Mining Association Rules in Long Sequence

Teerath Prasad Vishwkarma*
Jitendra Kumar Mishra**

Abstract

Discovering interesting patterns in long sequences, and finding confident association rules within them, is a popular area in data mining. Most existing methods define patterns as interesting if they occur frequently enough in a sufficiently cohesive form. Based on these frequent patterns, association rules are mined in the traditional manner. Recently, a new interestingness measure, combining cohesion and frequency of a pattern, has been proposed, and patterns are deemed interesting if encountering one event from the pattern implies with a high probability that the rest of the pattern can be found nearby. It is quite clear that this probability is not necessarily equally high for all the events making up such a pattern, which is why we propose to introduce the concept of association rules into this problem setting. The confidence of such an association rule tells us how far on average from a particular event, or a set of events, one has to look, in order to find the rest of the pattern. In this paper, we present an efficient algorithm to mine such association rules. After applying our method to both synthetic and real-life data, we conclude that it indeed gives intuitive results in a number of applications.

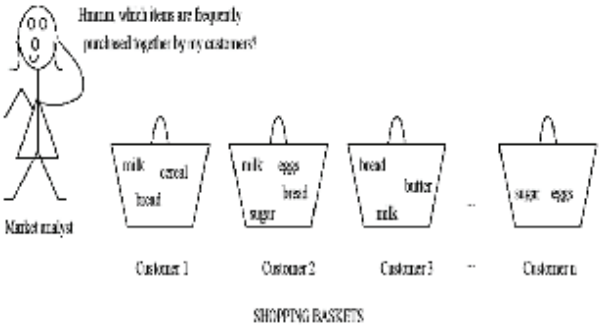
Introduction

Data mining, also referred to as database mining or knowledge discovery in databases (KDD), is a new research area that aims at discovery of useful information from large datasets. Data mining uses statistical analysis and inference to extract interesting trends and events, create useful reports, support decision making etc. It exploits the massive amounts of data to achieve business, operational or scientific goals. An important goal of current research is to provide methods for on-line analytical mining (OLAM) [6]. On-line analytical mining implies that data mining is performed in a way similar to on-line analytical processing (OLAP), i.e. mining can be performed interactively, for different portions of a database and at different conceptual levels.On-line analytical mining requires a high-performance and rapid-response environment that assists users in data selection, rule generation and rule filtering [5],[8], [11].

Area of Application

Market Basket Analysis

Understanding customer's buying habits and preferences is essential for retailers to make decisions including what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc, Association rules mining can provide such information An effective mining



application in the retail environment is market basket analysis or shopping basket analysis.

It analyzes the attributes of customer's shopping basket from Electronic Point of Sale data and applies the findings to launch effective promotions and advertising. For example, all rules that have "Diet Coke" as consequent may help plan what the store should do to boost the sales of "Diet Coke".

Association Rule

As we mentioned above, mining association rules is to find interesting association or correlation relationships among a large set of data. i.e. to identify sets of attribute values (predicate or item) that frequently occur together, and then formulate rules that characterize these relationships. A formal definition is given below.

Definition: An association rule is a rule in the form of

$A1,A2,\ldots\ldots Am \Rightarrow B1,B2,\ldots\ldots Bn$

*Student, Pioneer Institute of Professional Studies, Indore
**Student, Pioneer Institute of Professional Studies, Indore

where Ai and Bj are predicates or items.Such rules are usually interpreted as ***"When items A1,A2,……Am occur, it is often the case that items B1,B2,…….Bn occur as well in the same transaction***"

What exactly constitutes an item or a transaction depends on the application.

Let L -{*I1,I2,……..Im*} be a set of literals, called items. Let a non empty set of items T be called an *itemset*. Let D be a set of variable length itemsets, whee each itemset $T \subset L$.We say that an itemset Tsupports an item x⊃ L if x in T. We say that an itemset T supports an itemset X⊂ L if T supports every item in the set X.

An association rule is an implication of the form X⇒Y, where X⊂ *L, Y*⊂ *L,.X* ∩ *Y*=ϕ Each rules has associated measures of its statistical significance and strength, called support and confidence. The support of the rule X⇒Y in the set D is:

$$\text{support (XY,D)} = \frac{|\{T \in D \mid T \text{ sup } ports\_XUY\}|}{|D|}$$

In other words, the rule X⇒Y holds in the set D with support s if s% of itemset in D support XUY. The confident of the rule X⇒Y in the set D is :

$$\text{confident(X⇒Y,D)} = \frac{|\{T \in D \mid T \text{ sup } ports\_XUY\}|}{|\{T \in D \mid T \text{ sup } port\_X\}|}$$

In other words, the rule X⇒Y has confidence c if c% of itemsets in D that support X also support Y.

For Example

Let I = {i1, i2, . . . , in} be a set of n binary attributes called items. Let D = {t1, t2, . . . , tm} be a set of transactions called the database. Each transaction in D has an unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X => Y where X, Y ⊂ I and X \ Y = ϕ. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule.

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I = {milk, bread, butter, beer} and a small database containing the items is shown in Table 1. An example rule for the supermarket could be {milk, bread}) {butter} meaning

Table 1: An example supermarket database with five transactions.

transaction ID	Items
1	milk, bread
2	bread, butter
3	Beer
4	milk, bread, butter
5	bread, butter

that if milk and bread is bought, customers also buy butter. To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

The support supp(X) of an itemset X is defined as the roportion of transactions in the data set which contain the itemset. In the example database in Table 1, the itemset {milk, bread} has a support of 2/5 = 0.4 since it occurs in 40% of all transactions (2 out of 5 transactions).

The confidence of a rule is defined conf(X) Y) = supp (X [Y)/supp(X). For example, the rule {milk, bread}) {butter} has a confidence of 0.2/0.4 = 0.5 in the database in Table

Frequent Itemset Mining

The first algorithm developed to mine confident association rules ([Agrawal et al., 1993]) was divided into two phases. In the first phase all frequent item sets are generated.

The second phase is made up of the generation of all frequent and confident association rules. Many of the subsequent association rule mining algorithms also comply with this two phased strategy.

For a large number of items, it becomes infeasible to generate all itemsets and determine their support in order to find the frequent ones. That is, for |I| items there are $2^{|I|}$ possible item sets. The naive approach of finding all items quickly becomes intractable. For example, in the very typical case of a thousand items, the number of possible item sets is approximately 10301, which is already larger than the well know googol number (10^{100}) that in its turn is larger than the number of atoms in the observable universe (= 10^{79}). Of course we do not need to consider all possible item sets and can limit ourselves to the itemsets that occur *at*

least once in the transaction databases. Unfortunately for databases containing large transactions the number is mostly still too large. When generating itemsets we would ideally only want to generate the frequent ones. Unfortunately, this ideal solution is impossible in general. We will therefore have to consider several *candidate itemsets* and determine if these are frequent or not. Every considered candidate entails memory usage and computation time in order to obtain the support from the database. The goal is therefore to reduce the amount of candidate itemsets as much as possible in order to obtain an efficient algorithm. One property exploited by most of the itemset mining algorithms is the anti-monotonicity of support with respect to the set inclusion relation:

Important Definitions

As our work is based on an earlier work [2], we now reproduce some of the necessary definitions and notations that we will use here. An event is a pair (i, t), consisting of an item and a time stamp, where i ⌋ I, the set of all possible items, and t ⌋ N. Two items can never occur at the same time. We denote a sequence of events by S. For an itemset X, the set of all occurrences of its items is denoted by ***N(X) = {t | (i, t) ⌋ S and i ⌋ X}.*** The coverage of X is defined as the probability of encountering an item from X in the sequence, and denoted

$$P(X) = \frac{|N(X)|}{|S|}$$

The length of the shortest interval containing itemset X for each time stamp in N(X) is computed as

$$W(X,t) = \min\{t2 - t1 + 1 \,|\, t1 \le t \le t2$$
$$and \forall i \in X, \exists (i,t1) \in S, t1 \le t \le t2\}$$

The average length of such shortest intervals is expressed as

$$\overline{W}(X) = \frac{\sum_{t \in N(X)} W(X,t)}{|N(X)|}$$

The cohesion of X is defined as the ratio of the itemset size and the average length of the shortest intervals that contain it, and denoted

$$C(X) = \frac{|X|}{\overline{W}(X)}$$

Finally, the interestingness of an itemset X is defined as I(X) = C(X)P(X).Given a user defined threshold min int, X is considered interesting if I(X) exceeds min int. An optional parameter, max size, can be used to limit the output only to itemsets with a size smaller or equal to max size.

We are now ready to define the concept of association rules in this setting. The aim is to generate rules of the form if X occurs, Y occurs nearby, where X ⌋ Y=ϕ and X ⌋ Y=ϕ is an interesting itemset. We denote such a rule by X Y , and we call X the body of the rule and Y the head of the rule. Clearly, the closer Y occurs to X on average, the higher the value of the rule. In other words, to compute the confidence of the rule, we must now use the average length of minimal windows containing X U Y, but only from the point of view of items making up itemset X. We therefore define this new average as

$$\overline{W}(X,Y) = \frac{\sum_{t \in N(X)} W(XUY,t)}{|N(X)|}$$

The confidence of a rule can now be defined as

$$c(X \Rightarrow Y) = \frac{|XUY|}{\overline{W}(X,Y)}$$

A rule X Y is considered confident if its confidence exceeds a given threshold,min conf.We now return to our running example. Looking at itemset cd, we see that the occurrence of a c at time stamp 1 will reduce the value of rule c d, but not of rule d c. Indeed, we see that W(cd, 1) = 12, and the minimal window containing cd for the other three occurrences of c is always of size 3. Therefore, W(c, d) = 21/4= 5.25, and c(c d) = 2 /5.25 = 0.38. Meanwhile, the minimal window containing cd for all occurrences of d is always of size 3. It follows that

W(d, c) = 9/3 = 3 and c(d c) = 2/3 = 0.67. We can conclude that while an occurrence of a c does not highly imply finding a d nearby, when we encounter a d we can be reasonably certain that a c will be found nearby. We also note that, according to our definitions, c(a b) = 1 and c(g c) = 1, as desired.

Improved Interesting Itemsets Algorithm

The algorithm proposed in [2] and given in Algorithm 1, finds interesting itemsets as defined in Section 3 by going through the search space (a tree) in a depth-first manner, pruning whenever possible. The first call to the algorithm is made with X empty, and Y equal to the set of all items.

Algorithm 1 INIT(<X, Y>) finds interesting itemsets

if UBI(<X, Y >) ◊min_int and size(X) ≤ max size then

if Y = ϕ; then
output X
else
Choose a in Y
INIT(<X U {a}, Y \ {a}>)
INIT(<X, Y \ {a}>)
end if
end if

References

1. R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 207-216, 1993.
2. B. Cule, B. Goethals and C. Robardet, A new constraint for mining sets in sequences, Proc. SIAM Int. Conf. on Data Mining (SDM), pp. 317-328, 2009.
3. G. Das, K-I. Lin, H. Mannila, G. Renganathan and P. Smyth, Rule discovery from time series, Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 16-22, 1998.
4. G. C. Garriga, Discovering Unbounded Episodes in Sequential Data, Proc. 7th Eu-ropean Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 83–94, 2003.
5. S. K. Harms, J. Saquer and T. Tadesse, Discovering Representative Episodal Association Rules from Event Sequences Using Frequent Closed Episode Sets and Event Constraints, Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 603-606, 2001.
6. S. Laxman and P. S. Sastry, A survey of temporal data mining, SADHANA, Academy Proceedings in Engineering Sciences, volume 31, part 2, pp. 173–198,2006.
7. H. Mannila, H. Toivonen and A. I. Verkamo, Discovery of Frequent Episodes in Event Sequences, Data Mining and Knowledge Discovery, volume 1(3), pp. 259–289, 1997.
8. N. M´eger and C. Rigotti, Constraint-Based Mining of Episode Rules and Optimal Window Sizes, Proc. 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 313–324, 2004.