# Mining Association Rule by ECLAT Method Using Transaction Data

Pan Myat Mon, Renu, Thet Lwin Oo
*Computer University (Myeik), Myanmar*
*panmyatmon86@ gmail.com, renushi@ gmail.com*

## Abstract

*Association rule mining is a process that identifies links between sets of correlated objects in transactional databases where each transaction contains a list of items. Association rule is one of the well-defined algorithms, whose significance is measured via support and confidence factor, are intended to identify rules of the type. This system is the development of transactions data analysis system. The important problems of data mining are mining frequent itemsets and generating association rules from databases of transactions where each transaction consists of a set of items. Our proposed system is based on Association Rule Mining using Equivalence CLASS Transformation (ECLAT) method to find frequent-patterns. This method can also reduce the number of candidate itemsets. It is not required scanning the complete database over and over again. So, it also saves the time.*

**Keywords**: Data Mining, Association Rules Mining (ARM), Frequent-pattern Mining Algorithm, Performance Improvements, Knowledge Discovery in Database (KDD).

## 1. Introduction

Data mining is the task of discovering intersection patterns from large amount of data, where the data can be stored in databases, data warehouses, or other information repositories.

Association rule mining searches for interesting relationships among items in a given data set [4]. Association rule mining has a wide range of applicability such as Market basket analysis, Medical diagnosis/ research, Website navigation analysis, Homeland security and so on [6].

Association rules are used to identify relationships among a set of items in database. These relationships are not based on inherent properties of the data themselves (as with functional dependencies), but rather based on co-occurrence of the data item.

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional data sets. Frequent pattern mining techniques can also be extended to solve many other problems, such as iceberg-cube computation and classification. Thus the effective and efficient frequent pattern mining is an important and interesting research problem.

By performing data mining, interesting knowledge, regularities, or high-level information can be extracted from databases and viewed or browsed from different angles. The discovered knowledge can be applied to decision making, process control, information management, and query processing. Therefore, data mining is considered one of the most important frontiers in database and information systems and one of the most promising interdisciplinary developments in the information technology [4].

Association rule and frequent itemset mining became a widely researched area, and hence faster and faster algorithms have been presented.

This system is to develop a method for analysis buying patterns of the customers in the “Food Departmental Store”. Our proposed system is mined the frequent itemsets on transaction data of that store and then the important decisions are made by applying strong association rule. . By using this system, the food departmental stores are promoted sales and developed.

## 2. Related Work

Today there are several efficient algorithms that cope with the popular and computationally expensive task of association rule mining. Actually, these algorithms are more or less described on their own. J.Hipp, et.al, [5] explained the fundamentals of association rule mining and moreover derive a general framework. Based on this they described today's approaches in context by pointing out common aspects and differences. After that they thoroughly investigated their strengths and weaknesses and carry out several runtime experiments. It turned out that the runtime behavior of the algorithms was much more similar as to be expected.

A fast algorithm has been proposed for solving data structure or reduces the cost in frequent pattern mining. M.H.Margahny, et.al, used the "*TreeMap*" which was a structure in Java language. Also they presented "*Arraylist*" technique that greatly reduces they need to traverse the database. Moreover they presented experimental results which showed their structure outperforms all existing available algorithms in all common data mining problems. [6]

T.Calders, et.al, [2] presented a new itemset search space travel strategy, which allowed depth-first itemset mining algorithms to exploit the frequency knowledge of all subsets of a given candidate itemset. Second they presented a depth-first non-derivable itemsets mining algorithm, which used that traversal as it needs for every generated candidate itemset, all of its subsets. Third, they generalized the diffsets technique and store the cover of an itemset containing several negated items. These claims are supported by several experiments on real life datasets.

## 3. Motivation

The rapid development of computer technology, especially increased capacities and decreased costs of storage media, has led businesses to store huge amounts of external and internal information in large databases at low cost. Mining useful information and helpful knowledge from these large databases has thus evolved into an important research area [6]. Among them association rule mining has been one of the most popular data-mining subjects, which can be simply defined as finding interesting rules from large collections of data.

This system is to develop a new effective method for analyzing buying patterns of the customers in "Food Departmental Store". This system is used the association rule approach with ECLAT method, it can substantially reduce the total cost of vertical format mining of frequent itemsets. Furthermore, it can also reduce the number of candidate itemsets. It is not required scanning the complete database over and over again. So, it also saves the time. Our proposed system is performed on transaction data of the store and then the important decisions are made by applying strong association rules. This system is supported the food departmental stores' owners for promoting sales and developing their store.

## 4. Theoretical Background

Let $I = \{ a_1, a_2, . . . ,a_m\}$ be a set of items and a transaction database DB = $\{ T_1, T_2, . . . , T_n\}$ where $T_i$ (i = [1 . . . n] ) is a transaction which consists of a set of items in i. The support (or) occurrence frequency of a pattern A, where A is a set of items, is the number of transactions containing A in DB. A pattern A is frequent if A's support is no less than a predefined minimum support threshold, s.

### 4.1. Association Rule Mining

Association rule mining finds interesting association or correlation relationships among a large set of data items. Association rule mining can be viewed as a two steps process:

1. **Find all frequent itemsets**: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.
2. **Generate strong association rules from the frequent itemsets**: By definition, these rules must satisfy minimum support and minimum confidence [4].

### 4.2. Proposed Equivalence CLASS Transformation (ECLAT) Method

Mine frequent patterns from a set of transactions in item-TID-set format (that is, {items: 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. First, transform the horizontally formatted data to the vertical format by scanning the data set once. Mining can be performed on this data set by intersecting the TID-sets of every pair of frequent single item. The support count of an itemset is simply the length of the TID-set of the itemset. If the minimum support count is 2, the association rules can be generated from any frequent itemsets [4].

ECLAT is introduced that combines Depth First Search (DFS) with TID-list intersection to address the memory problem. When using DFS, it suffices to keep the TID-list on the path from the root down to the node itemset currently investigates. Splitting the database done by partition is no longer needed. ECLAT employs an optimization called “fast intersection,” in that whenever two TID-lists are intersected, we only consider the resulting TID-list if its cardinality reaches minimum support. In other words, each intersection is eliminated as soon as it does not meet the minimum support [5].

### 4.2.1 ECLAT Algorithm

The algorithm for ECLAT method is as follow:

```
Input: $D$, s, $I \subseteq \mathcal{I}$
Output: $\mathcal{F}[I](D, s)$
$\mathcal{F}[I]: = \{\}$
for all $i \in \mathcal{I}$ occurring in $D$ do
    $\mathcal{F}[I]: = \mathcal{F}[I] \cup \{I \cup \{i\}\}$
    //Create $D^i$
    $D^i: = \{\}$
    for all $j \in \mathcal{I}$ occurring in $D$ such that $j>i$ do
        $C: =$ cover $(\{i\}) \cap$ cover $(\{j\})$
        if $|C| \geq$ s then
            $D^i: = D^i \cup \{(j, C)\}$
        end if
    end for
    //Depth-first recursion
    Compute $\mathcal{F}[I \cup \{i\}](D^i, s)$
    $\mathcal{F}[I]: = \mathcal{F}[I] \cup \mathcal{F}[I \cup \{i\}]$
end for
```

Given a transaction database $D$ and a minimum support threshold s, denote the set of all frequent itemsets with the same prefix $I \subseteq \mathcal{J}$ by $\mathcal{F}$ [$I$] ($D$, s). ECLAT recursively generates for every item $i \in \mathcal{J}$ the set $\mathcal{F}$ [{$i$}] ($D$, s). (Note that $\mathcal{F}$ [{}] ($D$, s) $=\cup_{i \in \mathcal{J}} \mathcal{F}$ [{$i$}] ($D$, s) contains all frequent itemsets).

For the sake of simplicity and presentation, we assume that all items that occur in the transaction database are frequent. In practice, all frequent items can be computed during an initial scan over the database, after which all infrequent items will be ignored.

In order to load the database into main memory, ECLAT transforms this database into its vertical format. I.e., instead of explicitly listing all transaction, each item is stored together with its cover (also called tidlist). In this way, the support of an itemset X can be easily computed by simply intersecting the covers of any two subsets. Note that a candidate itemset is represented by each set $I$ U {$i,j$} of which the support is computed at line 7 of the algorithm. A technique that is regularly used is to reorder the items in support ascending order. In ECLAT, such reordering can be performed at every recursion step between line 11 and line 12 in the algorithm [2].

### 4.2.2 The Advantages of ECLAT Method

The advantages of ECLAT Method are

(1) The number of candidate itemsets that are generating reduced. By reordering, the generating sets tend to have lower support which results in fewer candidates.

(2) The generating sets tend to have lower support is that their tid-lists are smaller.

(3) It does not require scanning the complete database over and over again [2].

### 4.3. Association Rule Problem Description

The association rule mining problem can be formally stated as follows: Let $I$ = {$i$1; $i$2; _ _ _ ; im} be a set of items. Let $D$ be a set of transactions, where each transaction $T$ is a set of items such that $T \subseteq I$ associated with each transaction is a unique identifier, called its TID. We say that a transaction $T$ contains $X$, a set of some items in $I$, if $X \subseteq T$. An association rule is an implication of the form, $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \phi$. The rule $X \Rightarrow Y$ holds in the transaction set $D$ with confidence c if c% of transactions in $D$ that contain $X$ also contain $Y$ . The rule $X \Rightarrow Y$ has support s in the transaction set $D$ if s% of transactions in $D$ contain $X$ U $Y$ [7]. This is taken to be the conditional probability, P ($Y \backslash X$). That is,

$$\text{Supp } (X \Rightarrow Y) = \text{P } (X\text{U}Y) = \text{Supp-count}(X \text{ U } Y)$$

$$\text{Conf } (X \Rightarrow Y) = \text{P } (Y \backslash X) \quad = \quad \frac{\text{Supp-count } (X\text{U}Y)}{\text{Supp-count } (X)}$$

Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called strong [4].

## 5. Design Diagram of the Proposed System

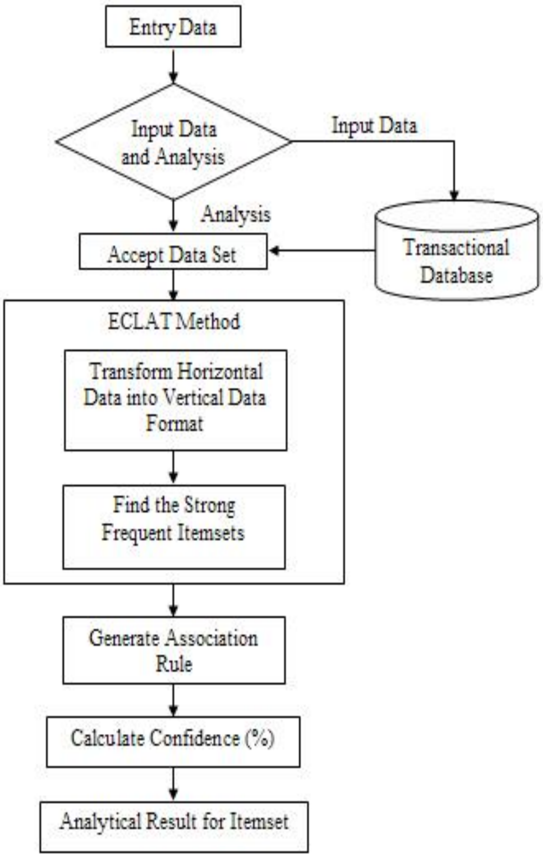
Entry Data
Input Data and Analysis
Input Data
Analysis
Transactional Database
Accept Data Set
ECLAT Method
Transform Horizontal Data into Vertical Data Format
Find the Strong Frequent Itemsets
Generate Association Rule
Calculate Confidence (%)
Analytical Result for Itemset

**Figure 1.** Design Diagram of the Proposed System

The above Figure 1 is the design diagram of the proposed system. This system is an association rule mining system for "Food Departmental Store". When the data set of the authorized person is entered, this system checks the data is input or to analyze. If the data is input, data then stores to it in the transactional database. If the data is analyzed, this system finds the frequent itemsets for transaction data according to ECLAT method.

First, transforms the horizontally formatted data to the vertical format by scanning the data set once. Then, it finds the strong frequent itemsets on this data set by intersecting the TID sets of every pair of frequent single item. If the minimum support count is 2, the association rules can be generated from any frequent itemsets. Then, the confidence is calculated on the association rules. If the minimum confidence is 70%, generates the strong association rules.

### 5.1. Process Flow of the System

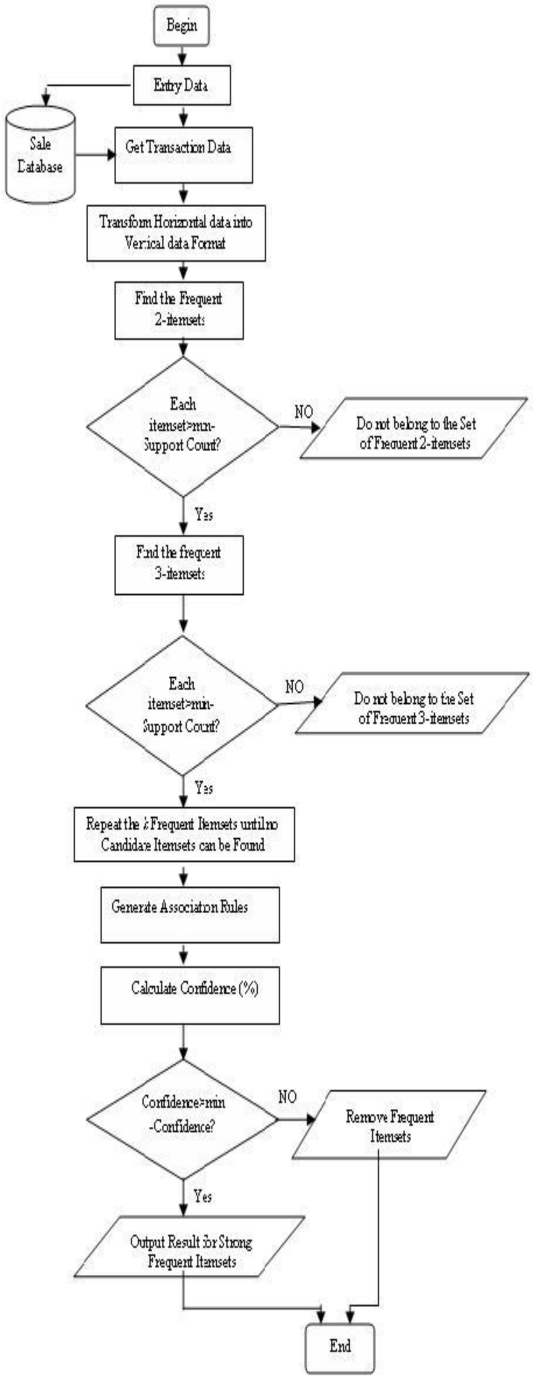
Begin
Entry Data
Sale Database
Get Transaction Data
Transform Horizontal data into Verical data Format
Find the Frequent 2-itemsets
Each itemset>min-Support Count?
NO
Do not belong to the Set of Frequent 2-itemsets
Yes
Find the frequent 3-itemsets
Each itemset>min-Support Count?
NO
Do not belong to the Set of Frequent 3-itemsets
Yes
Repeat the k Frequent Itemsets until no Candidate Itemsets can be Found
Generate Association Rules
Calculate Confidence (%)
Confidence>min -Confidence?
NO
Remove Frequent Itemsets
Yes
Output Result for Strong Frequent Itemsets
End

**Figure 2.** Process Flow of the System

The above figure 2 is the process flow of our proposed system. First, when the data set of the authorized person is entered, the data is recorded in sale database. This process get transaction data from sale database and transform the horizontally formatted data to the vertical format by scanning the data set once. Then, it finds the 2-frequent itemset on this data by intersecting the TID sets of every pair of frequent single item. The minimum support count is 2. If minimum support count is less than 2, then the itemset do not belong to the set of frequent 2-itemsets. If minimum support count is greater than 2, finds the 3-frequent itemset by intersecting the TID sets of every pair of frequent 2-itemset. If minimum support count is less than 2, then the itemset do not belong to the set of frequent 3-itemsets. This process repeats, until no frequent itemset or no candidate itemsets can be found. Finally, the association rule is generated from any frequent itemsets. Then, the confidence is calculated on the association rules. If the confidence is less than 70%, these association rules are removed. If the minimum confidence is 70%, these association rules are generated as strong association rules.

## 6. Experimental Data Sets and Result

In this section, we present the frequent pattern-mining algorithm. All the experiments are performed on a 1.7 GHz Pentium PC machine with 128 MB main memory, running on Window XP operating system. All programs were developed under the visual basic compiler, version 6.0.

The first algorithm proposed to solve the association rule mining problem was divided into two phases. In the first phase, all frequent itemsets are generated (or all frequent rules of the form X$\Rightarrow$ {}). The second phase consists of the generation of all frequent and confident association rules. Almost all association rule mining algorithms comply with this two phased strategy.

We report experimental results on Mining Association Rules with ECLAT method for Food Departmental Store as its domain. This store has one hundred selling items. This system leads to focus on the search of SQL database which stores many transactions for one year. Among them, we pick up 1500 transaction records to analyze. Based on the resulted frequent patterns, we mine association rules. Association rules are often used by retail stores to analyze market basket transaction. Market basket analysis is the analysis of customer's buying habits by finding association between the different items that customers place in our "shopping baskets".

In a food departmental store, we would like to learn more about the buying habits of customers. For instance, if customer A buys bread and butter, he/she also buys coffee from the food departmental store, then customer's identifier is T100 and list of items are {bread, butter, coffee}. If customer B buys butter and sugar, then customer's identifier is T200 and list of items are {butter, sugar}. If customer C buys butter and eggs, then customer's identifier is T300 and list of items are {butter, eggs} and so on.

Suppose a transactional database D with five items *I* = {bread, butter, eggs, sugar, coffee}. There are nine transactions in this database, that is, D = 9.

This database is the horizontal data format of the transaction database as shown in Table 1.

**Table 1.** Horizontal Data Format for Food Departmental Store

| TID | List of Itemsets |
|---|---|
| T100 | {Bread, Butter, Coffee} |
| T200 | {Butter, Sugar} |
| T300 | {Butter, Eggs} |
| T400 | {Bread, Butter, Sugar} |
| T500 | {Bread, Eggs} |
| T600 | {Butter, Eggs} |
| T700 | {Bread, Eggs} |
| T800 | {Bread, Butter, Eggs, Coffee} |
| T900 | {Bread, Butter, Eggs} |

First, the transaction dataset transforms the horizontally formatted data to the vertical data format by scanning the dataset once as shown in Table 2. The support count of an itemset is simply the length of the TID-set of the itemset.

**Table 2.** Vertical Data Format for Food Departmental Store

| Itemsets | TID-Set | Support-Count |
|---|---|---|
| Bread | {T100, T400,T500,T700,T800,T900} | 6 |
| Butter | {T100, T200,T300, T400,T600,T800,T900} | 7 |
| Eggs | {T300,T500,T600,T700,T800,T900} | 6 |
| Sugar | {T200,T400} | 2 |
| Coffee | {T100,T800} | 2 |

Mining can be performing on this data set by intersecting the TID-sets of every pair of frequent single items. The minimum support count is 2. Because every single item is frequent in Table 2, there are 10 intersection performed in total, which lead to 8 nonempty 2-items as shown in Table 3. Because the itemsets {Bread, Sugar} and {Eggs, Coffee} each contain only one transaction, they do not belong to the set of frequent 2-itemsets.

**Table 3.** The 2- itemsets in Vertical Data Format for Food Departmental Store

| Itemsets | TID-Set | Support-Count |
|---|---|---|
| {Bread, Butter} | {T100, T400, T800, T900} | 4 |
| {Bread, Eggs} | {T500, T700, T800, T900} | 4 |
| {Bread, Sugar} | {T400} | 1 |
| {Bread, Coffee} | {T100, T800} | 2 |
| {Butter, Eggs} | {T300, T600, T800, T900} | 4 |
| {Butter, Sugar} | {T200, T400} | 2 |
| {Butter, Coffee} | {T100, T800} | 2 |
| {Eggs, Coffee} | {T800} | 1 |

A given 3-itemset is a candidate 3-itemset only if every one of its 2-itemsets subsets is frequent. The candidate generation process here will generate only two 3-itemsets: {Bread, Butter, Eggs} and {Bread, Butter, Coffee}. By intersecting the TID-sets of any two corresponding 2-itemsets of these candidate 3-itemsets as shown in Table 4, where there are only two frequent 3-itemsets: {Bread, Butter, Eggs: 2}and {Bread, Butter, Coffee:2}.

**Table 4.** The 3-itemsets in Vertical Data Format for Food Departmental Store

| Itemsets | TID-Set | Support-Count |
|---|---|---|
| {Bread, Butter, Eggs} | {T800, T900} | 2 |
| {Bread, Butter, Coffee} | {T100, T800} | 2 |

Suppose the data contain frequent itemset L={Bread, Butter, Coffee:2}.The nonempty subsets of L are {Bread, Butter} and {Bread, Coffee}, {Butter, Coffee}, {Bread}, {Butter} and {Coffee}.The association rules that can be generated from L are as shown below, each listed with its confidence:

- *Bread and Butter ⇒ Coffee ,confidence = 2/4 = 50%*
- *Bread and Coffee ⇒ Butter ,confidence = 2/2 = 100%*
- *Butter and Coffee ⇒ Bread ,confidence = 2/2 = 100%*
- *Bread ⇒ Butter and Coffee ,confidence = 2/6 = 33%*
- *Butter ⇒ Bread and Coffee ,confidence = 2/7 = 29%*
- *Coffee ⇒ Bread and Butter , confidence = 2/2 = 100%*

If the minimum confidence is 70%, then only the second, third and last rule above are output, because these are the only ones generated that are strong.

According to the output of second rule, when an item, such as bread, has been designated to go on sale, management determines what other items are frequently purchased with bread. We find that coffee is purchased with bread 50% of the time and that butter are purchased with it 25% at the time. Based on the associations, special displays of butter and coffee are placed near the bread which is on sale. These actions are aimed at increasing overall sales volume by taking advantage of the frequency with which these items are purchased together. So, the food departmental stores are promoted sales and developed.

## 7. Conclusion

Data mining refers to extracting or mining knowledge from large amounts of data. Association rule mining is a process that identifies links between sets of correlated objects in transactional databases where each transaction contains a list of items. Determining frequent objects is one of the most important fields in data mining. In this paper we present an implementation that solved association rule and frequent itemset mining problem.

This system is implemented Association Rule Approach with ECLAT method. ECLAT is the fast algorithm which is used for mining of frequent patterns from the transaction database. ECLAT algorithms have been shown to perform much better and have far less costly candidate generation phases and do not required scanning the complete database over and over again. Our proposed system is performed on the retail data of customer's transaction of the store and then the important decisions are made by applying the strong association rules. So, it also saves the time and provides the display of items in a minimum space.

## References

[1] R.Agrawal, T.Imielinski, and A.Sawmi. "Mining association rules between sets of items in large databases", in proceeding of the ACM SIGMOD Conference on Management of Data, pp. 207-216, 1993.

[2] T.Calders, and B.Goethals, "Depth-First Non-Derivable Itemset Mining", University of Antwerp, Belgium, 1997.

[3] H.M.Dunham, and Y.Xiao, "A Survey of Association Rules", Department of Computer Science and Engineering, Southern Methodist University.

[4] J.Han and M.Kamber, Data Mining: Concepts and Techniques, second edition, Academic Press, India, 2006.

[5] J.Hipp, U.Guntzer, and N.Nakhaeizadch, "Algorithms for Association Rule Mining- a General Survey and Comparison", ACM SIGKDD Explorations 2, pp.58-65, July 2000.

[6] M.H.Margahny, and A.A.Mitwaly, "Fast Algorithm for Mining Association Rules", AIML 05 Conference, CICC, Cairo, Egype, 19-21 December 2005.

[7] J.Vaidya, and C.Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data", Department of Computer Sciences, Purdure University, West Lafayette, Indiana, 2002.