**Frequent Pattern mining: Dorsum edge pruning and logical analysis approach**

P. Dhana Lakshmi  
Assistant Professor  
Department of Computer Science Systems Engg.  
SEVC, A. Ragampet  
Email: dhanap_05@yahoo.com

Dr. B. Eswara Reddy  
Associate Professor & Head  
Department of Computer Science Engg.  
JNTUACE, Anantapur  
Email: eswarcsejntu@gmail.com

Abstract: Past observations have shown that a frequent item set mining algorithm are supposed to mine the closed ones as the end gives a compact and a complete progress set and better efficiency. Anyhow, the latest closed item set mining algorithms works with candidate maintenance combined with test paradigm which is expensive in runtime as well as space usage when support threshold is less or the item sets gets long. Here, we show, closed pattern mining algorithm, which is a capable algorithm used for mining closed sequences without candidate. It implements a novel sequence closure checking format that based on Dorsum edge pruning and logical analysis approach in short can refer as DEP&LA. A complete observation having sparse and dense real-life data sets proved that DEP&LA performs greater compared to older algorithms as it takes low memory and is faster than any algorithms those cited in literature frequently.

**Keywords:** Association Rule, edge Pruning, Qualified Support, Logical analysis Approach.

1. INTRODUCTION

Sequential item set mining, is an important task, having many applications with market, customer and web log analysis, item set discovery in protein sequences. Capable mining techniques are being observed extensively, including the general sequential item set mining [1, 2, 3, 4, 5, 6], constraint-based sequential item set mining [7, 8, 9], frequent episode mining [10], cyclic association rule mining [11], temporal relation mining [12], partial periodic pattern mining [13], and long sequential item set mining [14]. Recently it’s quite convincing that for mining frequent item sets, one should mine all the closed ones as the end leads to compact and complete result set having high efficiency [15, 16, 17, 18], unlike mining frequent item sets, there are less methods for mining closed sequential item sets. This is because of intensity of the problem and CloSpan is the only variety of algorithm [17], similar to the frequent closed item set mining algorithms, it follows a candidate maintenance-and-test paradigm, as it maintains a set of readily mined closed sequence candidates used to prune search space and verify whether a recently found frequent sequence is to be closed or not. Unluckily, a closed item set mining algorithm under this paradigm has bad scalability in the number of frequent closed item sets as many frequent closed item sets (or just candidates) consume memory and leading to high search space for the closure checking of recent item sets, which happens when the support threshold is less or the item sets gets long.

Finding a way to mine frequent closed sequences without the help of candidate maintenance seems to be difficult. Here, we show a solution leading to an algorithm, DEP&LA, which can mine efficiently all the sets of frequent closed sequences through a sequence graph protruding approach. In DEP&LA, we need not eye down on any historical frequent closed sequence for a new pattern’s closure checking, leading to the proposal of Sequence graph edge pruning technique and other kinds of optimization techniques.

The observations display the performance of the DEP&LA to find closed frequent item sets: The comparative study claims some interesting performance improvements over other frequently cited algorithms.

In section II most frequently cited work and their limits explained. In section III the Dataset adoption and formulation explained. In section IV, introduction to DEP&LA and its utilization for Sequence Graph protruding explained. In section V, the algorithms used in DEP&LA described. In section VI, results gained from a comparative study briefly and followed by conclusion of the study.

II. RELATED WORK

The sequential item set mining problem was initiated by Agrawal and Srikant, and the same developed a filtered algorithm, GSP [2], basing on the Apriori property [19]. Since then, lots of sequential item set mining algorithms are being developed for efficiency. Some are SPADE [4], PrefixSpan [5], and SPAM [6]. SPADE is on principle of vertical id-list format and it uses a lattice-theoretic method to decompose the search space into many tiny spaces, on the other hand PrefixSpan implements a horizontal format dataset representation and mines the sequential item sets with the pattern-growth paradigm: grow a prefix item set to attain longer sequential item sets on building and scanning its database. The SPADE and the PrefixSpan highly perform GSP. SPAM is a recent algorithm used for mining lengthy sequential item sets and implements a vertical bitmap representation.
Its observations reveal, SPAM is better efficient in mining long item sets compared to SPADE and PrefixSpan but, it still takes more space than SPADE and PrefixSpan. Since the frequent closed item set mining [15], many capable frequent closed item set mining algorithms are introduced, like A-Close [15], CLOSEST [20], CHARM [16], and CLOSET+ [18]. Many such algorithms are to maintain the ready mined frequent closed item sets to attain item set closure checking. To decrease the memory usage and search space for item set closure checking, two algorithms, TFP [21] and CLOSET+2, implement a compact 2-level hash indexed result-tree structure to keep the readily mined frequent closed item set candidates. Some pruning methods and item set closure verifying methods, initiated the can be extended for optimizing the mining of closed sequential item sets also. CloSpan is a new algorithm used for mining frequent closed sequences [17]. It goes by the candidate maintenance-and-test method: initially create a set of closed sequence candidates stored in a hash indexed result-tree structure and do post-pruning on it. It requires some pruning techniques such as Common Prefix and Backward Sub-Item set pruning to prune the search space as CloSpan requires maintaining the set of closed sequence candidates, it consumes much memory leading to heavy search space for item set closure checking when there are more frequent closed sequences. Because of which, it does not scale well the number of frequent closed sequences. BIDE [26] is another closed pattern mining algorithm and ranked high in performance when compared to other algorithms discussed. Bide projects the sequences after projection it prunes the patterns that are subsets of current patterns if and only if subset and superset contains same support required. But this model is opting to projection and pruning in sequential manner. This sequential approach sometimes turns to expensive when sequence length is considerably high. In [27] can find discussions on some other interesting works published in recent literature.

Here, we bring Sequence Graph protruding that based on edge projection and pruning, an asymmetric parallel algorithm for finding the set of frequent closed sequences. The giving of this paper is: (A) an improved sequence graph based idea is generated for mining closed sequences without candidate maintenance, termed as Dorsum Edge Pruning and Logical Analysis (DEP&LA) for closed item set mining, we create a algorithms for Forward edge projection and dorsum edge pruning(C) the performance clearly signifies that proposed model has a very high capacity: it can be faster than an order of magnitude of CloSpan but uses order(s) of magnitude less memory in several cases. It has a good scalability to the database size. When compared to Logic based pattern discovery [28] the model is proven as equivalent and efficient in an incremental way that proportional to increment in pattern length and data density.

III. DATASET ADOPTION AND FORMULATION
Item Sets I: A set of diverse elements by which the sequences generate.

\[ I = \bigcup_{i=1}^{n} j_i \]  \hspace{1cm} (1)

Note: ‘I’ is set of diverse elements

Sequence set ‘S’: A set of sequences, where each sequence contains elements each element ‘e’ belongs to ‘I’ and true for a function \( p(e) \). Sequence set can formulate as

\[ S = \bigcup_{j=1}^{m} (p(e_i), e_i \in I) \]  \hspace{1cm} (2)

Represents a sequence’s’ of items those belongs to set of distinct items ‘I’.

‘m’: total ordered items.

\( p(e_i) \): a transaction, where \( e_i \) usage is true for that transaction.

\[ S = \bigcup_{j=1}^{t} s_j \]  \hspace{1cm} (3)

S: represents set of sequences

‘t’: represents total number of sequences and its value is volatile

\( s_j \): is a sequence that belongs to \( S \)

Subsequence: a sequence \( s_p \) of sequence set ‘S’ is considered as subsequence of another sequence \( s_q \) of Sequence Set ‘S’ if all items in sequence \( s_p \) belongs to \( s_q \) as an ordered list. This can be formulated as

\[
\text{If } \left( \bigcup_{i=1}^{n} s_{pi} \in s_q \right) \Rightarrow (s_p \subseteq s_q)
\]
Then $\bigcup_{i=1}^{n}s_{pi} \leq \bigcup_{j=1}^{m}s_{qj}$ \hspace{1cm} (4)

Where $s_p \in S$ and $s_q \in S$

Total Support ‘ts’ : occurrence count of a sequence as an ordered list in all sequences in sequence set ‘S’ can adopt as total support ‘ts’ of that sequence. Total support ‘ts’ of a sequence can determine by following formulation.

$$f_{ts}(s) = \left| s_i < s_p, (foreachp = 1, |DB_s|) \right|$$ \hspace{1cm} (5)

$DB_S$ is set of sequences

$\forall i$, $f_{ts}(s_i)$ : Represents the total support ‘ts’ of sequence $s_i$ is the number of super sequences of $s_i$

Qualified support ‘qs’, The resultant coefficient of total support divides by size of sequence database adopt as qualified support ‘qs’. Qualified support can be found by using following formulation.

$$f_{qs}(s_i) = \frac{f_{ts}(s_i)}{|DB_s|}$$ \hspace{1cm} (6)

Sub-sequence and Super-sequence: A sequence is sub sequence for its next projected sequence if both sequences having same total support.

Super-sequence: A sequence is a super sequence for a sequence from which that projected, if both having same total support.

Sub-sequence and super-sequence can be formulated as

If $f_{ts}(s_i) \geq rs$ where ‘rs’ is required support threshold given by user

And $s_i < s_p$ for any $p$ value where

$\forall i$, $f_{ts}(s_i) \equiv f_{ts}(s_p)$

IV. DORSUM EDGE PRUNING AND LOGICAL ANALYSIS

A. Pre-process

As a first stage of the proposal we perform dataset pre-processing and item sets database initialization. We find item sets with single element, in parallel prunes item sets with single element those contains total support less than required support.

B. Forward Edge Projection:

In this phase, we select all item sets from given item set database as input in parallel. Then we start projecting edges from each selected item set to all possible elements. The first iteration includes the pruning process in parallel; from second iteration onwards this pruning is not required. In first iteration, we project an item set $s_p$ that spawned from selected item set $s_j$ from $DB_S$ and an element $e_i$ considered from ‘I’. If the $f_{ts}(s_p)$ is greater or equal to $rs$, then an edge will be defined between $s_i$ and $e_i$. If $f_{ts}(s_i) \equiv f_{ts}(s_p)$ then we prune $s_i$ from $DB_S$. This pruning process required and limited to first iteration only.

From second iteration onwards project the item set $S_p$ that spawned from $S_p$ to each element $e_i$ of ‘I’. An edge can be defined between $S_p$ and $e_i$ if $f_{ts}(s_p)$ is greater or equal to $rs$. In this description $S_p$ is a projected item set in previous iteration and eligible as a sequence. Then apply the following validation to find closed sequence.

If any of $f_{ts}(s_p) \equiv f_{ts}(s_p)$ that edge will be pruned and all disjoint graphs except $s_p$ will be considered as closed sequence and moves it into $DB_S$ and remove all disjoint graphs from memory.

If $f_{ts}(s_p) \equiv f_{ts}(s_p)$ and there after no projection spawned then $s_p$ will be considered as closed sequence and moves it into $DB_S$ and remove $s_p$ from memory.

The above process continues till the elements available in memory those are connected through direct or transitive edges and projecting item sets i.e., till graph become empty.

V. ALGORITHMS USED IN DEP&LA

This section describes algorithms for initializing sequence database with single elements sequences; spawning item set projections and pruning edges from Sequence Graph SG.

Algorithm 1: Generate initial $DB_S$ with single element item sets

Input: Set of Elements ‘I’.

47
Algorithm 2: Projecting Item sets algorithm

Input: $DB_s$ and ‘I’;

L1: For each sequence $s_i$ in $DB_s$
Begin:
L2: For each element $e_i$ of ‘I’
Begin:
C1: if $\text{edgeWeight}(s_i, e_i) \geq rs$
Begin:
Create projected item set $s_p$ from $(s_i, e_i)$
If $f_{ix}(s_i) \geq f_{ix}(s_p)$ then prune $s_i$ from $DB_s$
End: C1.
End: L2.
End: L1.

L3: For each projected Item set $s_p$ in memory
Begin:
$s_p^r = s_p$
End: L3.

L4: For each $e_i$ of ‘I’
Begin:
Project $s_p$ from $(s_p^r, e_i)$
End: L4.

C2: If $f_{ix}(s_p^r) \geq rs$
Begin
Spawn SG by adding edge between $s_p^r$ and $e_i$
End: C2.
End: L4.

C3: If $s_p^r$ not spawned and no new projections added
Begin
Remove all duplicate edges for each edge weight from $s_p^r$ and keep edges unique by not deleting most recent edges for each edge weight.
Select elements from each disjoint graph as closed sequence and add it to $DB_s$ and remove disjoint graphs from SG.
End C3
End: L3.

If $SG \neq \emptyset$ go to L3.

(a) First iteration
VI. DESCRIPTION OF LOGICAL ANALYSIS

Set I = \{i_1, i_2, ..., i_m\} be the universe of items composed of m different attributes, i_k (k=1,2,...,m) is item. Transaction database D is a collection of transactions. A transaction t = (tid, X) is a tuple where tid is a unique transaction ID and X is an item set. The count of an item set X in D, denoted by count(X), is the number of transactions in D containing X. The support of an item set X in D, denoted by supp(X), is the proportion of transactions in D that contain X. The negative rule \( X \Rightarrow \neg Y \) holds in the transaction set D with confidence conf \( (X \Rightarrow \neg Y) = \frac{\text{supp}(X \cup \neg Y)}{\text{supp}(X)} \).

In Transaction database, each transaction is a collection of items involved sequences. The issue of mining association rules is to get all association rules that its support and confidence is respectively greater than the minimum threshold given by the user.

The issues of mining association rules can be divide into two sub-issues as follows:

a). Find frequent item sets; generate all item sets that support is greater than the minimum support;

b). Generate association rules from frequent item sets.

In logical analysis, the direct calculation of support logical analysis is not convenient, to calculate the support and confidence of negative associations using the support and confidence of positive association that is known: set A, B\( \subset I \), \( A \cap B = \emptyset \) then:

\[
\text{sup}(\neg A) = 1 - \text{sup}(A);
\]

\[
\text{sup}(A \cup \neg B) = \text{sup}(A) - \text{Sup}(A \cup B);
\]

\[
\text{sup}(\neg A \cup B) = \text{sup}(B) - \text{sup}(A \cup B)
\]

\[
\text{sup}(\neg A \cup \neg B) = 1 - \text{sup}(A) - \text{sup}(B) + \text{sup}(A \cup B);
\]

Based on the above formulas we perform the logical analysis to derive the actual support of the patterns that improves the rule coherency.

Description of the logical analysis:

Let A, B\( \in I \) where I is item set generated with the association of A, B are individual items or subsets.

Let actual support of I is s1;

Under logical analysis we determine support of \( \neg A \cup \neg B \) as s2, support of \( A \cup \neg B \) as s3 and support of \( \neg A \cup B \) is s4.

The support s1, s2 we consider as positive support and s3, s4 we consider as negative support.

Finally we determine actual support \( s = s_1 + s_2 - s_3 - s_4; \)

VII. COMPARATIVE STUDY

This segment focuses mainly on providing evidence on asserting the claimed assumptions that 1) The DEP&LA is actually a sealed series mining algorithm that is competent enough to momentarily surpass results when evaluated against other algorithms such as CloSpan, spade and logic based pattern discovery(LBPD) [28]. 2) Utilization of memory and momentum is rapid when compared to the Logic based pattern discovery (LBPD) [28] algorithm. 3) There is the involvement of an enhanced occurrence and a probability reduction in the memory exploitation rate with the aid of the trait equivalent prognosis and also rim snipping of the DEP&LA. This is on the basis of the surveillance done which concludes that DEP&LA’s implementation is far more noteworthy and important in contrast with the likes of LBPD, to be precise.

JAVA 1.6_ 20th build was employed for accomplishment of the DEP&LA and LBPD algorithms. A workstation equipped with core2duo processor, 2GB RAM and Windows XP installation was made use of for investigation of the algorithms. The parallel replica was deployed to attain the thread concept in JAVA.

Dataset Characteristics
Pi is supposedly found to be a very opaque dataset, which assists in excavating enormous quantity of recurring clogged series with a profitably high threshold somewhere close to 90%. It also has a distinct element of being enclosed with 190 protein series and 21 divergent objects. Reviewing of serviceable legacy’s consistency has been made use of by this dataset. Figure 5 portrays an image depicting dataset series extent status.

In assessment with all the other regularly quoted forms like spade, prefixspan and CloSpan, LBPD has made its mark as a most preferable, superior and sealed example of mining copy, taking in view the detailed study of the factors mainly, memory consumption and runtime, judging with DEP&LA.

In contrast to DEP&LA and LBPD, a very intense dataset Pi is used which has petite recurrent closed series whose end to end distance is less than 10, even in the instance of high support amounting to around 90%. The diagrammatic representation displayed in Figure 3 explains that the above mentioned two algorithms execute in a similar fashion in case of support being 90% and above. But in situations when the support case is 88% and less, then the act of DEP&LA surpasses LBPD’s routine. The disparity in memory exploitation of DEP&LA and LBPD can be clearly observed because of the consumption level of DEP&LA being low than that of LBPD.

Figure 5: Sequence length and number of sequences at different thresholds in Pi dataset

VIII. CONCLUSION

It has been scientifically and experimentally proved that clogged prototype mining propels dense product set and considerably enhanced competency as compared to recurrent prototype of mining even though both these types project similar animated power. The detailed study has verified that the case usually holds true when the count of recurrent moulds is considerably large and is the same with the recurrent bordered models as well. However, there is the downbeat in which the earlier formed clogged mining algorithms depend on chronological set of recurrent mining outlines. It is used to verify whether an innovative recurrent outline is blocked or else if it can nullify few previously mined blocked patterns. This leads to a situation where the memory utilization is considerably high but also leads to inadequacy of increasing seek out space for outline closure inspection.

This paper anticipates an unusual algorithm for withdrawing recurring closed series with the help of Sequence Graph. It performs the following functions: It shuns the bight of contender’s maintenance and test exemplar, supervises memory space expertly and ensures recurrent closure of clogging in a well-organized manner and at the same
instant guzzling less amount of memory plot in comparison with the earlier developed mining algorithms. There is no necessity of preserving the already defined set of blocked recurrences; hence it very well balances the range of the count of frequent clogged models. A Sequence graph is embraced by DEP&LA and has the capability of harvesting the recurrent clogged pattern in an online approach. The efficacy of dataset drafts can be showcased by a wide-spread range of experimentation on a number of authentic datasets amassing varied allocation attributes. DEP&LA is rich in terms of velocity and memory spacing in comparison with the LBPD and CloSpan algorithms. On the basis of the amount of progressions, linear scalability is provided. It has been proven and verified by many scientific research studies that limitations are crucial for a number of chronological outlined mining algorithms. Future studies include proposing of claiming a deduction advance on perking up the rule coherency on predictable item sets.

ACKNOWLEDGEMENT

The authors are thankful to the Department of CSE, JNTUACE University for providing the necessary facilities.

REFERENCES


Dr B.Eswara Reddy was born at T.R.Puram, Chittoor(Dt), A.P, India on 20th May, 1972. He earned B.Tech (CSE) from S.K.University, Anantapur, A.P, India in 1995 and M.Tech(SE) from IPGSR, JNT University, Hyderabad in 1999. He received Ph.D (CSE) degree in 2008. His major areas of research includes Image
He served as Assistant Professor from 1996 to 2006. He is working as Associate Professor in CSE Dept., since 2006 and currently acting Head of CSE Dept. at Jawaharlal Nehru Technological University College of Engineering, Anantapur, A.P, India. He is one of the author’s of the text book titled ‘Programming with the Java’ (Pearson Publishers, 2011). He has published more than 30 research papers in various national and International Conferences and Journals.

Dr Reddy is life member of ISTE, IE(I), ISCA and member of CSI and IEEE.

Place: Chandragiri, Chittoor, Andhra Pradesh, India.

She is currently working as a Assistant Professor in CSSE Dept, SVEC, A. Ragam pet.

Membership: Seminar & Conference attended (National & International): 03
She is having Membership in ISTE.