MONASH University

MONASH BUSINESS SCHOOL

Market Basket Analysis & Association Analysis

Unsupervised Learning Technique

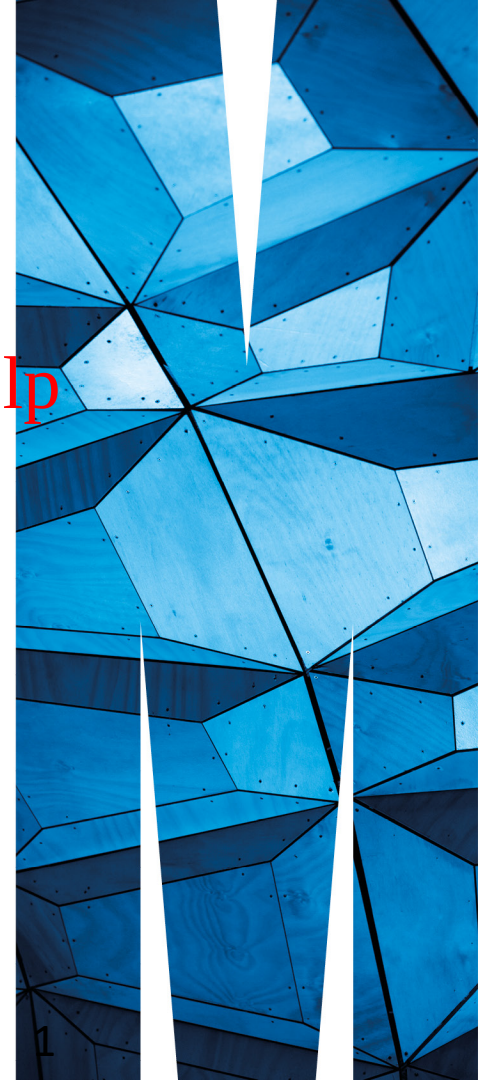
Presented by Klaus Ackermann

GROUP OF EIGHT AUSTRALIA

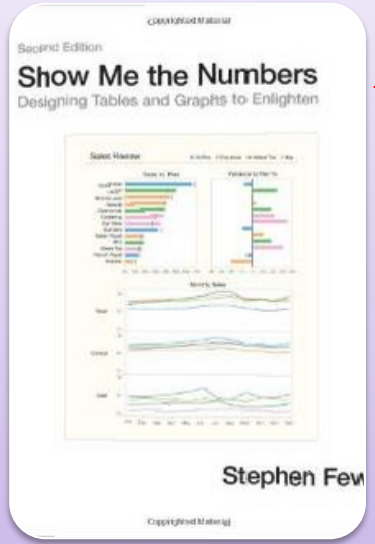
AACSB ACCREDITED

EFMD EQUIS ACCREDITED

ASSOCIATION AMBA ACCREDITED



Association Analysis Scenarios



Supermarket

- Would like to increase sale
  - By placing products in such a way that customers see items they are likely to add to their basket
  - Providing vouchers to encourage the purchase of likely items
- Available data: items purchased during each transaction, hence know:
  - What items were commonly purchased together
  - Does the purchase of item A make the purchase of item B more likely?

Association Analysis Scenarios

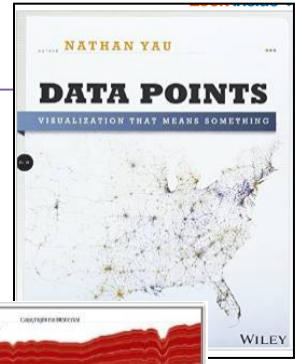
MONASH University

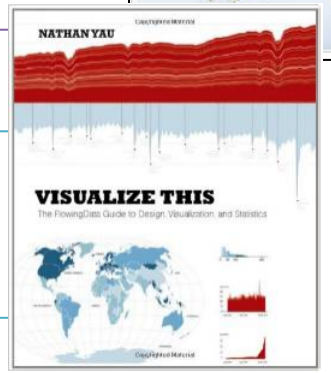
Online book stores

- Would like to increase sales by drawing customers' attention to books they would like to buy

Movie rentals

- Ex. Netflix: similar idea as above





Market Basket Analysis

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction
- Example: Hy-Vee Grocery store. Aims: to suggest:
  - How to increase sales through product placement
  - How to increase sales through cross-product promotions

Consider a Transaction

Transaction = {A, B, C, D, E, F}, thought of as a ***basket of items***

We consider particular groups of items, e.g. {A, C, F}. Such a group is called an ***itemset***

We may consider the baskets in which {A, C, F} all occur

The data may suggest that when A and F occur in a basket, then C is likely to occur in that basket. This may be represented as an ***Association rule: {A, F} → {C}***

Transaction	Bread	Eggs	Milk	Fruit	Jam	Cream	Soft drink	Vegetables	Crisps	Yoghurt	Cheese	Maple Syrup	Butter
1	1	1	1	1	1	1	0	0	0	0	0	1	0
2	1	1	1	1	1	1	0	1	1	0	1	1	1
3	0	0	0	1	0	1	0	0	0	0	0	1	1
4	1	0	0	1	1	1	1	0	0	0	0	0	0
5	0	1	1	1	1	1	1	0	0	0	0	0	0
6	1	0	1	1	1	1	1	0	1	0	0	0	1
7	0	0	1	1	0	0	1	0	1	0	0	0	0
8	0	1	1	1	0	0	1	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	1	1	0	0
10	0	0	0	0	0	0	0	1	0	1	1	0	1

Shopping Basket Transactions

How to conclude some general rules from shopping basket transaction data?

What goes with what?

Consider the association rules:

{*Eggs, Milk*} → {*Bread*}

{*Bread, Milk*} → {*Eggs*}

Shopping Basket Transactions

Consider the itemset {Bread, Milk, Eggs}

There are several possible association rules, such as

$$\{Eggs, Milk\} \rightarrow \{Bread\} \qquad \{Bread, Milk\} \rightarrow \{Eggs\}$$

The association rules are probabilistic → People who purchase eggs and milk don't always purchase bread and so on

Shopping Basket Transactions

The **support** of the itemset $\{Eggs, Milk, Bread\}$
is the proportion of all transactions that include Eggs, Milk and Bread.

The **support count** of the itemset $\{Eggs, Milk, Bread\}$
is the number of transactions that include Eggs, Milk and Bread

The **support of the association rule** $\{Eggs, Milk\} \rightarrow \{Bread\}$
is the support of the set comprising items in the antecedent and items in the consequent.

Shopping Basket Transactions

- {*Eggs, Milk*} occurs in transactions 1, 2, 5, 8 → that's 4/10

- Among these 4 transactions, {*Bread*} occur in transactions 1, 2 → that's 2/4

- Thus {*Bread*} occurs in half the transactions that contain {*Eggs, Milk*}

- We say the rule $\{Eggs, Milk\} \rightarrow \{Bread\}$ has confidence = 50%

Shopping Basket Transactions

- The **support** of an itemset is the proportion of all transactions in which that itemset occurs
- Calculate the support of $\{Bread, Milk, Eggs\}$

- The **confidence** of an association rule is the *proportion* of times that the consequent occurs, among all transactions where the antecedent occurs
- Calculate the confidence of the rule $\{Bread, Milk\} \rightarrow \{Eggs\}$

Terminology

- **Association rules**: if-then statements
  - If certain items are purchased, convey the likelihood of another specific item (or several other items) also being purchased
- **Antecedent**: The collection of items (or itemset) corresponding to the *if* portion of the rule
- **Consequent**: The collection of items (or itemset) corresponding to the *then* portion of the rule

- The itemset corresponding to the association rule consists of the items in the antecedent and the items in the consequent
- The **support** of an itemset is the proportion of transactions in which it appears
- The **support count** of an itemset is the *number* of transactions in which it appears

Terminology: Confidence

- **Confidence**: Among those transaction that include the antecedent, the proportion of transactions where the consequent appears

$$\frac{Support\ of\ \ antecedent\ and\ consequent}{Support\ of\ \ antecedent}$$

Terminology: Lift

- Are you more likely to find the **consequent (Bread)** in a basket that contains the **antecedent (Eggs and Milk)**? How much more likely?
  This is called the **Lift**

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

- Numerator = likelihood of **bread** in a basket that contains **eggs and milk**
- Denominator = likelihood of **bread** in any basket
- So lift measures how many times more likely the **consequent** is when the **antecedent** is present

Terminology: Lift

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

- For an association rule to be useful, the lift needs to be more than 1

What's 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

Netflix Prize

Netflix offered a prize of $1million 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.
  - Netflix movies also have customers' ratings of movies
- Develop recommendation systems
  - Suggest purchases based on what the individual has already purchased and their ratings

Direct Approach

How would you determine what movies to recommend based on past choices and likes?

- Rate movies by factors such as
  - How much comedy,
  - How complicated plot,
  - How much blood and gore
  - Handsomeness of lead actor…
  - and a couple of dozen more
- Classify the viewer's taste in the same way
- Recommend movies that coincide well with viewer's taste

But this isn't making best use of the available data

Start with *random* factors assigned to movies, each a linear combination of available variables

Tune the factors to align more and more with viewers' tastes

The factors may not be as comprehensible as "how much comedy" etc., but they predict viewers' ratings of movies very well

- $1million was won this way
- Actually a little more complex – see
- http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/
- Also see the Wikipedia article "Netflix prize"
- Our example (online radio) is a little simpler

lastfm artists data

lastfm.csv contains

- 300,000 records of artist selections, by
- 15,000 users
- Also supplies user's country and gender


```
  user                artist sex country
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
> length(lastfm$user)
[1] 289955
> lastfm$user <- factor(lastfm$user)
```

R Commands

- *split* divides up a data frame by the values of a categorical variable (factor) – in this case, “user”
  - We use *split* to produce a list for each user of the bands they listen to. This resulting object is a list, we give it the name *playlist*

- *lapply* applies a function to every element of a list or vector.
- In this example we will apply *unique* to playlist
- *unique* removes duplicated elements from the playlist for each user

R Commands

- *as* (playlist, "transactions") instructs R to view playlist as a list of transactions.
  - *transactions* is a data class defined in *arules*

- *itemFrequency* is another rules command: provides the *relative* frequency with which each band occurs among the 15000 users. Also *itemFrequencyPlot*

R Commands

- Finally, we get to actual association analysis
- *apriori*(playlist) mines playlist for frequent itemsets and association rules
- By default it lists rules with support ≥ 0.1, confidence ≥ 0.8.
- To change the default, the syntax is

  musicrules<-apriori(playlist, parameter = list(support = 0.01, confidence = 0.5))
- Example of restricting to rules for which lift > 5:
- *inspect*(*subset*(musicrules, subset = lift>5))

MONASH University

So Many Items to Consider

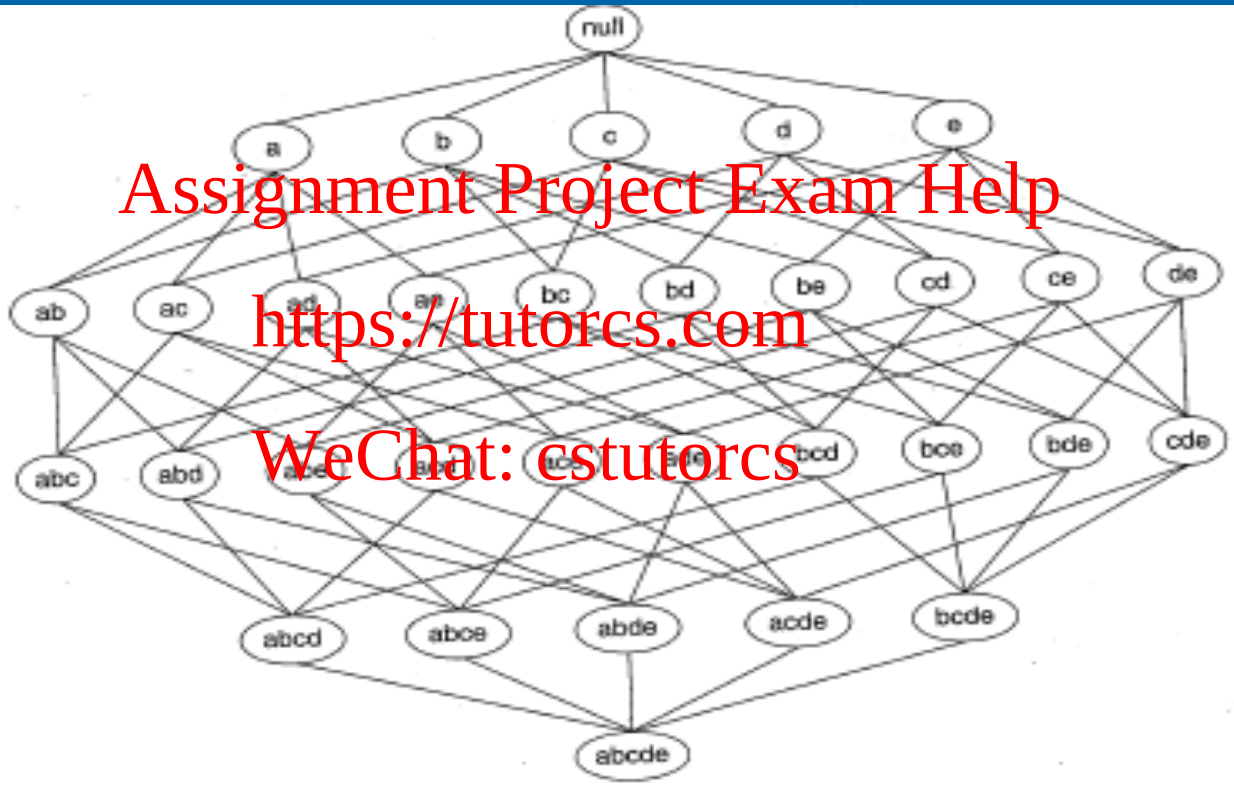


Figure 6.1. An itemset lattice.

MONASH University

So Many Items to Consider

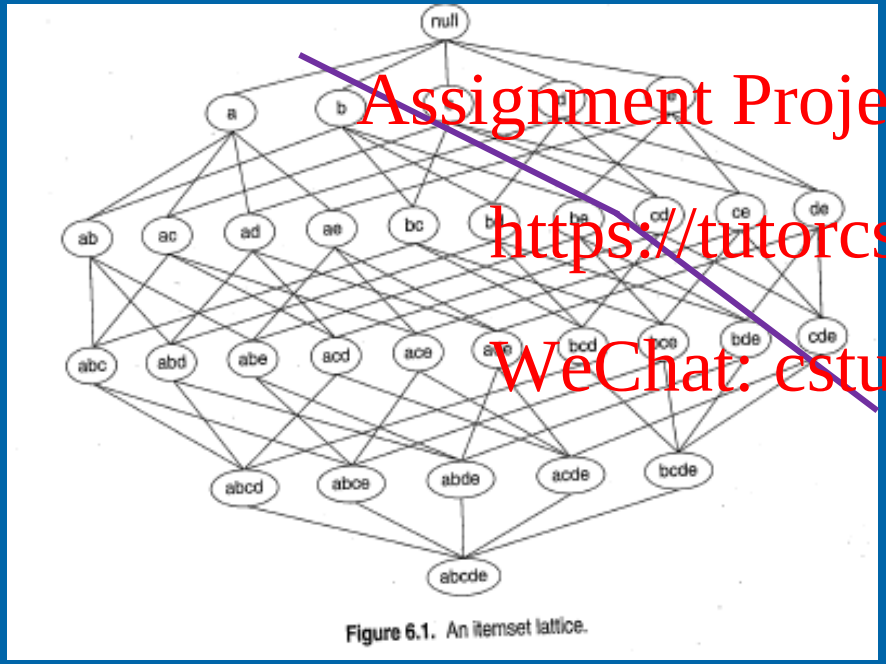


Figure 6.1. An itemset lattice.

If *cde* is frequent, so is every itemset above the line.

This is an example of the Apriori principle

MONASH University

So Many Items to Consider

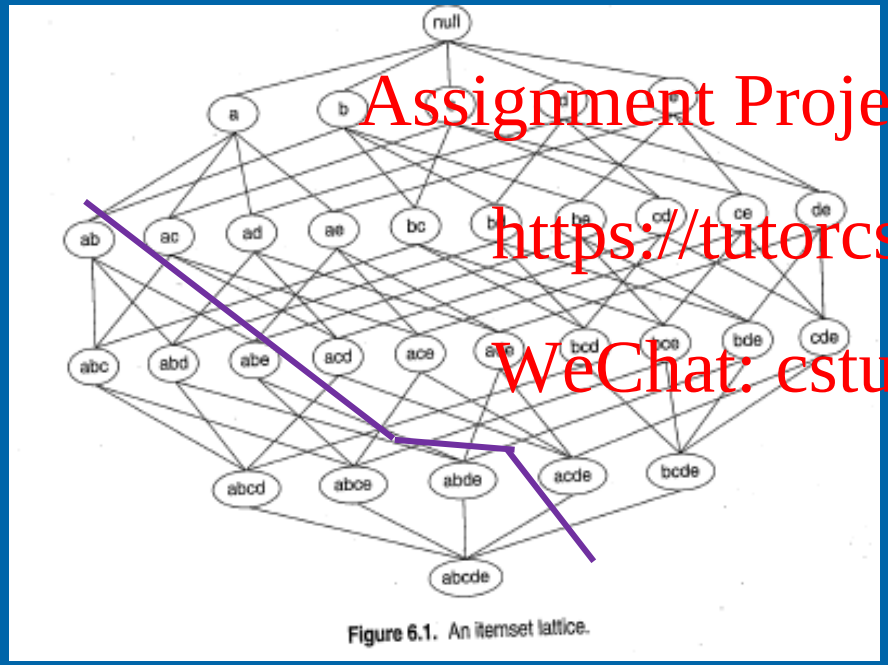


Figure 6.1. An itemset lattice.

If *ab* is infrequent, so is every itemset below the line

MONASH University

MONASH BUSINESS SCHOOL

GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED

EFMD EQUIS ACCREDITED

ASSOCIATION OF MBAS ACCREDITED