[DS-08] Association rules

Miguel-Angel Canela
Associate Professor, IESE Business School

What is an association rule?

A **rule** takes usually the form $A \Rightarrow B$. A is the **antecedent** (LHS) and B is the **consequent** (RHS). The rule is read as *if* A, *then* B. **Association rules** represent patterns in the data without a specified target attribute (i.e. there is no Y, only X's). So, mining association rules mining is an example of **unsupervised learning**. Association rules are, in general, **local patterns**, which apply only to a small proportion of instances.

Association rules are typically searched in **transactions data**. Each transaction contains a set of **items**. For instance, in a typical application to supermarket data, the transactions are the visits of the customers to an outlet, and the items are the different products purchased: milk, sugar, etc. An example of an association rule is butter ⇒ whole milk. In **clickstream analysis**, the transactions are the pages visited by an individual in an uninterrupted sequence.

The antecedent and the consequent of a rule are always **disjoint**, that is, they do not have common items. Also, in the rules discussed in this note, the consequent consists of a single item. The antecedent can contain more than one item.

In the supermarket example, we look at every transaction as a set of items purchased at a time. Nevertheless, transaction data rarely come in this way, because such data could only be processed by a program which could manage a set of vectors of different length. In practice, transaction data are structured data, with rows and columns. There are two typical formats:

- **Transactional data**, with two fields: (a) an ID field, which identifies the transaction, and (b) a content field, which contains the items involved in the transaction, one per row. Thus, one transaction covers as many rows as items it contains. We have data in this format in the project.
- **Tabular data**, with transactions in the rows and items in the columns. For each combination transaction/item, the entry is one if that particular transaction contained this particular item and zero if it was not so. An ID column, identifying the transactions, can also be included. The data of the example come in tabular form.

Since the number of items in each transaction is usually low, most of the entries in the transaction-item matrix are zeros. So, tabular data are **sparse**, which makes them inefficient. The best way to store these data is listing the coordinates (row and column) of the nonzero entries. This is not discussed in this lecture.

Frequent itemsets

An **itemset** is a set of items. The **support of an itemset** is the proportion of transactions that contain that itemset. So, if A is an itemset, $N(A)$ is the number of transactions containing the items of A and N is the total number of transactions, the support is

$$\mathrm{supp}(A) = \frac{N(A)}{N}\,.$$

To illustrate this, suppose that a supermarket data set covers 9,835 transactions, and that whole milk is included in 2,513 transactions, butter in 545 transactions, and both together in 271 transactions. Then, the supports of these three itemsets are, respectively:

$$\text{supp}\big(\text{whole milk}\big) = \frac{2,513}{9,385} = 0.256, \qquad \text{supp}\big(\text{butter}\big) = \frac{545}{9,385} = 0.505,$$

$$\text{supp}\big(\text{butter, whole milk}\big) = \frac{271}{9,385} = 0.028.$$

The support of an itemset can be regarded as an estimate of the **probability** that a transaction includes that itemset. So, we may think that the probability that a customer buys whole milk in a visit to the supermarket is 25.6%.

The most popular algorithm for searching frequent itemsets is the **Apriori algorithm**, which uses a minimum support threshold, picking first one-item frequent sets, then two-item sets containing only items already selected, etc. If the analyst is interested in finding rules, a second step follows, in which the rules are generated from the frequent itemsets, based the **parameters** described below.

Parameters for mining association rules

Formally defined, a rule is a pair of disjoint itemsets, the antecedent and the consequent. Rules are evaluated by means of three parameters, the support, the confidence and the lift. The **support of a rule** of is the same as the support of the itemset resulting from putting antecedent and consequent together, i.,e the proportion of transactions that contain both antecedent and consequent,

$$\text{supp}(A \Rightarrow B) = \frac{N(AB)}{N}\,.$$

Here, AB denotes the itemset resulting from the merger of A and B. The **confidence** of a rule is the proportion of transactions that contain both the antecedent and the consequent among those that contain the antecedent,

$$\text{conf}(A \Rightarrow B) = \frac{\text{supp}(AB)}{\text{supp}(A)} = \frac{N(AB)}{N(A)}\,.$$

In the same way as the support is taken as a probability, the confidence is taken as the **conditional probability** of the consequent given the antecedent. In the example, the support of butter (which occurs in 545 transactions) is 0.055. So,

$$\text{conf}\big(\text{butter} \Rightarrow \text{whole milk}\big) = \frac{\text{supp}\big(\text{butter, whole milk}\big)}{\text{supp}\big(\text{butter}\big)} = \frac{0.028}{0.055} = 0.497$$

or, equivalently,

$$\text{conf}\big(\text{butter} \Rightarrow \text{whole milk}\big) = \frac{N\big(\text{butter, whole milk}\big)}{N\big(\text{butter}\big)} = \frac{271}{545} = 0.497.$$

This can be read as *if you are buying butter, the probability that you are also buying whole milk is 49.7%*. Finally, the **lift** is the ratio of the confidence of a rule (a conditional probability) to the support of the consequent (an unconditional probability),

$$\text{lift}(A \Rightarrow B) = \frac{\text{conf}(A \Rightarrow B)}{\text{supp}(B)}\,.$$

Although it is not always used as a parameter for mining the rules, the lift is sometimes the better measure of how relevant is a rule. In the example,

$$\text{lift}\big(\text{butter} \Rightarrow \text{whole milk}\big) = \frac{\text{conf}\big(\text{butter} \Rightarrow \text{whole milk}\big)}{\text{supp}\big(\text{whole milk}\big)} = \frac{0.497}{0.256} = 1.941,$$

which is read as *the customer who buys butter buys whole milk 1.941 times more often than the average customer*.

The Apriori algorithm

In the typical implementation of the Apriori algorithm, the user specifies a minimum support and a minimum confidence. Since the objective is to discover local patterns, the support can be low (e.g. less than 10%) when the number of transactions is high. Note that finding itemsets with frequency with a minimum support can be computationally intensive if that minimum support is low. Once frequent itemsets are extracted, mining association rules with a minimum confidence is straightforward.

The Apriori algorithm can also be used for classification. This is done by setting the consequent as one of the values of the class attribute. Then the association rules become **classification rules**. Example: applying classification rules to fraud detection, we can identify profiles of potential fraudsters.

Recommendation based on association rules

A **recommender system** seeks to predict the rating or preference that a user would give to an item (e.g. music, books or movies) or social element (e.g. people or groups) that he/she had not yet considered. Recommender systems are common in e-commerce. Some examples are:

- When viewing a product on Amazon, the store recommends additional items, based on information of what other shoppers bought along with the currently selected item.
- Netflix offers predictions of movies that you may like to watch based on your previous ratings and watching habits, also taking into account the characteristics of the film.

Association rules can be used on **generic recommendations** (not user specific), such as those of Amazon: *customers interested in this product are also interested in* .... We set the antecedent of the rule equal to a particular item and the length of the rule to 2, taking the rules with high confidence as recommendations.

References

1. MJA Berry & GS Linoff (2011) *Data Mining Techniques for Marketing, Sales and Customer Relationship Management*, Wiley.

2. D Jannach, M Zanker, A Felfernig & G Friedrich (2010), *Recommender Systems — An Introduction*, Cambridge University Press.

3. DT Larose (2005), *Discovering Knowledge in Data*, Wiley.