Mining Top-k Co-occurrence Patterns using MinRpset Algorithm and Thresholding Computation across Multiple Streams

[1] Archita B. Deore,
[1]PG Student (ME Computer),
[1]Department of Computer Engineering Gokhale Education Society's,
[1]R. H. Sapat College of Engineering Management Studies and Research, Nashik-05, India

Abstract : *This Big information and IoT has numerous applications. In each stream huge information and IoT are giving its commitment. When it comes to mining real-time mining from data streams supports many domains. Finding frequent pattern in continuous stream of transaction is difficult in applications like social network services, retail market web usage mining etc. and various algorithms are introduced for the same. Mining of item sets from data streams is difficult as computational complexity is high. Paper proposes CP- Graph can effectively compute the depend of a given pattern and update the reply at the same time pruning pointless patterns, a hybrid index of graph and inverted file structures. This paper has interested by the monitoring of co-occurrence of patterns. The drawback of Mining Top-k Co-occurrence Patterns in the course of multiple Streams is addressed by way of sliding window. In this each pattern is ranked established on the count and it is going to be dynamic. Dynamic nature may just exchange the rank of the count which is a task to monitoring the top-k answers in actual-time. Outcome exhibit the effect and scalability of the proposed approach. Thus we are achieving results of the paper as compare to the previous methods.*

Index Terms:–*–Top-kco-occurrence patterns, Multiple streams; CP-Graph.*

I. INTRODUCTION

In this era data streams have lured much attention towards it. The growing sets of streaming applications such as retail mar- ket[2], social network, network traffic monitoring[6], market basket data analysis, credit card fraud detection is the reason to it. In all these applications data mining plays vital role. Online stream mining has also increased. Frequent item sets are a problem in many domains [1] corresponding to bioinformatics [3], ecology [4], web click stream mining [5] and so on. From data streams mining frequent item sets have proved to be extremely difficult seeing that of calculate complexity as well as require for actual-time response. Consequently, mining frequent item set in excess of data streams has been widely deliberate [8], [9], [10], [11], [12], [13], [14], [15], [18], [19]. This This paper specializes in an environment of multiple streams, also and addresses the novel concern of continuous problem of mining of top-k closed co- occurrence patterns across multiple streams. Here co- occurrence pattern means identical group of objects appear repeatedly in multiple streams over small time span, signal tight correlations between this objects [18]. Applications such as web click stream mining [5], crime prevention [18] and so on generates objects and is involved in multiple streams. Objects can be the observations which have multiple occurrences in the different streams. Suppose S1, S2, S3 S|S|are the multiple streams generating consecutively transaction. Sliding window will maintain the transaction. µP is calculated from amount of streams to contain generated pattern P. Here, pattern means set of the objects. If µP≤2 also there is no p0i P such that µP=(µ(P0)), P is a co-occurrence pattern. Constant mining of k co-occurrence patterns through the maximum µbe able to support a lot of current applications. Examples:

Example.1 - ASSOCIATION MINING

Application id is used as identifier of the executed application in the smart phone. If we assume that tuples are generated by the smart phones which have information about location and executed application. Association rules can be mined between location and executed application from top-k co-occurrence tuples.

Example.2 - WEB USAGE PATTERN MINING

By mining co-occurrence patterns of click pattern or streams generated by multiple users, popular web sites and web search patterns of many users can be found.

Example.3 - LOCATION-BASED SERVICES

Again, smart phone users have check-in apps by which user check-in to the located places. Discover FCPs throughout the streams where stream consist of the check in locations by which groups can be discovered of persons hanging out together. This is useful in betterment of location-based advertising.The rest of this paper is organized as follows. In Section2, the related works are discussed. Section 3 describes the methodology. Section 4 presents the results of the proposed work, section 5 concludes the paper.

II LITERATURE SURVEY

In this section, we review some literatures that relate to our work. The current enlargement of importance in data stream systems and data stream mining is due to the reality that in many applications, data should be processed continuously, either because of real time necessities or because the stream is too very big for a store-now amp; process-later approach. Common patterns are summarizes in compact manner which reduces the database scans. Same kinds of problems were seen in mining top-k frequent closed item sets [5]. In Han et al. [20], came up with fp-tree which helps in storing transactions in compact with given a minimum threshold support. Then author has proposed an algorithm called FP-growth for mining an fp-tree where two passes are used over the window for finding frequent items and item sets. After this much work is done in this field. Chi et al. [21] has proposed the Moment algorithm for maintaining closed frequent item sets over slid- ing windows while Cats Tree [21] and Can Tree [22] support the incremental mining of frequent item sets. In[23] author has proposed hash-based counting whereas for efficiently counting item sets Brin et al. [24], proposed DIC (a dynamic algorithm) The pattern mining problem has been extensively studied during the past two decades, and literature introduces that more than a thousand papers study the pattern mining difficulty. In [25] author was the first to try to maintenance of frequent pattern mining on single stream. Error guaranteed answer and highly accurate are provided in[26]. Authors in [27], [28], [29] have worked with sliding window (single stream) on the problem of mining top-k patterns. In [28] author has maintained the number of transaction which consist of object for individual object while index [27] maintains in SWTP- tree[28] and closed patterns. Work is also done in common pattern mining on static data Agrawal et al. [30] has given the A priori algorithm which discovered the association rules in databases. A priori employs a BFS that enumerates all common patterns of size 1 and generates a candidate set of common patterns of size 2. Unless all common patterns are mined this process is repeated. When observation can be mined through single database mining frequent patterns from multiple databases is an option. In [31] there are common patterns that matches user specified conditioned are mined. The proposed method is not useful for our problem because the method considers ad-hoc queries over static databases. In [32] author has proposed H-Stream algorithm was proposed which provides an estimated answer, which is different from our objective.

III. METHODOLOGY

The A Co-occurrence Pattern Graph is proposed with an algorithm. CP-Graph sorts the data i.e. it removes the unimportant patterns and CP Graph is exploited by an algorithm which incrementally updates the answer. Objects in valid transaction are summarised by in memory index [16] [17]. Assume that a set of tuples where individual tuple consists of closed c- occurrence patterns answer be A and its count i.e. ≤P; P≥. For updating of A following steps are to be followed:

Step 1: (Index update) Updating means deleting the old and invalid transaction from CPG and inserting the new transaction.

Step 2: CP-Graph and answer is used for computing the highest estimated kth highest µnamwhich is threshold, as per the threshold A is updated.

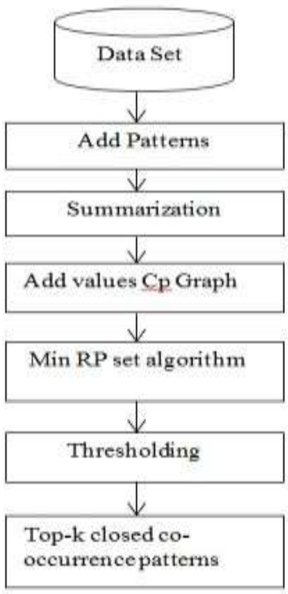

Fig. 1: System Architecture

A. CP-Graph

CP-Graph holds the responsibility of elimination and in-sertion so that the answer can be updated. Enumeration of the necessary closed co-occurrence patterns and computes their counts are the requirements which are satisfied by CP-Graph. Co-occurrence patterns Graph consists E and V, set of vertices is denoted by V (E) at the current time cyclenow. Valid transactions is regarded as a vertex v i and for representation of patterns on the window edge are created between vertices.

TABLE I: SYMBOLS RELATED TO CP-GRAPH

Symbols	Desciption
V	The set of vertices v of CP-Graph
vq	The vertex generated by oq
(j,c)	The tuple (A stream identifier, time-cycle)
v.S	The set of tuple (j,c) held by v
v.f	The boolean value held by v
v.I	The inverte file held by v
E	The set of edges e of CP-graph
ep,q	The edges between vp and vq
(P,j,c)	The label (a set of objects, a stream identifier, time-cycle)
(P,j,c)	The pointer to the label (P,j,c)
e,L	The ordered set of labels held by e

Two requirements must be satisfied for the creation of the edges. In a given valid transaction t =hj; Oi. Following are the requirements:

1) Two vertices of the corresponding objects that appear sequentially in O are connected through edges.
2) Corresponding vertices have edges between them, if there is a closed pattern in O. necessity of this requirement is for enumeration of closed co-occurrence patterns and traverse closed pattern directly.

B. CP-Graph update

Updating of CP-Graph requires insertion and deletion. CP- Graph needs to satisfy two requirements while creating edges. It is straight forward to satisfy the first requirement. CP graph have objects connected through edges which appear sequentially in a given transaction. Satisfying first requirement doesnt assure about satisfaction of second requirement. The problem is how to efficiently create necessary edges to satisfy the second requirement.

1) Insertion

For inclusion in Cp-Graph, while making edges, the CP-Graph needs to finish the two necessities, as pre- sented. The CP-Graph essentially makes edges be- tween objects (vertices) showing up successively in a given exchange. Be that as it may, fulfilling the primary necessity does not ensure fulfilling the second one. Let op, or be a shut example. As per operation or poten- tially may not show up successively in the legitimate exchanges. Inclusion calculated for a given arrangement of new exchanges at cnow. Given an exchange t = hi, Oi, created vertices and edges. Let or be the jth object in O. If vr ∈/ V , created vr. Then updated vr. S and set vr. f = 1. Here, assume that j ≥ 2, and let oq be the (j 1)th object in O. focused on θq,r. Created θq,r if θq,r

∈/ E. Let P be the object set consisting of the 1st to

the (j 1)th objects of O, and added the label hP, i,cnowh toθq,r ,L . Then, for∀ ∈ P, the pointer to h P, i, cnowi is added to vr.I(oq) . If j also satisfies j ≥ 3, confirmation of whether θq0∃ ,r ∈ E such that . If it is false, patterns P = op, ...,or on the window, where |P| ≥ 3, absolutely contain oq, so oq,r r is guaranteed to be a non-closed pattern[33].

2) Deletion

Each element of Ldel does not have whole information on the corresponding transaction then too deletion of the transaction is possible as e. L is an ordered set. By using property 5 we can identify the expired labels with c; v; o; f. In terms of memory usage this approach is considered as efficient approach as two identifiers are used for identification of expired labels. Deletion algorithm is as follows

Deletion Algorithm:

Given ≤Cnow,-w, vr, oq ,f≥ Ldel there can be three cases (i) oq =∅, in this case expired transaction contains single object o r hence we can set vr. S and set vr.f=1. (ii) f=0 here, eq,r has been created by the chain update and eq,r is deleted if eq,r.L=∅. In case (iii) the corresponding edges and vertices are traversed from vr, while updating vr.S and vr.f and deleting the expired labels of the edges. We have assumed op,...,oq,i,cnoww) be the rst label of eq,r.L.

C. MinRPset

MinRPset Algorithm Description:

1: Mine patterns with support less than or equal to (root)
2: DFS Search CXs (root);
3: Eliminate non-closed entry as of CXs;
4: To find representative patterns and output by applying the greedy set cover algorithm on CXs;

min sup(□)and store them in a CFP-tree; let root be the root node of the tree;

D. Threshold computation:

For prevention of the misuse of the authenticated nodes or from cheating the authenticated nodes the keys are dynamically changed. HMAC code is generated by using the same key. In this case attacker cannot fool the user and not be able to use previous keys[33].

E. Top-k co-occurrence patterns update

refine A by traversing the CP-Graph. Since had the apriori property and T, skip all vertices v where |v:S| ≤T. So as to be, specified a vertex v i where |vi:S| ≤T, updated A by enumerate co-occurrence patterns P = oi, , oj where (P) ≤T.

IV. RESULTS AND ANYLYSIS

F. Varying k

Index update time is not affected by k. Index update time of CP-Graph is more than that of Seg-tree. CP-Graph executes additional operation while Seg-tree cretaes edges between the vertices and objectss vertices in O. While in case of deletion, CP-Graph needs more index update time than Seg-tree because inverted file of every vertex is updated by CP-Graph. To compaire index updation time of proposed system with CP- Graph and Seg-tree is less.

G. Varying w

In chain-update of CP-Graph takes longer as it has to check multiple vertices, edges, and labels. When w is large, the numbers of them are also large which also affects longer deletion time. When total update time is seen of both the running time of Seg tree is slower than that of CP-Graph.

H. Varying Δ

Experiments are conducted on OnlineRetail. Δ is employed in count based on sliding window protocol

Since CP-Graph and Seg-tree update their indices for each transaction both CP- Graph and Seg-tree need linear index update time w.r.t Δ.

I. Mathematical Model:

Let S be the system such that

S=D,F,O

Where,

D = (D1,D2,...,Dm) : Data set,

F = (F1,F2,...,Fm) : Function set, O = Frequent Patterns

Input: D-≥ (D1,D2,...,Dm) : Retail Dataset

Output: FI where fi is the frequent itemsets which will be retrieve based on Thresholding and profit and filter using MINRP set algorithm.

Functions:

F = F1, F2, F3, F4,F5,F6

F1: Add Delete Records

F2:Dataset

F3: Summarization

A set of objects generated by streams is summarized by a graph-based structure, which is useful for representing patterns. Given a pattern foi; ojg, oi and oj are regarded as vertices (vi and vj) in the CP-Graph, and the CP-Graph has an edge between these vertices.

F4: Update CP-Graph

To update the answer quickly, it is desirable that we can efficiently enumerate necessary closed co-occurrence patterns and compute their counts.The CP-Graph satisfies these requirements, and consists of V and E, where V (E) denotes the set of vertices (edges) at the current time-cycle cnow.

F5: MinRPSet Algorithm

Mine patterns with support ≤ min sup and store them in a CFP-tree

F6: Top K Co-occurrence.

J. Performance Measures Used

We had compared our system performance based on speed up which in turn based on Update time it results computation of the counts of given patterns and a tight threshold, resulting in quick update. We will also compare our system based on closed utility patterns. Proposed system will provide most frequent item set compared to existing system.

K. Software requirements & specification:

TABLE II: Client Side

Software Required (Client)	Description
C# .Net	Windows Operating Environment

TABLE III: Server Side

Software Required (Server)	Description
OS	Windows
Dataset	Online Retail data sets

TABLE IV: Developer Side

Hardware Requirement	Description RAM
Minimum 4MB Hard Disk	
Minimum 500GB	
Processor	Minimum i3 processor

V. RESULTS

TABLE V: Comparison Table for Total Update time

K(online retail)	total updated time (msec) CP- Graph	Seg- Tree	Proposed	% speed up
10	1000	100	72	0.72
20	1012	118	79	0.79
30	1099	128	81	0.81
40	1100	136	85	0.85
50	1185	147	88	0.88
60	1197	151	90	0.90
70	1210	160	91	0.91
80	1266	174	93	0.93
90	1299	184	94	0.94
100	1326	190	96	0.96

$$\%SpeedUp = \frac{T_{seg} - T_{proposed}}{T_{seg\ proposed}} * 100 \qquad (1)$$

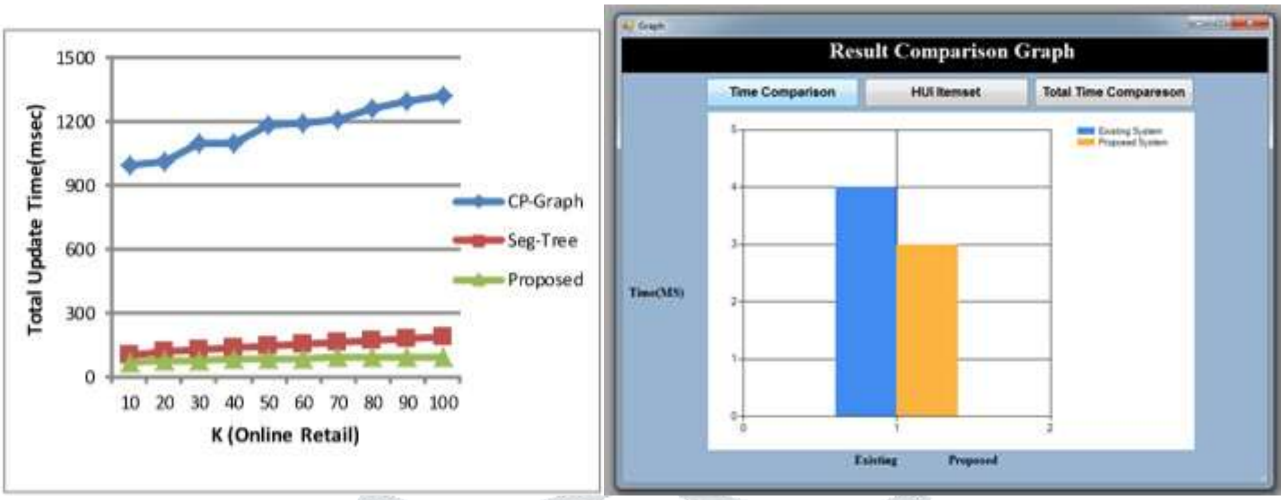

Fig. 2: Comparison Graph for Total Update Time (msec)

L. Efficiency calculation:

Total update time: the averaged running time for each slide of the window.

VI. CONCLUSION

This paper has centers around the trouble of mining top-k closed co-event designs transversely numerous streams. A strategy is proposed a technique which coordinates and rearranged le structures which figures the check and CP-Graph which gauges edge and updates the appropriate response. Results demonstrate the progressions caused by proposed strategy.

VII. REFERENCES

[1] 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.

[2] 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

[3] S. Su, S. Xu, X. Cheng, Z. Li, and F. Yang, Differentially private frequent itemset mining via transaction splitting, TKDE, vol. 27, no. 7, pp. 1875–1891, 2015.

[4] M. Celik, Partial spatio–temporal co–occurrence pattern mining, KIS,vol. 44, no. 1, pp. 27–49, 2015.

[5] A. Salam and M. S. H. Khayal, Mining top–k frequent patterns with- outminimum support threshold, KIS, vol. 30, no. 1, pp. 57–86, 2012.

[6] R. Gwadera and F. Crestani, Discovering significant patterns in multi- streamsequences, in ICDM, 2008, pp. 827832.

[7] V. E. Lee, R. Jin, and G. Agrawal, Frequent pattern mining in datas- treams, in Frequent Pattern Mining, 2014, pp. 199224.

[8] T. Calders, N. Dexters, J. J. Gillis, and B. Goethals, Mining frequen- titemsets in a stream, Information Systems, vol. 39, pp. 233255, 2014.

[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. 1536, 2012.

[10] M. Dallachiesa and T. Palpanas, Identifying streaming frequent items inad hoc time windows, DKE, vol. 87, pp. 6690, 2013.

[11] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, Mining frequent pat- terns in data streams at multiple time granularities, Next generationdata mining, vol. 212, pp. 191212, 2003.

[12] C. K.–S. Leung and Q. I. Khan, Dstree: a tree structure for the miningof frequent sets from data streams, in ICDM, 2006, pp. 928932.

[13] Y. Lim, J. Choi, and U. Kang, Fast, accurate, and space–efficient tracking of time–weighted frequent items from data streams, in CIKM, 2014, pp.11091118.

[14] B. Mozafari, H. Thakkar, and C. Zaniolo, Verifying and mining frequent patterns from large windows over data streams, in ICDE, 2008, pp.

179188.

[15] 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. 38433865, 2009.

[16] J. Guo, P. Zhang, J. Tan, and L. Guo, Mining frequent patterns across multiple data streams, in CIKM, 2011, pp. 23252328

[17] J. X. Yu, Z. Chong, H. Lu, Z. Zhang, and A. Zhou, A false negative approach to mining frequent itemsets from high speed transactional datastreams, Information Sciences, vol. 176, no. 14, pp. 19862015, 2006.

[18] Z. Yu, X. Yu, Y. Liu, W. Li, and J. Pei, Mining frequent co– occurren- cepatterns across multiple data streams. in EDBT, 2015, pp. 7384.

[19] L. Troiano and G. Scibelli, Mining frequent itemsets in data stream- swithin a time horizon, DKE, vol. 89, pp. 2137, 2014.

[20] J. Han, J. Pei, and Y. Yin, Mining frequent patterns without candidate generation, in SIGMOD, 2000.

[21] C.–S. Leung, Q. Khan, and T. Hoque, Cantree: A tree structure for efcient incremental mining of frequent patterns, in ICDM, 2005.

[22] W. Cheung and O. R. Zaiane, Incremental mining of frequent patterns without candidate generation or support, in DEAS, 2003. [Online]. Available: citeseer.ist.psu.edu/622682.html.

[23] J. S. Park, M.-S. Chen, and P. S. Yu, An effective hash-based algorithm for mining association rules, in SIGMOD, 1995, pp. 175186.

[24] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, Dynamic itemset counting and implication rules for market basket data, in SIGMOD,1997, pp. 255264

[25] G. S. Manku and R. Motwani, Approximate frequency counts over data streams, in VLDB, 2002, pp. 346357.

[26] R. C.–W. Wong and A. W.–C. Fu, Mining top–k frequent itemsets from data streams, Data Mining and Knowledge Discovery, vol. 13, no. 2, pp.193217, 2006.

[27] H.–F. Li, Interactive mining of top–k frequent closed itemsets from data streams, Expert Systems with Applications, vol. 36, no. 7, pp.1077910788, 2009.

[28] H. Chen, Mining top–k frequent patterns over data streams sliding window, Journal of Intelligent Information Systems, vol. 42, no. 1, pp.111131, 2014.

[29] P.S.Tsai,Miningtop–k frequent closed item sets over data streams using the sliding window model, Expert Systems with Applications, vol. 37, no. 10, pp. 69686973, 2010.

[30] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In VLDB, pages 487499, 1994.

[31] X. Zhu and X. Wu, Discovering relational patterns across multiple databases, in ICDE, 2007, pp. 726735.

[32] J. Guo, P. Zhang, J. Tan, and L. Guo, Mining frequent patterns across multiple data streams, in CIKM, 2011, pp. 23252328.

[33] Daichi Amagata,Takahiro Hara, Mining Top–k Co– occurrence Patterns across Multiple Streams IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. XX, NO. Y, ZZ 2017