International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

Association Rule Hiding using Hash Tree

Karan Sharma

U.G. Student, Department of Computer Science & Engineering, Galgotias University, Gr. Noida, U.P, India[1]

ABSTRACT: As broad accounts of data contain arranged standards that must be secured before appropriated, association rule hiding breezes up one of the basic privacy protection information mining issues. Data sharing between two affiliations is ordinary in different application zones for example business planning or marketing. Overall profitable pattern can be found from the consolidated dataset. Regardless, some sensitive patterns that should have been kept hidden could in like manner be revealed. Immense exposure of delicate patterns could reduce the mighty furthest reaches of the information owner. Database outsourcing is turning into an important business approach in the ongoing distributed and parallel frameworks for unremitting things identification**.**

This paper centers around introducing a couple of adjustments to safeguard both client and, server privacy. Adjustment methodologies like the hash tree to existing APRIORI algorithm are suggested that will help in protecting the accuracy, utility loss and data privacy and result is produced in little execution time. We execute the modified algorithm to two custom datasets of different sizes.

KEYWORDS*:*Association Rule Mining,Modified APRIORI, Freqeunt Dataset Mining, Hash Tree

I.INTRODUCTION

Data mining removes novel and gainful learning from wide files of data and has changed into a powerful assessment and decision strategies in the organizations. The sharing of information for data mining can bring a lot of focal points for research and business support; in any case, enormous storage of data contains private information and sensitive guidelines that must be confirmed before distributed. Awakened by the different conflicting necessities of information sharing, insurance, and learning discovery, privacy-preserving data mining has changed into an examination hotspot in data mining and database security fields.

Two issues are tended to in privacy saving data mining, one is the security of private data; another is the affirmation of touchy standards (learning) contained in the data. The past settles how to get ordinary mining results when private information can't be gotten to precisely; the last settles how to ensure sensitive guidelines contained in the information from being found, while non-touchy standards can at present be mined routinely. The last issue is called data hiding in the database in which is converse to learning discovery in the database. Unequivocally, the issue of learning disclosure can be depicted as seeks after.

II.RELATED WORK

Data mining is one of the basic reasoning strategy that takes care of various business arranged issues, all things considered, among association rule mining is one of the essential perspectives for learning revelation. R. AGARWAL addressed intrigued association rules among the diverse datasets. Mining progressive examples is an important part of mining particular thing sets in database applications, for instance, sequential pattern and mining association rules, etc. As indicated by specialist Sergey Brian ETAL recommended a dynamic item set counting(DIC) using APRIORI calculation to collected broad thing set and makes its subset similarly vast so it will increase memory and time complexity. All calculations proposed before are recovering regular thing sets constantly using association rule mining with APRIORI calculations.

Each dimension of all subsets of perpetual model is moreover recovered every now and then. By these calculations, generous progressive patterns with candidate keys are generated. By the earlier frameworks, we have to filter the database constantly, thusly capability of mining is also decreased. Due to these impediments, an analyst JIAWEI HAN proposed a calculation without generation a candidate key, by scanning the database-less times, we will make a FP-development estimation to increase the efficiency decreased with the past calculation of association rule mining using the APRIORI calculations. By dodging the candidate age process and less disregards the database, FP-Tree sets up to be speedier than the APRIORI calculations. The disadvantage of using FP-mining will be finished thing sets for which if there is an extensive unending thing sets with size X subset, about 2X subset of thing sets are created consequently. At any rate, to creating an enormous number of contingent FP-trees in mining the capability of association rule mining using FP-development is having disadvantages. In this paper, we propose a hash-tree based computation.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

III.PROBLEM STATEMENT

To design and actualize hash tree APRIORI algorithm in order to decrease time and memory complexity in the execution and illuminate the integrity and security issues in conveyed data.

IV.PROPOSED ALGORITHM

Rules for the Efficient improvement

We can improve the proficiency of the APRIORI by:

1. Prune all k-1 subsets without checking it.

2. Join L k-1 subsets without circling over the whole set.

3. Accelerating matching and looking

4. Lessening out the total number of transactions

5. Lessening the quantity of passes on data.

6. Lessening the quantity of subsets per transaction that are

to be considered.

7. Lessening the number of candidate for recurring item set generation.

This can be possible by utilizing hash trees.

This calculation was executed on a Python domain with Intel 2.9 GHz Intel Core i5 processor. The performance of the rules created is examined utilizing backing and confidence.

We need support because if we use confidence only some of the rules may create by some coincidence. So support causes us to discover item set that individuals only sometimes purchase together so that we can produce association rules out of them. Confidence gives unwavering quality of the deduction that can be inferred by the rules. Higher the confidence, higher its likely it is for Y to be present in the transaction 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) entity then the rules generated will be the waste computing power.

So the 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 recurrent then all of its subsets must be recurrent. Conversely if item set is not recurrent then all of its supersets are not recurrent.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

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

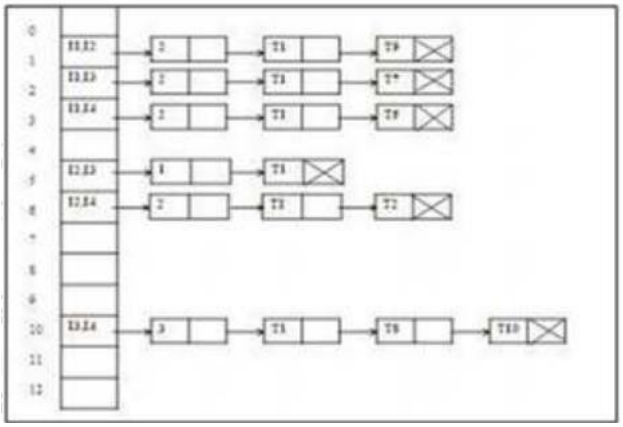
Candidate generation and pruning:

- Candidates -> Ck is set of every single imaginable candidate.
- Fk is set of regular candidate

Here after APRIORI we use Hash Tree with the goal that candidate item sets are apportioned into different bucket and stored in hash tree.

During the support counting, item sets contained in each transaction are likewise hashed into fitting buckets. That route as opposed to contrasting every transaction and each candidate item set, it is matched uniquely against candidate item set that have a place in the same bucket.

This without a doubt helps in decreasing time as well as provide security to the data.



V.RESULT AND CONCLUSION

For actualizing the Modified APRIORI Algorithm, weutilized two custom datasets of different sizes.

The small dataset comprised of 1000*9 irregular random dataset with missing qualities.

The bigger dataset comprised of 852433*3 irregular random number 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 actualizing the calculation in Python and looking at the outcomes with Original Unmodified APRIORI Algorithm we see that APRIORI Algorithm with hast tree works a lot quicker for datasets than the first one. Thus by utilizing the modified APRIORI algorithm using a hash tree, we can improve the security of the data as well as the general effectiveness.

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

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 5, May 2020||

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

[6] 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.

[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..