CCA-VW-FIM: Concept change aware variable window based frequent itemsets mining over data streams

D.Sujatha, B.L.Deekshatulu

Abstract— Considering the continuity of a data stream, the accessed windows information of a data stream may no longer useful as a concept change is effected on further data. Variable size sliding window is used for performing data stream mining which is suitable for observing recent changes in the set of frequent itemsets over data streams. The windows size id determined dynamically based on amounts of concept change that occurs within the arriving data stream.The window expands as the concept becomes more stable and shrinks when a concept change occurs. Using a variable size window, results in vast usage of memory as the window size increases gradually for a set of frequent items. To reduce memory usage of the frequent itemsets a time based window is used and a variable window is implemented as a concept change occurs in the recent window.

Index Terms— data stream mining; frequent itemsets; variable size window; concept change

1 INTRODUCTION

The notion of context is a set of hidden circumstances that entail a certain set of high-support itemsets. For instance, contexts can be defined by seasons or demographic factors. Although the context can change in time, the user may not know about it.

After the arrival of the new block of transactions, the system updates the support values in the current list of itemsets, and then decides whether this list has been significantly altered. Significant change would indicate that the underlying context might have changed.

A data stream is a massive unbounded sequence of data elements continuously generated at a rapid rate. Consequently, the knowledge embedded in a data stream is likely to be changed as time goes by. However, most of mining algorithms or frequency approximation algorithms for a data stream should be able to extract the recent change of information in a data stream adaptively, which in fact not found in stated models in recent literature.

A data stream is a huge unbounded sequence of data elements always generated at a rapid rate. Due to this reason, it is impracticable to maintain all the elements of data streams [1]. This rapid making of continuous streams of information has challenged our storage, computation and communication capabilities in computing systems. The main challenge is that 'data-intensive' mining is embarrassed by limited resources of time, memory, and illustration size. Data Stream mining refers to informational cofiguration extraction as models and patterns from continuous data streams. Traditional methods would require the data to be first stored and then progression off-line using complex algorithms that make several pass over the data, but data stream is endless and data generates with high rates, so it is impossible to store it [12].

- *D.Sujatha is currently pursuing Ph.D program in computer science and engineering inJNTUniversity,Hyderabad, India, E-mail: sujatha.dandu @gmail.com*
- *B.L.Deekshatulu Distinguished Fellow, IDRBT. E-mail: deekshatulu @hotmail.com*

2 EXISTING SYSTEM:

Chang and Lee (2005) proposed the algorithm called estWin which uses sliding window to find the recent frequent patterns adaptively over the data stream. This algorithm defines another support known as significant support in addition to minimum support to approximate the frequent patterns over the data stream. Tsai (2009) proposed a framework using the weighted sliding window model. In this model the user is allowed to specify the size of the window, number of window and a weight for each window. Thus users can achieve significant data by specifying a higher weight, which are the user's requirements. Li and Lee (2009) proposed MFI-TransSW. This algorithm is based on the Apriori algorithm. This algorithm finds all the frequent itemsets in the recent window of transactions and uses a bit string for every item in the window. In the bit string of an item in a transaction shows '1' if it appears and '0' if it is not present in another transaction. Leung & Khan, (2006) proposed an algorithm namely DSTree. This algorithm divides all the transactions into a number of batches(or panes) in a window and uses Prefix tree to maintain the information. Every node in the prefix tree shows items in the transactions and are sorted in canonical order. Batch by batch the transactions are inserted and removed from the tree and frequent itemsets are mined using FP-Growth method. Another recently proposed prefix tree based algorithm is the CPS-Tree (Tanbeer et al., 2009) which is similar to the DSTree algorithm. This approach maintains the support in the descending order nodes and dynamically reconstructs the prefix tree.It uses pre-

fix tree to reduce the memory usage and enhance the time for mining. Recently, a sliding window based method for frequent itemset mining has been proposed (Deypir & Sadreddini, 2011) which is based on dynamically maintaining the transactions by using lists. The SWIM (Mozafari et al., 2008) is a pane based algorithm in which frequent itemsets in one pane of the window are considered for further analysis to find frequent itemsets in whole of the window. It keeps the union of frequent patterns of all panes and incrementally updates their supports and prunes infrequent ones. Chang and Lee (2006) introduced an algorithm called estDec based on time decay model in which each transaction has a weight decreasing with age. In this method, in order to reduce the effect of old transactions in the set of frequent patterns a decay rate is defined .In Woo and Lee (2009) algorithm was proposed which was similar to the estDec algorithm which was proposed for mining maximal frequent itemsets on damped model over data stream. Mahmood Deypir, Mohammad Hadi Sadreddini, Sattar Hashemi (2013) proposed VSW (Variable Size sliding Window frequent itemset mining) which is suitable for observing recent changes in the set of frequent itemsets over data streams. The window size is determined dynamically based on amounts of concept change that occurs within the arriving data stream. The window expands as the concept becomes stable and shrinks when a concept change occurs. In all of the above mentioned methods for sliding window time and in decay algorithms, the window size or decay rate must be determined initially and remains fixed during data stream mining. Prior setting of these parameters is trivial for the user. If the information about the rate of change in the stream is known, then the user can set a suitable value for the above parameters. This can be done by observing the recent changes in the incoming data with respect to these parameters. But in a data stream, the rate of change is unpredictable and can vary as time passes in the stream. If the size of the window is very large then there may be a context change within the window itself and may contain stale information that belongs to older concept. Hence the sliding window may not reflect the recent change in the data stream. And if the window size is very small then it may drop any new changes in the stream.

3 PROPOSED SYSTEM

If streaming data is input to mining strategies such as frequent itemsets mining, the traditional approaches are not suitable, since those works on by the multiple passes through entire dataset. Henceforth the mining strategies opted for streaming data considers the tuples of the streaming transactions as windows and these windows are used as input to the mining algorithms. The significant issue here in this model is fixing the window size. In the case of data streams with transitional and temporal state transactions, the transitional and temporal state identifications can be used to fix the window size. In the other cases that are not having any transitional and temporal state identities for streaming transactions, fixing window size is a big constraint to achieve quality factors such as result accuracy, process scalability. In this regard, we propose a novel context variation based dynamic window size fixing approach for mining frequent itemsets over data streams. The proposed frequent itemsets mining strategy targets the following qualities:

- The window size should be optimal and dynamic
- The window size should fix dynamically between minimal and maximal size given as thresholds
- The size of the window should be within the range of minimal and maximal size and should fix based on the context variation observed in input transaction from the data stream.

In regard to implementing the proposed model, the only significant constraint related to the streaming data is that the context change of the transactions should be in an order.

The minimal memory utilization and less computational cost are two main quality metrics expected from this proposal. The exploration of the proposed window fixing strategy is follows:

Let ds be the DataStream, and stream transactions as horizontal partitions of the transactions, let each partition having one transaction. Let n be the total count of the attributes used to form the transactions by ds. Let a_{set} be the attributes set that contains attributes $\{a_1, a_2, \ldots \ldots a_i, a_{i+1}, \ldots \ldots a_n\}$, which are used to form the transactions. Let $\{t_1, t_2, t_3 \ldots . t_{i,} t_{i+1}, \ldots . . t_{i+m}, t_{i+(m+1)}, \ldots \ldots\}$ be the transactions streaming in the same sequence. Let WS_{min} be the minimum window size and WS_{max} be the maximal window size. Let W_{tran} be the transaction window and W_{cca} be the context change analysis window. Let S(W_{cca}) be the size of W_{cca}. The initial values to WS_{min}, WS_{max} and S(W_{cca}) will be set during the preprocessing step.

The transactions of count WS_{min} from the given data stream ds will be moved initially to W_{tran}, then following transactions of count S(W_{cca}) will be moved to W_{cca}. Then context variation analysis (CVA) process will be initialized. The exploration of the CVA process is follows:

The attributes involved to generate the transactions moved into W_{tran} will be collected as $al(W_{tran})$, and attributes involved to form the transactions found in W_{cca} will also be collected as $al(W_{cca})$. Then the similarity score of these two attribute lists $al(W_{tran}), al(W_{cca})$ will be found as follows (Eq1), which is derived from jaccard similarity measuring approach.

$$SS_{(W_{tran} \leftrightarrow W_{cca})} = \frac{al(W_{tran}) \cap al(W_{cca})}{al(W_{tran}) \cup al(W_{cca}} \ldots\ldots\ldots [Eq1]$$

If similarity score $SS_{(w_{tran} \leftrightarrow w_{cca})}$ is greater than the given similarity score threshold SS_{τ} then the transactions of w_{cca} will be moved to w_{tran} (see Eq2).

$$w_{tran} = w_{tran} \bigcup w_{cca} \ldots . \text{(Eq2)}$$

If size of the W_{tran} is greater than or equals to WS_{max}then the W_{tran}will be finalized and initiates process of mining frequent itemsets from the transactions of W_{tran}, else the further streaming transactions of size S(W_{cca}) will moved to W_{cca} and continues CVA process.
Once the W_{tran} is finalized and mining of frequent itemsets is initiated, then $W_{tran}$and W_{cca} will be cleared and continues the process explored to prepare the window W_{tran}will be continued for further transactions streaming from data stream ds.
The above said process continues till transactions found from data stream ds. The mining frequent itemsets from the finalized window will be done by using APRIORI or FPTree.

algorithmic exploration of the Fixing Variable Window Size by Context Variation Analysis
Inputs:
Data stream ds
Minimal transaction window size $ws_{\min}$
Maximal transaction window size $ws_{\max}$
Similarity score threshold ss_{τ}
Size of the context change analysis window $s(w_{cca})$
Begin
For each transaction $\{t \forall t \in ds\}$ Begin
If $(|w_{tran}| < ws_{\min})$ $w_{tran} \leftarrow t$
Else Begin
$w_{cca} \leftarrow t$
If $(|w_{cca}| \geq s(w_{cca}))$
$ss \leftarrow CVA(w_{tran}, w_{cca})$
if $(ss \geq ss_{\tau})$ Begin
$w_{tran} \leftarrow (w_{tran} \bigcup w_{cca})$
If $(|w_{tran}| \geq ws_{\max})$ Begin
Finalize window w_{tran}
Initiate $TIFIM(w_{tran})$
set $w_{tran} \leftarrow \phi$ // empty w_{tran}
set $w_{cca} \leftarrow \phi$ //empty w_{cca}
End of 10
End of 8
Else Begin
Finalize window w_{tran}
Initiate $TIFIM(w_{tran})$
set $w_{tran} \leftarrow \phi$ // empty w_{tran}
set $w_{tran} \leftarrow w_{cca}$ //move transactions of window w_{cca} to new window w_{tran}
set $w_{cca} \leftarrow \phi$ //empty w_{cca}
End of 17
End of 4
End of 2
End of 1

algorithmic exploration of the Context Variation Analysis
$CVA(w_{tran}, w_{cca})$ Begin
Set $fs_{tran} \leftarrow \phi$ // initiate field set fs_{tran} of transaction window w_{tran} empty
Foreach transaction $\{t \forall t \in w_{tran}\}$ Begin
$fs_{tran} \leftarrow fs_{tran} \bigcup t$
End of 3
Set $fs_{cca} \leftarrow \phi$ // initiate field set fs_{cca} of transaction window w_{cca} empty
Foreach transaction $\{t \forall t \in w_{cca}\}$ Begin
$fs_{cca} \leftarrow fs_{cca} \bigcup t$
End of 7
$ss = \frac{fs_{tran} \bigcap fs_{cca}}{fs_{tran} \bigcup fs_{cca}}$ //measuring similarity score of w_{tran} and w_{cca}
Return ss
End of 1

Frequent Itemsets Mining (FIM)
Mining Frequent itemsets using FPTree
……..

An algorithmic representation of the caching processes:

Input: At an event of time a window w_i with transaction t_i received

For each transaction t_i:

$$\{(a_1, a_2, a_3, ...a_i) \forall$$
$$(a_1, a_2, a_3, ...a_i) \in t_i \wedge$$
Let set of attributes $(a_1, a_2, a_3, ...a_i) \subseteq A_{set}\}$

For each pair of attributes $\{(a_m, a_n) \forall (a_m, a_n) \in t_i\}$, if found a bush $\{b_i \exists (a_m, a_n)\ as\ root\}$ then add transaction t_i as node to bush b_i, else prepare a bush such that $\{b_i \exists (a_m, a_n)\ as\ root \wedge t_i\ as\ node\}$

4 Empirical Analysis

Dataset characteristics

Multiple sets of data streamed to perform the experiments, and the characteristics of these streaming data are as follows:

- To achieve the sparseness in streaming transactions, the range of fields considered as 75,100, 125 and 150, the max transaction length set in the range of 12 to 18, the min transaction range set to 5 and the total number of transactions has taken in the range of 1000 to 10000.
- To achieve the denseness in streaming transactions, the range of fields considered as 20, 30, 40 and 50, the max transaction length set in the range of 10 to 15, the min transaction range set to 5 and the total number of transactions has taken in the range of 1000 to 10000.

4.1 Experimental Results

We compare our algorithm with frequent itemsets mining model for data streams devised in [14], which is a Variable size sliding window model (VSSWM) for frequent itemset mining over data streams[14] algorithm for data streams. The implementation of our CCA-VW-FIM and model VSSWM done by using java 7 and set of flat files as streaming data sources. The streaming environment is emulated using java RMI and parallel process involved in proposed CCA-VW-FIM is achieved by using java multi threading concept. The three parameters of each synthetic dataset are the total number of transactions, the average length, and divergence count of items, respectively. Each transaction of a dataset is scanned only once in our experiments to simulate the environment of data streams. In regard to measure the computational cost and scalability, the algorithms run under divergent coverage values in the range of 10% to 90%.

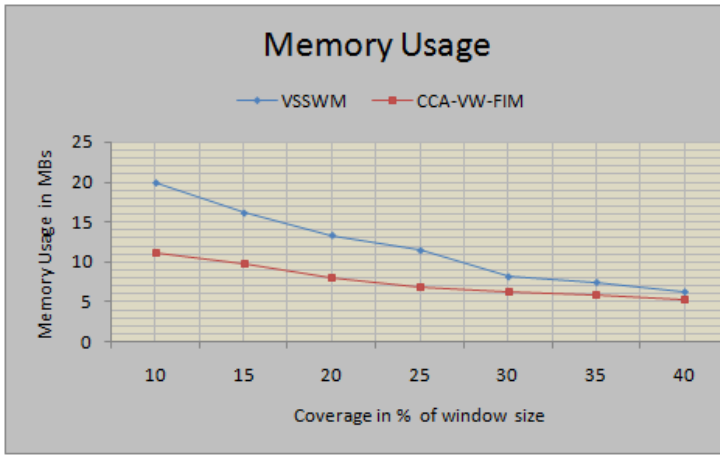

Figure 1: CCA-VW-FIM advantage over VSSWM (Towards a variable size sliding window model) [14] in Memory usage

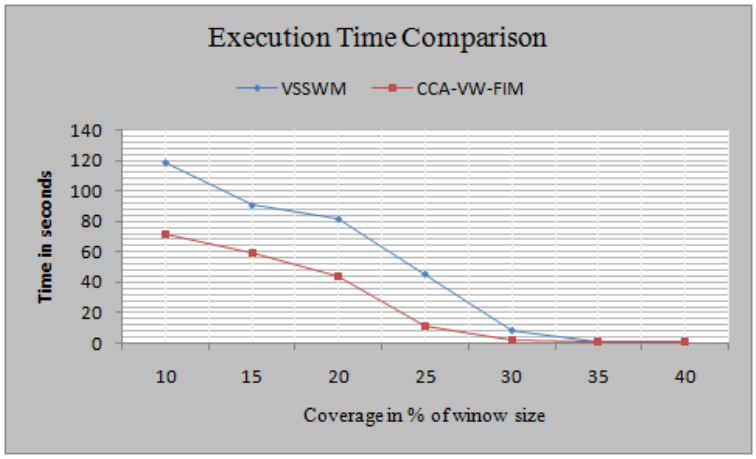

Figure 2: CCA-VW-FIM advantage over VSSWM (Towards a variable size sliding window model) [14] in terms of execution time

Figure 1 shows the comparison of the Memory usage, execution time under divergent coverage values range given from 10% to 40% respectively. In Figure 1, the horizontal axis is the coverage given and the vertical axis is the memory in unit of mega bytes and time in unit of seconds respectively. In figure 2, the horizontal axis is the streaming data size given in unit of transactions and the vertical axis is the execution time in unit of seconds and percentage of elapsed time in unit of seconds respectively. The figure 3 explores the scalability of CCA-VW-FIM over VSSWM under divergent streaming data sizes respectively. As the coverage value decreases the average increment in memory usage for VSSWM and CCA-VW-FIM are 2.29 and 0.7 respectively (see Figure 1) and average execution time increment for VSSWM and CCA-VW-FIM are 83.2 and 27.9 respectively (see figure 2). The results obtained here clearly indicating that the performance of CCA-VW-FIM is miles ahead than the VSSWM. The performance of CCA-VW-FIM is scalable as VSSWM is taking average of 14.16% elapsed time under uniform increment of streaming data size with 1500 transactions.

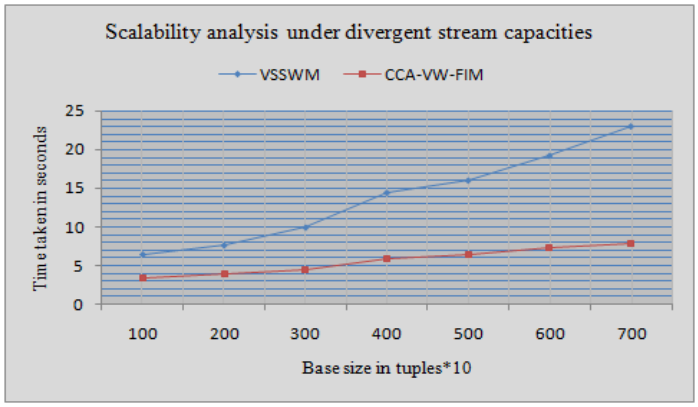

Figure 3: CCA-VW-FIM advantage over VSSWM (Towards a variable size sliding window model) [14] about Scalability under divergent streaming data size

5. CONCLUSION

We explored a novel approach for mining the frequent itemsets from a data stream. We have implemented an efficient tree based incremental frequent itemsets mining model [14] in our earlier research paper, further here we developed an approach for mining frequent itemsets using variable window size by context variation analysis (CCA-VW-FIM) over data streams. Due to the factor of fixing window size dynamically by concept variation analysis, the said model is identified as optimal and scalable. AN apriori or fptree approach is adapted to perform frequent itemsets mining over data streams. We extended VSSWM by introducing windowing the streaming transaction with variable window size technique in regard to achieve efficient memory usage and execution time. The experiment results confirm that the CCA-VW-FIM is scalable under divergent streaming data size and coverage values. In future this model can be extended to perform utility based frequent itemset mining over data streams.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In Proceedings of the 1993 International Conference on Management of Data, pp07-216 ,2003.

[2] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487-499, 1994

[3] J.H. Chang & W.S. Lee, "A sliding window method for finding recently frequent itemsets over online data streams', Journal of Information Science and Engineering, 20(4), 2004, pp. 753–762

[4] Y. Zhu & D. Shasha, "Stat Stream: statistical monitoring of thousands of data streams in real time", Proc. 28th Conf. on Very Large Data Bases, Hong Kong, China, 2002, pp. 358–369.

[5] K Jothimani, Dr Antony Selvadoss Thanmani, "MS: Multiple Segments with Combinatorial Approach for Mining Frequent Itemsets Over Data Streams", IJCES International Journal of Computer Engineering Science, Volume 2 Issue 2 ISSN : 2250:3439.

[6] J.H. Chang & W.S. Lee, "A sliding window method for finding recently frequent itemsets over online data streams", Journal of Information science and Engineering, 20(4), 2004, pp. 753–762.

[7] J. Cheng, Y. Ke, & W. Ng," Maintaining frequent itemsets over high-speed data streams", Proc. 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Singapore, 2006, pp.462–467.

[8] C.K.-S. Leung & Q.I. Khan, "DSTree: a tree structure for the mining of frequent sets from data streams," Proc. 6th IEEE Conf. on Data Mining, Hong Kong, China, 2006, pp. 928–932.

[9] Y. Chi, H. Wang, P.S. Yu, & R.R. Muntz, "Moment: maintaining closed frequent itemsets over a stream sliding window", Proc. 4th IEEE Conf. on Data Mining, Brighton, UK, 2004, pp. 59–66.

[10] K.-F. Jea & C.-W. Li, "Discovering frequent itemsets over transactional data streams through an efficient and stable approximate approach, Expert Systems with Applications", 36(10), 2009, pp. 12323–12331.

[11] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference, was supported in part by the National Science Council in 2006.

[12]] F. Bodon, "A fast APRIORI implementation", Proc. ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03), 2003.

[13] Frequent Itemset Mining Implementations Repository (FIMI). Available: http://fimi.cs.helsinki.fi/

[14] Mahmood Deypir, Mohammad Hadi Sadreddini, Sattar Hashemi, Towards a variable size sliding window model for frequent itemset mining over data streams, Computers & Industrial Engineering, Volume 63, Issue 1, August 2012, Pages 161-172, ISSN 0360-8352, http://dx.doi.org/10.1016/j.cie.2012.02.008.