Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 9, Issue, 02, pp.46068-46073, February, 2017

INTERNATIONAL JOURNAL
OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

(ARBSI): PROPOSED ALGORITHM ASSOCIATION RULES BASED ON SCANNING ITEM SETS

*Hebah H.O. Nasereddin

Factually of Information Technology, Middle East University (MEU), Jordan

ARTICLE INFO

Article History:
Received 04th November, 2016
Received in revised form
06th December, 2016
Accepted 18th January, 2017
Published online 28th February, 2017

Key words:
Data mining, Insert, Update,
Delete, Itemsumation.

ABSTRACT

The paper; discuss the concept of estimating and building the model process using association rule model, scanning item sets with their counts and design a novel, efficient, dynamic mining algorithm. (ARBSI) will not require rescanning the original database after collecting the data, even if a number of transactions have been newly inserted, and this will work regardless of the support value used and regardless of the confidence value used. (ARBSI) can work in conventional form, this is more efficient and will reduce the time when its performance is compared with the previous techniques used, in such away as: It will know the number of items used from the last process after normalization sub-process which will reduce the time for scanning each transaction, It will know the types of modification insert, update, and/or delete, In case there is an new inserted record (ARBSI) can translate this record to numeric using dummy table for attribute without duplicate (especially for nominal values)

Citation: Hebah H.O. Nasereddin, 2017. "(ARBSI): Proposed algorithm association rules based on scanning item sets", *International Journal of Current Research*, 9, (02), 46068-46073.

INTRODUCTION

Data mining is the task of discovering interesting and hidden patterns from large amounts of data where the data can be stored in databases, data warehouses, OLAP (on line analytical process) or other repository information (Maria Halkidi, 2000). It is also defined as knowledge discovery in databases (KDD) (Fayyad *et al.,* 1996; Jiawei Han *et al.,* 2001). Data mining involves an integration of techniques from multiple disciplines such as database technology, statistics, machine learning, neural networks, information retrieval, etc. Data mining process is a step in Knowledge Discovery Process consisting of methods that produce useful patterns or models from the data (Jiawei Han *et al.,* 2001). In some cases when the problem is known, correct data is available as well, and there is an attempts to find the models or tools which will be used, some problems might occur because of duplicate, missing, incorrect, outliers values and sometimes a need to make some statistical methods might arise as well. The KDD procedures are explained bellow (Hebah Nasereddin, 2011), in a way to help us focus on data mining process. It includes five processes:

- Defining the data mining problem,
- Collecting the data mining data,
- Detecting and correcting the data,
- Estimating and building the model,
- Model description, and validaion as seen in Figure 1

Corresponding author: Hebah H.O. Nasereddin,
Factually of Information Technology, Middle East University (MEU), Jordan.

Estimating and Building the Model (Hebah Nasereddin, 2012): This process includes four parts: 1) select data mining task, 2) select data mining method, 3) select suitable algorithm 4) extract knowledge as can be seen in Figure 2. Many Data mining techniques have been developed over the last 30 years. Depending on the type of databases processed, these mining approaches may be classified as working on transaction databases, relational databases, and multimedia databases, among others. On the other hand, depending on the classes of knowledge consequent, the mining approaches may be classified as finding association rules, classification rules, and clustering rules (Mehmed Kantardzic, 2003), among others. From past research, it is clear that association rules in transaction databases are the most common in data mining (Park *et al.,* 1997). This paper is closely related more specifically, to Association Rules. Thepaper is divided into five sections. Section 2 describe Data Mining Process Using Association Rules, section 3 discusses, Estimating and Building the Model Process Using Association Rules. Section 4 presents Definition of the Proposed algorithm (ARBSI).whileSection5 presents conclusion

Data mining process using association rules

In previous research, mining association rules algorithms form transactions were proposed, most of which were executed by scanning single items first, then scanning with two items, and this was repeated, continuously adding one more item each time, until some criteria were met. These algorithms are designed to work with static database.

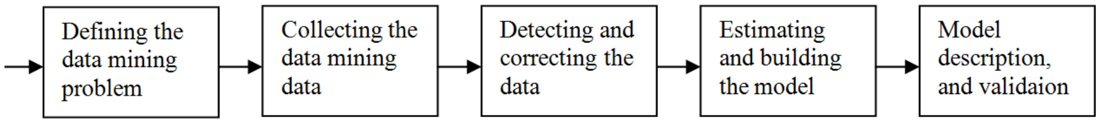

Figure 1. Data mining process (Hebah H. O. Nasereddin, 2011)

Figure 2. Estimating and building the model (Hebah H. O. Nasereddin, 2012)

However In real-world applications, new transactions are usually inserted into databases, and designing a mining algorithm that can maintain association rules as a database grows is thus critically important. One application of data mining is to induce association rules from transaction data, such that the presence of certain items in a transaction will imply the presence of certain other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data (Agrawal *et al.,* 1993; Agrawa and Srikant, 1994; Agrawal *et al.,* 1997). They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the count of an itemset appearing in the transactions was larger than a pre-defined threshold value (minimum support), the itemset was considered as a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items (Hebah Nasereddin, 2008). This process was repeated until all the large itemsets have been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (minimum confidence) were given out as association rules.

Estimating and building the model process using association rules

The original association rules may become invalid, when new transactions are added to databases, or new valid rules may appear in the resulting updated databases (Cheung *et al.,* 1996; Cheung *et al.,* 1997; Lin and Lee, 1998; Zhang, 1999). In these cases, mining algorithms must re-process the entire updated databases to find final association rules. This will cause two problems: Algorithms do not, however, use previously mined information and require rescanning the database which cost nearly twice the computational time to mine the databases. If new transactions appear often and the original databases are large, these algorithms are thus inefficient in maintaining association rules (Hebah Nasereddin, 2012). Transactions databases grow over time in real-world applications, which means re-evaluated association rules mined because new association rules may be generated and old association rules may become invalid when the new entire databases are considered. Apriori (Agrawal *et al.,* 1993) and DHP (Park *et al.,* 1997) solved this problem by re-processing entire new databases when new transactions are inserted into the original databases. These algorithms have two disadvantages: First, increasing the computation time for each insert / update and/or delete transaction. If the original database is large, much computation time is wasted in maintaining association rules whenever update transactions are generated. Second, information previously mined became meaningless (Anju Kakkad and Anita Zala, 2013). The importance of dynamic estimating and building process becomes essential due to the time consumption problem. Many researchers tried to solve these problems. Such as The Fast Update Algorithm (FUP) (Cheung *et al.,* 1996), Pre-large itemsetes (Tzung-Pei Hong *et al.,* 2001) and Record Deletion Based on the Pre-Large (Tzung-Pei Hong and Tzu-Jung Huang, 2007) they provided solution for the insert operation but failed to do the same for the other two cases namely update and delete.

Association rules based on scanning the itemsets (ARBSI)

Although the FUP algorithm (Tzung-Pei Hong and Tzu-Jung Huang, 2007) and Pre-large Itemsets algorithm () focused on the newly inserted transactions and thus save much processing time by incrementally maintaining rules, both of them must still scan the original database to handle cases of newly inserted transactions, both of them solve the insertion case but ignore the update and delete cases. Another disadvantage is if the number of newly inserted transactions (Tzung-Pei Hong *et al.,* 200) is less than the safety threshold, no action is done in this case, this situation may occur frequently, especially when the number of new transactions is small. In additional; to the problem of being not flexible, for example when the support value changes that means both techniques will be meaningless. Any way their techniques start after static association rule mining, after scanning and finding the large itemsets and it is dependent on the support value from the beginning. (ARBSI) presents solutions to the disadvantages of the above techniques. It deals with:

- The new transactions (insert/ update/delete).
- The support value is flexible it depends on the user as he/she chooses this value before and/or during running the data mining process.
- It only scans the original database once to find all itemsets with their appropriate counts.

Also (ARBSI) can work either in this dynamic process from scratch, which is more efficient than previous techniques such as: it knows the number of itemsetes from the last process after normalization sub-process which will reduce the time for scanning each transaction, it knows the types of modification insert, update, and/or delete, (ARBSI) after generates a mathematical summation value for each transaction (Hebah Nasereddin, 2012). If a new transaction is to take place, a new summation value will be generated based on the new status, which will also be reflected in a dedicated file stored in a predefined local database, which will be used to compare with itemsets selected in the initial scan.

Definition of the proposed algorithm (ARBSI)

The proposed algorithm is to induce association rules from transaction data, such that the presence of certain items in a transaction will imply the presence of certain other items by dividing the mining process into two phases. In the first phase, all itemsets will be generated and counted by scanning of the original database without any consideration to the threshold value (minimum support) as in (Agrawal *et al.,* 1993; Agrawal and Srikant, 1994; Agrawal *et al.,* 1997). Number of all itemsets will be equal ($2^{\#items} - 1$). Number of items will be easy to calculate when we run the last normalization sub-process in previous pre-processing process. This process will be repeated until all the itemsets and there counts have been found. In the second phase, association rules are induced from the large itemsets found in the first phase, after setting the sets that contain the count of each set and the total number of the transactions, we can activate the association rule any time as follows:

- Input the support values (changeable).
- Divide every set by the total number of transactions

(Support {set} = count {set}/ count of transactions).

- Find the sets where Support {set} >= support value.
- Calculate the confidence.

All possible association combinations for each large itemset are formed, and those with calculated confidence values larger than a predefined threshold (minimum confidence) are given out as association rules.

Note:

- Itemsets with their counts in preceding runs are recorded for later use in maintenance.
- For the original database is scanned once only at the beginning and the counts are keep for any modifications in later stages.
- No support value will be added until running data mining, it will be inserted manually.

In the case were a new transaction is taken place, a new summation value is calculated for this transaction. This is stored in a predefined location (file). Scan the new transaction; calculate the number of all sets that equal ($2^{\#\text{ of new items}} - 1$). Once the numbers of itemsets are calculated the following may take place based on the individual new transaction.

Input a new transaction

If the transaction contains the same items that exist in the original set, add (+1) to each set and (+1) to the total number of transactions. If the transaction contains a new item that does not exist in the original set, break the transaction into {$2^{\#\text{ of new items}}$ -1} and add this new sets to the original sets, add (+1) to each old set, and (+1) to each new set and (+1) to the total number of transactions.

Delete an exist transaction

There is no interpretations, cause the transaction and the sets already exists, so add (-1) to each set and (-1) to the total number of transactions.

Update an existing transaction

In case of update an existing transaction all we have to do is delete an exist transaction (Delete exist transaction step), and then input a new transaction (Input a new transaction step). Note here we can continue as above; we have all the updated sets and there counts and the total number of updated transactions. (Hebah Nasereddin, 2012) Proposed an algorithm to generate a mathematical summation for each transaction. Based on these summation values the exact transaction in the local database that have been modified and needs to be replaced can be identified. In other words, if there are any modification affecting one or a number of transactions, it simply selects the transactions summation for the particular transaction; delete the old transaction then insert the new updated one, and make the changes needed related to the transaction with the modified summation value, this will result in the replacement of the transactions by their changed value from the source DB

Presentation of the (ARBSI)

The (ARBSI) is presented; the notations used in the algorithm are:

D: the original database;
T: the set of new transactions;
d: the number of transactions in D;
t: the number of transactions in T;
S: the support threshold;
C_k *: the set of all candidate k-itemsets from D;*
#items: the number of items from normalization sub process;
#new items: the number of updated items;

The (ARBSI) steps are explained as follows

INPUT: A support threshold *S*, is a set of transaction in D consisting of (*d*) transactions, and a set of *t* new transactions, and #items.

OUTPUT: A set of final association rules for the D and T.

STEP 1: Calculate the number of all sets equal $2^{\#items} - 1$.
STEP 2*:* Find all k-itemsetsC_k and their counts from the transactions.
STEP 3: Input S.
STEP 4: divide every set by the total number of d.
Support {set} = count {set}/ count of d.
STEP 5: Set the sets where Support {set} >= S. All possible association combinations for each large itemset are formed.
STEP 6: Calculate the confidence, those with calculated confidence values larger than a predefined threshold (minimum confidence) are given out as association rules.
STEP 7: If T is not empty (there is a new transaction): from the previous technique [16] we can find:

1. Wither it's an insert, delete and/or update case.
2. The item-summation, are recalculated and stored along with modification time.

Sup step 7.1: If Input is a new transaction:

1- Calculate the item-summation value.
2- Calculate the number of all sets equal ($2^{\#new\ items} - 1$).
3- Scans the sets to generate sets itemsets.
4- If the transaction contains some of the items that exist in the original sets, add (+1) to each set and (+1) to the total number of transactions.
5- If the transaction contains a new item that doesn't exist in the original set, break the transaction into {$2^{\#new\ items}$} and add these sets to the original set. Addition of (+1) to each new set and (+1) to the total number of transactions.

Sup step 7.2: If deleting an exist transaction

1- Select the transaction from the old item-summation
2- Calculate the number of all sets equal ($2^{\#new\ items} - 1$).
3- Scans the sets to generate sets itemsets.
4- Break the transaction into its sets and add (-1) to each set and (-1) to the total number of transactions.

Sup step 7.3: If updating an exist transaction

1. Select the transaction from the old item-summation.
2. Calculate the item-summation for the new modified transaction.
3. Calculate the number of all sets equal ($2^{\#new\ items} - 1$) for the old transaction, scans the sets to generate itemsets, break the transaction into its sets and add (-1) to each set and (-1) to the total number of transactions.
4. Calculate the number of all sets equal ($2^{\#new\ items} - 1$) for the modified transaction, scans the sets to generate itemsets, if the transaction contains some of the items that exist in the original sets, add (+1) to each set and (+1) to the total number of transactions, if the transaction contains a new item that doesn't exist in the original set, break the transaction into {$2^{\#new\ items}$} and add these sets to the original set. Addition of (+1) to each new set is necessary and (+1) to the total number of transactions.

End

The proposed algorithm (ARBSI) can thus find all large 1-itemsets for the entire updated database. After that, candidate 2-itemsets from the newly inserted transactions are formed and the same procedure is used to find all large 2-itemsets. This procedure is repeated until all large itemsets have been found.

Illustrative Examples

In this Section, an example is given to illustrate (ARBSI). Assume the initial data set includes 8 transactions, which are as shown in table 1. Note that TID number (200 and 800) and TID number (400,500) are having the same items which mean the same item-summation.

Table 1. An original database with *TID* and *Items*

TID	Items	Item-summation
100	ACD	I
200	BCE	I I
300	ABCE	V
400	ABE	I II
500	ABE	I II
600	ACD	I
700	BCDE	X I
800	BCE	I I

From the previous sub process (normalization) we know that the number of items is (5) which mean number of sets will be = ($2^{\#\ of\ items} - 1$), and equal ($2^5 - 1$) = 31. The sets of itemsets are shown in Table 2. The minimum support threshold *S* is not record here.

Table 2, The itemsets for the original database

Items	Items	Items	Items	Items
A	B	C	D	E
AB	AC	AE	AD	BC
BE	BD	CE	CD	ED
ABC	ABE	ABD	ACE	ACD
AED	BCE	BCD	CDE	BDE
ABCE	ABDE	ACDE	BCDE	ABCD
ABCDE				

All itemsets were generated and counted by scanning the original database (just once) without the consideration of the threshold value (minimum support), the sets of itemsets and there counts are shown in Table 3.

Suppose the minimum support threshold *S* is set at 50%, using a conventional mining algorithm such as the Apriori algorithm, all large itemsets with counts larger than or equal to 4; (8∗50% = 4) are found, as shown in Table 4.

{B, C, E} can be found to be a large 3-itemset. Next, the large itemsets are used to generate association rules. According to the condition probability, the possible association rules generated is shown in Table 5.

Since the user specified minimum confidence is 80%, the final association rules are shown in Table 6.

Table 3. The itemsets and there counts

Items	Count	Items	Count	Items	Count	Items	Count	Items	Count
A	5	B	6	C	6	D	3	E	6
AB	3	AC	3	AE	3	AD	2	BC	4
BE	6	BD	1	CE	4	CD	3	ED	1
ABC	1	ABE	3	ABD	0	ACE	1	ACD	2
AED	0	BCE	4	BCD	1	CDE	1	BDE	1
ABCE	1	ABDE	0	ACDE	0	BCDE	1	ABCD	0
ABCDE	0								

Table 4. All large itemsets from an original database with *s*=50%

Large itemsets					
1 item	Count	2 items	Count	3 items	Count
A	5	BC	4	BCE	4
B	6	BE	6		
C	6	CE	4		
E	6				

Table 5. Possible association rules

Rule	Confidence
IF B,C, Then E	Count(B,C,E)/Count(B,C)=4/4
IF B,E, Then C	Count(B,C,E)/Count(B,E)=4/6
IF C,E, Then B	Count(B,C,E)/Count(C,E)= 4/4
IF B, Then C,E	Count(B,C,E)/Count(B)=4/6
IF C, Then B,E	Count(B,C,E)/Count(C)=4/6
IF E, Then B,C	Count(B,C,E)/Count(E)=4/6
IF C, Then B	Count(B,C)/Count(C)=4/6
IF B, Then C	Count(B,C)/Count(B)=4/6
IF B, Then E	Count(B, E)/Count(B)=6/6
IF E, Then B	Count(B,E)/Count(E)=6/6
IF C, Then E	Count(C,E)/Count(C)=4/6
IF E, Then C	Count(C,E)/Count(E)=4/6

Table 6. The final association rules for this example

Rule	Confidence
IF B,C, Then E	Count(B,C,E)/Count(B,C)= 1
IF C,E, Then B	Count(B,C,E)/Count(C,E)= 1
IF B, Then E	Count(B, E)/Count(B)= 1
IF E, Then B	Count(B,E)/Count(E)= 1

Conclusion

Data mining algorithms have at least two issues that characterize a database perspective of examining data mining concept: Efficiency and Scalability. Ideally any solution to data mining problems must be able to perform well against real-world databases. As far as the efficiency is concerned some parallelization is used to improve or overcome this issue. Dynamic data mining pose significant challenges. It can discover up-to-date patterns invaluable for timely strategic decisions, but this has to be done accurately and quickly with limited computation resources. Mining process can expose long-term trends and more complicated patterns that lead to deeper insights, but more than often meaningful patterns can only be found in subspaces, which incur high complexity in pattern mining. This paper presents a two part solutions to the problem of Dynamic data mining. The first is concerned with process of detecting an update on the data after it has been collected for the data mining from its original source. The second deals with the process of maintaining the association rules based on the updates that have taken place on the original data in its original location. These two solutions when combined will allow the (ARBSI) to solve the problem of dynamic data mining only one scan to the original source of data. This will provide an efficient dynamic data mining technique. (ARBSI) works with massive real-world databases regardless of the amount of data and/or the amount of memory available. This algorithm also copies all updates that might take place in the original database to a dummy table specially created. This dummy table will contain a copy of the update records plus their summation value. And based on the summation value all the updated records are identified and all the necessary updates (insert, update, and delete) are carried out on the data used in the data mining process. The second part of the algorithm is used to maintain the association rules produced by the data mining process according to all updates carried out on the original sources of data. This process carries out this process using the data available in the dummy database containing the updated records and their summation value. Once it finished its task it clears the dummy database and waits for any new updates to take place. The paper also presents several examples to support the claims made. The results of the test showed that (ARBSI) is capable of carrying out a data mining process on a dynamic database that is being continuously updated, covering all the three updates (insert, update, and delete) transactions. This algorithm was also tested using both static and dynamic databases in both cases the proposed algorithm achieved its task with high efficiency. From the above it is clear that the goal of this paper has been accomplished, in the form of the development of a unique technique to deal with both static and dynamic Data Mining process. The results obtained proved that (ARBSI) is able to solve some of the problem related to the Dynamic Data Mining process

REFERENCES

Agrawal, R. and R. Srikant, "Fast algorithm for mining association rules," The International Conference on Very Large Data Bases, pp. 487-499, 1994

Agrawal, R., R. Srikant and Q. Vu, "Mining association rules with item constraints," The Third International Conference on Knowledge Discovery in Databases and Data Mining, pp. 67-73, Newport Beach, California, 1997.

Agrawal, R., T. Imielinksi and A. Swami, "Mining association rules between sets of items in large database,“ The ACM SIGMOD Conference, pp. 207-216, Washington DC, USA, 1993.

Anju k.kakkad, Anita Zala, "Incremental Association Rule Mining by Modified Approach of Promising Frequent Itemset Algorithm Based on Bucket Sort Approach", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 11, November 2013, pp 4390-4393.

Babu, S. and Widom, J. (2001). "Continuous Queries over Data Streams", Stanford University, SIGMOD Record, SIGMOD Record, 30:109–120.

Cheung, D.W., J. Han, V.T. Ng, and C.Y. Wong, "Maintenance of discovered association rules in large databases: An incremental updating approach," The Twelfth IEEE International Conference on Data Engineering, pp. 106-114, 1996.

Cheung, D.W., S.D. Lee, and B. Kao, "A general incremental technique for maintaining discovered association rules," In Proceedings of Database Systems for Advanced Applications, pp. 185-194, Melbourne, Australia, 1997.

Domingos, P. and G. Hulten. "Mining high-speed data streams". In Proc. of the 2000 ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pages 71–80, August 2000.

El-Hajj, M. and O. R. Za¨ıane.“ Non Recursive Generation of Frequent K-itemsets from Frequent Pattern Tree", representations. In In Proc. of 5th International Conference on Data Warehousing and Knowledge Discovery (DaWak'2003), pages 371–380, September 2003.

Fayyad, U. M., G. P. Shapiro, P. Smyth. "From Data Mining to Knowledge Discovery in Databases", 0738-4602-1996, AI Magazine (Fall 1996): 37–53

Hebah H. O. Nasereddin , "Stream Data Mining″, *International Journal of Web Applications,* Volume 3, Number 2, June 2011.pp 90- 97.

Hebah H. O. Nasereddin, "An Enhanced Item-Summation for Dynamic Data Mining Algorithm", *International Journal of Web Applications,* Volume 4, Number 4, December 2012.pp 173- 184

Hebah H. O. Nasereddin, "New Technique to Deal with Dynamic Data Mining in the Database", *International Journal of Research and Reviews in Applied Sciences,* Volume 13, Issue3, December 2012. Pp 806-814

Hebah H. O. Nasereddin, “Dynamic Data Mining Process” has been published in the ICITST-2008 conference, 23-28 June in Dublin, Ireland. pp 23-26.

Jiang, N., L. Gruenwald. “Research Issues in Data Stream Association Rule Mining” SIGMOD Record, Vol. 35, No.

Jiawei Han, Micheline Kamber. “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, Champaign: CS497JH , Fall 2001, www.cs.sfu.ca/~han/DM_Book.html.

Lin, M.Y. and S.Y. Lee, “Incremental update on sequential patterns in large databases,” The Tenth IEEE International Conference on Tools with Artificial Intelligence, pp. 24-31, 1998

Maria Halkidi, 2000. “Quality assessment and Uncertainty Handling in Data Mining Process” http://www.edbt2000. uni-konstanz.de/phd-workshop/papers/Halkidi.pdf

Mehmed Kantardzic J. B. “Data Mining: Concepts, Models, Methods, and Algorithms”, ISBN: 0471228524, IEEE Computer society, Wiley-Interscience, Hoboken, NJ, 2003.

Mohamed Medhat Gaber, ArkadyZaslavsky and Shonali Krishnaswamy. “Mining Data Streams: A Review”, VIC3145, Australia, ACM SIGMOD Record Vol. 34, No. 2; June 2005.

Muthamilselvan, T., N. Senthil Kumar, I. Alagiri, “Finding Association Rules Based on Maximal Frequent Itemsets over Data StreamsAdaptively”, *International Journal of Advanced Research in Computer Science*, Volume 3, No. 2, March-April 2012, pp 118-120.

Park, J.S., M.S. Chen, P.S. Yu, “Using a hash-based method with transaction trimming for mining association rules,” IEEE Transactions on Knowledge and Data Engineering, Vol. 9, No. 5, pp. 812-825, 1997

Qingguo Zheng, Ke Xu, Shilong Ma; “When to Update the Sequential Patterns of Stream Data”; Pacific-Asia Conf. on Knowledge Discovery and Data Mining; 2003.

Sarawagi, S., Thomas, S., and Agrawal, R. 1998. “Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications, In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD’98), Seattle, WA, pp. 343–354.

Tzung-Pei Hong, Ching-Yao Wang, Yu-Hui Tao. “A new incremental data mining algorithm using pre-large itemsets” Intelligent Data Analysis, Issue: Volume 5, Number 2 / 2001, pages: 111–129.

Tzung-Pei Hong, Tzu-Jung Huang. “Maintenance of Generalized Association Rules for Record Deletion Based on the Pre-Large Concept”, Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February16-19,2007,www.wseas.us/e-ibrary/conferences/2007corfu/papers/ 540-410.pdf

Yaqiong Jiang , Jun Wang, “An Improved Association Rules Algorithm based on Frequent Item Sets”, Procedia Engineering, Volume 15, 2011, Pages 3335–3340

Zhang, S., “Aggregation and maintenance for database mining,” Intelligent Data Analysis, Vol. 3, No. 6, pp. 475-490, 1999.

****