IMPROVED FREQUENT ITEM-SETS MINING IN PHARMACOVIGILANCE

Kamatchi Sankar, Latha Parthiban

Abstract: ***Mining frequent item-sets is an important concept that deals with fundamental and initial task of data mining. Apriori is the most popular and frequently used algorithm for finding frequent item-sets which is preferred over other algorithms like FP-growth due to its simplicity. For improving the time efficiency of Apriori algorithms, Jiemin Zheng introduced Bit-Apriori algorithm with the enhancement of support count and special equal support pruning with respect to Apriori algorithm. In this paper, a novel Bit-Apriori algorithm, that deletes infrequent items during trie2 and subsequent tries are proposed which can be used in pharmacovigilance to identify the adverse event.***

Index Terms**:** *Pharmacovigilance, Data mining, Adverse Events*

I. INTRODUCTION

Data mining deals with extracting meaningful patterns from huge *data*sets using hybridized method involving computational intelligence, artificial intelligence. Therefore, the study of frequent item-sets mining is important in frequent pattern mining . An attempt is made in the present work to prune unimportant patterns in the item-sets mining.

II. LITERATURE REVIEW

An important challenge for medical industry in developing new drug is the Adverse Drug Reaction (ADR) which is the fourth leading cause for death. The study of the adverse drug reactions of the newly released drugs is called Pharmacovigilance which educate the people on the benefit and risk of drugs and warn them. Data mining algorithms are very useful in finding commonly occurring ADRs due to the drugs. The pseudo code for Apriori is given below.

1.1. The psuedocode for apriori algorithm

Apriori (T, ε)

$L1 \leftarrow \{\text{large } 1-\text{item sets that appear in more than } \mathcal{E} \text{ transactions}\}$
$k \leftarrow 2$
while $L_{K-1} \neq \emptyset$
$C_k \leftarrow \text{Generate } (L_{k-1})$
for transactions $t \in T$
$C_t \leftarrow \text{Subset}(C_k, t)$
for candidates $c \in C_t$
$\text{count}[c] \leftarrow \text{count}[c] + 1$
$L_k \leftarrow \{ c \in C_k \mid \text{count}[c] \geq \varepsilon \}$
$k \leftarrow k + 1$
return $\bigcup_k L_k$

1.2. The psuedocode for bit- apriori algorithm

Bit-Apriori uses the data structure and techniques of Apriori [1] algorithm. The pseudo code for Bit-Apriori is shown below.

Input: datasetD, min _sup
Output: frequent itemsets
1: Scan database D once and delete infrequent items;
2: Sort frequent single items into non decreasing order by supports
3: Scan database again and find all frequent 2 itemset, then establish the Trie with the binary string in each leaf;
4: k ← 3;
5: while the height of the Trie is increased do

Revised Manuscript Received on July 05, 2019.

Kamatchi Sankar, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Tamilnadu, India.
Latha Parthiban, Department of Computer Science, Pondicherry University CC, Puducherry, India

6: *generation*(k);

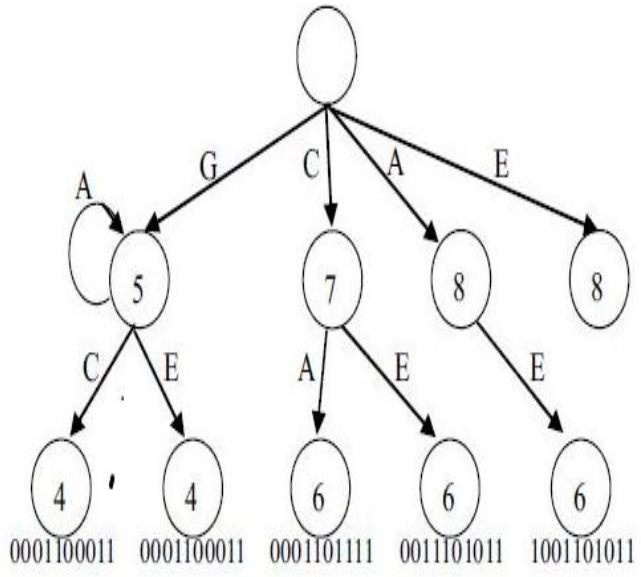
7: $k \leftarrow k+1$

8: *end while*

9: *write out the frequent itemsets from Trie*;

Using the algorithm, a trie with binary string in each leaf is established, except for non-frequent ones is shown in below

Fig. 1. Trie At Second Generation



During the consequent iterations, element 'E' can be ignored by considering it as non-frequent item set. The computation time can be considerably reduced when the non- frequent item sets are removed.

III. PROBLEM STATEMENT

To find out frequent item-sets, both Apriori[3] and Bit-Apriori[1] algorithms used to search elements in the entire item-sets starting from 1 to N. When the total support count for an item is zero or lesser than the support count, then the elements are not required for the consecutive iterations. While forming tires Apriori and Bit-Apriori algorithms are considering these elements. Hence there is a scope for improvement by eliminating such items during tries formation. An enhanced algorithm is proposed to improve the performance, resource utilization, time and efficiency.

The adverse events that happened because of medical drugs and medication errors are collected from around the world and stored in a database called FAERS. The adverse reaction due to specific drug is extracted from this database and preprocessed for analysis using data mining techniques. On applying the proposed algorithm the frequently occurring adverse event can easily be identified and appropriate remedy can be taken

IV. PROPOSED ALGORITHM

A new algorithm has been developed which deletes the infrequent items during the trie2 and subsequent iterations. The removal of infrequent items results in improvement in computation time. The pseudo code for the proposed algorithm is shown in Table 1.

Table 1 Psuedocode For Proposed Algorithm

Input*: *dataset D*, min _*sup

Output*: *frequent item sets

1: *Scan database D once and delete infrequent items*;

2: *Sort frequent single items into non decreasing order by supports*

3: *Delete the items of total support < support*;

4: *Scan database again and find all frequent 2 itemset, then establish the Trie with the binary string in each leaf*;

5: $k \leftarrow 3$;

6: *while the height of the Trie is increased do*

7: *generation*(k);

8: $k \leftarrow k+1$

9: *end while*

10: *write out the frequent itemsets from Trie*;

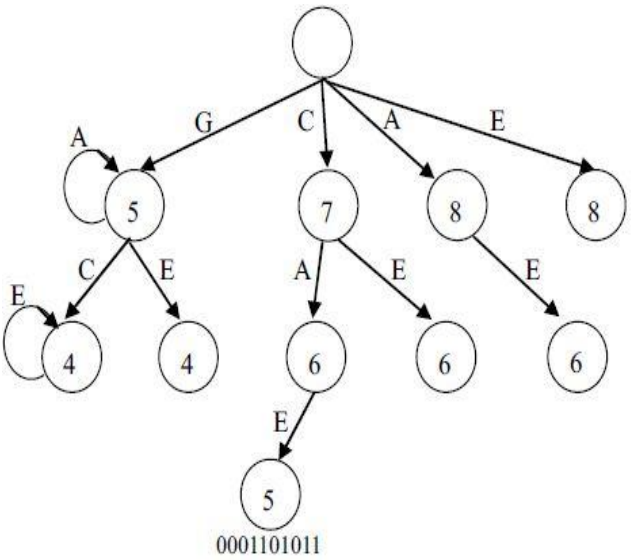


Fig. 2. Trie After Completion

V. EXPERIMENTAL RESULTS

The algorithm designed is tested and the experimental results are shown in Fig. 3.

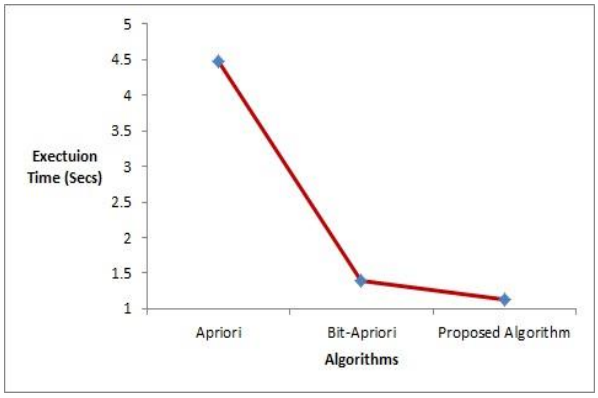


Fig. 3. Execution Time Of Algorithms

The proposed algorithm consumes considerably a lesser amount of time compared to Bit-Apriori and Apriori algorithms. Interesting finding is that, when the occurrence of non-frequent item-sets is higher, then the execution time gets reduced drastically. The experimental result shows that the proposed algorithm not only decreases the computation time but also decreases the resources used.

The algorithm designed is tested and the experimental results are shown in Fig. 3.

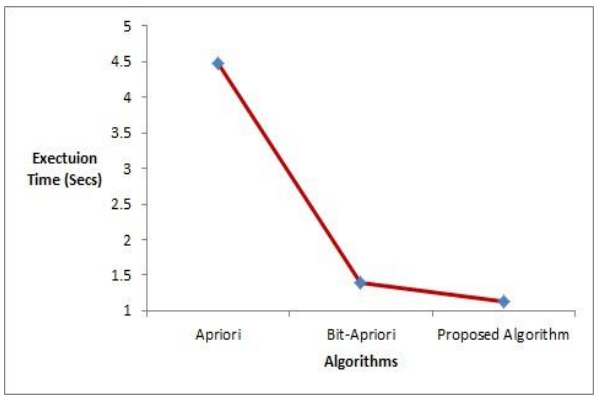


Fig. 3. Execution Time Of Algorithms

The proposed algorithm consumes considerably a lesser amount of time compared to Bit-Apriori and Apriori algorithms. Interesting finding is that, when the occurrence of non-frequent item-sets is higher, then the execution time gets reduced drastically. The experimental result shows that the proposed algorithm not only decreases the computation time but also decreases the resources used.

VI. CONCLUSIONS

In this paper, the modified Bit-Apriori technique improves the performance of Bit-Apriori, by eliminating the search of infrequent item-sets. It also improves the computational efficiency. Experimental results prove that modified Bit-Apriori algorithm outperforms the fast Bit-Apriori, especially when the occurrence of the non-frequent item-sets is more. When the database is large, the Bit-Apriori may suffer from the problem of memory scarcity due to large number of bitwise operations. Future work can be done in the direction of replacing bitwise operations.

Acknowledgments

The authors would like to acknowledge BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH for their support in doing this work.

References

1. Jiemin Zheng., Defu Zhang, Stephen C.H.Leung , Xiyue Zhou, "*An efficient algorithm for frequent itemsets in data mining*" in International Conference on Service Systems and Service Management(ICSSSM), 2010 [t]: 28-30 June 2010.

2. Agrwal R.,R.Srikant, "*Fast algorithms for mining association rules*", The International Conference on Very Large Dabases, pp. 487-499, 1994.

3. Zaki M.J., S. Parthasarathy, M.Ogihara, W.Li," *New algorithms for fast discovery of association rules*", in Proceedings of the 3rdInternational Conference on Knowledge Discovery and Data Mining, pp. 283-296,1997.

4. Han J., J. Pei, Y. Yin, "*Mining frequent patterns without candidate generation*" in Proceedings of the2000 ACM SIGMOD international conference on Management of data, ACM Press, pp. 1-12,2000.

5. Pork J.S., M.S. Chen, P.S. Yu, "*An effective hash based algorithm for mining association rules*" ACM SIGMOD, pp. 175-186, 1995.

6. Brin S., R. Motwani, J.D. Ullman, S. Tsur,"*Dynamic itemset counting and implication rules for market basket data*",inProceedings of the ACMSIGMOD International Conference on Management of Data, pp. 255–264, 1997.

7. Brin S., R. Motwani, C. Silverstein, "*Beyond market baskets: generalizing association rules to correlations*", in Proceedings of the ACM SIGMOD International Conference on Management of Data,Tuscon, Arizona, pp. 265-276, 1997.

8. Toivonen H., "*Sampling large databases for association rules*", in Proceedings of 22ndVLDBConference, Mumbai, India, pp. 134-145, 1996.

9. Savasere A., E. Omiecinski, S.B. Navathe, "*An efficient algorithm for mining association rules in large databases*", in Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95), Zurich, pp. 432-444, 1995.

10. Tsay Y.J., J.Y. Chiang, "*CBAR: an efficient method formining association rules*," Knowledge Based Systems, 18 (2-3), pp. 99-105, 2005.

11. Liu G., H. Lu, W. Lou, Y. Xu, J.X. Yu, "*Efficient mining of frequent patterns using ascending frequency Ordered prefix-tree*", Data Mining and Knowledge Discovery, 9 (3), pp. 249-274, 2004.

12. Grahne G., J. Zhu, "*Fast algorithms for frequent itemset mining using FP-Trees*", IEEE Transaction on Knowledge and Data Engineering, 17 (10), pp.1347-1362, 2005.

13. Zaki M.J., "*Scalable algorithms for association mining*" IEEE Transactions on Knowledge and DataEngineering, 12 (3), pp. 372-390, 2000.

14. Zaki M.J., K. Gouda, "*Fast Vertical Mining Using Diffsets*", in Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, pp. 326-335, 2003.

15. Dong J., M. Han, "*BitTableFI: an efficient mining frequent itemsets algorithm*" Knowledge Based Systems, 20 (4), pp. 329-335, 2007.