Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review on Algorithms for Mining Frequent Itemset Over Data Stream

Vikas Kumar
Department of Computer Science and engineering
Ajay kumar Garg engineering college, Ghaziabad
India.

Sangita Satapathy
Department of Computer science and engineering
Ajay kumar Garg engineering college, Ghaziabad
India.

Abstract: Frequent itemset mining over dynamic data is an important problem in the context of data mining. The two main factors of data stream mining algorithm are memory usage and runtime, since they are limited resources. Mining frequent pattern in data streams, like traditional database and many other types of databases, has been studied popularly in data mining research. Many applications like stock market prediction, sensor network, retail market data analysis, have a critical use of frequent itemset mining over continuous data streams. This paper is devoted to provide overview of various algorithms developed for extraction of frequent itemset from transactional databases using sliding window model.

Keywords: data stream, data stream mining, frequent itemset, closed frequent itemset, sliding window model.

I. Introduction

With the emergence of new application, include network traffic analysis, web click stream mining, network intrusion detection; the data we process is continuous data stream not static one. Today's information society has become a restless generator of data of various kinds. Huge amount of data is being generated, from web click stream, credit card records, telephone calls records, traditional retail market store transaction. These records are an extremely valuable source of information. The will and need to benefit from this largely available data, was the fertile land where the discipline of data mining was grown. Being continuous, unbounded flow of data which can be read only once, there is several limitation of data stream mining. Firstly, each item can be examined only once. Secondly, limited memory, though the data generated continuously. Finally, the mining results must be as fast as possible. To overcome these limitation, a data stream mining algorithm must be a single pass algorithm, must extract the useful information from the current data stream, which can be used to derive various information required once the data stream has been expired. Transaction arriving in series, forms data stream. Itemset is a set of items. A k-itemset is a set of k items. The ***support*** of itemset X (sup(X)) is a percentage of transactions containing the itemset. ***Frequent itemset*** is an itemset X whose sup(x) ≥ s, where "s" is the minimum support given by user. Any subset of frequent itemset is also frequent (*priori property)*. Due to which frequent itemset mining algorithms suffer from the problem of combinatorial explosion. To alleviate this problem, maximal frequent itemset and closed frequent itemset were discovered. A frequent itemset is ***maximal frequent itemset*** if *none* of its proper supersets is frequent. ***Closed frequent itemset*** is a frequent itemset, having none of its supersets with *same* support as it has. Various Time models, like L*andmark window model, Time-fading model, Sliding window model*, are used to mine continuously generated data streams. *Landmark window model* performs mining over all the data from landmark point (usually the time the system start) to current time of mining. *Time fading model, does* not treat all the data equally as, landmark model, and distinguish new data from old data by assigning different weights. It also considers data from start of stream up to current time.

Both landmark model and time fading model does not provide time-sensitivity while performing mining and support count of entire data, from start to end is to be counted. These limitations are being overcome in *sliding window model,* which fulfil all the requirements for data stream mining; **Time-sensitivity, Approximation, Adjustability.** Sliding window models utilizes only latest W transaction received, W is the size of window, defined by the user.

II. Comparision Study

Number of algorithms has been proposed using different time models, for the extraction of frequent itemset from data stream. One challenge all these algorithm faces is to develop a data structure that can capture full stream content, in a memory-efficient manner, with a single pass. In this paper we are discussing algorithms proposed using the sliding window time model for mining data streams.

A. Moment Algorithm:

Chi et al.(2006) introduced one algorithm, ***Moment***, to extract *closed frequent itemset* within sliding window. They designed, a prefix based data structure, **CET** (Closed Enumeration Tree), to maintain closed frequent itemset. CET maintains the boundary between *closed frequent itemset* and rest of itemset, which makes the boundary relatively stable, whenever any itemset changes its state (frequent to non-frequent vice-versa), ultimately reducing the updating cost. An

efficient algorithm to incrementally update the CET, which update the CET when newly arrived transaction change the content of window or oldest transaction being deleted from the window.

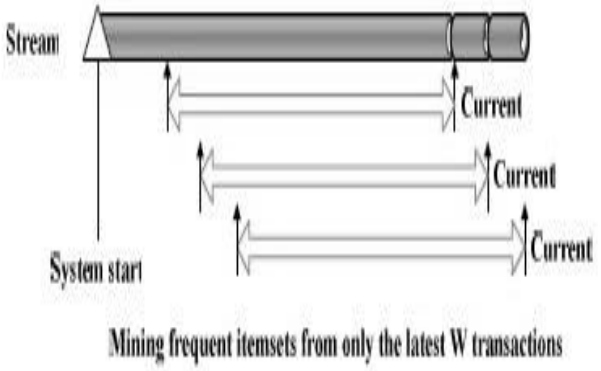

Figure 1: Sliding window model

When a transaction arrives/expires, Moment traverse the part of CET related to that transaction. For each node visited, Moment, update the CET by incrementing/decrementing its frequency.

The merit of Moment is that it computes the exact set of *closed frequent itemset* over a sliding window. Although an update to a node may result in a propagation of the node insertion and deletion in the CET, most of the nodes related to an incoming or expiring transaction do not change their type often. Therefore, the average update cost in the CET is small. Limitation of Moment algorithm is, it stores transaction using data structure, FP tree (Han et al.2004), which require a considerable amount of memory. Secondly, if the size of window is too large, CET can huge.

B. WSW algorithm:

Pauray S.M. Tsai(2009), proposed a new framework data stream mining, called weighted sliding window model, which allows user to specify the number of windows, weight of window and size of window. A single pass algorithm, called WSW, has been proposed to extract frequent itemset from data streams.

The motivation for weighted sliding window model come from the fact, that the size of traditional sliding window model is fixed or defined by the number of transaction, say W. Though recent W transactions are considered, the time to cover these W transactions may be long or varies, which may effectively decrease the mining result. While the weighted sliding window model proposed in this paper defines window size by time not by the number of transaction and user can specify the number of windows, with each window assign with different weight (sum of all window weight equals to 1). For example data may be more influential at current moment and hence, should be assigned higher weight.

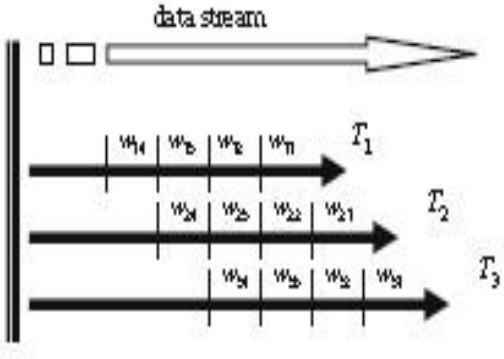

Figure 2: Weighted sliding window model

The algorithm WSW scans the data once in each window, and calculates the support count for each item present in current window, to find out frequent k-itemset (k=1). Based on this information candidate (k+1)-itemset are pointed out to find frequent (k+1)-itemset. The process terminates when no further candidates itemset can be generated.

If the number of windows increases the time to determine frequent itemset increases. To reduce this runtime they proposed an improvement of WSW algorithm called, WSW-imp, which further reduce the time of deciding whether a candidate itemset is frequent or non-frequent itemset.

C. TMoment Algorithm:

Fatemeh Nori et al. (2012), proposed another efficient algorithm, called **TMoment**, for *closed frequent itemset* mining using sliding window model. This algorithm uses single novel prefix tree based data structure, **TCET** (Transaction Translate Closed Enumeration Tree), for storing both transaction of the window and closed frequent itemset. This reduces the considerable amount of main memory usage.

The proposed algorithm, Tmoment, consist of four steps: Computing support using transaction, Building TCET, Eliminating the oldest transaction, adding the new transaction. Tmoment overcomes the limitation of Moment, high memory usage, with the help of TCET. Tmoment does not require a separate data structure (FP-Tree), as used in moment for storing transactions of window. TCET stores both transaction and corresponding frequent itemset, which considerably reduces the memory usage. On the other hand TCET affects the runtime in negative manner. List-intersection has been used to find out support of itemsets, which makes the run-time of Tmoment no better than Moment.

D. VSW Algorithm:

Mahmood Depyir et al. (2012) proposed a new algorithm, VSW (Variable Size sliding Window frequent itemset mining) which is suitable for observing recent changes in set of frequent itemset. In Transactional sliding window the window size is to be kept constant, which is being obtained from user. In order to determine the precise size of window, the user must have advance knowledge about time and scale of changes within data stream, which cannot be easily determined due to unpredictable changing nature of data streams. To overcome this limitation of fixed window size, VSW, algorithm has been developed. In this window size is determined dynamically based on amount concept change that occurs within the arriving data streams. Initially window size is given by user and then adjusted based on the change of frequent pattern embedded in the incoming data stream. The window size expands as the concept becomes stable and shrinks when a concept change occurs.

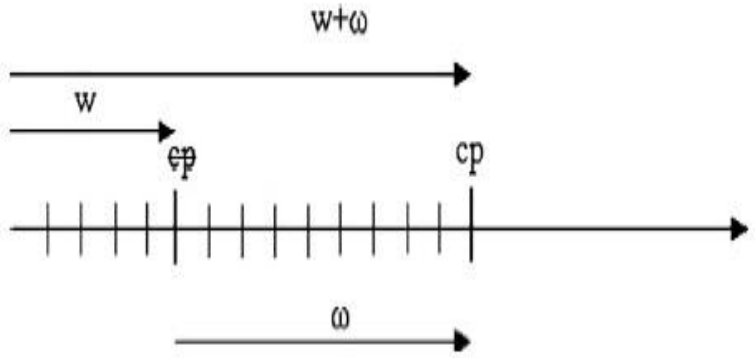

Figure 3: Window size reduction and checkpoint (cp) movement after change detection.

Fig. 3 shows the process of cutting old transaction and forming the new window. Concept change is calculated with respect to checkpoint (tid of last transaction received).If a concept change is detected, all information before the checkpoint is removed from the window and new window formed (window shrinks). If not detected, window size continues to grow with further no action.

III. Conlusion

In this study, we have overviewed various approaches for mining frequent itemset within sliding window over data streams. The algorithms reviewed in this paper shows that mining of data streams is growing into a new branch of knowledge discovery, with its own unique research problems. The need for online processing with time and memory constraints forces researchers to focus on resource usage while designing accurate classifiers. Additionally, concept drift introduce the requirement for a forgetting mechanism that dynamically removes outdate data.

Refernces

[1] F.Nori, Mahmood Deypir. "*A sliding window based algorithm for frequent closed itemset mining over data streams*". J. Syst. Software (2012), Elsevier

[2] Gannella,c.,Han,J.,pei,j.,Yan,X.,& Yu,P.s." *Mining frequent pattern in data stream at multiple time granularities*." Next generation data mining, pp.191-210, (2003)

[3] Hua-Fu Li, Chin-Chuan Ho. "*incremental updates of closed frequent itemsets over continuous data streams*." Expert Systems with Applications pp.2451-2458, Elsevier 2009.

[4] Jiawei Han, Hong Cheng ·& Dong Xin ·Xifeng Yan. "*Frequent pattern mining: current status and future directions*".Springer (2007).

[5] Mahmood Deypir, M Sadreddini, & S Hashemi. "*Towards a variable size sliding window model for frequent itemset mining over data streams*".Elsevier(2012).

[6] Manku,g., & motwani, R." *Approximate frequency count over data streams*" In proceeding of the VLDB conference, pp.346-357 , 2002.

[7] Pauray S.M Tsai." *Mining frequent itemsets in data streams using the weighted sliding window mode*l", Expert Systems with Applications, Vol. 36,pp. 11617–11625, Elsevier 2009.

[8] Yun Chi, Haixm Wang. "*Catch the moment: maintaining closed frequent itemset over a data stream sliding window*." Springer-Verlag 2006

