MONASH University

ETX2250/ETF5922: Data Visualization and Analytics

Market Basket Analysis

Lecturer: *Lauren Kennedy*

Department of Econometrics and Business Statistics

lauren.kennedy1@monash.edu

Week 9

Market basket analysis

- Also known as:
  - Association analysis
  - Affinity analysis
- When would we use market basket analysis?
- Supermarket would like to increase sales by :
  - Placing products in such a way that customers see items they are likely to add to their basket (e.g., chips and dip)
  - Providing voucher to encourage the purchase of likely items (e.g., fancy cat toys to owners who buy super premium cat food)
- What data do we have? Items purchased during each transaction:
  - What items are commonly purchased together
  - Does the purchase of item A make the purchase of item B more likely?

Can you think of an example where you have experienced the result of this analysis?

Online book stores increase sales by drawing customer's attention to particular books

Netflix helping you find the next movie

Terminology and Notation

- A transaction can be thought of as a **basket of items**. The letters represent the items. This transaction contains items A, B, C, D, E

$$Transaction = \{A, B, C, D, E\}$$

- A particular group of items is called an **itemset**. The following group contains items A, C and F

$$ItemSet = \{A, B, C, D, E\}$$

- The data might suggest that where A and F occur in a basket, C is also likely to occur. This is represented with an **Association Rule**

$$\{A, F\} \rightarrow \{C\}$$

- How do we find these association rules?

A transaction by product matrix

Transaction	A	B	C	D	E
1	1	1	1	0	0
2	1	1	1	1	1
3	0	0	0	1	0
4	1	0	0	0	0
5	0	1	1	0	0
6	1	0	1	1	0
7	0	0	1	0	0
8	0	1	1	0	0

- Rows represent transactions (different baskets)
- Columns represent different items that might be purchased
- 1 means the item was in the basket, 0 means it was not

Consider the itemset $\{A, B, C\}$

- There are several possible associative rules $IF \rightarrow THEN$
  - $\{A, B\} \rightarrow \{C\}$
  - $\{C, B\} \rightarrow \{A\}$
  - $\{A, C\} \rightarrow \{B\}$
- These are probabilistic. Baskets with A and B don't necessarily include C, etc.
  - Two baskets include A and B, both include C, so this rule works all the time
  - Four baskets include C and B, and of those, two include A, so this rule works half the time
  - Three baskets include A and C, and of those, two include B, so this rule works 2/3 of the time.

How often does the rule $\{C, D\} \rightarrow \{B\}$ work?

More formal language

In the rule

$$\{A, B\} \rightarrow \{C\}$$

- $\{A, B\}$ is called the antecedent
- $\{C\}$ is called the consequent
- The **support** of the itemset $\{A, B, C\}$ is the proportion of **all** transaction that include A, B and C
- The **support count** of the itemset $\{A, B, C\}$ is the number of transactions that include A, B and C
- The **support of the association rule** $\{A, B\} \rightarrow \{C\}$ is the support of $\{A, B, C\}$

Confidence

- How can we describe uncertainty about this rule?

$$\text{Confidence} = \frac{\text{Support Count}\{A, B, C\}}{\text{Support Count}\{A, B\}}$$

- Confidence is a conditional probability, Pr(Consequent|Antecedent)

$$\text{Confidence} = \frac{p(\text{Antecedent AND Consequent})}{p(\text{Antecedent})}$$

Consider $\{C, D\} \rightarrow \{B\}$

What is the association rule?

What is the antecedent?

What is the consequent?

What is the itemset?

What is the support?

What is the support count?

What is the confidence?

Lift

- What if C is just super common in the data?
- Are you more likely to find the consequent(C) in a basket that contains the antecedent (A and B)? How much more likely?

$$\text{Lift of an association rule} = \frac{\text{Confidence of association rule}}{\text{Support of consequent}}$$

- Numerator = likelihood of C in a basket that contains A and B
- Denominator = likelihood of C in any basket
- **Lift** measures how many times more likely the **consequent** is when the **antecedent** is present
- For the rule to be useful, lift needs to be more than 1

What is a good association rule?

- An association rule is ultimately judged on how actionable it is and how well it explains the relationship between item sets.
  - Ex. Wal-Mart mined its transactional data to uncover strong evidence of the association rule, “If a customer purchases a Barbie doll, then a customer also purchases a candy bar.”
- An association rule is useful if
  - Its antecedents strongly increase the likelihood of the consequent, (measured by the lift) and
  - It explains an important previously unknown relationship
  - It should also have reasonably large support

Computational challenges & the apriori algorithm

- How do we generate these candidate rules?
- We could use all possible rules, but why couldn't we do this for very big data?
  - Take a three item set $\{A, B, C\}$
  - How many possible rules are there?
  - If p = number of items in set, $3^p - 2^{p+1} + 1$
  - In the case of 3 items, this is 12 $A \to B$, $A \to C$, $B \to A$, $C \to A$, $B \to C$, $C \to B$, $\{A, B\} \to C$, $\{A, C\} \to B$, $\{B, C\} \to A$, $A \to \{B, C\}$, $B \to \{C, A\}$, $C \to \{A, B\}$
  - In the case of 6 items, 602; 7 items 1932
  - If we have a large number of items there will be too many rules!
- Often focus on frequent item-sets (see support)

A priori algorithm

1. Generate frequent itemsets with just one item (count how many transactions contain that item in the data)
2. Drop items that have support below the desired minimum support
3. Generate frequent two item sets using only the frequent one item sets.
4. Generate the frequent three item sets using only the frequent two item sets...

- The idea is that if the smaller itemset doesn't have sufficient support, a larger itemset won't have sufficient support.

An example of the power of associations: The Netflix Prize

- Netflix offered a prize of $1 million to anyone who could improve their recommendations by 10%
- Online selling provides data
  - Observe what movies are rented together or by the same individual at different times (this is before the monthly fee version of netflix)
  - Netflix movies also have customers' ratings of movies
- Develop recommendation systems
  - Suggest purchases based on what the individual has already purchased and their ratings

Two approaches:

- The direct approach:
  - Rate movies by factors such as: amount of comedy, complication of plot, amount of blood and gore, how handsome the lead actor was, and a couple more.
  - Classify the viewer's taste in the same way
  - Recommend movies that coincide well with the viewer's taste
- Alternatively:
  - Start with random factors assigned to movies, each a combination of available linear variables
  - Tune the factors to align more and more with the viewer's tastes
  - The factors may not be as comprehensible as "how much comedy" but they predicted viewer's ratings of movies very well
  - Actually even more complex (http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/)

How to in R

Shopping basket transactions at the Hy-Vee Grocery store.

- Aims: to suggest:
- How to increase sales through product placement
- How to increase sales through cross-product promotions
- First read in the data


```
transaction_matrix_v1 <-  read_csv(here::here("data/HyVeetr.csv"), col_names = F
```


- Is this a good format? Rows = baskets, columns = item (no particular order)

Shopping basket transactions at the Hy-Vee Grocery store.

- Let's use a special function from the `arules` function to read in as transactions

```
library(arules)
transaction_matrix <-  read.transactions(here::here("data/HyVeetr.csv"), sep = "
transaction_matrix
```



```
## transactions in sparse format with
##  10 transactions (rows) and
##  15 items (columns)
```

- This is a very different structure to what we have used before, what does it look like?

```
str(transaction_matrix)

## Formal class 'transactions' [package "arules"] with 3 slots
##   ..@ data       :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
##   .. .. ..@ i       : int [1:53] 1 3 6 7 8 10 1 2 3 4 ...
##   .. .. ..@ p       : int [1:11] 0 6 15 19 25 31 36 42 46 49 ...
##   .. .. ..@ Dim     : int [1:2] 15 10
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : NULL
##   .. .. ..@ factors : list()
##   ..@ itemInfo   :'data.frame':    15 obs. of  1 variable:
##   .. ..$ labels: chr [1:15] "Beer" "Bread" "Cheese" "Cream" ...
##   ..@ itemsetInfo:'data.frame':    0 obs. of  0 variables
```


- Complex!

Getting back to a binary incidence matrix

- The storage of the transaction matrix is designed for efficient memory and computation (remember the computation issues!)
- This is a deviation from our usual tidy data, purely for computational efficiency
- But I want to visually inspect our transaction matrix so I am going to extract it out. You won't be expected to do this in an exam


```
binary_incidence <- as(t(transaction_matrix@data), "matrix")
colnames(binary_incidence)<- transaction_matrix@itemInfo$labels
```

```
kableExtra::kable(binary_incidence)
```

Beer	Bread	Cheese	Cream	crisps	Crisps	Eggs	Fruit	Jam	MapleSyrup	Milk	Softdrink	Steak	Ve
FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FA
FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TR
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FA
FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FA
FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FA
FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FA
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FA
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FA
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FA
TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TR

Consider $\{Bread, Milk\} \rightarrow \{Cheese\}$

What is the association rule?

What is the antecedent?

What is the consequent?

What is the itemset?

What is the support?

What is the support count?

What is the confidence?

Frequency of items

- Let's get the most frequent items using the `eclat` function
- Specify the support (`supp`) and the maximum number of items in each item set (`maxlen`)

```
frequentItems <- eclat (transaction_matrix,
                        parameter = list(supp = 0.5, maxlen = 3))
```



```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen          target  ext
##     FALSE     0.5      1      3 frequent itemsets TRUE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 5
```

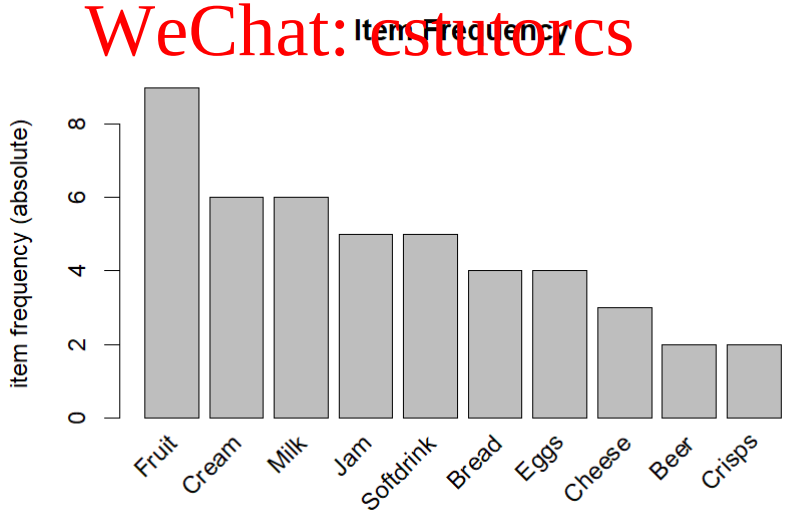
```
inspect(frequentItems)
```

```
##      items                 support transIdenticalToItemsets count
## [1]  {Fruit,Softdrink}     0.5     5                         5
## [2]  {Cream,Fruit,Jam}     0.5     5                         5
## [3]  {Fruit,Jam}           0.5     5                         5
## [4]  {Cream,Jam}           0.5     5                         5
## [5]  {Fruit,Milk}          0.6     6                         6
## [6]  {Cream,Fruit}         0.6     6                         6
## [7]  {Fruit}               0.9     9                         9
## [8]  {Cream}               0.6     6                         6
## [9]  {Milk}                0.6     6                         6
## [10] {Jam}                 0.5     5                         5
## [11] {Softdrink}           0.5     5                         5
```

Item Frequency plot

- Use the command `itemFrequencyPlot`
- Specify that we only plot the 10 most frequent items (with the `topN` argument).
- Specify we want the absolute frequency (rather than the relative) using the `type` argument, and add a title using `main`.


```
itemFrequencyPlot(transaction_matrix, topN=10,
                  type="absolute", main="Item Frequency")
```

Mine the rules (find the most common)

- Use the `apriori` function to use the apriori algorithm
- Need to specify the minimum support (`supp`), the minimum confidence (`conf`) and the minimum length considered.

```
basket_apriori <-apriori(transaction_matrix,
                         parameter = list(supp   = 0.5,
                                          conf   = 0.8,
                                          minlen = 2 ))
```

```
basket_apriori <-apriori(transaction_matrix,
                         parameter = list(supp   = 0.5,
                                          conf   = 0.8,
                                          minlen = 2 ))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.5      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 5
##
```

Extract the candidate rules

```
inspect(basket_apriori)

##     lhs                rhs     support confidence coverage lift     count
## [1] {Softdrink}     => {Fruit} 0.5     1.0000000  0.5      1.111111 5
## [2] {Jam}           => {Cream} 0.5     1.0000000  0.5      1.666667 5
## [3] {Cream}         => {Jam}   0.5     0.8333333  0.6      1.666667 5
## [4] {Jam}           => {Fruit} 0.5     1.0000000  0.5      1.111111 5
## [5] {Milk}          => {Fruit} 0.6     1.0000000  0.6      1.111111 6
## [6] {Cream}         => {Fruit} 0.6     1.0000000  0.6      1.111111 6
## [7] {Cream,Jam}     => {Fruit} 0.5     1.0000000  0.5      1.111111 5
## [8] {Fruit,Jam}     => {Cream} 0.5     1.0000000  0.5      1.666667 5
## [9] {Cream,Fruit} => {Jam}   0.5     0.8333333  0.6      1.666667 5
```

What makes a rule good?

- When the antecedents hold, the consequent is very likely to hold
  - That is, the confidence is high
- The antecedents strongly increase the likelihood of the consequent;
  - That is the lift is reasonably large
- It explains a previously unknown relationship.
  - This is beyond what the calculations can tell you.
- A rule will not be useful if it applies to hardly any baskets. However, if the dataset (and your business) is sufficiently large, you may be interested in rules whose itemsets have relatively low support, as we will see with the lastfm example below.

How often does data come like this?

- The form of the grocery data was that each row represented a transaction, and the items in that transaction were listed, seperated by commas.
- There are other common forms of data, but they take a little more data munging before we can use them.
- One example is the data you will use in your tutorial. You will need extra data munging tools than the ones we covered in week 3 (R has an extensive way of manipulating data, but the basics we covered in week 3 are sufficient in many scenarios).

Last fm artist data

- lastfm.csv records 300,000 selections of artist selections by 15,000 users
- Also have the user's country and gender

```
library(tidyverse)
lastfm <- read_csv(here::here("lectures/data/week09/lastfm.csv")) %>%
  mutate(user = as.factor(user))
head(lastfm)
```



```
## # A tibble: 6 x 4
##   user  artist                  sex   country
##   <fct> <chr>                   <chr> <chr>
## 1 1     red hot chili peppers   f     Germany
## 2 1     the black dahlia murder f     Germany
## 3 1     goldfrapp               f     Germany
## 4 1     dropkick murphys        f     Germany
## 5 1     le tigre                f     Germany
## 6 1     schandmaul              f     Germany
```

How is this scenario like the supermarket scenario?

- user = shopper
- artist = item bought
- How is the data structure different from the structure of the supermarket scenario?

Group split

```
lastfm_split <- lastfm %>%
  group_by(user) %>%
  group_split()
```

- Now we have a new data object type! This is something called a list, which let's us store a large amount of dataframes in one object.
- Here we use `group_split()` to make a dataframe(or a tibble to be more accurate) for every user, which we store in this list for easy access

Using map from purrr

- This list of dataframes(tibbles) means we can apply some tidyverse data munging for each dataframe (user) using our usual dataframe munging.
- To see what's going on, let's focus in on the first user. We use the double square brackets to index into a list

```
user1 <- lastfm_split[[1]]
```

Data munging for user1

```
head(user1)
```

- We want to:
  - Select only the artist
  - Make sure we only have the unique artists (so drop any multiple entries)
  - Take it out of a dataframe

Data munging for user1

```
user1 %>%
  select(artist)%>%
  unique() %>%
  deframe()
```

Map from the purrr package.

- The map function in purrr lets us do this over every dataframe (tibble) in our list, with just one function call! Neat!
- We do this by creating a function that takes in x (in this case each data.frame/tibble in our list), and then runs our little cleaning routine over it.

```
playlist<- lastfm_split %>%
  purrr::map(function(x) x %>%
               select(artist) %>%
               unique() %>%
               deframe())
```


- Let's check

```
playlist[[1]]
```

```
##  [1] "red hot chili peppers"   "the black dahlia murder"
##  [3] "goldfrapp"               "dropkick murphys"
##  [5] "le tigre"                "schandmaul"
##  [7] "edguy"                   "jack johnson"
```

Almost finished with the data munging

- Now we have a data.frame every artist the user has listened to, for every user, stored in a list.
- Fortunately this is sufficient to coerce into the transactions matrix from before - phew!

```
playlist_tr<-as(playlist, "transactions")
playlist_tr
```



```
## transactions in sparse format with
##  15000 transactions (rows) and
##  1004 items (columns)
```

A popular story (from 2012)

- A father walks into a store of a local Target, furious.
- Turns out (as the story goes) his teenage daughter has been sent coupons and advertising surrounding maternity and newborn goods
- The storemanager apologizes profusely and the man leaves
- A few days later, the father calls the store and explains that his daughter was in fact pregnant.
- This was a widely publicized example, but I cannot find actual verification for it.
- Still it is a good cautionary tale of both the benefits and ethical risks of these methods

MONASH University

Lecturer: *Lauren Kennedy*

Department of Econometrics and Business Statistics

lauren.kennedy1@monash.edu

Week 9