Mining frequent closed itemsets (FCIs) in data streams has become a noticeable and challenging area, which serves as a fundamental and essential technique for various emerging applications, such as sensor networks, network traffic analysis, trend analysis and fraud detection, and stock market analysis [7]. These applications, on one hand, share the common features of data streams: data are coming in a continuous, high-speed and unbounded way with its distribution changing over time, and very often patterns embedded in the most recent data (the current sliding window) is the focus. These characteristics, while pose a great challenge to the mining techniques that are required to achieve high resource-aware efficiency both in time and space, can also be proactively utilized to design an optimal strategy for mining data streams. For example, the sliding window is usually confined to a limited size by bounded and constant memory due to the demand of real-time response. This limitation on memory-based algorithms also provides opportunity to narrow the candidate space for mining.

On the other hand, various applications of data streams are realized in relation to practical context and user-specified needs that may be adjusted during the continuous operation process. Two mostly concerned parameters are the transaction size and support threshold, which vary under different circumstances. For example, in intrusion detection, the support threshold may be low so as to detect abnormalities, and different levels of safety guarantee exist among dynamic network and hierarchical users. There are also some cases that the transaction size is greatly concerned, such as web logs and business transactions where transaction size is large. Moreover, the size information is sometimes not priori due to the unbounded feature of data streams. Therefore, a general approach for mining data streams should take scalability, flexibility and completeness into consideration.

Closure-based mining not only shares attractive features of frequency-based summarization of subsets, but also has outstanding advantages in greatly reducing the number of frequent sets, maintaining complete information and being able to generate non-redundant association rules. Therefore it draws much attention, especially in the field of traditional database [1-3]. In data streams algorithms of mining FCIs are categorized into exact or approximate approaches. Representatives of the former are Moment [4] and CFI-Stream [5]. Moment keeps much extra information besides the current frequent closed itemsets and updates the FCIs through switching the states among four types of nodes (i.e. the infrequent, unpromising, intermediate and closed nodes) by complex rules, which is both time and space consuming especially when the support threshold is low. CFI-Stream stores only closed itemsets in memory and can output current FCIs based on user-defined thresholds, but has to check all the subsets of the incoming transaction. IncMine[6] is a typical false-negative approximate algorithm with the notion of semi-FCIs that increases the minimum support threshold for an itemset as it is retained longer in the window where performance is related to the given parameter of relaxation rate as well as minimum support threshold.

In this paper, we address the problem in a distinct way. Firstly, we utilize the limitations on the sliding window at initial to produce the closed itemsets introduced by the incoming/outgoing transaction through just one direct scan of the current window, which greatly narrow down the checking space. Secondly, the subsets generated are in order such that the whole process of updating the existing closed itemsets and identifying new ones is implemented linearly and efficiently. Thirdly, the interdependent relationship between addition and deletion of the closed itemsets over the sliding window is explored and realized by keeping only one extra attribute produced along with the initial scan. Based on these points we develop an algorithm, called SWM, to incrementally maintain the closed itemsets over the sliding window in data streams and flexibly output the frequent closed itemsets according to user-specified support threshold.

REFERENCES


