Article can be accessed online at http://www.publishingindia.com

A Novel Algorithm for Predicting Valuable Items in Data Streams

S. Vijayarani Mohan*

Abstract

A data stream is a real time, continuous, structured sequence of data items. Mining data stream is the process of extracting knowledge from continuous arrival of rapid data records. Data can arrive fast and in continuous manner. It is very difficult to perform mining process. Normally, stream mining algorithms are designed to scan the database only once, and it is a complicated task to extract the knowledge from the database by a single scan. Data streams are a computational challenge to data mining problems because of the additional algorithmic constraints created by the large volume of data. Popular data mining techniques namely clustering, classification, and frequent pattern mining are applied to data streams for extracting the knowledge. This research work mainly concentrates on how to predict the valuable items which are found in a transactional data of a data stream. In the literature, most of the researchers have discussed about how the frequent items are mined from the data streams. This research work helps to predict the valuable items in a transactional data. Frequent item mining is defined as finding the items which occur frequently, i.e. the occurrence of items above the given threshold is considered as frequent items. Valuable item mining is nothing but finding the costliest or most valuable items of a database. Predicting this information helps businesses to know about the sales details about the valuable items which guide to make crucial decisions, such as catalogue drawing, cross promotion, end user shopping, and performance scrutiny. In this research work, a new algorithm namely VIM (Valuable Item Mining) is proposed for finding the valuable items in data streams. The performance of this algorithm is analysed by using the factors, number of valuable items discovered, and execution time.

Keywords: Data Streams, Frequent items, Valuable Items, VIM algorithm

Introduction

A data stream is a real time, ordered sequence of instances. In many applications, data stream mining can read the data base only once. Some of the application areas of data streams are computer network traffic, mobile phone conversations, ATM transactions, network searches, and sensor records(Agarwal, 2007; Agarwal, Imielinksi, & Swami, 2000). Data stream mining can be considered as a subfield of data mining and machine learning. The main goal of mining data stream is to forecast the class or value of new instances in the data stream which gives some knowledge about the class membership or values of previous instances in the data stream. Machine learning techniques are used to learn this prediction task from labeled examples in an automated fashion (Bifet, 2011).

Stream data can be a continuous, potentially infinite flood of information as opposed to finite, statically stored datasets. Besides querying data streams, another important application is to mine data streams for interesting patterns or anomalies as they take place. In data streams, the volume of data is usually too large and it is not possible to store the entire data on permanent storage devices and also it is very difficult to scan the data more than once.

* Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, India.
Email: vijimohan_2000@yahoo.com

With the help of the data stream generator the user gets information and to apply any of the data mining algorithms the user can get the required output. In data streams, data analysis requires single pass algorithm because the size of the data is huge and also the single pass algorithm saves the processing time as well as the memory space.

Data streams have several unique properties: infinite measurement lengthwise, concept-drift, concept-evolution, feature-evolution and restricted labeled data. Concept-drift occurs in data streams when the underlying concept of data changes over time (Last, 2002; Wang, Fan, Yu, & Han, 2003). When new classes evolve in data stream data the concept-evolution occurs. Feature-evolution occurs when feature set varies with time in data streams. Data streams also suffer from insufficiency of labeled data since it is not possible to manually label all the data points in the stream. All these properties make a challenge to mine stream data.

This valuable item mining research helps to find the most valuable items in a transactional database. This can be achieved by providing the cost of an individual item and assigning an individual threshold for each and every item in a transaction. This gives the information about the sales details of every item at a particular time period. This information also provides whether the business may achieve its profit or not. With this valuable item mining analysis, the business organisations can improve their business strategy.

This paper will focus on the following sections. Second section discusses the related works. Third section gives the objective of the problem. The proposed VIM algorithm is described in fourth section. Experimental results are analysed in fifth section. Conclusion and future work are given in sixth section.

Related Works

(Tabeer, Ahmed, Jeong, and Lee (2008) and Han, Pei, and Yin (2001)had proposed a prefix-tree structure called CPStree (Compact Pattern Stream tree). The CPS tree uses a new technique called as dynamic tree restructuring technique to handle the stream data. This tree constructs a compact-prefix tree structure with single pass scanning. Its performance is same as FP tree growth technique. After creating the CPS tree, they refresh the tree at each window. For restructuring the CPS tree they used an efficient restructuring mechanism called as BSM method and path adjusting method. The algorithm uses bottom up technique to generate exact set of recent frequent patterns.

Paurayand Tsai (2009) had proposed a new technique called the weighted sliding window WSW algorithm. This model allows the user to specify the number of windows for mining, the size of the window and the weight of each window. Using this algorithm the user can specify minimum weighted threshold value. They split the transaction into equal number of windows. Using the WSW algorithm, in every window the weight of each transaction is calculated. If the weighted support count of an item is greater than or equal to minimum weighted threshold value it is called as frequent item set. Using the Apriori algorithm the user can generate the candidate item set also. When a candidate item set is generated then they determine whether it is frequent or not by using the WSW algorithm.

Li and Li (2009) proposed an efficient bit-sequence based algorithm called MFI-Trans SW (Mining Frequent Item sets with in a Transaction Sensitive Sliding window). MFI algorithm worked on three phases. They are window initialisation, window sliding, and pattern generation. In window initialisation phase, the items in the transaction are encoded in an efficient bit sequence representation. The second phase of the algorithm uses the left bit shift sequence technique to slide the windows efficiently. In final phase, the complete set of frequent item sets within the current sliding window is generated. Based on the MFI-TransSW they proposed another algorithm called MFI-TimeSW to find the set of frequent item sets over time sensitive sliding window.

Objective of the Problem

The main objective of this research work is to predict the valuable items in data streams. A new algorithm VIM is proposed for finding the valuable items in data streams. The efficiency of this algorithm is verified by the two performance factors namely number of valuable items found in each window and the execution time.Fig. 1 shows the system architecture of this research work.

VIM Algorithm

The acronym VIM stands for Valuable Item Mining and this algorithm is designed to find the valuable items in

Fig.1: System Architecture

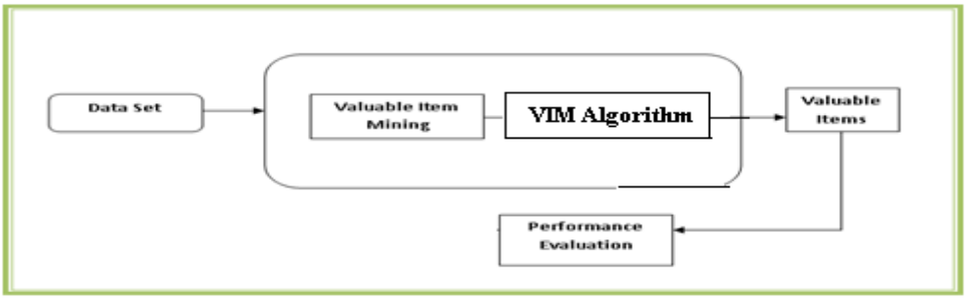


Table 1: VIM Algorithm

Input: data stream $\{W_1, W_2.....W_n\}$ Output: List of valuable items. 1. For each window $W_i = \{T_1, T_{2....}T_k\}$ where $T_1, T_2....T_k$ are transactions
2. For each transaction $T_j = \{I_1, I_2....I_l\}$ where $I_1, I_2....I_l$ are items
3. Assign cost to each item$\{C(I_1), C(I_2)....C(I_l)\}$
4. Assign individual threshold to each item $\{\sigma_1(I_1), \sigma_2(I_2)... \sigma_l(I_l)\}$ //Cost Calculation
5. For m=1 to l
6. {
7. Total_cost (I_m) =number of occurrences (I_m)* cost (I_m)
8.}
9. K=1 // Finding the Valuable Item
10. For m=1 to l
11. Total_cost $(I_m) >= \sigma(I_m)$
12. {
13. Val_item(K) = I_m
14. K = K+1

15.}
//candidate generation//
1.Combine the L^{th}-item set to form the L+1 item-set (Initially L=1)
2.Calculate the occurrences and total cost
3.Calculate the average threshold for L+1 item set
4.Check if total cost >=avg threshold then
5. {
6.Consider the L+1 item set as valuable item set
7. Else
8.Ignore the L+1 item set }
9. Consider the L+1 item set. Now, L=L+1
10. Repeat the steps from 1 to 9 until the termination has occurred.

data streams. This algorithm requires single pass to find out the valuable items. In this algorithm the data owner should assign cost for each and every item in a transactional database and also the data owner must provide individual threshold for each item. Then, this algorithm calculates the total cost of each item based on their number of occurrences. Then the total cost is compared with their corresponding threshold. If the total cost is greater than the threshold value it is considered to be a valuable item. After finding the candidate-1 items the user finds the candidate-2 itemset by using the same cost and same threshold value. The user can add the cost of two items which are bought together. Here the averages of two items thresholds are compared with the total cost. If the total cost is greater than the average threshold value it is considered to be a valuable item. All other candidate items are generated using the same concept. The pseudo code of this algorithm is given in Table 1.

Table 2 shows the sample transactional data with three windows. Window 1 has three transactions. Window 2 has five transactions and Window 3 has two transactions.

Table 2: Transactional Data

W1		*W2*		*W3*	
TID	*Item Set*	*TID*	*Item Set*	*TID*	*Item Set*
1	{1,3,6}	4	{1,3,5}	9	{2,9}
2	{2,3,7}	5	{2,7}	10	{3,4,5}
3	{3,6}	6	{3,5,6}	-	-
-	-	7	{1,7}	-	-
-	-	8	{1,3,5}	-	-

The items which are present in window 1 are 1,2,3,6 and 7. Data owner assigns the cost and threshold for these items. Based on the item cost and its occurrences the total cost is calculated. This is given in Table 3.

Table 3: Total Cost Calculation Using Item Cost

Items	*Cost*	*Occurrences*	*Total cost*	*Threshold*
1	25000	1	25000	50000
2	100	1	100	200
3	20	3	60	100
6	300	2	600	500
7	100	1	100	500

From the above calculations, we come to know that the item 6 is considered as valuable item because their total cost is greater than their threshold value. Using this valuable item the user can generate the candidate-2 itemsets but here 6 is the only valuable item hence this process is terminated at this stage. The same process is repeated for windows 2 and 3 also. The items which are present in window 2 are 1,2,3,5,6,7 and the data owner assigns the cost and threshold values.

Table 4 gives the candidate-1 item set for window 2. The items 1 and 5 are predicted as valuable items, because their total cost is greater than their corresponding threshold values. Next 2 candidate items are generated.

Table 4: Candidate-1 Item for Window2

Items	*Cost*	*Occurrences*	*Total cost*	*Threshold*
1	25000	3	75000	50000
2	100	1	100	200
3	20	3	60	100
5	25000	3	75000	50000
6	300	1	300	500
7	100	2	200	300

Table 5: Candidate-2 Items for Window 2

Items	*Cost*	*Occurrences*	*Total cost*	*Threshold*
1,5	25000+25000=50000	2	100000	50000+50000/2=50000

Table 6: Candidate-1 Items for Window3

Items	*Cost*	*Occurrences*	*Total cost*	*Threshold*
2	100	1	100	200
3	20	1	20	100
4	50000	1	50000	30000
5	25000	1	25000	50000
9	30000	1	30000	25000

From Table 5, the items (1,5) together called as valuable items because their total cost is greater than their threshold value. The same process will be continued for all other candidate generation. After the candidate-2 generation there is no items purchased together. So the process is terminated at this stage.

Table 6 gives two valuable items namely 4 and 9. Using 4 and 9 the user can generate candidate-2 items but those two items are not purchased together so the process is terminated at this phase.

From the above example, we come to know that the final sets of valuable items in a whole data set are {6}, {4}, {9}, {1}, {5} and {1, 5}. This is given in Table 7.

Table 7: Valuable Items

Windows	*Valuable Items*
Window 1	{6}
Window 2	{1},{5},{1,5}
Window 3	{4}{9}

Performance Evaluation

Experimental results of VIM algorithm is discussed in this section. Implementation is done using Microsoft visual studio 2008 with SQL server 2000. For finding the valuable items over transactional data, synthetic data stream data sets are used. The synthetic data used in this work is Kosakshi from IBM data generator. This data set contains 88054 transactions and 46 attributes. The dataset has been retrieved from the linkhttp://fimi.ua.ac.be/data/retail.dat.There are total five windows used, W1, W2, W3, W4, and W5. Three different sizes of transactions namely 100, 500, 1000 are tested and their results are obtained. The average length of the transaction is 22 when the window size is 100. The window contains 500 transactions; its average length is 20. The average length is 10 for1000 transactions.

Number of Valuable Items

In this research work window sliding concept is used. Therefore W1 contains three types of transactions. After finding the valuable items in W1 the next window W2 automatically slides (Ahmed &Jeong, 2011; Chelche & Sadreddini, 2010). The other three windows are processed, using the same sliding window concept.

Table 8: Number of Valuable Items

Windows	*Number of transactions*	*VIM algorithm*
W1	100	13
	500	50
	1000	99
W2	100	12
	500	49
	1000	97
W3	100	13
	500	48
	1000	102
W4	100	12
	500	51
	1000	105
W5	100	11
	500	48
	1000	100

Table 8 illustrates the number valuable items identified by VIM algorithm. Window 4 finds out more number of valuable items in all the transactions i.e. 100, 500, and 1000 compared with other four windows. At the same time, window 2 finds out less number of valuable items compared with other four windows.

Fig. 2 shows the number of valuable items predicted by VIM algorithm in different windows.

Table 10: Execution time

Windows	*Number of transactions*	*VIM algorithm (in sec)*
W1	100	6.84
	500	12.54
	1000	19.56
W2	100	5.92
	500	10.82
	1000	18.97
W3	100	5.52
	500	12.19
	1000	19.98
W4	100	6.11
	500	11.87
	1000	20.11
W5	100	5.12
	500	13.01
	1000	19.72

Fig. 2: Number of Valuable Items

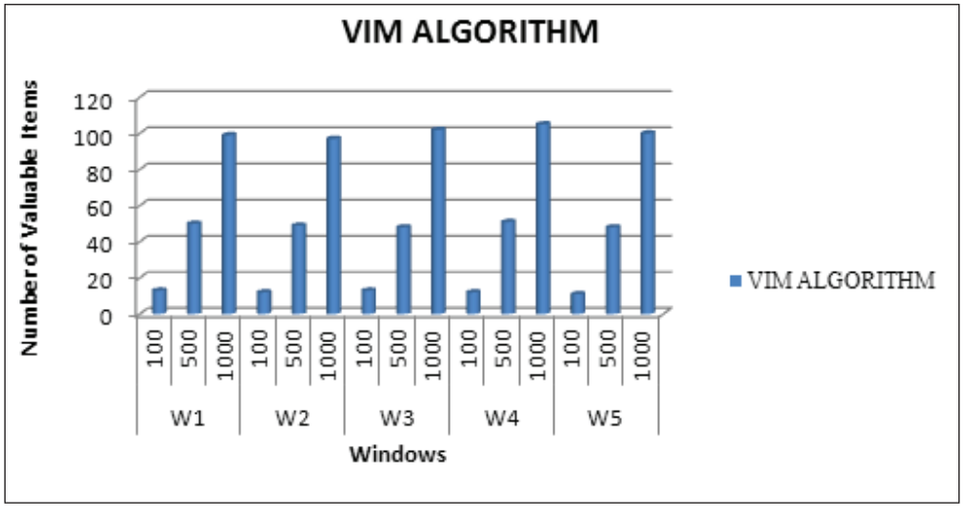
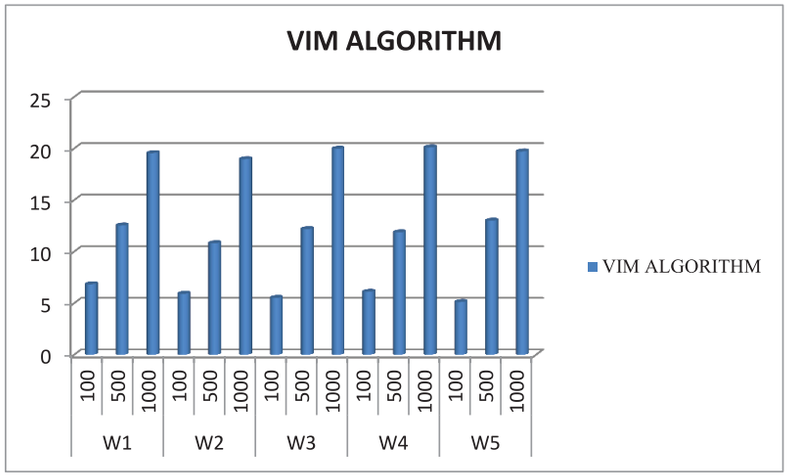


Fig. 3: Execution time



Execution Time

Table 10 shows the execution time required for VIM algorithm. Execution time is nothing but how much time required for predicting the valuable items in each window by the VIM algorithm.Fig. 3 gives the execution time for VIM algorithm.

Conclusion

Mining data stream is the process of extracting unseen knowledge from continuous, rapid data records. Data arrives faster; hence it is a difficult task to mine that data. In this research work, valuable items are mined from the stream data. A new algorithm VIM is proposed to predict the valuable items in data streams. Many new algorithms and techniques are to be required to perform the detailed analysis on data streams. In future new algorithms are to be developed to reduce the execution time.

References

Agarwal, C. (2007). *Data streams: Models and algorithms*. Springer.

Agarwal,R, Imielinksi, T., &Swami, A. (2000).*Fast algorithms for mining association rules between sets of items in large databases*. Research Report RJ9839, IBM Almaden Research Center, San Jose, California.

Ahmed, C. F., &Jeong, B. S. (2011). *Efficient mining of high utility patterns over data streams with a sliding window model*, Springerlink.com.

Bifet, A. (2011). Data Stream Mining: A Practical Approach.

Chelche, E. A., &Sadreddini, M. H. (2010). Using candidate hashing and transaction trimming in distributed frequent item set mining. *World Applied Science Journal*, 9(12), 1353-1358.

Han, J., Pei, H., & Yin, Y. (2001).*Mining frequent patterns without candidate generation*.In Proceedings of Conference on the Management of Data.

Last, M. (2002). Online classification of non-stationary data streams. *Intelligent Data Analysis*, 6(2), 129-147.

Li, H. F., & Li, S. (2009). *Mining frequent item sets over data streams using efficient window sliding technique*, Elsevier Publication.

Liu, B., Hsu, W., & Ma, Y, (1999).*Mining association rules with multiple minimum supports*. Proceedings of Knowledge Discovery and Data Mining Conference, (pp. 337-341).

Pauray, S., &Tsai, M. (2009). *Mining frequent item sets in data streams using the weighted sliding window model*, Elsevier Publication.

Tabeer, S. K., Ahmed, C. F., Jeong, B. S., & Lee, Y. K. (2008). *Efficient frequent pattern mining over data streams*. Elsevier Publication.

Wang, H., Fan, W., Yu, P., &Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD).

Salam, A., & Kulsun, U. (2002). Savings behavior in India: An empirical

Study. *The Indian Economic Journal*, July-September, 50(1), 77-80.