Negative Association Rules in Data Mining

Olena Daly
Monash University, Australia

David Taniar
Monash University, Australia

INTRODUCTION

Data Mining is a process of discovering new, unexpected, valuable patterns from existing databases (Chen, Han & Yu, 1996; Fayyad et. al., 1996; Frawley, Piatetsky-Shapiro & Matheus, 1991; Savasere, Omiecinski & Navathe, 1995). Though data mining is the evolution of a field with a long history, the term itself was introduced only relatively recently in the 1990s. Data mining is best described as the union of historical and recent developments in statistics, artificial intelligence, and machine learning. These techniques then are used together to study data and find previously hidden trends or patterns within.

Data mining is finding increasing acceptance in science and business areas that need to analyze large amounts of data to discover trends that they could not otherwise find. Different applications may require different data mining techniques. The kinds of knowledge that could be discovered from a database are categorized into association rules mining, sequential patterns mining, classification, and clustering (Chen, Han & Yu, 1996).

In this article, we concentrate on association rules mining and, particularly, on negative association rules.

BACKGROUND

Association rules discover database entries associated with each other in some way (Agrawal, Imielinski & Swami, 1993; Agrawal & Srikant, 1994; Agrawal, et. al., 1996; Mannila, Toivonen & Verkamo, 1994; Piatetsky-Shapiro, 1991; Srikant & Agrawal, 1995). The example could be supermarket items purchased in the same transaction, weather conditions occurring on the same day, stock market movements within the same trade day, words that tend to appear in the same sentence. In association rules mining, we search for sequences of associated entities, where each subsequence forms association rules as well. Association rule is an implication of the form $A \Rightarrow B$, where A and B are database itemsets. A and B belong to the same database transaction. The discovered sequences and later on association rules allow to study customers' purchase patterns, stock market movements, and so forth.

There are two measures to evaluate association rules: support and confidence (Agrawal, Imielinski & Swami, 1993). The rule $A \Rightarrow B$ has support s, if $s\%$ of all transactions contains both A and B. The rule $A \Rightarrow B$ has confidence c, if $c\%$ of transactions that contains A also contains B. Association rules have to satisfy the user-specified minimum support (minsup) and minimum confidence (minconf). Such rules with high support and confidence are referred to as *strong rules* (Agrawal, Imielinski & Swami, 1993).

The generation of the strong association rules is decomposed into the following two steps (Agrawal, Imielinski & Swami, 1993; Agrawal & Srikant, 1994): (i) discover the frequent itemsets; and (ii) use the frequent itemsets to generate the association rules.

Itemset is a set of database items. Itemsets that have support at least equal to minsup are called *frequent itemsets*. *1-itemset* is called an itemset with one item (A), *2-itemset* is called an itemset with two items (AB), *k-itemset* is called an itemset with k items. The output of the first step is all itemsets in the database that have their support at least equal to the minimum support.

In the second step, all possible association rules will be generated from each frequent itemset, and the confidence of the possible rules will be calculated. If the confidence is at least equal to minimum confidence, the discovered rule will be listed in the output of the algorithm.

Consider an example database in Figure 1. Let the minimum support be 40%, the minimum confidence 70%; the database contains 5 records and 4 different items *A, B, C, D* (see Figure 1). In a database with five records, support 40% means two records. For an item (itemset) to be frequent in the sample database, it has to occur in two or more records.

Figure 1. A sample database

1	A,D
2	B,C,D
3	A,B,C
4	A,B,C
5	A,B

Discover the Frequent Itemsets

- **1-itemsets:** The support of each database item is calculated

Support(A)= 4 (records)
Support(B)= 4 (records)
Support(C)= 3 (records)
Support(D)= 2 (records)

All items occur in two or more records, so all of them are frequent. Frequent 1-itemsets: *A, B, C,* and *D*

- **2-itemsets:** From frequent 1-itemsets candidate 2-itemsets are generated: AB, AC, AD, BC, BD, CD. To calculate the support of the candidate 2-itemsets, the database is scanned again.

Support(AB) =3 (records)
Support(AC) =2 (records)
Support(AD) =1 (record)
Support(BC) =3 (records)
Support(BD) =1 (record)
Support(CD) =1 (record)

Only itemsets occurring in two or more records are frequent. Frequent 2-itemsets: *AB, AC,* and *BC*

- **3-itemsets:** From frequent 2-itemsets candidate 3-itemsets are generated: ABC. In step 3 for candidate 3-itemsets, it is allowed to join frequent 2-itemsets that only differ in the last item. AB and AC from the step 2 differ only in the last item, so they are joined and the candidate 3-itemset ABC is obtained. AB and BC differ in the first item so a candidate itemset cannot be produced here; AC and BC do not differ in the last item, no candidate itemset produced. So there is only one candidate 3-itemset ABC. To calculate the support of the candidate 3-itemsets the database is scanned again.

Support(ABC) =2 (records)
Frequent 3-itemsets: ABC

- **4-itemsets:** In step 3, there is only one frequent itemset, so it is impossible to produce any candidate 4-itemsets. The frequent itemsets generation stops here. If there was more than one frequent 3-itemset, then step 4 (and onwards) would be similar to step 3.

The list of frequent itemsets: A, B, C, D, AB, AC, BC, and ABC.

Use the Frequent Itemsets to Generate the Association Rules

To produce an association rule, at least frequent 2-itemset. So the generation process starts with 2-itemsets and goes on with longer itemsets. Every possible rule is produced from each frequent itemset:

AB:	$A \Rightarrow B$
	$B \Rightarrow A$
AC:	$A \Rightarrow C$
	$C \Rightarrow A$
BC:	$B \Rightarrow C$
	$C \Rightarrow B$
ABC:	$A \Rightarrow BC$
	$B \Rightarrow AC$
	$C \Rightarrow AB$
	$BC \Rightarrow A$
	$AC \Rightarrow B$
	$AB \Rightarrow C$

To distinguish the strong association rules among all possible rules, the confidence of each possible rule will be calculated. All support values have been obtained in the first step of the algorithm.

Confidence($A \Rightarrow B$)=Support(AB)/Support(A)
Confidence($AB \Rightarrow C$)=Support(ABC)/Support(AB)

Association rules have been one of the most developed areas in data mining. Most of research has been done in positive implications, which is when occurrence of an item implies the occurrence of another item. Negative rules also consider negative implications, when occurrence of an item implies absence of another item.

MAIN THRUST

Negative itemsets are itemsets that contain both items and their negations (e.g., $AB{\sim}C{\sim}DE{\sim}F$). $\sim C$ means negation of the item C (absence of the item C in the database record).

Negative association rules are rules of a kind $A \Rightarrow B$, where A and B are frequent negative itemsets. Negative association rules consider both presence and absence of items in the database record and mine for negative implications between database items.

Examples of negative association rules could be Meat$\Rightarrow$~Fish, which implies that when customers purchase meat at the supermarket, they do not buy fish at the same time, or ~Sunny$\Rightarrow$Windy, which means no sunshine