Relationship Mining

Association Rules

Sequential Pattern Mining

Study of
“What Goes with What?”

Association Rules

Study of “what goes with what”

Also called *market basket analysis, affinity analysis*

Origin: Study of customer transaction databases to determine dependencies between purchases of different items

POS Transaction Data

Large number of transaction records

- Data collected using bar-code scanners
- Each record lists all items purchased by a customer on a single purchase transaction

Are certain groups of items consistently purchased together?

How would you use this knowledge?

How can the knowledge be used?

Promotion on one item, raise price of related item

Placement in store

Stocking

DSS transaction no. 00M511265
LOADED QUERIED ANALYZED
AT 6:32 PM EVERY WEDNESDAY, OWEN BLY BUYS DIAPERS AND BEER. DO NOT JUDGE OWEN.

Toy Example:
cell phone faceplates

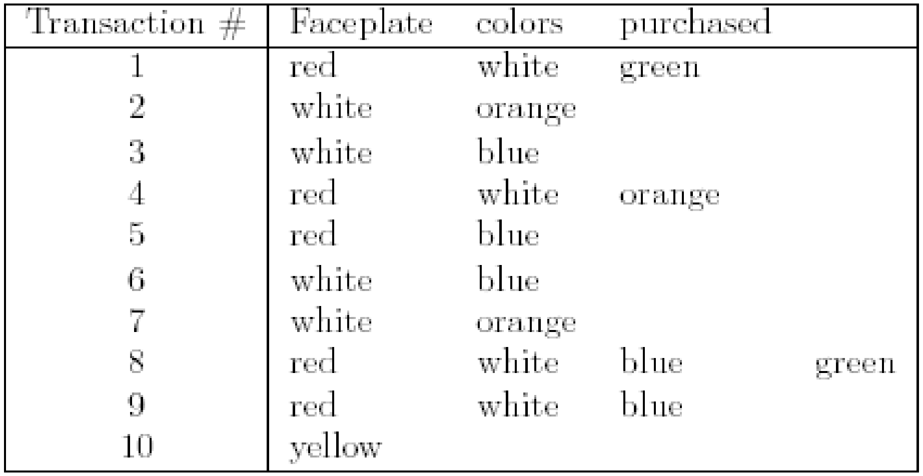
A store that sells accessories for cellular phones runs a promotion on faceplates

Buy multiple faceplates from a choice of 6 different colors and get a discount!

The store managers would like to know what colors of faceplates customers are likely to purchase together

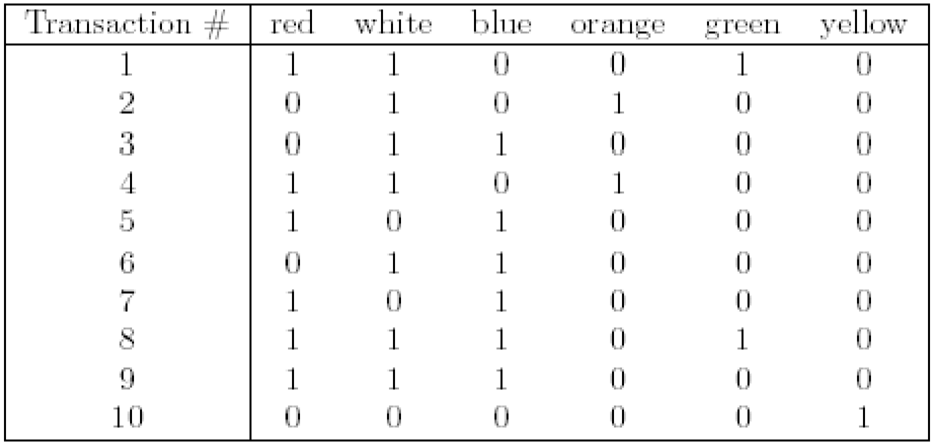
Data from first week of promotion (tiny example – 10 transactions)

List Format



Transaction #	Faceplate	colors	purchased	
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

Binary Matrix Format



Transaction #	red	white	blue	orange	green	yellow
1	1	1	0	0	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	0
4	1	1	0	1	0	0
5	1	0	1	0	0	0
6	0	1	1	0	0	0
7	1	0	1	0	0	0
8	1	1	1	0	1	0
9	1	1	1	0	0	0
10	0	0	0	0	0	1

Association Rules are probabilistic “if-then" statements

Basic idea:

- Examine all possible rules between items in “if-then“ format
- Select only rules most likely to indicate true dependence

Example: Rules from {red, white, green}

1. If {red, white} Then {green}
2. If {red, green} Then {white}
3. If {white, green} Then {red}
4. If {red} Then {White, Green}
5. If {white} Then {red, green}
6. If {green} Then {red, white}

Problem

Many rules are possible

How to select the TRUE/GOOD rules
From all generated rules

Terminology for Rules

Terminology:
“IF” part = Antecedent
“THEN” part = Consequent

- “If red and white, then green” (“If red and white faceplates are purchased, then so is a white one”)
  - Antecedent: Red and White
  - Consequent: Green

Performance Measure #1: Support

Consider only combinations that occur with higher frequency in the database

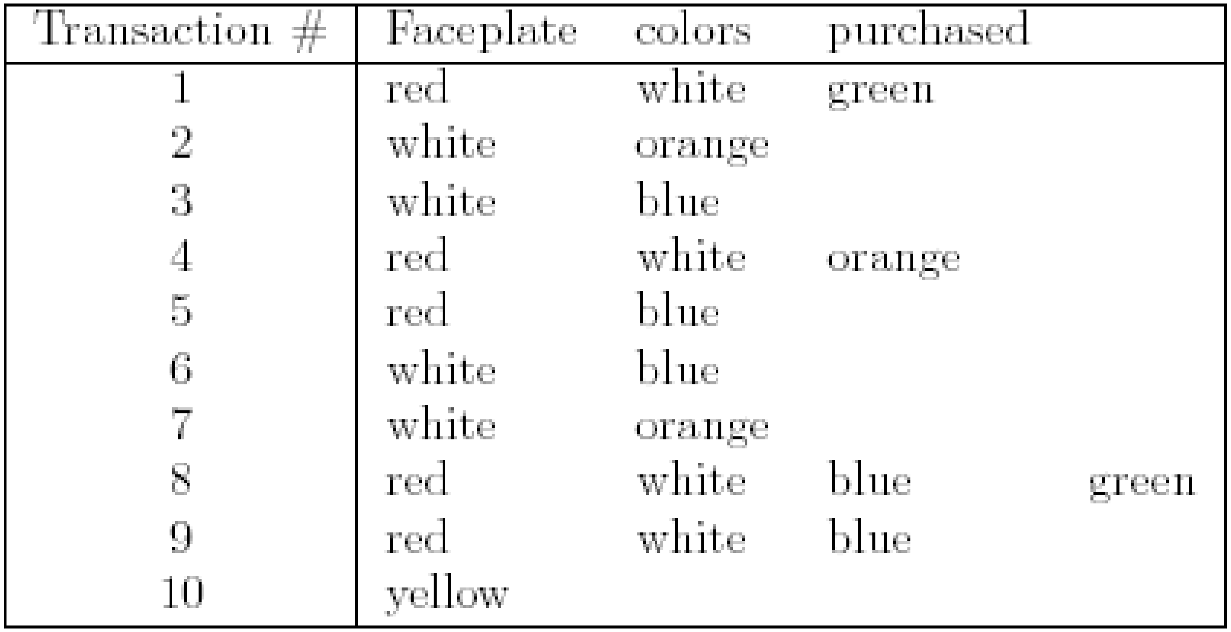
Criterion for “frequent”: *Support*

Support of a rule =

% (or number) of transactions in which antecedent (IF) and consequent (THEN) appear in the data

What is the support for “if white then blue”? (choose one or more)

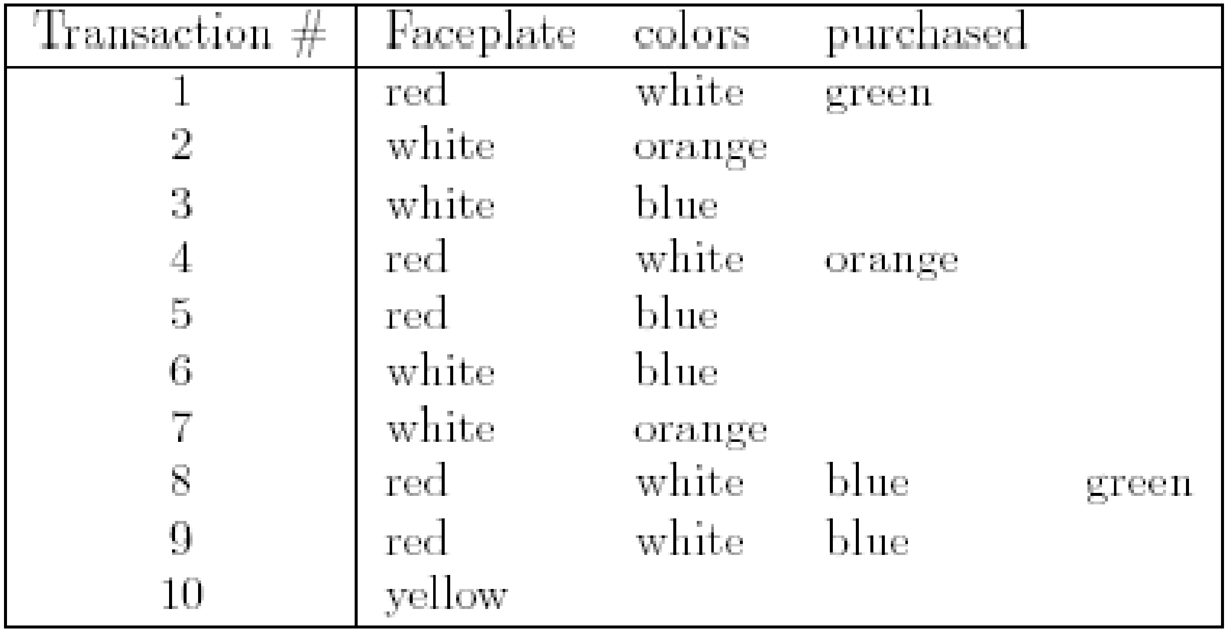
1. 4
2. 40%
3. 2
4. 90%



Transaction #	Faceplate	colors	purchased	
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

What is the support for “if blue then white”? (choose one or more)

1. 4
2. 40%
3. 2
4. 90%



Transaction #	Faceplate	colors	purchased	
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

Problem: Generating all possible rules is exponential in the number of distinct items

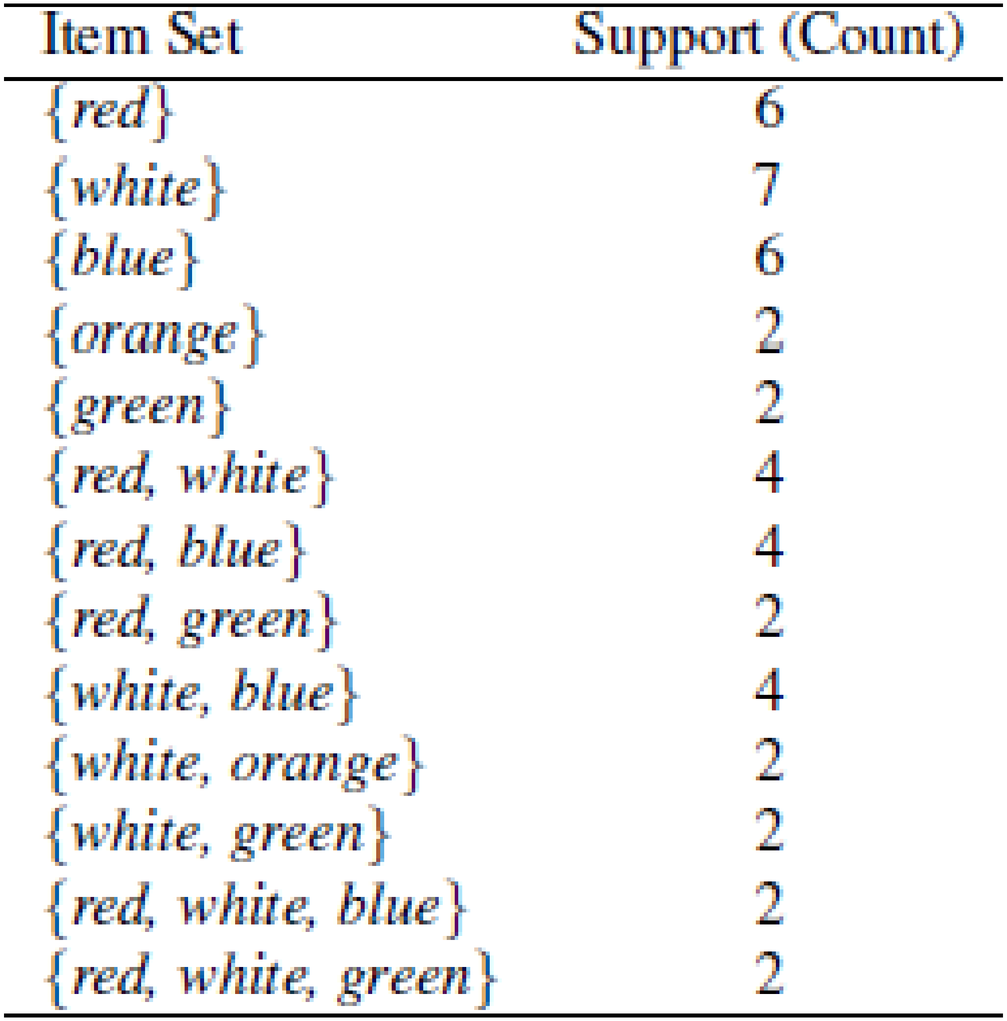
Solution: *Frequent item sets* using Apriori algorithm

Generating frequent item sets: The Apriori Algorithm

For *k* products…

1. Set minimum support criterion
2. Generate list of one-item sets that meet the support criterion
3. Use list of one-item sets to generate list of two-item sets that meet support criterion
4. Use list of two-item sets to generate list of three-item sets that meet support criterion
5. Continue up through *k*-item sets

Example: Generating frequent item sets: The Apriori Algorithm (when minimum support equals 2)



Item Set	Support (Count)
{*red*}	6
{*white*}	7
{*blue*}	6
{*orange*}	2
{*green*}	2
{*red, white*}	4
{*red, blue*}	4
{*red, green*}	2
{*white, blue*}	4
{*white, orange*}	2
{*white, green*}	2
{*red, white, blue*}	2
{*red, white, green*}	2

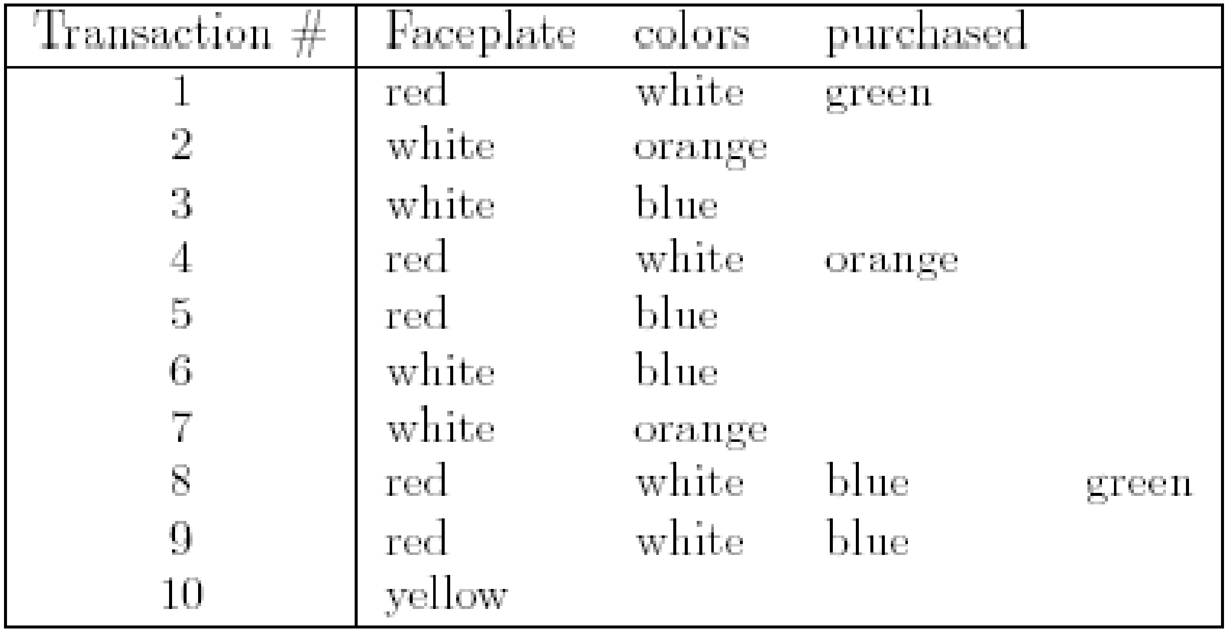
Performance Measure #2: Confidence

Confidence: % of antecedent (IF) transactions that also have the consequent (THEN) item set

$$\frac{\text{\# transactions with both antecedent \& consequent item sets}}{\text{\# transactions with antecedent item set}}$$

What is the confidence for “if white then blue”? (choose one or more)

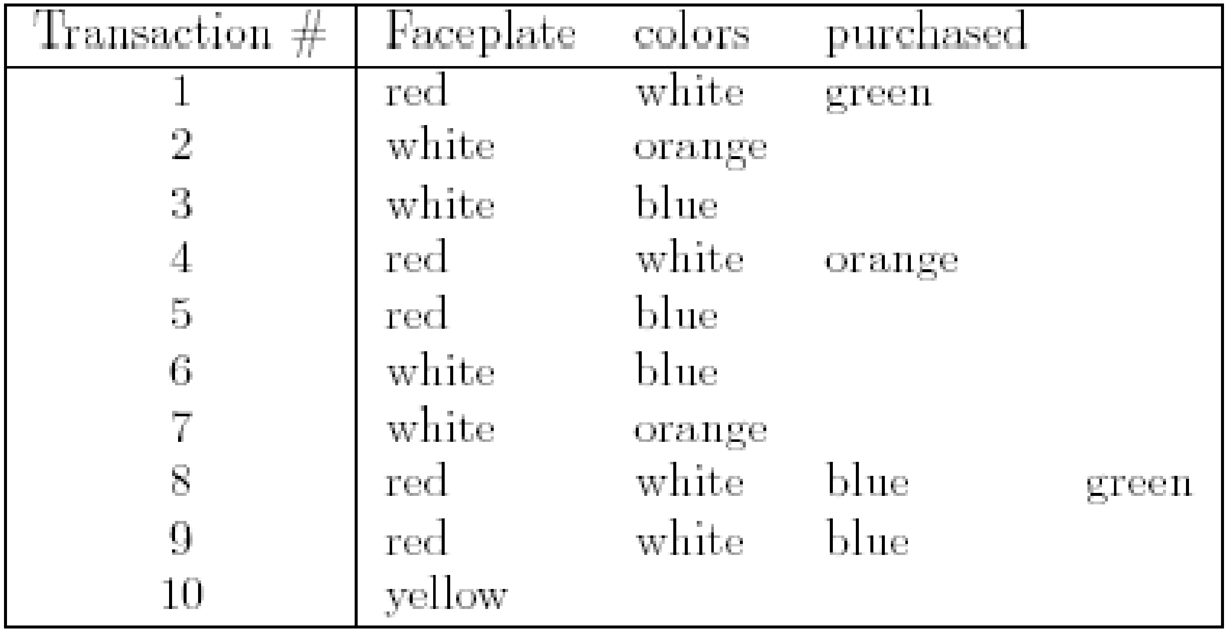
1. 4/5
2. 5/8
3. 5/8
4. 4/8



Transaction #	Faceplate	colors	purchased	
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

What is the confidence for “if blue then white”? (choose one or more)

1. 4/5
2. 5/8
3. 5/8
4. 4/8



Transaction #	Faceplate	colors	purchased	
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

The weakness of confidence

If antecedent and/or consequent have high support
→ High Confidence

Performance Measure #3: Lift ratio

Lift ratio= *confidence*/(*benchmark confidence*)

Benchmark assumes independence between antecedent and consequence:

P(antecedent & consequent) = P(antecedent) x P(consequent)

Benchmark confidence:

P(C|A) = P(C&A) / P(A) = P(C) x P(A) / P(A) = P(C) =

$$\frac{\text{\# transactions with consequent item sets}}{\text{\# transactions in database}}$$

Interpreting Lift

Lift > 1 indicates a rule that is useful in finding consequent items sets (i.e., more useful than selecting transactions randomly)

Interpretation revisited

- *Lift ratio* shows how effective the rule is in finding consequents vs. random (useful if finding particular consequent is important)
- *Confidence* shows the rate at which consequents will be found (useful in learning costs of promotion)
- *Support* measures overall impact (% transactions affected)

Process of Rule Selection

Generate all rules that meet specified support & confidence

- Find frequent item sets (those with sufficient support)
- From these item sets, generate rules with sufficient confidence

Caution: The Role of Chance

Random data can generate apparently interesting association rules

The more rules you produce, the greater the danger

Rules based on large numbers of records are less subject to this danger

Summary

- Association rules (*affinity analysis, market basket analysis*) produce rules on associations between items from a database of transactions/events
- The most popular method to enumerate rules rule: Apriori algorithm
- To reduce computation, we consider only “frequent” item sets (=support)
- Performance is measured by *confidence* and *lift ratio*
- Can produce a profusion of rules; review is required to identify useful rules and to reduce redundancy

Example: detecting a flu outbreak

A supermarket database has 100,000 POS transactions.

2,000 transactions include both orange juice and Strepsils

OJ
NUTRITION
Lozenges for sore throats and mouth infections
Strepsils
Original Flavour
Double antibacterial action
24 Lozenges
Strepsils

800 of the above 2,000 include soup purchases

What is the support for rule “IF (orange juice and Strepsils) are purchased THEN (soup) is purchased on the same trip" ?

1. 0.8%
2. 2%
3. 40%

What is the support for rule “IF (orange juice and Strepsils) are purchased THEN (soup) is purchased on the same trip" ?

1. 0.8%
2. 2%
3. 40%

Estimated P(antecedent & consequent) = 800/100,000 = 0.8%

What is the confidence for rule “IF (orange juice and Strepsils) are purchased THEN (soup) is purchased on the same trip" ?

1. 0.8%
2. 2%
3. 40%

What is the confidence for rule “IF (orange juice and Strepsils) are purchased THEN (soup) is purchased on the same trip" ?

1. 0.8%
2. 2%
3. 40%

Estimated P(consequent | antecedent) = 0.8% / 2% = 40%

To compute lift ratio for rule “*IF (orange juice and Strepsils) are purchased THEN (soup) is purchased on the same trip*" what additional info is needed? (choose 1 or more)

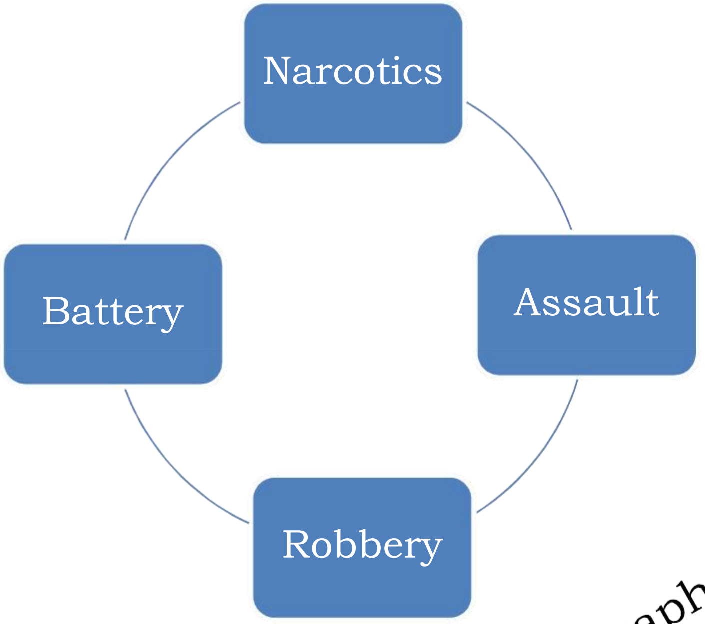
1. # transactions with only “soup”
2. # transactions of “soup” without orange juice or strepsils
3. # transactions with “soup & strepsils” or “soup & orange juice”
4. # transactions with soup alone or with anything else

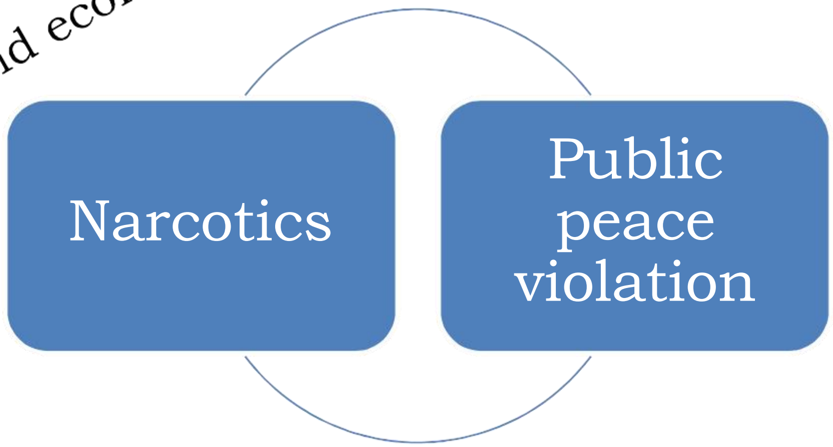
Other applications: Product-~~Product~~ Store association

- Identify rules to design product assortment in a store
  - Which product to stock in which store
  - E.g., which tea bags/coffee products to keep in the Café Coffee Day inside college campus vis-à-vis the one in a City Mall

Event-based databases

Association of Crimes at District Level





May reflect cultural, demographic, and economic differences across districts

Sequential Pattern Mining

Unlike,

- Purchases/events occur at the same time

More like,

- If many have taken “Big Data” in Kelly, have also enrolled for “Data Science” in Kelly.
- Recommend “Data Science” to those who have enrolled for “Big Data”.

Association Rules vs. Sequential Pattern mining

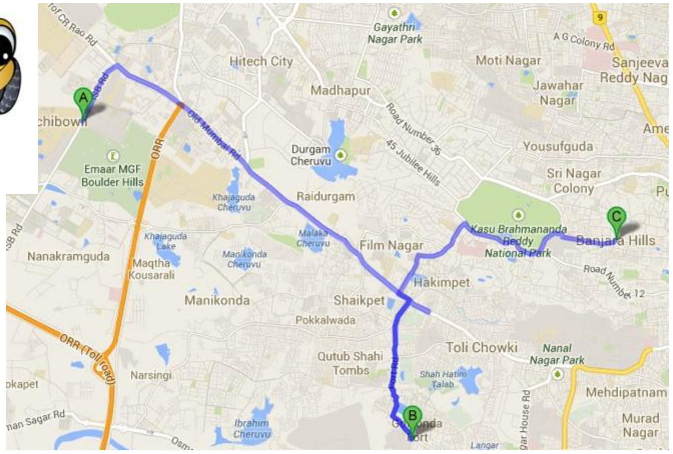
- Look for *temporal* patterns
- Order/sequence of *a* & *b* matters for a rule "*b follows a*"
- However, what happens in between *a* & *b* doesn't matter

- Recollect the dataset used for AR example
  - Association among items (bought within the same week) were discovered
  - How about finding what they would buy next week or the week after, if they had bought x in this week?

Example

- Identify popular taxi routes
  - Sequential pattern from GPS tracks—spatiotemporal records of taxi trajectories
  - How about, first identifying collocated customers using clustering techniques?
  - Routes:

    AB, AC, ABC, etc.



R Code

- install.packages("arules") ## if not already installed
- library("arules")
-
- mydata<-read.xlsx("Association Rules phone faceplates.xlsx",1)
- rules = apriori(as.matrix(mydata[,2:7]), parameter=list(support=.2, confidence=0.7,minlen=2)) ## the first column in mydata has transaction id
- inspect(head(sort(rules, by="lift")))