ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Improving Efficiency of MIKE Algorithm by Reducing Set Size

Gargi Narula[1], Sunita Parashar[2]

Research Scholar, CSE, HCTM, Kaithal, India[1]

Associate Professor, CSE, HCTM, Kaithal, India[2]

Abstract: Erasable item set mining was introduced to approach data mining in production planning. The erasable item set mining isthe process of finding erasable item sets that satisfy the constraint i.e. user defined value. This paper proposes an efficient algorithm for finding Top-Rank-K erasable item sets. Since the MIKE Algorithm was proposed to generate the top-rank-k erasable item sets. In last few years there have been several methods to improve its performance. But they do not consider the time and space constraint. If rank is high value then MIKE takes a lot of time and space to generate candidate set. In this paper, we proposed an Improved MIKE (I-MIKE) which reduces time and space by using efficient approach to generate candidate set.

Keywords: MIKE algorithm, data mining, Erasable item set, Apriori algorithm.

I. INTRODUCTION

Data Mining refers to extracting or mining knowledge from large amounts of data [1] . Data mining has gain popularity in the recent years due to wide availability of huge amount of data and the impending need of further turning it into useful information and knowledge .Frequent pattern Mining [5] is the fundamental problem in data mining which discovers frequent item set in database. It is the integration of various techniques from multiple disciplines such as statistics, machine learning, pattern reorganization, neural networks, image processing and database management system and so on[8] [10].

In this paper we mainly concentrate on erasable item sets mining problem. The problem of mining erasable item set [3] originate from production planning. Consider a manufacturing factory, which produces a large collection of products which constitutes. of some components are known as items (I= i1, i2, i3, i4……in). Sometimes, due to financial crises a manufacturing industry cant purchase all these items. Thus, finding the components which can be erased and without which the loss in profit is controllable is known as erasable item set mining. And these components are known as erasable item sets. The original motivation for finding erasable item set has been raised from the need to control the loss in profit due to absence of some component. Erasable item sets is helpful for manufacturer to decide how to purchase raw material or help to select which components can be rejected used for manufacturing products in case of some financial problem.

In this paper, we proposed a method to improve the efficiency of existing MIKE algorithm for erasable item sets mining.

The organization of the paper is as follow: Section 2 defines the problem of mining erasable item sets. Section 3 give the problem statement and basic concepts related to erasable item sets mining. Section 4 give the description and limitation of MIKE algorithm for mining erasable item sets. Section 5 will described the new improved approach and its Comparitive working example with old algorithm example .And finally we concludes in section 6 with a discussion of future work.

II. ERASABLE ITEM SET MINING PROBLEM IS DECOMPOSED INTO 2 SUB PROBLEM

A. First sub problem is calculating gain of item sets which are arranged by PID_list [4]

B. Second sub problem is to generate erasable item sets by using pruning technique for example top rank K value .Item sets upto this top rank K value (user specified) are known as erasable item sets.

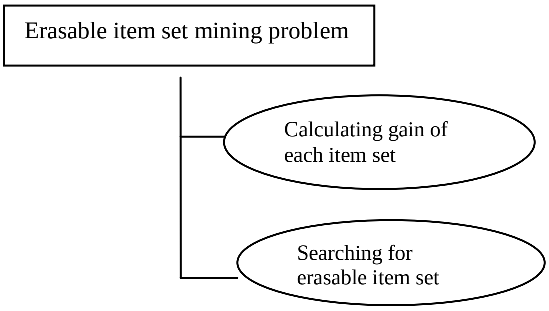
Erasable item set mining problem
Calculating gain of each item set
Searching for erasable item set

Fig 1. . Erasable item set mining problem

III. PROBLEM STATEMENT

Let I = {i1, i2, i3, i4…….im} is the set of m different literals known as_items which are abstract representation of components and a PDB = {P1, P2…. …....Pn} is the product database over I .and represented in the form of <PID, Items, Val>. PID is the identifier of Pi. Items are all items that constitutes pi. Val is the profit gained or obtained by selling pi.

TABLE 1
Horizontal Format

Product	PID	Items	Value
P1	1	{i1 , i2 , i4 }	200
P2	2	{i2 , i5 }	300
P3	3	{ i5 }	100
P4	4	{ i3 , i6 }	900
P5	5	{ i1 , i3 }	1200

A. *Basic Concepts [3]*

Definition (3.1*) (Gain) Let A (⊆ I) is an item set (i.e. set of items), the gain of A is defined as*

Gain (A) = ∑ Pk .Val **(1)**

{Pk | A∩Pk .Items≠ ∅}

Let A[I] be a set of items , the gain of A is defined by – " Sum of Profits of all products that include atleast one item in A as their component .

Definition (3.2) *(The Rank of an item set)[4] : It is the constraint which is used for discovering the erasable item sets and denoted as K .*

Given a product database DB and a pattern A (⊆ I), the rank of A, RA, is defined by

RA = |{Gain(X)|X ⊆ I|

Note that |Y| is the number of elements in Y.

Definition (3.3) (Top-Rank-k ErasableItem sets). Given a product database DB and a threshold k, an item set A (⊆I) is called to be a top-rank-k erasable item set if and only if RA is no greater that k. That is, RA ≤ k.

Based on the above definitions, the problem of mining top-rank-k erasable item sets can be described as follows: Given a transaction database DB and a threshold value k, the top-rank-k, erasable item sets mining is the task of finding the complete set of erasable item sets whose ranks are no greater than k. That is, the set of top-rank-k erasable item sets is equal to Stop-k, which is {X | X ⊆ I and RX ≤ k}[6].

Lets take an example of a product database in vertical format shown in Table 2.

TABLE 2
Vertical Format

Items	PID_list
I1	<1,200><5,1200>
I2	<1,200><2,300>
I3	<4,900><5,1200>
I4	<1,200>
I5	<2,300><3,100>
I6	<4,900>

IV. MIKE USING PID LIST

MIKE is the abbreviation for Mining TOP Rank –K Erasable item sets using Anti-monotone property. Different algorithms have been proposed for finding erasable item sets. MIKE by PID_listis the well-known application proposed by Z. Deng, Guo-Dong Fang and Z. Wang (2013) [7]. There are two main steps of this algorithm: first step is to generate a set of candidate item sets. Second is to find the gain of each candidate set in database and all truncate disqualified candidates.

A. *Anti-monotone Property*

If A is not a Top- rank- K erasable item set, any item set B containing A, which are also caleed super sets of A cant be a Top-rank-k erasable item set. This property is used to perform mining all top-rank-k item sets from short item sets to long item sets. This algorithm uses an iterative approach known (2) as levelwise search, which is also adopted by theapriori algorithm in frequent pattern mining.

B. *Terms related to this algorithm are*

Erasable item set [3]: The sets of item which satisfy the constraint (threshold) and it is denoted by Ei for ithitem set.

Apriori Property [2] : Any subset of erasable item set must be erasable .

Join Operation:To find Ek, a set of candidate k-item sets is generated by joining Ek-1 with itself[9].

Prune Step: MIKE uses two pruning techniques, first is based on the threshold (threshold for erasable item sets should be less than the user specified threshold) and second is, any (p-1)-item sets that is not erasable cannot be subset of an erasable p-item sets [5].

C. Pseudo-Code for MIKE [7]

Step 1: Initially set top-rank-k table ==null. Scan DB to compute the gain of each item. Collect the set of top-rank-k erasable 1-item sets and Insert these top-rank-k erasable 1-item sets into the top-rank-k table. For any two or more item sets having same gain are put into same tuple .

Step 2: Use the 1-item sets in the top-rank-k table to generate candidate 2-item sets. If the gain of a candidate 2-item set is no less than the smallest value of gain of the top-rank-k table, the candidate 2-item set is inserted into the top-rank-k table. After each inserting operation, the top-rank-k table is checked to ensure the number of tuples is not more than k.

Step3: Repeat procedure (2) by using l-item sets in thethe top-rank-k table to generate top-rank-k(l + 1)-item sets until no new item sets can be inserted into top-rank-k table.

Example of algorithm:

Scan database to compute the gain of each item

TABLE 3
Gain of each item

Items	Value
i4	200
i5	400
I2	500
i6	900
i1	1400
i3	2100

Let us suppose k=4

TABLE 4
Initial Table K

Items	Value
i4	200
i5	400
I2	500
i6	900

TABLE 5
TR1 // Temporary table for 1- item set

Item set	Value
i4	200
i5	400
i2	500
i6	900

TABLE 6
Candidate set for level-2 by using TR1

Item sets	Value
{i2, i4}	500
{i2 ,i5}	600
{i4 , i5}	600
{i4 ,i6}	110
{i5 ,i6}	1300
{i2 ,i6}	1400

TABLE 7
TR2 //temporary top-rank-k table for level 2

Item sets	Value
{i2, i4}	500
{i2 ,i5}	600
{i4 , i5}	600

TABLE 8
Candidate set for level-3 by using TR2

Item sets	Value
{i2, i4 , i5}	600

There is no eligible item set for further level therefore Loop stops.

TABLE 9
Complete top-rank-4 erasable item sets

Rank	Erasable Item sets	Value
1	{i4}	200
2	{i5}	400
3	{i2} , {i2,i4}	500
4	{i2,i5},{i4,i5} ,{i2,i4,i5}	600

D. Limitation of MIKE

If value value of Rank is high then a large no of candidate set is produced by using old approach of generating candidate set .And a lot of space consumed and time spends to deal with large candidate item sets. And due to large number of records in database results in much more input/output cost.

V. PRINCIPAL FOR IMPROVED MIKE (I-MIKE)

In this section an optimized method for MIKE algorithm [2] by using efficient approach to generate Candidate set. In Previous algorithm examples,it is analyze that candidate set at level 2 or more consist of superset of first few starting tuples of a initial table.So by analyzing these patterns of generating candidate set, in this paper we introduced an efficient method in such a way that initial candidate set is generated by a superset of selected top "n" items so that there is no need to generate candidate set at

each individual level and reduces the time and space complexity.
n= ceil((log_2(k)+0.1))to roundoff a value of n range of(0.1-0.4) can be added.
Where n are top selected items such that 2^n>=k, means size of candidate set must be equal or greater than the rank value k.
In this new method Item sets from candidate set are selected more successfully as compared to previous work.
Time complexity is O(n).

A. *Description of the Algorithm*

Input: **:** a product database *DB* and a rank value *k*.
Output: a top-rank-K table,which include the complete set of top-rank-K erasable item sets.

```
Step 1: Initialize Top-Rank-K table =∅
Step2:Scan database to compute the gain of each item and Sort I , the set of all items according to gain ascending order .
Step 3: Select top 'n' items from sorted list such that $n^2$>=k
Step 4: Initialize Candidate Set with superset of top 'n' items such that |Candidate Set|>=k
Where |Set|=number of items in Set
Step 5: Initialize selected_candidate == ∅
Step 6: For each item set in Candidate-set do step (a) to (c)
(a) If value of item seti ==value of any row j in top-rank-k table
additem seti in the same row j of top-rank-k table  and selected_candidate set .
goto step (6)
 (b) If |Tabk| < k
 Create a new row r with item seti .and add a row r in top-rank table and selected_candidate set .
goto step (6)
(c) If value of last row in Top-Rank table is greater than gain
ofitem seti , the item set is inserted into the top-rank-k table.
If |TabK| >k then delete last row.
Step 7:  ++n // n is used to index next item from database to be added in candidate set
Step 8: If value of last row ot top rank k table is equal or greater than the value of i[n]
Then candidate set = {selected candidate ∪i[n]}

Do step 6 to 7 while ([selected candidate] > 0)
```

Example of improved algorithm:

Let us suppose rank value K=4

TABLE 10
Initial Candidate –Set

Item sets	Value
i4	200
i5	400
i2	500
{i2,i4}	500
{i4 ,i5}	600
{i2 ,i5}	600
{i2 ,i4,i5}	600

TABLE 11
Initial Tab-k

Item sets	Value
{i4}	200
{i5}	400
{i2} ,{ i2 ,i4}	500
{i2,i5},{i4,i5} ,{i2,i4,i5}	600

Now value of n becomes 4 that is is i[6] .
Value of i[6] = 900 which is greater than the value of last tuple in TabK therefore no further new candidate Set is generated and algorithm stops .

TABLE 12
Complete top-rank-4 erasable item sets

Item sets	Value
{i4}	200
{i5}	400
{i2} ,{ i2 ,i4}	500
{i2,i5},{i4,i5} ,{i2,i4,i5}	600

VI. Conclusion

In last few years a lot of people have given several algorithms to solve the erasable item set problem as efficiently as possible .In this paper, MIKE algorithm is improved by using the concept of generating initial candidate set which consist of superset of first few initial tuples of a table inspite of generating candidate set at each individual level .
For high value rank,there was a performance bottleneck because a large size of candidate set is produced .Thus it is very important to improve its performance. In this new approach, we gives the key idea to improve its performance for high value rank. The idea of generating candidate set by improved method will definitely open new scope for Young researchers to work in the field of erasable item set data mining .Although this improved MIKE is efficient but in case of low value rank it has same performance as previous algorithm .For future researchers can found the new approaches to make it efficient in all range of values of a rank.

References

[1] Han J, Camber M. Data mining: Concepts and techniques. (2nd Ed.). Amsterdam, the Netherlands: Elsevier; 2006
[2] Aggarwal R, Imielinski T, Swami. A. Mining association rules between sets of items in large databases. In: SIGMOD 93; 1993. pp 207–216.
[3] Deng, Z., Fang, G., Wang, Z., Xu, X. Mining Erasable Item sets. In: 8th IEEE International Conference on Machine Learning and Cybernetics, pp. 67–73. IEEE Press, New York (2009).

[4] ZhihongDeng and XiaoranXu, China.Mining Top-rank-K Erasable Item sets, in ICIC Express Letters, ICIC International @ 2011 ISSN 1881-803X, Volume 5, Number 1, January 2011.
[5] K. Geetha, Sk. Mohiddin. An Efficient Data Mining Technique for Generating Frequent Item Sets, In: Proceeding of IJARCSSE, volume 3, Issue 4, April 2013.
[6] Hu J, Mojsilovic. A. High-utility pattern mining, a method for discovery of High-utility item sets. Pattern Recogn 2007;40(2007):3317–3324.
[7] Zhihiong Deng. Mining Top-Rank-K Erasable Item sets by PID_lists, International Journal Of Intelligent Systems,Vol.28,pages 366-379(2013)
[8] TianLan, Runtong Zhang and Hong Dai. A New Frame of knowledge discovery, In preceedings of 1st International workshop on Knowledge discovery and data mining,WKDD 2008,Jan. 2008,pp. 607-611.
[9] Dr. KanwalGarg,Shweta. Improving efficiency of META algorithm using record reduction,international journal of computers and technology,vol 8,no1.
[10] G.Cormode, Fundamentals of Analyzing and Mining Data Streams, WORKSHOP ON DATA STREAM ANALYSIS, San Leucio, Italy 2007.
[11] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi, ―Improving Efficiency of Apriori Algorithm Using Transaction Reduction‖, In: proceeding of IJSRP, Volume 3, Issue 1, January 2013.

BIOGRAPHY

Ms. Gargi Narula received the B. Tech degree in computer science and engineering from Kurukshetra University in 2011. And now doing M.Tech in computer science from Kurukshetra University, 2013 batch . Her Research interest includes Data Mining.