PSS718 - Data Mining

Lecture 7 - Association Analysis

Asst.Prof.Dr. Burkay Genç

Hacettepe University

November 6, 2016

What is it?

Definition (Association Analysis)

Association analysis identifies relationships or correlations between observations and/or between variables in our datasets.

- Particularly successful in mining very large transactional databases, like shopping baskets and on-line customer purchases
- Association analysis is one of the core techniques of data mining

Motivation Example

- 0.5% of all customers bought books A and B together
  - Not very interesting!
- 70% of these customers (who bought A and B) purchased book C
  - Interesting!
- How do we find such relations?

Transactions

- Each transaction is represented as an itemset
  - $\{A, B, C, D, E, F\}$
- The aim is to identify collections of items that appear together in multiple baskets
  - such as $\{A, C, F\}$
- From these itemsets, we identify rules
  - $\{A, F\} \implies C$

Association rules

- The outcome of an association analysis is association rules
  - $\mathcal{A} \rightarrow \mathcal{C}$
- Both $\mathcal{A}$ and $\mathcal{C}$ are itemsets. $\mathcal{A}$ is called the *antecedent* and $\mathcal{C}$ is called the *consequent*.
- Examples:
  - *milk* $\rightarrow$ *bread*
  - *beer&nuts* $\rightarrow$ *potato crisps*
  - *cigkofte* $\rightarrow$ *marul&nar eksisi*
- This can be extended to variable - value pairs:
  - $(WindDir3pm = NNW) \rightarrow (RainToday = No)$

Basis

- The basis of an association analysis algorithm is the generation of frequent itemsets.

Definition

A frequent itemset is a set of items that occur together frequently enough to be considered as a candidate for generating association rules.

- The obvious approach is quite expensive. Why?

Obvious approach

1. Let T be all transactions
2. Let L be the list of all items occuring in T
3. Let S_L be all possible combinations of the items in L
4. For each $s_i \in S_L$ count the number of times it occurs in T
5. Return significantly large s_i counts

Complexity

$$O(|T| \times |S_L|) = O(|T| \times 2^{|L|}) = O(2^{|L|})$$

Alternative approach

1. Let T be all transactions
2. For each $t_i \in T$
   - Compute S_{t_i}, all possible subsets of t_i
   - For each $s \in S_{t_i}$ increase the count by 1

Complexity

$O(\sum_{i=1}^{|T|} 2^{|t_i|})$

How to make it faster?

Idea

All subsets of a frequent itemset must also be frequent

- If we have many $\{milk, bread, cheese\}$ sets, then we must have *at least as many* $\{milk, bread\}$, $\{bread, cheese\}$, $\{milk, cheese\}$, $\{milk\}$, $\{bread\}$ and $\{cheese\}$ sets.
- Contraposition: If we don't have many $\{milk\}$, then we don't have many $\{milk, bread, cheese\}$
- Now we can count bottom-up:
  - Count individual items
  - Eliminate items with very low frequencies
  - Construct 2-item sets and count them
  - Eliminate 2-item sets with low frequencies
  - Repeat with 3-item, 4-item, ... sets

Complexity

- Runtime depends on how fast we prune the search space
- We eliminate all items/sets below a certain threshold, called *support*
- If we have a low support, the speed will be lower
- If we have a high support, the speed will be higher

Next phase

- Once the frequent itemsets are found, create possible association rules

Example

For subset $\{bread, milk, cheese\}$, create:

- $\{milk\} \rightarrow \{bread, cheese\}$
- $\{bread\} \rightarrow \{milk, cheese\}$
- $\{cheese\} \rightarrow \{milk, bread\}$
- $\{bread, milk\} \rightarrow \{cheese\}$
- $\{milk, cheese\} \rightarrow \{bread\}$
- $\{bread, cheese\} \rightarrow \{milk\}$

Confidence

- Now, compute *confidence* of each rule

Definition (Confidence)

Confidence of a rule $A \rightarrow C$ is the ratio

$$\frac{c(C \cup A)}{c(A)}$$

where $c()$ represents counts.

Example

For $T = \{A, B, C\}, \{A, B\}, \{B, C, D\}, \{A, C\}, \{B, D\}, \{A, C, D\}$
confidence of $\{A\} \rightarrow \{B\}$ is $2/4 = 0.5$

- We accept only rules with a certain level of confidence, such as 90%

Support

- The minimum support is expressed as a percentage of the total number of transactions in the dataset

Definition (Support)

Support for a collection of items $\mathcal{I}$ is the proportion of all transactions in which all items in $\mathcal{I}$ appear. The support for an association rule is expressed as

$$support(A \rightarrow C) = P(A \cup C)$$

- Typically, we use small values for support, such as 5%.

Confidence

- The minimum confidence is also expressed as the proportion of the total number of transactions in the dataset

Definition (Confidence)

$$confidence(A \rightarrow C) = P(C|A) = P(A \cup C)/P(A)$$

or,

$$confidence(A \rightarrow C) = support(A \rightarrow C)/support(A)$$

- Typically, we use high values for confidence, such as 90%.

Lift

- Another measure used in Rattle and R is *lift*

Definition (Lift)

Lift compares the confidence of a rule with the support of the consequent

$$lift(A \rightarrow C) = confidence(A \rightarrow C)/support(C)$$

or,

$$lift(A \rightarrow C) = \frac{support(A \rightarrow C)}{support(A) \times support(C)}$$

- A rule with lift equal to 1 means the antecedent and consequent appear in transactions independently. A lift greater than 1 means the rule can be successfully used for making predictions

Leverage

- Another measure used in Rattle and R is *leverage*

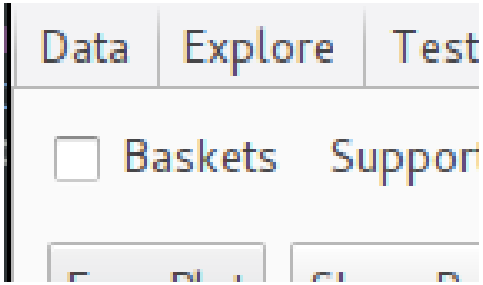
Definition (Leverage)

$$leverage(A \rightarrow C) = support(A \rightarrow C) - support(A) \times support(C)$$

- A rule with leverage equal to 0 means the antecedent and consequent appear in transactions independently. A positive leverage points at a potential association rule.

Basket Analysis

- The baskets checkbox allows you to do a market transaction analysis, assuming ident variable represents baskets, and target variable represents items.



Example

Ident	Target
1	Bread
1	Milk
2	Milk
2	Cheese

Basket Example

- Load the dvdtrans.csv file into Rattle
  - First load weather data, then click on the "filename" button
- Goto Association tab
- Check Baskets
- Execute

Loading the dataset

- Load the dataset from file:

```
> library(arules)
> library(rattle)
> dvdtrans <- read.csv(system.file("csv", "dvdtrans.csv", package = "rattle"))
```

- Convert into "transactions" format to be processed:

```
> data <- as(split(dvdtrans$Item, dvdtrans$ID), "transactions")
> data
transactions in sparse format with
 10 transactions (rows) and
 10 items (columns)
```

Running the model

```
> model <- apriori(data, parameter=list(support=0.2, confidence=0.1))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.1    0.1    1 none FALSE            TRUE       5     0.2      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 10 transaction(s)] done [0.00s].
sorting and recoding items ... [7 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [20 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Inspecting the rules

```
> inspect(sort(model,by="confidence")[1:5])
    lhs                        rhs             support confidence lift
[1] {LOTR1}                 => {LOTR2}         0.2     1          5.000000
[2] {LOTR2}                 => {LOTR1}         0.2     1          5.000000
[3] {Green Mile}            => {Sixth Sense}   0.2     1          1.666667
[4] {Patriot}               => {Gladiator}     0.6     1          1.428571
[5] {Patriot,Sixth Sense} => {Gladiator}     0.4     1          1.428571
```