MOUNT ROYAL UNIVERSITY

Department of
MATHEMATICS AND COMPUTING

COMP 4522

Database II: Advanced Databases

Contents

EXERCISE

TEN

ASSOCIATION RULES

Define association rules
Use support and confidence and lift to evaluate rules
Understand the A-Priori algorithem

Association Rules

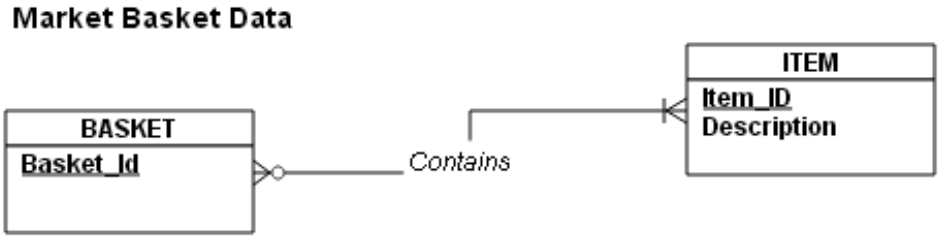
Finding associations is common goal in data mining. For example, go to http://chapters.ca and select a book for purchase. You will immediately be presented with a selection of books that others also looked at or purchased. Which other books to display is determined by association rules that have been found in the purchase data at Chapters.

A grocery store may find, from data mining, that customers who buy milk often also buy bread, so they can place the bread and milk in the store in such a way that customers are enticed to buy other stuff as well (tofu, pork rinds).

By Sander van der Wel from Netherlands [360/365] Private, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=34928584

Market Baskets and Rules

Most often data used for finding associations is organized into a *Market Basket* model.



The market basket model, despite its rather specific name is quite versatile. Items can be items for purchase in a store, or other things like hashtags in a tweet. Usually there are a large number of baskets and some smaller number of items.

We are interested in finding patterns in what goes into the baskets, that is, what items were bought together, or which hashtags appear together in a set of tweets.

These patterns are usually expressed as *rules*.

The formula $milk \implies bread$ denotes an *association rule* between milk and bread. Formally this is read as *milk implies bread*. milk is the *antecedent* and bread is the *consequent*.

For an association to make any sense it must have a *population*.

The population is a set of *instances*, sometimes called baskets (like a shopping basket, get it?).

For example, at a grocery store an instance would be the list of all products purchased during one trip to the store. For the on-line book sellers an instance may be all items that the customer has ever bought, because when they bought the book may be less important. For the tweet example a basket may be a single tweet.

Properties

The first property to understand is the *Support*.

Support

Support is defined as the percentage of baskets that have a particular item or items in them.

For example, suppose we have a population P where $|P| = 10,000$ of

grocery store baskets.

If there are 100 baskets that contain tofu then

$Supt(tofu) = 100 \div |P| = 0.01 = 1\%$

If there are 500 that contain pork rinds then

$Supt(porkrinds) = 500 \div |P| = 0.05 = 5\%$

and if there are two instances with both tofu and pork rinds in the same basket,

$Supt(porkrinds, tofu) = 2 \div |P| = 0.02\%$

Given a rule $R : A \implies B$, the support of R is the percentage of instances in the population that have both the A and B in them.

Confidence

Given a rule $R : A \implies B$, the confidence of R is the likelihood that B appears in a basket given that A is also there. If someone buys milk, what is the likelihood they will also buy bread?

It can be calculated by finding the support of the rule and dividing by the support of antecedent.

$Conf(R : A \implies B) = \frac{Supt(R)}{Supt(A)}$

From the example above, if we have a rule $R : tofu \implies porkrinds$ then

$Conf(R) = \frac{Supt(R)}{Supt(A)} = \frac{0.02}{1} = 0.02$

Not much confidence in that rule. For another example, suppose $Supt(bread) = 30$, $Supt(milk) = 40$ and $Supt(bread, milk) = 26$

Then for rule $R : bread \implies milk$

$Supt(R) = 26$

and

$Conf(R) = \frac{Supt(bread,milk)}{Supt(bread)} = 26 \div 30 = 87\%$

Lots of confidence in that rule!

Note that while $Supt(A \implies B) = Supt(B \implies A)$, $Conf(A \implies B) \neq Conf(B \implies A)$.

For example, if the rule is $S : milk \implies bread$, then

$Supt(S) = 26$

and

$Conf(S) = \frac{Supt(milk,bread)}{Supt(milk)} = 26 \div 40 = 65\%$

We have more confidence that bread implies milk than milk implies bread.

Lift

The *Lift* of a rule $R : A \implies B$ is the ratio of how many times B will appear when A appears. To calculate lift

$Lift(R : A \implies B) = \frac{Conf(R)}{Supt(B)}$

Other times you may see Lift defined as:

$Lift(R : A \implies B) = \frac{Supt(R)}{Supt(A) \times Supt(B)}$

But the confidence of R is $\frac{Supt(R)}{Supt(A)}$ so they are equivalent.

A lift of 1 means there is no association between A and B. Lift greater than one means there is a likelihood A and B will appear together. A lift less than 1 means there is no likelihood A and B will be bought together.

Interest

There is another measure, the *Interest*, that may be of interest.

If we have a rule, $R, A \implies B$ then the interest is defined as

$I(R) = Conf(R) - Supt(B)$

If R has no influence on B the fraction of baskets including A and B would be the same as the fraction of all baskets that contain B, and so would have an interest of 0.

High positive interest means R influences B to appear while high negative interest mean R influences B to not appear.

Finding Frequent Itemsets

To find rules from a data set one first finds *frequent itemsets*. These are sets of items that occur often, or formally, they have high support.

If there are only a small number of items one could produce a list of all the possible subsets of the items and check the support of each. However, the number of subsets grows exponentially, and the problem quickly becomes intractable. For example, if a store sells 1000 different items (a modest amount by today's big store standards) then there are 2^{1000} possible combinations of purchases or possible instances. Clearly it is not possible to check them all for support or confidence.

Many algorithms have been developed to find frequent itemsets. One is the A-priori method that works like this.

1. Decide on a threshold of support, for example 60%.
2. Create all the possible single item itemsets and calculate the support for each one.
3. Discard all items that have less than the threshold of support.
4. With the reamining items create all possible pair itemsets and calculate the support for each of these.
5. Discard all pairs with less than the threshold of support.
6. Create itemsets with three items and continue the process.

This works by *pruning the search space* and by the fact that for a set to have high support, all of its subsets must also have high support. Think about that and be sure to understand it.

Now that we have a set of frequent itemsets we need to find rules from them.

Finding Rules

If we have an itemset J with n items, we can create n rules of the form: $J - \{j\} \Longrightarrow j$ for all $j \in J$. For example, if we have an itemset $\{A, B, C\}$ we can generate three rules:

- $B, C \Longrightarrow A$
- $A, C \Longrightarrow B$
- $A, B \Longrightarrow C$

Which of these rules should we use? We can calculate the confidence, support and lift for each rule, which is not hard as we already calculated the support figures when we found the frequent itemsets. We want high support, high confidence, high lift rules.

Example

An example with phrases.

Item	Text
1	Cat, and, dog, bites
2	Yahoo, news, claims, a, cat, mated, with, a, dog, and, produced, viable, offspring
3	Cat, killer, likely, is, a, big, dog
4	Professional, free, advice, on, dog, training, puppy, training
5	Cat, and, kitten, training, and, behavior
6	Dog, &, Cat, provides, dog, training, in, Eugene, Oregon
7	“Dog, and, cat”, is, a, slang, term, used, by, police, officers, for, a, male-female, relationship
8	Shop, for, your, show, dog, grooming, and, pet, supplies