CASW: Context Aware Sliding window for Frequent Itemset Mining over Data Streams

V. Sidda Reddy\textsuperscript{1}, T.V.Rao\textsuperscript{2} and A.Govardhn\textsuperscript{3}

\textsuperscript{1}Department of CSE, Sai Tirumala NVR Engineering College, A.P, India.
\textsuperscript{2}Department of Computing, K.L.University, A.P, India.
\textsuperscript{3}Department of CSE, JNTUHE, T.S, India.

Abstract

In recent years, advances in both hardware and software technologies coupled with high-speed data generation has led to data streams and data stream mining. Data generation has been much faster in data stream applications and scores of data is generated in quick turnaround time. Hence it becomes obvious to perform mining, data on arrival that is usually termed as data stream mining. General frequent pattern mining methods are envisaging limitations and do not support in responding to a massive quantum of data being streamed. In order to address such limitations, data mining researchers have focused on methods for conducting more efficient and effective mining tasks by scanning a database only once. As a process of evolution, sliding window model that perform mining operations focusing on updating accumulated parts over data streams, are proposed. It is hard to consider all of the frequent patterns in data stream environment as generated patterns were remarkably increasing as data streams get extended continuously. Hence, methods for efficiently compressing patterns that are generated are essential to address the limitations. Considering the challenges and shortcoming in the earlier solutions, in this paper, focus is on incremental mining of frequent patterns from the window and a solution of CASW (Context Aware Sliding Window) is proposed. There are well defined boundaries for frequent and infrequent patterns for specific patterns. In this research article, we adapt usage
of window size change for representing conceptual drift in the information stream. An experimental study carried out on the model depicts significant developments and has affirmed that the algorithm has been designed with a more efficient system than that of existing solution.

**Keywords:** Data Mining, Data Streams, Data Stream Mining, Frequent Itemset, Frequent Itemset Mining, Variable Window, Sliding Window.

1. **INTRODUCTION**

Data mining solutions have become an integral part of analytical solutions and knowledge discovery process. Different data mining solutions have been applied for mining interesting patterns on huge amount of data. Among different data mining solutions frequent pattern mining is one of the significant areas of research with varied approaches and widely accepted in vivid areas of industrial sectors. Numerous algorithms were developed to find frequent patterns, a few of the well-known fundamental frequent pattern mining algorithms are Apriori [1] constituting Breadth First Search and FP-growth [2] that is reliant on Depth First Search. Many improvised solutions have been developed based on the aforesaid fundamental algorithms. For instance, certain algorithm models like frequent pattern mining comprising a minimum support threshold that is specified by users [3], [4], [5] and sequential frequent pattern mining [6], [7], [8]. Frequent pattern mining is utilized in vivid range of applications including in the medical domain [9] weblog and web page click analysis [10], [11] and many other applications as well.

Apart from applying frequent pattern mining in the databases that are static, it is also applied to the data streams with high score volume of data that is constantly updating in to the data systems in a real-time environment, which signifies continuous and unlimited features. However, certain requirements that are to be addressed by data stream mining are [12]:

- Only once the data elements required for data streaming has to be analyzed.
- Despite the fact that there is constant stream of data inflow in to the database, still the issues like the memory usage for mining has to be limited for having acceptable and constant range.
- The data elements that are updated in to the information system have to be processed in quick turnaround time.
- Results of data stream analysis should be of very insightful and resourceful and it can be accessible instantly.

The frequent pattern mining methods that were proposed in earlier times do not adhere to such standards and many methods scan database multiple times to mine frequent patterns. Some of the methods that could be effectively resourceful for mining approaches, and the ones that could be resourceful in addressing the
requirements are proposed in few of the studies [13], [14]. However the proposed data stream mining methods has succeeded in extraction of frequent patterns over data streams effectively, still one of the key issues that are encountered is with the data elements getting added regularly, and the size of the data is also raising high, it becomes significant challenge in mining the time spent on accumulation of transaction data, and thus resulting in lapse in addressing one of the requirements of data streaming and also the immediate processing. In order to address such issues, CFP and MFP notations[4], [12], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] representing more compact forms of general frequent patterns, which can be utilized.

The MFP notation assures more efficient pattern of compression with more compact forms than the CFP notation, however there is a slight pattern loss that occur when the conversion of general ones to MFPs takes place. Also, if MFP method without outstanding compressibility is adapted for data streaming, valid patterns over data streams with more efficient system is observed. As data is accumulated over data streams continuously, importance of certain data in the system might be ignored or no longer be essential and relatively the importance of recently added data might be high.

Many window model based mining approaches were proposed in [29] [30], [31], [32], [12], [33], [34], [35], [14], [5] for the aforesaid context. Predominantly, the fixed window, sliding window and adaptive window techniques are chosen as per the requirements of data streams. Data streams are composed of varied items and every item denotes certain objects from real-world. For instance, in retail market data stream, item could represent regarding products in the supermarket section, in the case scenario of medical records, it could be records of the patients with varied classification etc. and significance to each item might actually be different. Hence, obtaining high-quality mining results reflecting not only items, frequency and also their importance by working on a weight factor in the data stream mining.

In this paper, the model is proposed for Mining Frequent Itemsets using Variable Window Size fixed by Context Aware Sliding Window (CASW). The proposed solution shall be effective in memory usage and computational complexity. Context variation analysis is used in the solution proposed reliant on dynamic window oriented transaction storage is used in the solution proposed and the mining of frequent item sets from the concluded window are performed using TIFIM [36].

2. RELATED WORK

In the contemporary models from the recent past, Apriori [1] proposed model performs only in the static database solutions using BFS (Breadth First Search) strategy, which scans database recursively. In the instances of larger transactions in the database, scanning task to be implemented with varied number of items present. Whereas in FP-Growth [2] DFS (Depth First Search) is adapted for addressing the issues quoted in the aforesaid model. As two fixed database scans are adapted for mining by FP-Growth, algorithm effectively performs and do not generate any candidate patterns.
Though mining methods that are reliant on FP-Growth have an effect on static databases, it might not right fit model for data streams that are accumulating data in continuous manner. The other challenge is that it may not deal with data streams instantly, as these methods have to perform scans of more than two databases. In the other dimension, as the trees are constructed with items that are remaining after infrequent items that are deleted, the trees that are generated earlier had to be discarded and new trees have to be built again whenever the new transaction data are added in to data streams. In data streams, though a certain item might be infrequent in current time frame, it might turn to be a frequent one according to addition of new transaction data.

But the challenge is that, the model for reading the databases right from start has to run both the scan based methods as they have already excluded the infrequent items in the earlier steps. For addressing such issues, mining methods comprising suitable data streams are proposed [37], [38], [13] that can perform mining tasks having only one database scan and also respond to the changes of data streams in immediate manner.

Accordingly, the sliding window-based frequent pattern mining approach is proposed [30], [31], [32], [12], [33], [34], [35], [14], [5] which mines frequent patterns based on the latest transaction data of large data streams.

In [13], [14] it is discussed that if the general data sizes are large, implementation of static database that can cause computational overheads. In sliding window-based data stream models, only the latest windows are considered and the earlier ones are ignored, the overheads can be reduced, but still avoiding them could result in increasing size of windows or the volume of windows might go up.

Considering such implications, it is imperative that MFP notation which can compress the frequent patterns that are generated in to small forms that are compressed can be used for a mining process and many MFP based mining methods were proposed [15], [16], [12], [18], [17], [19], [20], [21], [22], [23], [26], [27], [28].

Vertical bitmap representation is adapted by MAFIA [20] which reduces the tree traversals. Once the bitmap is constructed, MAFIA can observe pattern’s frequency thorough and also the operation of bitmap despite that it might not traverse the trees. One of the significant methods of state-of-art MFP mining algorithm is FP Max [17] in which an additional data structure for mining MFPs in quick turnaround time was proposed. The model succeeds in decreasing tree traversal times in considerable manner. As FP-array has information of patterns, algorithm can more easily estimate them early even before the trees are actually traversed whilst working on the growth. Also, the technique not only reduces effectively the tree traversal operations, it also supports in pruning efficiency by curtailing any needless conditional trees being generated. However, the challenge in the aforesaid model is that they are profoundly the scan based processes and shall not be appropriate for process of data stream mining.
3. CONTEXT AWARE SLIDING WINDOW FOR FREQUENT ITEMSET MINING OVER DATA STREAM.

Multiple scan models are not suitable for frequent itemset mining over data streams as they do not consider the tuples of streaming transactions, but the data streams should consider tuples of records as windows for the mining algorithms. The key element of the proposed model is fixation of window size in the case of data streams which comprises transitional and temporal kind of state transactions. In the other way, for the windows that do not any kind of transitional or temporal state for identifying streaming actions for streaming, fixing of window size shall be a major challenge towards attaining quality aspects like accuracy of results and stability of the process. Taking such factors in to account, the contemporary model of context variation based dynamic window size fixing approach for mining frequent item sets over the data streams, and the proposed solution focus on the following attributes:

- Optimal and Dynamic window size is necessary.
- Size of window should be fixed dynamically, amid of the threshold limits considered.
- Minimal and Maximal size range has to be considered whilst fixing the window size.
- It should be fixed on the basis of context variation observed for input transaction.

To implement the proposed model, key constraints envisaged are about handling context change issues for transactions. Computational cost and minimal memory utilization of two key metrics adapted in the proposed solution of context aware window sliding, to overcome the constraints.

Proposed window fixing strategy exploration is carried out as follows:

Let $D$ be the Data stream and stream transactions as horizontal partitions of the transactions, one transaction is comprised for every partition. When $n$ be attributes total count, for forming the transactions by $D$. Let $a_{1},a_{2},......a_{i},a_{i+1},......a_{n}$ be the attributes set that are used for forming transactions, then $\{t_{1},t_{2},t_{3},......t_{i},t_{i+1},......t_{\text{min}}\}$ be the transactions streaming in the similar sequence. When minimum window size be $\text{min}$ and the maximal window size be denoted as $\text{max}$ ,then $w_{t}$ be the transaction window and $w_{c}$ shall be context change analysis window. The initial values of $\text{min}$,$\text{max}$ shall be set with pre-processing step.

The transactions of count $\text{min}$ generated from data stream $D$ shall be initially moved to $w_{t}$, accordingly the further transactions of count $\text{min}$ shall be moved to $w_{c}$. Context similarity analysis process shall be initiated and the exploration of context similarity is carried out as follows:
The attributes set $a(w_t)$ contains all of the attributes, which are found in the transactions of window $w_t$, similarly the attribute set $a(w_c)$ contains all of the attributes comprised to form the transactions found in $w_c$. Further the hamming distance between these two sets will be assessed that explored in sec 3.1.

### 3.1 Assessing Context Similarity

Value of hamming distance is observed in the model for denoting the contextual difference amidst attributes $a(w_t)$ and attributes $a(w_c)$. It is a one of the key method adapted for assessing the difference between elements in coding theory. Such strategy is applied for identifying the distance between unique values that are identified in transaction window $w_t$ and context variance analysis window $w_c$.

For two vectors chosen, $a(w_t) = \{t_1, t_2, \ldots, t_n\}$ and $a(w_c) = \{c_1, c_2, \ldots, c_m\}$ of size $n$ and $m$ respectively. Hamming distance is assessed as follows:

$$\text{hd}_{a(w_t)\leftrightarrow a(w_c)} = \sum_{j=1}^{\max(m,n)} \left\{\begin{array}{ll} 1 & \text{if } r_i \equiv c_j \\ 0 & \text{otherwise} \end{array} \right.$$  \hspace{1cm} (1)

$\text{hd}_{a(w_t)\leftrightarrow a(w_c)}$ is the hamming distance between $a(w_t)$ and $a(w_c)$.

In furtherance contextual similarity amidst of two windows are evaluated as:

$$S_{a(w_t)\leftrightarrow a(w_c)} = \frac{1}{\text{hd}_{a(w_t)\leftrightarrow a(w_c)}}$$

If similarity score $S_{a(w_t)\leftrightarrow a(w_c)}$ is higher than the given similarity score threshold $s\tau$ in such instances, transactions of $w_c$ will be moved to $w_t$ (2).

$$w_t = w_t \cup w_c$$  \hspace{1cm} (2)

The window $w_t$ will be finalized with the number of records, are max size defined or above and then $w_t$ will be used as input to the process of mining frequent itemsets.

When the $w_t$ gets finalized and frequent itemsets mining are originated, then $w_t$ and $w_c$ shall be cleared and the process exploration for preparing window $w_t$ will be sustained in further transactions streaming from data stream $D$.

The aforesaid process is performed till the transactions are observed using the data stream $D$. The mining frequency item sets gathered from the window that is finalized will be carried out with TIFIM, wherein the earlier research contribution [36], the tree based incremental frequent item sets mining approach has been proposed.
3.2 Estimating the window size by context variation

Let notations $D$, $w_i$, $\min$, $\max$, $s$ and $c_w$ be the data stream, transaction window minimum records count of the $w_i$, maximum records count of the $w_i$, similarity score threshold, and cached window of the context variation analysis respectively.

1. Begin
2. $\forall \{r_i | r_i \in D\}$ Begin //For each transaction of the given data stream $D$
3. If $(|w_i| > \min)$ Begin
4. $w_i \leftarrow r_i$
5. If $(|w_i| \geq \min)$
6. $s \leftarrow \text{contextSimilarity}(w_i, w_c)$ (see sec 3.1)
7. if $(s \geq s\tau)$ Begin
8. $w_i \leftarrow w_c$ // move all transactions of $w_c$ to $w_i$
9. If $(|w_i| \geq \max)$ Begin
10. Finalize window $w_i$
11. Initiate $\text{TIFIM}(w_i)$
12. $w_i \leftarrow \phi$ // empty $w_i$
13. $w_i \leftarrow \phi$ //empty $w_c$
14. End of 10
15. End of 8
16. Else Begin
17. Finalize window $w_i$
18. Initiate $\text{TIFIM}(w_i)$
19. set $w_i \leftarrow \phi$ // empty $w_i$
20. set $w_i \leftarrow w_i$ //move transactions of window $w_i$ to new window $w_i$
21. set $w_i \leftarrow \phi$ //empty $w_c$
22. End of 17
23. End of 3
24. Else // of condition in line 3
25. $w_i \leftarrow r_i$
26. End of 2
27. End of 1

3.3 Frequent Itemsets Mining using (TIFIM)

Primary representations pertaining to transaction of a data stream $D$ has been detailed above. Asynchronous parallel processes are carried out for indentifying the frequent item sets in an incremental phase.Bush indicates item sets comprising two attribute pair like the attributes belonging to $f_i$ and transactions constituting the pair. For measuring the frequency of item sets, coverage can be considered and set in as context of window $w_i$ size. Coverage of two
attribute item sets with count of childs towards a bush is denoted for all the pair of attributes.

FIF (Frequent Item Sets Finder) which is an asynchronous parallel process is performed as follows:

- Initially it selects the bushes comprising coverage more than the chosen coverage threshold \( cv \).
- Develop new bushes from every two bushes by union of roots and intersects the Childs, and shall retain them in instance of a bush coverage which is new, being higher or equal to \( cv \) else discards.
- Process is continued until no new bush is formed.

### 3.3.1 Process of Pruning

A bush \( b_i \) considered to be sub-bush to bush \( b_j \) if \( r_i \subseteq r_j \) any \( cv(b_i) \leq cv(b_j) \). As sub-bush \( b_j \) denoted by \( b_j \), then bush \( b_j \) shall be pruned from the bush-set \( B \).

### 3.3.2 Find frequent items

During an instance, frequent item sets are reviewed as:

Roots of bushes comprising higher value than the given \( cv \) shall be claimed as frequent itemset.

A bush ‘\( b_j \)’ coverage shall be detected as:

If a bush \( b_j \) institute to be such that \( b_i \subseteq b_j \) and coverage value of \( b_j \) turns higher than any other bush \( b_k \) such that \( b_i \subseteq b_k \), in such instance, the coverage of \( b_i \) said to be

\[
(cv(b_i) + cv(b_j))
\]

### 4. EMPIRICAL ANALYSIS

#### 4.1. Characteristics of a Dataset

Numerous sets of data are streamed for carrying out the experiments and the characteristics of the data streams are assessed as follows:

To achieve sparseness in the streaming transactions, fields that are in range of 75, 100, 125 and 150 are considered for max transaction length, and the range of 12 to 18 are sent in range, with minimum transaction range set as 5. Total numbers of transactions are set in the range of 1000 to 10000.

For achieving the denseness of streaming transactions, fields in the range of 20, 30, 40 and 50 are chosen, and the range of 10 to 15 for maximum transaction length, with min transaction range set as 5, the total numbers of transactions are taken in the range of 1000 to 10000.
4.2. Experimental results

Experimental results of the proposed solution are compared with the frequent item sets mining model for data streams that are devised in [29] the model of MFIM (Matrix based frequent item sets) are carried out in Java 7, and set of flat files as streaming data sources. Streaming environment is enumerated using Java RMI and also parallel process is adapted for the CASW which is achieved using Java Multi-threading concept. Three parameters for every synthetic dataset are cumulative of the aggregate volume of transactions. Every transaction of a data set is scanned only once in the experimental study, for simulating the data streams environment. For measuring computational cost and scalability, algorithms are performed under vivid coverage values in range of 10% to 90%.

![Figure 1: CASW advantage over MFIM in Memory usage.](image1.png)

![Figure 2: CASW advantage over MFIM in terms of execution time](image2.png)
Figure 1 and 2 depict the comparative analysis of memory usage and execution time under vivid range of coverage values in range of 10% to 40% respectively. Figure 3 denotes the scalability in both the models of CASW and MFIM and how CASW has outperformed the other model.

For graphical representation in Figure 1 and 2, the coverage given is denoted in horizontal axis and the memory in units of MBs and time mentioned in unit seconds are denoted in vertical axis respectively. Streaming of data size denoted in units of transactions is detailed in horizontal axis in Figure 3, and execution times denoted in units of seconds are represented in vertical axis. Also the percentage of time elapsed is also represented in units of seconds. With the decreasing value of coverage, the avg. increment in memory usage for matrix that is based in FIM and the CASW are 2.29 and 0.7 respectively.

Also, the average execution time increment for the matrix that is reliant on FIM and CASW has resulted as 83.2 and 27.9 respectively. Results that are obtained from the experimental study indicate performance of CASW, which has outperformed the matrix based FIM.

Performance of CASW is scalable as matrix, when FIM is taking average of 14.16% elapsed time in the instance of increment of streaming data size comprising 1500 transactions.

Figure 3: CASW advantage over MFIM about Scalability under divergent streaming data size.

5. CONCLUSION

In the proposed model of contemporary approach for mining the frequent item sets from a data stream, the model of an efficient tree based incremental frequent item sets mining model TIFIM [36] proposed in earlier research paper, has been further developed for an approach of mining frequent item sets by using Contest Aware
Sliding Window over the data streams. Using the concept variation analysis, the factor of fixing window size in dynamic manner, the solution proposed has been more resourceful in terms of optimality and scalability. Also, a parallel process determining frequent item sets using the concept of cached bush structures as proposed in our earlier proposal model TIFIM [36] it performs frequent item sets mining over the data streams.

In this paper, the solution is extended by introducing windowing the streaming transaction with sliding window size technique for achieving effective usage of memory and execution time. Experimental results from the study confirm that CASW is scalable with coverage values and also with divergent streaming data size. In the future model, the solution can be extended to perform utility based frequent item set mining over the data streams.

ACKNOWLEDGMENT

The authors would like thanks to the anonymous reviewers for their valuable comments. We also grateful to the authors of all reference journals for valuable information and helping us set up the paper.

REFERENCES


Farzanyar, Zahra, Mohammadreza Kangavari, and Nick Cercone. "MaxFISM: Mining (recently) maximal frequent itemsets over data streams using the sliding window model." Computers & Mathematics with Applications 64.6 (2012): 1706-1718.


[32] Deypir, Mahmood, Mohammad Hadi Sadreddini, and Sattar Hashemi. "Towards a variable size sliding window model for frequent itemset mining"


[38] Chen, Lei, and Changliang Wang. "Continuous subgraph pattern search over certain and uncertain graph streams." IEEE Transactions on Knowledge and Data Engineering 22.8 (2010): 1093-1109.