When to Update the Sequential Patterns of Stream Data?

Qingguo Zheng Ke Xu Shilong Ma

National Lab of Software Development Environment
Department of Computer Science and Engineering
Beijing University of Aeronautics and Astronautics, Beijing 100083
{zqg, kexu, slma }@nlsde.buaa.edu.cn

Abstract. In this paper, we first define a difference measure between the old and new sequential patterns of stream data, which is proved to be a distance. Then we propose an experimental method, called TPD (Tradeoff between Performance and Difference), to decide when to update the sequential patterns of stream data by making a tradeoff between the performance of increasingly updating algorithms and the difference of sequential patterns. The experiments for the increasingly updating algorithm IUS on the alarm data show that generally, as the size of incremental windows grows, the values of the speedup and the values of the difference will decrease and increase respectively. It is also shown experimentally that the incremental ratio determined by the TPD method does not monotonically increase or decrease but changes in a range between 20 and 30 percentage for the IUS algorithm.

1 Introduction

To enhance the performance of algorithms of data mining, many researchers [1,2,3,4,5,6,7] have focused on increasingly updating association rules and sequential patterns. But if we update association rules or sequential patterns too frequently, the cost of computation will increase significantly. For the problem above, Lee and Cheung [8] studied the problem "Maintenance of Discovered Association Rules: When to update?" and proposed an algorithm called DELL to deal with it. The important problem about "When to update" is to find a suitable distance measure between the old and new association rules. In [8], the symmetric difference was used to measure the difference of association rules. But Lee and Cheung only considered the difference of association rules, and did not consider that the performance of increasingly updating algorithms will change with the size of added transactions. Ganti et al. [9,10] focused on the incremental stream data mining model maintenance and change detection under block evolution. However, they also didn't consider the performance of incremental data mining algorithms for the evolving data.

Obviously, with the increment of the size of incremental windows, the performance of increasingly updating algorithms will decrease. If the difference

between the new and old sequential patterns is too high, the size of increasingly window will become too large, therefore the performance of increasingly updating algorithm will be reduced greatly. On the other hand, if the difference is too small, the increasingly updating algorithm will update the old sequential patterns very frequently, which will also consume too many computing resource. In all, we must make a tradeoff between the performance of the updating algorithms and the difference of the sequential patterns. In this paper, we use the speedup as a measure of increasingly updating algorithms and define a metric distance as the difference measure to detect the change of the sequential patterns of stream data. Based on those measures, we propose an experimental method, called TPD (Tradeoff between Performance and Difference), to estimate the suitable range of the incremental ratio of stream data by making a tradeoff between the performance of the increasingly updating algorithm and the difference of sequential patterns.

By the TPD method, we estimate the suitable range of incremental ratio of stream data for the increasingly updating algorithm IUS [12]. The experiments on the alarm data in Section 4 show that generally, as the size of incremental windows grows, the values of the speedup and the values of the difference will decrease and increase respectively. By the experiments, we can discover that as the size of original windows increases, the incremental ratio determined by TPD method does not monotonically increase or decrease but changes in a range between 18 and 30 percentage for the IUS algorithm.

2 Problem Definition

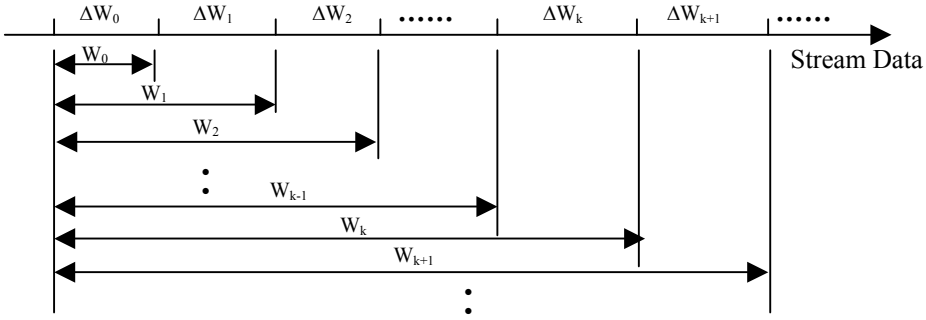

Fig. 1. Sliding stream viewing windows on the stream queue

1. Definitions of initial window, incremental window and incremental ratio
Given stream view window W_i (i=0,1,2,3, …), ΔW_i (i=0,1,2,3, …) is called **incremental window**, iff i=0, ΔW_0 is called **initial window**, where $W_{i+1}=W_i+\Delta W_{i+1}$ (i=0,1,2,3, …). The ratio of the size of incremental window to that of the initial window, i.e. $|\Delta W_1|/|W_0|$, is called **incremental ratio** of stream data.

2. Definition of difference measure between the old and new sequential patterns
Before updating the frequent sequences L^{W_K} in W_k, we must estimate the difference between L^{W_K} and $L^{W_{K+1}}$. If the difference is very large, then we should update the sequential patterns as soon as possible. But if the difference is very small, then we do

not need to update the sequential patterns. In order to measure the difference between the old and new frequent sequence sets, we can define a measure as follows

$$d(L^{W_k}, L^{W_{k+1}}) = \frac{|L^{W_k} \Delta L^{W_{k+1}}|}{|L^{W_k} \bigcup L^{W_{k+1}}|} \quad \text{if} \quad L^{W_k} \neq \Phi \quad \text{or} \quad L^{W_{k+1}} \neq \Phi \quad , \quad \text{otherwise} \quad d(L^{W_k}, L^{W_{k+1}}) = 0 \tag{1}$$

, where $L^{W_k} \Delta L^{W_{k+1}}$ is the symmetric difference between L^{W_k} and $L^{W_{k+1}}$.

We know that a measure is not necessarily a metric. A metric distance must satisfy the triangle inequality which means that for three objects A, B and C in the metric space, the distance between A and C is greater than the distance between A and B plus the distance between B and C. For the measure defined above, we can prove that it is also a metric (please see Appendix A in [13]).

3 The TPD (Tradeoff between Performance and Difference) method of deciding when to update sequential patterns

Lee and Cheung [8] only considered the difference between the old and new association rules, but that they didn't consider the change of the performance of increasingly updating algorithms. As mentioned before, too large difference between the new and old sequential patters will result in poor performance of increasingly updating algorithms, while too small difference will increase the computations lose significantly. Therefore, we must make a tradeoff between the difference of sequential patterns and the performance of increasingly updating algorithms and find the suitable range of the incremental ratio of stream data.

We propose an experimental method, called TPD (Tradeoff between Performance and Difference), to find the suitable range of incremental ratio of the initial window for deciding when to update sequential patterns of stream data. The TPD method uses the speedups as the measurement of increasingly updating algorithms and adopts the measure defined in Section 3 as the difference between the new and old sequential patterns of stream data. With the increment of the size of incremental window, the speedup of the algorithm will decrease, while the difference of the old and new sequential patterns will increase. According to two main factors of the increasingly updating algorithms, the TPD method maps the curve of the speedup and the difference changing with the size of incremental windows into the same graph under the same scale, and the points of intersection of the two curves are the suitable range of the incremental ratio of the initial windows for the increasingly updating algorithm.

In this paper, by the experiments in Section 4, we study the suitable range of incremental ratio of the initial windows for the increasingly updating algorithm: IUS [12] by TPD (Tradeoff between Performance and Difference) method. In the experiments, the speedup ratio of the IUS algorithm is defined as **speedup=the execution time of Robust_search / the execution time of IUS**, where Robust_search is an algorithm to discover sequential patterns from stream data [11] and IUS is an increasingly updating sequential patterns algorithm based on Robust_search algorithm

[12]. We use the distance i.e. $d(L^{W_k}, L^{W_{k+1}})$ defined above as the difference measure between the old frequent sequences L^{W_k} and the new frequent sequences $L^{W_{k+1}}$.

The experiments of Section 4 show that generally, as the size of incremental windows grows, the values of the speedup and the values of the difference will decrease and increase respectively. By making data transform, called Min-max normalization [14] for the values of the speedup and the difference, we can map the speedup and the difference with the increment of the size of incremental windows into the same graph under the same scale, and then from the graph we can discover the intersection point of two lines, obviously, by which we can compute the suitable range of incremental ratio of the initial window to update the sequential patterns.

4 Experiments

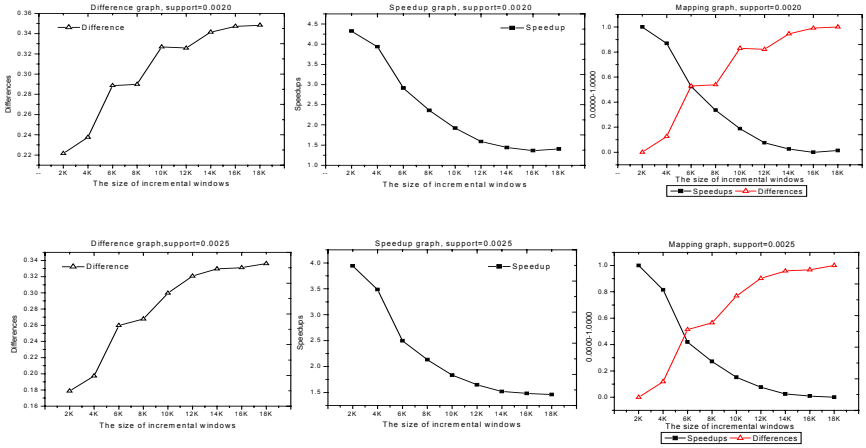

Fig. 2. Experiment 1 on Data_1 |Initial window|=20k

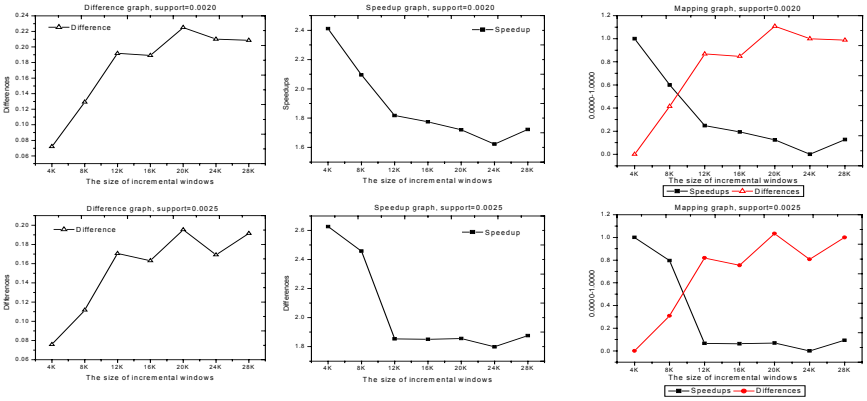

Fig. 3. Experiment 2 on Data_1 |Initial window|=40K

The experiments were on the DELL PC Server with 2 CPU Pentium II, CPU MHz 397.952211, Memory 512M, SCSI Disk 16G. The Operating system on the server is Red Hat Linux version 6.0. The data_1 in experiments are the alarms in GSM Networks, which contain 194 alarm types and 100k alarm events. The time of alarm events in the data_1 range from 2001-08-11-18 to 2001-08-13-17.

The Speedup graph in Figure 2,3 is the speedup of IUS Algorithm [11] to the Robust_search Algorithm [12] with the size of incremental windows. The Difference graph in Figure 2,3 is the difference measure of frequent sequences between the initial window and the incremental windows with the size of that increasing. In the experiments of this paper, we map the value of difference and speedup into the same range [0,1] by let $new\_min_A=0$ and $new\_max_A=1$ [14]. Then, we map the broken lines of the difference and speedup into the same graph i.e. Mapping Graph under the same scale.

In the experiment 1 on data_1, we choose the initial window $|W_0|$=20K, and update the initial sequential patterns by the incremental size of 2K, 4K, 6K, 8K, 10K, 12K, 14K, 16K, and 18K, i.e. the size of incremental window ΔW_i. The results of experiment 1 are illustrated in Figure 2. In order to find the suitable size of incremental windows, we map the graphs of speedup and difference into the same graph by the data transform above, by which we can locate the intersection point of the two lines. The intersection point is a tradeoff between the speedup and the difference, which is a suitable point to update sequential patterns. In the mapping graphs of Figure 2, the intersection point is about 6K, so the suitable range of incremental ratio of initial window is about 30 percent of initial windows W_0.

In the experiment 2 on data_1, we choose the initial window $|W_0|$=40K, and update the initial sequential patterns by the incremental size of 4K, 8K, 16K, 20K, 24K, 28K, and 32K. The results of experiment 2 are illustrated in Figure 3. In the mapping graphs of Figure 3, the intersection point is between 9K and 10K, so the suitable range is about 22.5 to 25 percent of initial windows W_0.

In all, by the experiments above, in general, as the size of incremental windows grows, the values of the speedup and the values of the difference will decrease and increase respectively. Based on the TPD method we proposed, it is shown experimentally that the suitable range of incremental ratio of initial windows to update is about 20 to 30 percent of the size of initial windows for the IUS algorithm.

5 Conclusion

In this paper, we first proposed a metric distance as the difference measure between the sequential patterns. Then we present an experimental method, called TPD (Tradeoff between Performance and Difference), to decide when to update sequential patterns of stream data. The TPD method can determine a reasonable ratio of the size of incremental window to that of original window for increasingly updating algorithms, which may depend on the concrete data. We also do two experiments of IUS algorithm [11] to verify the TPD method. From the experiments, we can see that as the size of original windows increases, the incremental ratio determined by the TPD method does not monotonically increase or decrease but changes in a range

between 20 and 30 percentage. So in practice, by use of the TPD method, we can do some initial experiments to discover a suitable incremental ratio for this kind of data and then use this ratio to decide when to update sequential patterns is better.

Acknowledgement

Thanks Professor Jiawei Han's Summer Course in Beijing 2002. Thanks Professor Wei Li for the choice of subject and guidance of methodology, Professor YueFei Sui and the anonymous Reviewers' suggestions for this paper. This research was supported by National 973 Project of No.G1999032701 and No.G1999032709.

Reference

1. D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, "Maintenance of discovered association rules in large databases: An incremental update technique," *In Proceedings of 12th ICDE'9)*, pages 106-114, Louisiana, USA, February 1996.
2. D. W. Cheung, S. D. Lee, and B. Kao, "A General Incremental Technique for Maintaining Discovered Association Rules," *In Proceedings of the 5th DASFAA'97*, pages 185-194, Melbourne, Australia, April 1997.
3. F. Masseglia, P. Poncelet and M.Teisseire, "Incremental Mining of Sequential Patterns in Large Databases (PS)," *BDA'00*, Blois, France, October 2000.
4. S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas, "Incremental and Interactive Sequence Mining," *In Proceedings of the 8th CIKM'99*, pages 251-258, USA, November 1999.
5. Necip Fazil Ayan, Abdullah Uz Tansel, and Erol Arkun, "An efficient algorithm to update large itemsets with early pruning," *Proceedings of the fifth KDD'99*, August 15-18, 1999, San Diego, CA USA pp.287-291.
6. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, "An efficient algorithm for the incremental updating of association rules in large database," *In Proceedings of the 3rd KDD'97*, pages263-266, California, USA, August 1997.
7. Ahmed Ayad, Nagwa El-Makky and Yousry Taha, "Incremental Mining of Constrained Association Rules," *ICDM'2001 MINING*, April 5-7, 2001, USA
8. S.D. Lee and D.W. Cheung, "Maintenance of Discovered Association Rules: When to update?," *Proc. 1997 Workshop on Data Mining and Knowledge Discovery (DMKD'97)* with ACM-SIGMOD'97,Tucson, Arizona, May 11, 1997.
9. V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh, "A frame work for measuring changes in data characteristics," In Proceedings of the 18th Symposium on Principles of Database Systems, 1999.
10. V. Ganti, J. Gehrke and R. Ramakrishnan, "Mining Data Streams under Block Evolution," SIGKDD Explorations, Volume 3, Issue 2, pp.1-11, January, 2002.
11. Q. Zheng, K. Xu, W. Lv, S. Ma, "Intelligent Search of Correlated Alarms from Database Containing Noise Data," Proc. of the 8th International IFIP/IEEE NOMS 2002, Florence, Italy, April, 2002, http://arXiv.org/abs/cs.NI/0109042
12. Q. Zheng, K. Xu, W. Lv, S. Ma, "The Algorithms of Updating Sequential Patterns" The Second SIAM Data mining'2002: workshop HPDM , Washington, USA, April 2002. http://arXiv.org/abs/cs.DB/0203027
13. Q. Zheng, K. Xu, S. Ma, "When to Update the sequential patterns of stream data", Tecnical Report, NLSDE, China, 2002, http://arXiv.org/abs/cs.DB/0203028
14. J. Han and M. Kambr, "DATA MINING Concepts and Techniques", p.115, Morgan Kaufmann Publisher, 2000