International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-04, July 2018

Top-K Utility Frequent Itemset Mining from Multiple data Streams

[1]Monika Vikas Deore, [2]Prof. N. R. Wankhade

[1]M. E. Student, [2]H.O.D, Late G. N. Sapkal College Of Engineering, Nashik, India.

[1]*monikanjali@gmail.com* , [2]*nileshrw2000@yahoo.com*

Abstract **Lot of application in current era generates continuous data. In data mining techniques utility itemset extraction and frequent itemset extraction are important research area. Mainly in retail market analysis, these two techniques are required for market basket analysis. In existing work these two techniques are studied independently. The proposed work aims to find utility frequent itemset over multiple data stream. Along with the frequent itemset extraction it also takes in to account the utility of product. There is change in transaction dataset due to insertion and/or deletion of transactions. The system extracts top k itemset at the current movement. The system follows sliding window protocol to update the top k itemset from multiple data streams. For extraction of utility frequent itemset: utility table, CP-Graph, a hybrid index of graph and inverted file structures are used. The performance evaluation parameter compares the results with existing top k frequent itemset mining technique in terms of extracted itemset, time and memory evaluation.**

Keywords - CP-graph, data streams, frequent itemset, utility itemset, top k itemset.

I. INTRODUCTION

In frequent itemset mining, set items are extracted those are frequently occur in dataset with minimum support value. The support value is the user defined threshold value. Support value adds the minimum occurrence constraint on frequent itemset mining strategy. Frequent itemset/ pattern mining is applicable in variety of domains such as market basket analysis, retail market analysis, intrusion detection, web click mining, network monitoring, bioinformatics, etc.

In real life scenarios, the data is generated in streaming format. A single or multiple streams are generated in variety of application. Hence frequent itemset mining over data stream is important task. Frequent itemset mining play vital role in variety of applications such as:

- In social networks like facebook , twitter, etc generates bulk streaming in every day. The relationship among multiple post/tweets can be extracted by matching keyword in it. More frequently occurred tweets/post can define a social media trend. Frequent itemset mining is useful in trend analysis.
- In E-COMMERCE strategy product promotions, recommendations can be performed using frequent pattern analysis. In ecommerce the purchase history of multiple

  users can be traced to find sequence of items in purchase strategy.
- Association mining: With the help of application usage statistics in smart phones, usage pattern can be extracted. Usage pattern may contain location specific application access, user profile specific application access. This helps in statistical study of various application categories.

In multiple data streaming, all streams data is merged together to generate a dataset. Along with the support evaluation closed co-occurrence patterns analysis is done in multiple streaming data mining. In closed co-occurrence pattern analysis, pattern occurrence is checked with be 2 or more streams. The frequent item should be present in at least 2 streams.

There are various algorithms are proposed for frequent pattern analysis like apriory, Eclat, Fp-Growrh, etc. These algorithms cannot be directly applied to continuously changing the streaming data. There is need to implement different strategy for stream data analysis.

A utility itemset extraction is a technique in which itemsets are extracted from a dataset that generates higher profit. The frequent itemset extraction and utility itemset evaluation are two important techniques in retail market analysis.

 DOI : 10.18231/2454-9150.2018.0449

In the following section various techniques related to frequent itemset mining and utility extraction are studied.

II. REVIEW OF LITERATURE

Arnaud Giacometti, Dominique H. Li, Patrick Marcel, Arnaud Soulet proposes a survey based on last 20 years work in the domain of pattern mining[2]. This survey provides the overview of 1,087 publications papers. The work includes variety of pattern extraction using association riles and itemset. The pattern mining techniques are mainly classified in following 6 categories:

a. Pattern mining on static data:

A whole dataset is provided to the system at ones. The system will extract the patterns by analyzing complete data. For such analysis apriory[3] and FP growth[4] are two main techniques used in literature.

In apriori algorithm the execution is divided in two phases. In first phase candidate items are extracted and in second phase frequent items are extracted. This apriory technique uses breadth-first search to find next probable frequent itemset.

FP growth algorithm technique uses FP tree for frequent itemset evaluation. This technique reduces the database scan and it does not generate the candidate itemset. Fp growth technique is faster than apriory technique. A combine techniqe of FP growth and apriori algorithm is present in [5].

Mining top-k frequent closed itemsets[6] is proposed to extract top k items using apriory and fp growth technique. Differential privacy based frequent itemset[7] mining technique provides the privacy in frequent data mining. This technique adds the noise in data before analysis to provide data privacy.

b.Frequent pattern mining over a single stream:

G. S. Manku and R. Motwani[8] proposed a technique of mining frequent itemset over streaming data. FP-Stream[9] technique is used to find current frequent patterns and prediction of future pattern occurrence. These techniques provide approximate results. varying-size sliding window technique[10] is used to provide high accuracy in solution.

To extract exact frequent patterns from dataset DStree[11] technique is proposed. But this technique has several limitations like tree structure is not compact, storage overhead issue, etc. To overcome these problems CPS-tree[12] technique is proposed. This three has compact tree structure and hence reduces the storage overheads.

To mine frequent itemset over streaming data apriory based[13] and FP-growth[14] base techniques are also proposed. These techniques update the data structure incrementally with respect to every sliding window batch.

Mining Top k closed patterns from data stream [15][16] is also proposed in literature. The patterns are extracted from a single streaming window.

c. High utility itemset extraction:

Frequent itemset mining technique extracts the items those are occurs frequently in transactions but it may discover the low value itemset i.e. low profit itemset and can lose the information of high valued itemsets. User is interested in finding high profit itemset in the transactional dataset. Apriori-based algorithm for mining High utility Closed itemsets[17] extracts the high utility itemset based on predefined threshold. To overcome the problem of user defined threshold value, Mining top k High utility itemset technique is proposed to extract the top k high profit elements from the dataset [18].

d. Utility based frequent itemset extraction:

The utility itemset extraction along with frequent itemset extraction is proposed in letature[19]. It introduces a new concept as utility frequent itemset. This algorithm is run in two phases initially frequent items are extracted as a candidate items and then based on the utility value the items are filtered.

e. Frequent pattern mining across multiple databases:

Multiple static databases are considered to find frequent itemset. Multiple databases with user defined threshold values for inter-frequency and intra-frequency matching of itemset is proposed in[20].

f. Frequent pattern mining across multiple streams:

Data with multiple streams is analyzed using seg-tree[21]. In this technique segment tree is generated as a data structure to find patterns in transaction dataset. A CoolMine algorithm is proposed to travel the seg-tree and obtain the common patterns. But this algorithm is computationally expensive.

In[1] cp-grap based multiple streaming frequent itemset mining is proposed. This technique overcome the drawback of CoolMine algorithm[21]. This technique finds top k frequent itemset from multiple stream using closed co-occurrence patterns technique.

III. PROBLEM FORMULATION

Multiple real life examples generate data streams. A same set of object can appear in more than one stream. The existing work includes variety of techniques such as single stream processing, multiple stream processing. The utility based frequent itemset are extracted from static dataset. There is

 DOI : 10.18231/2454-9150.2018.0449

need of such system that provides solution for utility based frequent itemset extraction technique for multiple data streams.

IV. SYSTEM ARCHITECTURE

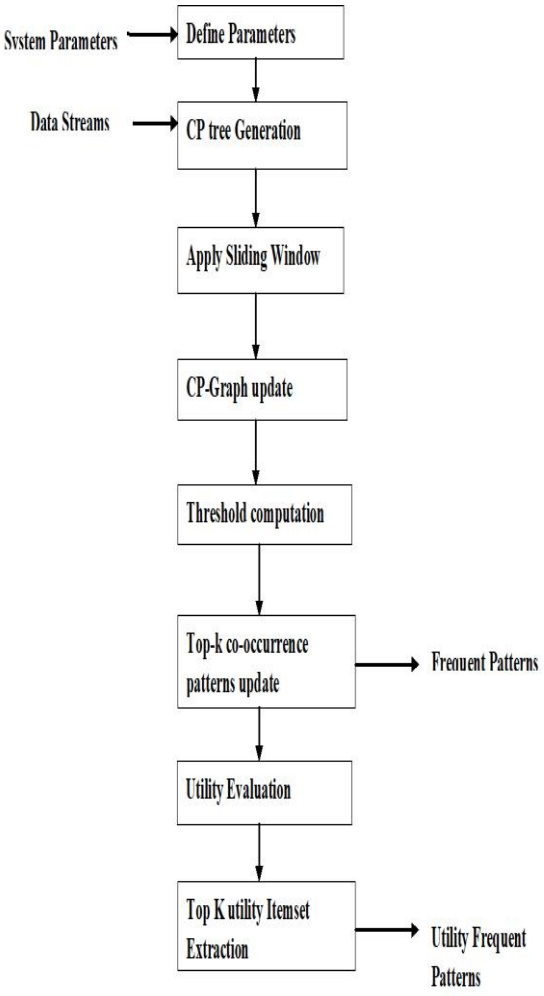

Fig 1: System Architecture

Following figure 1 represents the architecture of system.
The system works with multiple data streams. System read input streams and generates CP-Graph. Based on the cp graph, system finds set of tuples consist of closed co-occurrence patterns and its occurrence count. The utility value is evaluated for the filtered high co-occurrence count patterns and system generates high utility frequent itemset.

To read multiple data streams system follows the sliding window protocol. After receiving transaction slot after every sliding window, system performs following actions.

1. Index update: CP-graph is updated as per the new transaction entry. The expired transactions are deleted and accordingly CP-Graph is updated.
2. Mining top-k closed co-occurrence patterns: A set of closed co-occurrence patterns are extracted after every sliding window slot and accordingly the threshold U is updated
3. Mining top k Utility-frequent itemset: Based on the extracted closed co-occurrence patterns and utility values system extracts high utility itemset from frequent closed co-occurrence patterns.

CP graph:

This is undirected graph. It contains set of vertices and edges. This graph is generated based on the transaction dataset. Each vertex represents the item in transaction. Let vi and vj are two vertices in a graph. If any transaction in a dataset contains both items a and b in a consecutive manner then there is an edge between these two vertices.

With every vertex in a graph, a data structure is preserved. It contains a touple <Stream identifier S, flag, and inverted file structure>. Stream identifier represents the list of stream identifier in which item belongs, flag is set to 1 if the vertex is present in current sliding window else it is 0. An inverted file contains the frequent pattern identifiers list in which the vertex i.e. item belongs.

With the edge E ordered set of labels is preserved. It contains start point vertex, stream identifier and a time cycle.

Work Flow:

For every sliding window, number of transactions are read by the system. According to the items present in a transaction the CP graph is updated.

U(P) is the count of number of streams in which pattern P occurs. This count consider the single occurrence of a stream. For mining co-occurrence pattern among multiple streams the threshold is set to 2. i.e. the pattern should occur in more than 2 streams.

After inserting the items in a CP graph the minimum threshold T1 is calculated for top k frequent itemset based on the inverted file structure and edge labels. For calculating the threshold it initially sorts the pattern frequency . It checks for the co-occurrence pattern count among multiple stream and set. And top k items are extracted from the sorted set.

From the sorted list utility of each pattern is calculated based on the purchase quantity and product price. Utility of itemset is given as

$$\sum_{i=1}^{n} pi * ui$$

Where pi is purchase count of product I in temset and Ui is the unit price of an itemset. The frequent items having high utility

 DOI : 10.18231/2454-9150.2018.0449

value are extracted from a sorted list to generate frequent utility itemset.

V. ALGORITHMS

- **Insertion Algorithm:**

Input: CP graph,
Transastion T
Output: updated Cp graph:
Processing

1. Read itemset in transaction
2. For every item in dataset
3. If item not present in Graph then
   Initialize vertex with data structure
   Initialize edge with from previous node
4. If edge has label is already preserve with pattern path then update label
5. Else add label
6. Set flag = 1

- **Utility Frequent Itemset extraction:**

Input: D dataset
K: itemset count
Output: {fq1, fq2,..fqn} : frequent itemset
{ufq1,ufq2,..,ufqn} : utility frequent itemset
CP : co-occurrence pattern graph
Processing:

1. Generate data stream {S1, S2,..,Sn}
2. S1: Read first sliding window
3. V: Generate vertex set
4. E: Generate edge set
5. CP: Generate CP graph
6. Find itemset from graph
7. SF: Sort itemset with occurrence count
8. T: Find minimum threshold
9. Find utility of frequent items in sorted set SF
10. fq1: Find Top k frequent itemset
11. ufq1: Find Top k utility frequent itemset
12. Read Next sliding window i from S =2 to n
13. Update vertex set V
14. Update Edge set E
15. Update CP graph
16. Find latest itemset from graph
17. SF: Sort itemset with occurrence count
18. T: update minimum threshold
19. Find utility of frequent items in sorted set SF
20. fqi = update Top k frequent itemset
21. fqi = update Top k utility frequent itemset

VI. MATHEMATICAL MODEL

The system S can be defined as:
S= {I, O, F} where
I = {D, k,w}, Set of Inputs
D= Dataset
k= Top itemset count
w = Window Size
O = {CG,A,AU, T }, Set of Outputs
CG = CP-Graph
A= Top k Set of tuples consist of closed co-occurrence patterns
AU = Set of tuples consist of high utility closed co-occurrence patterns
T = Updated threshold
F= {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13}, Set of Functions
F1= Upload Dataset
F2 = Generate streams
F3 = Apply Sliding Window
F4 = Generate CP –Graph
F5 = Update CP graph
F6 = Find Patterns
F7 = Sort Pattern set
F8= Get Top k closed co-occurrence patterns
F9 = update Top k closed co-occurrence patterns
F10 = Update threshold
F11 = Calculate Utility
F12 = Get top k utility co-occurrence patterns
F13 = Update top k utility co-occurrence patterns

VII. IMPLEMENTATION

a) Experimental Setup:
The system is implemented using java environment on windows platform with 4 gb ram and i3 processor.

b) Dataset:
OnlineRetail dataset is downloaded from UCI repository[21]. This dataset contains transaction records in between 01/12/2010 and 09/12/2011. We have generated 100 different streams. The stream id is randomly assigned to the transaction record. For each transaction item, a unit price is randomly defined in between 1 to 10 and quantity of purchase for each transaction is randomly assigned between 1 to 5[6].

c) Performance Metric:

1. Processing Time : The time required for frequent itemset mining and utility frequent itemset mining is evaluated.
2. Itemset Similarity: The ratio of count of similar itemset extracted between frequent itemset and utility frequent itemset is extracted.

 DOI : 10.18231/2454-9150.2018.0449

VIII. RESULT ANALYSIS

The frequent itemsets are frequent utility itemset are extracted using existing and proposed system for data window size 2000,3000 and 4000 for BMS and Online retail dataset. The minimum support value of proposed system is less than the minimum support value of existing system to extract top k itemset.

Window Size	Min Support value for existing system[1]	Min Support value for Proposed system
20000	346	270
	434	355
	344	257
	302	253
30000	408	384
	407	328
	242	196
40000	720	530
	651	492

Table 1: Minimum support Analysis for BMS dataset

Window Size	Min Support value for existing system	Min Support value for Proposed system
2000	120	144
	534	180
	117	103
	105	133
	114	108
	2	2
3000	279	152
	164	175
	157	163
	65	67
4000	223	201
	219	217
	117	108

Table 2: Minimum support Analysis for Online Retail dataset

Following figure 2 and figure 3 shows the graphical analysis of existing and proposed system for different window sizes for BMS and online retail dataset respectively.

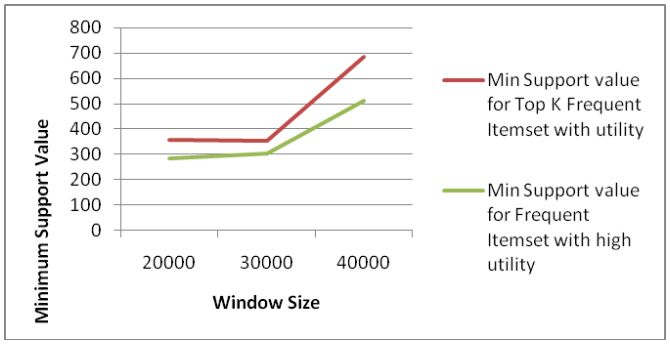

Fig 2: Minimum support Analysis for BMS dataset

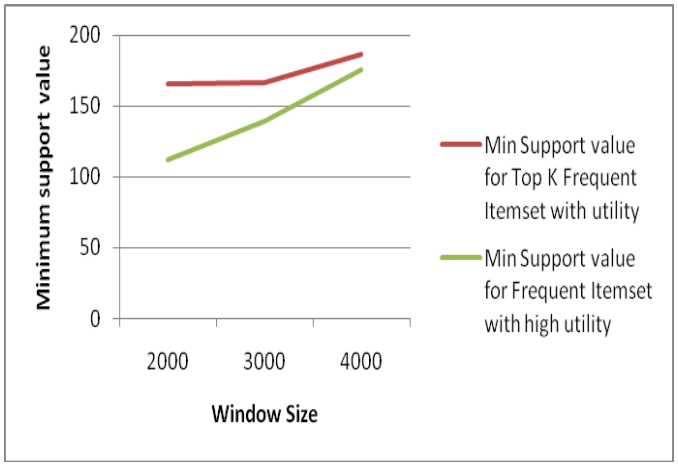

Fig 3: Minimum support Analysis for Inline Retail dataset

The number of top k utility frequent itemset count is input to the system. Based on the input value, existing system extracts less top k utility frequent items than the proposed system. Existing system extracts frequent items but all frequent items are not utility frequent. The proposed technique overcome this drawback and extracts all utility frequent items as per the user input. Following table 3 and 4 shows the results for BMS and online retail dataset.

Dataset	Top K Frequent Itemset count	Frequent Itemset with high utility (Existing system)	Frequent Itemset with high utility (Proposed system)
BMS2	25	19	25
BMS2	50	37	50
BMS2	75	63	75
BMS2	100	87	100

Table 3: Top k Frequent itemset Analysis for BMS dataset

Dataset	Top K count	Top K Frequent Itemset count	Frequent Itemset with high utility
online retail	25	25	19
online retail	50	50	47
online retail	75	75	51
online retail	100	100	58

Table 4: Top k Frequent itemset Analysis for Online Retail dataset

Following figure 3 and figure 4 shows the graphical representation of utility frequent items extraction from BMS dataset and online retail dataset respectively.

 DOI : 10.18231/2454-9150.2018.0449

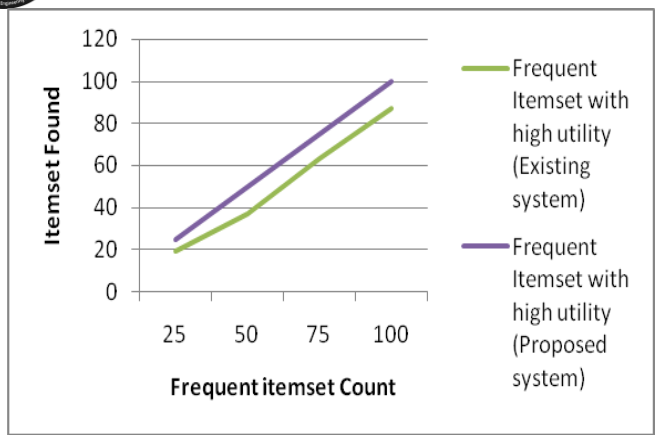

Fig 4: Top k Frequent itemset Analysis for BMS dataset

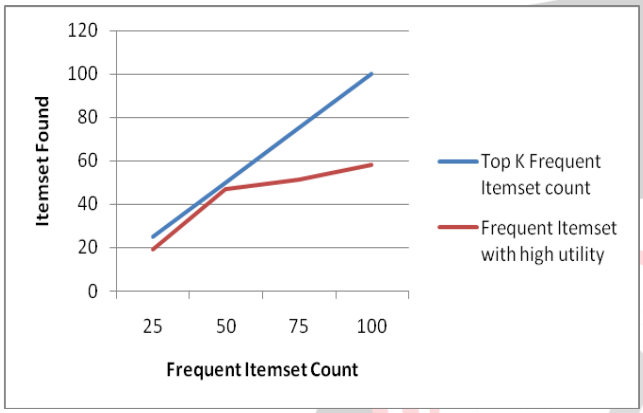

Fig 5: Top k Frequent itemset Analysis for Online Retail dataset

IX. CONCLUSION

Lot of application in current era generates continuous data. Some applications generate multiple data streams. The technique applied for single data stream analysis is not applicable to multiple data streams. The system works with multiple data streams. System read input streams and generates CP-Graph. Based on the cp graph, system finds set of tuples consist of closed co-occurrence patterns and its occurrence count. The utility value is evaluated for the filtered high co-occurrence count patterns and system generates high utility frequent itemset. In future, system can be implemented using distributed environment to improve the system performance.

REFERENCES

[1] Daichi Amagata, Takahiro Hara,"Mining Top-k Co-Occurrence Patterns across Multiple Streams", in IEEE Transactions on Knowledge and Data Engineering, Vol. 29, Issue 10, pp. 2249 - 2262, Oct 2017

[2] A. Giacometti, D. H. Li, P. Marcel, and A. Soulet, “20 years of pattern mining: a bibliometric survey,” ACM SIGKDD Explorations Newsletter, vol. 15, no. 1, pp. 41–50, 2014.

[3] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in VLDB, 1994, pp. 487–499

[4] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in SIGMOD, 2000, pp. 1–12.

[5] A. Salam and M. S. H. Khayal, “Mining top-k frequent patterns without minimum support threshold,” KIS, vol. 30, no. 1, pp. 57–86, 2012.

[6] Sen Su, Shengzhi Xu, Xiang Cheng, Zhengyi Li, and Fangchun Yang, "Differentially Private Frequent Itemset Mining via Transaction Splitting", in IEEE Transactions on Knowledge and Data Engineering,Vol 27, Issue 7, pp. 1875 - 1891, July 2015

[7] G. S. Manku and R. Motwani, “Approximate frequency counts over data streams,” in VLDB, 2002, pp. 346–357.

[8] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, “Mining frequent patterns in data streams at multiple time granularities,” Next generation data mining, vol. 212, pp. 191–212, 2003.

[9] H. Chen, L. Shu, J. Xia, and Q. Deng, “Mining frequent patterns in a varying-size sliding window of online transactional data streams,” Information Sciences, vol. 215, pp. 15–36, 2012.

[10] C. K.-S. Leung and Q. I. Khan, “Dstree: a tree structure for the mining of frequent sets from data streams,” in ICDM, 2006, pp. 928–932.

[11] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, “Sliding window-based frequent pattern mining over data streams,” Information sciences, vol. 179, no. 22, pp. 3843–3865, 2009.

[12] B. Mozafari, H. Thakkar, and C. Zaniolo, “Verifying and mining frequent patterns from large windows over data streams,” in ICDE, 2008, pp. 179–188.

[13] L. Troiano and G. Scibelli, “Mining frequent itemsets in data streams within a time horizon,” DKE, vol. 89, pp. 21–37, 2014.

[14] H.-F. Li, “Interactive mining of top-k frequent closed itemsets from data streams,” Expert Systems with Applications, vol. 36, no. 7, pp. 10 779–10 788, 2009.

[15] V. S. Tseng, C.-W. Wu, P. Fournier-Viger, and P. S. Yu, “Efficient algorithms for mining the concise and lossless representation of high utility itemsets,” TKDE, vol. 27, no. 3, pp. 726–739, 2015.

[16] V. S. Tseng, C.-W. Wu, P. Fournier-Viger, and P. S. Yu, “Efficient algorithms for mining top-k high utility itemsets,” TKDE, vol. 28, no. 1, pp. 54–67, 2016.

[17] Vid Podpecan, Nada Lavrac and Igor Kononenko, "A Fast Algorithm for Mining Utility-Frequent Itemsets", in academia.edu, 2007

[18] X. Zhu and X. Wu, “Discovering relational patterns across multiple databases,” in ICDE, 2007, pp. 726–735.

[19] Z. Yu, X. Yu, Y. Liu, W. Li, and J. Pei, “Mining frequent co-occurrence patterns across multiple data streams.” in EDBT, 2015, pp. 73–84.

https://archive.ics.uci.edu/ml/datasets.

 DOI : 10.18231/2454-9150.2018.0449