IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 8, Issue 5 (Jan. - Feb. 2013), PP 79-84
www.iosrjournals.org

An Adjunct Quick Mining in Closed Persistent Patterns (AQCPP)

K.V.Subbaraju[1], RohiniVarma Pusapati[2]

[1]*(CSE Department, M.V.G.R. College of Engineering, India)*
[2]*(CSE Department, M.V.G.R. College of Engineering, India)*

Abstract: *Closed persistent patterns in data mining is biggest challenge in these days. The existing models find these patterns from traditional and transactional databases. This paper approach a novel model to mine closed persistent patterns using the invert matrix for any given traditional data file or for any raw sequential data and adjunct closed persistent element matrix which reduces the consumption of number of iterations or search space for retrieving various persistent patterns of deferent elements. We approach adjunct quick mining in closed persistent patterns (AQCPP) matrix to reduce the iterate levels i.e. the l and l+1 levels in mining conditions. Iteration time improvement is the clear output over the ancestors work in this process. Our experiments resulted efficient quick mining process throughout our analysis the algorithms performed efficiently with less computations. As a case study we also implemented our inline mechanism over Service Patterns to analyze the performance of the proposed AQCPP algorithm over lightweight directories and our results are satisfactory.*

Keywords:*AQCPP, Closed Persistent Patterns, Data mining, Invert Matrix, Service Patterns.*

I. Introduction:

Though the introduction of Sequential Pattern Mining received considerable attention with the rapid growth in the usage of the amount of huge stored digital data using data mining techniques among researchers with broad applications like bioinformatics, web access traces, system utilization logs, etc., the sequential pattern algorithms face uneasiness while mining long sequential patterns or while mining at low supported thresholds. One possible solution to overcome this issue is by mining closed persistent patterns.

A closed persistent pattern is said to be a narrowed sequential pattern which has no super sequence with the same happening frequency [1]. The main advantage of mining closed persistent patterns are, this can reduce the time and space cost when generating volatile numbers of frequent sequence patterns by mining only closed frequent subsequences (these closed sequences generally does not contain any super sequence with sample support) which results in avoiding unnecessary traversing search space.

However, there are not so many methods proposed in order to mine closed sequential patterns due to the problem of complexity, out of these Apriori [2], GSP [3], CloSpan [4] etc., algorithms are enforced in mining closed frequent item sets. All these algorithms follows a sustenance- and- test epitome, that is, these mechanisms needs to sustain a set of already mined closed sequences that can we further used to prune the search space to check if a newly found frequent set is promising to be closed one. But this sustenance may lead to poor scalability as a large number closed patterns will occupy large memory and search space which also effects the threshold measures.

In this paper we concentrated mining closed sequential patterns based on adjunct using inverted and adjunct based inline element matrices named Adjunct Quick Mining in Closed Persistent Patterns (AQCPP). This study proposes and feasible approach extending Quick mining in closed persistent patterns (QCPP) in order to find deferent inline patterns by reducing the scan/search space and minimizes number of scans of the original input. This mechanism consumes less memory and runs over an order of faster magnitude for mining closed persistent patterns especially when the support threshold is low.

II. Literature Review:

The Sequential pattern mining was first posed by Agarwal and Srikanth [4] and later many algorithms were proposed by many others for performance improvements. Initially the inventors Agarwal and Srikanth [4] refined their algorithm [4] by introducing Generalized Sequential Patterns (GSP) based on the based on implementing apriori dimensions [3].

Few other interesting algorithms on persistent pattern mining are SPADE [7], SPAM [8], PrefixSpan [9]. The SPADE uses lattice theoretic approach based on vertical ID list in order to optimize original search space into optimized smaller spaces. Later SPAM was introduces for mining long patterns this algorithm espouses a vertical bitmap representation and the performance scale shows that SPADE is more efficient than SPAM but still there is a flaw in this algorithm as it consumes more space compared with the SPAM algorithm. The other algorithm is the PrefixSpan algorithm which intern utilizes horizontal dataset format presentation and prefix based pattern growth mechanism for mining sequential data. But this algorithm shows less performance with increased scalability.

After the introduction of adopting the mining mechanisms of closed persistent patterns some of few interesting algorithms like CHARM [10], CLOSET [11], CLOSET+ [12], BIDE [13] and several other methods [14] were devised. But most of these algorithms were designed in order to sustain the already mined frequent patterns to proceed for pruning process among these in order to reduce the search space and memory, the algorithm TFP [15] was adopted for a compact hash levelled trees were devised in order to store the closest item sets. Later BIDE [13] was introduces in order to discover closed persistent patterns without maintaining candidate database.

III. Related Work:

The primary aim of mining or theory invention is to make utilize the pre available data to invent new points and to unhide the new groups that were with non-existing in the past with feasible solutions by utilizing the space and less time consumption. The repetitive item set mining plays a vital role in many mining process and applications, such as relational rules, sequential patterns and clustering. Frequent item set construction has been a major research area over the years and several techniques have been proposed in many papers to address the problem of mining correlation. PIM (persistent invert matrix) algorithm is widely classified into two classifications. The first classification is Apriori and other classification is Persistent Pattern growth.

Persistent pattern is a temporal domain and trend research for this study. The preexisting techniques in this temporal domain is totally based and grown Apriori candidate generation logic and suffer from duplicate input iterations setback. Adjunct span, a deferent of PP-growth is came into research for feasible inline pattern mining needs inputs to be in the form of tuples and does not have any track of temporal sequence across continuous pattern mines or executable tuples.

A feasible solution for persistent temporal pattern (FPTP) exposes the sequential pattern mining on one big inline, but have practical issues in duplicate scans of the first copy of inputs and iterates of the tree.

A sequence ***Sq0***closed if there is no master Sequence of ***Sq0*** with the same support in the main data file or raw data. So many inventions went in recent years to propose to mine nearest sequential patterns.
Mining sequential patterns with closed with closed patterns may reduce huge amount in number of patterns produced in the task and without loss of any data because it can be used to get all set of sequential patterns. Mining closed patterns may significantly reduce number of patterns produced is information lossless because it can be used to derive the complete set of closed patterns.

IV. Methodology:

4.1. Problem Definition:

This section discusses the basic concepts in inline pattern mining and the problem of mining repetitive closed inline patterns in an inline basic data. Let ***34*** be an input categorical sequence dataset and be a set of items in the dataset ***34***.

A sequence ***Sq0 = <s1, s2,….,sk>, (si Í I)*** for ***1≤i≤k***, is an ordered list. We assume that there exists a linear order in sequence ***Sq0***. A sequence ***Sq0***is closed *if* no *super sequences* of ***Sq0*** with the same support exist in the basic given data.

Given a minimum support threshold min_sup and an input categorical dataset ***34***, the task of prefix based mining of sequential patterns is to mine the complete set of frequent patterns based on prefix in the dataset ***34***.

Practical approach: Take the basic inline data like ***12342413421324341234*** in which the patterns to be mined and assume that the support count is 2. The persistent patterns of item 1 of this sequence would be for frequent 1-itemset = ***{1,2,3,4},*** frequent 2-itemset = ***{12,13, 14},*** frequent 3-itemset = ***{123, 124,132,134,143B}*** and frequent 4-itemset **=** ***{1234,1324, 1342}***.

4.2 Algorithms:

An inverted matrix approach to correlate each item with its inline segments those are sub in lines of the inline in which it occurs as an adjunct and to relate all items in each segment using pointers. Similar to the vertical approach in the transactional database representation, the item is the key of each record

In this layout. The difference between this approach and the vertical approach is that each attribute on the inverted matrix is not the transaction ID, but a pointer points to the location of the next item on the same inline segment. The construction of the inverted matrix is assumed to be pre-processing of the mining process. The inverted matrix that is made of two parts: the index and the inline array.

The index contains the items and the inline array is a set of rows in which each row is associated with one item in the index part. Each row is made of pairs representing pointers, where each pair holds two information: the physical address in the index part of the next item in the same sequence segment, and the physical address in the row of the next item in the same sequence segment. The entries of the sequence segments with items are done as follows: the given sequence is read, the location ofthe first item in the inline is sought and an entry to its inline array is added that holds the location of the next item in this inline. For the second time the same process occurs, in which an entry in the inverted matrix of the second item is added to hold the location of the third item in the sequence.

Table-1 represents inverted matrix for the given following data @1.
12342413421324341234*----*@1

Table-1: Inverse matrix generation of the above sequence

Index	**1**	**2**	**3**	**4**	**5**	**6**
1 – 1	**2,1**	**3,2**	**3,3**	**2,5**		
2 – 2	**3,1**	**4,2**	**1,3**	**4,4**	**3,5**	
3 – 3	**4,1**	**4,3**	**2,4**	**4,5**	**4,6**	
4 – 4	**2,2**	**1,2**	**2,3**	**3,4**	**1,4**	**$,$**

The proposed mechanism for **AQCPP** consists of three algorithms namely the creation of invert matrix, mining of close persistent patterns and adjunct patterns respectively. These three algorithms were developed to mine the closed persistent patterns in sequence database. The invert matrix is used to store the sub adjunct of adjunct levels having various elements as adjunct in the sequence the mining phase steps are described in 2 steps namely, step 1 and step 2 is given in algorithm2. The process of mining nearest patterns in described in algorithm 3.

Algorithm1: Generation of invert matrix:

Step 1: Monitor the input sequence and find the repeated item set for the given seed value (threshold value). Assign the sequence index value for repeated item and create the invert matrix with index of items in rows creation.
Step 2: Monitor the input sequence and seek for the current index in the inverted matrix and an entry to sequential data representation is appended the represents the index of the next element in that particular sequence.
Step 3: Do the above step repeatedly till end of the sequence is framed.

Algorithm 2: Mining of close persistent patterns:

Phase 1:
For each item in frequent 1- itemset
Step1: Get all all 2-item set having the item as adjunct.
Step2: Monitor the sequence levels relevant item in the inverted matrix using the index of the adjunct item in the item set and get the frequencies of 2-item set. Prune the items to ignore the non-frequent itemsets like Apriori.

Phase2:
For each item in K-Item set
Step1: Make a adjunct based sequence element matrix having itemsets as adjunct value and remaining items as rows and columns. The diagonal elements are the frequency of k+2 item set.
Step2: Monitor the sequence levels having the sequence element matrix adjunct as sequence adjunct in the inverted matrix, find the frequencies of k+1 – itemset and k+2 itemset and update entries in the regular matrix.
Step 3: Prun to remove non frequent itemsets before generating next level itemset and repeat the above steps in Phase-2.

For the sequence ***1234241342132434123***given in practical approach , the following adjunct based sequence element matrices with 12 as adjunct is first created using the frequent 2-itemset {12, 13,14} for patterns of item 1. The diagonal elements represent the frequency of 123 and 124 and non-diagonal represent the frequency of 123 and 124 and the non-diagonal elements represent the frequency of 1234 and 1243.

Table-2 represents Adjunct based sequence element matrix adjunct 12

12	**3**	**4**
3	3	3
4	1	4

Likewise the adjunct based sequence matrices are created for each itemset at level k in the mining process in order to find the frequent sequential patterns. The profit of the process is that the number of monitors is minimized by monitoring sequence elements at k and k+1 levels at the same time. That is the frequency of pattern 123 and 1234 are found using single monitor and same process is done for patterns 124 and 1243in a single monitor.

Alogorithm3: Mining of frequent closed adjunct patterns:

Step1: Classify the frequent sequential patterns using the transaction in which they appear.
Step2: Figure out their equivalence classes containing elements sharing the same supporting transactions.(For itemsets belong to the same equivalence class iff they are supported by the same set of transactions).
Step3: Find the persistent sequences which are more elements in each equivalent class.

Practical Approach2:
Let the transaction segments of item 1 be as shown in following table:

Table-3:Transaction level table for for item 1:

Level ID	Sequence
1	**1234**
2	**134**
3	**1324**
4	**1234**

The existence of continuous patterns with prefix 1 and support count as 2 in the transaction levels, are as follows:

12 = {1, 3, 4}
13 = {1, 2, 3, 4}
14 = {1, 2, 3, 4}
123 = {1, 4}
124 = {1, 3, 4}
134 = {1, 2, 3, 4}
1234 = {1, 4}

Table-4 the equivalence classes for the above patterns are shown in the following table:

Equivalence class	patterns
{1,4}	**123 , 1234**
{1,3,4}	**12 , 124**
{1,2,3,4}	**13 , 14, 134**

Here the maximal frequent item sets 1234, 124 and 134 identified from each equivalence class are closed and the maximal frequent closed item set is 1234.

V. Expermental Results:

This part gives us the reports of the performance testing of AQCPPwith splice dataset. The splice datasets is distributed as part of the UCI KDD archive. The data set encompasses a wide variety of data types, analysis, tasks and other application areas.

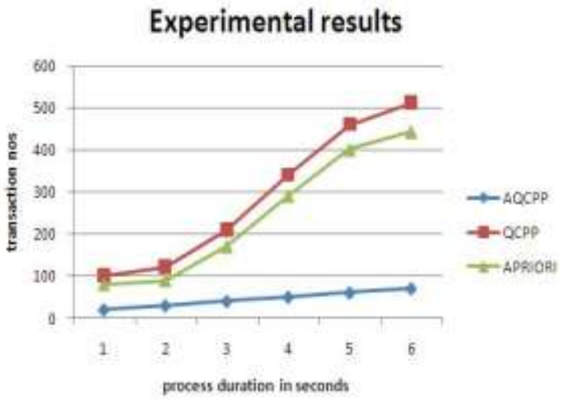

Figure-1: comparison analysis of proposed mechanism with other mining mechanisms

The above graph clearly proves that the proposed AQCPP mechanism mines the persistent pattern with less consuming process time when compared to Apriori and QCPP

5.1 A Case Study on Service Patterns:

In order to analyze the performance of our proposed mechanism, we deployed our proposed AQCPP mechanism over Service Patterns to analyze the performance for better quick mining methodology and compared our results with the other two algorithms namely QCPP which is the common prefix mining algorithm used by many data mining applications and APRIORI mining method.

Table-5: Comparative analysis over Service Patterns

AQCPP	QCPP	APRIORI	PP with Band width over service
15	10	5	5
25	15	10	4
35	20	13	8
45	25	16	6
55	30	18	8

The above table shows the comparative performance efficiency of the proposed one with the other two algorithms over service patterns.

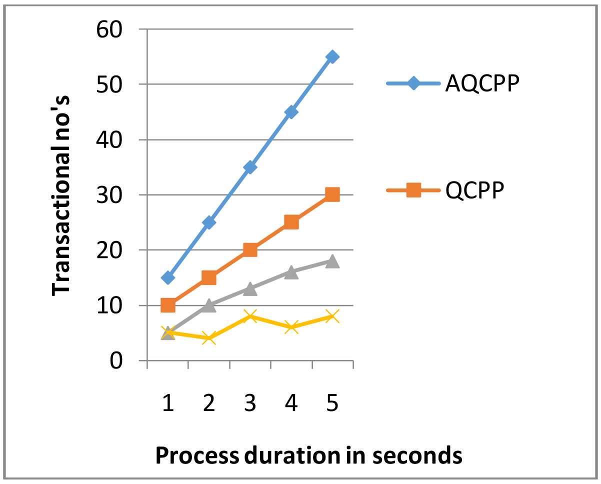

Figure-2: comparison analysis of proposed mechanism with other mining mechanisms over Service Patterns

VI. Conclusion:

In this paper we proposed an efficient pattern mining algorithm which performs quick mining over closed persistent patterns which performs well compared with the other experimental algorithms which follows a sustain and maintain mechanism in order to prune closed persistent patterns. Our inline pattern mining mechanism implementing Inverse matrix mechanism and the sequential mining algorithm reduces the performance complexity in order to figure out closest persistent patterns and also reduces the number of iterations compared with the others. Our offline and online (i.e. over service patterns) experimental results are satisfactory.

References:

[1] Shengnan Cong, Jiawei Han and David Padua, "Parallel Mining Of Closed Sequential Patterns", in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 562 – 567, Chicago, Illinois, USA, 2005.

[2] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. Int’l Conf. Very Large Data Bases (VLDB’94), pp. 487-499, Sept. 1994.

[3] R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements,” Proc. Int’l Conf. Extending Database Technology (EDBT ’96), pp. 3-17, Mar. 1996.

[4] X. Yan, J. Han, and R. Afshar, “CloSpan: Mining Closed Sequential Patterns in Large Databases,” Proc. SIAM Int’l Conf. Data Mining (SDM ’03), pp. 166-177, May 2003.

[5] R. Agrawal and R. Srikant, “Mining Sequential Patterns,” Proc. Int’l Conf. Data Eng. (ICDE ’95), pp. 3-14, Mar. 1995.

[7] M. Zaki, “SPADE: An Efficient Algorithm for Mining Frequent Sequences,” Machine Learning, vol. 42, pp. 31-60, 2001.

[8] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, “Sequential Pattern Mining Using a Bitmap Representation,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’02), pp. 429-435, July 2002.

[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, “Mining Sequential Patterns by Pattern- Growth: The PrefixSpan Approach,” IEEE Trans. Knowledge and Data Eng., vol. 16, no. 11, pp. 1424-1440, Nov. 2004.

[10] M. Zaki and C. Hsiao, “CHARM: An Efficient Algorithm for Closed Itemset Mining,” Proc. SIAM Int’l Conf. Data Mining (SDM’02), pp. 457-473, Apr. 2002.

[11] J. Pei, J. Han, and R. Mao, “CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets,” Proc. ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD ’00), pp. 21-30, May 2000.

[12] J. Wang, J. Han, and J. Pei, “CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD’03), pp. 236-245, Aug. 2003.

[13] J. Wang, J. Han, and Chun Li, "Frequent Closed Sequence Mining Without Candidate Maintenance", IEEE Transactions on Knowledge and Data Engineering, Vol.19, No.8, August 2007

[14] Proc. ICDM Workshop Frequent Itemset Mining Implementations(FIMI ’04), R.J. Bayardo Jr., B. Goethals, and M.J. Zaki, eds., Nov.2004.

[15] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, “Mining Top-K Frequent Closed Patterns without Minimum Support,” Proc. IEEE Int’l Conf. Data Mining (ICDM ’02), pp. 211-218, Dec. 2002.

[16] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent pattern tree approach. Data Mining and Knowledge Discovery, 2003.

[17] Mohammad El-Hajj and OsmarR.Zaiane, “Inverted Matrix: Efficient Discovery of Frequent Items in Large Datasets in the Context of Interactive Mining”, SIGKDD’03, August 24-27, 2003.