International Journal of Computer Engineering & Technology (IJCET)
Volume 7, Issue 1, Jan-Feb 2016, pp. 18-25, Article ID: IJCET_07_01_003
Available online at
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=1
Journal Impact Factor (2016): 9.3590 (Calculated by GISI) www.jifactor.com
ISSN Print: 0976-6367 and ISSN Online: 0976–6375

A NOVEL APPROACH TO MINE FREQUENT PATTERNS FROM LARGE VOLUME OF DATASET USING MDL REDUCTION ALGORITHM

P. Alagesh Kannan
Assistant Professor, Department of Computer Science,
MKU College, Madurai, Tamil Nadu, India

Dr. E. Ramaraj
Professor, Department of Computer Science & Engg.,
Alagappa University, Karaikudi, Tamil Nadu, India

ABSTRACT

In this paper, MDL based reduction in frequent pattern is presented. The ideal outcome of any pattern mining process is to explore the data in new insights. And also, we need to eliminate the non-interesting patterns that describe noise. The major problem in frequent pattern mining is to identify the interesting patterns. Instead of performing association rule mining on all the frequent item sets, it is feasible to select a sub set of frequent item sets and perform the mining task. Selecting a small set of frequent item sets from large amount of interesting ones is a difficult task. In our approach, MDL based algorithm is used for reducing the number of frequent item sets to be used for association rule mining is presented. MDL based approach provides good reduction of frequent patterns on all types of data such as sequences and trees. Experimental results show that reductions up to three orders of magnitude is achieved when MLD algorithm is used.

Key words: Frequent item sets, Pattern Mining, MDL, Minimum Description Length, Interestingness, Data Mining, Association Rule Mining and ARM

Cite this Article: P. Alagesh Kannan and Dr. E. Ramaraj. A Novel Approach to Mine Frequent Patterns from Large Volume of Dataset using MDL Reduction Algorithm. *International Journal of Computer Engineering and Technology*, **7**(1), 2016, pp. 18-25.
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=1

1. INTRODUCTION

Set of items, sequences or structures that appear frequently in a data set are said to be frequent patterns. The frequency of appearing should not be less than the user specified threshold [1]. For example, if milk and bread appears frequently together in a transaction, then these two items (known as item set) is a frequent item set. Frequent patterns are the set of items, sub sequences, structures that occur frequently in a data set[1].

Frequent item set plays an important role in many data mining applications. Frequent item sets are used to find interesting patterns from databases, classifiers, clusters, sequences, correlations, episodes, etc[2]. The term frequent item set was first coined by Agarwal et al, 1993 to analyze the customer behaviour during shopping, leading to a famous market analysis problem called market basket analysis[2]. Frequent pattern mining is also having wide applications in cross marketing, catalogue design, campaign analysis, DNA sequence analysis, web log analysis, etc.[3]

Finding relationships between different items that customers place in their cart helps to increase the sales by helping retailers to do selective marketing and arrange their items as per customer choice[3].

2. RELATED WORK

Tanna et al [4] proposed frequent pattern mining based on apriori algorithm. Apriori is the basic mining algorithm used for mining frequent patterns. Apriori algorithm reduces number of database scans to extract frequent patterns. The algorithm finds possible item sets and terminates when no further successful extensions are found. Apriori algorithm uses bread-first search strategy and tree structure for counting candidate item sets.

Cornelia Gyorodi et al [5] presented a comparative study on association rules mining algorithms. The comparison was made between classical frequent pattern mining algorithms which uses candidate set generation and algorithms without candidate set generation. A representative algorithm for both categories such as the Apriori, FP-growth and DynFP-growth was chosen. The experiments were conducted on these data and it can be concluded that the DynFP-growth algorithm is superior than FP-growth algorithm. FP-growth algorithm needs at most two scans of database whereas the candidate generation algorithm (Apriori) increases the number of scans proportional to the dimensions of the candidate itemsets.

Hui Cao et al[6] proposed frequent pattern mining algorithms based on partition method which divides the database into number of non-overlapping partitions. Frequent item sets local to the partition are generated for each partition. Partition algorithm need minimum of two database scans with generation of frequent item sets in the first scan and generating global item sets in the second scan[1]. In partition algorithms, a special data structure called TIDLIST is used which contains transaction IDs of all the transactions corresponding to an item set in the partition[1].

Kanakubo, M and Hagiwara [7] proposed frequent pattering mining based on Sampling algorithm which picks random samples from the database and tries to find frequent item sets in the samples. Finding frequent item sets is based on using support that is less than the user specified minimum support for the database. Then the algorithm also finds candidate item sets that did not satisfy minimum support. Performance of this algorithm relies on the quality of the sample chosen.

Chin-Chen Chang et al [8] proposed an efficient algorithm for incremental mining of frequent patterns. Incremental algorithms can manipulate earlier mining to get final mining outputs. The algorithm uses backward approach and scanning incremental database. Instead of scanning original database for frequent item sets, occurrence counts of newly generated frequent item sets are accumulated and infrequent item sets are deleted. The running time of NFUP is directly proportional to transaction number of incremental database.

Ya Han Hu and Yen Liang Chen [9] proposed an algorithm for mining association rules with multiple minimum supports. The algorithm is the improvement of traditional apriori based MSapriori (Minimum Support) algorithm proposed by Liu et a l[5]. The proposed algorithm is two fold : with MIS-tree construction to store the crucial information about frequent patterns in the first step. In the second step, appropriate thresholds for all items at a time are set. Generally, users tune item supports and run the mining algorithm repeatedly till a satisfactory value is reached.

Farah Hanna Al-Zawaidah et al [10] proposed an improved algorithm for mining association rules in large databases. Key challenge in developing association rule mining algorithm is that rules generated in extremely large databases makes algorithm inefficient. Further, understanding the generated rules by the end users is difficult. The algorithm presented is derived from conventional apriori approach with additional features.

A.Zemirline et al [11] proposed an efficient association rule mining algorithm for classification. The algorithm name is Association Rule Mining algorithm for classification (ARMC) and it extracts the set of rules, specific to each class. The algorithm uses fuzzy approach to select the items and it does not require the user to provide thresholds. This algorithm contain different features like covering all training instances and leaving no unclassified instances, requires only one pass to discover rules and uses novel model for building classification model. The quality features of this algorithm are not available in traditional associative classification methods.

The only problem with all the above methods is that all the patterns are analyzed for satisfying some interesting measures. It will be good if small set of non redundant interesting patterns can be selected and analyzed. This avoids the pattern explosion problem. In other words, the best set of patterns is selected for performing association rule mining. Minimum Description Length (MDL) approach is used for selecting best set of patterns[12]. Hence this paper presents a novel method of frequent pattern mining approach using MDL algorithm.

The paper is organized as follows: Section 1 provides introduction to frequent pattern mining, section 2 provides literature survey and section 3 provides background work related to our research such as MDL approach, code table, problem definition and ordering patterns. Section 4 provides detailed experimental results, section 5 describes conclusion and paper finishes by defining the list of papers referenced in this research work.

3. BACKGROUND

3.1 Minimum Description Length

Minimum Description Length (MDL) is close to Minimum Message Length (MML) which is a practical version of Kolmogorov complexity [13]. Developed by Li and Vitanyi, MDL provides a generic solution to the model selection problem. Let H = $\{I_1, I_2, I_3 \ldots\ldots I_n\}$ be the set of patterns from the data set D. The best set of patterns Bp is the one which minimizes the sum of L(H,D) = L(H) + L(D|H) where L(H) is the length of the description and L(D|H) is the length of the description when encoded [13].

3.2 Code Tables

The basic of MDL principle relies on code table. This table contains two columns with first one containing patterns and second column defining codes relevant to that pattern[14]. The two basic assumptions that are used in code tables are :-

1. Each code table must contain a single pattern
2. The pattern entries are ordered

Let Db be the structured database, e be the element in Db. Let CT represents code table. The code table for the pattern p_i, where $p_i \subseteq e$, all occurrences are replaced by C_i and p_i. The total number of occurrences is the frequency of the pattern Db and the length is replaced by l, where l=(CT,Db)(p_i). This strategy contains certain properties:

1. Each element 'e' of the structured database Db is covered by non-overlapping patterns.
2. There is a great distinction between code table entries and database cover.
3. The algorithm will terminate when code table contains single pattern.

3.3 Problem Definition

Give a database D and code table CT, the frequency of a pattern pi, which is the number of times it covers a database element is represented by freq(p_i). The relative frequency of a pattern p_i is given by

$$-\log\left(\frac{freq(p_i)}{\sum_{p_j} freq(p_j)}\right)$$

In order to apply MDL principle, we have to determine the size of the code table. We know that initial code table contains only singleton patterns. If the patterns are arranged in descending order based on support value in the database, the resulting table is the standard code table. The size of the code table is computed as

Size of the code Table = $\sum_{p_i,\, freq(p_i)} l(_{ST,Db})(p_i) + l(_{CT,Db})(_{C_i})$

3.4 Ordering of Patterns

Let P = {p_1, p_2, p_3, …. p_n} be the set of frequent patterns and CT as code table. The patterns are entered in the code table in an ordered manner as follows:

1. If p_1 is bigger than p_2, then p_1 will enter before p_2. In other words, p_1 will have longer sequence.
2. If p_1 and p_2 have the equal size but p_1 is having larger support in the database, then p_1 will enter before p_2.
3. If both the measures are same, then the order can be random.

The next duty is to compress the patterns in the code table. The following algorithm is used.

```
Procedure Compress (P, CT, D, Dsize)
    // P represents set of patterns, CT represents Code Table,
    // D represents the database and Dsize is the size of the database//
    CT = {singleton patterns}
    // Initially code table contains only singleton patterns//
    minDsize = ComputeSize(CT);
    for each pi in P
    {
            CT.add(pi); // in its place //
            Dsize = ComputeSize(CT);
    If (Dsize < minDsize)
            minDsize = Dsize;
    else
            CT.remove(pi);
    }
return CT
end
```

In the above procedure, ComputeSize routine computes the size of MDL i.e. For this, it computes cover first and then computes the size of the code table. If the new size is smaller than the minimal size, the pattern is allowed to be in the code table else it is removed.

Further more, the code table is reduced by pruning the code table tree. This is done by applying greedy pruning algorithm which starts from the bottom of the table to remove non-contributing smallest patterns. This reduces frequent patterns to large extent. Reduction is based on re-computing the cover. The pattern is removed if the re-computed results are better else it is reinserted. In this fashion, all the patterns in the code table are visited. The following procedure prunes the code table tree.

```
Procedure PRUNE (CT, DSize)
    // CT represents Code Table, Dsize is the size of the database//
for each code in CT
    {
            code table.remove(code);
            newsDsize = ComputeSize(CT);
            if (newDsize ≤ minDsize)
            minDsize = newDsize;
            else
            CT.add(code);
            endif
    }
return CT
end
```

4. EXPERIMENTAL RESULTS

KDD Cup is one among the leading knowledge discovery competitions in the world which is organized by ACM SIGKDD. Hence KDD Cup 2000 data set is used for our experiment. It consists of click streams and customer data of e-commerce retail website. It contains around 777,780 clicks divided over 234,954 sequences. From this, code table is generated, compression is applied on frequent sequences. Finally pruning is applied to remove the items with low frequency that are left during compression stage. It is noted that around 40% of patterns are retained in compression phase and only 10% of patterns are fully available in code table after pruning phase.

4.1 Prune trees

Pruning the code table is nothing but removing reverse support order code table elements that cannot contribute database compression no more. To test the efficiency of the pruning step, logml, US304 and US2430 web log data is used[15]. The experimental results show that a huge reduction in patterns in final code table as much as 0.17% to the original code table size. The reduction ratios are functions of various support levels and different characteristics of data. The below table shows the performance of prune algorithm.

Table 1 Sequence Reduction Results

Window Size	60 sec	120 sec
minsup	02%	0.2%
No. of sequences	3,076	3,076
No. of CT	1,983	2,264
No. of CT_p	311	648

4.2 Coverage analysis

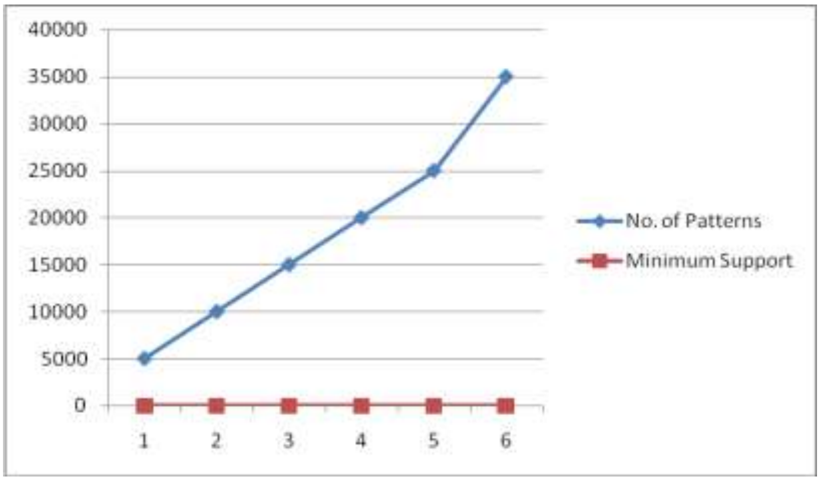
In order to further access the patterns that exists in the code table, coverage analysis is calculated. Coverage is the values that represents the frequency of a rule that can be applied or percentage of times that it can be applied[16]. Here it represents the interestingness of the final patterns and it is calculated by

$$partial\_cover(1,x) - \sum_{i=1}^{x} freq(C_i) \times l(_{CT,Db})(_{C_i})$$

and for a particular pattern, it is calculated by

$$\frac{\Delta partial\_cover(1,x)}{\Delta i} - freq(C_i) \times l(_{CT,Db})(_{C_i})$$

The experimental results show that most of the covering patterns are close to the top of the code table and these patterns will appear in early part of the evaluation state. Patterns of specific window size covers large portion of the database while reaching a specific size in code table [17-18].



5. CONCLUSION

Association rule mining using MDL principle is discussed in this paper. MDL principle is much helpful in reducing the frequent item sets size. The method is both information and useful. MDL algorithm selects small informative set of patterns from potentially large amount of set of structured frequent patterns. These reductions can be up to three orders of magnitude. The reduction in size of pattern sets is higher in low threshold levels. Moreover, code table is friendly in terms of evaluation and most interesting patterns are listed on top of the code table. Based on the experimental results, it is concluded that MDL based algorithm reduces the size of frequent sets at great level than traditional algorithms, there by achieving very good level of compression.

6. REFERENCES

[1] Jiawei Han, Hong Cheng, Dong Xin and Xifeng Yan, "Frequent pattern mining: current status and future Directions", Data Min Knowl Disc (2007) 15:55–86, DOI 10.1007/s10618-006-0059-1

[2] Vanhoof, K.; Depaire, B., "Structure of association rule classifiers: a review," Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference, vol., no., pp.9,12, 15-16 Nov. 2010 doi: 10.1109/ISKE.2010.5680784

[3] Wenxue Huang; Krneta, M.; Limin Lin; Jianhong Wu, "Association Bundle - A New Pattern for Association Analysis," Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference, pp.601,605, Dec. 2006 doi: 10.1109/ICDMW.2006.33

[4] Paresh Tanna and Yogesh Ghodasara, "Foundation for Frequent Pattern Mining Algorithms' Implementation", International Journal of Computer Trends and Technology(IJCTT) – Volume 4 Issue 7 -July2013

[5] Cornelia Gyorodi, Robert Gyorodi and Stefan Holban, "A Comparative Study of Association Rules Mining Algorithms", Department of Computer Science, University of Oradea, Str. Armatei Romane 5

[6] Hui Cao; Gangquan Si; Yanbin Zhang; Lixin Jia, "A density-based quantitative attribute partition algorithm for association rule mining on industrial database," American Control Conference, 2008 , vol., no., pp.75,80, 11-13 June 2008

[7] Kanakubo, M.; Hagiwara, M., "Speed-up Technique for Association Rule Mining Based on an Artificial Life Algorithm," Granular Computing, 2007. GRC 2007. IEEE International Conference on , vol., no., pp.318,318, 2-4 Nov. 2007

[8] Jong Soo Park , Ming-Syan Chen , Philip S. Yu, Efficient parallel data mining for association rules, Proceedings of the fourth international conference on Information and knowledge management, p.31-36, November 29-December 02, 1995,

[9] Ya-Han Hu, Yen-Liang Chen, Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism, Decision Support Systems, Volume 42, Issue 1, October 2006, Pages 1-24, ISSN 0167-9236,

[10] Farah Hanna AL-Zawaidah, Yosef Hasan Jbara, Marwan AL-Abed Abu-Zanona, " An Improved Algorithm for Mining Association Rules in Large Databases", World of Computer Science and Information Technology Journal, Vol. 1, No. 7, pp. 311-316, 2011.

[11] A.Zemirline, Lecornu, B.Solaiman, and A. Echcherif, "An Efficient Association Rule Mining Algorithm for Classification ", L. Rutkowski et al. (Eds.): ICAISC 2008, LNAI 5097, pp. 717 728, 2008. Springer, Verlag Berlin Heidelberg 2008

[12] Kudo, M.; Shimbo, M., "Selection of classifiers based on the MDL principle using the VC dimension," Pattern Recognition, 1996., Proceedings of the 13th International Conference on , vol.2, no., pp.886,890 vol.2, 25-29 Aug 1996 doi: 10.1109/ICPR.1996.547203

[13] Peter Grünwald " Introducing the Minimum Description Length Principle", Centrum voor Wiskunde en Informatica, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

[14] Michael Y. Levin Benjamin C. Pierce, "Type-based Optimization for Regular Patterns", Technical Report MS-CIS-05-13 Department of Computer and Information Science University of Pennsylvania

[15] Khabbaz, M.; Kianmehr, K.; Alhajj, R., "Employing Structural and Textual Feature Extraction for Semistructured Document Classification," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions vol.42, no.6, pp.1566, 1578, Nov.2012, doi : 10.1109/TSMCC.2012.2208102

[16] Palacios, M.; Garcia-Fanjul, J.; Tuya, J.; Spanoudakis, G., "Coverage-based testing for Service Level Agreements," Services Computing, IEEE Transactions vol.PP, no.99, pp.1,1 doi: 10.1109/TSC.2014.2300486

[17] Y. Angeline Christobel, P. Sivaprakasam. Improving the Performance of K-Nearest Neighbor Algorithm for the Classification of Diabetes Dataset with Missing Values. *International Journal of Computer Engineering and Technology*, **3**(3), 2012, pp. 155-167.

[18] Shailesh Singh Panwar and Dr. Y. P. Raiwani. Data Reduction Techniques to Analyze NSL-KDD Dataset. *International Journal of Computer Engineering and Technology*, **5**(10), 2014, pp. 21-31.