RESEARCH ARTICLE OPEN ACCESS

FREQUENT ITEMSET MINING USING ECLAT WITH RELATIVE PROFIT AND PRICE

Siddhrajsinh Solanki[1], Neha Soni[2]
[1] *Computer Engineering Department, SVIT Vasad, India*
[2] *Computer Engineering Department, SVIT Vasad, India*

----------**********----------

Abstract:

Data mining is the process of extracting useful information from the huge amount of data stored in the databases. Data mining tools and techniques help to predict business trends those can occur in near future. Many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes, such as catalog design, cross-marketing, and customer shopping behavior analysis. Eclat is a classical algorithm for mining frequent itemsets, which is based on vertical layout databases. It is greatly different from those algorithms based on horizontal layout databases, such as algorithm Apriori and FP-Growth. Eclat uses vertical data format for frequent pattern mining. It is depth first search technique. It is proved that Eclat is better then apriori algorithm. It needs less database scan compare to apriori. Eclat is faster then apriori.

Eclat is not used with any kind of utility. In market basket analyses retailers have to analyze frequent itemset which have high profit corresponding to its price of that itemset. Investment is basic constrained in any business. While purchasing items from company or agency retailers have limited money and they have to invest in varying items with certain quantity by that limited money. While offering package of items retailers has to aware about limit of cost. Therefore, this work investigates working of eclat with utility called "Relative Profit" and with "Price". The main aim of this research is to look for and manipulate relative profit and price with algorithm to find itemset with high relative profit with price limit.

Keywords — Frequent Itemset Mining, Eclat, Equivalence class, Vertical data format, Relative Profit

----------**********----------

I. INTRODUCTION

Frequent patterns are the patterns which appear frequently in database. For example a set of items, such as milk and bread that appear frequently together in a transaction data set is a frequent itemset. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern. If Pattern occurs frequently, it is called a frequent pattern. Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Frequent pattern mining has become an important data mining task and a focused theme in data mining research.

There are various algorithms proposed for frequent itemset mining [12]. First algorithm named apriori was proposed by Rakesh Agrawal and Shrikant [11]. Eclat [5] was proposed by M. J. Zaki in 1997 and he proved that Ecalt is better than apriori. It uses only one database scan. In 2000 it had been proved that Eclat is scalable algorithm [7] and can be used for large datasets. In 2003 new technique Diffset [10] is proposed to use with Eclat algorithm to improve its memory usage. Christian Borgelt implement [9] Eclat and Apriori and he did comparative study of them. In 2004 Lars Schmidit examine the features of Eclat [8] Algorithm. In

2010 Kan Jin proposed new algorithm [6] based on Ecalt algorithm. In 2013 Ecalt is used in user behavior analysis through web log usage mining [1]. Eclat is also used to mine frequent itemsets on data stream[2]. In 2014, Eclat Algorithm is used in framework for rule mining on XML data[3]. In 2014 Eclat is implemented on GPU[4] to examine its performance and compared to apriori.

The main goal of any type of market basket analyses is to increase profit by making strategies for combine selling or cross marketing. In any business it is advantageous to gain more profit with less investment. So anyone can invest remaining money in other items or in more quantity. Profit is absolute term and it cannot give idea about investment. So we introduced utility Relative Profit (RP) to relate profit with investment.

In any business investment is basic constrained so if price is limitation then retailer have to analyze only those itemset which have total price less then predefined cost.

Below are scenarios where Relative Profit and Price constrain needs Consideration.

(1) When retailer want to invest more money on items which can give more profit on investment, and should be frequent also.

(2) When retailer want to find itemsets which are giving high return on investment, to make strategies to increase profit.

(3) When retailer want to find itemsets which have high profit margin so they can adjust discount on MRP.

(4) When investment is limitation at that time retailer have to find itemset which can be possible below some predefined price.

(5) When planning package sales at that time they have to maintain package price which can be suitable for average customer.

DEFINITIONS

- Transaction database: It is a collection of sets of items (transactions).
- Itemset -A collection of one or more items

Example: {Milk, Bread}

- k-itemset -An itemset that contains k items
- Support count (□)

-Frequency of occurrence of an itemset

-Number of transactions that contain an itemset

- Frequent Itemset

-An itemset whose support count is greater than or equal to a minsup threshold

- Association Rule
- An implication expression of the form X □ Y, where X and Y are itemsets
- Example:

{Milk} □ {Bread}

- Support of an association rule X□Y equals the support of X U Y
- Confidence of an association rule X□Y

= Support(X□Y) / Support(X)

II. OVERVIEW OF ECLAT ALGORITHM

Both the Apriori and FP-growth methods mine frequent patterns from a set of transactions in TID-itemset format (that is, {TID : itemset}), where TID is a transaction-id and itemset is the set of items bought in transaction TID. This data format is known as horizontal data format. Alternatively, data can also be presented in item-TID set format (that is, {item : TID set}), where item is an item name, and TID set is the set of transaction identifiers containing the item. This format is known as vertical data format. In this section, we look at how frequent itemsets can also be mined efficiently using vertical data format, which is the essence of the ECLAT (Equivalence CLASS Transformation) algorithm developed by Zaki. Mining can be performed on this data set by intersecting the TID sets of every pair of frequent single items. First, we transform the horizontally formatted data to the vertical format by scanning the data set once. The support count of an itemset is simply the length of the TID set of the itemset. Starting with k = 1, the frequent k-itemsets can be used to construct the candidate (k+1)-itemsets based on the Apriori property. The computation is done by intersection of the TID sets of the frequent k-itemsets to compute the TID sets of the corresponding (k+1)-itemsets. This process repeats, with k incremented by 1 each time, until no frequent itemsets or no candidate itemsets can be found.

Besides taking advantage of the Apriori property in the generation of candidate (k+1)-itemset from frequent k-itemsets, another merit of this method is that there is no need to scan the database to find the support of (k+1) itemsets. This is because the TID

set of each k-itemset carries the complete information required for counting such support. However, the TID sets can be quite long, taking substantial memory space as well as computation time for intersecting the long sets.

A. HORIZONTAL DATA FORMAT

For Example Given transaction database contains Transaction ids and items which it consist.

TID	List of item IDS
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Fig 1 (a) Horizontal data format

Above given Horizontal database can be represented as vertical database format. In vertical database format it consist items and set of transaction which contains it.

itemset	TID_set
I1	{T100,T400,T500,T700,T800,T900}
I2	{T100,T200,T300,T400,T600,T800,T900}
I3	{T300,T500,T600,T700,T800,T900}
I4	{T200,T400}
I5	{T100,T800}

Fig 1(b) Vertical data format

B. EQUIVALENCE CLASS

Let's take an example

Let L2= {I1I2, I1I3, I1I4, I1I5, I2I3, I2I4, I2I5, I3I4, I3I5, I4I5}.

Then

C3={I1I2I3,I1I2I4,I1I2I5,I1I3I4,I1I3I5,I1I4I5,I2I3I4,I2I4I5,I3I4I5}

Assuming that L_{k-1} is lexicographically sorted, we can partition the itemsets in L_{k-1} into equivalence classes based on their common k-2 length prefixes.

Candidate k-itemsets can simply be generated from itemsets within a class by intersecting all pairs.

For our example L2 above, we obtain the equivalence classes:

SI1 = [I1] = {I1I2, I1I3, I1I4, I1I5}
SI2 = [I2] = {I2I3, I2I4, I2I5},
SI3 = [I3] = {I3I4, I3I5}
SI4 = [I4] = {I4I5}

C. ALGORITHM

Input: F_k = {I1..In} frequent k Itemsets

Output: $F_{|R|}$ Frequent Item Sets

Bottom-Up(F_k):

for all $I_i \in F_k$ **do**

$F_{k+1} = \phi$;

for all $I_j \in F_k$, i < j do

N = $I_i \cap I_j$; // I_i and I_j Both should be //from same equivalence //class

if N.sup ≥ minsup **then**

$F_{k+1} = F_{k+1} \cup \{N\}$; $F_{|R|} = F_{|R|} \cup \{N\}$

end;

if F_{k+1} != ϕ ; **then**

Bottom-Up(F_{k+1});

end;

III. OVERVIEW OF RELATIVE PROFIT AND PRICE

In Market Basket Analysis retailers try to analyze buying habit of customers to make strategies to gain more profit by selective marketing or to plan their shelf. So in direct or indirect way goal of market asket analysis is to find frequent itemset which are more likely to be buy together by customer and make strategies to gain profit.

Frequent itemset mining algorithm find itemsets which are frequent but they dont consider profit at the time of mining frequent itemset. So after that whatever the itemsets are found that consist itemsets with high profit and low profit both. So if our goal is to increase profit than we have to focus on only high profit items.

For any business, retailers try to gain more profit by less investment. In simple words business firms or retailers try to increase profit with decreasing investment. Only profit term don't give any idea of investment. So retailers have to focus on items which are giving high profit relative to its Price.

For Example:
If there are two frequent products A and B.
Profit of A= 10
Price of A= 50
Profit of B= 15
Price of B=100

It is simple that retailer can get profit of 20 by investing 100on product A instead of 15 by investing 100 on product B.

Every retailer will choose product which have high return.

If retailer want to analyses this kind of relative profit then they have to find frequent itemset by existing algorithm and then they have to compute relative profit and then select those itemset which have high relative profit.

PRICE:

Investment is basic constrained in any business. So for retailers it is also constrained. Investment have some limits. So retailers have to invest their money in itemset which have cost below some predefined limit. Or while making strategies for package sale at that time they have to think for pocket size of customer. For Example instead of finding itemset which have cost of 15000 they can find itemsets which have cost less then 2500.

For that kind of analyses they have to find frequent itemsets by existing algorithm and then they have to check for price of itemsets. While checking if itemset (A,B) have cost greater than predefined limit than there is no need to check itemset (A,B,C,D). Because if (A,B) have high cost then (A,B,C,D) have high cost.

DEFINITIONS

(1) Cost: Cost is the price in which retailer buy items from company or agency.

(2) MRP: Maximum Retail Price is the maximum price in which retailer can sell item.

(3) Selling Price: It is the price in retailer actually sell item.

(4) Profit: It is difference between Selling Price (Or MRP) and Cost, where SP or MRP is greater then Cost.

(5) Relative Profit: It is percentage Profit on Price. Where price can be Cost or MRP.

D. RELATIVE PROFIT CALCULATION

In below Calclulation,

→RP can be calculated with reference to Cost or MRP based on requirement.

When RP calculated with reference to Cost at that time profit

Profit = SP – COST

When Rp is Calculated with reference to MRP at that time

Profit = MRP - Cost

→In Below Calcilation Price can be COST or MRP

→We can relate profit with price by Relative Profit (RP)

$$RP = \frac{Total\ Profit}{Total\ Price} * 100$$

For N items

$$RP_N = \frac{\sum_{i=1}^{N} Profit(i)}{\sum_{i=1}^{N} Price} * 100$$

→RP is different than simple Profit Because

$$RP(A) = \frac{Profit(A)}{Price(A)} * 100$$

$$RP(B) = \frac{Profit(B)}{Price(B)} * 100$$

$$RP(A,B) = \frac{Profit(A) + Profit(B)}{Price(A) + Price(B)} * 100$$
$$\neq \frac{Profit(A)}{Price(A)} + \frac{Profit(B)}{Price(B)} * 100$$

$$RP(A,B) \neq RP(A) + RP(B)$$

For simple profit

Profit(A,B) = Profit(A) + Profit(B)

→METHOD TO CALCULATE RP OF ITEMSET FROM COMPUTED RP

We can handle RP in different way.

TABLE II
EXAMPLE

Item	Relative Profit (%)
A	20
B	15
C	10

When relative profit previously computed then we can interpret it as profit when cost is 100.

In above table A gives 20 when cost is 100. So we can take above information as temporary profit and cost which is already based on original values of profit and cost.

We can interpret above table as

Profit (A) = 20 when Price (A) = 100

Profit (B) = 15 when Price (B) = 100

Profit(C) = 10 when Price (B) = 100

If we compute RP of Itemset (A,B) then

$$RP(A,B) = \frac{Profit(A) + Profit(B)}{Price(A) + Price(B)} * 100$$

But Price is scale down to 100 so

$$RP(A,B) = \frac{RP(A) + RP(B)}{100 + 100} * 100$$

$$= \frac{RP(A) + RP(B)}{2 * 100} * 100$$

$$RP(A,B) = \frac{RP(A) + RP(B)}{2}$$

Same can be prove for N items. By above equation we can get relative profit of itemset (A, B) by precomputed values of RP(A) and RP(B).

By above equation we can understand that relative profit of any item set is average of relative profit of items which are consist in it.

E. PRICE CALCULATION

Price is the absolute term so we can directly add Price of items to find cost of itemsets.

$$Price(A,B) = Price(A) + Price(B)$$

For N items,

$$Price(N\ items) = \sum_{i=1}^{N} Price(i)$$

III. PROPOSED SCHEME

To Handle Relative Profit and Price here is proposed scheme. We are assuming that Relative Profit of all item are previously computed and given before applying method. And price of all item are known.

ItemDB is Itemtable with precomputed relative profit and Price.

TABLE IIII
ITEM DB

Item	Relative Profit	Price

Notation:

1. VDB: Vertical database
2. ItemDB: Itemdatabase which contain RP and Price Value of Item
3. RP: Relative Profit of Item
4. minRP: minimum Relative Profit value given by user
5. P: Price of Item
6. maxP: maximum price value given by user
7. minsup: minimum support given by user

F. Proposed Algorithm

Input: vertical tid-list database VDB, minsup, maxP, ItemDB

Output: 1-frequent itemsets with P<maxP

Procedure find_frequent_1-itemsets

For all items Ii∈ VDB **do**

For all transactions Tj ∈ Ii **do**

|tid-list(Ii)|=|tid-list(Ii)|+1;

end

For all Ii∈ VDB,

If(|tid-list(Ai)|≥ min_ Supp

& P(Ii) < maxP) **then**

add Ii to F_1

end

end

Input: F1, minsup, minRP, maxP, ItemDB

Output: S: set of frequent itemsets with P<maxP and RP >minRP

Bottom-Up(F_k):

for all $I_i \in F_k$ **do**

$F_{k+1} = \phi$;

for all $I_j \in F_k$, i < j do

$N = I_i \cap I_j$; // I_i and I_j Both should be from same equivalence class

if N.sup ≥ minsup **then**

{

if (computeP(N) < maxP) **then**

{

$F_{k+1} = F_{k+1} \cup \{N\}$;

if (computeRP(N) >minRP) **then**

add N to S;

}

}

end;

if $F_{k+1} \neq \phi$; then

Bottom-Up(F_{k+1});

end;

IV. IMPLEMENTATION

We have taken datasets from www.fimi.ac.ua.be , which are already preporocessed for itemset mining. We have generated Item DB with random values as txt file which consist Rp value and Price for every item and in that line number is item id. We Analyse the working of our algorithm by passing different values of minRP and maxP.

G. HARDWARE SPECIFICATION

OS: Windows 7, Ubuntu 14.04 LTS

Processor: Intel(R) Core(TM) i3 2330M CPU 2.20 Ghz

Ram: 4GB

H. Dataset Description

All databases are publicly available at the Frequent Itemset Mining Implementations Repository (http://fimi.ua.ac.be/). Some of their Characteristics are as below.

TABLE IVII
DATASET

Dataset	Items	Avg. Lenth	Transaction
Retail	16469	10.3	88162
Mushroom	119	23	8124
Connect	129	43	67557

I. Output of Eclat

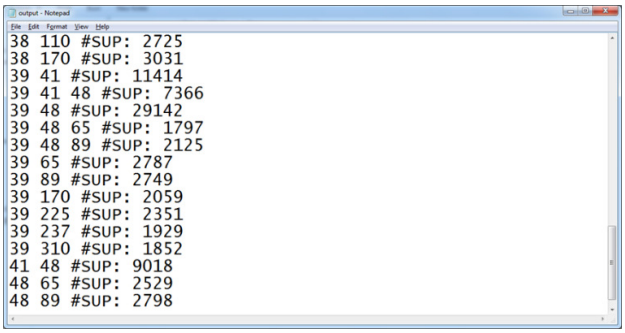

Fig 2 Eclat Output

J. Output of Eclat with Relative Profit and Price

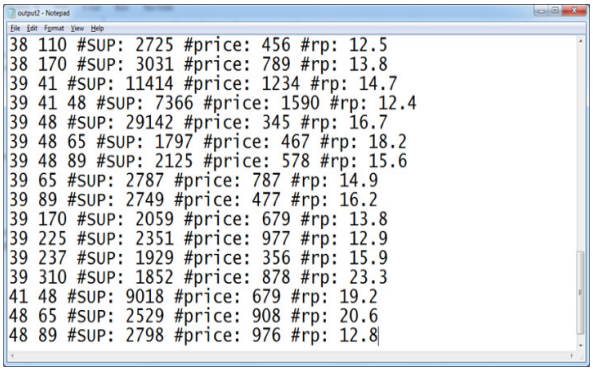

Fig 3 Eclat with Relative Profit and Price

K. Observations

- Eclat algorithm can be work well using RP and Price as utility and it gives frequent itemsets which have Relative Profit greater than minimum relative profit and which have total price less than maximum price.
- By maximum price value we can prune early procedure of searching branches which have higher price compare to specified maximum price.

V. CONCLUSION AND FUTURE ENHANCEMENT

In this research work, we have examine working of Eclat algorithm. We introduced new utility call "Relative profit" which have not been used with any algorithm and is not linier utility. As Eclat is faster than apriori we use relative profit and price with Eclat algorithm to find more suitable itemsets to invest more money in business with specific limitation on the price of item. Because of Relative profit computation algorithm take time but on the other hand we get itemset with high Relative profit. We can also proone search space by price limit.

Relative profit is not linier utility so it can not be handle easily. So methods should be found to handle it in easy way. Relative profit can be used with other algorithm in future.

ACKNOWLEDGMENT

We express our gratitude to management of SVIT, Vasad for proving opportunity and support for such an activity.

REFERENCES

[1] Bina Kotiyal, Ankit Kumar, Bhaskar Pant, R.H. Goudar, Shivali Chauhan and Sonam Junee "User Behavior Analysis in Web Log through Comparative Study of Eelat and Apriori" Proceedings of7'h International Conference on Intelligent Systems and Control (ISCO 2013) IEEE

[2] S.Vijayarani P.Sathya " Mining Frequent Item Sets over Data Streams using Éclat Algorithm" Proceedings published in International Journal of Computer Applications® (IJCA) 2013

[3] Rohini L. Damahe, Veena Kulkarni "An Efficient Framework for Rule Mining Using Eclat on XML Data" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 8, August 2014

[4] Sarika S.Kadam,Sudarshan S.Deshmukh "Eclat Algorithm for FIM on CPU-GPU co-operative & parallel environment" IOSR Journal of Computer Engineering (IOSR-JCE) Volume 16, Issue 2, Ver. VIII (Mar-Apr. 2014)

[5] M. J. Zaki, S. ParthaSarathy, M. Ogihara, W. Li "New Algorithms for Fast Discovery of Association Rules" KDD 97

[6] Kan Jin "A new algorithm for discovering association rules" Logistics Systems and Intelligent Management, 2010 International Conference on (Volume:3) 9-10 Jan. 2010 Ieee

[7] Mohammed J. Zaki," Scalable Algorithms for Association Mining" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 12, NO. 3, MAY/JUNE 2000

[8] Lars Schmidt-Thieme "Algorithmic Features of Eclat" FIMI, volume 126 of CEUR Workshop Proceedings, CEUR-WS.org, (2004)

[9] Christian Borgelt "Efficient Implementations of Apriori and Eclat" Workshop of Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL, USA)

[10] Mohammed J. Zaki and Karam Goudaz "Fast Vertical Mining Using Diffsets" KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining ACM New York, NY, USA ©2003

[11] Rakesh Agrawal Ramakrishnan Srikan "Fast Algorithms for Mining Association Rules"

[12] Dr. Kanwal Garg Deepak Kumar " Comparing the Performance of Frequent Pattern Mining Algorithms" International Journal of Computer Applications (0975 – 8887) Volume 69– No.25, May 2013

[13] en.wikipedia.org

[14] http://fimi.ua.ac.be