

a) Distribution functions

The OpenMSC provides different distributions to select. The distributions are used to determine the UE starting times as well as for the latencies between each communication descriptor. As possibility distributions are introduced into the data generate process, the distance between each individual items are not fixed any more. And position of patterns is formed randomly as well, which is more close to real world scenarios.

The first distribution tested is exponential, the results show that different items have different mean distance value, but items belonging to same pattern trend to have very close distance values, shows in figure 4. Each short line consists of seven item points which in the same level. That means by clustering these mean distance values, we can find the number of items in a pattern which is seven in this case. Using Gaussian and Uniform distribution funcations, the same trnd is found as illustrated in figure 5 and figure 6.

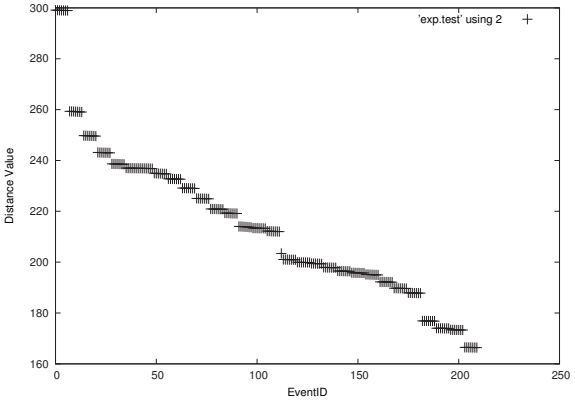

Fig. 4: The mean distance of items with exponential distribution.

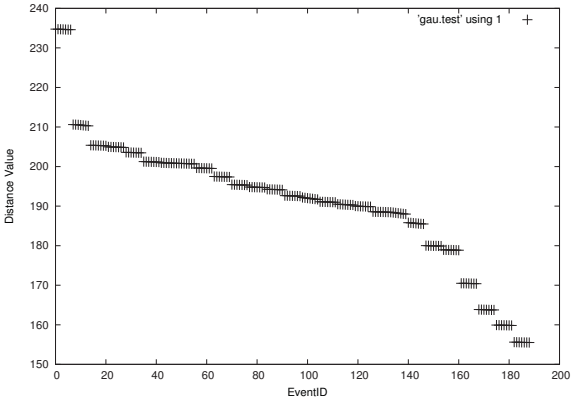

Fig. 5: The mean distance of items with gaussian distribution.

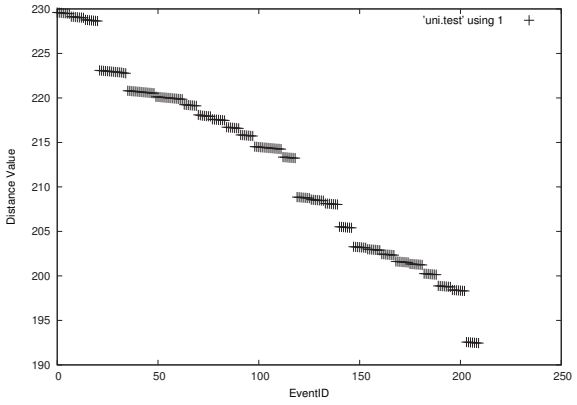

Fig. 6: The mean distance of items with uniform distribution.

Simulation results obtained using the three different distribution funcations all procide rough indications on how manr patterns in a dataset, and how mant items in these patterns. While data mining algorithms need to find the exact patterns, the distance information indicates the minimum transaction length required, which is very useful in deciding the sliding windows during the data mining process.

b) Noise detach

To push this method further, we want to find out how this method perform when dealing with a more complex situation, in which a trace file contains not only patterns, but also noise. The noise is items irrelevant to patterns and only waste resource in data mining process. It is a great help if noise can be removed before mining. So another experiment is taking to test how this method detects the noise from dataset.

The result in figure 7 shows the noise do not have

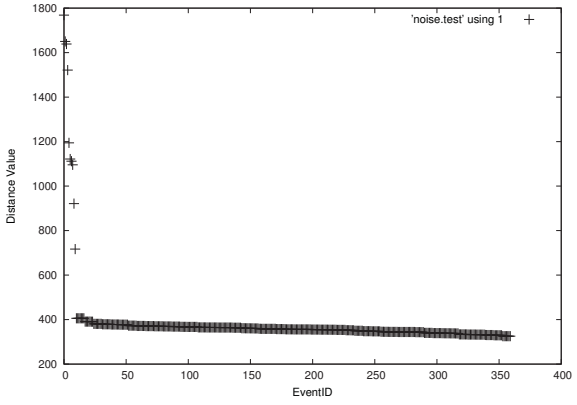

Fig. 7: The mean distance of items and noise.

the same curve as patterns do. As items in patterns trend to appear more regular and more frequent, so it is clear to distinguish infrequent noise from frequent pattern. For those noise which appear as frequent as patterns, can be classified into meaningless items, and no way to distinguish them statistically.

IV CONCLUSION

Stream mining algorithms are widely used in different areas. However, with the improvement of algorithm processing time, few research have been put on the parameters setting of mining process. In this paper, we discussed one of the most important parameter in stream data mining process: transaction length. By showing how transaction length affect the result and mining process of a emulated telecommunication network trace file, we can clearly see the influence of transaction length on pattern length and processing time. Further more, we propose a distant measurement method to predict the most suitable transaction length setting for mining process. By using this method as a pre-processing step of mining process, we can separate noise from data, and give a suitable parameter setting for the mining process, which will increase the meaningful patterns and reduce mining time.
As the concept of this method is based on locating frequently appeared itemsets, the drawback is that the method could no longer useful to find the distance attribute when frequent random noise and burst data stream interfered into the pattern. When variable length random noise are mixed into the patterns, there is no way this method can still put correlated itemsets together. Though stream with different distribution have been tested through this method, we still could not count on the stream in real world will exhibit the same overall distribution.
In the future, we hope to deploy this method on different applications to deal with different stream. Though this is a generic approach to infer the parameter setting, different stream with different feature will somehow have specific requirements on the parameter. Besides, we have not tested how could this method handle large transaction length as we are restricted to the telecommunication data generator and the computing ability of PrefixSpan algorithm.

V Acknowledgements

This work is funded by Enterprise Ireland Innovation Partnership Programme with Ericsson Ireland under grant agreement IP/2011/0135 [24].

References

[1] Jiandong Huang and JA Stankovic. Experimental evaluation of real-time transaction processing. *Real Time Systems ...*, 1989.

[2] Lukasz Golab and MT Özsu. Issues in data stream management. *ACM Sigmod Record*, 32(2):5–14, 2003.

[3] Brian Babcock, S Babu, and M Datar. Models and issues in data stream systems. *... of database systems*, pages 1–30, 2002.

[4] ZG Qu, XX Niu, and J Deng. Frequent itemset mining over stream data: Overview. *... (IETICT 2013), IET ...*, 2013.

[5] MM Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.

[6] Barzan Mozafari, Hetal Thakkar, and Carlo Zaniolo. Verifying and mining frequent patterns from large windows over data streams. *Data Engineering, 2008. ...*, 2008.

[7] James Cheng, Yiping Ke, and Wilfred Ng. A survey on algorithms for mining frequent itemsets over data streams. pages 1–27, 2008.

[8] Yun Chi, Haixun Wang, PS Yu, and RR Muntz. Moment: Maintaining closed frequent itemsets over a stream sliding window. *Data Mining, 2004. ICDM'04 ...*, 2004.

[9] M Datar, A Gionis, P Indyk, and R Motwani. Maintaining stream statistics over sliding windows. *SIAM Journal on Computing*, 4, 2002.

[10] Hua-fu Li, Chin-chuan Ho, and Suh-yin Lee. Incremental updates of closed frequent itemsets over continuous data streams. *Expert Systems With Applications*, 36(2):1466–1477, 2009.

[11] Toon Calders, N Dexters, and B Goethals. Mining frequent items in a stream using flexible windows. *Intelligent Data Analysis*, pages 1–14, 2008.

[12] Hua-fu Li and Suh-yin Lee. Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems With Applications*, 36(2):1466–1477, 2009.

[13] Michael Leonard and Brenda Wolfe. Data Mining and Predictive Modeling Mining Transactional and Time Series Data Data Mining and Predictive Modeling. pages 1–26, 2001.

[14] Balakrishnan Gokulakrishnan, Pavalanathan Priyanthan, Thiruchittampalam Ragavan, Nadarajah Prasath, and AShehan Perera. Opinion mining and sentiment analysis on a Twitter data stream. In *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*, pages 182–188. IEEE, December 2012.

[15] Nan Jiang and Le Gruenwald. Research issues in data stream association rule mining. *ACM Sigmod Record*, 35(1), 2006.

[16] Xiaoyan Liu, Xindong Wu, Huaiqing Wang, Rui Zhang, James Bailey, and Kotagiri Ramamohanarao. Mining distribution change in stock order streams. *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 105–108, 2010.

[17] Albert Bifet and Ricard Gavald. Adaptive Parameter-free Learning from Evolving Data Streams.

[18] Jimeng Sun and C Faloutsos. Graphscope: parameter-free mining of large time-evolving graphs. *... discovery and data mining*, 2007.

[19] Conny Junghans, Marcel Karnstedt, and Michael Gertz. Quality-driven resource-adaptive data stream mining? *ACM SIGKDD Explorations Newsletter*, 13(1):72, August 2011.

[20] P Haghighi, M Gaber, Shonali Krishnaswamy, and A Zaslavsky. An architecture for context-aware adaptive data stream mining. 2007.

[21] Toon Calders, Nele Dexters, and Bart Goethals. Mining Frequent Itemsets in a Stream. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 83–92, October 2007.

[22] Daniel Kifer, S Ben-David, and Johannes Gehrke. Detecting change in data streams. *... on Very large data bases-Volume 30*, pages 180–191, 2004.

[23] Sebastian Robitzsch. OpenMSC - An Open Source MSCgen-Based Control Plane Trace Emulator for Communication Networks, 2014.

[24] Dublin City University and Ericsson. E-Stream Project.