Concurrent Edge Prevision and Rear Edge Pruning Approach for Frequent Closed Itemset Mining

Anurag Choubey[1]
Dean Academic, Technocrats
Institute of Technology,
RGPV,
Bhopal (M.P.), India

Dr. Ravindra Patel[2]
Reader & Head, Dept. of Computer
Application,
UIT-RGPV,
Bhopal (M.P.), India

Dr. J.L. Rana[3]
Former Professor & Head, Dept. of
CSE & IT MANIT,
Bhopal (M.P.), India

***Abstract*- Past observations have shown that a frequent item set mining algorithm are purported to mine the closed ones because the finish provides a compact and a whole progress set and higher potency. Anyhow, the newest closed item set mining algorithms works with candidate maintenance combined with check paradigm that is pricey in runtime yet as space usage when support threshold is a smaller amount or the item sets gets long. Here, we show, CEG&REP that could be a capable algorithm used for mining closed sequences while not candidate. It implements a completely unique sequence finality verification model by constructing a Graph structure that build by an approach labeled "Concurrent Edge Prevision and Rear Edge Pruning" briefly will refer as CEG&REP. a whole observation having sparse and dense real-life knowledge sets proved that CEG&REP performs bigger compared to older algorithms because it takes low memory and is quicker than any algorithms those cited in literature frequently.**

Keywords*- *Data mining; Closed Itemsets; Pattern Mining; sequence length; graph structure.

I. 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, CEG&REP, which can mine efficiently all the sets of frequent closed sequences through a sequence graph protruding approach. In CEG&REP, 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 CEG&REP 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 CEG&REP and its utilization for Sequence Graph protruding explained. In section V, the algorithms used in CEG&REP described. In section V1, results gained from a comparative study briefed and fallowed 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], 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 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 our earlier literature[27] we discussed 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 Concurrent Edge Prevision and Rear Edge Pruning (CEG&REP) based Sequence Graph Protruding for closed itemset mining. The Edge Projection is a forward approach grows till edge with required support is possible during that time the edges will be pruned. During this pruning process vertices of the edge that differs in support with next edge projected will be considered as closed itemset, also the sequence of vertices that connected by edges with similar support and no projection possible also be considered as closed itemset

(B) in the Edge Projection and pruning based Sequence Graph Protruding for closed itemset mining, we create a algorithms for Edge Prevision and Rear 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 BIDE 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.

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

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

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

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.

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

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 is belongs to s_q as an ordered list. This can be formulated as

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

Then $$\bigcup_{i=1}^{n} s_{pi} \mathrel{<:} \bigcup_{j=1}^{m} s_{qj}$$ 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 fallowing formulation.

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

DB_S Is set of sequences

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

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

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

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

And $s_t <: s_p \; for\, any\, p\, value$ where $f_{ts}(s_t) \cong f_{ts}(s_p)$

IV. Concurrent Edge Prevision and Rear Edge Pruning

A. Preprocess:

As a first stage of the proposal we perform 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 required support.

B. Edge Prevision:

In this phase, we select all itemsets from given itemset database as input in parallel. Then we start projecting edges from each selected itemset to all possible elements. The first iteration includes the pruning process in parallel, from second iteration onwards this pruning is not required, which we claimed as an efficient process compared to other similar techniques like BIDE. In first iteration, we project an itemset s_p that spawned from selected itemset s_i 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 $f_{ts}(s_i) \cong f_{ts}(s_p)$ then we prune s_i from DB_S. This pruning process required and limited to first iteration only.

From second iteration onwards project the itemset 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 itemset in previous iteration and eligible as a sequence. Then apply the fallowing validation to find closed sequence.

If any of $f_{ts}(s_{p'}) \cong f_{ts}(s_p)$ 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 $f_{ts}(s_{p'}) \cong f_{ts}(s_p)$ and there after no projection spawned then s_p will be considered as closed sequence and moves it into DB_S and remove $s_{p'}\; and\; s_p$ from memory.

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

1) Algorithm used in CEG&REP:

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

Algorithm 1: Concurrent Edge Prevision to build graph structure and Rear Edge Pruning

```
Step 1:
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.
Step 2:
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: $\text{if edgeWeight}(s_i, e_i) \geq rs$
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 edge 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 edges for each edge weight from $s_{p'}$ 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 $SG \neq \phi$ go to L3
```

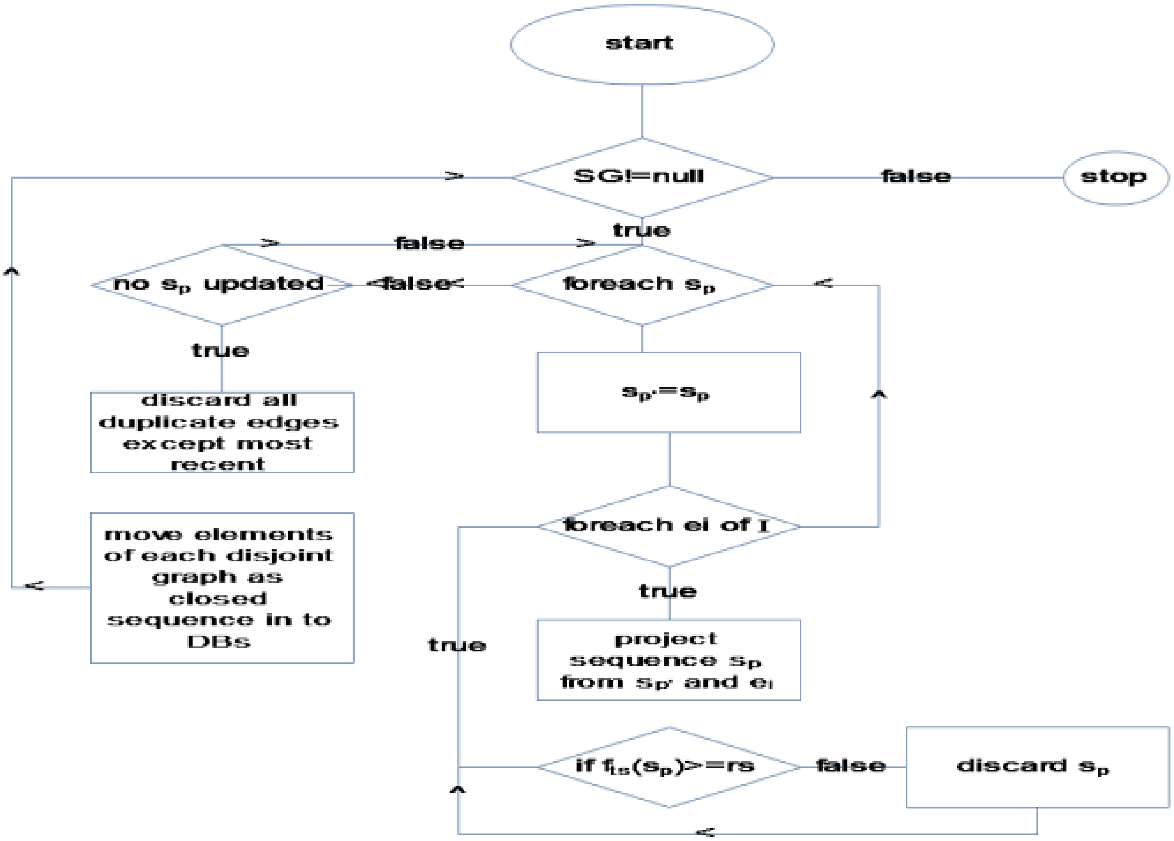
start
SG!=null
false
stop
true
false
no s_p updated
false
foreach s_p
true
discard all duplicate edges except most recent
$s_{p'}=s_p$
move elements of each disjoint graph as closed sequence in to DBs
foreach ei of I
true
true
project sequence s_p from $s_{p'}$ and e_i
if $f_{ts}(s_p)>=rs$
false
discard s_p

Fig 1: Concurrent Edge Prevision to build graph structure and Rear Edge Pruning

VI. COMPARATIVE STUDY:

This segment focuses mainly on providing evidence on asserting the claimed assumptions that 1) The CEG&REP is similar to BIDE which is actually a sealed series mining algorithm that is competent enough to momentously surpass results when evaluated against other algorithms such as CloSpan and spade. 2) Utilization of memory and momentum is rapid when compared to the ColSpan algorithm which is again analogous to BIDE. 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 CEG&REP. This is on the basis of the surveillance done which concludes that CEG&REP’s implementation is far more noteworthy and important in contrast with the likes of BIDE, to be precise.

JAVA 1.6_ 20th build was employed for accomplishment of the CEG&REP and BIDE 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.

A. 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. Fig. 5 portrays an image depicting dataset series extent status.

In assessment with all the other regularly quoted forms like spade, prefixspan and CloSpan, BIDE 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 CEG&REP.

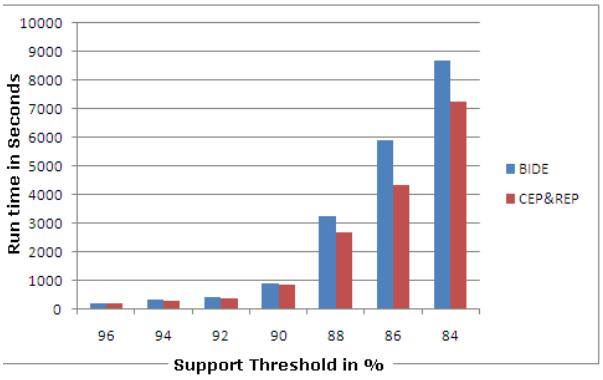

Fig 3: A comparison report for Runtime

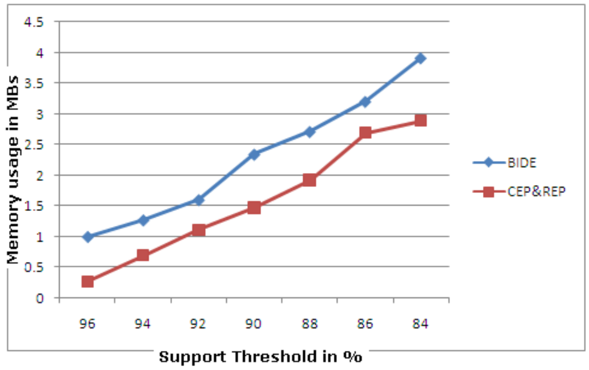

Fig: 4: A comparison report for memory usage

In contrast to CEG&REP and BIDE, 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 Fig 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 CEG&REP surpasses BIDE's routine.

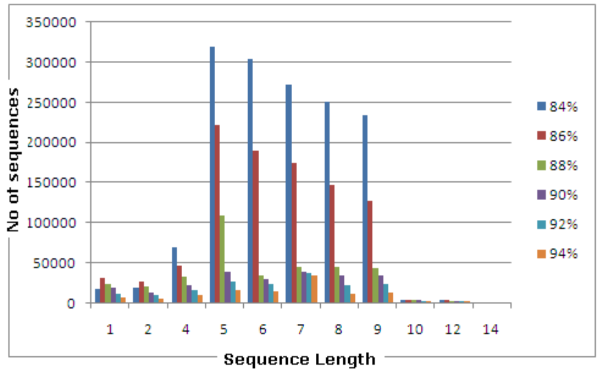

Fig 5: Sequence length and number of sequences at different thresholds in Pi dataset

The disparity in memory exploitation of CEG&REP and BIDE can be clearly observed because of the consumption level of CEG&REP being low than that of BIDE.

V. 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 te following functions: It shuns the blight 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 CEG&REP 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. CEG&REP is rich in terms of velocity and memory spacing in comparison with the BIDE 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 itemsets.

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] Anurag Choubey, Dr. Ravindra Patel and,Dr. J.L. Rana. Article: Frequent Pattern Mining With Closeness Considerations: Current State Of The Art. GJCST Issue 11, Volume 17, August 2011. Published by Global Journals, 25200 Carlos Bee Blvd. #495, Hayward, CA 94542, USA

AUTHORS PROFILE

Anurag Choubey[1]: He has completed B.Sc. (Electronics) in 1990 from Dr. Hari Singh Gaur Vishwavidyalaya, Sagar, M.P., India and M.Sc. (Applied Physics) in 1993 from Govt. Engineering College, Jabalpur, M.P., India. He has completed MCA in 2008 from Guru Ghasidas Vishwavidyalaya (Presently central University), Bilaspur C.G., India. He has worked as a Lecturer at Govt. Engg. College, Jabalpur and Hitkarni College of Engineering & Technology, Jabalpur from 1998 to 2000. From October 2000 onwards he has been working in Technocrats Institute of Technology, Bhopal. Currently working as Dean Academic, Technocrats Institute of Technology, Bhopal M.P., India. He posses more than 13 years of experience in teaching and has worked in different capacities like controller of exam, admission in-charge and other administrative post. email: anuragphd11@gmail.com

Dr. Ravindra Patel[2]: Associate Professor and Head, Department of Computer Applications at Rajiv Gandhi Technological University, Bhopal, India. He has awarded Ph.D. degree in Computer Science. He possessed more than 10 years of experience in teaching post-graduate classes. He has published more than 15 papers in international and national journals and conference proceedings. He is member of International Association of Computer Science and Information Technology (IACSIT). Email: ravindra@rgtu.net

Dr. J.L. Rana[3]: Former Professor and Head, Department of Computer Science and Engineering & Information Technology at MANIT, Bhopal, India with 32 years of vast experience of teaching, out of which 19 years as Professor of Computer Science & Engg. and Information Technology. Currently he is working as Group Director of Radha Raman Group of Institute, Bhopal. He has completed B.E. (Hons) in 1968 from GEC, Jabalpur and M.S. Computer Control from University of Hawaii (UH), USA in 1972. He has awarded Ph.D. degree in Computer from Indian Institute of Technology, Bombay in 1987. He also posses more than 25 years of experience in post graduate teaching He has published more than 30 papers in international and national journals and 65 international and national conference proceedings. He has guided 12 nos. of Ph.D. and 5 no. in progress He is a senior life member of CSI and Chairman CSI Bhopal Chapter for two terms. Email: jl_rana@yahoo.com