OPTIMIZATION OF CMAP BASED ALGORITHMS FOR EXTRACTING RARE SEQUENCE PATTERNS

Sonamdeep Kaur[1], Mrs Sarika Chaudhary[2], Mrs. Neha Bishnoi[2]

[1]Department of Computer Science and Engineering, Amity University, Haryana (IND)

[2,3]Department of Computer Science and Engineering, Amity University, Haryana (IND)

ABSTRACT

Frequent item-set pruning in SPAM and SPADE like state-of-the-art algorithms are only based on minimum support value, In our version of modified algorithm we have tried to approach another way to extract both frequent and rare item-sets by using an additional window in a small region of pruned group considering this small group is rare. This simple method proposed here is efficient, fast and is having low memory print. The algorithm is based on the fact that out of all the item-sets there one dimensional property for pruning is minimum support value. It simply divides an infinitely extended one-dimensional number line into two halves. One group becomes frequent and another becomes non-frequent. If we look at rare items, the rare items have a support range only. This important notion is used in our modified version of above mentioned algorithms by creating a window in that small range and then extracting those item-sets along with frequent item-sets extracted using simple minimum support based extraction.

KEYWORDS- Sequence Pattern Mining, Vertical Databases, CM-SPAM, CM-SPADE, Rare Itemset.

INRODUCTION

Data Mining is an analytic method planned to see the sights data (usually large amounts of data - typically business or market related - also known as "big data") in investigation of dependable patterns and/or systematic associations between variables, and then to validate the findings by implementing the detected patterns to new subsets of data. The eventual goal of data mining is calculation - and extrapolative data mining is the most common type of data mining and one that

has the most direct business applications. The process of data mining consists of three stages: (1) the initial exploration, (2) model making or pattern recognition with validation/verification, and (3) deployment (i.e., the use of the model to new data in order to produce predictions).

Sequential Pattern Mining finds interesting sequential patterns among the large database. A subsequence is a Sequential Pattern if it frequently appears in a sequence database and its frequency is no less than a user specified minimum support threshold minsup. It finds out frequent subsequences as patterns from a sequence database. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining sequential patterns from their database. Sequential pattern mining is one of the most well-known methods and has broad applications including web-log analysis, customer purchase behavior analysis and medical record analysis. In the retailing business, sequential patterns can be mined from the transaction records of customers. For example, having bought a notebook, a customer comes back to buy a PDA and a WLAN card next time. The retailer can use such information for analyzing the behavior of the customers, to understand their interests, to satisfy their demands, and above all, to predict their needs. In the medical field, sequential patterns of symptoms and diseases exhibited by patients identify strong symptom/disease correlations that can be a valuable source of information for medical diagnosis and preventive medicine. In Web log analysis, the exploring behavior of a user can be extracted from member records or log files. For example, having viewed a web page on "Data Mining", user will return to explore "Business Intelligence" for new information next time. These sequential patterns yield huge benefits, when acted upon, increases customer royalty.

Constraint-based mining usually represent user's interest and focus, which confines the patterns to be found to a particular subset satisfying some strong conditions. A constraint C for sequential pattern mining is a boolean function $C(\alpha)$ on the set of all sequences. The problem of constraint-based sequential pattern mining is to find the complete set of sequential patterns satisfying a given constraint C. Constraints can be examined and characterized from different points of views.

LITERATURE REVIEW

Agrawal and Srikant in their paper "Mining Sequential Patterns" (1995) introduced a new problem of mining sequential patterns from a database of customer sales transactions and presented three algorithms for solving this problem. Two of the algorithms, AprioriSome and

AprioriAll, have comparable performance, although AprioriSome performs a little better for the lower values of the minimum number of customers that must support a sequential pattern.

Zaki in the paper "SPADE: An Efficient Algorithm for Mining Frequent Sequences" (2001) has presented SPADE, a new algorithm for fast discovery of Sequential Patterns. The existing solutions to this problem make repeated database scans, and use complex hash structures which have poor locality. SPADE utilizes combinatorial properties to decompose the original problem into smaller sub-problems that can be independently solved in main-memory using efficient lattice search techniques, and using simple join operations.

Jian Pei et.al in their paper "Constraint-based sequential pattern mining: the pattern-growth methods" (2005) studied the problem of pushing various constraints deep into sequential pattern mining. They characterized constraints for sequential pattern mining from both the application and constraint-pushing points of views. A general property of constraints for sequential pattern mining, prefix monotone property, is identified. It covers many commonly used constraints. An efficient algorithm, PG, is developed to push prefix-monotone constraints deep into the mining process.

Jiwei Han et.al in their paper "Frequent pattern mining: current status and future directions" (2007) presented a brief overview of the current status and future directions of frequent pattern mining. With over a decade of extensive research, there have been hundreds of research publications and tremendous research, development and application activities in this domain.

Philippe et.al in their paper "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information" (2014) presented a novel data structure named CMAP for storing co-occurrence information to address the problem of generation of a large amount of infrequent candidates by vertical algorithms. They explained how CMAPs can be used for pruning candidates generated by vertical mining algorithms.

PREVIOUS WORK

Candidate Generation in CMSPAM

The SEARCH procedure takes a sequential pattern *pat* and two sets of items to be appended to *pat* as input parameters to generate candidates. The items to be appended to pat by s-extension are

represented by first set Sn. The s-extension of a sequential pattern {I1, I2 ... Ih} with an item y is defined as {I1, I2 ... Ih ∪ {y}}. The items that are to be appended to pat by i-extension are represented by second set Si. The i-extension of a sequential pattern {I1, I2 ... Ih} with an Item x is defined as {{I1, I2, I3} ∪ {s}}. CMSPAM calculates the support for each candidate pattern generated by an extension to determine if the pattern is frequent. To do that, a join operation and count of number of sequences where pattern appears operation are performed. The Id List representation used by CMSPAM is based on bitmaps to get faster operations. If the pattern *pat* is frequent, then it is again used in a recursive call to SEARCH to generate patterns starting with the prefix *pat*. CMSPAM does not extend infrequent patterns thus prune it by default and reduce time.

Candidate Generation in CMSPADE

CMSPADE is another very fast and efficient algorithm. It takes a sequential database SDB and minimum support threshold *minsup* as input to its procedure. It first creates the vertical representation of the horizontal sequence database as V (SDB). It then identifies the set of frequent sequential patterns F1 containing frequent items. Then it calls another procedure ENUMERATE with equivalence class of size 0 as input. An *equivalence class* of size n is defined as a set of all the frequent patterns containing n items sharing same prefix of n-1 items. For F1 patterns there is only 1 equivalence class of size 0 containing F1. The output of each member of the equivalence class Ai, is a frequent sequential pattern. Then, a set Ti representing the equivalence class of all frequent extensions of Ai is initialized to the empty set. Then, the pattern Ai is merged with Aj for each Aj ∈ F such that i lex j, to form larger pattern(s). Then, the support is calculated for each pattern r, by performing join operation between IdLists of Ai and Aj. Now, again the frequent pattern is determined by testing its cardinality against *minsup* . The frequent patterns are added to Ti. Finally, Ti contains the whole equivalence class of patterns starting with prefix Ai. The EUMERATE is called again recursively to determine larger patterns with Ai as prefix. When all loops terminate, we get all the frequent sequential patterns.

Proposed Methodology

In our proposed work, we have presented an addition in the concept of frequent pattern mining by adding an extra window to extract or mine rare patterns as well. We have used a very basic

method for mining of rare pattern which uses an extra range defined as *minimum rare support* and *maximum rare support* which creates an extra window and extracts rare pattern as well. The consideration on how it works is simple.

Rare patterns are those patterns whose support is low as well as whose importance is high as a useful pattern. In CMSPAM or CMSPADE, only frequent patterns are mined. Thus all the infrequent patterns whose support is less than minsup are pruned or neglected. For a given value of minsup we can define a range of support in which the itemsets can be considered as rare since their support is not very less as well as their support is not very high. This method is based on the property that rare items have low support but not lower than a threshold value. We used this method to generate a window whose size and starting point i.e. minimum rare support is dependent on the user input of minimum support.

Proposed Methodology to find rare itemsets:

a)Steps of proposed methodology for CMSPAM :

Step1. We calculate our minimum support as a function of size of the dataset and the minsupRel i.e relative minimum support.

```
minsup = (int) Math.ceil((minsupRel * sequencesSize.size()));
```
...Eq:1

Step2. The value of minraraesup and maxraresup is dependent on minsup.

i. `minraresup = (minsup == 1) ? 1 : (int)(Math.floor(minsup/5));` ...Eq:2

ii. `maxraresup = (minraresup == minsup)?minraresup :(int)(Math.floor(minraresup/1.25) +minraresup);` ...Eq:3

b)Steps of proposed methodology for CMSPADE :

Similar to the CMSPAM, CMSPADE also extracts frequent and rare item sets , but with the help of minSupAbsolute and minSupRelative.

Step1. `this.minSupAbsolute = (int) Math.ceil(minSupRelative * database.size());` ...Eq:4

```
if (this.minSupAbsolute == 0) {
    this.minSupAbsolute = 1;
}
```

Step 2. minRareSup = (minSupAbsolute == 1) ? 1 : (int)(Math.*floor*(minSupAbsolute/5));...Eq:5

```
maxRareSup = (minRareSup == minSupAbsolute) ? minRareSup :
(int)(Math.floor(minRareSup/1.25) + minRareSup);
```
...Eq:6

EXPERIMENTAL RESULTS

To verify the concept proposed, we have conducted an experiment in which we took CMSPAM and CMSPADE & their modified versions and ran them on a dataset BMS1.text and generated the results. The results show the information like runtime, rare patterns extracted, *minimum rare support* and rare support window range with respect to *minsup*. The value of minsupRel is varied in the main algorithm, which will therefore vary the value of window.

The value of minsupRel is varied from 0.01(597) to 0.045(2683) and time is calculated in ms(milli- sec).

The results show that while the algorithm works perfectly for both CMSPAM and CMSPADE and extracts frequent patterns with a window which both moves and expands linearly with linear increase in *minsup*. It also shows that while algorithm is comparably faster for CMSPAM, it is really slow when compared with CMSPADE.

Round	MinSup	RareSeq	Time(ms)(New)	MinRareSup	MaxRareSup	WinSize
1	597	190	1997	119	214	95
2	895	73	1553	179	322	143
3	1193	40	1477	238	428	190
4	1491	14	1245	298	536	238
5	1789	9	1184	357	642	285
6	2087	6	1191	417	750	333

7	2385	2	1091	477	858	381
8	2683	1	1078	536	964	428

Table 1: Minimum Support & Rare Patterns extracted with time and window size for our proposed methodology with CMSPAM

Round	MinSup	RareSeq	FreqSeq	Time(ms)(Base)
1	597	0	77	1235
2	895	0	36	1179
3	1193	0	22	1036
4	1491	0	13	1005
5	1789	0	11	1032
6	2087	0	7	950
7	2385	0	5	970
8	2683	0	5	1051

Table 2: Minimum Support & Rare Patterns extracted with time and Frequent Patterns extracted for CMSPAM

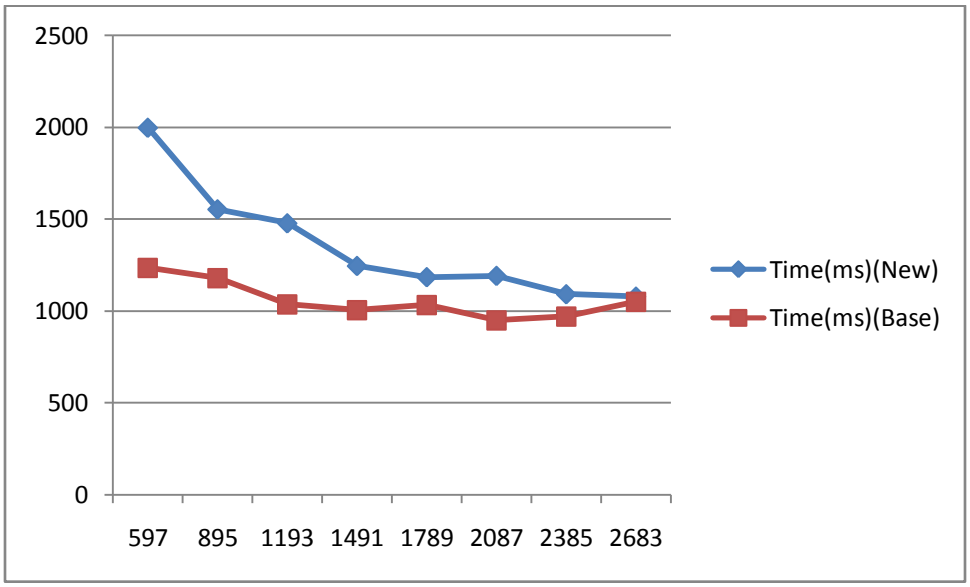

Fig 1: Showing the Runtime of CMSPAM & Updated-CMSPAM Vs *minsup*

Round	MinSup	RareSeq	FreqSeq(Base)	Time(ms)(Base)
1	597	0	77	440
2	895	0	36	300
3	1193	0	22	231
4	1491	0	13	205
5	1789	0	11	185
6	2087	0	7	170
7	2385	0	5	150
8	2683	0	5	130

Table 3: Minimum Support & Rare Patterns extracted with time and Frequent Patterns extracted for CMSPADE

Round	MinSup	RareSeq	Time(ms)(New)	MinRareSup	MaxRareSup	WinSize
1	597	182	2372	119	214	95
2	895	73	1688	179	322	143
3	1193	40	1217	238	428	190
4	1491	14	1270	298	536	238
5	1789	9	980	357	642	285
6	2087	6	839	417	750	333
7	2385	2	705	477	858	381
8	2683	1	632	536	964	428

Table 4: Minimum Support & Rare Patterns extracted with time and window size for our proposed methodology with CMSPADE

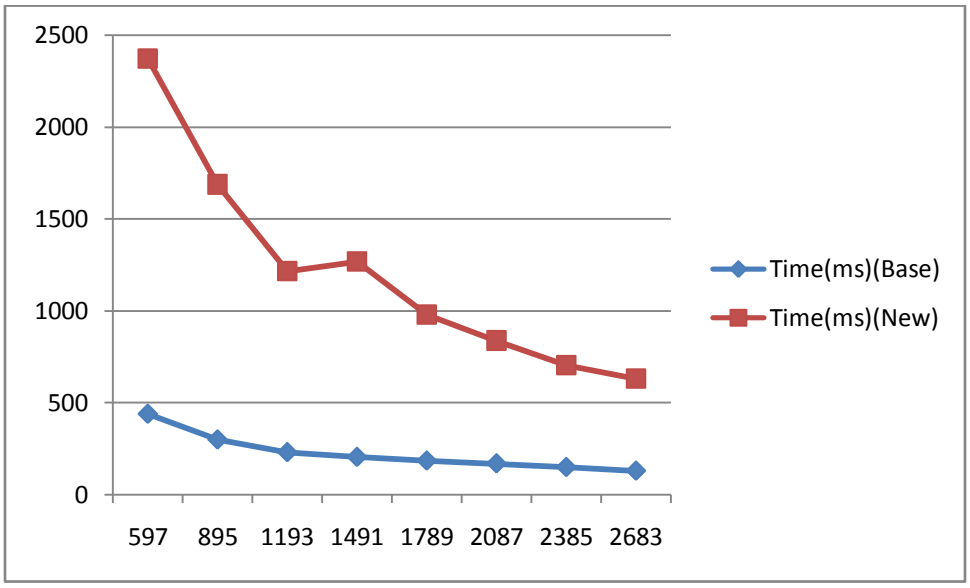

Fig 2: Showing the Runtime of CMSPADE & CMSPADE-Updated Vs *minsup* with our proposed method.

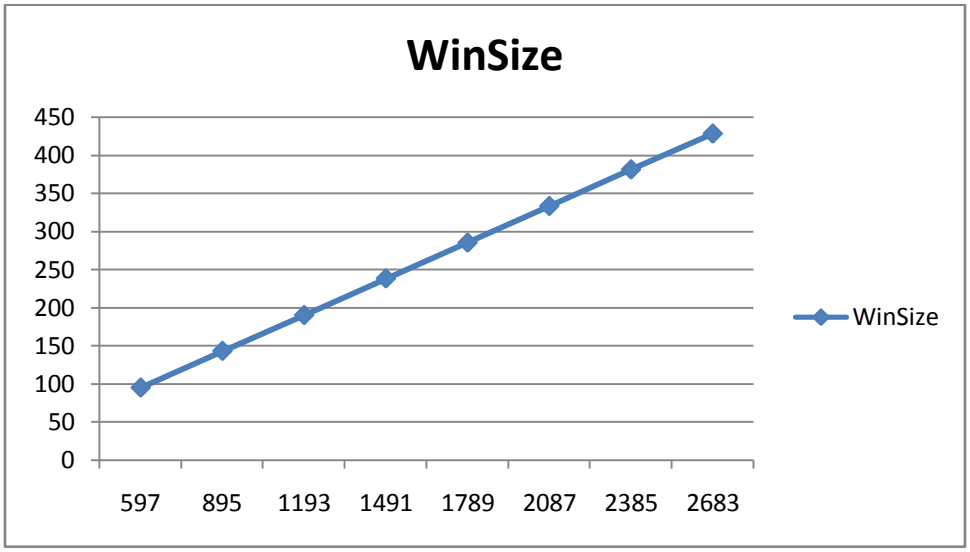

Fig 3: Showing the trend of window size variation with variation in *minsup* in the proposed method.

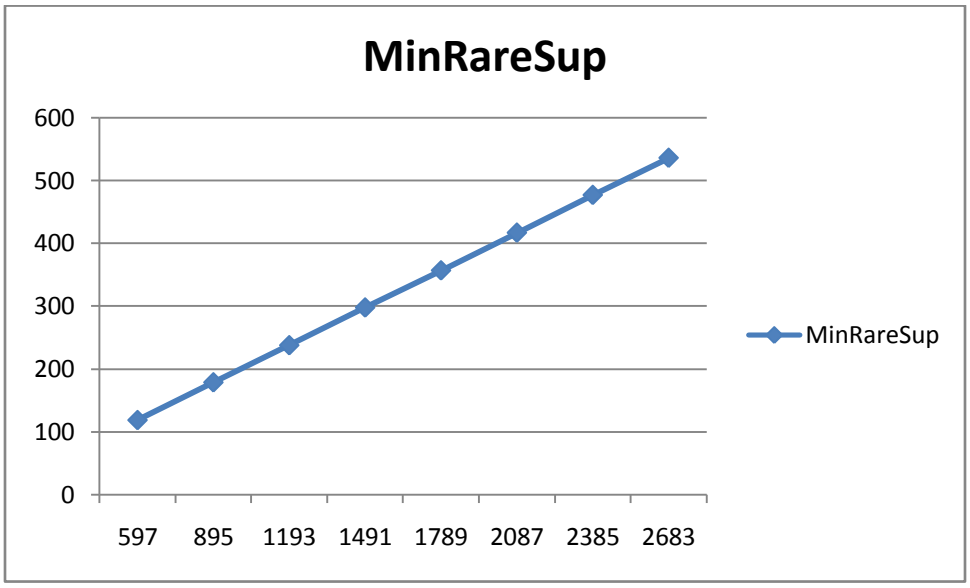

Fig 4: Showing the trend of *minimum rare support* variation with variation in *minsup* in the proposed method.

CONCLUSION & FUTURE WORK

Sequential Pattern mining algorithms using the vertical format are very efficient as they can calculate the candidate patterns by avoiding costly database scans. In this paper we have added a range of support which is less than minsup that will help in finding rare itemsets. So, CM based pruning in the state of the art vertical sequential algorithms minsup will find the frequent sequential patterns as minraresup and maxraresup range will extract the rare sequential patterns as well.

So, by using this improvised methodology of adding extra support window we are able to find rare frequent patterns as well. The basic use of rare items comes handy when the duration of database taken is wide like an year or two where finding rare items cannot be ignored for the development and progress in various sectors.

In this paper we have proposed a simple way to extract rare patterns using a moving window. The algorithm is applied on both CMSPAM and CMSPADE and the results were generated. The algorithm performs well in both cases and extracts rare patterns effectively. While the runtime is increased, if compared to CMSPAM original, our algorithm is faster but when CMSPADE comes to play, the algorithm is relatively slow. We can analyze the property which has caused the runtime of CMSPADE to deteriorate and work on it to make the algorithm faster too.

ACKNOWLEDGEMENT

I am thankful to Mrs. Sarika Chaudhary, Assistant Prof, ASET, Mrs. Neha Bishnoi, Assistant Prof, ASET for his constant guidance and encouragement provided in this endeavour. I also thanks my parents for their continuous support, understanding and patience without whose support and understanding this endeavour would never been fruitful. I also thanks all my friends for helping me out in completing this project and helping me in solving various problems encountered during the progress of this project.

REFERENCES

[1] Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas (2014), "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information": Springer International Publishing Switzerland 2014.

[2] Prof. Alpa Reshamwala, Ms. Neha (2014)Analysis of Sequential Pattern Mining Algorithms Mishra International Journal of Scientific & Engineering Research, Volume 5, Issue 2, February-2014.

[3] Gomariz, A., Campos, M., Marin, R., Goethals, B (2013), ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences. PAKDD 2013, Part I. LNCS, vol. 7818, pp. 50–61. Springer, Heidelberg (2013)

[4] Aseervatham, S., Osmani, A., Viennet, E (2006), bitSPADE: A Lattice-based Sequential Pattern Mining Algorithm Using Bitmap Representation. In: Proc. 6th Intern. Conf. Data Mining, pp. 792–797. IEEE (2006).

[5] Anurag Choubey, Ravindra Patel, J.L. Rana. (2011)"A Survey of Efficient Algorithms and New Approach for Fast Discovery of Frequent Itemset for Association Rule Mining(DFIARM)". 2011. International Journal of Soft Computing and Engineering (IJSCE)

[6]] Han, Jiawei, Hong Cheng, Dong Xin, and Xifeng Yan. "Frequent pattern mining: current status and future directions." Data Mining and Knowledge Discovery 15, no. 1 (2007)..

[7] Y Hirate (2006), Generalized Sequential Pattern Mining with Item Intervals. JOURNAL OF COMPUTERS, VOL. 1, NO. 3, JUNE 2006.

[8] Yu HIRATE and Hayato YAMANA (2006), "On generalizing of Sequential Pattern Mining with Time Intervals," Proc. of the 17th IEICE Data Engineering Workshop (DEWS2006) (2006).

[9] Han, J., Kamber, M.: Data Mining(2006), Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco

[10] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M(2004), Mining sequential patterns by pattern-growth: the PrefixSpan approach. IEEE Trans. Knowledge Data Engineering 16(11), 1424–1440 (2004)

[11] Ming-Yen Lin and Suh-Yin-Lee (2004), Efficient Mining of sequential patterns with time constraints by delimited pattern growth.

[12] Xifeng Yan , Jiawei Han , Ramin Afshaf (2003), CloSpan: Mining Closed Sequential Patterns in Large Datasets (2003).

[13] Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation(bitmap SPAM). In: Proc. 8th ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining, pp. 429–435. ACM (2002).

[14] Jian Pei Jiawei Han ,Wei Wang:Mining Sequential Patterns with Constraints in Large Databases (2002)

[15] Zaki, M.J (2001), SPADE: An efficient algorithm for mining frequent sequences. MachineLearning 42(1), 31–60 (2001).

[16] Agrawal, R., Imielinski, T., and Swami, A(1993), "Mining association rules between sets of items in large databases." SIGMOD, 1993.

[17] http://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php

[18] http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

[19] http://www.eclipse.org/documentation/