ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Parallel Edge Projection and Pruning (PEPP) Based Sequence Graph Protrude Approach for Closed Item Set Mining

CH.Shanthi
M.Tech In Software Engineering
Department of CSE
Aurora Technological And Research Institute
Hyderabad, Telangana.

N.Nirmala Jyothi
Senior Assistant Professor
Department of CSE
Aurora Technological And Research Institute
Hyderabad, Telangana.

Abstract:

Recent observations have revealed that a frequent item set mining algorithm be supposed to mine the congested ones as the end gives a condensed and a complete evolution set and better efficiency. Anyway, the latest closed item set mining algorithms mechanism with candidate protection combined by means of test paradigm which is expensive in runtime as well as space procedure when sustain threshold is less or the item sets gets extended. Here, we show, PEPP, which is a capable algorithm used for mining closed sequences without candidate. It apparatus a novel succession closure checking format that based on Sequence Graph protruding by an approach labeled "Parallel Edge projection and pruning" in short can refer as PEPP. A complete surveillance having dense and dense real-life data sets prove that PEPP achieve greater evaluate to older algorithms as it obtain low memory and is faster than any algorithms those cited in prose frequently.

Introduction

Sequential item set mining, is an important task, having many applications with market, customer and web log examination, item set find in protein succession. Capable mining techniques are being observed extensively, including the general sequential item set mining [1, 2, 3, 4, 5, 6], constraint-based in order item set mining [7, 8, 9], frequent occurrence mining [10], cyclic association rule mining [11], sequential relation mining [12], partial episodic pattern mining [13], and long in order 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 trail a candidate maintenance-and-test paradigm, as it maintains a set of readily supply blocked 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 below this paradigm has bad scalability in the quantity of frequent blocked item sets as many frequent closed item sets (or just candidates) consume memory and leading to high search break for the closure checking of recent item sets, which happens when the holdup threshold is less or the item sets gets extended.

Finding a technique to extract frequent closed sequences lacking the help of candidate preservation seems to be complex. Here, we show a solution leading to an algorithm, PEPP, which can mine efficiently all the sets of frequent closed sequences through a sequence graph protruding approach. In PEPP, 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 Graph pruning technique and other kinds of optimization techniques.

The observations display the performance of the PEPP to find closed frequent itemsets using Sequence Graph:

The comparative study claims some interesting performance improvements over BIDE and 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 PEPP and its utilization for Sequence Graph protruding explained. In section V, the algorithms used in PEPP described. In section V1, results gained from a comparative study briefed and fallowed by conclusion of the study.

Related Work

The sequential item set mining difficulty was initiated by Agrawal and Srikant , and the same urbanized a filtered algorithm, GSP [2], basing on the Apriori assets [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 configure and it uses a lattice-theoretic method to fester the search space into many tiny places, on the other hand PrefixSpan implements a parallel 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], CLOSET [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 seek out space for item set closure examination, 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 our earlier literature[27] we discussed some other interesting works published in recent literature.

Here, we bring Sequence Graph protruding that based on Graph projection and pruning, an asymmetric parallel algorithm for finding the set of frequent closed succession. The giving of this paper is: (A) an improved sequence graph based idea is generated for mining closed sequences without candidate maintenance, termed as Parallel Edge Projection and pruning (PEPP) based Sequence Graph Protruding for closed itemset mining. The Graph Projection is a forward approach grows till Graph with required support is possible during that time the Graphs will be pruned. During this pruning process vertices of the Graph that differs in support with next Graph projected will be considered as closed itemset, also the sequence of vertices that connected by Graphs with

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

similar support and no projection possible also be considered as closed itemset (B) in the Graph Projection and pruning pedestal Sequence Graph Protruding for closed itemset mining, we create a algorithms for Forward Graph projection and back Graph 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 BIDE the model is proven as equivalent and efficient in an incremental way that proportional to increment in pattern length and data density.

Dataset adoption and formulation

Item Sets I: A position of diverse basics by which the sequences produce.

$$I = \bigcup_{k=1}^{n} i_k$$

Note: 'I' is set of diverse essentials

Sequence set 'S': A position of sequences, where both sequence contains basics each element 'e' belongs to 'I' and accurate for a function p(e). Sequence set can prepare as

$$s = \bigcup_{i=1}^{m} < e_i \,|\, (p(e_i), e_i \in I) >$$

Symbolize a sequence's' of items those belong to set of dissimilar items 'I'.

'm': total controlled items.

$P(e_i)$: a contract, where e_i usage is true for that operation.

$$S = \bigcup_{j=1}^{t} s_j$$

S: characterize set of sequences

't': signify total number of sequences and its assessment is volatile

s_j: is a sequence that belong to S

Subsequence: a sequence s_p of progression set 'S' is measured as subsequence of an additional sequence s_q of Sequence Set 'S' if all items in progression S_p is belongs to s_q as an controlled list. This can be prepare as

$$\text{If} \quad (\bigcup_{i=1}^{n} s_{pi} \in s_q) \Rightarrow (s_p \subseteq s_q)$$

$$\text{Then} \quad \bigcup_{i=1}^{n} s_{pi} <: \bigcup_{j=1}^{m} s_{qj} \quad \text{where} \quad s_p \in S \; and \; s_q \in S$$

Total sustain 'ts' : happening count of a sequence as an controlled list in all sequences in sequence set 'S' can assume as total support 'ts' of that progression. Total sustain 'ts' of a sequence can establish by subsequent formulation.

$$f_{ts}(s_t) = | s_t <: s_p \;\; (for\, each\, p = 1.. | DB_S |) |$$

DB_S Is position of sequences

$f_{ts}(s_t)$: Represents the total sustain 'ts' of progression s_t is the number of super sequences of s_t

Practiced support 'q_s': The consequential coefficient of total sustains divides by size of progression database assume as qualified hold up 'qs'. Qualified hold up can be initiates by using fallowing formulation.

$$f_{qs}(s_t) = \frac{f_{ts}(s_t)}{| DB_S |}$$

Sub-sequence and Super-sequence: A progression is sub progression for its next predictable sequence if equally sequences enclose same total sustain.

Super-sequence: A progression is a fabulous sequence for a succession from which that predictable, if both enclose same total support.

Sub-sequence and super-sequence can be create as

If $f_{ts}(s_t) \geq rs$ where 'rs' is necessary support threshold specified by user

And $s_t <: s_p \; for\, any\, p\, value$ where

$$f_{ts}(s_t) \cong f_{ts}(s_p)$$

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Parallel Edge Projection and Pruning Based Sequence Graph protrude

Preprocess

As a first stage of the offer we achieve dataset preprocessing and itemsets Database initialization. We find itemsets with single element; in parallel prunes itemsets with single element those contains total support less than essential support.

Forward Graph Projection:

In this segment, we choose all itemsets from given itemset database as input in equivalent. Then we establish projecting Graphs starting each preferred itemset to all achievable elements. The foremost iteration includes the pruning progression in parallel, from second iteration onwards this pruning is not necessary, which we maintain as capable process compared to other parallel techniques like BIDE. In first iteration, we assignment an itemset s_p that spawned from preferred itemset s_i from DB_S and an aspect e_i considered from 'I'. If the $f_{ts}(s_p)$ is greater or identical to rs, then an Graph will be distinct between s_i and e_i. If $f_{ts}(s_i) \cong f_{ts}(s_p)$ then we prune s_i from DB_S. This pruning progression required and inadequate to first iteration only.

Beginning second iteration past project the itemset S_p that spawned since $S_{p'}$ to each aspect e_i of 'I'. An Graph can be distinct among $S_{p'}$ and e_i if $f_{ts}(s_p)$ is greater or identical to rs. In this description $S_{p'}$ is a estimated itemset in preceding iteration and adequate as a sequence. Then concern the fallowing validation to locate closed sequence.

If any of $f_{ts}(s_{p'}) \cong f_{ts}(s_p)$ that Graph will be reduce and all replace graphs except s_p will be measured as closed sequence and moves it into DB_S and eliminate all disjoint graphs from recollection.

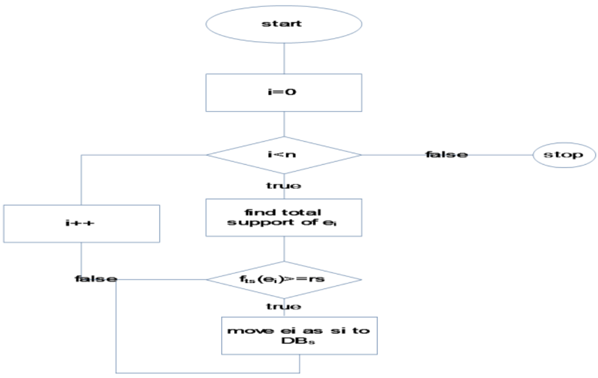
If $f_{ts}(s_{p'}) \cong f_{ts}(s_p)$ and there after no projection spawned then s_p will be measured as closed sequence and moves it into DB_S and eliminate $s_{p'}$ *and* s_p from recollection.

The exceeding process continues dig the elements obtainable in memory those are linked during direct or transitive Graphs and prognostic itemsets i.e., till graph happen to empty

Algorithms used in PEPP:

This section describes algorithms for initializing sequence database with single elements sequences, spawning itemset projections and pruning Graphs from Sequence Graph SG.

Fig 1: Generate initial DB_S with single element itemsets

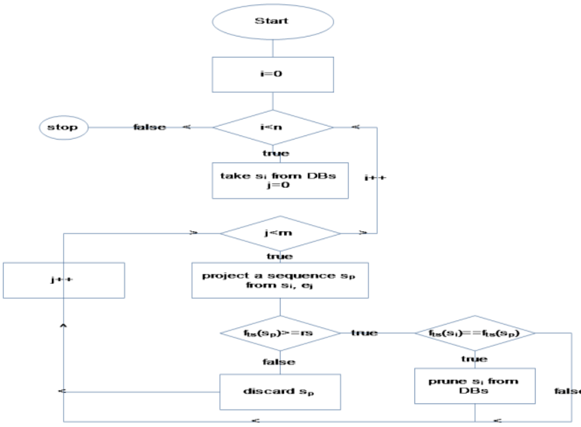


Algorithm 1: Generate initial DB_S with single element itemsets

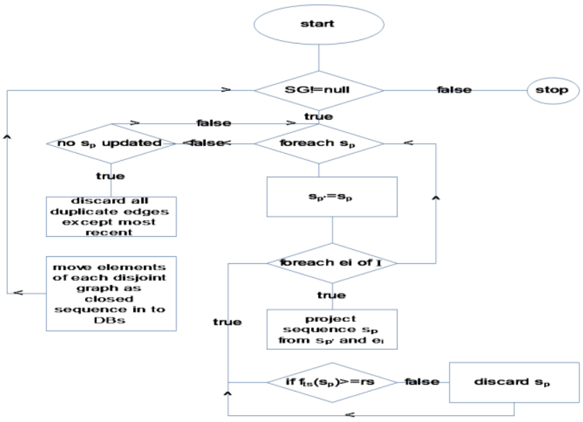
Input: Set of Elements 'I'.
Begin:
L1: For each element e_i of 'I'
Begin:
Find $f_{ts}(e_i)$
If $f_{ts}(e_i) \geq rs$ then
Move e_i as sequence with single element to DB_S
End: L1.
End.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Fig 2: spawning projected Itemsets and protruding sequence graph



First iteration



Rest of all Iterations

Algorithm 2: spawning projected Itemsets and protruding sequence graph

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 itemset s_p from (s_i, e_i)

If $f_{ts}(s_i) \cong f_{ts}(s_p)$ then prune s_i from DB_S

End: C1.

End: L2.

End: L1.

L3: For each projected Itemset s_p in memory

Begin:

$s_{p'} = s_p$

L4: For each e_i of 'I'

Begin:

Project s_p from $(s_{p'}, e_i)$

C2: If $f_{ts}(s_p) \geq rs$

Begin

Spawn SG by adding Graph between $s_{p'}$ *and* e_i

End: C2

End: L4

C3: If $s_{p'}$ not spawned and no new projections added for $s_{p'}$

Begin:

Remove all duplicate Graphs for each Graph weight from $s_{p'}$ and keep Graphs unique by not deleting most recent Graphs for each Graph 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 $SG \neq \phi$ go to L3

Comparative Study

In this segment, we will current our methodical experimental results in regulate to testify the following claims: (1)The PEPP is accurately designed frequent closed progression mining algorithm like BIDE, can considerably outperform compared to other algorithms like CloSpan and spade.(2) PEPP consumes much less memory and can be faster than CloSpan and similar to BIDE. 3). the feature parallel projection and Graph pruning of the PEPP, improves the performance and minimize the memory utilization cost. In the context of dense data the comparative study observed that PEPP

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

significantly performed better when compared with existing models, in particular with BIDE.

The implementation of the BIDE and PEPP algorithms was done using JAVA 1.6 20th build. Both the algorithms tested on a computer with core2duo processor and 2GB RAM and Windows XP installed. Java thread concept was used to achieve the parallel model.

Dataset Characteristics:

We discover a very opaque dataset, Pi, from which a huge number of common closed sequences can be mined yet with a very high sustain threshold like 90%. This dataset is furthermore a bio-dataset which contains 190 protein sequences and 21 dissimilar items. This dataset has yet been used to evaluate the reliability of efficient inheritance [22]. Dataset sequence length status can be found in fig 5.

Since the Bide already proven as better closed pattern mining model when compared to other frequently cited models like spade, prefixSpan and cloSpan, our comparative study in particular for memory utilization and run time, consider the performance comparison between BIDE and PEPP.

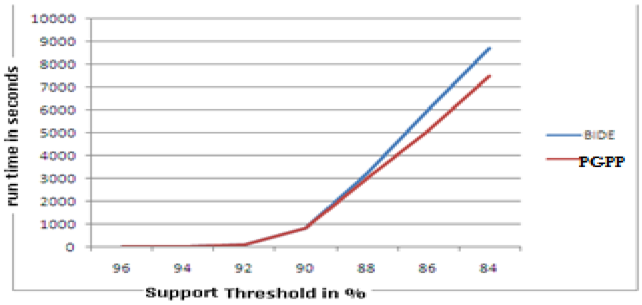


Fig 3: A evaluation report for Runtime

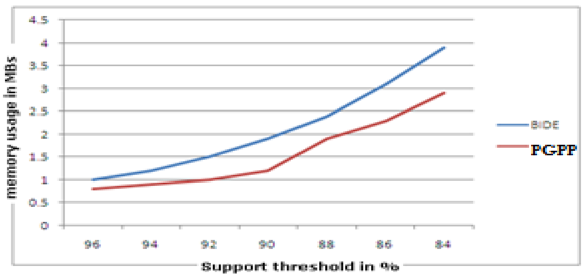


Fig: 4: A evaluation report for memory usage

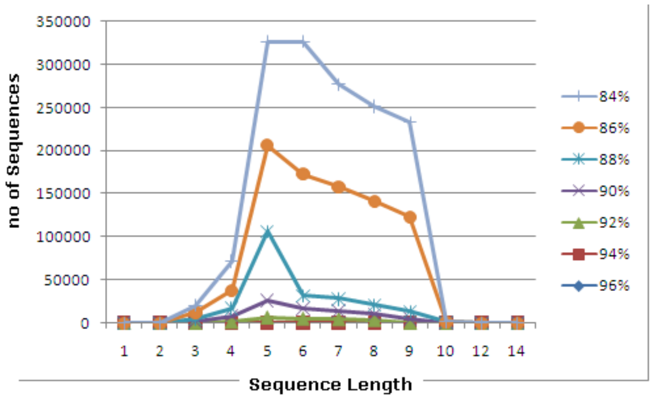


Fig 5: Sequence length and number of sequences at dissimilar thresholds in Pi dataset

We used extremely dense dataset, Pi, to compare PEPP with BIDE. Since In this dataset, we canister observe that still with a very elevated support like 90%, there can be a huge number of diminutive frequent congested sequences with a span less than 10. Fig. 3 shows that with a support higher than 90%, these two algorithms have very similar performance, but once the support is 88% or less, we can observe the outperform of PEPP over BIDE. For example, at support 88%, PEPP performance can be observable, which is faster than BIDE. From Fig. 4 we can observe the considerable difference in memory utilization between PEPP and BIDE, where PEPP always uses considerable less memory than BIDE. At support 88% and less, the less utilization of the memory by PEPP compared to BIDE is in high.

Conclusion

Plenty researchers have developed that closed pattern mining offers the similar significant power as which of all frequent pattern mining even leads to additional compact consequences set and substantially better performance. Our research demonstrated that this is normally true when the quantity of frequent patterns is excessively huge, in that case the amount of frequent closed patterns is additionally likely very significant. However, most of the formerly designed closed pattern mining algorithms depend on the traditional set of frequent closed patterns to assess if a recently found frequent pattern is restricted or if it can invalidate

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

certain definitely mined closed candidates. Simply because the set of already excavated frequent closed patterns holds growing through the mining process, not really will it intake more memory, but also contribute to inefficiency because of the growing query space for pattern closure monitoring. In this paper, we suggested PEPP, a novel algorithm for mining frequent closed sequences making use of sequence Graph. It prevents the curse of the prospect maintenance-and-test paradigm, manages the memory space conveniently by pruning Graphs perfectly and checks the method closure in a additional efficient way although consuming much reduced memory in distinction to the formerly developed closed pattern mining algorithms. It will not need to preserve the set of historic closed patterns, thus it machines very well in the amount of frequent closed patterns. PEPP chooses a Sequence Graph and can produce the frequent closed patterns in an on the web fashion. A comprehensive set of studies on several genuine datasets with assorted distribution functions have revealed the performance of the algorithm design: PEPP utilizes less memory while can be efficient than the CloSpan and BIDE algorithms. It also has additive scalability in terms of the number of sequences in the database. Numerous studies have demonstrated that constraints are recommended for many sequential pattern mining purposes. In the upcoming, we plan to utilize the inference strategy on projected itemsets to develop the rule coherency.

References

[1]F. Masseglia, F. Cathala, and P. Poncelet, The psp approach for mining sequential patterns. In PKDD'98, Nantes, France, Sept. 1995.

[2]R. Srikant, and R. Agrawal, Mining sequential patterns: Generalizations and performance improvements. In EDBT'96, Avignon, France, Mar. 1996.

[3]J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, FreeSpan: Frequent pattern-projected sequential pattern mining . In SIGKDD'00, Boston, MA, Aug. 2000.

[4]M. Zaki, SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning, 42:31-60, Kluwer Academic Pulishers, 2001.

[5]J. Pei, J. Han, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In ICDE'01, Heidelberg, Germany, April 2001.

[6]J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, Sequential PAttern Mining using a Bitmap Representation. In SIGKDD'02, Edmonton, Canada, July 2002.

[7]M. Garofalakis, R. Rastogi, and K. Shim, SPIRIT: Sequential PAttern Mining with regular expression constraints. In VLDB'99, San Francisco, CA, Sept. 1999.

[8]J. Pei, J. Han, and W. Wang, Constraint-based sequential pattern mining in large databases. In CIKM'02, McLean, VA, Nov. 2002.

[9]M. Seno, G. Karypis, SLPMiner: An algorithm for finding frequent sequential patterns using lengthdecreasing support constraint. In ICDM'02,, Maebashi, Japan, Dec. 2002.

[10]H. Mannila, H. Toivonen, and A.I. Verkamo, Discovering frequent episodes in sequences . In SIGKDD'95, Montreal, Canada, Aug. 1995.

[11]B. Ozden, S. Ramaswamy, and A. Silberschatz, Cyclic association rules. In ICDE'98, Olando, FL, Feb. 1998.

[12]C. Bettini, X. Wang, and S. Jajodia, Mining temporal relationals with multiple granularities in time sequences. Data Engineering Bulletin, 21(1):32-38, 1998.

[13]J. Han, G. Dong, and Y. Yin, Efficient mining of partial periodic patterns in time series database. In ICDE'99, Sydney, Australia, Mar. 1999.

[14]J. Yang, P.S. Yu, W. Wang and J. Han, Mining long sequential patterns in a noisy environment. In SIGMOD' 02, Madison, WI, June 2002.

[15]N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, Discoving frequent closed itemsets for association rules. In ICDT'99, Jerusalem, Israel, Jan. 1999.

[16]M. Zaki, and C. Hsiao, CHARM: An efficient algorithm for closed itemset mining. In SDM'02, Arlington, VA, April 2002.

[17]X. Yan, J. Han, and R. Afshar, CloSpan: Mining Closed Sequential Patterns in Large Databases. In SDM'03, San Francisco, CA, May 2003.

[18]J. Wang, J. Han, and J. Pei, CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. In KDD'03, Washington, DC, Aug. 2003.

[19]R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In VLDB'94, Santiago, Chile, Sept. 1994.

[20]J. Pei, J. Han, and R. Mao, CLOSET: An efficient algorithm for mining frequent closed itemsets . In DMKD'01 workshop, Dallas, TX, May 2001.

[21]J. Han, J. Wang, Y. Lu, and P. Tzvetkov, Mining Top- K Frequent Closed Patterns without Minimum Support. In ICDM'02, Maebashi, Japan, Dec. 2002.

[22]P. Aloy, E. Querol, F.X. Aviles and M.J.E. Sternberg, Automated Structure-based Prediction of Functional Sites in Proteins: Applications to Assessing the Validity of Inheriting Protein Function From Homology in Genome Annotation and to Protein Docking. Journal of Molecular Biology, 311, 2002.

[23]R. Agrawal, and R. Srikant, Mining sequential patterns. In ICDE'95, Taipei, Taiwan, Mar. 1995.

[24]I. Jonassen, J.F. Collins, and D.G. Higgins, Finding flexible patterns in unaligned protein sequences. Protein Science, 4(8), 1995.

[25]R. Kohavi, C. Brodley, B. Frasca, L.Mason, and Z. Zheng, KDD-cup 2000 organizers' report: Peeling the Onion. SIGKDD Explorations, 2, 2000.

[26]Jianyong Wang, Jiawei Han: BIDE: Efficient Mining of Frequent Closed Sequences. ICDE 2004: 79-90

[27]Kalli Srinivasa Nageswara Prasad and Prof. S Ramakrishna. Article: Frequent Pattern Mining and Current State of the Art. International Journal of Computer Applications 26(7):33-39, July 2011. Published by Foundation of Computer Science, New York.