Journal of Computer Science 8 (4): 599-605, 2012
ISSN 1549-3636

Mining of Datasets with an Enhanced Apriori Algorithm

[1]Nandagopal, S., [2]V.P. Arunachalam and [3]S. Karthik
[1]Department of CSE, Sasurie College of Engineering, Tirupur, Tamilnadu, India
[2]SNS College of Technology, Coimbatore, Tamilnadu, India
[3]Department of CSE, SNS College of Technology, Coimbatore, Tamilnadu, India

Abstract: Problem statement: Classical association rules are mostly mining intra-transaction associations i.e., associations among items within the same transaction where the idea behind the transaction could be the items bought by the same customer on the same day. The goal of inter-transaction association rules is to represent the associations between various events found in different transactions. **Approach:** In this study, we break the barrier of transactions and extend the scope of mining association rules from traditional single-dimensional, intratransaction associations to N-Dimensional, inter-transaction associations. With the introduction of dimensional attributes, we lose the luxury of simple representational form of the classical association rules. Mining inter-transaction associations pose more challenges on efficient processing than mining intra-transaction associations because the number of potential association rules becomes extremely large after the boundary of transactions is broken. **Results:** Various tests also conducted using the data set collected from different Stock Exchange (SE).Various experimental results are reported by comparing with real life and synthetic datasets and we show the effectiveness of our work in generating rules and in finding acceptable set of rules under varying conditions. **Conclusion/Recommendations:** This study introduce the notion of N-Dimensional inter-transaction association rule, define its measurements: support and confidence and develop an efficient algorithm called Modified Apriori.

Key words: Association rules, intra-transaction associations, inter-transaction associations, n-dimensional, apriori

INTRODUCTION

Among all the data mining problems, discovering association rules from large databases is probably the most significant contribution from the database community to the field (Agrawal *et al*., 1993; Agrawal and Srikant, 1994; Dong and Han, 2007; Feng *et al*., 2002; Han and Fu, 1995; Kamber *et al*., 1997; Shankar *et al*., 2009). The most often cited application of association rules is market basket analysis using transaction databases from supermarkets and departmental stores. We can discover rules like:

R1: 80% of customers who bought diaper also bought beer (diaper => beer (20 and 80%))

where, 80% is the confidence level of the rule and 20% is the support level of the rule indicating how frequent the rule holds.

Association rules for prediction: The same concept can be applied to other applications as well. For example, to predict the stock market price movement (Tung *et al*., 2003), we can construct a transaction database in such a way that: each record (transaction) in the database represents one trading day and contains a list of winners (closing price is x% more than the previous day's closing price where x% is the trading overhead). Thus we can find rules like:

R2: When the prices of IBM and SUN go up, 80% of time the price of Microsoft goes up (on the same day).

While rule R2 reflects some relationship among the prices, its role in price prediction is limited. It is rather obvious that the traders may be more interested in the following kind of rules:

R3: If the prices of IBM and SUN go up, Microsoft's will most likely (80%of time) go up the next day.

Unfortunately, current association rule miners cannot discover this kind of rules.

The fundamental difference: There is a fundamental difference between rule R3 and the other

Corresponding Author: Nandagopal, S., Department of CSE, Sasurie College of Engineering, Tirupur, Tamilnadu, India

rules. The classical association rules express the associations among items purchased by one customer or share price movement within a day, i.e., associations among items within the same transaction record. We call them intra-transaction association rules. Sequential pattern discovery is also intra-transaction mining in nature because each sequence is treated as one transaction and the mining process is to find similarities among the sequences. On the other hand, rule R3 expresses the association among items from different transaction records. We call it inter-transaction association.

N-dimensional inter-transaction Association rules: In this stock movement prediction application, the association is along one dimension, the trading days. The concept can be extended further. If a database contains records about the time and location of buildings and facilities of a new city under development, we may be able to find such a rule:

R4: After McDonald and Burger King open branches, KFC will open a branch two months later less than a mile away.

Based on what have been described above, we propose N-dimensional inter-transaction association rules with the classical association rules as a special case.

The transaction database: Definition 1 let E = {e_1, e_2, …,e_u} be a set of literals, called events. Let D_1, D_2,…,D_n be a set of attributes. A transaction database is a database containing records in the form of (d_1, d_2,…,d_n, E_i) such that k (1≤k≤n) (d_k Є Dom (D_k)) where Dom(D_k) is the domain of attribute D_k and $E_i \subseteq$ E.A transaction database with n attributes is called an n-dimensional transaction database.

The attributes in an n-dimensional transaction database are called dimensional attributes. They describe the properties associated with the events, such as time and place. There are a wide range of application databases that can be viewed as n-dimensional transaction databases. The stock price movement database is a 1-dimensional transaction database. The example of urban development project can use a 2-dimensional transaction database where the two dimensional attributes are month and block number and the event list includes the buildings or facilities completed during the month at a particular block. In the current study, we will assume that the domain of a dimensional attribute can be divided into equal length intervals. For example, time can be divided into day, week, month, etc. and distance into meter, mile. The intervals can be represented by integers 0, 1, 2,… without losing generality. If we divide the space into n-dimensional cells each of which is identified by the associated n-ary tuple (d_1, d_2,… d_n), each transaction in the database represents a non-empty cell with some points (events) inside it.

Definition 2: Let T_i = (d_{i1}, d_{i2}, …, d_{in}, E_i) be a record in the transaction database. (d_{i1}, d_{i2}, …. , d_{in}) is the address of event e_i Є E_i. An event associated with its address is called an event instance, denoted by $\bar{e}_i$= e_i(d_{i1}, d_{i2}, …, d_{in}).

Figure 1 depicts a 2-dimensional transaction database. The dimensional attribute values of D_1 and D_2 have been mapped to integers; and there are four types of events, a, b, c and d. The database contains transactions:

T_1(1,1,a,b,c),T_2(2,1,b),….,T_{24}(5,5,c).

From the database, we can identify such event instances as a(1,1), b(1,1), c(1,1), b(2,1), c(5,5) and so on.

N-dimension inter-transaction association rules: The objective of inter-transaction association rules is to represent the associations between various events found in different transactions. With the introduction of dimensional attributes, we lose the luxury of simple representational form of the classical association rules. Some definitions are needed before we formally define such rules.

Definition 3: Given a set of event instances $\bar{E}$ = { $\bar{e}_1$, $\bar{e}_2$,…., $\bar{e}_m$} where $\bar{e}_i$ is in the form of e_i(d_{i1},d_{i2},…..,d_{in}) (1≤ i ≤ m).An n-ary tuple (d_{01},d_{02},…..,d_{0n}) with d_{0k}=Min(d_{ik})(1≤ k ≤ n,1≤ I ≤ m) is called the base address for event instance set $\bar{E}$, denoted by E-BASE($\bar{E}$).(d_{i1}-d_{01},d_{i2}-d_{02},….,d_{in}-d_{0n}) is the relative address of all member event instances in set $\bar{E}$ form the address of event instance set $\bar{E}$,denoted by E-ADDR($\bar{E}$). In Fig. 1, there are three shadowed areas indicating three sets of event instances:

$\bar{E}_1$={a(1,1),c(1,2),d(2,2)}, $\bar{E}_2$={a(3,2),b(4,2),d(4,1)},and $\bar{E}_3$={a(1,3),c(1,4),d(2,4)}.

According to the above definition, we have:
E-BASE ($\bar{E}_1$) = (1,1),E-BASE($\bar{E}_2$)=(3,1) and E-BASE ($\bar{E}_3$) = (1, 3).The addresses for $\bar{E}_1$ and $\bar{E}_2$ are
E-ADDR ($\bar{E}_1$) = {(0, 0),(0,1),(1,1)} and
E-ADDR($\bar{E}_2$) = {(0,1),(1,1),(1,0)},respectively

Since two events in a set may have the same address of the set will have those duplicates removed. For example, referring to Fig. 1, the address of event instance set {a(1,1),b(1,1),c(1,2),d(2,2)} will be {(0,0),(0,1),(1,1)}.

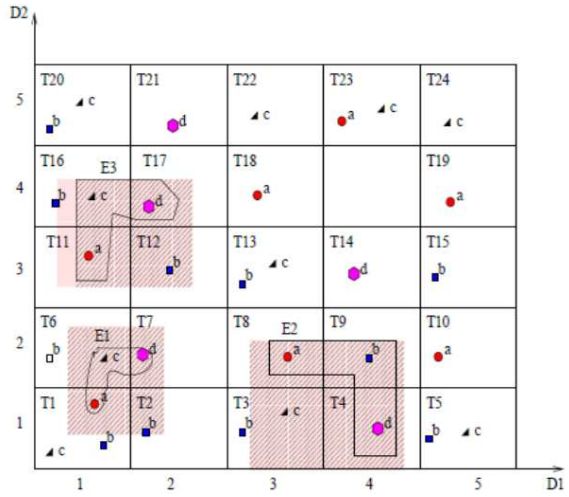

Fig. 1: Graphical representation of a 2-dimensional transaction database

Note that, by using relative address, two sets of event instances may have the same address with respect to their base addresses. For set:

$\bar{E}_3$={a(1,3),c(1,4),d(2,4)},E-BASE($\bar{E}_3$) = (1, 3)

And E-ADDR ($\bar{E}_3$)={(0,0),(0,1),(1,1)},which is the same as E-ADDR($\bar{E}_1$).In other words, event instance sets $\bar{E}_1$ and $\bar{E}_3$ have the same address but different base addresses.The notion of base address and address can be extended to the set of transactions.

Definition 4: Given a set of transactions T={T_1,T_2,…,T_8} where transaction T_j is in the form of (d_{j1},d_{j2},…,d_{jn}, E_j) (1 ≤ j ≤ s). An n-ary tuple (d_{01},d_{02},…,d_{0n}) with d_{0k}=Min(d_{jk})(1 ≤ k ≤ n, 1 ≤ j ≤ s) is called the base address for transaction set T, denoted by T-BASE(T).The relative address of all member transactions in T form the address of transaction set T, denoted by T-ADDR(T).

In our example, for transaction set T={T_1,T_2,T_6} and:

T'={T_4,T_8,T_9},T-BASE(T)=(1,1)and T-BASE(T')=(3,1).

The address of these two transaction sets are:

T-ADDR(T)={(0,0),(1,0),(0,1)}

And:

T-ADDR(T')={(1,0),(0,1),(1,1)}, respectively.

Definition 5: Given a set of transactions T={T_1,T_2,…,T_8} where T_j is in the form of (d_{j1},d_{j2},….,d_{jn},E_j) (1 ≤ j ≤ s) and a set of event instance $\bar{E}_T$={ $\bar{e}_1$, $\bar{e}_2$,…, $\bar{e}_m$ } where $\bar{e}_i$ is in the form of e_i (d_{i1},d_{i2},…,d_{in}) (1 ≤ i ≤ m).T is said to contain $\bar{E}_T$ if (1) for every $\bar{e}i$ ε $\bar{E}_T$, there exists a transaction T_j ε T so that e_i ε E_j ,and the relative address of $\bar{e}i$ in E-ADDR($\bar{E}_T$) is the same as the relative address of T_j in T-ADDR(T). (2) |E-ADDR($\bar{E}_T$) = |T-ADDR(T)|'

In the definition, the first condition guarantees that each event is among certain event list of a record in the transaction database. The second condition requires the transaction set is a minimum set. In our example, transaction set {T_1,T_6,T_7} contains event instance set {a(0,0),c(0,1),d(1,1)}.{T_{11},T_{16},T_{17}} and {T_8,T_{13},T_{14}} contain the same set of event instances.

Now we are ready to define n-dimensional inter-transaction association rules.

Definition 6: An inter-transaction association rule is an implication of the form X ==> Y, where (1) X and Y are sets of event instances in the form of e_i(d_{i1},d_{i2},…,d_{in}) where (d_{i1},d_{i2},…,d_{in})is the address of e_i relative to E-BASE(X U Y),ie.,(d_{01},d_{02},….,d_{0n});(2) e_i ε E, d_{0k} ε Dom(D_k),(d_{ik}+d_{0k}) ε Dom(D_k)(1 ≤iI ≤ u,1 ≤ k ≤ n) and (3) X ∩ Y = Φ.

For the database shown in Figure 1, one such association rule is a (0, 0), c (0, 1) => d(1, 1).

Since the inter-transaction association rules involve more than one transaction, the definitions of support and confidence, which are widely used as the objective interestingness measure of association rules in intra-transaction association rules, need to be modified. The reason is that, the number of transactions in the database can no longer be used as the measure. To address the problem, we introduce the following notion.

Definition 7: Let T_{xy} be the set of transaction sets containing event instance set X UY, T_{xy} be the set of transaction sets that possibly contain X U Y and T_x be the set of transaction sets containing X, the support and confidence of an inter-transaction association rule X=> Y are defined as:

Support = $|T_{xy}|/|T_{xy}|$
Confidence = $|T_{xy}|/|T'_{xy}|$

Respectively, where, T'_{xy} = {τ | (τ ε T_{xy} } ^. ℓ (ℓ ε T_x) (ℓ⊂ τ)}

As an example, we compute the support and confidence of the association rule:

A(0,0),c(0,1)⇒d(1,1)

In database shown in Figure 1. Here, X = {a(0,0),c(0,1)} and Y = {d(1,1)}.There are three transaction sets that contain the event instance set X U Y:

$T_{xy} = \{\{T_1,T_6,T_7\},\{T_8,T_{13},T_{14}\},\{T_{11},T_{16},T_{17}\}\}, |T_{xy}| = 3$

The transaction database contains 24 records. The number of transaction sets that possibly contain X U Y is $|T_{xy}| = 13$. Note that, the database does not contain any transaction with address (4,4), which reduces the number of transaction sets that possibly contain the event instance set. In addition to the transaction set in T_{xy}, $\{T_{18},T_{22},T_{23}\}$ is a transaction set that possibly contain XUY and surely contains X: $T'_{xy} = \{\{T_1,T_6,T_7\},\{T_8,T_{13},T_{14}\},\{T_{11},T_{16},T_{17}\},\{T_{18},T_{22},T_{23}\}\}|T'_{xy}| = 4$.Therefore, the support and confidence for the above rule is 3/13 and 3/4, respectively. Note that we do not count the event a and c in transaction T_{19} and T_{24} when computing the confidence, as no transaction set can be formed with T_{19} and T_{24} that possibly contains X U Y.

Mining 1 Dimensional inter-transaction association rules: Mining n-dimensional inter-transaction rules is obviously a computation intensive problem (Lee *et al*., 2006). Comparing to the classical association rules, the search space is much bigger as the number of possible rules increases dramatically with both the number of transactions and the number of dimensions. To investigate the feasibility of mining inter-transaction rules, we implemented two algorithms by extending the Apriori-based algorithm to mine 1-dimensional inter-transaction association rules and applied it to the problem of stock price movement prediction. To limit the search space, we used an additional mining parameter, MAX_INTERVAL, to define a sliding window. Only the associations among the events that co-occurred within the window are interested. In general, the mining process of n-dimensional inter-transaction rules can be divided into three phases: data preparation, Frequent-item set discovery and candidate generation.

Data preparation: The transaction database is prepared for mining from operational databases. The major task in this phase is to organize the transactions based on intervals of the dimensional attribute(s). For example, to find the long term movement regularities of stock prices across different weeks (months), we need to transform daily price movement into weekly (monthly) group. After such transformation, each record in the database will contain an interval value and a list of items.

Frequent-Item set discovery: In this phase, we find the set of all frequent item sets. A k-item set is of the form $\{i_1(d_{i1}), i_2(d_{i2}), \ldots, i_k(d_{ik})\}$, where event ij, $1 \leq j \leq k$, is attached by a non-negative value d_{ij} indicating the relative address with respect to the base address of the set. For example, a 3-itemset {a(0), b(1), c(3)} contains three event instances expressed in relative addresses along the dimension. That is, taking a transaction containing event a as the base transaction, b(1) is an event b contained in a transaction with 1 unit distance away from the base transaction and c(3) represents an event c in a transaction 3 unit distances away from the base transaction. This is quite different from the classical definition of item set $\{i_1,i_2,\ldots,i_k\}$ in which all items lie within the same transactions.

To find the frequent item sets, two algorithms, E-Apriori and M-Apriori, were implemented which are extensions of Apriori based algorithms (Agrawal and Srikant, 1994; Han *et al*., 2000; Chu *et al*., 2009; Srikant and Agrawal, 1995). Let L_k represent the set of frequent k-item sets and C_k the set of candidate k-item sets. Both algorithms make multiple passes over the database. Each pass consists of two phases. First, the set of all frequent (k-1) item sets L_{K-1}; found in the (k-1)th pass, is used to generate the candidate item set C_k. The candidate generation procedure ensures that C_k is a super set of the set of all frequent k-item sets. The algorithms now scan the database. For each list of consecutive transactions, they determine which candidates in C_k are contained and increment their counts. At the end of the pass, C_k is examined to check which of the candidates are actually frequent, yielding L_k. The algorithms terminate when L_k becomes empty.

As previously reported in (Han *et al*., 2000; Feng *et al*., 2002), the processing cost of the first two iterations (i.e., obtaining L_1 and L_2) dominations the total mining cost. The reason is that, for a given minimum support, we usually have a very large L_1, which in turn results in a huge number of itemsets in C_2 to process. In the inter-transaction association rules, this situation becomes much more serious as a lot of additional 2-itemsets like {a(0),a(1)} may be added into C_2,thus leading to a huge amount of $|C_2|$. In order to construct a significantly smaller C_2, EH-Apriori adopts a similar technique of hashing as (Park *et al*., 1995; Han *et al*., 2000) to filter out unnecessary candidate 2-itemsets. When the support of candidate C_1 is counted by scanning the database. EH-Apriori accumulates information about candidate 2-itemsets in advance in such a way that all possible 2-itemsets are hashed to a hash table. Each bucket in the hash table consists of a number to

represent how many itemsets have been hashed to this bucket thus far. Such resulting hash table can be used to greatly reduce the number of 2-itemsets in C_2.In the following, we describe how E-Apriori and M-Apriori generates candidates and count their supports.

Candidate generation:
Pass 1: Let I={1,2,…m} be a set of items in a database. To generate the candidate set C_1 of 1-itemsets, we need to associate all possible intervals with each item. That is:

C1={1(0),1(1),…..,1(MAX_INTERVAL),
2(0),2(1),…..,2(MAX_INTERVAL),
……m(0),m(1),…..,m(MAX_INTERVAL)}

Starting from transaction $T_c(1\leq c \leq|D|)$, T_{c+di} transaction is scanned to determine whether item i exist. If so, the count of $\{i(d_i)\}$ increases by one. Through one scan of the database ,we can get the large set L_1.

Pass 2: For any two 1-itemsets of L_1,a(0) and $b(d_b)((d_b=0\text{^}a<b)V(d_b \neq 0))$, we generate a 2-itemset $\{a(0),b(d_b)\}$, that is $C_2=\{\{a(0),b(d_b)\}|(d_b \neq 0)V(d_b=0\text{^}a<b)\}$. Of all 2-itemsets in C_2, the minimal interval value is always 0:

Pass k > 2

Given L_{k-1}, the set of all frequent (k-1) item sets, the candidate generation procedure returns superset of the set of all frequent k-item sets. This procedure has two parts. In the join phase, we join L_{k-1} with L_{k-1}:

Insertinto C_k Select $p.item_1(d_{item1}),p.\ item_2(d_{item2})$,
…..,$p.item_{k-1}(d_{itemk-1}),q.item_{k-1}(d_{itemk-1})$ From L_{k-1} p,L_{k-1} q

Where:
p. $item_1(d_{item1})$ = $q.item_1(d_{item1})$,….,
p. $item_{k-2}(d_{itemk-2})$ = $q.item_{k-2}(d_{itemk-2})$,
p. $item_{k-1}(d_{itemk-1})<$ = $q.item_{k-1}(d_{itemk-1})$

Counting support of candidates: To facilitate the efficient support counting process, a candidate C_k of k-itemsets is divided into k groups,with each group Go containing o number of items whose interval is $0(1 \leq o \leq k)$. For example, a 3-item set:

C_3={{ a(0),a(1),b(2)},{c(0),d(0),d(2)},
{a(0),b(0),h(3)},{l(0),m(0),n(0)},
{p(0),q(0),r(0)}}

Is divided into three groups:

G_1={{a(0),a(1),b(2)}},
G_2={{ c(0),d(0),d(2)},{a(0),b(0),h(3)}},
G_3={{ l(0),m(0),n(0)},{p(0),q(0),r(0)}}

Each group is stored in a modified hash-tree. Only those items with interval 0 participate the construction of this hash-tree, e.g., in group 2, only {a(0),b(0)},{(c(0),d(0)} enter the hash-tree. The construction process is similar to that Apriori (Agrawal and Srikant, 1994; Rahman and Balasubramanie, 2009). The rest items, e.g., h(3),d(2), are simply attached to the corresponding itemsets, e.g., {a(0),b(0)} and {c(0),d(0)} respectively, in the leaves of the tree. Upon reading one transaction of the database, every hash-tree is tested. If one itemset is contained, its attached itemsets whose intervals are larger than 0 will be checked against the successive transactions. In the above example, if {a(0),b(0)} exists in the current transaction tc, then tc+3 transaction will be scanned to see whether it contains item h. If so, the support of 3-itemsets {a(0),b(0),h(3)} will increase by 1.

E-Apriori and M-Apriori share the same procedures, except that Pass 1, M-Apriori hashes all 2-itemsets like $\{i_1(0),i_2(d_{i2})\}(d_{i2} \neq 0)$ contained in the current series of transactions into the corresponding buckets of a hash Table and prunes unnecessary 2-itemsets from C_2 in pass 2, whose corresponding bucket values in the Hash Table are less than support threshold.

MATERIALS AND METHODS

To assess the performance of the proposed algorithms, some preliminary experiments were conducted using synthetic data. Table 1 listed one set of the results obtained using a transaction database with 10,000 records with each records containing 5 items on the average. The total number of items is 500. The maximum interval is set to 3. (T – ave_tran_size, N – item_num, D – tran_num,R – max_interval). The results indicate that, with the given setting, the execution time is acceptable, especially if M-Apriori algorithm is used.

Table 1: Comparisons of E-Apriori and M-Apriori

	T5-R3-N500-D10K			
	Support = 0.6%		Support = 0.7%	
	E-Apriori	M-Apriori	E-Apriori	M-Apriori
$\|L_{1\|}$	348.0	348.0	319.0	319.0
t_1	0.4s	3.9s	0.4s	3.9s
$\|C_2\|$	363312.0	43047.0	305282.0	17403.0
$\|L_2\|$	86.0	86.0	29.0	29.0
t_2	537.6s	65.5s	224.8s	18.1s
t_1+t_2	538.0s	69.4s	225.2s	22.0s
Total mining time	538.0s	70.0s	226.2s	23.0s

It is also found that although the execution time of the fist pass of M-Apriori is slightly longer than that of E-Apriori due to the extra overhead required for building Hash Table, it incurs significantly smaller execution time than E-Apriori in later Pass 2 and less $|C_2|$ results in much less time to test against each transaction of the database.

RESULTS AND DISCUSSION

Some tests conducted using the data set collected from Singapore Stock Exchange (SES). The available stock price data was used to generate two data sets, WINNER and LOSER. A stock is a winner if its closing price of the day is 3% more than the previous day closing. A stock is a loser otherwise. The WINNER (LOSSER) data set contains the date and the winners(losers) of that day. Each data set contains 250 records corresponding to 250 trading days in 2006. Since the major trend for SES in 2006 is down side, there are a few of winners everyday but a large number of losers. From the LOSER set, one example rule found is {UOL(0), SIA(1)} ⇒ DBS(2). That is, if UOL goes down and SIA goes down the following day, DBS will go down the second day with confidence more than 99%. Since the WINNER data set is small, we do not have rules with large support. However, if after lowering the support, we can find rules such as {HAISUNWT(0), KIMENGWT(0)} ⇒ HAISUNWT(1). The following table shows the performance of the proposed algorithm.

The necessity of having N-dimensional inter-transaction association rules is clear. The definition of such rules is lengthy (based on our study).

CONCLUSION

We believe that, the proposed n-dimension inter-transaction association rules represent a uniform treatment to a few association-related data mining problems. Furthermore, there seems to be highly promising to apply such notions in textual mining, spatial data mining, multi-media data mining.

REFERENCES

Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules. Proceedings of the 20th Very Large Data Bases Conference Santiago, (VLDBCS' 94), Chile, pp: 1-13.

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD Conference on Management of Data, (MD' 93), ACM, New York, USA, pp: 207-216. DOI: 10.1145/170035.170072

Chu, C.J., V.S. Tseng, T. Liang, 2009. An efficient algorithm for mining high utility itemsets with negative item values in large databases. Applied Math. Comput., 215: 767-778. DOI: 10.1016/j.amc.2009.05.066

Dong, J. and M. Han, 2007. IFCIA: An efficient algorithm for mining intertransaction frequent closed itemsets. Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery, Aug. 24-27, IEEE Xplore Press, Haikou, pp: 678-682, DOI: 10.1109/FSKD.2007.352

Feng, L., J.X. Yu, H. Lu and J. Han, 2002. A template model for multidimensional inter-transactional association rules. VLDB J., 11: 153-175. DOI: 10.1007/s00778-002-0069-6

Han, J. and Y. Fu, 1995. Discovery of multiple-level association rules from large databases. Proceedings of the 21st Conference on Very Large Data Bases, (VLBD' 95), Morgan Kaufmann Publishers Inc. San Francisco, pp: 420-431.

Han, J., J. Pei and Y. Yin, 2000. Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM, New York, pp: 1-12. DOI: 10.1145/342009.335372

Kamber, M., J.H. Jenny, Y. Chiang, J. Han and J.Y. Chiang, 1997. Metarule-guided mining of multi-dimensional association rules using data cubes. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, California, pp: 207-210.

Lee, A.J.T., W.C. Lin., C.S. Wang, 2006. Mining association rules with multi-dimensional constraints. J. Syst. Softw., 79: 79-92. DOI: 10.1016/j.jss.2005.03.005

Park, J.S., M.S. Chen and P.S. Yu, 1995. An effective hash-based algorithm for mining association rules. Proceedings of the ACM SIGMOD Conference on Management of Data, (DM' 95), ACM, New York, pp: 175-186, DOI: 10.1145/223784.223813

Rahman, A.M.J.M.Z. and P. Balasubramanie, 2009. Weighted association rule mining using closed itemset lattices in parallel. Int. J. Comput. Sci. Network Security, 9: 247-253.

Shankar, S., T. Purusothaman, S. Jayanthi and N. Babu, 2009. A Fast Algorithm for Mining High Utility Itemsets. Proceedings of the IEEE International Advance Computing Conference, IEEE Xplore Press, Mar. 6-7, Patiala, pp: 1459-1464. DOI: 10.1109/IADCC.2009.4809232

Srikant, R. and R. Agrawal, 1995. Mining generalized association rules. Future Gene. Comput. Syst., 13: 161-180. DOI: 10.1016/S0167-739X(97)00019-8

Tung, A.K.H, H. Lu., J. Han and L. Feng, 2003. Efficient mining of intertransaction association rules. IEEE Trans. Know. Data Eng., 15: 43-56. DOI: 10.1109/TKDE.2003.1161581