FREQUENT ITEMSET MINING OVER STREAM DATA: 
OVERVIEW

Z. G. Qu1,2,3,4, X.X. Niu4, J. Deng3, C. McArdle3, X.J. Wang3

1Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, China, 210044, qzghhh@126.com
2School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, China, 210044
3School of Electronic Engineering, Dublin City University, Dublin, Ireland
4Key Laboratory of Network and Information Attack & Defence Technology of MOE, Beijing University of Posts and Telecommunications, Beijing, China, 100876

Keywords: Stream data; Stream data mining; Frequent itemset mining.

Abstract

During the past decade, stream data mining has been attracting widespread attentions of the experts and the researchers all over the world and a large number of interesting research results have been achieved. Among them, frequent itemset mining is one of main research branches of stream data mining with a fundamental and significant position. In order to further advance and develop the research of frequent itemset mining, this paper summarizes its main challenges and corresponding algorithm features. Based on them, current related results are divided into two categories: data-based algorithms and task-based algorithms. According to its taxonomy, the related methods belonging to the different categories and sub-categories are comprehensively introduced for better understanding. Finally, a brief conclusion is given.

1 Introduction

In the past decades, with the rapid development of computer network, wireless communication network and wireless sensor network, a large amount of stream data arises with a variety of forms in a widespread applications, such as environment and astronomical detection, network monitoring, network traffic management, etc. In order to effectively transform the redundant stream data to some useful knowledge, which eventually can be applied to various fields of social life, many researchers and experts have been making their great efforts to advance the research of stream data mining. Currently, many achievements have been proposed in various application fields.

As a classical traditional data mining, sequential data mining based on static databases has gotten great achievements, and it has been playing an important role in economic development and daily life of the people. The typical results include [1,2]. Different with the data of traditional static database, stream data is an infinite and continuous data flow with many unique characteristics, which makes impossible to directly adopt the algorithms of sequential data mining to meet the challenges of mining frequent patterns over stream data. Simply speaking, the reasons caused by the characteristics of stream data are mainly embodied in the three points: one scan to dataset, high computation complexity and huge memory space.

The characteristics of stream data are described as follows.

1) Fast arrival rate: It means that stream data shall be processed as simply and effectively as possible;
2) Unbounded volume: Theoretically infinite arrival stream data means that it is very necessary to deliberate how to manage limited memory space to process the stream data with unbounded volume;
3) Wide territory: Stream data usually has a huge number of attributes with wide value range, which makes it is impossible to store all of the information about the data;
4) Time-varying: When stream data continuously arrives, its statistical characteristics will keep changing over time, such as variance, quantile, probability distribution, etc, which leads to an issue of concept-drifting [3].

Based on them, the related research results of stream data mining show us there are three main challenges needed to be cautiously resolved.

1) For fast arrive rate, a standard algorithm of stream data mining shall implement real-time or as fast as possible processing the data to timely response corresponding results to users. From this perspective, the relevant algorithms must satisfy the requirement of one-scan to dataset;
2) For unbounded volume and wide territory of stream data, there are a significant need of efficiently managing the limited internal/external memory space to realize timely storing, deleting or updating unbounded and continuous incoming stream data;
3) For time-varying property of stream data, it usually needs to design a good update strategy, which can realize real-time maintenance or timely updating the frequent itemsets or patterns for dealing with the problem of concept-drifting.

Integrated with the characteristics of stream data and based on the different data models and algorithm's features of existing results, stream data mining can be mainly divided into two categories: data-based category and task-based category [5]. The data-based algorithm is based on different data models.
while the task-based algorithm is based on different application functions.

(1) According to time and space features of related algorithms, data-based category can be further divided into: time-sensitive based sub-category and result-sensitive based sub-category. As a natural extension, time-sensitive based sub-category can be subdivided into three processing time models of stream data: Landmark, Damped and Sliding Windows [6], while result-sensitive based sub-category can be subdivided into two models: accurate result and approximate result.

(2) According to different application functions of data mining, task-based algorithm can be subdivided into four parts: clustering, classification, frequent itemset mining, discovering association rules. Clustering refers to how to partition stream data into different classes according to its different characteristics. Classification refers to identifying the class of stream data. Frequent itemset mining attempts to find out all of the frequent itemsets with the occurrence numbers surpassing a minimum user-specified threshold. Based on frequent itemsets, association rules discovery aims at finding out the associative rule to confirm the real patterns by calculating candidate itemsets’ Confidence or Lift [4].

Recently, many applications prove that meaningful patterns with valuable knowledge often frequently occur in certain scope of time or space, such as network anomaly detection and stock market fluctuating anomaly detection. In Addition, as mentioned above, frequent itemset mining is the necessary premise condition of association rule discovery. Hence, frequent itemset mining is very crucial to stream data mining with an significant status. In general, similar with stream data mining, frequent itemset mining also can be divided into two categories: data-based algorithms and task-based algorithms.

2 The taxonomy of frequent itemset mining

According to the conclusions about stream data mining mentioned above, a standard frequent itemset mining algorithm over stream data shall have the essential features as follow:

1) One-scan to the data. As we have described in Section 1, fast processing and timely responding to users only allow us to scan the database or dataset once;
2) An efficient time window model, such as Sliding Window. In practical applications, the people usually have more concerns about the data flows generated in recent time. Therefore, most of the algorithms of mining frequent itemsets prefer to deploy their mining process based on Damped time model or Sliding Window model. Among that, the algorithm based on Sliding Window guarantees that the discovered frequent patterns occurred in a recent period of time, while the algorithm based on Damped Window better reflects on the fluctuating rule of the data within a long history records by adding a time decaying function. However, it’s difficult for Damped Window model to search the periodic time of patterns. Hence, currently, Sliding Window is the most popular time window model of stream data mining;
3) The exact result or the approximate result with a limited error. Due to that it is impossible to store all the data for stream data mining, so that only the approximate results can be obtained instead of the exact result at most time. However, the approximate results may affect the accuracy of mining patterns, so that it is very necessary to limit the error for the algorithms. Consequently, choosing a suitable method of extracting or selecting synopsis data structure with more accuracy becomes a good feasible choice to develop frequent itemset mining. In general, sampling, histogram, wavelet transforming, hashing and load shedding are widely used for achieving this goal;
4) Some specified compact data structure for time and space efficiency, such as Frequent Pattern Tree (FP-Tree) widely used in sequential data mining;
5) One effectively and adaptively updating process or strategy of frequent patterns that is used to deal with the problem of concept-drifting in stream data. Most of time, the updating process will be activated periodically according to time or batch of the data, or when a user-specified condition is met;
6) Low computation complexity enough to suit for real-time response to users.

As described above, frequent itemset mining can be divided into two categories similar with stream data mining: data-based algorithms and task-based algorithms. The data-based algorithm attempts to extract synopsis data structure over the entire stream dataset to mine frequent itemset by adopting some specific data models to select or transform the data, such as sampling or sketching, etc. The task-based algorithm aims at improving performances of the space and time efficiencies by modifying existing results or designing a new one.

According to different models of processing or pre-processing the data, the data-based algorithm can be further subdivided into six sub-categories: sampling, histogram, wavelet transforming, hashing, sketching and load shedding. According to different targets of mining process, the task-based algorithm also can be further subdivided into three sub-categories: general frequent itemset mining, maximum frequent itemset mining and closed frequent itemset mining.

3 The data-based algorithm

The data-based algorithm focus on projecting some subset of the data or summarizing the data to resolve the essential issue of stream data mining, which need effectively balance the contradiction between limited memory space and unbounded volume of the data. There are six related methods widely used in the current results, including sampling, histogram, wavelet transforming, hashing, sketching and load shedding. All the methods can be used to generate synopsis subsets of the stream data for convenience of mining frequent itemsets.
Among them, sampling and hashing are two of most popular methods for stream data mining.

3.1 Sampling
As one kind of classical statistical techniques [7,8], sampling enables to generate synopsis information of stream data by randomly extracting some sampling fractions to represent the whole dataset with similar statistical characteristics. And then the final patterns are mined over the extracted sample data. According whether or not the probability of selecting data fractions are same, sampling can be divided into uniform sampling and bias sampling. In uniform sampling, each fraction of the data will be selected with the same probability, such as reservoir sampling [9] and concise sampling [10], while different fractions are selected with different probability in bias sampling, such as counting sampling [10]. Comparing to reservoir sampling, concise sampling has better space efficiency. Count sampling further improves concise sampling, so that it can effectively obtain the list of frequent patterns or itemsets in stream data.

3.2 Histogram
Histogram divides a large dataset into many continuous buckets (also called as small dataset), and each bucket is denoted by a digital number to represent its characteristic. Histogram has four sub-categories, equi-width histogram [11], compressed histogram [12], v-optimal histogram [13] and end-biased histogram. It enables to intuitively and concisely outline a large data set, so that it has been widely used in many commercial databases. Since synopsis data usually cannot include all the characteristics of the dataset, approximate results will be produced when using such data structures.

3.3 Wavelet transforming
Wavelet transforming is a general digital signal processing technique. Similar with Fourier transforming, wavelet transforming can transform the inputted analog quantity into a series of small wave parameters, and approximately restores the original signal [14] according to the extracted top \( n \) high energy parameters. It has been widely applied to the field of data or signal processing. There are many kinds of wavelet and Harr wavelet is one of the most famous wavelet.

3.4 Hashing
Adopting hashing function to generate synopsis data structure in the field of computer is a very common way. As a result, for saving the memory space or simplifying computation complexity, it is often used to facilitate the process of data mining in stream data either. For example, the Bloom Filter [15] uses a small piece of memory to represent a large dataset with the size far more than its memory. And [16] improves [15] by using counters to replace bits at every position, so that it not only can judge whether or not there are elements, but also can estimate the elements' value.

3.5 Sketching
Sketching [17, 18] is the process of randomly project a subset of the features of the data by vertically sampling stream data. The major drawback of sketching is lack of some accuracy so that it is difficult to adopt it to mine the patterns in the context of stream data. Principal Component Analysis (PCA) [19] maybe is a better solution that has been applied in many applications of stream data.

3.6 Load shedding
Load shedding [20] refers to the process of dropping a sequence of stream data. However, there are some probabilities to drop useful chunks of stream data for it, so the same problems as sampling makes it difficult to be used for exactly mining itemsets in stream data either.

The methods mentioned above are used to generate some approximate synopsis data structure of stream data, so that it's difficult to guarantee the accuracy of mining results for them. With the advances of stream data mining, exact results or approximate results with more accuracy gradually becomes the new goal of frequent itemset mining in the future.

4 The task-based algorithm
According to different intra structures of the mining target, the task-based algorithm can be separated into three parts, general frequent itemset mining, maximal frequent itemset mining and closed frequent itemset mining.

(1) General frequent itemset mining: The related algorithms enable to discover frequent itemsets that are supported by at least a min-sup (Minimum support, a user-specified threshold). Currently, most of the results of stream data mining belong to it;

(2) Maximal frequent itemset mining: If any supersets of a frequent itemset are non-frequent itemsets, then this itemset will be called as maximal frequent itemset, denoted by the MFI. Mining maximal frequent itemsets can obtain the advantage of a relatively small number of candidates according to Apriori Theory. Therefore, compared to mining frequent itemset and mining frequent closed itemset, if the entire stream data contains many longer frequent patterns with a lot of items, mining maximal frequent itemset is a very effective method to improve its time and space efficiency. However, mining the set of MFIs instead of general frequent itemsets will cause to the problem of losing the supports of its subset of frequent itemsets;

(3) Closed frequent itemset mining: If there is no true super-itemset \( Y \) of \( X \), and \( X \) and \( Y \) own the same support in dataset \( S \), while \( X \) ‘s occurrence number is more than the minimum support user-specified, \( X \) will be regarded as a closed frequent itemset of the data set \( S \). To discover all the closed frequent itemsets is a very useful and efficient method for increasing time and space efficiency of stream data mining. According to its definition, it is easy to know that closed frequent itemset mining is more easier than maximal frequent itemset mining to obtain all
fundamental to stream data mining tasks. Most of the frequent patterns can be recovered by its corresponding set of closed frequent itemsets, it also is regarded as one kind of lossless compression method. As a matter of fact, less numbers of candidate itemsets, lower computation complexity and simultaneously keeping the support value of itemsets are the most valuable advantages of it for stream data mining.

4.1 General frequent itemset mining

Frequent itemset mining has been well recognized to be fundamental to stream data mining tasks. Most of the achievements related to frequent itemset mining in stream data [21-31] focus on this issue. In 2002, Datar proposed Ref. [24] to deal with the problem of maintaining statistics over sliding windows and deduce upper and lower space bounds for various problems. It also considers problems like updating other statistics such as variance, clustering such as maintaining k-medians, etc, in the sliding window model.

In 2002, Manku proposed Ref. [21] and implemented an approximate frequent itemset mining in the stream data. The implemented algorithm uses all the previous historical data to incrementally discover frequent itemsets. In 2008, Mozafari presented Ref. [30] that developed a frequent itemsets mining algorithm over stream data. It proposes to use tilted window model to find frequent itemsets and adopts an incremental algorithm to maintain and update the FP-stream which is used as frequent tree data structure to represent the frequent itemsets.

In 2008, Mozafari developed one AOG-based algorithm [28]: Lightweight frequent pattern mining named as LWF. It enables to find most of frequent itemsets with an approximate solution over the incoming stream data by using adaptation and releasing the least frequent itemsets regularly in order to count more frequent itemsets.

In 2009, Tanbeer proposed an efficient method [27] to discover the complete set of recent frequent itemsets from a high-speed stream data over a sliding window. It presents a Compact Pattern Stream tree (CPS-tree) to capture the recent stream data content and efficiently remove the obsolete, old stream data content. And it also introduces the concept of dynamic tree restructuring in the CPS-tree to generate a highly compact tree structure of frequency-descending at runtime.

In 2010, Lam [29] studied the problem of finding the K most frequent itemsets over stream data for the recently proposed max-frequency measure. Based on the characteristics of an item, the max-frequency of an item is counted over a sliding window, while its length keeps dynamically changing. Besides being parameterless, this way of measuring the support of itemsets is shown to have the advantage of a faster detection to bursts in stream data, especially to heterogeneous itemsets.

4.2 Maximal frequent pattern mining

In 2005, Chang proposed a maximal frequent itemset mining algorithm named as estDec [32], which checked each transaction in stream data without any candidate generation by adopting a prefix tree to store those itemsets needed to be monitored closely in the memory. This method introduces two major operations: delayed-insertion and pruning to minimize the total number of itemsets in the memory, while the former delays to insert a new itemset in a newly incoming transaction until this itemset became significant enough, and the latter responds to prune a monitored itemset when it turned out to be insignificant. However, estDec disables to dynamically resolve the problem caused by the memory overflow due to large size of the prefix tree.

To deal with the problem, Lee proposed an improved method of estDec, called as estDec+ [33], which presented a CP-tree (Compressed-prefix tree structure) to effectively find frequent or maximal frequent itemsets over stream data. Unlike a prefix tree, a node of a CP-tree can merge and maintain the information of several itemsets together. By introducing CP-tree and an adaptive memory utilization scheme, new method enables to maximize the accuracy of mining results within a limited memory space adaptively.

In 2007, based on estDec [32], Woo presented a new method named estMax [34] to mine maximal frequent itemsets over stream data. By modifying the internal node structure of a prefix tree used in the estDec, it is capable to discover all of MFIs without additional superset/subset checking while the original estDec can still work normally. In this method, two additional constructs including maximum lifetime and maximality mark, are introduced to distinguish the possible MFIs from non-MFIs, so that it is possible to trace the set of MFIs without searching the entire prefix tree recursively for superset/subset checking. As a result, its time and space efficiency can be increased greatly with better accuracy of mining results.

Comparing with two-phase data mining methods [32-34], which store meta-itemsets and re-mine internal data structures to mine the meta-itemsets over stream data, INSTANT [35] is a single-phase mining method and it stores the itemsets in the array structure for each support. It enables to obtain the frequent itemsets in a real-time way after scanning the data once. However, it has two drawbacks including a very huge memory space required and no need of efficient superset/subset checking mechanism which is easy to lead to worst performance of mining results when the transaction's average length is large.

4.3 Closed frequent pattern mining

In 2004, Chi proposed the Moment algorithm [36] to mine closed frequent itemsets over a sliding window of stream data. It introduces a compact data structure, the closed enumeration tree (CET), to maintain a dynamically selected set of itemsets which includes four types of nodes: infrequent gateway nodes, unpromising gateway nodes, intermediate nodes, and closed
nodes. Moment judges the closed itemsets indirectly through node property checking and excludes them from the other three types of boundary nodes stored in the internal data structure. Due to that much more information about current closed frequent itemsets is needed to be stored, the need to consume much memory is an obvious demerit, especially when the user-specified support threshold is low.

In 2006, Jiang proposed an algorithm, called CFI-Stream [37], to incrementally mine closed itemsets without the need of any support information. The closed itemsets is maintained in a derived data structure. When a new transaction arrives, it performs the closure checking on the itemsets which means only associated closed itemsets and their support information can be incrementally updated. This algorithm achieves both time and space efficiency, especially when a dataset contains highly correlated transactions. The output of closed frequent itemsets is capable to be implemented in real time based on any user-specified thresholds. Hence, it is easy to resolve the problem of concept-drifting for it.

In 2009, Li proposed an efficient one-pass algorithm, called as NewMoment [38], which can maintain the set of closed frequent itemsets over stream data with a transaction-sensitive sliding window. A new summary data structure named NewCET2, which is based on a prefix tree structure, is presented to maintain the core information of closed frequent itemsets in the recent transactions of stream data. And it also introduced an effective bit-sequence representation of items to reduce the time and memory requirements of sliding the window. Comparing to the Moment, it can not only obtain highly accurate mining results, but also run more faster and consume less memory space.

5 Conclusion
In this paper, an overview of research on frequent itemset mining over stream data is presented. It attempts to introduce both early and recent results related to mining frequent itemsets over stream data. After summarizing the characteristics of stream data and the features of stream data mining, the main issues of frequent itemset mining are listed out, followed by the descriptions of its taxonomy and related hierarchical structure. Based on them, according to data-based algorithms and task-based algorithms, the related methods belonging to the different categories or sub-categories are comprehensively introduced for better understanding. Moreover, it also addresses the merits and limitations of the related algorithms with an overall analysis, which can provide the insight for researchers in designing or developing an appropriate algorithm for different streaming environments and various applications.

Acknowledgements
A Project Funded by the Priority Academic Program Development of Jiangsu Higer Education Institutions (PAPD).

References


