International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Co-occurrence Patterns Extraction over Data Streaming

Miss Ankita Uday Manekar and Prof.Amar Chandgude

Computer Department SND COE RC, Babhulgaon, Yeola, Dist Nasik

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

Abstract

Lot of real life applications generates bulk data in streams. The analysis of real time data is required in variety of domains. The applications like market basket analysis requires the analysis of utility based co-occurrence patterns as well as frequency based co-occurrence patterns. In literature these techniques are studied on independently on data stream. The proposed work focuses on the mining of frequency based and utility based cooccurrence pattern extraction from multiple data streams. The system extracts top k patterns from multiple data streams. A sliding window protocol updates the top k co-occurrence patterns. For cooccurrence pattern extraction cp-graph, pattern utility table and inverted file structure is used. The system performance is tested on the basis of window size , processing time and memory required.

Keywords: *Co-occurrence patterns, utility patterns, top k itemset, streaming data, static data, multiple stream,cp-graph*

Introduction

In data mining, extracting interesting patterns in a dataset is important task. Co-occurrence patterns include itemset. There is no order of occurrence of item in a set. It only considers the existence in set. Co-occurrence pattern mining is useful in variety of domain such as retail market analysis, market basket analysis, intrusion detection, network monitoring, bioinformatics, web click analysis, etc.

In retail market analysis or in market basket analysis, transaction analysis is done. A transaction is nothing but an order placed by a customer. From such transaction dataset interesting occurrence patterns can be extracted. For example bread is frequently sold with butter rather than a chocolate.

Co-occurrence patterns are extracted with the help of predefined minimum purchase count. A user defined threshold value called as minimum support is required to find frequently occurred co-occurrence patterns. The minimum support values add the constraints on the extraction of patterns from a dataset. With the growing internet age, lots of applications such as facebook, twitter, online shopping carts are often used by users. Such applications generate lots of data continuously called as streaming data. Co-occurrence pattern mining is useful in such streaming data analysis.

In social network, a topic is extracted from matched keywords in multiple posts. Frequently occurred post or tweets defines the trend in market. Frequently co-occurrence patterns help to identify trends in a market. Where as in ECOMMERCE, frequently co-occurred products analysis helps to generate recommendation and also used in product promotional activities.

Frequently co-occurred data can be extracted from stream or from multiple streams. Frequently co-occurred patterns from single stream may not be always frequently occurred with multiple streams. The multiple streams can be defined for multiple sites for product sale or multiple regions based on user location. Co-occurrence pattern extraction over single and multiple stream data are important task in e-commerce data mining.

In multiple data stream analysis the whole data in all streams is merged together and analysis is done. Along with the minimum support value occurrence of pattern in multiple stream is also checked. The pattern is labeled as co-occurrence pattern if it occurred in more than 2 streams.

Association mining is again data mining task which uses coconcurrence patterns for finding items association. The statistical analysis of applications accessed using smart phones is one of the examples of association mining. The application analysis includes association discovery of application and its access time, location, user profile, etc.

Multiple algorithms for frequent co-occurrence pattern extraction are proposed in literature. apriory, Eclat, Fp-Growth are the algorithms for frequent pattern extraction over static dataset. The streaming data is continuously changing data. These algorithms can not directly applied to single or multiple streaming data. There is need to modify these algorithms or create a new strategy for streaming data analysis.

For Utility itemset extraction one phase and two phase algorithms are proposed to find itemset giving higher profit. The frequent itemset and utility itemset extraction are two important techniques in market analysis. Following section II includes an overview of literature work related to co-occurrence pattern extraction and utility pattern extraction techniques. Section III includes the analysis and problem formulation. Based on the problem definition a new system is proposed in section IV. The implementation details are mentioned in section V followed by the conclusion.

Literature Survey

P. Marcel, A. Giacometti, A. Marcel and D. H. Li[2] presented a survey paper for various techniques in pattern mining. This survey includes last 20 years work summary. The paper covers the 1087 different techniques for pattern mining. It includes pattern extraction and association rule discovery techniques.
The pattern mining algorithms are categorized in the following 6 sections based on the nature of dataset like static dataset, streaming data, multiple stream data, and analysis type like co-occurrence pattern extraction, utility pattern extraction, association rule extraction etc.

A. Pattern mining on static data:

In static dataset analysis the whole data is present before analysis in a database or in flat file. The complete data is provided for co-occurrence pattern extraction technique. The Apriori[3] and FP growth[4] are two widely used techniques for frequent co-occurrence pattern extraction.

Apriori algorithm is a two phase algorithm. In first phase it extracts the candidate results and in another phase it filters out the candidate and finds the final frequent co-occurrence patterns. For pattern extraction this algorithm uses breadthfirst search strategy to find next probable pattern[3].

To improve system efficiency this algorithm executes in one phase. This algorithm reduces the database scan and finds the final patterns without the candidate generation. This technique uses FP –tree for pattern extraction. FP tree uses divide-and-conquer strategy [4].

Mining frequent itemsets need minimum support value as an input. If the support value is less then too many frequent patterns are extracted and if the value is high then very few or no itemset can be extracted from the dataset. The nature of frequent patterns varies with respect to the nature of dataset. Hence there is no standard value for minimum support. To overcome this problem A. Salam and M. S. H. Khayal[5] proposes a technique to extract top k frequent patterns without predefined minimum support value. It dynamically adjusts the minimum support value based on the dataset. For itemset extraction it uses apriori algorithm.

Sen Su et. al added privacy to the dataset so that the original data should not be disclosed to the analyst. A privacy preserving frequent itemset extraction technique is proposed in literature. Differential privacy based frequent itemset[6] technique uses transaction splitting strategy and add some noise in original dataset to preserve original transaction dataset privacy.

B. Pattern Mining Over single Stream:

G. S. Manku and R. Motwani[7] proposes two algorithms named as Sticky Sampling algorithm and Lossy Counting algorithm for approximate co-occurrence pattern extraction over single stream data. This algorithm uses efficient memory utilization and can run on low configuration systems.

FP stream[8] is one more technique to find current frequent patterns over streaming data. This strategy also generates prediction for future pattern occurrence. This technique generates approximate result for future prediction. It uses pattern-tree-based structure.

Varying size window using sliding window protocol is one more technique is proposed. Apriori[9] and FP growth[10] algorithm are used to find frequent patterns over streaming data. These techniques update the data structure incrementally with respect to every sliding window batch.

Based on the FP growth algorithm, D-Tree[11] algorithm is proposed. This technique has few limitations such as efficiency in tree structure, storage overhead, etc. CPS-Tree [12]is the technique proposed to overcome these issues. CPStree provides Compact tree structure and hence reduces traversal and storage time.

Based on the streaming data mining top k pattern extraction is also proposed in literature. This techniques uses sliding window protocol[13].

C. Pattern mining across multiple databases:

X. Zhu and X. Wu propose a new technique for relational pattern extraction across multiple database. Rather than considering streaming data, this technique uses multiple static databases to find frequent co-occurrence patterns. This technique uses 2 threshold values as inter-frequency and intrafrequency for itemset filtration over multiple databases[14].

D. Frequent pattern mining across multiple streams:

Sagment-tree[15] is a technique proposed for analyzing frequent patterns over multiple data streams. Segment tree is used to finds patterns in transaction dataset. A CoolMine algorithm is proposed to travel the segment-tree. But the tree traversal technique is computationally expensive.

To overcome the drawback of CoolMine algorithms, a new technique CP graph is proposed. Graph

generation and traversal is used for multiple stream transaction data. This technique finds top k frequent itemset from multiple stream using closed co-occurrence patterns technique[1].

E. High utility itemset extraction:

All patterns extracted from frequent itemset extraction technique do not have high profit values. The frequent itemset may include low profit itemset. User has interested in finding high profit itemset in the transactional dataset. Apriori-based algorithmfor mining High utility is proposed[16]. It requires predefined minimum support value. 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 [17].

F. Utility frequent itemset extraction:

Utility patterns with frequent itemset extraction[18] is proposed in literature. This is called 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.

Problem Formulation

Lot of real life applications generates bulk data in stream. Multiple streams data is generated from same application based on the regional sale. The existing work includes variety of techniques such as single stream processing, multiple stream processing, and multiple database processing. Along with the frequent itemset user is interested in mining Utility itemset. Utility frequent patterns extraction is the best solution for any end user. Utility frequent patterns are extracted from static dataset and not from streaming data. There is need of such system that provides solution for utility based frequent cooccurrence pattern extraction technique for multiple data streams.

Proposed Methodology

The streaming data is input to the system. The streaming data contains 2 or more stream data. To read multiple data streams, system follows the sliding window protocol. System read the transaction records in the streaming data and generates the CP graph for first sliding window. For next sliding window, CP graph is updated. Based on the CP graph, system extracts the tuples containing 2 or more items occurred together. The touples represents closed co-occurrence patterns. Along with the patterns its occurrence count is also extracted. The utility value of each co-occurrence pattern is evaluated using following equation:

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

Where pi is the purchase count and ui is unit price of item I in a transaction. Base on the utility value, pattern with high utility value are extracted.

The system does not take minimum support value as an input. It dynamically computes the minimum support threshold value and finds the top k high utility co-occurrence patterns.

CP Graph:

This is undirected weighted graph. The graph is generated from transaction dataset. The distinct items presents in a dataset represents the vertices set. Let vi and vj be the two vertices in a dataset. The vertices vi and vj are connected in a CP graph if an only if i and j items co-occurred in a transaction. The edge weight represents the co-occurrence count.

Every vertex in a graph preserves the information in terms of touple. It contains stream identifier, i.e. number of streams in which it belongs. Flag represents the current occurrence of item in sliding window and inverted file structure that presents the list of co-occurrence items in which it belongs.

A. Architecture

Following figure shows the architecture of system. The multiple stream data is input to the system. Frequent itemsets and frequent utility itemsets are output of the system.

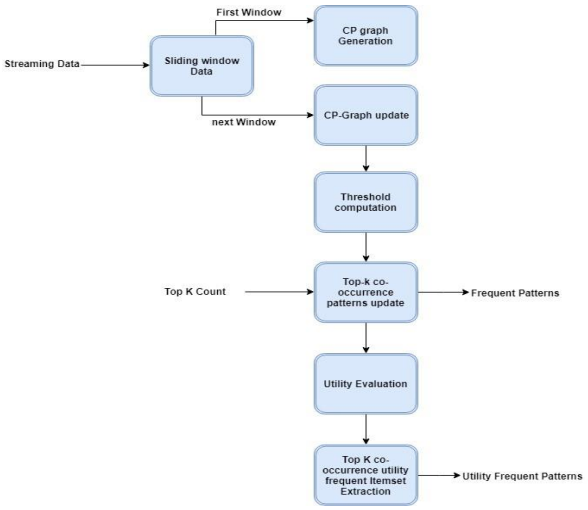

Fig 1: System Architecture

Algorithms

Utility Frequent Itemset extraction algorithm:
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. {S1, S2,..,Sn}: Generate n data streams
2. W1: Read first sliding window
3. V: Generate vertex set
4. E: Generate edge set
5. CP: Generate graph

6. Travel graph and 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. V: Update vertex set
14. E: Update Edge set
15. CP: Update 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

Result and Discussions

The system is implemented in java using jdk 1.8 on widows system with 4GB ram.

A. Datasets:

For testing transitional dataset is downloaded from UCI repository[19] and SPMF repository[20]. The online retail dataset contains transaction records occurring from 1-12-2010 to 9-12-2011. BMSWebView is the real click stream dataset. It contains records of seven months. A random stream id is assigned to each transaction record. The transaction id is generated between n0 to 99. For utility computation, the unit price is assigned to each item randomly between 1 to 10 and quantity of purchase is assigned randomly between 1 to 5[6]. Following table 1 represents the details of dataset:

Table I. Dataset

Sr. No.	Dataset	Number of Transaction	Number of Distinct Items
1.	Online Retail	541909	2603
2.	BMSWebView	77512	3340

B. Performance Measure:

1. Time: The processing time is captured for frequent itemset and utility frequent itemset extraction.
2. Memory: The memory required for processing is captured for frequent itemset and utility frequent itemset extraction.
3. Itemset Similarity: The ratio of number of elements present in frequent and utility frequent is extracted.

C. Implementation Status:

The system is implemented partially. The frequent itemset from multiple data streams are extracted. The following table 2 shows the results of BMS dataset. The results are taken by varying the top k value count. The memory required for processing and time required for processing is captured. As the value of top k count increases, the time required for processing is decreases and memory required for processing is increases.

Table II. BMS Dataset Evaluation

Top K count	Time For Execution(in Seconds)	Memory Required(in MB)
25	315.88	26.4278
50	297.71	49.3205
75	284.993	52.448
100	273.936	65.89335

System Comparison Following Chart shows the Comparative analysis of system with existing techniques.

	Frequent itemset Extractio n	Utility Itemset Extractio n	Single Strea m	Multipl e Streams
A New Algorithm for Frequent Itemsets Mining Based on Apriori and FPTree [5]	YES	-	-	-
Efficient Algorithms for Mining Top-K High Utility Itemsets[17]	-	YES	-	-
Mining Accurate Top-K Frequent Closed Itemset from Data Stream[16]	YES	-	YES	-
Mining Top-k CoOccurrence Patterns across Multiple Streams[1]	YES	-	YES	YES
Top-K Utility Frequent Itemset Mining from Multiple data Streams	YES	YES	YES	YES

Conclusions

Continuous data is generated in lot of applications. The continuous data is generated from single or multiple streams. The single stream data analysis technique is not applicable for multiple data stream analysis. The system provides a solution for analysis of data over

multiple streams. The system finds co-occurrence patterns using CP-graph the utility and frequency of patterns is calculated and top k utility frequent itemset are extracted. 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 201

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

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

[11]. 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.

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

[13]. 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.

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

[15]. 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.

[16]. 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.

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

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

[19]. Dataset:https://archive.ics.uci.edu/ml/datasets

[20]. Dataset: http://www.philippe-fournier-viger.com/spmf/index.php