Topic :

Association Rule Mining:

Description:

Association:

An association indicates a logical dependency between various things. Association rule mining means to derive all logical dependencies among different attributes given a set of records.

The association rule mining problem was first developed by Agarwal, Imielinski and Swami, "Mining association rules between sets of items in large databases", Proc. Of Inlt. Conf. on Management of Data, 1993] and is often referred to as the Market-Basket Analysis (MBA) problem.

Market-Basket Analysis: The concept of Market Basket Analysis was framed from the analysis of supermarket. Here, given a set of transactions of items the task is to find relationships between the presences of various items. In other words, the problem is to analyze customer's buying habits by finding associations between the different items that customers place in their shopping baskets. Hence, it is called Market-Basket Analysis. The discovery of such association rules can help the super market owner to develop marketing strategies by gaining insight into matters like "which items are most frequently purchased by customers". It also helps in inventory management, sale promotion strategies, supply-chain management, etc.

Suppose a group of customers purchase data are collected from a grocery store. The table given below shows a small sample of data. From this data, the store owner is interested to learn about the purchasing behaviour of the customers. That is, which items are occurring more frequent?

Basket	Items
1	bread, milk, diaper, cola
2	bread, diaper, beer, egg

3	milk, diaper, beer, cola
4	bread, milk, tea
5	bread, milk, diaper, beer
6	milk, tea, sugar, diaper

Table 1. Sample Customer purchase data

From the table 1, following are the items frequently bought.
$Bread–milk$
$Diaper–beer$

We can say that two rules
$\{bread\}\rightarrow\{milk\}$ [If customer buys bread then he also buys milk] and
$\{beer\}\rightarrow\{diaper\}$[If customer buys beer then he also buys diaper], or more precisely, two association rules suggesting a strong relationships between the sales of bread and milk, and beer, diaper exist (this means, customer, who purchases bread (or beer) also likely to purchase milk (or diaper).

The process of analyzing customer buying habits by finding association between different items can be extended to many application domains like medical diagnosis, web mining, text mining, bioinformatics, scientific data analysis, insurance schemas, etc. In general, an association rule can be expressed as

$$\{S_L\} \rightarrow \{S_R\}$$

Where S_L and S_R denotes a set of items (non-empty). However, there is no universally accepted notation for expression association rules and we may merely follow the popular convention. Further, note that in our convention an arrow is used to specify whether the relation is bi-directional. Sometimes, the relationship may be unidirectional, that is, $\{S_R\} \rightarrow \{S_L\}$ are both equivalents.

Classification rule of the form $X \rightarrow Y$ and association rule of the same form

$X \rightarrow Y$ look alike but they have different implications.

1. First, classification rules are concerned with predicting an entity to which it belongs. So, the right-hand part of the rule is always a single attribute called class label. Whereas $X \rightarrow Y$ representing an association rule, Y may be consists of one or more element(s).
2. Both the rules imply logical dependence of Y on X. In case of classification rule, X is a conjunctions of attributes and relations with their values. In other words, the left-hand part in classification rule potentially includes test on the value of any attribute or combination of attribute. On the other hand, $X(or\ Y)$ in association rule includes a set of values rather than tests.
3. Classification rule $\text{IF } X \rightarrow \text{THEN } Y$ is either fires (that is, satisfies) or not; whereas, in case of association rule $X \rightarrow Y$ it is a matter of degree of strength how X is related to Y.
4. In the case of classification rules, we are generally interested in the quality of a rule set as a whole. It is all the rules working in combination that determine the effectiveness of a classifier, not any individual rule or rules.

In the case of association rules, the emphasis is on the quality of each individual rule, instead of the whole set of rules.

Notation	Description
D	$Database\ of\ transactions$
t_i	$Transaction\ (ith\ )\ inD$
$X,\ Y$	$Itemsets$

$X \rightarrow Y$	$Association\ rule$
s	$Support$
α	$Confidence$
I	$Set\ of\ large\ itemsets$
i	$Large\ itemset\ in\ I$
C	$Set\ of\ candidate\ itemsets$

Table 2. Some notations commonly used in Association rule mining.

Transactions and itemsets

Suppose, D is a database comprising n transactions that is, D = $\{t_1, t_2, .., t_n\}$ where each transaction denotes a record (for example, in the context of market-basket, it is a set of items shopped). I denotes the set of all possible items. For an example, Table 3 shows a database of transaction. In this table, $a, b, c, d\ and\ e$ denote items. Here, $I = \{a, b, c, d, e\}$ comprising set of all items. Any one transaction, say $\{a, b, c\}$ is called an itemset.

Transaction Id	**Transaction (item set)**
1	$\{a, b, c\}$
2	$\{a, b, c, d, e\}$
3	$\{b\}$
4	$\{c, d, e\}$
5	$\{c, d\}$
6	$\{b, c, d\}$
7	$\{c, d, e\}$

8	$\{c, e\}$

Table 3. Transaction dataset

Note: $\{a, b, c\}$ and $\{c, b, a\}$ are the same itemset. For convenience, all items in an itemset are presented in an order (say alphabetical order). An itemset is a non-null set. Thus, the maximum number of transactions, that is possible, with m items in I, is $2^m - 1$ (it is all subsets from m elements).

Associations rule:

- Usually, a rule is stated as $r_i: X \rightarrow Y$. Here X is called body and Y is head. Also, $X\ and\ Y$ is called antecedent and consequent.
- Inorder to improve, the interpretability there are many optional parameters are included. Thus, $r_i: X \rightarrow Y$ [support, confidence] [start Time, end Time] [validity Period] etc. We shall discuss later about these parameters.
- The head part is usually a single attribute. If there are more than one attribute in head, then it can be splitted into further association rules. For example

$$r_i: \{a, b\} \rightarrow \{c, d\} \Rightarrow \begin{cases} \{a, b\} \rightarrow \{c\}\ and \\ \{a, b\} \rightarrow \{d\}\ etc. \end{cases}$$

A transaction t_i is said to contain an itemset$X$, if X is a subset of t_i. **Support count** refers to the number of transactions that contain a particular itemset.

Example: With reference to Table 3, the support count of the set $\{c, d\}$ is $= 5$

Support is the ratio (or percentage) of the number of itemsets satisfying both body and head to the total number of transactions.

Support of a rule $r_i: X \rightarrow Y$ is denoted as $s(r_i)$ and mathematically defined as $s(r_i) = \frac{\sigma(\mathrm{X} \cup Y)}{|D|}$ ……(19.2)

Example: From Table 19.3:

$$s(\{c, d\} \rightarrow \{e\}) = \frac{3}{8} = 0.375$$

$$s(\{a, b,\ c\} \rightarrow \{d, e\}) = \frac{1}{8} = 0.125$$

Confidence of a rule $r_i: X \rightarrow Y$ in a database D is represented by $\alpha(r_i)$ and defined as the ratio (or percentage) of the transactions in D containing X that also contain Y to the support count of X. More precisely,

$$\alpha(X \rightarrow Y) = \frac{\sigma(\mathrm{X} \cup Y)}{\sigma(X)}$$

So, alternatively, confidence of a rule $X \rightarrow Y$ is the conditional probability that Y occurs given that X occurs.

Support (s) is also called "relative support" and confidence (α) is called "absolute support" or "reliability".

It is customary to reject any rule for which the support is below a minimum threshold (μ). This minimum threshold of support is called minsup and denoted as μ.

Typically the value of $\mu = 0.01\ (i.e.,\ 1\%)$. Also, if the confidence of a rule is below a minimum threshold (τ), it is customary to reject the rule. This minimum threshold of confidence is called minconf and denoted as τ.

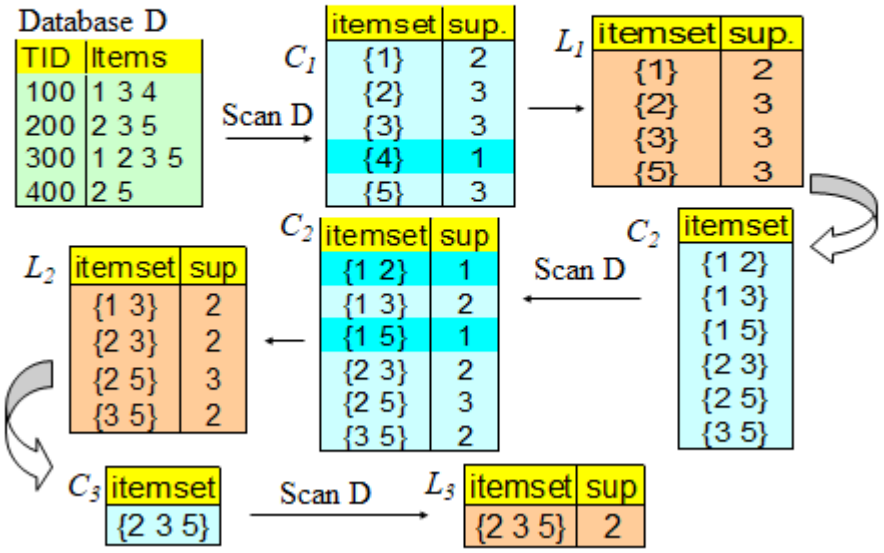
Frequent Itemset

Let μ be the user specified minsup. An itemset i in D is said to be a frequent itemset in D with respect to μ if $s(i) \geq \mu$

- Join Step: C_k is generated by joining L_{k-1} with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code:

C_k: Candidate itemset of size k
L_k : frequent itemset of size k
L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
C_{k+1} = candidates generated from L_k;
for each transaction t in database do
increment the count of all candidates in C_{k+1} that are contained in t
L_{k+1} = candidates in C_{k+1} with min_support
end
return $\cup_k L_k$;

The Apriori Algorithm — Example



Example of Generating Candidates

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4={*abcd*}

Examples: