An Incremental Approach for Mining Erasable Itemsets

Suchi Shah
ME Computer Engg.
SVIT, Vasad

Jayna Shah
AssistantProfessor
SVIT,Vasad

ABSTRACT

A factory has a production plan to produce products which are created from number of components and thus create profit. During financial crisis, the factory cannot afford to purchase all the necessary items as usual. Mining of erasable itemsets finds the itemsets which can be eliminated and do not greatly affect the factory's profit. The managers uses erasable itemset (EI) mining to locate EIs. If the manager wants to determine which new products are beneficial for the factory, we have to apply EI mining on the original database with new products from the scratch. So, here the incremental approach to mine erasable itemsets is proposed which scans only new products and update the EIs which were found previously from original database.

Keywords

Data mining, Erasable itemset mining, pidset, dpidset

1. INTRODUCTION

Data mining is the boon of IT industry. Upto now we are using the frequent pattern mining to find that which items are frequent. For e.g. in retail store we see the buying habits and see that bread and butter are purchased frequently. But, now when we see a production plan than a problem of mining erasable itemsets originates. In a manufacturing factory, the products are produced on a very large scale. Also, each product is formed with a few components or materials. The factory has to spend the large amount to purchase and also store this component to manufacture their products. But, the factory cannot purchase all the needed components when there is a financial crisis. So, now a big question for the managers is that how to decide the production plan due to the limited money. As they cannot purchase all the components due to the financial crisis, hence one needs to stop the manufacturing of some products.

So, managers can create the new production plan by finding which itemsets can be erased i.e. eliminated. This is known as erasable itemsets. But, by stopping the manufacturing of some products the loss of factory's profit should be controllable. Several algorithms have been proposed to solve the EI mining problem, such as META (mining erasable itemsets with the anti-monotone property), VME (vertical format based algorithm for mining erasable itemsets) and MERIT (fast mining erasable itemsets).Also it proposes the algorithm called MERIT+ which is capable of mining EIs fully. Mining Erasable itemsets (MEI) scans the database many times when the new products are inserted. So, the time and the efforts required to mine the erasable itemsets is much more. Hence an algorithm called mining erasable itemsets from incremental database has been proposed. When new products are inserted the scan is not done from the starch only the scanning of new products is done and update is made in original database. The organization of the paper is Section 1 explains Introduction. Section 2 briefly explains the related work on different erasable itemset mining algorithms. Section 3 gives Problem statements and definitions. Section 4 introduces incremental approach for mining erasable itemsets. Section 5 presents the results on performance of MEI and incremental approach to mine erasable itemsets. Finally, conclusion and suggestions for future research are given in Section 6.

2. RELATED WORK

2.1 META:-

META Algorithm is used for Mining Erasable iTemsets with the Anti-monotone property. It adopts the Apriori algorithm and uses iterative approach with level wise search. In this level wise approach, first step is to find the set of erasable 1-itemsets.This set is denoted by E1.Next step is to find the set of erasable 2-itemsets from the erasable 1-itemsets.This is denoted by E2. Similarly, we need to find k- erasable itemset upto no more sets are found. Also, it uses the horizontal data format for finding the gain .It scans the database to find the total profit of the factory and k-times to find the erasable itemsets, where k is the maximum level of EIs .So, when we find k-erasable itemsets it scans the database k+1 times. And only the items with same prefix are combined. For eg consider 2-erasable itemsets {ab,ac,ad,bc,bd,cd}, when it considers the first element {ab} to combine with other elements having the same prefix. So, only {ac, ad} are combined and {bc,bd,cd} are redudant. Here, the itemsets may be redundant.[1]

2.2 VME: -

VME Algorithm uses the same approach as META, but the data structure used is vertical for mining erasable itemsets.PID_List is a set of pairs <PID,Val>, where PID is the product identifier and Val is the gain of this product. It uses the PID_list for storing the identification number of products, which stores in the format <Item, gain>. It can remove the irrevalant data easily. In level wise approach, first step is to find the set of erasable 1-itemsets.This set is denoted by E1.Next step is to find the set of erasable 2-itemsets from the erasable 1-itemsets.This is denoted by E2. Similarly, we need to find k- erasable itemset upto no more sets are found As is scans the databse twice ,the algorithm is faster then META but the weakness of VME is it scans the database to find the total profit of the factory and again to find the erasable itemsets. So,scanning the database takes a lot of time and memory. VME stores each value which leads to data duplication.[2]

2.3 MERIT:-

MERIT algorithm is used for mining fast erasable itemsets using NC-Sets. To keep the track of complete information the new structure is used called as NC-Sets (node code sets) as it is a compact structure. Also, it can efficiently remove the irrelevant data. Using NC-Sets, the gain of itemsets can be computed in a linear time complexity. Without generating candidate itemsets it can find erasable itemsets sometimes. The NC-Sets builds the WPPC Tree and WPP-Code with <(pre-order, post-order):weight>.But, MERIT uses the union strategy and hence requires the large memory for finding the large number of EIs. It scans the database many times and so takes lot of time. Also it stores the value which leads to the data duplication.[4]

2.4 MIKE:-

MIKE algorithm is used for mining top rank k erasable itemsets, where k is the maximum value to be mined. It is the task of finding the erasable itemsets whose ranks is no greater than k. Here the unwanted results is removed and only the required results are generated. So, the search space is also reduced to a larger extent. But, MIKE can only find the top rank k erasable itemset. Hence ,deciding the value of k is very difficult.[3]

2.5 MERIT+:-

MERIT+ is an algorithm which is used for mining erasable itemsets using difference of NC-Sets. It is similar to MERIT and which is then established for the foundation of dMERIT+.To improve the mining time the weight index, a hash table and the difference of Node Code Sets (dNC-Sets) are used. Also, the memory usage is reduced to a large extent. Here it is not capable of mining all EIs by checking all subsets (k-1) itemsets of a k-itemset whether X is erasable or not. So, not capable to say whether it is erasable or not by checking all its subsets. Also, not capable for enlarging the equivalence classes.[5]

2.6 MEI: -

MEI algorithm uses the divide and conquer strategy and dpidset i.e. difference of two pidsets concept for mining erasable itemsets. As only the difference is stored so the memory usage and time is also reduced. So from erasable 1 –itemset we can find erasable 2-itemsets and so on upto k-erasable itemsets. Hence, we can efficiently find the erasable itemsets. But, the problem is that it scans the database from the scratch many times when new products are inserted. So, the time and efforts required are more. So, here the original database is scanned and after that when new products are inserted than only scanning of that products are done and the original database is updated.

Table 1. Original Product Database

Product	Items	Val(million $)
P1	a	500
P2	a, b, c	200
P3	c, d	100
P4	b, e	100
P5	c d, e	50

3. PROBLEM STATEMENTS AND DEFINITIONS

Let I = {i1, i2, …, im} be a set of all items. A product database is denoted by DB={P1, P2, …, Pn}, where product is presented in the form of ⟨Items, Val⟩, where Items are the items (or components) that constitute Pi and Val is the profit that the factory obtains by selling the product Pi. In this database, {a, b, c, d, e} is the set of items (components) used to create all products {P1, P2, …, P5}. Product P2 is made from three components {abc}. The factory earns 200 million dollars by selling product P2.[7]

Terms related to Erasable itemset mining are as follows:

3.1 Gain of items:

Let X (⊆ I) be an itemset.The gain of X is defined as follows:

$$g(X)= \sum_{\{Pk\,/\,X \cap Pk.Items \neq \phi\}} Pk.Val$$

The gain of itemset X is the sum of profits of the products which include at least one item in itemset X. For instance, let X = {ac} be an itemset.

From the example database in Table 1,{P1, P2, P3,P5} are the products which include {a}, {c}, or {ac} as components.
So, g(X) = P1.Val + P2.Val + P3.Val + P5.Val
= 850 million dollars. [7]

3.2 When items are erasable:

Given a threshold ξ and a product database DB, let T be the total profit of the factory.

An itemset X is erasable if: g(X)≤T * ξ

Where T is computed as,T= $\sum_{Pk \in DB} Pk.Val$

The total profit of the factory is the sum of profits of all products. From the example database in Table 1, the total profit of the factory T is 950 million dollars.

An itemset X is called an EI if g(X)≤T*ξ. Let the threshold be 50% (ξ =50%). Now, g({b}) = 300 million dollars. Item e is called an EI.[7]

3.3 Pidsets –The set of product identifiers:

For an itemset X, p(X), the set of the product identifiers, is denoted as follows:

P(X) = $\bigcup_{A \in X} p(A)$ where A is an item in itemset X and p (A) is the pidset of item A, i.e., the set of product identifiers which includes A.

For the example database in Table 1, the pidset of {a} is {1, 2,} because P1, P2 include {a} as a component. Similarly, the pidset of {b} is { 2, 4}. The pidset of itemset X = {ab} is
P(X) = p({a}) Ụ p({b}) = {1, 2} Ụ { 2, 4}= {1, 2, 4}.[7]

3.4 Gain of an itemset based on pidset:

Let X is an itemset. The gain of X denoted by g(X) is computed as follows: g(X) = ∑ G[k] where G[k] is the element at position k of G.

For e.g. Pidset of {a} is {1, 2} because P1, P2 include {a} as a component. Similarly, the pidset of {b} is {2, 4}.

The pidset of itemset X = {ab} is

P (X) = p ({a}) Ų p({b}) = {1, 2} Ų{2, 4}= { 1,2, 4}. The gain of X is g(X) = G [1] + G [2] + G [4] = 800 million dollars.[7]

3.5 dPidset – The difference pidset of two pidsets:

Let XA and that of XB is denoted by p (XB). The dPidset of pidsets P (XA) and p (XB), denoted as dP (XAB), is defined as follows: dP(XAB) = p (XB)\p (XA).

e.g. XA = {ab} with p (XA) = {1, 2, 4} and XB= {ac} with P (XB)={1, 2, 3, 5}.

The dPidset of XAB is dP (XAB) = p (XB)\p(XA)

= {1, 2, 3, 5}\{1, 2, 4} = {3,5}. [7]

3.6 ⊕ operations:

Let XA and XB be two k-itemsets with the same prefix X. The operation for creating the (k+1) - itemset is defined as follows: XAB=XAXB

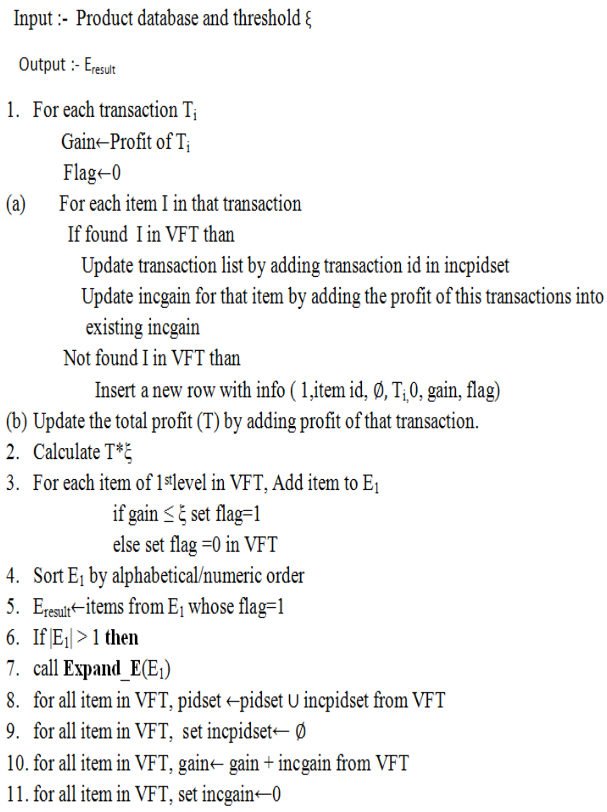
E.g. Let XA= {ab} and XB= {ac} be two 2-itemsets with the same prefix{e},XAB=XA-XB= {abc}.[7]

4. INCREMENTAL APPROACH FOR MINING ERASABLE ITEMSETS

The goal of this research is to mine incremental approach for erasable itemset. We will be using the vertical format table (VFT) with various fields. Here in vertical format table the attributes will be updated in format (level, item_id, pidset, incremental pidsets, gain, incremental gain and flag).The level field will show the level of items. The item id will be the id of the item. Initially the pidset and gain will be empty. As, the new item is inserted the entry is done in incremental pidset which shows the list of items at 1st level and then the dpidset i.e. difference of pidsets from next level. And according to that the gain is updated in incremental gain.

Once, every combination is there with their dpidset and gain. We, will copy the same to pidset and gain and clear it from incremental pidset and incremental gain. So, now when again the new items are inserted we will insert it in incremental pidset and update it in incremental gain. And according to it we will update the total profit, its gain and also the threshold value. The flag is set to identify whether the items are erasable or not. Also we will be sorting the items in the alphabetic order. Hence, there would be no redundant combination of the items produced in the database.

4.1 The algorithm for incremental approach for mining erasable itemsets



```
Input :- Product database and threshold ξ

Output :- $E_{result}$

1. For each transaction $T_i$
       Gain←Profit of $T_i$
       Flag←0
(a)    For each item I in that transaction
        If found I in VFT than
          Update transaction list by adding transaction id in incpidset
          Update incgain for that item by adding the profit of this transactions into
           existing incgain
       Not found I in VFT than
            Insert a new row with info ( 1,item id, ∅, $T_{i,}$0, gain, flag)
(b) Update the total profit (T) by adding profit of that transaction.
2. Calculate T*ξ
3. For each item of 1st level in VFT, Add item to $E_1$
              if gain ≤ ξ set flag=1
              else set flag =0 in VFT
4. Sort $E_1$ by alphabetical/numeric order
5. $E_{result}$←items from $E_1$ whose flag=1
6. If |$E_1$| > 1 then
7. call Expand_E($E_1$)
8. for all item in VFT, pidset ←pidset ∪ incpidset from VFT
9. for all item in VFT, set incpidset← ∅
10. for all item in VFT, gain← gain + incgain from VFT
11. for all item in VFT, set incgain←0
```

Procedure Exapand_E(E_v,level)

```
1.  for k← 0 to |$E_v$|-2 do
2.    $E_{next}$← ∅
3.    for j← (k+1) to |$E_v$|-1 do
4.        E.Items =$E_v$[k].Items ⊕$E_v$[j].Items
5.       (E.incpidsets, Gain, )←
 Sub_dPidsets ($E_v$[ k].incpidset, $E_v$[j] .incpidset)
6.       E. incgain= $E_v$[k].incgain + Gain
7.         if $E_v$[j].items is new and $E_v$[ k].items is old than
              E.incgain= E.incgain + Gain of $E_v$[k].items in VFT
8.      if E.gain <T* ξ than flag=1
9.          $E_{result}$←E
10.     else flag=0
11. $E_{next}$←E
12.      for each E in $E_{next}$
             if E.items is there than
             Update the value of incgain and incpidset in VFT for E.item
           else
             Insert (1, itemid, ∅, E.incpidset,0,E.incgain,flag)
13.   If |$E_{next}$|>1 then
14.    Expand_E($E_{next}$)
```

```
Procedure Sub_dPidsets

Input:  dPidsets d1,d2 and index of gain G

Output: dPidsets {d3} and its gain (Gain)

i←0
j←0
Gain←0
$d_3$←Ø
while i<|$d_1$| and j<|$d_2$| do
      if $d_1$[i] < $d_2$[j] then
            i++
else if $d_1$[i] == $d_2$[j] then
      i++
      j++
else
      Gain=Gain + G[$d_2$[j]]
      insert $d_2$[j] into $d_3$
      j++
while j < |$d_2$| do
    Gain = Gain + G[$d_2$[j]]
    Insert $d_2$[j] into $d_3$
    j++
return $d_3$ and Gain
```

4.2 Example

1. In this strategy we will use the Vertical Format Table(VFT) which consist

of 7 fields.

Level	Item id	Pidsets	Incpidsets	Gain	Incgain	Flag

2. When the new products are inserted then the values are updated in Incpidsets and Incgain. Also , sorting is done according to the alphabetic order. And erasable 1-itemsets are found and flag is updated.

3. Now to find 2-erasable itemsets the dipidsets concept is used which is calculated from 1-erasable pidsets . And the gain is calculated and again accordinly updates its flag.

4. Then it uses the Exapand_E procedure to implement this strategy to compute all erasable itemsets.

5. Once the computation is done than every contents of Incpidsets is copied to Pidsets and Incgain to Gian. And then we clear the contents of Incpidsets and Incgain.

6. Now when new products are inserted than again we insert it in Incpidsets and Incgain ,the process is repeated till it finds all erasable itemsets.

Table 2:-Consider the product database

Product	Items	Val(million $)
P1	a	500
P2	a, b, c	200
P3	c, d	100

The total profit of the factory is the sum of profits of all products. i.e. T=800(500+200+100) million dollars. Let the threshold be 50% (ξ =50%). So, the items will be erasable when gain ≤ 400 and flag will be set to 1 else flag will be set to 0.

Now the Vertical format table(VFT) enteries is as follows

Table 3:- Vertical Format Table

Level	Item id	Pid set	Inc pid set	Gain	Inc Gain	Flag
1	a	φ	**1,2**	0	**500+200 =700**	0
1	b	φ	**2**	0	**200**	1
1	c	φ	**2,3**	0	**200+100 =300**	1
1	d	φ	**3**	0	**100**	1
2	ab	φ	**φ**	0	**700**	0
2	ac	φ	**3**	0	**700+100 =800**	0
2	ad	φ	**3**	0	**700+100 =800**	0
2	bc	φ	**3**	0	**200+100 =300**	1
2	bd	φ	**3**	0	**200+100 =300**	1
2	cd	φ	**φ**	0	**300+0= 300**	1
3	abc	φ	**3**	0	**700+100 =800**	0
3	abd	φ	**3**	0	**700+100 =800**	0
3	acd	φ	**φ**	0	**800+0=800**	0
3	bcd	φ	**φ**	0	**300+0=300**	1
4	abcd	φ	**φ**	0	**800+0=800**	0

In the level 1 we writes its Product in which they are included and calculate its gain. And from 2nd level we find the difference i,e dpidset form that and according to it update its gain in incpidset and incgain.After this copy the contents of incpidset to pidset and incgain to gain and set incpidset← ∅ and incgain←0 .So. when new products are inserted than again insert it in the incpidset and incgain.

This two products are inserted in the original database

Table 4:- New Products Inserted

Product	**Items**	**Val(million $)**
P4	b, e	100
P5	c, d, e	50

When 2 new products are inserted than its total profit (T)= 950 (500+200+100+100+50) and let Threshold = 50%.So, gain ≤ 950 * 50 % than the items will be erasable. Hence, when gain ≤ 450 than items are erasable and flag is set to 1.

So, the vertical format table(VFT) when the new products are inserted

Table 5:- Vertical Format Table of New Items

Level	**Item id**	**Pid set**	**Inc pid set**	**Gain**	**Inc Gain**	**Flag**
1	a	1,2	**φ**	700	**0**	0
1	b	2	**4**	200	**100**	1
1	c	2,3	**5**	300	**50**	1
1	d	3	**5**	100	**50**	1
1	**e**	φ	**4,5**	**0**	**100+50=150**	**1**
2	ab	φ	**4**	700	**0+100=100**	0
2	ac	3	**5**	800	**0+50=50**	0
2	ad	3	**5**	800	**0+50=50**	0
2	ae	φ	**4,5**	0	**700+0+150=850**	0
2	bc	3	**5**	300	**100+50=150**	1
2	bd	3	**5**	300	**100+50=150**	1
2	be	φ	**5**	0	**200+100+50=350**	1
2	cd	φ	**φ**	300	**50+0=50**	1
2	ce	φ	**4**	0	**300+50+100=450**	1

Here when we see about b item than the original contents are copied into pidset and gain. Now as b is included in product P4 so its Incpidset is {4} and its Incgain is 100.

Similarly , it uses the Exapand_E procedure to implement this strategy to compute all erasable itemsets.

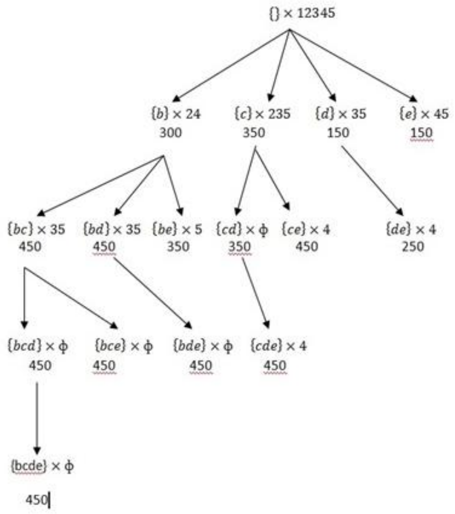
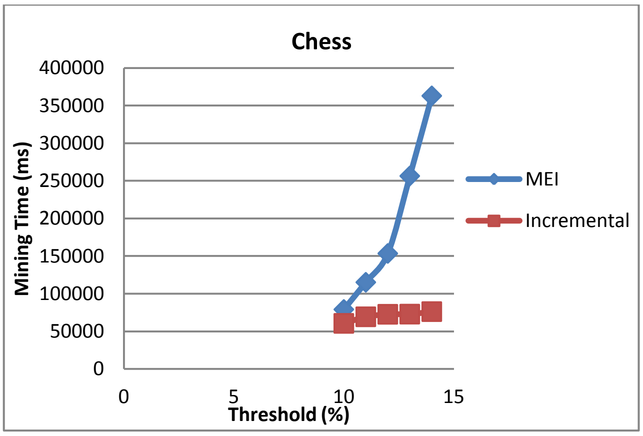
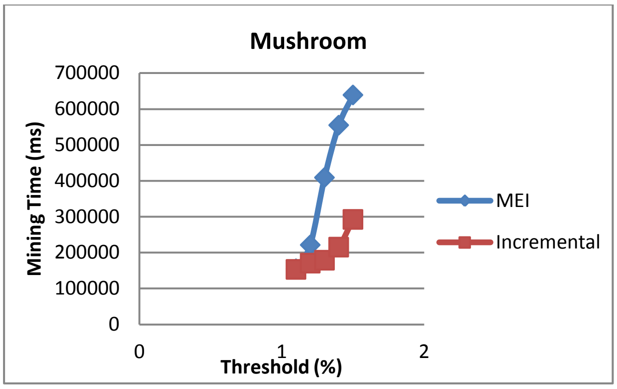


Fig 1:- Tree of all erasable nodes

5. RESULTS





6. CONCLUSION AND FUTURE WORK

This paper proposes the concept of incremental approach for mining erasable itemsets by using the concept of pidset and dpidsets. By sorting items into alphabetic order the redundant patterns are not generated. By using this approach the scanning is not done from the scratch when new products are inserted. And also by using the incpidset in VFT table, we can find the dpidsets of EIs in less time by intersecting the set of incpidsets having small size. So, the mining time and efforts are reduced.

In future work, we can mine the rules from erasable itemsets and also mining based on closed/maximal erasable itemsets. Mining an erasable itemsets from the huge database can be done. Also how to use erasable itemsets in recommendation system can be studied in future.

7. REFERENCES

[1] J. Han, M. Kamber, and J. Pei, ”Data mining: concepts and techniques” 3rd Edition, Morgan kaufmann, 2006.Deng, Z., Fang, G., Wang, Z., Xu, X., 2009. Mining erasable itemsets. In: ICMLC'09, pp. 67–73.

[2] Deng, Z.H., Xu, X.R., 2010. An efficient algorithm for mining erasable itemsets. In: ACDM'10, pp. 214–225.

[3] Deng, Z.H., Xu, X.R., 2012. Fast mining erasable itemsets using NC_sets. Expert Systems with Applications 39 (4), 4453–4463.

[4] Deng, Z.H., 2013. Mining top-rank-k erasable itemsets by PID_lists. International Journal of Intelligent Systems 28 (4), 366–379.

[5] Le,T., Vo,B., Coenen,F., 2013 An efficient algorithm for mining erasable itemsets using the difference of NC-Sets. IEEE Copmuter society , pp. 2270-2274

[6] Le,T., Vo,B., 2013.MEI: An efficient algorithm for mining erasable itemsets .Elsevier’13, pp.155-66

[7] Shweta,Garg,k.,2013V Searching the best strategies of mining erasable itemsets. International Journal of scientific &engg. research (4), pp. 673–677.

[8] Le, T.P., Vo, B., Hong, T.P., Le, B., 2012. An efficient incremental mining approach based on IT-tree. In: IEEE.

[9] Aggarwal, C.C., Li, Y., Wang, J., Wang, J., 2009. Frequent pattern mining with uncertain data. In: SIGKDD'09, pp. 29–38.

[10] Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: VLDB'94, pp. 487–499.

[11] Bernecker, T., Kriegel, H., Renz, M., Verhein, F., Zuefle, A., 2009. Probabilistic frequent itemset mining in uncertain databases. In: SIGKDD'09, pp. 119–127.

[12] Grahne, G., Zhu, J., 2005. Fast algorithms for frequent itemset mining using fp-trees. IEEE Transactions on Knowledge and Data Engineering 17 (10), 1347–1362.

[13] Gupta, R., Fang, G., Field, B., Steinbach, M., Kumar, V., 2008. Quantitative evaluation of approximate frequent pattern mining algorithms. In: SIGKDD'08, pp. 301–309.

[14] Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: SIGMOD'00, pp. 1–12.

[15] Le, B., Nguyen, H., Vo, B., 2011. An efficient strategy for mining high utility itemsets. International Journal of Intelligent Information and Database Systems 5 (2), 164–176.

[16] Lin, K.C, Liao, I.E., Chen, Z.S., 2011. An improved frequent pattern growth method for mining association rules. Expert Systems with Applications 38 (5), 5154–5161.

[17] Liu, B., Hsu, W., Ma, Y., 1998. Integrating classification and association rule mining. In: SIGKDD'98, pp. 80–86.

[18] Lucchese, B., Orlando, S., Perego, R., 2006. Fast and memory efficient mining of frequent closed itemsets. IEEE Transactions on Knowledge and Data Engineering 18 (1), 21–36.

[19] Vo, B., Le, B., 2011. Interestingness measures for mining association rules: combination between lattice and hash tables. Expert Systems with Applications 38 (9), 11630–11640.

[20] Vo, B., Coenen, F., Le, B., 2013. A newmethod for mining frequent weighted itemsets based on WIT-trees. Expert Systems with Applications 40 (4), 1256–1264.

[21] Wang, J., Han, J., Pei, J., 2003. CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: SIGKDD'03, pp. 236–245.

[22] Yun, U., Shin, H., Ryu, K.H., Yoon, E., 2012. An efficient mining algorithm for maximal weighted frequent patterns in transactional databases. Knowledge Based Systems 33, 53–64.