Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

The discovery of association rules in the rapid flow of data using a sliding chute window

Leila AZADİ SHİRİ[1], Reza NOURMANDİ-POUR[2,*]

[1]*Department of computer engineering, Sirjan Science and research branch, Islamic azad university, Sirjan, Iran & Department of computer engineering, Sirjan branch, Islamic azad university, Sirjan, Iran*

[2]*Department of computer engineering, Sirjan branch, Islamic azad university, Sirjan, Iran*

Received: 01.02.2015; Accepted: 05.05.2015

Abstract. Window method based on the decline all set items are stored alphabetically in a tree. Each node in the tree represents a set of pens. In addition, an entry for each node keeps his collection of relevant data item. Cnt value reflects the current iteration of the model, blurred and tid specifies the number of the last transaction that contains the item has been set. Since the data distribution may change over the data flow, the sliding window method based on Only recently have shifted their attention to the data observed. Some data sets may be part of the data stream, but may be very new data entry and increasing data volume of data in another part of the same set of data with very low or even zero. For Fixing these algorithms have been introduced based on sliding windows that are turning their attention to the part of the data and the data prior to the most recent data not be exposed to fewer repetitions gradually so that the non-items become and the list of data that are evaluated in each window are deleted.

Keywords: Items are repeated, sliding windows, windows deterioration, dependency rules, algorithms Vienna.

1. INTRODUCTION

Explore the laws of the board topics explored in the context of a research and discovery in respect of the interest and importance of relationships between data items in large data warehouses h leg of the transaction is that recent research efforts great it is important to account for these applications, they can[1] be called into question should donkey cart yale analysis, clustering and strap the board noted. The form of the laws of the **x** forums **y** is indicative of the type of prior x and y are the same for these items (x and y are two sets of pen choice of data storage) x and y, respectively between the front and the tal law called the idea of be. Various criteria for evaluation of the authenticity and value of the above laws provided that they can be based on good laws means of fluid accumulates from the list of laws may be selected. The most famous and beloved r yen this criteria 2 standard minimum y,y back at least to ensure a degree. Back y, x is a set of items such as the number of transactions include all items in x the total number of transactions.

The most interesting and useful not only our laws yen for that includes the frequencies from which to learn, not items that are rarely found to be data. For example, what strategy goods store with girls yell out things that rarely have a customer, the strategy will not be successful. Therefore this is often their methods assume that we seek is a collection of my items at least acceptable fraction of transactions occur together, ie the inverse of the other man at the back our not less. The so-called frequent item set for the items back goes high. O ensure a law **x** **y** as well as

[1] Decayed Rate
[2] Closed Enumeration Tree
[3] Infrequent Getway Nodes
[4] Unpromising Getway Nodes
[5] Intermediate Nodes
[6]Closed Nodes

*Corresponding author. *Email address*: *noormandi_r@iausirjan.ac.ir*

Special Issue: The Second National Conference on Applied Research in Science and Technology

the ratio of the number of repeated simultaneous x, y the number of occurrences of x alone is defined as the fraction of the transactions are included on the x or the y's are the towering included. A total of ¥ accepted rules of that quantity acceptable for a man to both the above criteria. In explored the issues of the size of the data that is usually mentioned time is not in main memory . Thus the evaluation of the performance of different methods to explore laws, such as a time standard need to read data from or the number of risks that must be read' data, use of the process.

To improve the algorithm has different The proposed research is to discover Prtkrartvst items for these focused efforts that will be Find items possible number of frequencies that are in need of recourse to the risks my Hajj data reserved seen the direct method Half the items to my surprise. Although the methods proposed are really efficient in this regard, but another important issue is rarely considered insignificant and that these methods in specific cases and in relation to the items on the work and in many cases result The calculation of the A set of items but not counting my repetitions. In this article we try to have efficient methods for the discovery of my association laws offer my ten single count a main building on the optimal number of data items.

1.1 Window-Based Methods Deterioration Algorithm

The algorithm [1] All of the items are stored alphabetically in a tree. Each node in the tree represents a set of pens. In addition, each node has one entry (cnt, id) for the maintenance of its respective item. Cnt value reflects the current iteration of the model, blurred and tid specifies the number of the last transaction that contains the item has been set. Astdk algorithm consists of four main phases which are: to update parameters, repeat update, add and choose font collection of duplicate items. Details of these phases and steps in Figure 1-2 are shown. When a new transaction is generated, all other phases of the fourth phase will be implemented in tandem. Notes that the fourth phase will be executed when new patterns are extracted.

At the beginning of the current transaction data streams in phase updating the following parameters to be updated**.**

$$|D|_k = |D|_k d + 1 \tag{1}$$

Update replication phase of a repeat of the items that have been seen in the new transaction to be updated. At the foot of the tree, all paths are generated by items in new transactions, navigation and the last entry of the node (cnt $_{p,}$ tid $_{p)}$ are updated as follows:

$$cnt_k = cnt_p d^{(k-tid_p)} + 1, \qquad tid_k = k \quad (p < k) \tag{2}$$

When support Date is less than the threshold of a pen, this pen set foot is removed from the tree to prune its operations say. The following figure shows the algorithm in pseudo-code.

After all of the items related to new transactions in the tree, were on stage Add item set to add a new set of items begins. All items are not important at the beginning of the new transaction are deleted, because the font set contains one or more items is not important, can not be recognized as an important item in the collection. For this purpose, all items in new transactions meet the relevant node in the tree are examined. If you have a current backup of the filter is less than the threshold. At the same time there was a new pen if the transaction was not in the tree, the tree will be added. New items are added can be divided into two groups. Those who first appear in the data flow, and those that have already appeared in previous transactions and have been pruned. The first iteration to zero, while for the second Is. Astdk algorithm maximum possible amount for a repeat of a previous transaction is considered to be a new item. In addition, the amount based on the elimination, reduction, and one to which there is added as a new transaction. Therefore, the current entry of a new item in the tree node monitoring is set as follows.

The discovery of association rules in the rapid flow of data using a sliding chute window

$$cnt_k = \left(|D|_{k-1} S_{sig}\right) d + 1, \qquad tid_k = k. \tag{3}$$

Input: A data stream D_k
Output: A complete set of recently frequent itemsets L_k

S_{min} : A minimum support
S_{sig} : A given significant support
d : A given decay rate
ML : A monitoring tree that maintains a set of itemset entries $e(cnt, tid)$'s

```
1   ML = ∅;
2   for each new transaction T_k in D_k {

        // Parameter updating phase
3       |D|_k = |D|_{k-1} × d + 1;
        // Count updating phase
4       for all itemset e ⊆ T_k in ML {
5           ML.e.cnt = ML.e.cnt × d^(k − ML.e.tid) + 1;
6           ML.e.tid = k;
7           if (ML.e.cnt / |D|_k) < S_sig      // Pruning
8               Eliminate e and all of it's descendent nodes from ML;
9       }

        // Itemset insertion phase
10      T̄_k = ∅;
11      for all itemset e ⊆ T_k s.t. |e| = 1 {
12          if e ∉ ML {
13              Insert e into ML;
14              ML.e.cnt = (|D|_{k-1} × S_sig) * d + 1; ML.e.tid = k;
15          }
16          else if S_k(e) ≥ S_sig      { T̄_k = T̄_k ∪ e ; }
17      }
18      for all itemset e in ML for all new itemset ē ⊆ T̄_k s.t. e ⊂ ē , |ē| = |e| + 1 {
19          Estimate C_k^max(ē) ;
20          if C_k^max(ē) > C_k^upper(ē)
21              C_k^max(ē) = C_k^upper(ē) ;
22          if (C_k^max(ē) / |D|_k) ≥ S_sig {
23              Insert ē into ML;
24              ML.ē.cnt = C_k^max(ē) ; ML.ē.tid = k;
25          }
26      }

        // Frequent itemset selection phase
27      L_k = ∅;
28      for all itemset e ∈ ML {
29          ML.e.cnt = ML.e.cnt × d^(k − ML.e.tid) ;
30          ML.e.tid = k;
31          if (ML.e.cnt / |D|_k) ≥ S_min
32              L_k = L_k ∪ {e};
33      }
34  }
```

Figure 1. Pseudo-code algorithms [9]

Once all items have been removed by ineffective transactions, a transaction filter is obtained. To add multiple font collection, monitoring tree traversal once again, but this time the transaction is blocked and its nodes are visited. When a node of a pen e met, the maximum possible repeat of the font set that is filtered e transaction and the remaining items filtered from the transaction is estimated. This estimate is derived using a subset. As for a brush with length n, n-1 subsets length are much closer to the formula used for the estimate.

$$C_k^{max}(\bar{e}) = \min\{C_k(\alpha) | \alpha \subseteq \bar{e}, |\alpha| = n - 1 \tag{4}$$

The minimum length of one less repeat all the subsets that are present in the tree. If the support is estimated at more than the threshold set by the filter is important, it is written as a series of important new item is added to the tree. Added to this method is called with a delay, because the addition of a new item, you must repeat a subset of important and each of them is more important than the threshold.

When a new item e is diagnosed, it can be as high as repeat Achieved. Consider that among k transaction that has been produced, at least by e n n n a transaction that includes all the subsets that are needed and also the tree of items to be added. However, although the number of transactions that are included in this set greater or equal to n, pens, pen sets e may be added to the tree is estimated to be less than the threshold of support is important. According to this view, the repetition of the item if the following conditions n n e Figure 2 , The maximum amount that will be:

1. One of the n Transaction is the first transaction data flow.

2. n-1 remaining transactions, transactions have recently been produced.

3. In the first transaction n From the moment a transaction earlier this year to support the collection item should be the maximum, but less than the threshold.

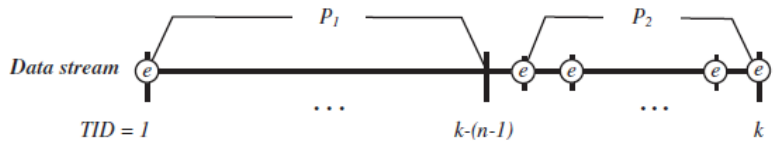

Figure 2. Repeated at the top of a new item e

Due to the high flow of data and is divided into two periods. Since the series in the first period between the transaction is 1 and pens e k- (n-1) is not repeated. The upper limit of this period, **)** Is obtained. After the second course in the series by n-1 transaction consecutive appearance, content***)*** that, n-1 times based on reduced blur. As a result of the high repetition in the entire data flow is calculated as follows.

$$C_k^{upper}(e) = S_{sig}|D|_{k-(n-1)}d^{(n-1)} + \{1 - d^{(n-1)}\}/(1-d). \quad (5)$$

According to this view, and what about the repeated estimates using a subset of the above, if the maximum number of possible **)** A new item e of the high repetition ***)*** The higher this value as the value of cnt repeated in the relevant entry in the tree is used to monitor the amount of more accurate estimates to repeat it. So the following entries for the new node is initialized.

$$cnt_k = \min\{C_k^{max}(e), C_k^{upper}(e)\}, \quad tid_k = k. \quad (6)$$

Alphabetical tree for features, all of the new items to be diagnosed organized and are added to the tree. In addition, the collection of items that are removed from the tree, if in the future new production transactions are seen repeatedly, with the addition of delay are added to the tree alphabetically.

1.2. Algorithm +

Lee and Lee [8] + EstDec algorithm to find the set relative to a set of data items have introduced Prtkrar maximal. This algorithm uses a reduction factor 2 are used. In this way, instead of simple prefix tree structure of a trie authors called CP-tree uses compressed. The tree structure is as follows.

D Consider a trie. D integration with a threshold that is between zero and one, if all data items stored in a formula to meet the D S under the tree, then S to a CP-tree node will be compressed.

$$\frac{freq(X)-freq(Y)}{N} \leq \delta \qquad (7)$$

Msal2-2 Figure 4 is a trie structure and an associated CP_tree shows. An S in the trie under the tree in a single node is called the compression CP_tree point shown by the arrow. The list of <b, c> b and c-nodes with labels in S are connected. Father of the list <> b means that the father node in S is currently in the first position of the list of elements in the tree is compressed and the father node c in S is currently in first position in the list of compressed tree is.

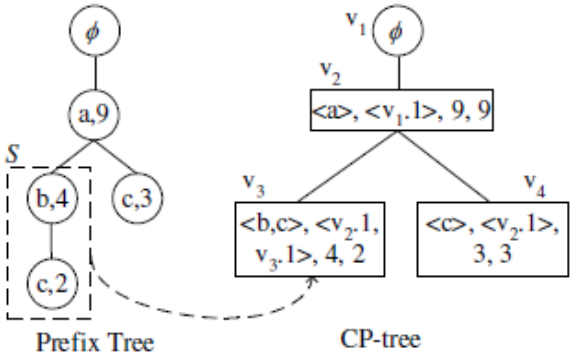

Figure 3. Trie compression algorithm Estdec +

To obtain maximal set of patterns, CP-tree navigation is to extract maximal patterns. The algorithm selected data sets and identify their ideas of [7] is taken.

The algorithm using those old factors, the effects of transactions older low and more recent data is more important. One problem with this method is that it can not be easily considered a range of error for the number of repetitions of the extracted patterns and many will have to answer False-positive. This is because the number of repetitions calculated for a data set may be much greater than the actual number of repetitions. On the other hand, however, this method uses a tree structure, but intensive compression operations, in turn, have a relatively high complexity that it would be much more complex with the increasing volume of input data.

2. METHODS BASED ON SLIDING WINDOW

In this section we algorithm based on sliding window [2,3,4,1,6,8] we studied. Since the data distribution may change in the flow of data, methods based on sliding window only focus their attention to the data that has to be observed. Some data sets may be part of the data stream, but may be very Prtkrar new data entry and increasing data volume of data in another part of the same set of data with very low or even zero. For example, consider a store cart. A particular item, such as ice cream can be sold in a particular season and warm season very much and have a lot of repetition in daily sales in a supermarket, but other seasons and winter to reach gradually the

number of repetitions to a decrease in Repeat or repeat low commodity becomes zero. To solve these problems, algorithms have been introduced based on sliding windows that are focused his attention on the part of the data prior to the most recent data are not Prtkrar data by reducing the number of repeat exposures are gradually making up data items become non Prtkrar and the list of data that are evaluated in each window are deleted. Since the algorithm based on sliding window select models only through | W | Recent data selects where | W | is the length of the window. With the arrival of new transactions under review with the model we presented several algorithms.

2.1. Algorithm Vienna

Chang and lee [2,10] called Win algorithm to protect data collection have introduced using a sliding window. Data sets generated by Win in a trie D structure is maintained. X is a data set D has the following fields As freq (x) indicates the frequency of X current location is listed to the current time. Denotes a high frequency X in the current window before it is included in D. The transaction ID is extracted X and the D has been inserted. With the arrival of each incoming transaction ID Y *t Where* t is the current time, the algorithm calculates Win number of repetitions for each subset of Y in D can be calculated. In this algorithm a data set X if at least one of the two following conditions is established as a non- data will be considered and should be removed.

1. Tid (X)> tid $_{1,}$ freq (x) $<\hat{I}\lceil\in(N-(tid(x)-tid_1)\rceil$
2. freq (x) $<\hat{I}\lceil\in N\rceil$

After updating, and deleting data sets ineffective, the algorithm Win new data set on the D Inserts. The algorithm first set of data to insert a data item that is in terms of their frequency of occurrence of 1 and 0 errors. Then, for each new data set X 2 X ⊂ Y is greater than or equal to the length of time since their entry into the D before D X to Y, then the algorithm adds. In this case, the number of occurrences of X equal to 1 tid *tid* $_{t.}$ The error is given by the formula x.

$$err(X) = min\Big(min(\{(\widetilde{freq}(X') + err(X')) \mid \forall X' \subset X \text{ and } |X'| = |X| - 1\}) - 1, \lfloor\epsilon(w - |X|)\rfloor + |X| - 1\Big) \quad (8)$$

Finally estWin algorithm for each data set X, X provided as output data sets algorithm and selects one of the following conditions occurred.

1) $Freq(x) \geq \lambda N \quad , \quad tid(X) \leq tid1$
2) $freq(x) + err(x)) \geq \lambda N \quad , tid(x) > tid1$

Sliding window model based on the data model changes the way Win recently been detected in the study. However, the update window for each new transaction that will delete the oldest transaction type should do so with lower performance than batch processing methods such as lossy counting. This complexity, especially when The search space is large with a small error at its most shows. But what is more remarkable is that the current batch of high-speed input data to quickly generate billions of transactions are not accountable.

2.2. Moment Algorithm

Chi and his colleagues [4, 5] to update the list of data collection and close have introduced an algorithm called Moment. The algorithm is based on a sliding window and a trie-based structure

could be stored in memory have introduced 3 called CET. The dynamic structure for a selected set of data sets on the sliding window is selected.

Let *n Node represents a data set* X is considered CET. A list of data sets corresponding tree nodes are classified in four different types.

- IGN 4 : *n One* IGN Is if 1) X r not. 2) *n Father node n And* y is a data set. 3) If *n A twin and* **x = Y È Y** ' Then Y '.
- 5 Ugn: *n One* UGN Is if 1) X and 2) some data collection Y There is so Y A data collection package Prtkrar And **X ⊂ Y** And **freq (Y) = freq (X**) And Y In alphabetical sorting of data collection before X Is.
- 6 IN: *n One* IN If you are
- 1) X A data set is
- 2) *n* Father *n Such that* freq (Y) = freq (X) And
- 3) *n One* UGN Not.
- 7 CN: *n One* CN If you are X a data set is closed.

Figure 2-1 shows an example of the structure of the CET.

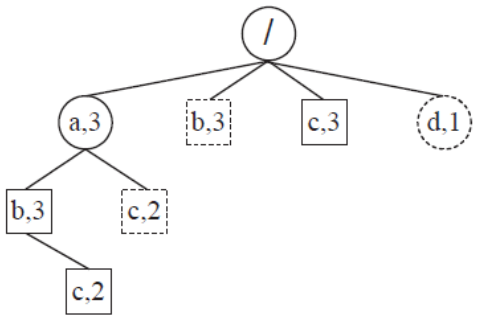

Figure 2-1. An example of structure CET

The picture above is an example of CET structure made from a series of transactions {cd, ab, abc, abc} shows where l = 0.5 and the number of transactions in the sliding window is equal to 4. CET structure is similar to the structure of trie except that there are four types of nodes in the structure. Nodes indicated by the dotted circles and squares dotted UGN IGN, a circle with lines and squares continuous-line steadily IN CN Are.

Moment advantage of the algorithm is to set clear and close patterns using the drop-down window. But there is a problem with this method is that with the arrival of a new transaction or transactions shall expire oldest many nodes are the nodes, change the time and reduce the effectiveness of the algorithm. [9,10].

3. CONCLUSION

Due to the increasing use of information banks and repositories my pilgrimage big transaction recent attention of many researchers toward efficient production methods to extract the laws of the forum was focused on the first phase of work. Most methods a repeat of its full items (simple and composite out) between all of the items in the data looks that this is repeated in need to read the data from the risks new methods which have been proposed to try on this event and the degree of the getting at least some of the items can be calculated directly Yama no data c., and less attention to this issue and how to approach optimal.

REFERENCES

[1] JH Chang and WS Lee. Finding Recent Frequent Itemsets Adaptively over Online Data Streams. In Proc. of KDD, 2003.
[2] JH Chang and WS Lee. estWin: Adaptively Monitoring the Recent Change of Frequent Itemsets over Online Data Streams. In Proc. of CIKM, 2003.
[3] JH Chang and WS Lee. A Sliding Window method for Finding Recently Frequent Itemsets over Online Data Streams. In Journal of Information Science and Engineering, Vol. 20, No. 4, July, 2004.
[4] Y. Chi, H. Wang, PS Yu and RR Muntz. Moment: Maintaining Closed Frequent Itemsets over a Stream Sliding Window. In Proc. of ICDM, 2004.
[5] Y. Chi, H. Wang, PS Yu and RR Muntz. Catch the Moment: Maintaining Closed Frequent Itemsets over a Data Stream Sliding Window. In KAIS, 10 (3): 265-294, 2006.
[6] C. Giannella, J. Han, J. Pei, X. Yan, and PS Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In Kargupta et al .: Data Mining: Next Generation Challenges and Future Directions, MIT / AAAI Press, 2004.
[7] K. Gouda and M. Zaki. Efficiently Mining Maximal Frequent Itemsets. In Proc. of ICDM, 2001
[8] D. Lee and W. Lee. Finding Maximal Frequent Itemsets over Online Data Streams Adaptively. In Proc. of ICDM, 2005.
[9] H. Li, S. Lee, and M. Shan. An Efficient Algorithm for Mining Frequent Itemsets over the Entire History of Data Streams. In Proc. of First International Workshop on Knowledge Discovery in Data Streams, 2004. 39
[10] D. Xin, J. Han, X. Yan, and H. Cheng. Mining Compressed Frequent-Pattern Sets. In Proc. of VLDB, 20 05.