

Kalman Filter based Flexible Sliding Window Algorithm for Mining Frequent Itemset over Data Stream

Shailvi Maurya

Department of Information Technology
SATI College
Vidisha, M.P. (India)

S. K. Shrivastava, Ph.D

Department of Information Technology
SATI College
Vidisha, M.P. (India)

ABSTRACT

A data stream mining is relevant issue in the context of information gain. The data arrived are large in amount limitless and high rate with time impractical to stock, excavating and testimony at identical measure of time to retrieve intelligence. Sliding window model utilized for frequent pattern mining data stream mining emphasis on recent data and minimum space consumed. In past algorithm window measurement change was steady to concept variation when stagnant and gets smaller when the concept variation happens. Renewed frequent patterns are moderately kept in the current concept whereas the stable transaction is moved out of window. Panes steadily combined to window and performing unnecessary mining for frequent itemsets, conduct is diminishing. Based on the sliding window model the new algorithm named KF_FSW (Kalman Filter based Flexible Sliding Window Model) which utilize Kalman filter function for prediction and measurement approach. The prediction and measurement method is done on basis of already existing information as measure. Thus coagulating the error for accurate position of behavior variation in window size fluctuates in streamed database. Test on standard dataset reports that proposed algorithm coagulates less number of windows for mining and even predicting efficiently the number of count for change ratio captured by occurring change variation.

Keywords

Data Mining, Data Stream, Sliding Window Model, Frequent itemsets, Kalman filter, Concept Change.

1. INTRODUCTION

Data Mining is being defined as extracting useful knowledge from huge set of data. Data mining is basically a process of discovering hidden patterns and information from existing data [1]. In other words data mining means mining knowledge from data. The process of discovering novel patterns, insightful, interesting as well as prognosis, explanative, understandable model out of large-scale data is called as data mining.

Due to the advancement in technology the data generated is continues unbounded massive and high speed called as data stream [2] often changes with time. Mining data streams is issue in data mining. Many applications generate large amount of data streams in online duration, such as sensor information accumulated from sensor link, real transaction arrival in retail link, click-streams Web record of in Web appositeness, records in telephonic conversation, and conduct coagulation in network observing and traffic maintenance [3]. Mining data stream must satisfy following requirement:

- **Memory usage:** Data stream is generated continuously and massive so the memory usage should be low for low cost.

- **Fast processing:** the processing speed should be high for better performance because slow processing will lead to high time consumption.
- **Scan:** The scanning must be done once due to the dynamic nature of the stream and limited amount of memory.
- **Updating:** the updated solution of investigation must be available rapid on user's request.

Mining frequent patterns refers to the discovery of the sets of items that frequently appear in a transaction database. Extraction of frequency of itemsets (or patterns) [4] is an interesting and fundamental problem in the context of knowledge discovery and data mining. This is due to its wide appositeness area in business planning, industry growth and science exploration. Various types of mining algorithms are Apriori algorithm [5], FP- growth tree, Partition algorithm, and Eclat algorithm [6], Clique algorithm [4] where apriori algorithm is the most usual algorithm for extracting frequent itemsets. A set of items is frequent in a real dataset if the number of its occurrences in the database is exceeds or equal to a user specified threshold. This threshold is coagulated by the user of the mining process. To extract data stream accurately many scanning are not acceptable. Mining on recent data is more attractive than the old history in data streams. It is challenging to extracting frequency of itemsets from recent data, because new items coming and old items overdue with high speed [7].

Sliding window [8] is one of the major models for data stream processing and extraction, in which a static measure of currently incoming data took in considered. In a sliding window over a transaction based data steam, by the incoming of unfamiliar transaction, the previous transaction is deleted from the window and the new transaction is injected into the window. Therefore, it always maintains the newest transactions. The window is usually accumulated and pileup within the main memory for rapid processing.

Concept refers to the target variables, which the model is trying to predict or to describe. Concept [9] change is the variation of the underlying behavior over time. The concept change is a common fact in data stream processing due to variant nature of incoming data. Moreover, due to changing behavior of data streams, a data stream mining algorithm must be able to adapt with the occurrence of concept change occurs when the data generating process changes from one data generating prototype to another one. Concept change can be categorized in two main types; concept shift and concept drift [10]. Drifting concept dissipated a progressive variation of the concept and concept shift occurs when a variation of two behaviors is more abrupt.

This change is the results of a change in the parameters of the model [11]. The concept change makes frequent itemset mining in data streams even more challenging than traditional

static databases. For Illustrations, in an online market, the buying preference of customers may change with time which affects the products that they asset together. In this problem, the user can be learned for new solutions when a concept change is captured [10].

The Kalman filter is a set of mathematical equation mensuration that gives an efficient coagulation (recursive) means to determine the state of a process in a way that reduces the mean of the squared error. The filter is efficient in many aspects: it supports prognosis of previous, current, and upcoming states, and it can do so even when the variant behavior of the prototype system is unknown [12].

The next sections in paper are organized as follow. Section 2, we discuss about related work of various existing algorithm. The innovation proposed is described in section 3. Section 4 explains the experimental interpretation about proposed algorithm. Lastly, conclusion of research is summarized in section 5.

2. RELATED WORK

In this section various algorithms for mining frequent itemsets over fixed or varying data was developed were Apriori algorithm [10] based on association rule was given by Agrawal and Srikant for stable database. The algorithm searched association guideline between large set of dataset of sales transactions. The algorithm has tremendous scale linearly with count of transitions thus reducing the execution time as item number increased within database. It comes with drawback of multiple scan over static data.

Next concept developed for association rule known as Eclat [4] algorithm (Equivalence Class Transformation) Mohammed J. Zaki for fixed database which utilized the structural characteristic of frequent itemsets for fast searching. The items are arranged into lattices which further distinguish into sub-lattice which can be solved in memory. Efficient traversing techniques are used which detect the long frequent itemsets quick. The algorithm requires less than 0.018 Mb memories which are 2% of whole database size, but cannot overcome with the problem of multiple scan over fixed database.

The estDec [11] method on data stream which is variant in nature was designed by Chang et. al.. The algorithm discovers recent frequent itemsets over an online data stream by decaying the measure of past transactions as time pass away. As a conclusion, the recent change of knowledge in a data stream adaptively mirrored the current mining solution of the data stream. The measure of knowledge in a transaction of a data stream is gradually reduced as time pass away while reducing rate can be variably controlled. Therefore, no transaction needs to be handled manually. It utilizes minimum execution time and memory as compare to other algorithms developed earlier along with requires many scan of the online data.

Another algorithm on data stream was given by Leung et.al. for extracting useful knowledge. The algorithm was based on a novel tree designing, called DSTree (Data Stream Tree) [13], which captures important data from the stream. It uses window behavior where tree detect the elements of transaction in a window and sort nodes to canonical form which is not affected by changes in item frequency. The algorithm has linear scalability along with has reduced execution time when support increases but requires multiple scan of real data.

Due to real-time activity and computational difficulty Mozafari et. al. which introduced a novel verification method

SWIM (Sliding Window Incremental Miner) [14] based on sliding window for extraction over data stream. This algorithm modifies effectiveness by allowing small message delay. Thus algorithm shows excess scalability, flexibility and efficiency for extracting frequent itemsets on data stream with huge windows where it needs single scanning of complete incoming stream.

Challenging features, like unknown or limitless size, possibly rapid incoming rate, inability to backtrack past arrived transactions, and missing of system power over the sequence in which the data arrived Hua-Fu Li et. al. gave an effective bit-series based, single-pass algorithm, called MFI-Trans-SW (Mining Frequent Itemsets within a Transaction-sensitive Sliding Window) [15], to mine set of frequent itemsets from data streams within a transaction-sensitive sliding window which consists of a fixed number of transactions. The proposed algorithm shows accurate results, runs faster and consumes less memory comparatively then previous algorithms.

Another algorithm based on prefix tree based structure called as BFI- tree [7] was innovated by Kun Li et. al. to maintain correct frequent itemsets from moving windows over data stream. Monitoring the edges between frequent itemsets and infrequency of itemsets, it stops update on small part of the tree. It requires one scan over streamed data for frequent itemset extraction along with time effective and returns correct solution by outperforming MFI-TransSW in time and storage complexity.

For extracting frequent patterns from recent data improves the investigation of data stream efficient approach Compact Pattern Stream tree (CPS-tree) [16] developed by Tanbeer et. al. search complete set of current frequent pattern from high rapid rate data stream over moving window. It also initiated the concept of variant tree restructuring in CPS-tree to generate frequency- decreasing tree structure at execution time. The algorithm is efficient in time complexity, low memory consumption over single scan of streamed database.

Sliding window prototype is widely utilized for extraction over data stream for mining current data. Mahmood Deypir et.al. given concept of VSW (variable Sliding Window) [17] where the window size measure is actively adjusted based on the volume of the nature variation incoming within the data stream. The window grows as the concept is fixed and gets smaller when variation in concept captured. The algorithm effectively captures concept change, resizes the window size and acquires its self to new concept by scanning once the data stream but come along with the drawback of adding panes every time for mining if concept variation is not captured.

Fatemeh Nori et. al. developed the structure of Tmoment [18] algorithm. The algorithm utilizes data structure for stocking transactions of the window and responding frequent closed itemsets. The Tmoment algorithm extract closed frequent itemsets within sliding window over data stream. It upgrades the new transactions incoming and old transactions are deleted. This innovation is suitable for high-speed and limitless with one scan over transactional data streams. The algorithm consumes less time for execution along with lower in memory level.

3. PROPOSED ALGORITHM

In this research work the proposed algorithm contains four main states:

Window initiation and frequent itemset mining

- 1) Pane insertion at initial stage and prediction using Kalman filter for rest of stream
- 2) Frequent itemsets updating
- 3) Dropping useless knowledge from the window

After the window initialization, the procedure of pane addition at initial state and then prediction is used to prognosis the behavioral change in the stream according to which the window actively adjust the window, upgrading frequent itemsets and removing previous information, steadily goes forever for limitless data streams.

3.1 Window Initiation and Frequent Itemset Mining

During the mining procedure the window is declared by user only at the initial state, next time the window size is automatically adjusted based on concept variation appearing in data stream. The initial and last point of window is defined by checkpoints. Assume that panes are added to the recent window from left to the right. Concept variation is coagulation with respect to checkpoint (cp) [17]. As shown in 4.1, if a concept variation is captured, all information before the checkpoint is deleted from the window and new window ω is created. The checkpoint is shifted to the new point where concept variation is captured.

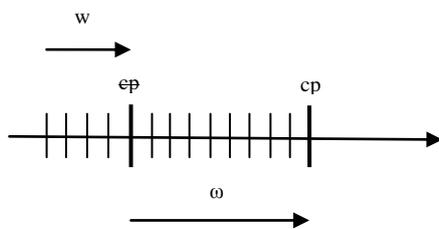


Fig 1: Window size reduction and checkpoint (cp) movement

After window is being initialized Eclat [6] algorithm is used for mining over the window. The Eclat algorithm for mining is very similar to Apriori algorithm [10] which was used first for mining frequent itemset over data stream. Eclat algorithm was given by M. J. Zaki for mining frequent itemset over static database.

3.2 Pane Insertion at Initial Stage and Prediction using Kalman Filter for Rest of Stream

In the next step after extraction of frequent itemset new pane is added only once. After the adding of the new pane again mining of window is performed to search new frequent itemsets. Distinguish of two concepts is explained based on the volume of change in the set of frequent patterns. If the difference greater from the user initialized threshold, we say that a concept change has been detected. Given two time duration called as T and T^1 in a data stream where $T^1 > T$, let F_T and F_{T^1} denote the set of frequency of itemsets at time T and T^1 . Therefore $F_T^+(T^1) = F_{T^1} - F_T$ is the set of the newly detected frequent itemsets at T^1 with respect to T . So therefore $F_T^-(T^1) = F_T - F_{T^1}$ is set of infrequency of itemsets at T^1 which was frequent at T . The change ratio of frequent itemsets at T^1 in comparison to T is declared as:

$$\text{Change ratio}_T(T^1) = (|F_T^+(T^1)| + |F_T^-(T^1)|) / (|F_T| + |F_T^+(T^1)|)$$

Where $|F_T|$ is the count of itemsets in set F_T . The change ratio is a value lies between 0 and 1.

If the Change ratio lies between 0 and 1 then concept variation has occurred and if the change ratio exceeds than minimum threshold the window is resizes actively. After resizing the window size value is measured. If the window size is exceeds 0, algorithm switches to the Kalman filter function.

The Kalman filter algorithm [19] works by predicting the position where the window can resize for further mining. The prediction contains noise. Based on the past knowledge the mensuration is done indicating the approximate location of the position with some noise added to the window. Error is the difference between the prediction and mensuration. The Kalman gain is also coagulated to find out whether the predicted position coagulated by giving any new information for further frequent mining and resize of the window based on the prediction techniques.

The Kalman filter model considers that the state of a system at a time t evolved from the past state at time $t-1$ according to the equation

$$x_t = F_t x_{t-1} + B_t u_t + w_t$$

where

- x_t is the vector of state containing the terms of interest for the system at time t
- u_t is the control inputs vector
- F_t is the state transition matrix
- B_t is the control input matrix
- w_t is the vector containing the process noise terms for each parameter in the state vector.

Measurements of system is performed as

$$z_t = H_t x_t + v_t$$

where

- z_t is the vector of measurements
- H_t is the transition matrix that maps the state vector parameters into the measurement domain
- v_t is the vector describing the measurement noise terms for each observation in the measurement vector.

3.3 Frequent Itemsets Updating

After the prediction of state vector [19] of the stream where concept variation will be captured and extraction is performed at particular window specified by position predicted. Similar to the Eclat [5], we compute the support of candidate itemsets by intersecting their transaction. Itemsets having support exceeding minimum support threshold are updated to the prefix tree and new candidate itemsets generated by new frequent itemsets are added to the set of candidates. The procedure moderately process until there is no more new candidate itemset.

3.4 Dropping Useless Knowledge from the Window

An itemset which are infrequent after a prediction is deleted from the prefix tree. Dropping of the static useless information is processed on the basis of the checkpoint. Thus the whole process moderately goes till to the maximum of the stream length [9].

Algorithm KF_FSW (Pane_Size, Window_Size, change_th, min_sup)

```
1) W = Window_Size; // window initialized
2) CP = W; //last transaction of window is set to initial value of checkpoint
3) CP_start = CP; // starting of checkpoint initialized to CP
4) CP_pre = 1; // first transaction of window is set to 1
5) W_val = W; //number of windows changed
6) Forever //the process continues unlimitedly
7) I = 1:100;
8) FISet = Eclat(W, min_sup) //initial window mined using the Eclat algorithm
9) If i == 1
10) FIS_pre = FISet; // initialize the itemset to pre frequent itemset
11) End
12) Forever i = 1:FIS_new
13) If sum(strcmp(FIS_pre, FIS_new) > 0 //comparing new and previous frequent itemset
14) FIS_val = 1; // stores values for common frequent itemsets
15) Else
16) FIS_val = 2; // stores the value for new frequent itemsets
17) End
18) End
19) FIS_new_rules = sum(FIS_val == 2); // adding the values of new frequent itemset
20) FIS_common_rules = sum(FIS_val == 1); // adding the values of common frequent itemset
21) FIS_remove_rules = FIS_pre - FIS_common_rules; // deleting infrequent itemset
22) FIS_CR=(FIS_new_rules+FIS_remove_rules)/(FIS_pre+FIS_new_rules); // calculating change ratio
23) If FIS_CR > change_th //comparing change ratio with minimum change threshold
24) CP_pre = CP_start; // previous checkpoint moves to start checkpoint
25) CP_start = CP; // start checkpoint initialized to current checkpoint
26) W_val = CP - CP_pre; //calculating window size value.
27) End if
28) If W_val > 1 // checking condition greater the 1
29) Fun_kalman_est(W_val); // Kalman filter function.
30) CP = CP + round(W_val_pred); // checkpoint is incremented
31) Else
32) CP = CP + Pane_Size; //checkpoint incremented based on pane size
33) End if
34) Else
35) CP = CP + Pane_Size; // checkpoint incremented based on pane size
36) End if
```

Fig 2: Kalman filter flexible sliding window algorithm (KF_FSW)

4. EXPERIMENTAL RESULTS

In this research work, the performance parameters have been coagulated on the window size and change ratio. To estimate the efficiency on these parameters we have used standard dataset. We use the BMS-Webview-1 [20] dataset. The dataset contains several months' worth of click stream data from an e-commerce web site. In this dataset, all product information pages viewed by a customer during a web session are considered as a transaction. We select this dataset due to the variant nature of clicking behavior of the customers [21]. For all tests about this dataset, we set the size of pane, window initialization size and minimum support to 1.5K transactions, 5 K transactions and 0.5% respectively. The implementation is performed in MATLAB 8.0.0, 2012. To explain the performance of the proposed technique a comparison has been done between proposed technique and the base paper method of mining over data stream by using sliding window model. It is worth noting that the proposed technique has proven better than the base paper technique as it out performs in efficiency as compare to the base paper. The comparison in graphical and tabular data shows drastic change in the window size and change ratio detected according to the concept change occurring in the data stream.

4.1 Performance Measures

The test evaluation of the proposed approach is performed using the following criteria:

The parameters which are required in the proposed work are

- 1) Size of the sliding window
- 2) Change ratio

4.1.1 Size of the Sliding Window

The size of the window varies when the change is captured by prediction and mensuration, and resizes the window size according to the new concept by removing static useless transactions belonging to the first concept. The algorithm resizes the window based on the observed amount of changes.

Table 1 a standard dataset is utilized in the proposed work to check the number of change in window size for 30000 transactions. The correlation of the conclusion of proposed algorithm KF_FSW with respect to the base algorithm VSW over 1.5K transactions in a single pane size and the table No. of window size changed in shown in terms of K transactions. The values of both proposed and base algorithm shows proposed algorithm the number of window size changes is very less as compare to the base algorithm. Because of the prediction and mensuration technique the size is reduced as it predicts were the concept is going to variant in the transaction added in form of panes

Table 1. Window size of KF_FSW with respect to VSW at pane size 1.5K

No. of window's size changed (K Transactions)	KF_FSW (K Transaction)	VSW (K Transaction)
Window 1	5000	5000
Window 2	5500	6000
Window 3	4783	1500
Window 4	5368	1500
Window 5	4902	1500

Window 6	-	1500
Window 7	-	1500
Window 8	-	1500
Window 9	-	1500
Window 10	-	1500
Window 11	-	1500
Window 12	-	1500
Window 13	-	1500
Window 14	-	1500

Figure 3 shows the graphical representation of the correlation between the new algorithm KF_FSW with respect to base algorithm at pane size 1.5K, initial window size 5K, minimum support is 5. The graph shows the variation in the number of window size. The vertical axis shows the K transactions which indicate the variation in the size of window changing every time. The horizontal axis shows the number of times the window has changed because of the changes occurring in the stream and those changes are proving any knowledge gain. In proposed algorithm 5 times window is changed whereas in base algorithm 14 times window is changed even though any knowledge gain is not retrieved. This variation shows that the proposed algorithms is effective in case of storage as less windows will needs less memory.

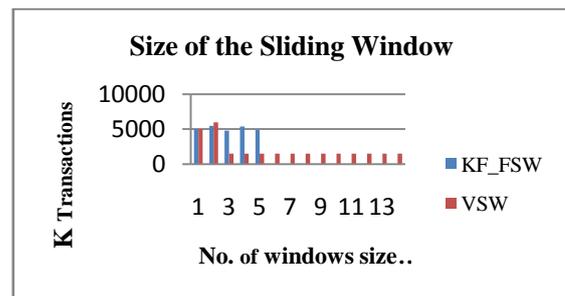


Fig 3: Pattern of window size for KF_FSW with respect to VSW window size at pane size 1.5K

4.1.2 Change Ratio

The concept change detection and window resizing behavior during a data stream mining depend on the changing behavior of the data stream and the given change threshold. A lower value for the change threshold leads to the capture of more concept changes, and consequently, this will have the effect of smaller window sizes during a data stream mining.

Table 2 a standard dataset is used in the proposed work to check the number of concept change 30000 transactions. The comparison of the result of proposed algorithm KF_FSW with respect to the base algorithm VSW over 1.5K transactions in a single pane size, in the table No. of change ratio in shown in terms of K transactions. The values of both proposed and base algorithm shows that in proposed algorithm the number of change ratio is less as compare to the base algorithm. Because of the prediction and measurement technique, the size is reduced as it predicts were the concept is going to change in the transaction inserted in form of panes.

The values obtained are called as change ratio in frequent itemset of the past window and recent window which is due to the variation in the data stream. The change ratio value lies between 0 and 1.

Table 2. Change ratio of KF_FSW with respect to VSW at pane size 1.5K

No. of change ratio greater than minimum threshold	KFS_VSW (K Transactions)	VSW (K Transactions)
Change ratio 1	0	0
Change ratio 2	0	0
Change ratio 3	0	0
Change ratio 4	0	0
Change ratio 5	0.5	0.5
Change ratio 6	0.888889	0.875
Change ratio 7	1	1
Change ratio 8	0.875	1
Change ratio 9	-	0.9
Change ratio 10	-	0.888889
Change ratio 11	-	1
Change ratio 12	-	0.875
Change ratio 13	-	0.9
Change ratio 14	-	0.875
Change ratio 15	-	1
Change ratio 16	-	0.75
Change ratio 17	-	0.75

Figure 4 shows the graphical representation of the comparison between the new algorithm KF_FSW with respect to base algorithm at pane size 1.5K, initial window size 5K, minimum support is 5. The graph shows the variation in the number of concept change which leads to change ratio. The vertical axis indicates the change ratio between 0 and 1 which indicate the variation in the concept of data stream changing every time. The horizontal axis shows the number of times No. of times concept change exceeds minimum threshold detected in the stream and these changes are not relevant for any knowledge gain. In proposed algorithm 8 times window is changed whereas in base algorithm 17 times window is changed even though any information gain is not retrieved. This variation shows that the proposed algorithms performs better in case of storage as less windows will require less memory. Thus this shows that the prediction technique in new algorithm KF_FSW in finding more correctly where concept change occurring in data stream due to dynamic nature.

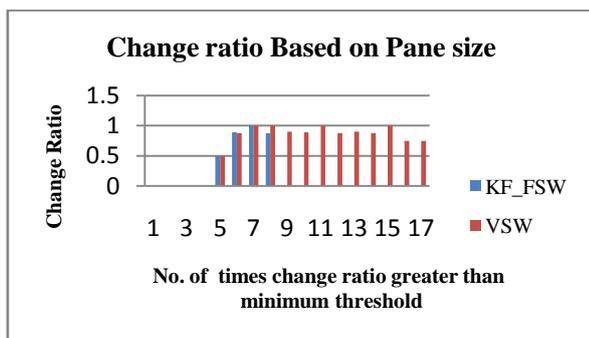


Fig 4: Change ratio for KF_FSW with respect to VSW window size at pane size 1.5K

5. CONCLUSION

The objective or the aim of this research work was to gain a deeper understanding about the Data Stream and their application and all the details that are required to know about mining data stream. It's a new task as it involved investigation by prediction and mensuration of new state in data stream where the concept variation is going to occurs utilizing Kalman filter function. Instead of joining panes in sliding window every time for mining the data stream, the state transition of the stream where the concept is likely to be change is captured and then mensuration is performed. Correction of state is done by coagulating the error between prediction and mensuration, thus finding the issues in previous technique of mining using sliding window over streamed data as compared to the proposed technique using Kalman filter function. The experimental interpretation insures that the number of window consumed for mining in Kalman filter based sliding window is comparatively very less than previous algorithm based on variable length sliding window. Even the number of times change ratio detected in KF_FSW is minimum than past approach because prediction and mensuration function is utilized for capturing nature variation of stream.

New technique opened way for further research in data stream mining. Different version of Kalman filter can be utilized based on the parameters for enhancing the attainment of proposed Kalman filter based flexible sliding window model algorithm for extracting knowledge in limitless, variant and moderately flowing data stream.

6. ACKNOWLEDGMENTS

I am thankful to Dr. S. K. Shrivastava for his guidance for helping in research work and making the research paper.

7. REFERENCES

- [1] PhridviRaj, M. S. B. and GuruRao, C. V. 2014. "Data mining – past, present and future – a typical survey on data streams", The 7th International Conference Interdisciplinarity in Engineering, Procedia Technology 12 255 – 263.
- [2] Aggarwal, C. C. 2007. "Data Streams – Models and Algorithms", Springer
- [3] Deypir, M., Sadreddini, M. H. and Hashemi, S. "A dynamic layout of sliding window for frequent itemset mining over data streams", The Journal of Systems and Software, 2012, 85 746– 759.
- [4] Zaki, M. 2000 "Scalable algorithms for association mining", IEEE Transactions on Knowledge and Data Engineering, 12(3), 372–390.
- [5] Borgelt, C., 2003. "Efficient Implementations of Apriori and Eclat", Workshop of Frequent Item Set Mining Implementations.
- [6] Thieme, L. S., 2005. "Algorithmic Features of Eclat", Computer-based New Media Group.
- [7] Li, K., Wang, Y. Y., Ellahi, M and Wang, H.-an. 2008. "Mining recent frequent itemsets in data streams", IEEE fifth Int. conf. on Fuzzy System and Knowledge Discovery.
- [8] Chandrika, J., Kumar, Dr. K. R. A. "Frequent Itemset Mining in Transactional Data Streams Based on Quality Control and Resource Adaptation", International Journal

- of Data Mining & Knowledge Management Process (IJDKP), 2012, Vol.2.
- [9] Kifer, D., David, S. B., Gehrke, J. 2004. “Detecting Change in Data Streams”, Proceedings of the 30th VLDB Conference.
- [10] Agrawal, R., and Srikant, R. 1994. “Fast algorithms for mining association rules”, In Proc. VLDB int. conf. very large databases, pp. 487–499.
- [11] Chang, J. and Lee, W. S. 2003. “Finding recently frequent itemsets adaptively over online transactional data streams”, Information Systems, 31(8), 849–869.
- [12] “The Kalman Filter”, Internet: www.ion.org/museum/item_view.cfm?cid=6&scid=5&iid=293.
- [13] Leung, C. K- S and Khan, Q. I. 2006. “DSTree: A tree structure for the mining of frequent sets from data streams”, In Proc. ICDM, pp. 928–932.
- [14] Mozafari, B., Thakkar, H. and Zaniolo, C. 2008. “Verifying and mining frequent patterns from large windows over data streams”, In Proc. int. conf. ICDE, pp. 179– 188.
- [15] Li, H.-F. and Lee, S.-Y. 2009. “Mining frequent itemsets over data streams using efficient window sliding techniques”, Expert Systems with Applications, 36(2), 1466–1477.
- [16] Tanbeer, S. K., Ahmed, C. F., Jeong, B. S. and Lee, Y. K. 2009. “Sliding window-based frequent pattern mining over data streams”, Information Sciences, 179 3843–3865.
- [17] Deypir, M., Sadreddini, M. H. and Hashemi, S. 2012. “Towards a variable size sliding window model for frequent itemset mining over data streams”. Computers & Industrial Engineering, 63 161–172.
- [18] Nori, F., Deypir, M., Sadreddini, M. H. and Hashemi, M. H. “A sliding window based algorithm for frequent closed itemset mining over data streams”, The Journal of Systems and Software, 2013, 86 615– 623.
- [19] Faragher, R. 2012. “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation”, IEEE Signal Processing Magazine
- [20] “BMSWebView1 (Gazelle) (KDD CUP 2000)”, <http://www.philippe-fournieviger.com/spmf/index.php?link=datasets.php>
- [21] Flouvat, F. 2005. “Experimental study of frequent itemsets datasets”.