International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 3 | Mar-Apr 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

Association Rule Hiding using Hash Tree

Garvit Khurana

School of Computer Science and Engineering Vellore Institute of Technology, Vellore, Tamil Nadu, India

How to cite this paper: Garvit Khurana "Association Rule Hiding using Hash Tree" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.787-789, URL: https://www.ijtsrd.com/papers/ijtsrd23037.pdf

IJTSRD23037

ABSTRACT

As extensive chronicles of information contain classified rules that must be protected before distributed, association rule hiding winds up one of basic privacy preserving data mining issues. Information sharing between two associations is ordinary in various application zones for instance business planning or marketing. Profitable overall patterns can be found from the incorporated dataset. In any case, some delicate patterns that ought to have been kept private could likewise be uncovered. Vast disclosure of touchy patterns could diminish the forceful limit of the information owner. Database outsourcing is becoming a necessary business approach in the ongoing distributed and parallel frameworks for incessant things identification.

This paper focuses on introducing a few adjustments to safeguard both customer and server privacy. Adjustment strategies like hash tree to existing APRIORI algorithm are recommended that will be helping in safeguarding the accuracy, utility loss and data privacy and result is generated in small execution time. We implement the modified algorithm to two custom datasets of different sizes.

KEYWORDS: *Association Rule Mining, Modified APRIORI, Frequent Itemset Mining, Hash Tree*

1. INTRODUCTION

Data mining expels novel and profitable learning from broad files of information and has transformed into an effective examination and decision strategies in organization. The sharing of information for data mining can bring a lot of points of interest for research and business participation; in any case, tremendous storage facilities of information contain private information and touchy rules that must be verified before distributed. Awakened by the various clashing necessities of information sharing, insurance and learning discovery, privacy preserving data mining has transformed into an examination hotspot in data mining and database security fields.

Two issues are tended to in privacy preserving data mining, one is the security of private information; another is the confirmation of sensitive rules (learning) contained in the information. The past settles how to get normal mining results when private information can't be gotten to precisely; the last settles how to guarantee delicate rules contained in the information from being found, while non-touchy principles can at present be mined routinely. The last issue is called information hiding in database in which is inverse to learning discovery in database. Emphatically, the issue of learning discovery can be portrayed as seeks after.

2. RELATED WORK

Data mining is the one of the critical thinking method takes care of numerous business arranged issues, all things considered, among association rule mining is one of the vital viewpoints for learning discovery. R. AGARWAL spoke to interested association rules among the diverse datasets. Mining successive patterns is a principal part in mining distinctive thing sets in database applications, for example, consecutive patterns and mining association rules and so on. According to specialist Sergey Brian ETAL suggested a dynamic item set counting (DIC) using APRIORI calculation to assembled extensive thing set and makes its subset likewise vast so it will increase memory and time complexity. All calculations proposed before are retrieving regular thing sets continuously using association rule mining with APRIORI calculations. Each dimension all subsets of incessant example are additionally recovered every now and again. By these calculations substantial successive patterns with candidate keys are generated. By the prior frameworks we have to filter the database continuously, consequently proficiency of mining is additionally diminished. Because of these deterrents, an analyst JIAWEI HAN proposed a calculation without generating a candidate key, by scanning the database less times, we are going to create a FP-development calculation to increase productivity contrasted with past calculations of association rule mining using APRIORI calculation. By avoiding the candidate age process and less ignores the database, FP-Tree establishes to be quicker than the APRIORI calculation. The disadvantages of using FP-mining are mining finished thing sets for which if there is an expansive incessant item sets with size X subset, nearly 2X subset of thing sets are generated consequently. Anyway to producing a huge number of contingent FP-trees

in mining the proficiency of association rule mining using FP-development is having disadvantages. In this paper we propose a hash-tree based calculation.

3. PROBLEM DEFINITION

To design and implement hash tree APRIORI algorithms in order to reduce time and memory complexity of execution and solve the integrity and security issues in distributed data.

4. PROPOSED ALGORITHM

Rule for an Efficiency Improvement

We can improve the efficiency of the APRIORI by:

1. Prune all k-1 subsets without checking it.
2. Join L k-1 subsets without looping over the entire set.
3. Speeding up matching & searching
4. Reducing the total number of transactions
5. Reducing the number of passes on data.
6. Reducing the number of subsets per transaction that are to be be considered.
7. Reducing number of candidates for frequent item set generation.

This can be done by using hash trees.

This algorithm was implemented on a Python environment with Intel 2.9 GHz Intel Core i5 processor.

The performance of the rules generated is analyzed using support and confidence.

We need support because if we use confidence only some of the rules might produce by chance. So support helps us to find item set that people seldom buy together so that we can generate association rules out of them. Confidence provides reliability of the inference that can be derived by the rule. Higher the confidence, higher its likely it is for Y to be present in the transactions that contain X.

Total possible rules:

3^d - 2^ (d + 1) + 1

X -> Y only depends upon the support of (xUy)

If support of (x U y) is less than all the 2*(|x| + |y| - 1) rules generated will waste computing power.

So problem is divided into two parts:

1. Frequent item set generation
2. Rule generation

Frequent Item set generation:

O (N*M*w)

Where, N is transactions, M is item set, w is max width of item set.

So two ways:

1. Reduce M
2. Reduce number of comparisons for finding support.

The APRIORI principle:

If an item set is frequent them all of its subsets must be frequent.

Conversely if item set is infrequent then all of its supersets are infrequent.

Support based pruning: Trimming exponential search space based on support measure.

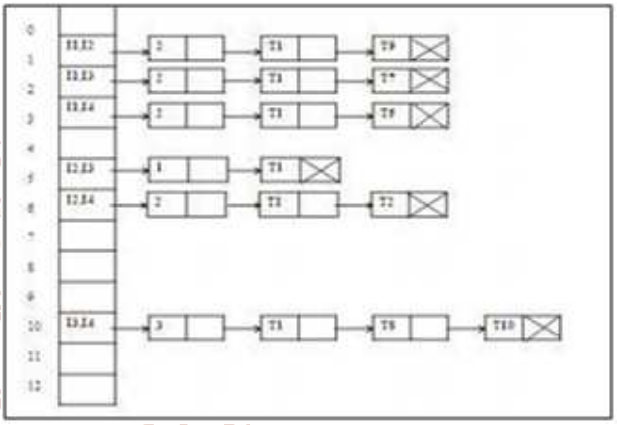
Candidate generation and pruning:

- Candidates -> Ck is set of all possible candidates.
- Fk is set of frequent candidates:

Here after APRIORI we use Hash Tree so that candidate item sets are partitioned into different buckets and stored in hash tree.

During support counting, item sets contained in each transaction are also hashed into appropriate buckets. That way instead of comparing each transaction with every candidate item set, it is matched only against candidate item set that belong to the same bucket.

This indeed helps in reducing time as well as provides security to the data



5. RESULTS AND DISCUSSION

For implementing the Modified APRIORI Algorithm, we used two custom datasets of different sizes.

*The small dataset consisted of 1000*9 random integer dataset with missing values.*

*The larger dataset consisted of 852433*3 random integer dataset without missing values.*

Dataset	Time Taken by APRIORI Algorithm	Time Taken by Modified APRIORI Algorithm
Small	0.831753969193	0.0556449890137
Large	39.800085783	6.41527199745

After implementing the algorithm in Python and comparing the results with Original Unmodified APRIORI Algorithm we see that APRIORI Algorithm with hast tree works much faster for datasets than the original one.

Hence by using the modified APRIORI algorithm using hash tree we can improve not only the security of the data but also the overall efficiency.

6. CONCLUSION

We see that computational complexity depends upon:

1. Threshold Support: Size of C increases.
2. Number of items: Size of both C, F may increase, requires more space and IO cost will increase

3. Number of transactions: Since APRIORI makes use number of passes on database
4. Average width of transactions: Increases hash tree traversals during support count phase.
5. Generation of frequent 1 item sets: O (N*w) where w is average width
6. Candidate generation:
7. Support counting:

O (N*sum (k*wCk*alpha))
Each transaction generates Ck item sets of size K and each of which requires K steps to go down the hash tree and alpha is the cost associated with updating count of candidate inside bucket.

REFERENCES

[1] J. Han, M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, San Francisco, USA, 2001, ISBN 1558604898.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," In Proc. of VLDB '94, pp. 487-499, Santiago, Chile, Sept. 1994.

[3] J. Vaidya& C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," In proc. Conf. Knowledge Discovery and Data Mining, pp. 639–644, July 2002.

[4] Komal shah, Amitthakkar & Amitganatra, "Association Rule Hiding by Heuristic Approach to Reduce Side Effects & Hide Multiple R.H.S. Items" International Journal of Computer Applications (0975 – 8887) Volume 45– No.1, May 2012.

[5] A. B. M Rezbaul Islam, Tae-Sun Chung "An Improved Pattern Tree Based Association Rule Mining Technique" IEEE International Conference on Information Science and Applications (ICISA), 201 I.

[6] Qihua Lan, Defu Zhang, Bo Wo, "A new algorithm for frequent item set mining based on APRIORI and FP-tree", Global Congress on Intelligent System 2009

[7] Bodon, F. 2005. A trie-based APRIORI implementation for mining frequent item sequences. In Proceedings 1st international workshop on open source data mining: frequent pattern mining implementations, ACM.

[8] Bodon, F. and Schmidt-Thieme, L. 2005. The relation of closed item set mining, complete pruning strategies and item ordering in APRIORI-based fim algorithms. In Knowledge Discovery in Databases: PKDD, Springer Berlin Heidelberg, 437-444.