HASHDIC : AN EFFICIENT APPROCH FOR FREQUENT ITEMSET MINING

[1]Krishna H. Odedara, [2]Dr.Vipul Vekariya, [3]Prof.Daxa V. Vekariya
[1]P.G. Student, [2]Associate Professor, [3]Assistant Professor
[1]Computer Department,
[1]Noble group of Institution, Junagadh, India

Abstract : **Association rule mining is used for finding frequent itemset form the Transaction database. There are many efficient algorithm used for finding support of the itemset which denote that itemset is frequent or not. From that algorithm. Eclat algorithm is used intersection for the support counting but it give low efficiency when number of transaction are large. In our algorithm considers the DIC(variation of apriori) approach it reduce the number of passes made over the transaction database. DIC maintains four itemsets dashed circle, dashed box, Solid circle and solid box for calculating the frequent itemset And used hash based technique for reduce the size of candidate itemsets. Using Hash based technique at that time collision sometime occur but overcome this problem by rehashing technique . Besides other efficiency improving methods the DIC give confirmed frequent itemset form transactional dataset and compare to other algorithm our proposed algorithm require less time.**

IndexTerms **- frequent itemset, transactional database, DIC, Hash Technique**

Introduction

Association rule mining was introduced by Agrawal, and initially used in large-scale transaction data recorded in supermarket to discovering interesting relations between variables in large databases. There are some efficient algorithms uses the downward-closure property of support which guarantees that for a frequent itemset. For example, one of properties used by the Apriori algorithm is that all subsets of a frequent itemset must also be frequent. Finding association rules is the core process of data mining and it is the most popular technique has been studied by many researchers.. It is mining for association rules in database of sales transactions between items which is important field of the research in dataset .Using different algorithm through finding frequent itemset from large transactional dataset.

Frequent itemset mining has wide applications. The research in this field is started many years before but still emerging. This is a part of many data mining techniques like association rule mining, classification, clustering, web mining and correlations. The same technique is applicable to generate frequent sequences also. In general, frequent patterns like tree structures, graphs can be generated using the same principle. There are many applications where the frequent itemset mining is applicable. In short, they can be listed as market-basket analysis, bioinformatics, networks and most in many analyses.

In this paper we uses minhash technique for finding frequent itemset. And using bucket address provide the address space to that frequent itemset. .And using Rehashing technique resolves the collisions that are encountered during various collision resolution techniques used in open addressing starategy.This is done by increasing the size of a hash table, and restoring all of the items into the hash table using the hash function h(k)=k%m where m is the new length of the hash table after increasing it..And also used dynamic counting method for adding item dynamically if required.

I. BACKGROUND

2.1 Minhash Technique

A hash table (hash map) is a data structure used to implement an associative array, a structure that can map keys to values. A hash table uses a hash function to compute an *index* into an array of *buckets* or *slots*, from which the correct value can be found. Hash functions are primarily used in hash tables, to quickly locate a data record given its search key. Hash technique through finding the bucket address and put that itemset on that index.But whenever collisions occur after mapping the frequent item sets then an immediate check for the number of buckets still vacant in the hash table must be done. If it is observed that the hash table is either half –filled or is more than half of the size of the hash table is occupied then it is appropriate to apply rehashing technique using which we can double the size of the hash table thus providing enough buckets for all frequent item sets without any collisions.

2.2 Rehashing Technique

Rehashing is a technique used in hash tables to overcome hash collisions, when two different values to be searched for producing the same hash key. It is a popular collision resolution technique used on hash tables. Like linear probing ,it uses one hash value as a starting point and then repeatedly steps forward an interval ,until the desired value is located; an empty location is reached , or the entire table has been searched. .In Linear probing, Quadratic probing and Double hashing, we have to guess the number of elements we need to insert into a hash table.

2.3 Dynamic Itemset Counting

This is an alternative to Apriori Itemset Generation. In this itemsets are dynamically added and deleted as transactions are read. It is based on the downward release property in which this calculates the itemsets to different point of time regarding the scan. This algorithm also used to ease the number of database for discovering the frequent itemsets by just counting the new element at any fact of time finished the run time

DIC maintains four sets of itemsets, namely Dashed Circle,Dashed Box, Solid Circle and Solid Box. Itemsets in the"dashed" sets are subjects for support counting while itemsetsin the "solid" sets do not need to be counted. "Circles" contain infrequent itemsets while "boxes" contain frequent itemsets.

II. PROPOSED WORK

In general the structure of the transactional database may be in two different format – Horizontal data format and Vertical data format. In this paper, transactions of database are stored in the vertical format. Vertical data format ,an "Item: TID" format in which "TID" is unique identifier for of a transaction and "Item" is an item in database.

In this paper, We use hash based technique for providing bucket address to frequent mining itemset. But sometime collision occur because size of hash table is less than the required size.so avoid this problem we use hash method. It can be used to increase the size of a hash table, and restoring all of the items into the hash table using the hash function h(k)=k%m where m is the new length of the hash table after.And this paper we also use DIC method for add any no. of transaction in given database when required, which possible using dynamic itemset counting method.

A frequent itemset is an itemset that occurs frequently. In frequent pattern mining check itemset occurs frequently or not. Find frequent itemset from the given dataset check count support of itemset. If count support is greater than or equal to the min support then that itemset is frequent otherwise not frequent. Different methods used for frequent itemset mining. Here,use Hash & DIC method.

The steps for HashDIC algorithm is as follows:

- Apply DIC Technique to dataset.
- Get no. of unique items from the dataset.
- Get frequent mining itemset from given database.
- If collision occur then apply Hash technique and then get frequent itemset.
- Calculate Execution time require for finding frequent itemset

III. EXPERIMENT

We apply the experiments on two fimi datasets : 1)Online retail dataset 2) T40I10D100K dataset which have been commonly used for many frequent itemsets mining algorithms.

Here we use parameter for speed of HashDIC algorithm. Proposed algorithm give more spped compare to other algorithm. Comparision of the results can be show in the graph. Compare to Apriori and éclat algorithm HashDIC require less time.We perform our experiment in eclipse using java language. Java is purely object oriented language. Object oriented programming is a method of implementation in which programs are organized as cooperative collection of objects, each of which represents an instance of a class, and whose classes are all members of a hierarchy of classes united via inheritance relationship.

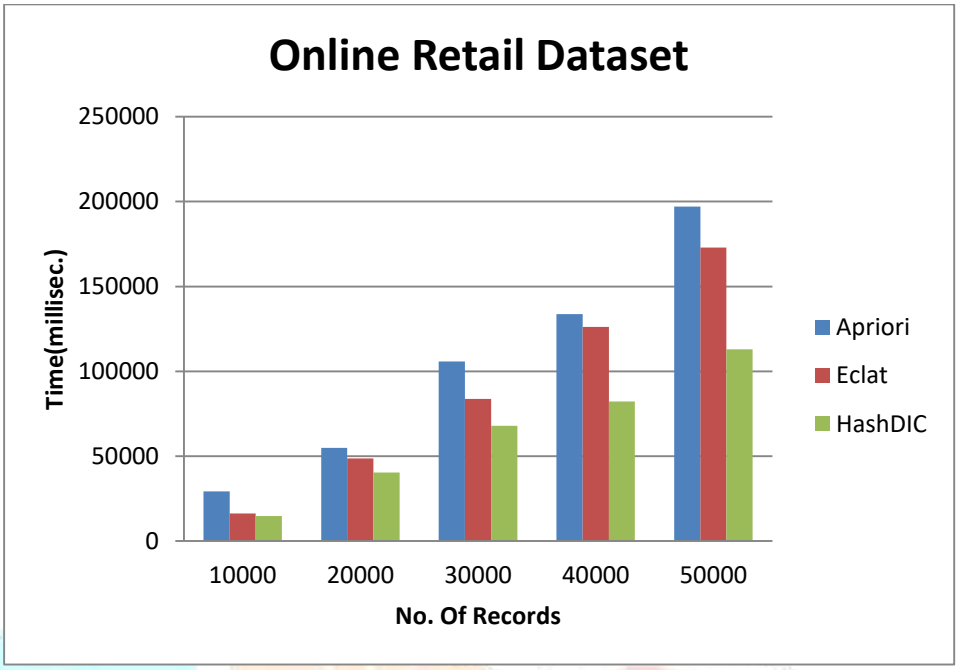

Fig 1: Comparison of HashDIC,Apriori and Eclat speed on Online retail dataset

No. of Records	Apriori(Time)	Eclat(Time)	HashDIC(Time)
10000	29306	16359	14720
20000	54896	48751	40473
30000	105854	83659	67953
40000	133702	126100	82177
50000	197020	172821	113031

TABLE I. Information of test online retail dataset

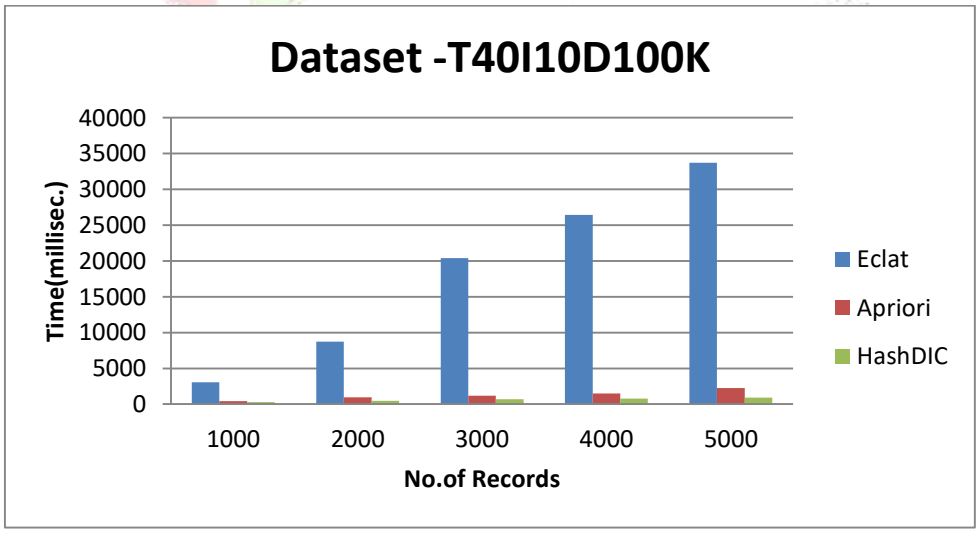

Fig 2: Comparison of HashDIC,Apriori and Eclat speed on T40I10D100K dataset

No. Of Records	Apriori(Time)	Eclat(Time)	Hash DIC(Time)
1000	406	3074	297
2000	951	8729	469
3000	1197	20397	671
4000	1518	26450	771
5000	2268	33702	924

TABLE II. Information of test T40I10D100K Dataset

IV. CONCLUSION

In this work, we proposed an approach to find frequent itemset using dynamic itemset counting & hashing method . Here, We presented the frequent itemset finding using DIC technique .And put that itemset in their bucket address but some collision occur due to one key put at same location.Our method we can overcome this collision problem using hashing technique through and get the unique bucket address for each frequent mining itemset. And finding the no. of unique items from the dataset and calculate the execution time for frequent itemset mining.And proposed algorithm is faster than the comparing to other algoithmFor the further improvement ,some other optimization methods can also be used to try in our framework.

ACKNOWLEDGMENT

I would like to express my sincere thanks to my guide Daxa V.Vekriya professor ,Computer department, for her vital support, valuable guidance and for providing me with all facility and guidance for presenting assisting me in times of need. I would also take this opportunity to express my heartfelt gratitude to Professor Ashutosh Abhangi, Head of the Department of Computer Engineering, for his valuable support and cooperation in the presentation of this paper.

REFERENCES

1) Chunkai Zhang , Xudong Zhang , Panbo Tian ," An approximate approach to frequent itemset mining" 2017 IEEE Second International Conference on Data Science in Cyberspace INSPEC Accession Number**:** 17098630 DOI -10.1109 /DSC 2017 .60
2) Hao Jiang and Xu He "An Improved Algorithm for Frequent Itemsets Mining" 2017 IEEE fifth international conference on advanced cloud and big data (CBD) INSPEC Accession Number**:** 17153579 DOI-10.1109/CBD 2017.61
3) Junrui Yang , Yingjie Zhang , Yanjun Wei "An Improved Vertical Algorithm for Frequent Itemsets Mining From Uncertain Database" IEEE 2017 9th international conference on Intelligent Human Machine Systems and Cybernetics INSPEC Accession Number**:** 17207441 ISBN**:** 978-1-5386-3022-8 DOI-10.1109/ IHMSC 2017.87
4) Mazaher Ghorbani and Masoud Abessi "A New Methodology for Mining Frequent Itemset On Temporal Data."IEEE Transaction on Engineering Management Volume: 64 INSPEC Accession Number: 17259900 DOI-10.1109/TEM .2017.2712206.
5) Sagar Bhise and Prof. Sweta Kale" An Efficient Algorithms To Find Frequent Itemset Using Datamining" International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 06 | June -2017. e-ISSN: 2395 -0056 ,p-ISSN: 2395-0072
6) O.Jamsheela, Raju.G, "Frequent Itemset Mining Algorithms :A Literature Survey", 2015 IEEE International Advance Computing Conference (IACC)
7) Wang L, Cheung D W, Cheng R, et al. Efficient Mining of Frequent Item Sets on Large Uncertai Databases[J]. Knowledge & Data Engineering IEEE Transactions on, 2012, 24(12):2170-2183.
8) R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no.2, pp. 207–216, 1993
9) Aakansha Saxena, Sohil Gadhiya , "A Survey on Frequent Pattern Mining Methods Apriori, Eclat, FP growth", 2014 IJEDR | Volume 2, Issue 1 | ISSN: 2321-9932 S.
10) Neelima, N. Satyanarayana and P. Krishna Murthy3,"A Survey on Approaches for Mining Frequent Itemsets", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-87.
11) Debajyoti Bera, and Rameshwar Pratap. Frequent-Itemset Mining using Locality-Sensitive Hashing[C]. Computing and Combinatorics: 22nd International Conference, 2016.
12) Xie Y, Palsetia D, Trajcevski G, et al. Silverback: Scalable association mining for temporal data in columnar probabilistic databases[C]. 2014 IEEE 30th International Conference on Data Engineering. IEEE, 2014: 1072-1083.

13) Ma Z, Yang J, Zhang T, et al. An Improved Eclat Algorithm for Minin Association Rules Based on Increased Search Strategy[J]. International Journal of Database Theory and Application, 2016, 9(5): 251-266.

14) A.M.J. Md. Zubair Rahman, P. Balasubramanie and P. Venkata Krihsna ―A Hash based Mining Algorithm for Maximal Frequent Itemsets using Linear Probing‖. Infocomp Journal of Computer Science 2009, Vol.8, No.1, pp.14-19.

15) Hao Jiang,You-Jin LIAO, Shi-Meng NI, A New Algorithm for Mining Frequent Itemset Using Efficient Data Structure International Conference on computer science and software engineering. 2014

16) Won D, McLeod D, An efficient approach to categorising association rules. International Journal of Data Mining, Modelling and Management, pp. 309-333, 2012.

17) C. Zeng, J. F. Naughton, and J.-Y. Cai, "On differentially private frequent itemset mining," Proc. VLDB Endowment, vol. 6, no. 1, pp. 25–36, 2012.

18) Hongjian Qiu, Yihua Huang, Rong Gu, Chunfeng Yuan, "YAFIM: A Parallel Frequent Itemset Mining Algorithm with Spark", 2014 IEEE 28th International Parallel & Distributed Processing Symposium Workshops

19) www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/DIC.html

20) www.justanswer.com/computer-programming/4np4s-data-mining-consists-five-major-elements-extract-transform.html

21) http://fimi.ua.ac.be/data/retail.dat

22) http://fimi.ua.ac.be/data/T10I4D100K.dat