ECLAT-Based Association Rules Mining for Education Training Centre

Hsu Hmwaye Aung, Khin Mar Myo
University of Computer Studies, Mawlamyine, Myanmar
hsuhmwaye@gmail.com, kmmyo09@gmail.com

Abstract

Frequent pattern mining has become an important data mining task because finding such frequent patterns play an essential role in mining associations, correlation and many other interesting relationships among data. This paper describes how effectively use the vertical association rule mining approach in finding correlated courses in Human Resource Centre. Although there are many algorithms for vertical approach, ECLAT (Equivalence CLASS Transformation) algorithm, developed by Zaki [6], is used for our system implementation. By using ECLAT method, useful frequent itemsets (courses) are obtained easily by saving the database scanning time. These frequent itemsets are effectively used in providing correlated information to the system users.

1. Introduction

Association rule mining has attracted the attention of data mining research community since the early 90s, as a mean of unsupervised, exploratory data analysis. The association rule mining paradigm involves searching for co-occurrences of items in transaction databases. Such a co-occurrence may imply a relationship among the items it associates. These relationship can be further analyzed and may reveal temporal or causal relationships, behaviors etc. An example of association rule might be "85% of students that attend English Grammar Basic Course also attend Grammar Advanced Course."[1].

Association rules are applied in many domains that range from decision support to telecommunications alarm diagnosis and prediction. However, the typical application of association rules is in analysis of sales data referred to as market basket data. Other applications of associations rules include cross marketing and attached mailing applications, catalog design, add-on sales, store layout and customer segmentation based on buying patterns [1].

Association rule mining, as originally proposed in with its Apriori algorithm, has developed into an active research area. Many additional algorithms have been proposed for association rule mining. Among these algorithms, ECLAT algorithm is evolves as a vertical association rule mining approach. This algorithm takes the advantages of Apriori property, besides these it can reduce the scanning time for database [2].

The main aim of our work is to apply the facilities of ECLAT algorithm in specific domain area. The Human Resource Training Centre is chosen as domain area in our system design and implementation. Firstly, frequent itemsets (frequent courses) are generated by using ECLAT method. Secondly, these itemsets are later used for rule generation. Finally, these rules are used to provide useful and recommended information for user as correlated courses.

This paper organized as follow. Section 2 describes the related work. Review on ECLAT method is described in Section 3. Section 4 describes the overview of system design. Detail explanation on system implementation is described in Section 5. Performance of the system is describes in Section 6. In Section 7, we describe the conclusion of our system.

2. Related work

The related works concerning with our system is summarized in this section. Some of papers describe how effectively use association rule mining approach in their domain area. Most of the papers pointed-out the problems of association rule mining for mining large itemsets and for generating association rules. There has been a lot of research in developing efficient algorithms for mining frequent itemset [3]. The paper [3] presented about Lattice-Based Algorithm for Incremental mining of Association rules. The history of rule mining algorithm is summarized in paper [4]. According to [4], one of the first algorithms for Association Rule Mining was the AIS algorithm. This algorithm was improved later to obtain the Apriori algorithm. Many variants of the Apriori algorithm have been also developed, such as AprioriTid, AprioriHybrid, direct hashing and pruning (DHP), dynamic itemset counting (DIC), partition algorithm etc. FP-growth is a well-known algorithm that uses the FP tree data structure to achieve a condensed representation of the database transaction.

According to [5], a number of vertical mining algorithms have been proposed recently for association rule mining, which has shown to be very effective and usually outperform horizontal approaches. ECLAT is the first algorithm to find frequent patterns by depth-first search and it has shown to perform well.

3. ECLAT Algorithm

A number of vertical mining algorithms have been proposed recently for association mining, which has shown to be very effective and usually outperform horizontal approaches. The main advantages of the vertical format is support for fast frequency counting via intersection operations on transaction ids (tids) and automatic pruning of irrelevant data. Among these vertical approaches, ECLAT algorithm is the first successful algorithm proposed to generate all frequent itemsets in a depth-first manner. This algorithm first transforms the given set of transactions in horizontal data format into vertical data format. If the itemset I dose not satisfy the minimum support threshold, min-sup, then I is not frequent; that is, P (I) <min-sup. If an item A is added to the itemset I, then the resulting itemset (i.e. I U A) cannot occur more frequently than I. Therefore, I U A is not frequent either; that is P (I U A) <min-sup. ECLAT algorithm makes the intersection of TID set in the join step and automatic pruning of irrelevant data.

The main problem with these approaches is when intermediate results of vertical tid lists become too large for memory, thus affecting the algorithm scalability.

3.1 Steps of ECLAT Algorithm

The processing step of ECLAT algorithm as follows. This algorithm takes transactions and support count as input. Firstly, transforms the database into its vertical format. Instead of explicitly listing all transactions, each item is stored together with its cover (also called *tidlist*).

Input: D, σ, I (*I* initially called with I= { })
Output: F [I]
D= all transaction database

σ = minimum support count

I= itemset

I= item

F [I] =frequent itemset

```
Input: D, σ, I (I initially called with I= { })
Output: F [I]
1: F [I]:= { };
2: For all: Є I occurring in D do
3: Add I U {i} to F [I]:
4: D: = { };
5: For all j Є I occurring in D such that j > I do
6: C: = cover ({i}) U cover ({j}):
7: if I c I ≥ σ then
8: Add (j, c) to D;
9: Compute F [I U {i}] (D, σ) recursively;
10: Add F [I U {i}] to F [I];
```

On line 3, each frequent item is added in the output set. After that, on lines 4-8 for every such frequent item I, the i-projected database is created. This is done by first finding every item j that frequently occurs together with i. The support of this set {i, j} is computed by intersecting the covers of both items (line 6). If {i, j} is frequent, then j is inserted into together with its cover (line 7, 8). On line 9, the algorithm is called recursively to find all frequent itemsets in the new database. A candidate itemset is represented by each set I {i, j} of which the support is computed at line 7 of the algorithm.

4. System Design

The system design is constructed based on the vertical approach. The system design consists of four main components. They are transaction data collector, frequent itemset miner, association rule generator and information provider.

4.1 Transaction data collector

This component collects the transaction data from the database. This component only focuses and extract on transaction field (tid) and itemsets (k).

4.2 Frequent Itemset Miner

Many algorithms are used for frequent itemset mining. In this paper, ECLAT algorithm based on

Apriori Property is used in frequent itemset mining. This algorithm takes transactions and support count as input. Firstly, the transaction data in horizontal data format (tids, itemsets) from the data base is transformed into vertical data format (items, tid-sets). After transforming the transactions, frequent itemset calculator calculates by intersecting the transactions with vertical data format and generating frequent itemset that are satisfied minimum support count.

4.3 Association Rule Generator

This component is responsible to generate the association rules based on the minimum support and minimum confidence. The number of rules may be differing based on the confidence.

4.4 Information Provider

The rules generated by above components are used to provide correlated patterns as suggestion to users.

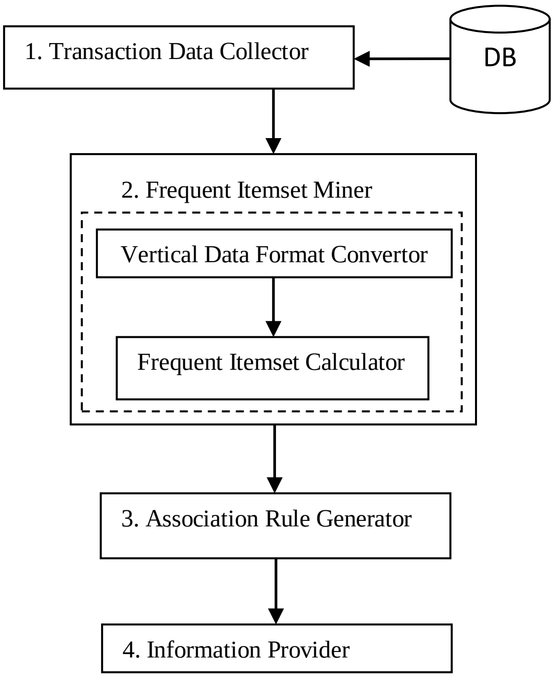
1. Transaction Data Collector
DB
2. Frequent Itemset Miner
Vertical Data Format Convertor
Frequent Itemset Calculator
3. Association Rule Generator
4. Information Provider

Figure 1. Overview of system

5. System Implementation

To measure the effectiveness and efficiency of proposed system design describe in Section 3, experiment is implemented. The dataset getting from Human Resources Centre are chosen as domain area for system implementation. There are two main users in this domain. The trainees and the owner and manager from training centre. Trainees usually want to know which courses are popular and which course should be attend. At that time manager want to know which courses are popular for administrative purposes. Our system provides correlated courses not only to trainees but also to manager by using the rules generating by ECLAT algorithm.

The original transactional data getting from Training Centre is illustrated in Table 1.

Table 1.Transaction Table (horizontal format)

TID	Itemsets
T1	OOP with C^{++}, .Net Framework with C#, ASP.Net with C#
T2	LCCI Level1&2, LCCI Level 3
T3	OOP with Core JAVA, J2SE, J2EE
T4	.Net Framework with C#, ASP.Net with C#
T5	LCCI Level1&2
T6	OOP with Core JAVA, J2SE, J2EE
T7	OOP with Core JAVA
T8	OOP with C^{++}, .Net Framework with C#, ASP.Net with C#
T9	J2SE, J2EE

5.1 Patterns Mining Using ECLAT

As describes in section 3.1, the first thing to do for ECLAT algorithm is to convert the dataset into vertical format. Therefore the table 1 is changed into vertical format as illustrated in Table 2.

Table 2.Transaction Table (vertical format)

Items	TID-set
OOP with C^{++}	{T1,T8}
OOP with Core JAVA	{T3,T6,T7}
.Net framework with C#	{T1,T4,T8}
ASP.NET with C#	{T1,T4,T8}
Advanced JAVA(J2SE)	{T3,T6,T9}
Web JAVA (J2EE)	{T3,T6,T9}
LCCI Level 1&2	{T2,T5}
LCCI Level 3	{T2}

The support count of an itemset is the same as the length of the TID-sets 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. By intersection of the TID-sets of the k-itemsets are used to mine the (k+1)-itemsets (2-itemsets). When the itemset are mined by using ECLAT algorithm, it is carried out by scanning data set once as shown in the following Table 3.

Table 3 Mining 2-itemsets using ECLAT

Items	TID-sets
OOP with C^{++}, .Net Framework with C#	{T1,T8}
OOP with C^{++}, ASP.Net with C#	{T1,T8}
OOP with Core JAVA, J2SE	{T3,T6}
OOP with Core JAVA, J2EE	{T3,T6}
.Net framework with C#, ASP.Net	{T1,T4,T8}
Advance JAVA(J2SE),Enterprise JAVA(J2EE)	{T3,T6,T9}

The results of 2-itemsets are compared with the minimum support count which is 2. For mining 3-itemsets, mining can be performed on 2-itemsets by intersecting the TID-sets of every pair of frequent single items. In order to mine next frequent itemsets, their corresponding previous itemsets are used and mined frequent itemsets based on Apriori property. This process repeats, with k incremented by 1 each time, until no frequent itemsets or no candidate itemsets can be found. Frequent 3-itemsets are mined from frequent 2-itemsets is as shown in below Table4.

Table 4. Mining 3-itemsets from 2-itemsets

items	TID-set
OOP with C^{++}, .Net Framework with C#, ASP.Net with C#	{T1,T8}
OOP with Core JAVA, J2SE, J2EE	{T3,T6}

5.2. Generating Association Rule

In this paper, the following equation is used for association rule.

Support = P (A, B) = probability that a transaction contains $A \cup B$

Support $(A \Rightarrow B) = P(A \cup B)$

Confidence = P (B|A) = conditional probability that a transaction having A also contains B =P(A, B)/P(A).

$$\text{Confidence } (A \Rightarrow B) = P(B|A) = \frac{\text{support}(AUB)}{\text{support}(A)}$$

$$= \frac{\text{support count }(AU\,B)}{\text{support count}(A)}$$

In this example,

Let {OOP with C^{++}, .Net Framework with C#, ASP.Net with C#} = {I1, I3, I4} and {OOP with Core JAVA, J2SE, J2EE} = {I2, I5, I6}.

l= {I1, I3, I4}, {I2, I5, I6}. The non empty subsets of ***l*** are {I1, I3}, {I1, I4}, {I3, I4}, {I2, I5}, {I2, I6}, {I5, I6}, {I1}, {I3}, {I4}, {I2}, {I5}, {I6}.

The resulting association rules for {I1, I3, I4} and {I2, I5, I6} are as shown below,

1. $I1 \wedge I3 \Rightarrow I4$, confidence =2/2=100%
2. $I1 \wedge I4 \Rightarrow I3$, confidence =2/2=100%
3. $I3 \wedge I4 \Rightarrow I1$, confidence =2/3=67%
4. $I1 \Rightarrow I3 \wedge I4$, confidence =2/2=100%
5. $I3 \Rightarrow I1 \wedge I4$, confidence =2/3=67%
6. $I4 \Rightarrow I1 \wedge I3$, confidence =2/2=67%
7. $I2 \wedge I5 \Rightarrow I6$, confidence =2/2=100%
8. $I2 \wedge I6 \Rightarrow I5$, confidence =2/2=100%
9. $I5 \wedge I6 \Rightarrow I2$, confidence =2/3=67%
10. $I2 \Rightarrow I5 \wedge I6$, confidence =2/3=67%
11. $I5 \Rightarrow I2 \wedge I6$, confidence =2/3=67%
12. $I6 \Rightarrow I2 \wedge I5$, confidence =2/3=67%

To generate the association rule, required threshold values (minimum confidence) must be inputted by user. If minimum confidence threshold is, say, 80%, then only the rule 1,2,4,7 and 8 are output.

6. Performance of the System

For the testing purpose, we use a test dataset for mining the frequent itemset. It includes 3500 transactions and 200 items. We mine the dataset with variation of support and confidence, such as 2% of support and 40% of confidence and 3% of support and 60% of confidence using Apriori and ECLAT algorithm. Results show that the total time consuming of ECLAT is always less than Apriori. The following table shows the result of the test comparing on two algorithms.

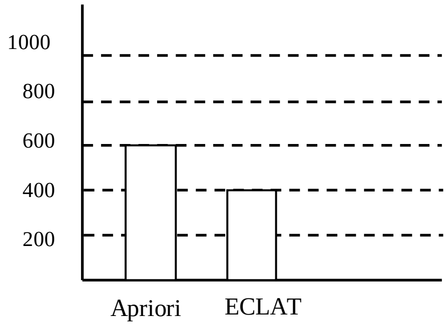

Figure2. Time consuming for two algorithms

7. Conclusion

Discovering the association rules in transaction databases are very useful tool for finding the relation of itemsets. This system analyzed the association of each course in training centre. This paper describes the detail implementation of association rule mining for Training Centre. The frequent courses which are attended by students can be easily extracted by ECLAT algorithm. By using this algorithm, the system is needed to scan the database only once time. The discovery of such association can help training centre to develop marketing strategies by gaining insight into which courses are frequently attended by students. By using this system, the trainee can also get the suggestion of which courses should be attended.

References

[1] C. Berberidis, G. Tzanis and I. Vlahavas. ”Mining for Contiguous Frequent Itemsets in Transaction Databases.” *IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 5-7 September 2005, Sofia, Bulgaria.*

[2] G. K. Palshikar, M. S. Kale and M. M. Apte. “*Association Rules Mining Using Heavy Itemsets”.* ADVANCES IN DATA MANAGEMENT 2005 Jayant Haritsa, T.M. Vihayaraman (Editors) © CSI 2005.

[3] R. Patel, D.K. Swami and K. R. Pardasani. *“Lattice Based Algorithm for Incremental Mining of Association Rules”*, International Journal of Theoretical and Applied Computer Sciences, Vol.1, No.1 (2006) pp. 119-128.

[4] M. Song and S. Rajasekaran “*A Transaction Mapping Algorithm for Frequent Itemsets Mining*”, IEEE Transactions on Knowledge and Data Engineering.

[5] M.J. Zaki and K. Gouda. *“Fast Vertical Mining Using Diffsets”*, Technical report, RPI, 2001. Tech. Report. 01-1.

[6] M.J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowledge and Data Engineering*, 12:372-390, 2000.

[7] A. P.Kyaing, K.A.Than. “Mining Frequent Itemsets by Using EClaT (Equivalence Class Transformation) method”, Master of Computer Science Thesis, 2008, University of Computer Studies, Yangon.