CCA-VW-FPTPM: Concept change aware dynamic variable window based Frequent Positive Transition Pattern Mining over data streams

Ravali Biravolu
M.Tech in Computer science Engineering
Aurora's Technological & Research Institute,
parvathapur, uppal, Hyderabad-500039

Mrs. Sujatha Dandu
Head of Department Information Technology M.Tech
(PH.D)
Aurora's Technological & Research Institute,
parvathapur, uppal, Hyderabad-500039

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. In order to support frequent item mining over data stream, the interesting recent concept change of a data stream needs to be identified flexibly. Based on this, an algorithm can be able to identify the range of the further window. A method for finding frequent itemsets over a data stream based on a sliding window has been proposed here, which finds the interesting further range of frequent positive transition patterns by the concept changes observed in recent windows. The Patterns of Positive Transitions denote the vibrant nature of the frequent patterns in the database. The considerable high points for the transaction database are the timestamps also called as time durations. They are the points that have the alteration in the recurrence of the prototypes.

Keywords: Data stream mining; frequent itemsets; Variable size window; Concept change

1. INTRODUCTION

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.

Frequent itemset mining is a KDD technique which is the essential of many other techniques, such as association rule mining, sequence pattern mining, categorization, clustering and so on. 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 configuration extraction as models and patterns from continuous data streams. Data Streams have dissimilar challenges in many aspects, such as computational, storage, querying and mining. Based on last researches, since of data stream requirements, it is necessary to design new techniques to replace the old ones. 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].Data from sensors like endure stations is an example of fixed-sized data, whereas again, market basket data are an example of unpredictable size data, because each basket contains a dissimilar number of items. By contrast, sensor measurements have a fixed size, as each set of measurements contains a fixed set of dimensions, like temperature, precipitation, etc. A typical loom for dealing is based on the use of so called sliding windows. The algorithm keeps a window of size W containing the last W data items that have arrived (say, in the last W time steps). When a new item arrives, the oldest constituent in the window is deleted to make place for it. The summary of the Data Stream is at every instant computed or rebuilt from the data in the window only. If W is of moderate size, this fundamentally takes care of the requirement to use low memory. The type of objects i.e. windows in a stream impacts the way the data stream is process. This is due to two facts: We have to handle windows of different types in a different way, and we are able to tailor the processing to the explicit properties of the objects. Hence, for our focus, appealing properties are the size of the windows, what information the objects characterize and how they relate to other windows in the stream.

ISSN: 2278 – 7798

2. RELATED WORK

Chang and Lee (2005) proposed the estWin that finds recent frequent patterns adaptively over transactional data streams using sliding window model. Tsai (2009) proposed a framework for data stream mining, called the weighted sliding window model. Li and Lee (2009) proposed an algorithm namely MFI-TransSW, which is based on the Apriori algorithm. Leung & Khan, (2006) proposed DSTree, another recently proposed prefix tree based algorithm is the CPS-Tree (Tanbeer et al., 2009). Recently, a sliding window based method for frequent itemset mining has been proposed (Deypir & Sadreddini, 2011). The SWIM (Mozafari et al., 2008) is another pane based algorithm. Chang and Lee (2006) introduced an algorithm called estDec based on time decay model In Woo and Lee (2009) algorithm. In all of the mentioned sliding window and time decay algorithms, the window size or decay rate must be determined initially and remains fixed during data stream mining. Setting these parameters is not trivial for the user. The user can determine suitable value for these parameters if information about the time scale of change in the stream is available. This is due to considering the recent changes of the incoming data in these models according to these parameters. In a data stream, the scale of change is unknown and unpredictable. Moreover, it varies within the stream as time passes. If the value of the window size is too large there may exists concept change(s) within the window and thus it contains stale information belong to the older concepts. Therefore, the set of frequent patterns is not accurate with respect to the recent concept and does not reflect recent change as the aim of sliding window model. On the other hand if the window size is too small, information from the new concept are dropped from the window and set of frequent patterns is not dependable and support values are not stable. Similar problem exists about determining decay rate in the time decay model.

The other drawback of these procedures is that they can produce huge amount of the patterns on the condition of low support threshold value. As this drawback being one of the reasons, maximum procedures are not utilized for the data mining job. So, as a case of evading the above mentioned ineffectual and outmoded patterns, the latest patterns closed frequent itemsets [10] were been established.

One of the common characteristic of these procedures is that they don’t determine the time stamps that are related to the actions in the database. This leads to the loss of exposure of the vibrant nature of the patterns. For an illustration let us consider an electronic showroom’s database. The amount of retails of refrigerator during the peak summer season is very much more when compared to that of the retails in the winter. But when we determine all the transactions in whole on an average the retails are frequent, while in actual they are frequent only in the peak season. So in order to find out such a vibrant characteristics, latest patterns have been established like Transitional patterns [13] [14] [17]. There are two kinds in them such as the positive patterns and the negative patterns. Positive transitional patterns have the characteristic of incrementing the recurrence of the pattern’s time stamp where as the negative patterns decrement the recurrence of the pattern’s time stamp.

3. Concept change aware dynamic variable window based frequent positive transition pattern mining (CCA-VW-FPTPM) over data steams

If streaming data is input to mining strategies such as frequent itemsets mining, the traditional approaches are not suitable, since those are mainly works 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 results accuracy, process scalability. In this regard here 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 is centric to following qualities targeted

- 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 $\{a_1, a_2, ...a_i, a_{i+1}, ...a_n\}$, which are used to form the transactions. Let $\{t_1, t_2, t_3 ...., t_i, t_{i+1}, ...t_{i+m}, t_{i+(m+1)}, ...\}$ 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.

$$ss_{(w_{trans} \leftrightarrow w_{cca})} = \frac{al(w_{tran}) \bigcap al(w_{cca})}{al(w_{trans}) \bigcup al(w_{cca})} \text{ …. (Eq1)}$$

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).

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

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 FREQUENT SETS OF transitional patterns from the finalized window will be done by using proposed model explored in section 3.3

3.1 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 $s(w_{cca})$

```
1.  Begin
2.  For each transaction $\{t \forall t \in ds\}$ Begin
3.  If $(| w_{tran} |< ws_{\min})$ $w_{tran} \leftarrow t$
4.  Else Begin
5.  $w_{cca} \leftarrow t$
6.  If $(| w_{cca} |\geq s(w_{cca}))$
7.  $ss \leftarrow CVA(w_{tran}, w_{cca})$
8.  if$(ss \geq ss_{\tau})$ Begin
9.  $w_{tran} \leftarrow (w_{tran} \bigcup w_{cca})$
10. If $(| w_{tran} |\geq ws_{\max})$ Begin
11. Finalize window $w_{tran}$
12. Initiate $TIFIM(w_{tran})$
13. set $w_{tran} \leftarrow \phi$ // empty $w_{tran}$
14. set $w_{cca} \leftarrow \phi$ //empty $w_{cca}$
15. End of 10
16. End of 8
17. Else Begin
18. Finalize window $w_{tran}$
19. Initiate $TIFIM(w_{tran})$
20. set $w_{tran} \leftarrow \phi$ // empty $w_{tran}$
```

```
21. set $w_{tran} \leftarrow w_{cca}$ //move transactions of window $w_{cca}$ to new window $w_{tran}$
22. set $w_{cca} \leftarrow \phi$ //empty $w_{cca}$
23. End of 17
24. End of 4
25. End of 2
26. End of 1
```

3.2 algorithmic exploration of the Context Variation Analysis

```
1.  $CVA(w_{tran}, w_{cca})$ Begin
2.  Set $fs_{tran} \leftarrow \phi$ // initiate field set $fs_{tran}$ of transaction window $w_{tran}$ empty
3.  Foreach transaction $\{t \forall t \in w_{tran}\}$ Begin
4.  $fs_{tran} \leftarrow fs_{tran} \bigcup t$
5.  End of 3
6.  Set $fs_{cca} \leftarrow \phi$ // initiate field set $fs_{cca}$ of transaction window $w_{cca}$ empty
7.  Foreach transaction $\{t \forall t \in w_{cca}\}$ Begin
8.  $fs_{cca} \leftarrow fs_{cca} \bigcup t$
9.  End of 7
10. $ss = \frac{fs_{tran} \bigcap fs_{cca}}{fs_{tran} \bigcup fs_{cca}}$ //measuring similarity score of $w_{tran}$ and $w_{cca}$
11. Return $ss$
12. End of 1
```

3.3 Frequent Sets of Positive Transitions

Input:

A Transactions Window (D), an appropriate milestone range that the user is interested (T_{ξ}), pattern support threshold (t_s), and transitional pattern threshold (t_t). Coverage

Output:

The group of transitional patterns (positive (S_{PTP}) and negative (S_{NTP})) including their significant milestones.

Algorithm:

1. Determine frequent patterns

In order to preserve the individual patterns, snip out the patterns that are as subset for the superset frequent patterns and has the equal support values as follows, which can result the count of the patterns gets reduced.

A pattern can be sniped out if it is a resident of other patterns and their support values are equal.

Assuming that $A \; and \; B$ are a couple of patterns such that

$$\sup(A) \; is \; \omega$$
$$\sup(B) \; is \; \omega^{'}$$
$$and \; \omega \cong \omega^{'}$$

$if \; A \subseteq B \; then \; A$ has a chance to be sniped out from the group of patterns.

2. Patterns of Positive Transitions

Check the positive and negative transition for every pattern in a group of transitional dates tdr_i as follows:

The development in the impact of the transitional patterns is done by checking T_{ξ} for the specified timely intervals of transactions rather than checking the particular time threshold value.

Assuming that $TD = \{td_1, td_2, ...td_n\}$ is a group of dates on which the transitions are occurred,

$$TDR = \{tdr_1, tdr_2, ...tdr_s\}$$
$$here$$
$$tdr_1 = \{td_1, td_2, ...td_m\}$$
$$tdr_2 = \{td_{m+1}, td_{m+2}, ...td_{m+i}\}$$
$$tdr_3 = \{td_{m+i+1}, td_{m+i+2}, ..., td_{m+i+x}\}$$
$$......$$
$$td_n = \{td_{m+i+x} ....td_n\}$$

TD denotes the group of transitional dates

TDR denotes the group of transition interval and every interval consists of the group of transition dates.

At this instant the transition patterns threshold T_ξ checked among the interval Transition range tdr_i.

As a result there is an apparent development in the count of the transition patterns that are based on tdr_i when compared to that of transition patterns based on td_i

$$TP(tdr_i) = TP(td_{i_i}) \bigcup TP(td_{i_2}) \bigcup TP(td_{i_3}) .... \bigcup TP(td_{i_t})$$

3. In order to evaluate the supports s_{ptp} for a pattern aroused preceding tdr_i and supports s_{ntp} for the same pattern aroused following tdr_i , examine the transaction data group.

```
$S_{PTP}$ =ø, $S_{NTP}$ =ø
for all k = 1 to n do
MaxTran ($P_k$) = 0, MinTran (($P_k$) = 0
$S_{FAM}$ ($P_k$) =ø, $S_{FDM}$ ($P_k$) =ø
end for
for all transactions $T_i$ whose position satisfying
$T_\xi$ do
 for k = 1 to n do
 if $T_i \sqsupseteq P_k$ then
            $C_k = C_k + 1$
    if $sup_{+}^{c_k}(P_k) \geq t_s$ and $sup_{-}^{c_k}(P_k) \geq t_s$
then
        if $tran^{c_k}(P_k) \geq t_t$ then
          if $P_k. \notin S_{PTP}$ then
           Add $P_k$ to $S_{PTP}$
          end if
    if $tran^{c_k}(P_k) > MaxTran(p_k)$ then
     $S_{FAM}(P_k) = \{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$
       $MaxTran(P_k) = tran^{c_k}(P_k)$
    else if $tran^{c_k}(P_k) = MaxTran(P_k)$ then
         Add
$\{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$ to $S_{FAM}(P_k)$
    end if
    else if $tran^{c_k}(P_k) \leq -t_t$ then
    if $P_k \notin S_{NTP}$ then
      Add $P_k$ to $S_{NTP}$
        end if
      if $tran^{c_k}(P_k) < MinTran(p_k)$ then
          $S_{FDM}(P_k) = \{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$
          $MinTran(P_k) = tran^{c_k}(P_k)$_
      else if $tran^{c_k}(P_k) = MinTran(P_k)$ then
           Add
    $\{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$ to $S_{FAM}(P_k)$
          end if
       end if
     end if
    end if
   end for
 end for
return $S_{PTP}$ and $S_{FAM}$ , ($P_k$) for each $P_k \in S_{PTP}$
return $S_{NTP}$ and $S_{FDM}$ , ($P_k$) for each $P_k \in S_{NTP}$
```

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%.

5. CONCLUSION

We explored a novel approach for mining the patterns of Positive transitions from a data stream. We have implemented an efficient positive Transition pattern mining model [15] in our earlier research paper, further here we developed an approach for positive Transition patterns from variable window size that defined by context variation analysis (CCA-VW-FPTPM) 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. Frequent Positive Transition Pattern Mining approach [15] is adapted to perform positive transitional pattern 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-FPTPM is scalable under divergent streaming data size and coverage values. In future this model can be extended to perform utility based frequent item-set 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, pp. 207-216, 1993.

[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.

[15] Sujatha Dandu et al; PTP-Mine: "Range Based Mining of Transitional Patterns in Transaction databases"; Volume 12 Issue 2 Version 1.0 January 2012; Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc. (USA)