Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 12 Issue 11 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Data Stream Mining: A Review on Windowing Approach

By Pramod S. & O.P.Vyas
Ravishankar Shukla University, Raipur

Abstract - In the data stream model the data arrive at high speed so that the algorithms used for mining the data streams must process them in very strict constraints of space and time. This raises new issues that need to be considered when developing association rule mining algorithms for data streams. So it is important to study the existing stream mining algorithms to open up the challenges and the research scope for the new researchers. In this paper we are discussing different type windowing techniques and the important algorithms available in this mining process.

Keywords: Data Stream Mining, Association Rule Mining, Data Mining, Online Data Mining.

GJCST-C Classification: D.2.2

DATA STREAM MINING A REVIEW ON WINDOWING APPROACH

Strictly as per the compliance and regulations of:

Data Stream Mining: A Review on Windowing Approach

Pramod S. [α] & O.P.Vyas [σ]

Abstract - In the data stream model the data arrive at high speed so that the algorithms used for mining the data streams must process them in very strict constraints of space and time. This raises new issues that need to be considered when developing association rule mining algorithms for data streams. So it is important to study the existing stream mining algorithms to open up the challenges and the research scope for the new researchers. In this paper we are discussing different type windowing techniques and the important algorithms available in this mining process.

Keywords : Data Stream Mining, Association Rule Mining, Data Mining, Online Data Mining.

I. Introduction

Once any company decided to use the data mining system for daily operations, management will be concerned with the system performance for their environment. If the mining take place on the historic data then the result could be used for the future strategic decision making[1]. But the amount of data collected over time is increased in daily basis in the database then that can reduce the accuracy of the result[2] of the mining process. This is where online data mining can play a vital role to improve the mining result and its accuracy[3].

a) Frequent Itemset Mining Approaches

The tentative nature of frequent itemset mining normally results in a large number of frequent itemset generations. The increase in the number of frequent itemset generated will result in the degradation of mining efficiency. The frequent closed itemset[9] mining is the solution for the above said problem. The FCI is a non redundant representation of the set of frequent itemsets[10]. The commendable reduction in the size of the result set leads to improved performance in the speed and memory usage. Different efficient FCI algorithms[8,11,12] are proposed by different authors. We noticed that the FCIs approach could not be applied over land mark window since the number of FCIs approaches that of frequent itemsets when the window becomes very large.

There is one another frequent item set mining technique called Frequent Maximal Item set[7]. Compare with the Frequent Closed Item set mining technique it will generate comparatively less number of item sets, due to this reason it is significantly more efficient [13] in terms of both CPU and memory. But the disadvantage of FMI mining is that it lose the frequency information of the subset of FMIs so the error bound will also increased. There are many concise representations of frequent item sets are proposed [14, 15, 16, 17, 18, 19], these are significantly saving memory space, CPU and shows better accuracy. This technique could be applied in stream mining with the efficient incremental technique and batch processing.

II. Windowing Approach to Data Stream Mining

One of the main issues in the stream data mining is to find out a model which will suit the extraction process of the frequent item set from the streaming in data. There are three stream data processing model[20] that are Landmark window, Damped window and Sliding window model. A transaction data stream is a sequence of incoming transactions and an excerpt of the stream is called a window. A window, W, can be either time-based or count-based, and either a landmark window or a sliding window. W is time-based if W consists of a sequence of fixed-length time units, where a variable number of transactions may arrive within each time unit. W is count-based if W is composed of a sequence of batches, where each batch consists of an equal number of transactions. W is a landmark window if $W = (T_1, T_2, . . . , T)$; W is a sliding window if $W = (T_{T-w+1}, . . . , T_T)$, where each T_i is a time unit or a batch, T_1 and T_T are the oldest and the current time unit or batch, and w is the number of time units or batches in the sliding window, depending on whether W is time-based or count-based. Note that a count-based window can also be captured by a time-based window by assuming that a uniform number of transactions arrive within each time unit.

The frequency of an item set, X, in W, denoted as freq(X), is the number of transactions in W support X. The support of X in W, denoted as sup(X), is defined as freq(X)/N, where N is the total number of transactions received in W. X is a Frequent Item set (FI) in W, if $sup(X) \geq \sigma$, where σ ($0 \leq \sigma \leq 1$) is a user-specified

Author α : Computer Science and Information Technology Department at Ravishankar Shukla University, Raipur, C.G. and working as Associate Professor in Information Technology Department in Christian College of Engineering and Technology, Bhilai, C.G, India.
E-mail : pramodsnair@yahoo.com
Author σ : Professor in Indian Institute of Information Tecnology-Allahabad, U.P., India. E-mail : dropvyas@gmail.com

minimum support threshold. X is a Frequent Maximal Item set (FMI) in W, if X is an FI in W and there exists no item set Y in W such that X ⊂ Y . X is a Frequent Closed Item set (FCI) in W, if X is an FI in W and there exists no item set Y in W such that X ⊂ Y and freq(X) = freq(Y).

a) Landmark Window Concept

In this section we will discuss some of important land mark window algorithms. One of the algorithm proposed by Manku and Motwani[23] is a lossy counting approximation algorithm. It will compute the approximate set of frequent item sets over the entire stream so far. In this algorithm the stream is divided into sequence of buckets. The lossy counting algorithm processes a batch of transactions arriving at a particular time. In this paper they are maintaining the item set, the frequency of item set and the error as the upper bound of the frequency of the item set. This algorithm uses three different modules, Buffer, Trie and Set Gen. The Buffer module keeps filling the available memory with the incoming transactions. This module frequently computes the frequency of every item in the current transactions and prune if it is less than N. The Trie module maintains set D, as a forest of prefix trees. The Trie forest as an array of tuples (X, freq(X), err (X), level) that correspond to the pre-order traversal of the forest, where the level of a node is the distance of the node from the root. The Trie array is maintained as a set of chunks. On updating the Trie array, a new Trie array is created and chunks from the old Trie are freed as soon as they are not required.

All the item sets in the current batch having the support will be generated by the Set Gen module. The Apriori-like pruning[21] will help to avoid the generation of superset of an item set if the frequency less than β in the current batch. The Set Gen implemented with the help of Heap queue. Set Gen repeatedly processes the smallest item in Heap to generate a 1-itemset. If this 1-itemset is in Trie after the Add Entry or the Update Entry operation is utilized, Set Gen is recursively invoked with a new Heap created out of the items that follow the smallest items in the same transactions. During each call of Set Gen, qualified old item sets are copied to the new Trie array according to their orders in the old Trie array, while at the same time new item sets are added to the new Trie array in lexicographic order. When the recursive call returns, the smallest entry in Heap is removed and the recursive process continues with the next smallest item in Heap.

The quality of the approximation mining results by using the relaxed minimum support threshold ∈ leads to the extra usage of memory and the processing power. That is, the smaller relaxed minimum support leads to increase of number of sub-FIs generated, so the increase of memory and the extra usage of processing power. , if ∈ approaches σ, more false-positive answers will be included in the result, since all sub-FIs whose computed frequency is at least $(\sigma - \in)N \approx 0$ are displayed while the computed frequency of the sub-FIs can be less than their actual frequency by as much as σN. The same problem is in other mining algorithms [21, 22, 23, 24, 13, 4] that use a relaxed minimum support threshold to control the accuracy of the mining result.

One of the algorithm called DSM-FI developed by Li[13], is to mine an approximate set of FIs over the entire history of the stream. This algorithm is used a prefix-tree based in memory data structure. DSM-FI is also using the relaxed minimum support threshold and all the generated FIs are stored in the IsFI-forest. The DSM-FI consists of Header Table(HT) and Sub-Frequent Itemsets tree(SFI-tree). For every unique item in the set of sub-FIs it inserts an entry with frequency, batch id and head link, it increments otherwise. The DSM-FI frequently prunes the items that are not satisfied the minimum support.

One of the approximation algorithm developed by Lee[4] used the compressed prefix tree structure called CP-tree. The structure of the CP-tree is described as follows. Let D be the prefix tree used in estDec. Given a merging gap threshold δ, where 0≤ δ ≤ 1, if all the itemsets stored in a subtree S of D satisfy the following equation, then S is compressed into a node in the CP-tree.

$$\frac{freq_T(X) - freq_T(Y)}{N_T} \leq \delta$$

Where X is the root of S and Y is an item set in S. Assume S is compressed into a node v in the CP-tree. The node v consists of the following four fields: item-list, parent-list, freqTmax and freqTmin where v.item-list is a list of items which are the labels of the nodes in S, v. parent-list is a list of locations (in the CP-tree) of the parents of each node in S, v. freqTmax is the frequency of the root of S and freqTmin is the frequency of the right-most leaf of S.

The use of the CP-tree results in the reduction of memory consumption, which is important in mining data streams. The CP-tree can also be used to mine the FIs, however, the error rate of the computed frequency of the FIs, which is estimated from freqTmin and freqTmax, will be further increased. Thus, the CP-tree is more suitable for mining FMIs.

b) Sliding Window Concept

The sliding window model processes only the items in the window and maintains only the frequent item sets. The size of the sliding window can be decided according to the applications and the system resources. The recently generated transactions in the window will influence the mining result of the sliding windowing, otherwise all the items in the window to be maintained. The size of the sliding window may vary depends up on the applications it may use. In this section we will discuss some of the important windowing approaches for stream mining.

An in memory prefix tree based algorithm proposed by Chi[26, 22] following the windowing approach to incrementally update the set of frequent closed item sets over the sliding window . The data structure used for the algorithm is called as Closed Enumeration Tree (CET) to maintain the dynamically selected set of item set over the sliding window. This algorithm will compute the exact set of frequent closed item sets over the sliding window. The updation will be for each incoming transaction but not enough to handle the handle the high speed streams.

One another notable algorithm in the windowing concept is estWin[3]. This algorithm maintains the frequent item sets over a sliding window. The data structure used to maintain the item sets is prefix tree. The prefix tree holds three parameters for each items set in the tree, that are frequency of x in current window before x is inserting in the tree, that is freq(x). The second is an upper bound for the frequency of x in the current window before x is inserted in the tree, err(x). The third is the ID of the transaction being processed, tid(x).b. The item set in the tree will be pruned along with all supersets of the item set, we prune the item set X and the supersets if $\mathrm{tid}(X) \leq \mathrm{tid}_1$ and $\mathrm{freq}(X) < \lceil \in N \rceil$, or (2) $\mathrm{tid}(X) > \mathrm{tid}_1$ and $\mathrm{freq}(X) < \lceil \in (N - (\mathrm{tid}(X) - \mathrm{tid}_1)) \rceil$. The expression $\mathrm{tid}(X) > \mathrm{tid}_1$ means that X is inserted into D at some transaction that arrived within the current sliding window and hence the expression $(N-(\mathrm{tid}(X)-\mathrm{tid}_1))$returns the number of transactions that arrived within the current window since the arrival of the transaction having the ID tid(X). We note that X itself is not pruned if it is a 1-itemset, since estWin estimates the maximum frequency error of an itemset based on the computed frequency of its subsets [84] and thus the frequency of a 1-itemset cannot be estimated again if it is deleted.

c) *Damped Window Concept*

In this section we will discuss some of the notable Damped window algorithms. The estDec[5] algorithm proposed to reduce the effect of the old transactions on the stream mining result. They have used a decay rate to reduce the effect of the old transactions and the resulted frequent item sets are called recent frequent Item sets. The algorithm, for maintaining recent FIs is an approximate algorithm that adopts the mechanism to estimate the frequency of the item sets.

The use of a decay rate diminishes the effect of the old and obsolete information of a data stream on the mining result. However, estimating the frequency of an item set from the frequency of its subsets can produce a large error and the error may propagate all the way from the 2-subsets to the n-supersets, while the upper bound is too loose. Thus, it is difficult to formulate an error bound on the computed frequency of the resulting item sets and a large number of false-positive results will be returned, since the computed frequency of an item set may be much larger than its actual frequency. Moreover, the update for each incoming transaction (instead of a batch) may not be able to handle high-speed streams.

Another approximation algorithm[6] uses a tilted time window model . In this frequency FIs are kept in different time granularities such as last one hour, last two hours, last four hours and so on. The data structure used in this algorithm is called FP-Stream. There are two components in the FP-Stream which are pattern tree based prefix tree and tilted time window which is at the end node of the path. The pattern tree can be constructed using the FP-tree algorithm[25]. The tilted time window guarantees that the granularity error is at most T/2, where T is the time units.

The updation of the frequency record will be done by shifting the recent records to merge with the older records. To reduce the number of frequency records in the tilted-time windows, the old frequency records of an item set, X, are pruned as follows. Let freqj(X) be the computed frequency of X over a time unit Tj and Nj be the number of transactions received within Tj , where $1 \leq j \leq \tau$. For some m, where $1 \leq m \leq \tau$, the frequency records freq1(X), . . . ,freqm(X) are pruned if the following condition holds:

$$\exists n \leq \tau, \forall i, 1 \leq i \leq n, \mathrm{freqi}(X) < \sigma Ni \text{ and}$$

$$\forall l, 1 \leq l \leq m \leq n, \sum_{j-1}^{l} \mathbf{freq}_j(x) < \in \sum_{j=1}^{j} N_J$$

The FP-stream mining algorithm computes a set of sub-FIs at the relaxed minimum support threshold, ∈ , over each batch of incoming transactions by using the FI mining algorithm, FP-growth [25]. For each sub-FI X obtained, FP-streaming inserts X into the FP-stream if X is not in the FP-stream. If X is already in the FP-stream, then the computed frequency of X over the current batch is added to its tilted-time window. Next, pruning is performed on the tilted-time window of X and if the window becomes empty, FP-growth stops mining supersets of X by the Apriori property [2]. After all sub-FIs mined by FP-growth are updated in the FP-stream, the FP-streaming scans the FP-stream and, for each item set X visited, if X is not updated by the current batch of transactions, the most recent frequency in X's tilted-time window is recorded as 0. Pruning is then performed on X. If the tilted-time window of some item set visited is empty (as a result of pruning), the item set is also pruned from the FP-stream.

The tilted-time window model allows us to answer more expressive time-sensitive queries, at the expense of some frequency record kept for each item set. The tilted-time window also places greater importance on recent data than on old data as does the sliding window model; however, it does not lose the information in the historical data completely. A drawback of the approach is that the FP-stream can become very large over time and updating and scanning such a large structure may degrade the mining throughput.

III. Conclusion

In this paper we have discussed some of the issues of the windowing concept for the online stream mining to develop an effective, performance oriented algorithm. We also discussed some of the important windowing algorithms in the different windowing concept and reviewed, for some extend, how the existing important algorithms could handle these different issues. The further study can be done on this field to develop an effective algorithm in the data stream mining. We have discussed the way the different algorithms handle the data stream mining so that the researchers can analyze and study further for the research work.

References Références Referencias

1. Fernando Crespoa, Richard Weberb. “A methodology for dynamic data mining based on fuzzy clustering”, Fuzzy Sets and Systems 150 (2005) 267–284.
2. David Hand, Heikki Mannila, Padhraic Smyth. “Principles of Data Mining”, ISBN: 026208290 MIT Press, Cambridge, MA, 2001.
3. Maria Halkidi, “Quality assessment and Uncertainty Handling in Data Mining Process” http://www.edbt2000. unikonstanz.de/phd-workshop/papers/Halkidi.pdf.
4. B. Liu, W. Hsu, and Y. Ma. Integrating Classification and Association Rule Mining. In Proc. of KDD, 1998.
5. J. H. Chang and W. S. Lee. Finding Recent Frequent Itemsets Adaptively over Online Data Streams. In Proc. of KDD, 2003.
6. C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In Kargupta et al.: Data Mining: Next Generation Challenges and Future Directions, MIT/AAAI Press, 2004.
7. D. Lee and W. Lee. Finding Maximal Frequent Itemsets over Online Data Streams Adaptively. In Proc. of ICDM, 2005.
8. Y. Chi, H. Wang, P. S. Yu and R. R. Muntz. Catch the Moment: Maintaining Closed Frequent Itemsets over a Data Stream Sliding Window. In KAIS, 10(3): 265-294, 2006.
9. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering Frequent Closed Itemsets for Association Rules. In Proc. of ICDT, 1999.
10. M. Zaki. Generating Non-Redundant Association Rules. In Proc. of KDD, 2000.
11. M. Zaki and C. J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining. In Proc. of SDM, 2002.
12. J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. In Proc. of KDD, 2003.
13. K. Gouda and M. Zaki. Efficiently Mining Maximal Frequent Itemsets. In Proc. of ICDM, 2001.
14. T. Calders and B. Goethals. Mining All Non-derivable Frequent Itemsets. In Proc. Of PKDD, 2002.
15. J. F. Boulicaut, A. Bykowski and C. Rigotti. Free-Sets: a Condensed Representation of Boolean Data for the Approximation of Frequency Queries. In DMKD, 7(1):5-22, 2003.
16. J. Pei, G. Dong, W. Zou, and J. Han. Mining Condensed Frequent-Pattern Bases. In KAIS, 6(5): 570-594, 2004.
17. D. Xin, J. Han, X. Yan, and H. Cheng. Mining Compressed Frequent-Pattern Sets. In Proc. of VLDB, 2005.
18. F. Bonchi and C. Lucchese. On Condensed Representations of Constrained Frequent Patterns. In KAIS, 9(2): 180-201, 2005.
19. J. Cheng, Y. Ke, and W. Ng. δ-Tolerance Closed Frequent Itemsets. To appear in Proc. of ICDM, 2006.
20. R. Jin and G. Agrawal. An Algorithm for In-Core Frequent Itemset Mining on Streaming Data. In Proc. of ICDM, 2005.
21. LTC Bruce D. Caulkins, J.Lee, M.Wang, “A Dynamic Data Mining Technique for Intrusion Detection Systems, 43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA. Copyright 2005 ACM 1-59593-059-0/05/0003.
22. Y. Chi, H. Wang, P. S. Yu and R. R. Muntz. Catch the Moment: Maintaining Closed Frequent Itemsets over a Data Stream Sliding Window. In KAIS, 10(3): 265-294, 2006.
23. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of Frequent Episodes in Event Sequences. In DMKD, 1:259-289, 1997.
24. Graham Cormode, S.Muthukrishnan; What's Hot and What's Not: Tracking Most Frequent Items Dynamically; ACM Transactions on Database Systems; March 2005.
25. Cheqing Jin, Weining Qian, Chaofeng Sha, Jeffrey X. Yu, Aoying Zhou; Dynamically Maintaining Frequent Items over a Data Stream; Int'l Conf. on Information and Knowledge Management; 2003.
26. Hua-Fu Li, Suh-Yin Lee, and Man-Kwan Shan; An Efficient Algorithm for Mining Frequent Itemsets over the Entire History of Data Streams; Int'l Workshop on Knowledge Discovery in Data Streams; Sept. 2004.