A Survey on sliding window based weighted maximal frequent pattern mining over data streams

Rahul Anil Ghatage, Dnyaneshwar A Rokade, Rahul V Chavan

Abstract— **The Frequent pattern mining is one of the important tasks used in data mining domain and frequent data mining approaches are widely applied onto static database as well as data stream but the data have been accumulated more quickly in recent years and the corresponding databases have also become huger, and thus, general frequent pattern mining methods have been faced with limitations that do not appropriately respond to the massive data, so it is necessary to conduct more efficient and immediate mining tasks by scanning databases only once. In this paper, we analyze the different sliding window approach which can perform mining operations focusing on recently accumulated parts over data streams and mine all of the frequent patterns in the data stream environment and efficiently compressing generated patterns to solve that problem. In addition, we focus on different algorithm which not only use support conditions but also weight constraints expressing items and mining weighted maximal frequent patterns over sliding window model-based data streams, and also we survey the different approaches which always extract mining results regarding the latest data over data streams, and can gain the resulting patterns more quickly through the maximal frequent pattern technique and weight conditions.**

Index Terms— **Data mining, Data stream, Sliding window, weighted maximal frequent pattern mining.**

I. Introduction

The well-known frequent pattern mining (FPM) algorithms are apriori based on breadth first search finds frequent patterns over static databases and to obtain complete results of frequent patterns, the algorithm should scan databases repeatedly and FP-growth on the basis of depth first search conduct mining work with two fixed database scans and does not generate candidate patterns in comparison to Apriori, but it is necessary to apply frequent pattern mining in dynamic data streams. Data streams mean that transaction data are added constantly, and thus, they have continuous and unlimited features.

Data stream mining has to satisfy the requirements in which each data element needed for data stream analysis has to be examined only once and all of the entered data elements have to be processed as soon as possible and results of data stream analysis should be available instantly as well as their quality should also be acceptable whenever users want the results. The old frequent pattern mining methods do not satisfy these requirements since they have to conduct two or more database scans to mine frequent patterns. Therefore, to resolve these problems apply mining approaches with only one scan and use data stream mining methods which extract frequent patterns over data streams effectively.

In data streams, data elements are constantly added and their sizes are continuously increased according to accumulation of transaction data. Therefore, frequent patterns generated over data streams also become large, which means spending a lot of time mining the patterns, and thereby it can violate one of the requirements for the data stream mining, immediate processing. As data have been accumulated in data streams continuously, importance of certain data entered a long time ago can decline or they may be no longer needed, while that of recently added data can be relatively high. To apply these characteristics in the mining process, a variety of window model-based mining approaches have been proposed, and damped window, landmark window, and sliding window techniques can be selectively applied according to characteristics of data streams. Especially since the sliding window-based mining approaches perform mining operations with only the most recent data among accumulated data streams, we can obtain recent high-quality results by using them.

A. *DATA STREAM APPLICATIONS*

Data streams are used in various applications. Some important applications are as follows.

1. Network monitoring in data stream
2. Intrusion detection in data stream
3. Sensor network analysis in data stream
4. Cosmological application in data stream
5. Environmental and weather data in stream

The tele communication companies have massive streams of data containing information about phone calls. It is important to analyze the underlying data in order to determine the broad patterns in the data.

In many applications the intrusions appear as sudden bursts of patterns in even greater streams of attacks. The intrusion makes the problem very difficult, because one cannot scan the data twice. Stream clustering turns outs to be

Manuscript received Oct, 2014.

Rahul Anil Ghatage, *Department of computer engineering, imperial college of engineering & research, Pune, India, 9975251285.*

Dnyaneshwar A. Rokade, *Department of computer engineering, imperial college of engineering & research, Pune, India, 9960488094,*

Rahul V. Chavan, *Department of computer engineering, sinhgad institute of technology, pune , Pune, India, 9503475101.*

quite useful for such problems. When known intrusions are received in the stream, they can be used in order to create class-specific clusters. These class specific clusters can be used to determine the nature of new clusters which arise from unknown intrusion detection.

B. *FREQUENT PATTERN MINING*

A collection of one or more items in a transaction is known as item set. Consider the example T= {beer, bread, chips, diaper} is an item set. An item set whose threshold value is greater than equal to minimum support and confidence is known as frequent item set.

ID	ITEMSET
1	A,B,D
2	A,C,D
3	A,D,E
4	B,E,F
5	B,C,D,E,F

Table 1 Transaction Database

In the above table there are five transactions occurred namely A,B,C,D,E,F. In that A occurred in 3 times. B occurred in 3 times. C occurred in 2 times and D, E, F occurred 3, 4, 2 times respectively. In that example we set 3 as threshold value. So according to the threshold value A, B, D, F are called as frequent items because their occurrence values are greater than the threshold value. C and F are called as infrequent item sets. So they are omitted. Thus this is called as frequent pattern mining. Frequent pattern mining is used in variety of areas. Some of them are:

1. Click stream analysis
2. Drug design
3. Market basket analysis
4. Web link analysis
5. Genome analysis

The weighted maximal frequent pattern mining over data streams based on sliding window model can always extract mining results regarding the latest data over data streams, and can gain the resulting patterns more quickly through the MFP technique and weight conditions. Weighted maximal frequent pattern mining can efficiently mine weighted maximal frequent patterns with only one scan over sliding window-based data stream environment.

II. RELATED WORK

Many previous studies contributed to the efficient mining of frequent itemsets over data streams (Chang & Lee, 2003, 2004; Chi et al., 2006; Giannella et al., 2003; Leeet al., 2005; Li et al., 2004, 2005; Manku & Motwani, 2002; Yu et al., 2006). According to the stream data processing model (Zhu & Shasha, 2002), the research of mining frequent itemsets in data streams can be divided into three categories: landmark-window based mining (Li et al., 2004, 2005; Manku & Motwani, 2002; Yu et al., 2006), damped-window based mining (Chang & Lee, 2003; Giannella et al., 2003), and sliding-window based mining (Chang & Lee, 2004; Chi et al., 2006; Lee et al., 2005).

As well-known fundamental frequent pattern mining algorithms, there are Apriori (Agrawal & Srikant, 1994) based on Breadth First Search and FP-growth (Han, Pei, Yin, & Mao, 2004) on the basis of Depth First Search. On the basis of those basic algorithms, a variety of pattern mining algorithms have been proposed, such as frequent pattern mining without the minimum support threshold specified by users (Chuang, Huang, & Chen, 2008; Li, 2009; Zhang & Zhang, 2011), sequential frequent pattern mining (Chang, Wang, Yang, Luan, & Tang, 2009; Muzammal & Raman, 2011; Yun, Ryu, & Yoon, 2011). Furthermore, frequent pattern mining has been utilized in extensive applications such as medical and bio data analysis (Sallaberry, Pecheur, Bringay, roche, & Teisseire, 2011; Xiong, He, & Zhu, 2010), stock market and protein networks (Sim, Li, Gopalkrishnan, & Liu, 2009), network environment (Fang, Deng, & Ma, 2009; Lin, Hsieh, & Tseng, 2010), traffic data analysis (Liu, Zheng, Chawla, Yuan, & Xing, 2011), analysis of web-click streams (Li, 2008; Li, Lee, & Shan, 2006), and so on. Frequent pattern mining can be applied not only in static databases like the above methods but also in data streams.

As an early frequent pattern mining algorithm, Apriori (Agrawal & Srikant, 1994) finds frequent patterns over static databases. The algorithm performs mining operations in Breadth First Search (BFS) manner and has to generate numerous candidate patterns in the process of actual frequent patterns. Moreover, to obtain complete results of frequent patterns, the algorithm should scan databases repeatedly, and especially in the worst case, the scanning task has to be performed as many as the number of items of the longest transaction in a database. Thereafter, FP-Growth algorithm (Han et al., 2004) based on Depth First Search (DFS) was proposed in order to overcome that problem, and most of the numerous algorithms suggested so far are on the basis of the framework and techniques of FP-growth. The algorithm can more efficiently conduct mining work with two fixed database scans and does not generate candidate patterns in comparison to apriori.

Finding frequent patterns in a continuous stream of transactions is critical for many applications such as retail market data analysis, network monitoring, web usage mining, and stock market prediction. Even though numerous frequent pattern mining algorithms have been developed over the past decade, new solutions for handling stream data are still required due to the continuous, unbounded, and ordered sequence of data elements generated at a rapid rate in a data stream. Therefore, extracting frequent patterns from more recent data can enhance the analysis of stream data so Compact Pattern Stream tree (Tanbeer et al., 2009) was proposed to capture the recent stream data content and efficiently remove the obsolete, old stream data content & use dynamic tree restructuring in compact pattern stream tree to produce a highly compact frequency-descending tree structure at runtime. The complete set of recent frequent patterns is obtained from the tree of the current window using an FP-growth mining technique

Online mining of frequent itemsets over a stream sliding window is one of the most important problems in stream data mining with broad applications. It is also a difficult issue since the streaming data possess some challenging characteristics, such as unknown or unbound size, possibly a very fast arrival rate, inability to backtrack over previously arrived transactions, and a lack of system control over the order in which the data arrive, so MFI-TransSW algorithm (Hua-Fu Li et al., 2007) was proposed by three phases: window initialization, window sliding and pattern generation. First, every item of each transaction is encoded in an effective bit-sequence representation in the window initialization phase. The proposed bit-sequence representation of item is used to reduce the time and memory needed to slide the windows in the following phases. Second, MFI-TransSW uses the left bit-shift technique to slide the windows efficiently in the window sliding phase. Finally, the complete set of frequent itemsets within the current sliding window is generated by a level-wise method in the pattern generation phase.

Mining frequent itemsets over high speed, continuous and infinite data streams is a challenging problem due to changing nature of data and limited memory and processing capacities of computing systems. Sliding window is an interesting model to solve this problem since it does not need the entire history of received transactions and can handle concept change by considering only a limited range of recent transactions. However, previous sliding window algorithms require a large amount of memory and processing time, new algorithm (Mhmood Deypir et al., 2013) based on a prefix tree data structure to find and update frequent itemsets of the window. In order to enhance the performance, instead of a single transaction, a batch of transactions is used as the unit of insertion and deletion within the window. Moreover, by using an effective traversal strategy for the prefix tree and suitable representation for each batch of transactions, both updating of current itemsets and inserting of newly emerged itemsets are performed together, thus improving the performance even further.

The new mining algorithm NTK (Zhi-Hong Deng et al., 2013) mostly used to mine top-rank-k- frequent patterns. The NTK algorithm employs a data structure, Node-list, to represent patterns. The Node-list structure makes the mining process much efficient.

The different data mining techniques use for finding recent frequent itemsets adaptively over an online data stream. The effect of old transactions on the mining result of the data steam is diminished by decaying the old occurrences of each itemset as time goes by. Furthermore, several optimization techniques are devised to minimize processing time as well as main memory usage.

Finally novel algorithm (Gangin Lee et al., 2014), weighted maximal frequent pattern mining over data streams based on sliding window model to obtain weighted maximal frequent patterns reflecting recent information over data streams by conduct more efficient and immediate mining tasks by scanning databases only once.

III. ANALYSIS - FPM OVER DATA STREAMS

A. *Sliding window-based frequent pattern mining over data streams*

The mining methods based on FP-Growth have an effect on static databases and they are not suitable for data streams accumulating data continuously. Since these methods perform more than two database scans, they do not deal with data streams instantly. Moreover, since they construct trees with items remained after infrequent items are deleted, they have to discard previously generated trees and build new trees again if new transaction data are added into data streams.

In data streams, although a certain item is currently infrequent, it can become frequent one according to addition of new transaction data. However, those two scan-based methods must read databases from the first again since they already eliminated infrequent items in the previous step. To solve this, mining methods suitable for data streams (Ahmed, Tanbeer, Jeong, Lee, & Choi, 2012; Chen & Wang, 2010; Tanbeer et al., 2009a) have been proposed, and they can perform mining tasks with only one database scan, thereby responding to changes of data streams immediately. After that, sliding window-based frequent pattern mining approaches (Ahmed et al., 2009; Chen et al., 2012; Deypir et al., 2012; Farzanyar et al., 2012; Li, 2011; Mozafari et al., 2008; Shie et al., 2012; Tanbeer et al., 2009b; Zhang & Zhang, 2011) have been proposed, which can mine frequent patterns considering the latest transaction data of large data streams. Especially in those paper (Tanbeer et al., 2009a, 2009b), an efficient tree-restructuring method, BSM was proposed. The method performs restructuring operations more effectively than previous ones such as the path adjusting method, etc. IWFP algorithm (Ahmed et al., 2012) is a weighted frequent pattern mining algorithm over data streams, applying the BSM method. Among accumulated data streams, the most important elements are recently added data in general. In other words, importance of previously added data can be lowered or meaningless, while that of lately accumulated ones can be relatively higher. Therefore, to reflect these characteristics, the sliding window model can be applied into mining process. The method divides data streams into windows composed of a set of constant-sized transactions and finds frequent patterns from recently generated windows, where the size of windows and the number of them can be assigned as various values by users. Through the sliding window-based approach, we can always obtain frequent patterns reflecting recent information. In Tanbeer et al. (2009b), Tanbeer et al. suggested a frequent pattern mining algorithm over sliding window-based data streams, applying the BSM technique to tree restructuring steps in order to raise efficiency of mining operations.

B. *Maximal frequent pattern mining over data streams*

Mining all frequent patterns over data streams as well as static databases can cause numerous computational overheads in general if data sizes are large. In sliding window-based data stream mining, since the remaining parts except for the latest windows are not considered, the overheads can be reduced, but we cannot still avoid causing them if the size of windows or the number of them becomes large. For this reason, the MFP notation, which can compress generated frequent patterns into a small number of compressed forms, can be utilized in the mining process, and a variety of MFP mining methods (Burdick et al., 2005; Chen et al., 2011; Farzanyar et al., 2012; Gouda & Zaki, 2005; Grahne & Zhu, 2005; Huang et al., 2007; Luo & Chung, 2008, 2012; Priya et al., 2012; Selvan & Nataraj, 2010; Yang et al., 2007; Yun et al., 2012; Zeng et al., 2009) have been proposed. In MAFIA algorithm (Burdick et al., 2005), vertical bitmap representation was proposed so as to help mine MFPs more efficiently.

The algorithm uses an additional data structure with a bitmap form to reduce the number of tree traversals. After the bitmap is constructed, MAFIA can know pattern's frequency through AND operation of the bitmap even though it does not try to traverse trees actually. FPmax* (Grahne & Zhu, 2005) is a state-of-the-art MFP mining algorithm, where FP-array, an additional data structure for mining MFPs more quickly, was proposed, thereby decreasing tree traversal times considerably. Since FP-array has information of patterns' supports, the algorithm can calculate them in advance before trees are actually traversed when growth processes are performed. Consequently, this technique not only can reduce tree traversal operations effectively but also can enhance pruning efficiency by preventing generation of needless conditional trees. However, since the above algorithms have two scan-based processes, they are not suitable for the data stream mining.

C. Applying weight conditions into frequent pattern mining over data streams

Each item existing in data streams has unique importance (or weight). Weights of items in data streams are used in the mining process after they are converted into normalized values within a certain range. The reason is that if a weight of any item is too large, it is hard to denote its weighted support as a finite number of digits. The main challenge of applying weights is to maintain the anti-monotone property. However, the application generally destroys that property since weighted infrequent patterns can become weighted frequent ones as pattern growth operations are conducted. For this reason, researchers have made efforts to maintain the anti-monotone property, and a variety of methods (Ahmed et al., 2009, 2012; Wang & Zeng, 2011; Yun & Ryu, 2011; Yun et al., 2011, 2012) have been proposed. WFPMDS (Ahmed et al., 2009) mines weighted frequent patterns over data stream environment based on the sliding window model. The algorithm conducts tree restructuring work with the BSM technique and provides the most recent mining results from the sliding window whenever users request them. In this study, the framework of the proposed algorithm, WMFP-SW is based on the state-of-the-art MFP mining algorithm, FPmax* and the outstanding tree restructuring technique, BSM.

V. CONCLUSION

In this paper we have studied the concept of data streams and how the frequent patterns are mined over data streams. In addition to this, we have analyzed the different existing research works of frequent pattern mining over data streams. The merits, demerits and future enhancements of the existing works are also discussed. In future, we will develop new techniques and algorithms for finding frequent patterns over data streams which helps to overcome the drawbacks of the existing techniques and also we survey the techniques and algorithms for mining weighted maximal frequent patterns over data streams based on the sliding window concept, which can perform mining operations focusing on recently accumulated parts over data streams and these data stream mining methods can extract frequent patterns over data streams effectively.

REFERENCES

[1] Gangin Lee , Unil Yun , Keun Ho Ryu," Sliding window based weighted maximal frequent pattern mining over streams" Expert Systems with Applications vol. 41, pp. 694–708, 2014.

[2] Chen, Y., Bie, R., & Xu, C. "A new approach for maximal frequent sequential patterns mining over data streams". International Journal of Digital Content Technology & its Application.Vol. 5(6), pp. 104–112, 2011.

[3] Agrawal, R., & Srikant, R. "Fast algorithms for mining association rules. In Proceedings of the 20th international conference on very large databases" pp. 487–499, September 1994.

[4] Ahmed, C. F., Tanbeer, S. K., Jeong, B. S., & Lee, Y. K. "An efficient algorithm for sliding window-based weighted frequent pattern mining over data streams".IEICE Transactions, Vol. 92-D(7),pp. 1369–1381, 2009.

[5] Chen, Y., Bie, R., & Xu, C." A new approach for maximal frequent sequential patterns mining over data streams". International Journal of Digital Content Technology and its Applications, Vol.5 (6), pp.104–112, 2011.

[6] Chen, H., Shu, L., Xia, J., & Deng, Q. "Mining frequent patterns in a varying-size sliding window of online transactional data streams". Information Sciences Vol.215, pp.15–36, 2012.

[7] Deypir, M., Sadreddini, M. H., & Hashemi, S. "Towards a variable size sliding window model for frequent itemset mining over data streams". Computers & industrial engineering, Vol. 63(1), pp.161–172, 2012.

[8] Farzanyar, Z., Kangavari, M. R., & Cercone, N. "Max-FISM: Mining (recently) maximal frequent itemsets over data streams using the sliding window model". Computers & Mathematics with Applications, Vol. 64(6), pp.1706–1718, 2012.

[9] Li, H. "A sliding window method for finding Top-k path traversal patterns over streaming Web click-sequences". Expert Systems with Applications, Vol.36 (3), pp.4382 - 4386, 2008.

[10] Thomas, L.T., Valluri, S.R., & Karlapalem, K. "MARGIN: Maximal frequent subgraph Mining". In Proceedings of the 6th IEEE international conference on data mining, pp. 1097–1101, 2006.

[11] Gouda, K., & Zaki, M. J. GenMax: "An efficient algorithm for mining maximal frequent itemsets. Data Mining and Knowledge Discovery". Vol. 11(3), pp.223–242, 2005.

[12] Luo, C., & Chung, S. M. "A scalable algorithm for mining maximal frequent sequences using a sample". Knowledge and Information Systems, Vol. 15(2), pp. 149–179, 2008.

[13] Mozafari, B., Thakkar, H., & Zaniolo, "Verifying and mining frequent patterns from large windows over data streams". In Proceedings of the 24th international conference on data, engineering pp. 179–188, 2008.

[14] Zhang, X., & Zhang, Y. "Sliding-window Top-k pattern mining on uncertain streams". Journal of Computational Information Systems, Vol. 7(3), pp. 984–992, 2011.

[15] Priya, R. V., Vadivel, A., & Thakur, R. S. "Maximal pattern mining using fast CP-tree for knowledge discovery". International Journal of Information Systems and Social Change, Vol. 3(1), pp. 56–74, 2012.

[16] Shie, B., Yu, P. S., & Tseng, V. S. "Efficient algorithms for mining maximal high utility itemsets from data streams with different models". Expert Systems with Applications, Vol.39 (17), pp. 12947–12960, 2012.

[17] Tanbeer, S. K., Ahmed, C. F., Jeong, B. S., & Lee, Y. K. "Efficient single-pass frequent pattern mining using a prefix-tree". Information Sciences, Vol. 179(5), pp.559–583, 2009.

[18] Tanbeer, S. K., Ahmed, C. F., Jeong, B. S., & Lee, Y. K. "Sliding window-based frequent pattern mining over data streams". Information Sciences, Vol. 179(22), pp. 3843–3865, 2009.

[19] Li, H. "A sliding window method for finding Top-k path traversal patterns over streaming Web click-sequences". Expert Systems with Applications, Vol. 36(3), pp. 4382–4386, 2008.

[20] Yang, C., Li, Y., Zhang, C., & Hu, Y. "A novel algorithm of mining maximal frequent pattern based on projection sum tree". Fuzzy Systems and Knowledge Discovery, Vol. 1, pp. 458–462, 2007.

[21] Zhang, X., & Zhang, Y. "Sliding-window Top-k pattern mining on uncertain streams". Journal of Computational Information Systems, 7(3), pp. 984–992, 2011.

[22] Mhmood Deypir, Mohammad Hadi Sadreddini, "An efficient sliding window based algorithm for adaptive frequent itemset mining over data streams", Journal of Information Science and Engineering, Vol. 29, pp.1001-1020, 2013

[23] Zhi-Hong Deng, "Fast mining Top-Rank-k frequent patterns by using Node-lists", expert systems with applications Vol. 4, pp. 1763–1768, 2014.

[24] Yun, U., Shin, H., Ryu, K., & Yoon, E. "An efficient mining algorithm for maximal weighted frequent patterns in transactional databases". Knowledge-Based Systems, vol. 33, pp.53–64, 2012.

[25] Wang, J., & Zeng, Y. "DSWFP: Efficient mining of weighted frequent pattern over data streams. Fuzzy Systems and Knowledge Discovery, Vol. 2, 942–946, 2011.