Global Journal of Computer Science and Technology
Volume 11 Issue 19 Version 1.0 November 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Mining Closed Itemsets for Coherent Rules: An Inference Analysis Approach

By Kalli Srinivasa Nageswara Prasad, Prof. S. Ramakrishna
Sri Venkateswara University

Abstract - Past observations have shown that a frequent item set mining algorithm are alleged to mine the closed ones because the finish offers a compact and a whole progress set and higher potency. Anyhow, the most recent closed item set mining algorithms works with candidate maintenance combined with check paradigm that is dear in runtime likewise as area usage when support threshold is a smaller amount or the item sets gets long. Here, we show, PEPP with inference analysis that could be a capable approach used for mining closed sequences for coherent rules while not candidate. It implements a unique sequence closure checking format with inference analysis that based mostly on Sequence Graph protruding by an approach labeled "Parallel Edge projection and pruning" in brief will refer as PEPP. We describe a novel inference analysis approach to prune patterns that tends to derive coherent rules. A whole observation having sparse and dense real-life information sets proved that PEPP with inference analysis performs larger compared to older algorithms because it takes low memory and is quicker than any algorithms those cited in literature frequently.

Keywords : Data mining, Association Rule Mining, Closed itemset, Frequent Itemset,KDD,PEPP.

GJCST Classification : H.2.8

MINING CLOSED ITEMSETS FOR COHERENT RULESAN INFERENCE ANALYSIS APPROACH

Strictly as per the compliance and regulations of:

Mining Closed Itemsets for Coherent Rules: An Inference Analysis Approach

Kalli Srinivasa Nageswara Prasad [α], Prof. S. Ramakrishna [Ω]

Abstract - Past observations have shown that a frequent item set mining algorithm are alleged to mine the closed ones because the finish offers a compact and a whole progress set and higher potency. Anyhow, the most recent closed item set mining algorithms works with candidate maintenance combined with check paradigm that is dear in runtime likewise as area usage when support threshold is a smaller amount or the item sets gets long. Here, we show, **PEPP** with inference analysis that could be a capable approach used for mining closed sequences for coherent rules while not candidate. It implements a unique sequence closure checking format with inference analysis that based mostly on Sequence Graph protruding by an approach labeled "Parallel Edge projection and pruning" in brief will refer as **PEPP**. We describe a novel inference analysis approach to prune patterns that tends to derive coherent rules. A whole observation having sparse and dense real-life information sets proved that **PEPP** with inference analysis performs larger compared to older algorithms because it takes low memory and is quicker than any algorithms those cited in literature frequently.

Keywords : Data mining, Association Rule Mining, Closed itemset, Frequent Itemset, KDD, PEPP.

I. Introduction

Association rule mining, introduced in [28], is considered as one of the most important tasks in Knowledge Discovery in Databases [29]. Among sets of items in transaction databases, it aims at discovering implicative tendencies that can be valuable information for the decision-maker. An association rule is defined as the implication $X \rightarrow Y$, described by two interestingness measures support and confidence, where X and Y are the sets of items and $X \cap Y = \phi$. Apriori [28] is the first algorithm proposed in the association rule mining field and many other algorithms were derived from it. It is very well known that mining algorithms can discover a prohibitive amount of association rules; Starting from a database, it proposes to extract all association rules satisfying minimum thresholds of support and confidence. For instance, thousands of rules are extracted from a database of several dozens of attributes and several hundreds of transactions. Furthermore, as suggested by Silbershatz and Tuzilin [30], valuable information is often represented by those rare low support and unexpected association rules which are surprising to the user. So, the more we increase the support threshold, the more efficient the algorithms are and the more the discovered rules are obvious, and hence, the less they are interesting for the user. As a result, it is necessary to bring the support threshold low enough in order to extract valuable information. Unfortunately, the lower the support is, the larger the volume of rules becomes, making it intractable for a decision-maker to analyze the mining result. Experiments show that rules become almost impossible to use when the number of rules overpasses 100. Thus, it is crucial to help the decision-maker with an efficient technique for reducing the number of rules.

To overcome this drawback, several methods were proposed in the literature. On the one hand, different algorithms were introduced to reduce the number of itemsets by generating closed [31], maximal [32] or optimal itemsets [33], and several algorithms to reduce the number of rules, using non redundant rules [34], [35], or pruning techniques [36]. On the other hand, post processing methods can improve the selection of discovered rules. Different complementary post processing methods may be used, like pruning, summarizing, grouping, or visualization [37]. Pruning consists in removing uninteresting or redundant rules. In summarizing, concise sets of rules are generated. Groups of rules are produced in the grouping process, and the visualization improves the readability of a large number of rules by using adapted graphical representations.

However, most of the existing post processing methods are generally based on statistical information in the database. Since rule interestingness strongly depends on user knowledge and goals, these methods do not guarantee that interesting rules will be extracted. In this paper, we propose a novel framework to identify closed itemsets. Associations are discovered based on inference analysis. The principle of the approach considers that an association rule should only be reported when there is enough interest gain claimed during inference analysis in the data. To do this, we consider both presence and absence of items during the mining. An association such as

beer $\rightarrow$ nappies will only be reported if we can also find that there are fewer occurrences of $\neg$beer $\rightarrow$nappies and beer $\rightarrow\neg$nappies but more of $\neg$beer $\rightarrow$ $\neg$

Author [α] : Research Scholar in Computer Science Sri Venkateswara University, Tirupati Andhra Pradesh, India, Mobile: 9490171769
Email : kallisnprasad@gmail.com
Author [Ω] : Department of Computer Science Sri Venkateswara University, Tirupati Andhra Pradesh, India,
Mobile : 944149572 Email : profsramakrishnasvu@gmail.com

Global Journal of Computer Science and Technology Volume XI Issue XIX Version I November 2011

nappies. This approach will ensure that when a rule such as
beer $\longrightarrow$ nappies is reported, it indeed has the strongest interest in the data as comparison was made on both presence and absence of items during the mining process.

II. Related Work

The sequential item set mining problem was initiated by Agrawal and Srikant, and the same was developed as 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 that 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.

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' is total ordered items and $P(e_i)$ is 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 and s_j: is a sequence that belongs to S.

Subsequence: is 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 $$\bigcup_{i=1}^{n} s_{pi} <: \bigcup_{j=1}^{m} s_{qj}$$ 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. Closed Itemset Discovery

a) PEPP: *Parallel Edge Projection and Pruning Based Sequence Graph protrude [28]*

i. *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.

ii. *Forward Edge Projection:*

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 following validation to find closed sequence.

iii. *Edge pruning:*

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.

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.

b) *Inference Analysis:*

Inferences:-

a. Pattern positive score is sum of no of transactions in which all items in the pattern exist, no. of transactions in which all items in the pattern does not exist.
b. Pattern negative score is no of transactions in which only few items of the pattern exist.
c. Pattern actual coverage is pattern positive score-pattern negative score.
d. Interest gain is Actual coverage of the pattern involved in association rule.
e. Coherent rule Actual coverage of the rule's left side pattern must be greater than or equal to actual coverage of the right side pattern.
f. Inference Support ia_s refers actual coverage of the pattern.
g. $f_{ia}(s_t)$ Represents the inference support of the sequence s_t.

V. Approach

For each pattern s_p of the pattern dataset, If $f_{ia}(s_t) < ia_s$ then we prune that pattern

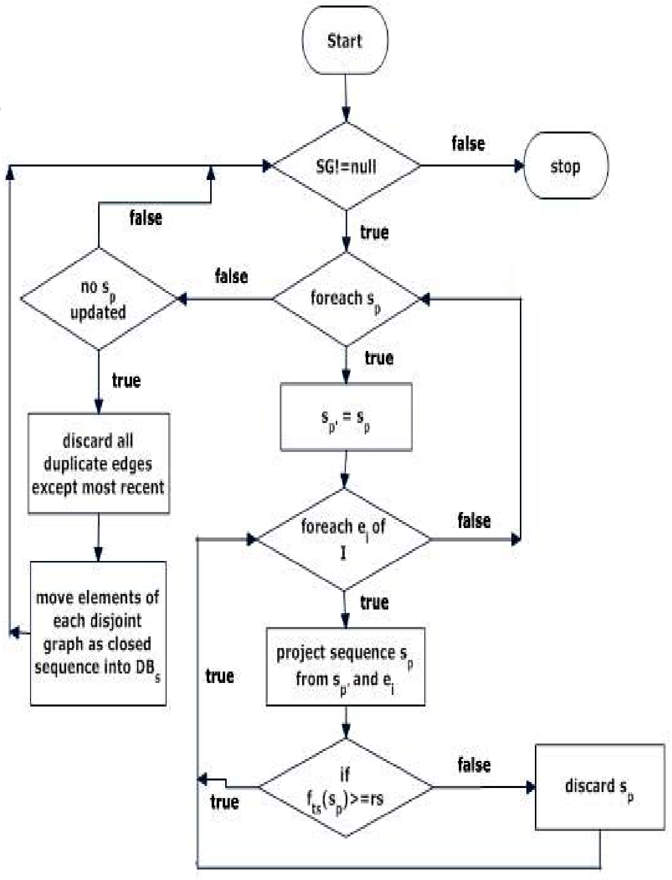
a) PEPP[1] *Algorithm:*

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

```
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 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.
```

Global Journal of Computer Science and Technology Volume XI Issue XIX Version I November 2011

```
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
true
s_p' = s_p
discard all duplicate edges except most recent
foreach e_i of I
false
true
move elements of each disjoint graph as closed sequence into DB_s
true
project sequence s_p from s_p' and e_i
if f_ts(s_p)>=rs
false
true
discard s_p

Fig.1 : flowchart for PEPP Algorithm

b) Description of Inference Analysis

Set I = {i1, i2… im} be the universe of items composed of m different attributes, ik(k=1,2,...,m) is item. Transaction database D is a collection of transaction T, A transaction t = (tid, X) is a tuple where tid is a unique transaction ID and X is an itemset. The count of an itemset X in D, denoted by count(X), is the number of transactions in D containing X. The support of an itemset X in D, denoted by supp(X), is the proportion of transactions in D that contain X. The negative rule X⇒¬Y holds in the transaction set D with confidence conf (X⇒¬Y) = supp $(X \cup \neg Y)$ /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:

Find frequent itemsets, Generate all itemsets that support is greater than the minimum support. Generate association rules from frequent itemsets. In logical analysis, the direct calculation of support 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⊂I , A∩B=Φ ,then:

$$\sup(\neg A) = 1 - \sup(A);$$
$$\sup(A \cup \neg B) = \sup(A) - Sup(A \cup B);$$
$$\sup(\neg A \cup B) = \sup(B) - \sup(A \cup B)$$
$$\sup(\neg A \cup \neg B) = 1 - \sup(A) - \sup(B) + \sup(A \cup B);$$

Based on the above formulas we perform the logical analysis to derive the actual support of the patterns that improves the rule coherency. Inference analysis by example: Let A,B∈I where I is itemset generated with the association of A,B are individual items or subsets.

Under logical analysis we determine $f_{ts}(\neg A \cup \neg B)$, $f_{ts}(A \cup \neg B)$ and $f_{ts}(\neg A \cup B)$. The support $f_{ts}(I)$, $f_{ts}(\neg A \cup \neg B)$ we consider as positive support and $f_{ts}(A \cup \neg B)$, $f_{ts}(\neg A \cup B)$ we consider as negative support. Finally we determine $f_{ia}(I) =$ $f_{ts}(I) + f_{ts}(\neg A \cup \neg B) - f_{ts}(A \cup \neg B) - f_{ts}(\neg A \cup B)$;

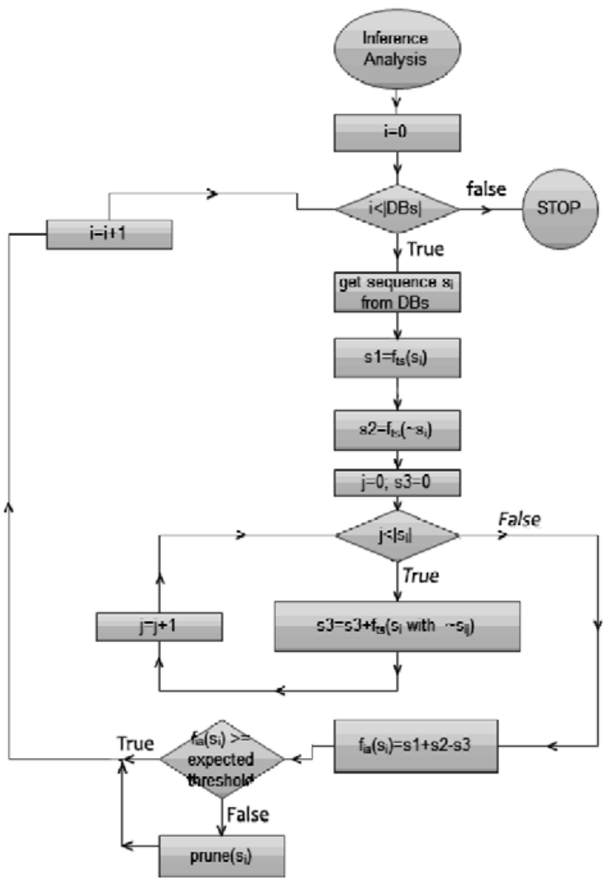


Fig.2 : flowchart for Inference Analysis.

VI. Comparative Study

This segment focuses mainly on providing evidence on asserting the claimed assumptions that 1) The PEPP 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 CloSpan 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 PEPP with inference analysis for no coherent pattern pruning. This is on the basis of the surveillance done which concludes that PEPP'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 PEPP 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.

VII. 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 PEPP.

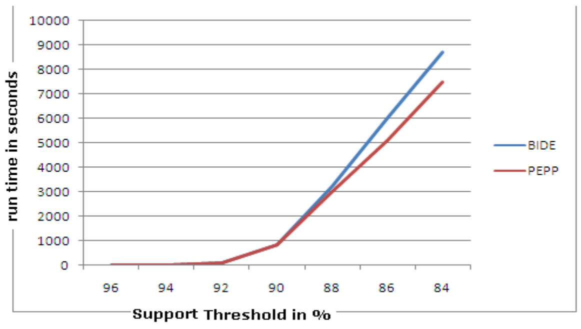


Fig.3 : A comparison report for Runtime.

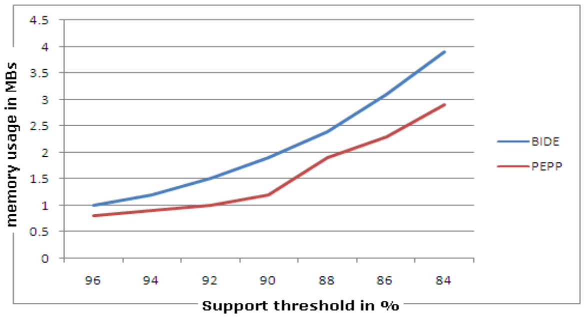


Fig.4 : A comparison report for memory usage.

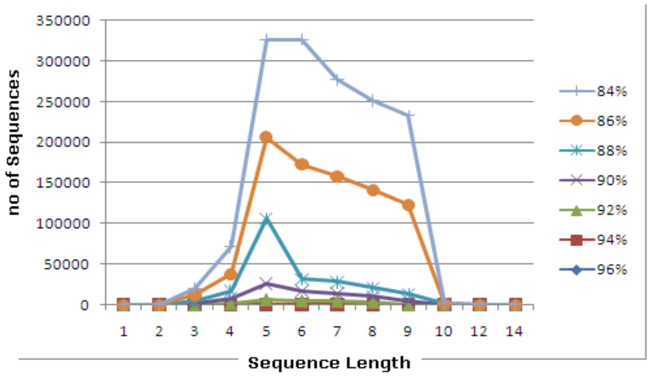


Fig.5 : Sequence length and number of sequences at different thresholds in Pi dataset.

Global Journal of Computer Science and Technology Volume XI Issue XIX Version I November 2011

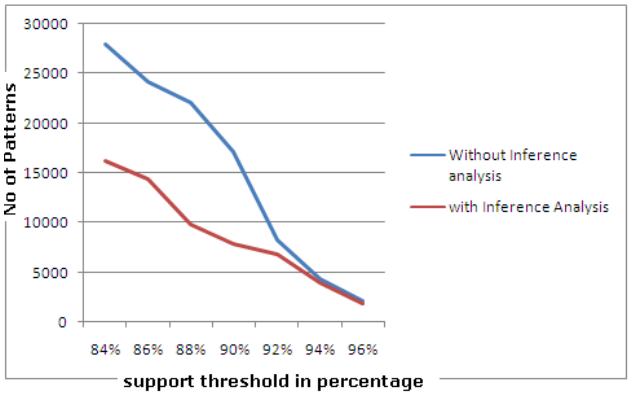


Fig.6 : No patterns detected by PEPP with and without inference analysis.

In contrast to PEPP 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 PEPP surpasses BIDE's routine. The disparity in memory exploitation of PEPP and BIDE can be clearly observed because of the consumption level of PEPP being lower than that of BIDE. The concept inference analysis we introduced here played a vital role in closed itemset detection. The significant improvement in closed itemset detection can be observable in our results, see the fig 6.

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 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 PEPP 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. PEPP 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 is also proven that PEPP is efficient to find closed itemsets under inference analysis. It has been proven and verified by many scientific research studies that limitations are crucial for a number of chronological outlined mining algorithms. In addition we improved closed itemset detection performance by introducing inference analysis as an extension to PEPP. Future studies include proposing of post processing and pruning of the rules based on categorical relations between attributes.

References Références Referencias

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 Publishers, 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 length decreasing 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, Discovering 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. 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. 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.
28. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD, pp. 207-216, 1993.
29. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
30. A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems," IEEE Trans. Knowledge and Data Eng. vol. 8, no. 6, pp. 970-974, Dec. 1996.
31. M.J. Zaki and M. Ogihara, "Theoretical Foundations of Association Rules," Proc. Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD '98), pp. 1-8, June 1998.
32. D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "Mafia: A Maximal Frequent Itemset Algorithm," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 11, pp. 1490-1504, Nov. 2005.
33. J. Li, "On Optimal Rule Discovery," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 460-471, Apr. 2006.
34. M.J. Zaki, "Generating Non-Redundant Association Rules," Proc. Int'l Conf. Knowledge Discovery and Data Mining, pp. 34-43, 2000.
35. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Efficient Mining of Association Rules Using Closed Itemset Lattices," Information Systems, vol. 24, pp. 25-46, 1999.
36. H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hatonen, and H. Mannila, "Pruning and Grouping of Discovered Association Rules," Proc. ECML-95 Workshop Statistics, Machine Learning, and Knowledge Discovery in Databases, pp. 47-52, 1995.
37. B. Baesens, S. Viaene, and J. Vanthienen, "Post-Processing of Association Rules," Proc. Workshop Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics with Sixth ACM SIGKDD, pp. 20-23, 2000.