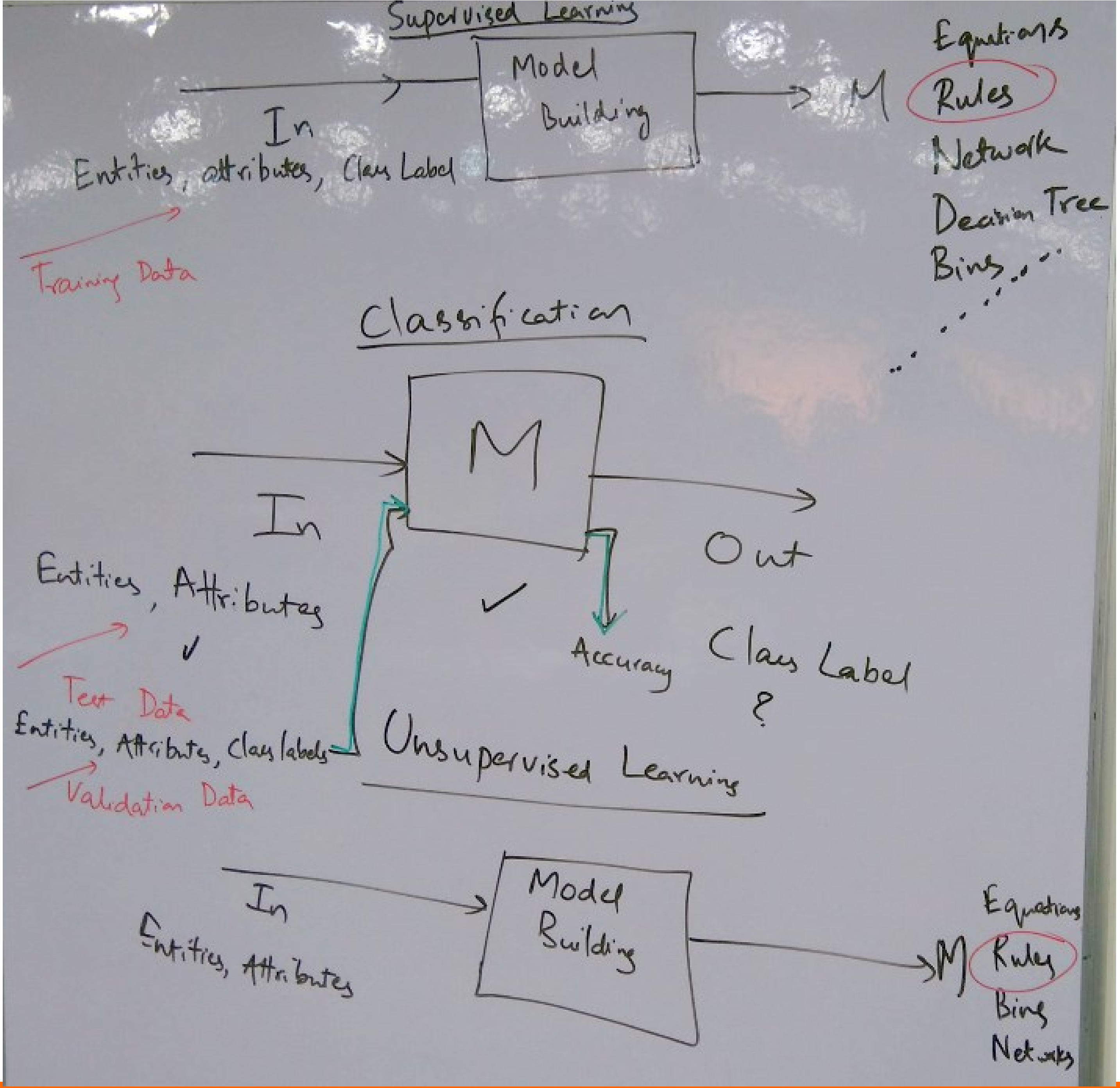
Inspire…Educate…Transform.

Methods and Algorithms in Machine Learning

Learning Rules from Data

Dr. Sreerama K. Murthy
CEO, Quadratyx

May 14, 2017


Supervised Learning
Model Building
In
Entities, attributes, Class Label
M
Equations
Rules
Network
Decision Tree
Bins, ...
Training Data
Classification
M
In
Entities, Attributes
Out
Accuracy
Class Label
?
Test Data
Entities, Attributes, Class labels
Validation Data
Unsupervised Learning
In
Entities, Attributes
Model Building
M
Equations
Rules
Bins
Networks

Rule-Based Classifier

- Classify records by using a collection of "if…then…" rules

- Rule: $(Condition) \rightarrow y$
  - where
    - *Condition* is a conjunctions of attributes
    - *y* is the class label
  - *LHS*: rule antecedent or condition
  - *RHS*: rule consequent
  - Examples of classification rules:

    (Blood Type=Warm) ∧ (Lay Eggs=Yes) → Birds

    (Taxable Income < 50K) ∧ (Refund=Yes) → Evade=No

Example

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Application of Rule-Based Classifier

- A rule *r* covers an instance **x** if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

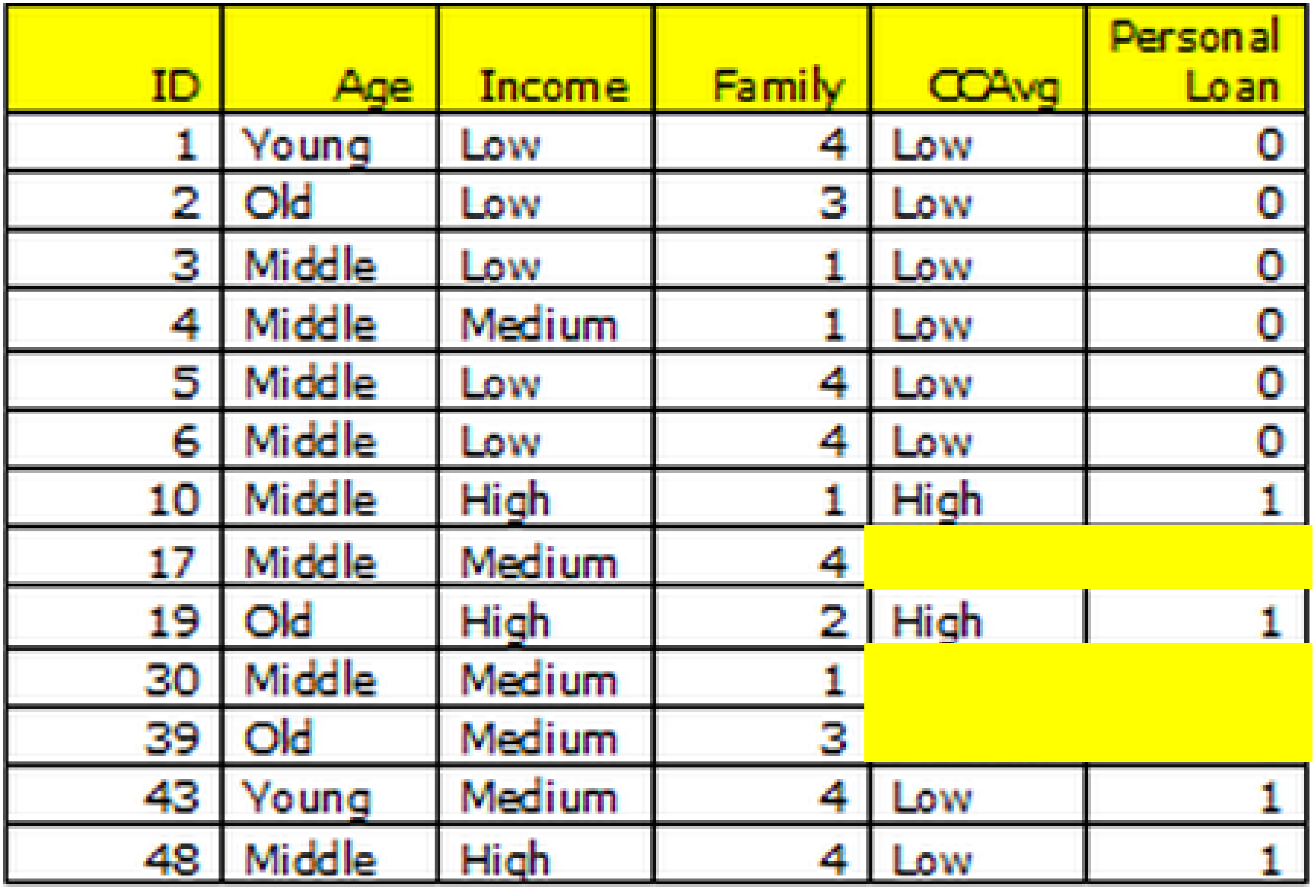
Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

GOODNESS METRICS FOR RULES

“if CCAvg is medium then loan = accept”



ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4		
19	Old	High	2	High	1
30	Middle	Medium	1		
39	Old	Medium	3		
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

3 in 13. 23% of the data is covered by this rule.

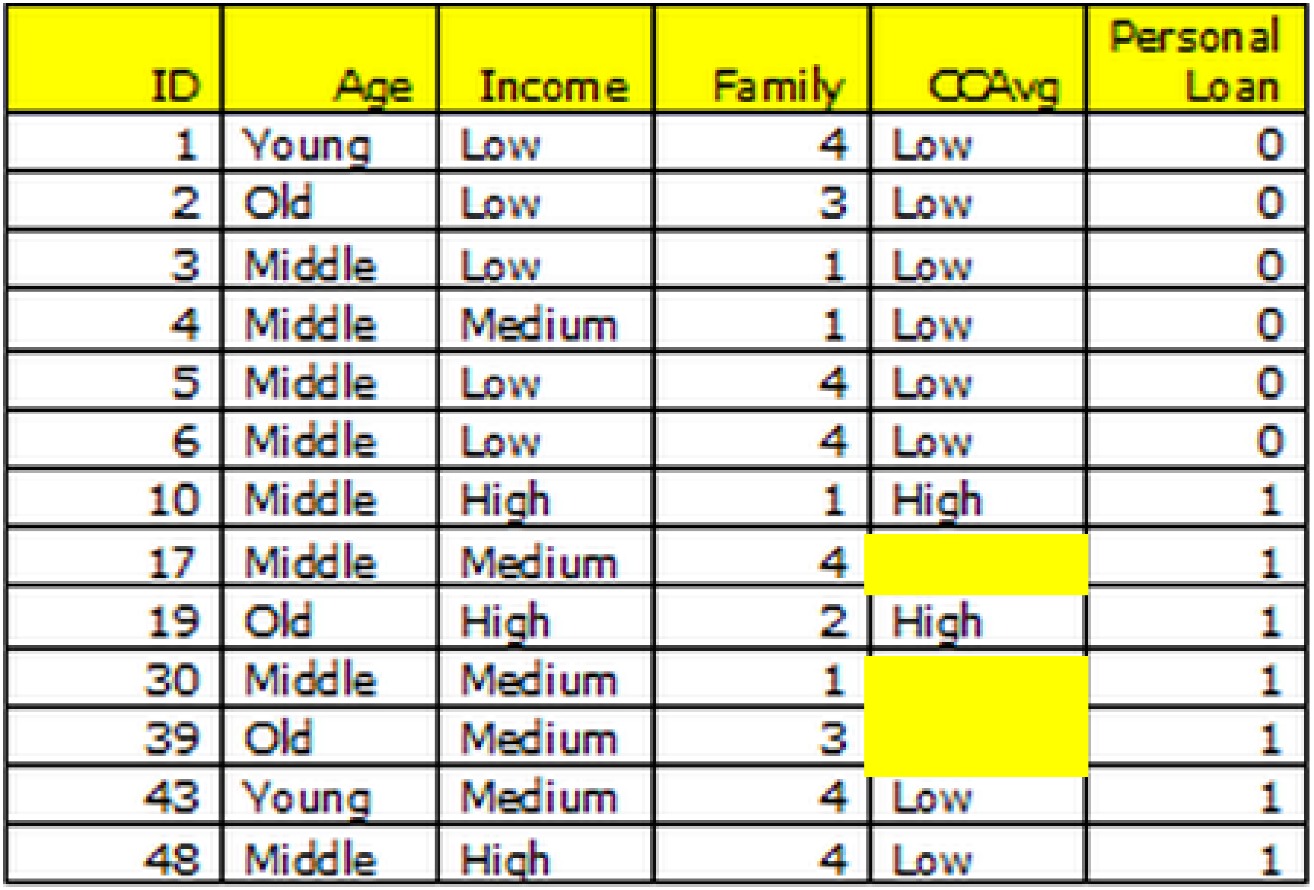
This is called **SUPPORT**.

Support is the % of cases in the data that contain both X and Y.

Recall P(X and Y).

* Adapted from Universal Bank data on predicting loan purchase likelihood of existing bank customers

“if CCAvg is medium then loan = accept”



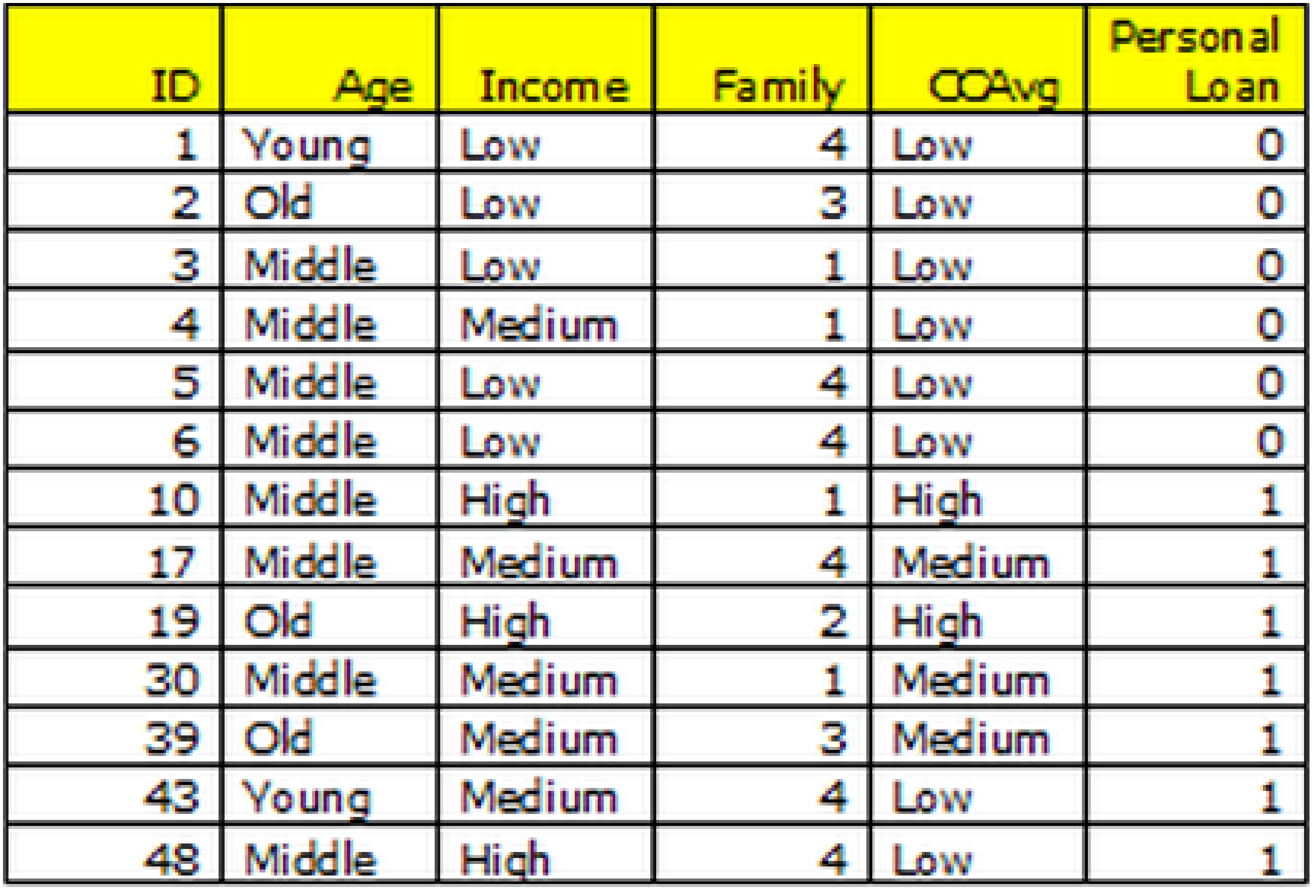
ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4		1
19	Old	High	2	High	1
30	Middle	Medium	1		1
39	Old	Medium	3		1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

Of the three occasions LHS is present, RHS too is present.

This rule has 100% **CONFIDENCE** / **ACCURACY**.

Confidence is the % of cases containing X that also contain Y. *Recall P(Y|X).*

“if loan = accept then CCAvg is medium”



ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

What are SUPPORT and CONFIDENCE now?

Support remains same at 3/13 (23%), but Confidence dips to 3/7 (43%).

Finding-1

- Support remains same if the antecedents and consequents of a rule are switched
- But confidence changes.

Rule Coverage and Accuracy

- **Coverage** of a rule:
  - Fraction of records that satisfy the antecedent of a rule
- **Accuracy** of a rule:
  - Fraction of records that satisfy both the antecedent and consequent of a rule (over those that satisfy the antecedent)

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	**Single**	125K	**No**
2	No	Married	100K	**No**
3	No	**Single**	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	**Single**	85K	**Yes**
9	No	Married	75K	**No**
10	No	**Single**	90K	**Yes**

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

Let us look at another case

- Let us say X is present in 25 transactions out of 100 transactions/ records. If Y is an obvious feature (e.g., “has mobile number”) present in all the records, what are the Support and Confidence of the rule “if x then y”?
  - Support = 25/100 = 25% (both items are present in a total of 25% of transactions)
  - Confidence = 25/25 =100% (if X is present then Y is always present).

Confidence is NOT Enough

- While this rule has very high confidence (100%), in reality, Y is present whether or not X is present.
- So, there is no REAL relationship between X and Y.

- Clearly, we need another metric to capture the goodness of a rule..

LIFT

- The *lift* of a rule *X* =>*Y* is defined as confidence of Y divided by the probability of Y.
  - ➢Lift = P(Y|X) / P(Y) = $\frac{P(X\ and\ Y)}{P(X)P(Y)}$

- In the earlier example, as Y is there in all transactions, P(Y)=1. So, LIFT =1.

One more example

- Total transactions 100. Y occurs in 20 where X also occurred (remember that X occurred 25 times?) and does not occur anywhere else. What are the Support, Confidence and Lift now?

Support of X->Y: 20/100 = 20% (where X and Y occur).

Confidence of X->Y: 20/25 = 80%.

Lift = 0.8/0.2 (confidence/probability of Y) = 4

One more example

- Y occurs in 20 records where X occurs and 70 transactions where X does not occur. Remember that X occurred total 25 times. What are the Support, Confidence and Lift now?

  – Support of X->Y: 20/100 = 20% (where X and Y occur).

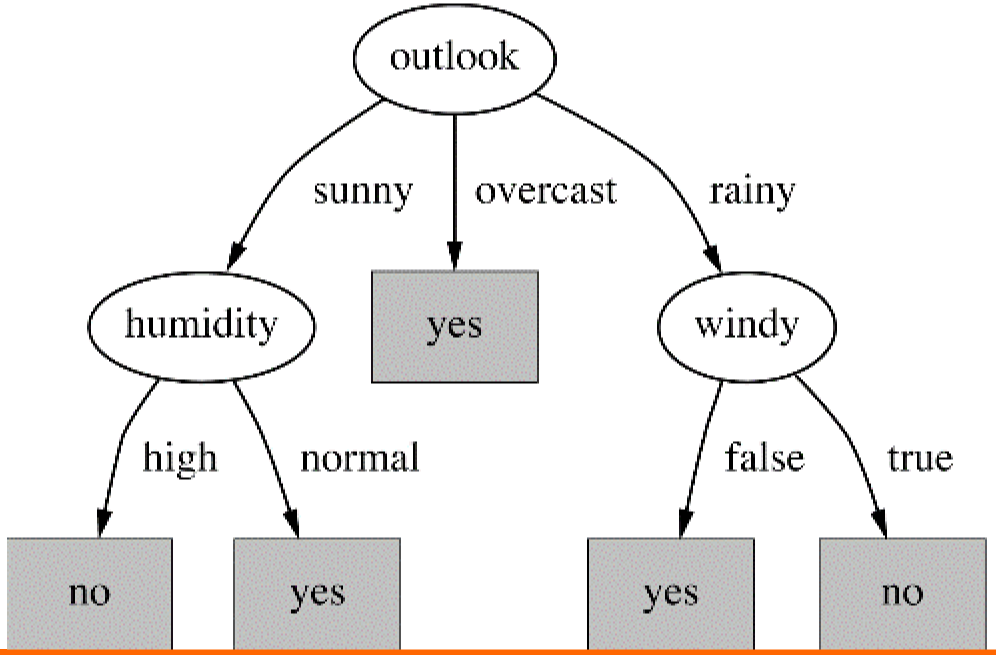
  – Confidence of X->Y: 20/25 = 80%

  – Lift = 0.8/0.9 (confidence/probability of Y) = 0.89

Decision Trees vs. rules

From trees to rules.

- Easy: converting a tree into a set of rules
  - One rule for each leaf:
  - Antecedent contains a condition for every node on the path from the root to the leaf
  - Consequent is the class assigned by the leaf
  - Straightforward, but rule set might be overly complex



Decision Trees vs. rules

From rules to trees

- More difficult: transforming a rule set into a tree
  - Tree cannot easily express disjunction between rules
- Example:

  If a and b then x

  If c and d then x

  - Corresponding tree contains identical subtrees (⇒“replicated subtree problem”)

How does Rule-based Classifier Work?

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) → Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) → Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Desiderata for Rule-Based Classifier

- Mutually exclusive rules
  - No two rules are triggered by the same record.
  - This ensures that every record is covered by **at most** one rule.

- Exhaustive rules
  - There exists a rule for each combination of attribute values.
  - This ensures that every record is covered by **at least** one rule.

Together these properties ensure that every record is covered by exactly one rule.

Rules

- Non mutually exclusive rules
  - A record may trigger more than one rule
  - Solution?
    - Ordered rule set
- Non exhaustive rules
  - A record may not trigger any rules
  - Solution?
    - Use a default class

Ordered Rule Set

- Rules are ranked ordered according to their priority (e.g. based on their quality)
  - An ordered rule set is known as a **decision list**
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule it has triggered
  - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

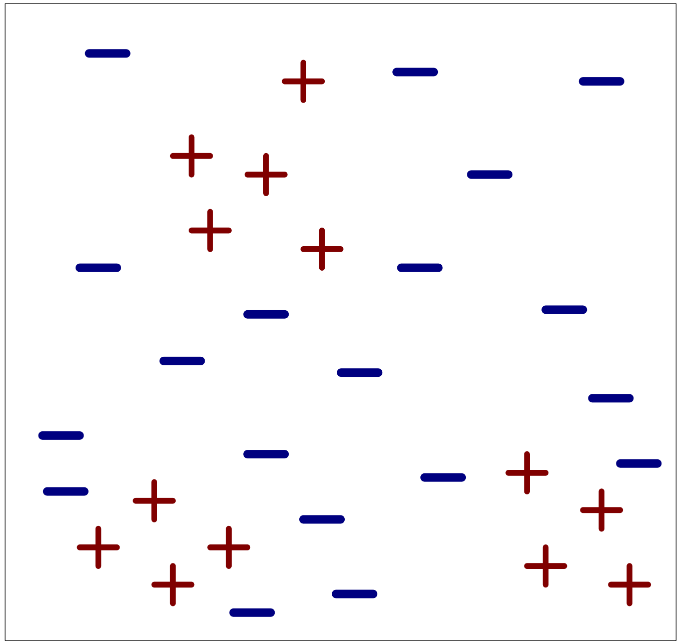
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

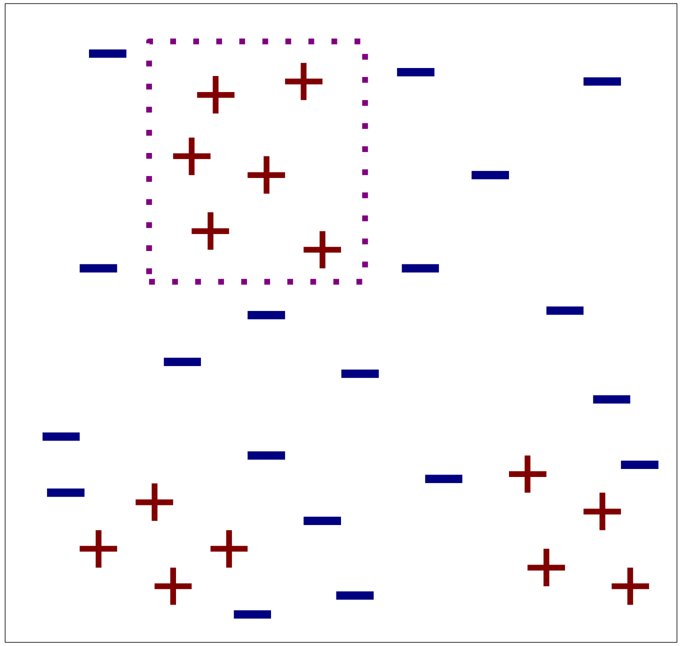
Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Building Classification Rules: Sequential Covering

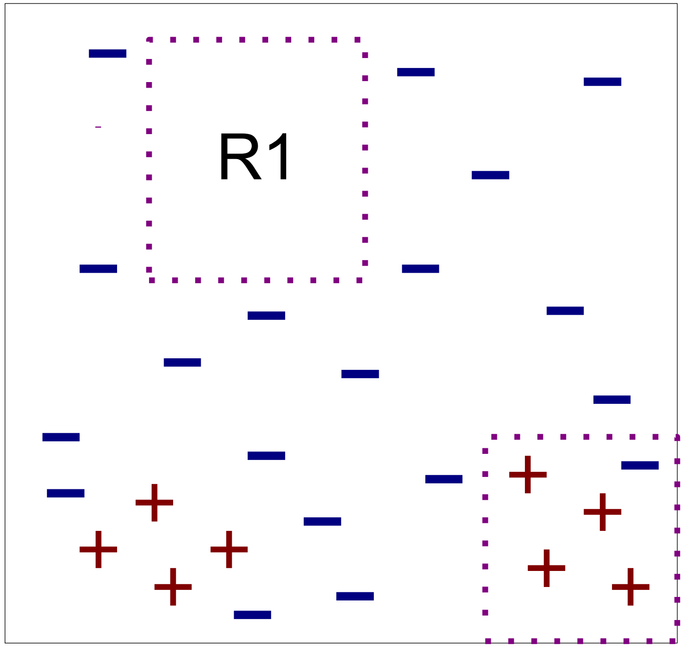
1. Start from an empty rule
2. Grow a rule using some **Learn-One-Rule** function
3. Remove training records **covered** by the rule
4. Repeat Step (2) and (3) until stopping criterion is met



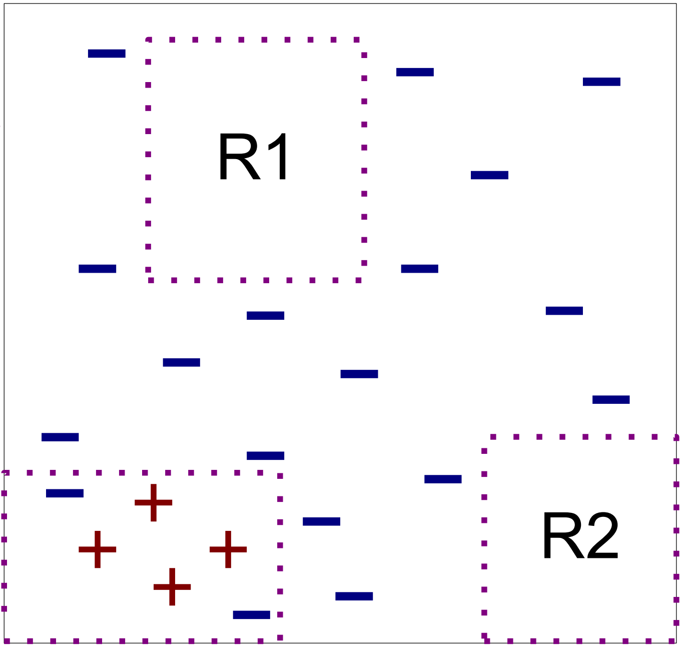
(i) Original Data



(ii) Step 1



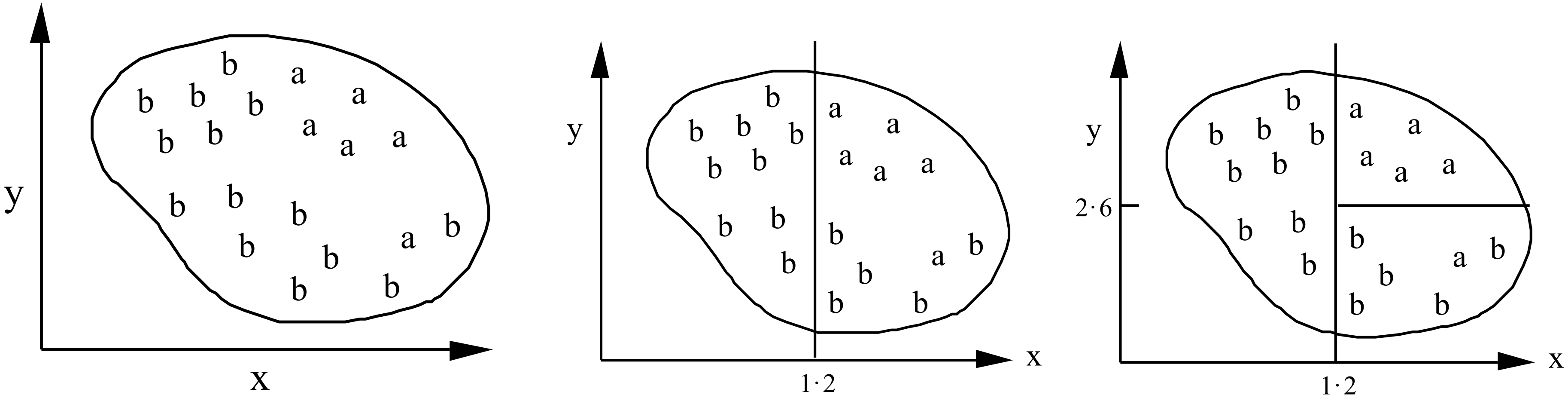
(iii) Step 2



(iv) Step 3

- This approach is called a **covering** approach because at each stage a rule is identified that covers some of the instances

Example: generating a rule



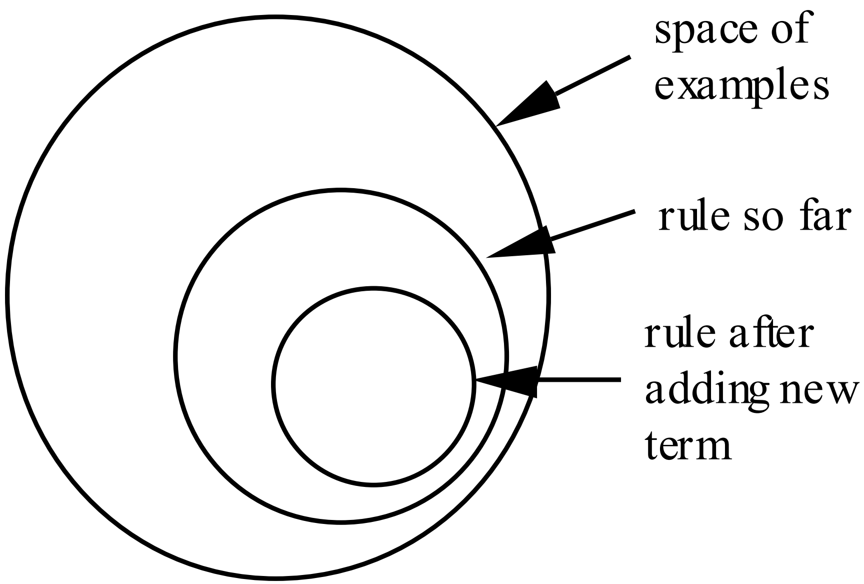
- Possible rule set for class “b”:
- More rules could be added for “perfect” rule set

If x ≤ 1.2 then class = b

If x > 1.2 and y ≤ 2.6 then class = b

A simple covering algorithm

- Generates a rule by adding tests that maximize rule’s accuracy
- Similar to situation in decision trees: problem of selecting an attribute to split on.
  - But: decision tree inducer maximizes overall purity
- Here, each new test (growing the rule) reduces rule’s coverage.



Selecting a test

- Goal: maximizing accuracy
  - **t**: total number of instances covered by rule
  - **p**: positive examples of the class covered by rule
  - **t-p**: number of errors made by rule

⇒ Select test that maximizes the ratio **p/t**

Example: contact lenses data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended Lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

Example: contact lenses data

- Rule we seek:

```
If ?
    then recommendation = hard
```

- Possible tests:

Age = Young	2/8
Age = Pre-presbyopic	1/8
Age = Presbyopic	1/8
Spectacle prescription = Myope	3/12
Spectacle prescription = Hypermetrope	1/12
Astigmatism = no	0/12
Astigmatism = yes	4/12
Tear production rate = Reduced	0/12
Tear production rate = Normal	4/12

The numbers on the right show the fraction of “correct” instances in the set singled out by that choice.
In this case, correct means that their recommendation is “hard.”

Modified rule and resulting data

- Rule with best test added:

```
If astigmatism = yes
    then recommendation = hard
```

- Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

The rule isn't very accurate, getting only 4 out of 12 that it covers. So, it needs further refinement.

Further refinement

- Current state:

```
If astigmatism = yes
     and ?
   then recommendation = hard
```

- Possible tests:

Age = Young	2/4
Age = Pre-presbyopic	1/4
Age = Presbyopic	1/4
Spectacle prescription = Myope	3/6
Spectacle prescription = Hypermetrope	1/6
Tear production rate = Reduced	0/6
Tear production rate = Normal	4/6

Modified rule and resulting data

- Rule with best test added:

```
If astigmatism = yes
    and tear production rate =
normal
  then recommendation = hard
```

- Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

Should we stop here? Perhaps. But let’s say we are going for exact rules, no matter how complex they become.

So, let’s refine further.

Further refinement

- Current state:

```
If astigmatism = yes
     and tear production rate = normal
     and ?
   then recommendation = hard
```

- Possible tests:

Age = Young	2/2
Age = Pre-presbyopic	1/2
Age = Presbyopic	1/2
Spectacle prescription = Myope	3/3
Spectacle prescription = Hypermetrope	1/3

- Tie between the first and the fourth test
  - ❑ We choose the one with greater coverage

The result

- Final rule:

```
If astigmatism = yes
   and tear production rate = normal
   and spectacle prescription = myope
   then recommendation = hard
```

- Second rule for recommending “hard lenses”: (built from instances not covered by first rule)

```
If age = young and astigmatism = yes
   and tear production rate = normal
   then recommendation = hard
```

- These two rules cover all “hard lenses”:
  - ❑ Process is repeated with other two classes

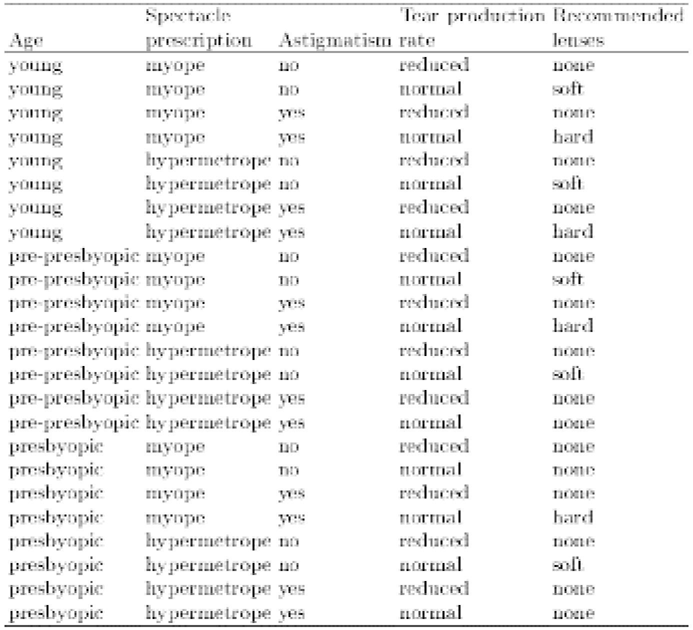
Other Rules from the Contact Lens Dataset

```
IF    TearProduction = reduced
THEN  ContactLenses = none        [#soft=0 #hard=0 #none=12]

IF    TearProduction = normal
  AND Astigmatism = no
THEN  ContactLenses = soft        [#soft=5 #hard=0 #none=1]

IF    TearProduction = normal
  AND Astigmatism = yes
  AND SpectaclePrescription = myope
THEN  ContactLenses = hard        [#soft=0 #hard=3 #none=0]

IF    TearProduction = normal
  AND Astigmatism = yes
  AND SpectaclePrescription = hypermetrope
THEN  ContactLenses = none        [#soft=0 #hard=1 #none=2]
```



Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

Pseudo-code for PRISM

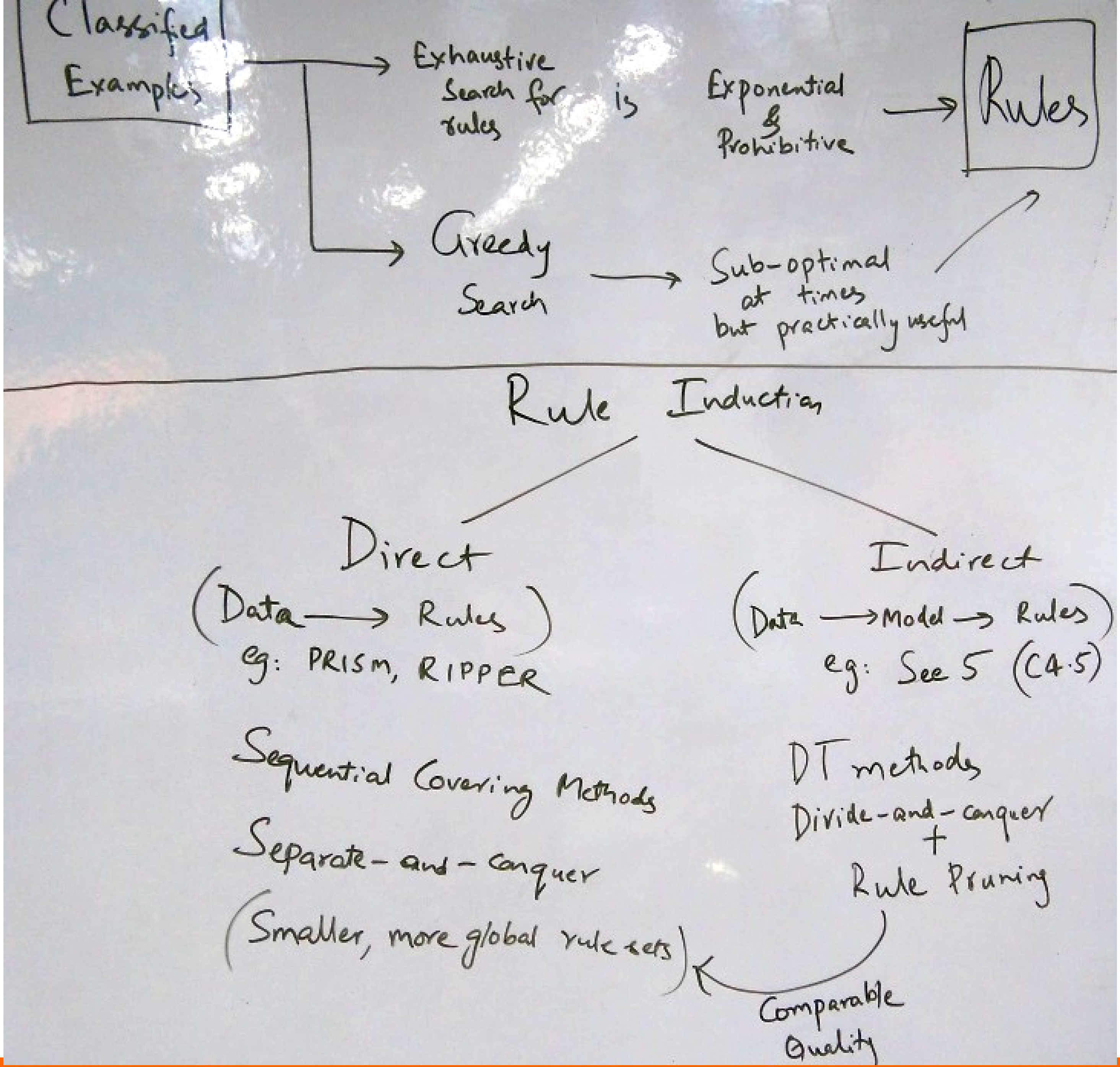
Heuristic: order C in ascending order of occurrence.

```
For each class C
    Initialize E to the instance set
    While E contains instances in class C
    Create a rule R with an empty left-hand side that
        predicts class C
    Until R is perfect (or there are no more attributes to
        use) do
        For each attribute A not mentioned in R, and each value v,
            Consider adding the condition A = v to the left-hand side of R
            Select A and v to maximize the accuracy p/t
            (break ties by choosing the condition with the largest p)
        Add A = v to R
    Remove the instances covered by R from E
```

RIPPER Algorithm is similar. It uses instead of p/t the info gain.

Separate and conquer

- Methods like PRISM (for dealing with one class) are **separate-and-conquer** algorithms:
  - First, a rule is identified
  - Then, all instances covered by the rule are separated out
  - Finally, the remaining instances are “conquered”
- Difference to divide-and-conquer methods:
  - Subset covered by rule doesn’t need to be explored any further



Classified Examples → Exhaustive Search for rules is Exponential & Prohibitive → Rules

Classified Examples → Greedy Search → Sub-optimal at times but practically useful → Rules

Rule Induction

Direct (Data → Rules)
eg: PRISM, RIPPER

Indirect (Data → Model → Rules)
eg: See 5 (C4.5)

Sequential Covering Methods
Separate-and-conquer
(Smaller, more global rule sets)

DT methods
Divide-and-conquer
+
Rule Pruning

Comparable Quality

METRICS TO FILTER RULES – THE BUSINESS PERSPECTIVE

Defining Minimum Support

- Rules with a certain minimum support only may be interesting.
- Example?
- But, how do we know *how much* minimum support?
    - Domain expertise
    - Subjective

Identifying Trivial Rules

- A short rule with high support and confidence

$$\frac{Support+Confidence}{Length}$$

  - Can you think of an example?
  - Also, always check with the business user

Actionability

- *If (the mother is B positive) and (she smoked during pregnancy) and (the child is eating a lot of carbohydrates), then the child is likely to get asthma*

- Are the actionabilities of the three features different?

Universal Bank Attributes For Loan Decisions – Actionability

- **Non-actionable**: Acts of God (weather), external factors (price of gold, rupee value, etc.)
- **Moderately Actionable**: Age, experience, income, family, education
- **Highly Actionable**: infoReq (information requested by Phone or Email)

Actionability of a Rule

- $Actionability = \dfrac{\sum Actionability\ of\ antecedents}{Total\ number\ of\ attributes\ in\ the\ antecedent}$

- Why not take the numerator alone?
  - distinguishing between a long rule and a short rule

- Subjective

Explicability

- More precedents, less explicable

Cost of a Rule

- Cost of a rule is the sum of the costs of collection of each attribute

Tuning a Rule

- **Generalizing:** removing an attribute
- **Specifizing:** Adding an attribute

- What does it do to support?
- What does it do to confidence?

Propositional Rule Induction

- Produces compact/understandable knowledge
- Can find every possible pattern and hence overfit
- Slow to induce
- Separate-and-conquer process

Other Rule Learning Approaches

- Translate decision trees into rules (C4.5)
- Sequential (set) covering algorithms
  - General-to-specific (top-down) (CN2, FOIL)
  - Specific-to-general (bottom-up) (GOLEM, CIGOL)
  - Hybrid search (AQ, Chillin, Progol)
- Translate neural-nets into rules (TREPAN)
- Inductive Logic Programming: first-order, higher-order
- Beam search

Unsupervised rule induction

ASSOCIATION RULES – AFFINITY ANALYSIS / MARKET BASKET ANALYSIS

- Popularized by the 1993 paper* by Agrawal *et al.* on finding regularities based on POS transactions in supermarkets.

- Market basket analysis **doesn't refer** to a single technique.

- Useful for cross-selling, up-selling, influencing sales promotions, loyalty programs, store layouts, discount plans, intrusion detection, bioinformatics, and many more applications.

**Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207.*

CAN WE REALLY GET INSIGHTS FROM MARKET BASKETS

It is not over yet

- Most likely he/she is a vegetarian!

- He/she has been exposed to some overseas culture (how many Indians eat pickles, not the Indian pickles!)

Market Basket Analysis

- Provides insight into **which products** tend to be **purchased together** and which are most amenable to **promotion**.

- The findings were that men between 30- 40 years in age, shopping between 5PM and 7PM on Fridays, who purchased diapers, were most likely to also have beer in their carts.

Market Basket Analysis

- Suppose the POS system has the following data:
  - Total transactions = 600,000
  - Transactions containing diapers = 7,500 (1.25%)
  - Transactions containing beer – 60,000 (10%)
  - Transactions containing both beer and diapers = 6,000 (1%)
- Assuming (null hypothesis) that beer and diaper purchases are independent (no association), and knowing that 10% of all transactions contain beer, 10% of the transactions containing diapers should be EXPECTED to contain beer.

Market Basket Analysis

- 10% of 7500 = 750. However, 6000 transactions containing diapers contained beer, which is an 8-fold **increase over the expected value**. So, the LIFT is 8.

$$\text{Recall, } Lift = \frac{P(X\ and\ Y)}{P(X)P(Y)} = \frac{0.01}{0.0125*0.1} = 8$$

Market Basket can give Rules that are

- Seemingly interesting, actually not
  - People buying conference calling facility also buy call forwarding service
- Trivial
  - People who buy shoes also buy socks
- Inexplicable
  - People who buy shirts also buy milk
- Actionable
  - People who bought Dove soap also bought Barbie doll. What should the business do?
  - Target store case

Market Basket can lead to interesting discoveries

Wal-Mart in Florida found in 2004 that **strawberry pop tart** sales before a hurricane had a lift of 7 over normal shopping days.

A major electronics store used association rule mining to find that customers who bought VHS players/recorders tended to return 3-4 months later to buy camcorders. They use discount coupons to successfully engage such customers.

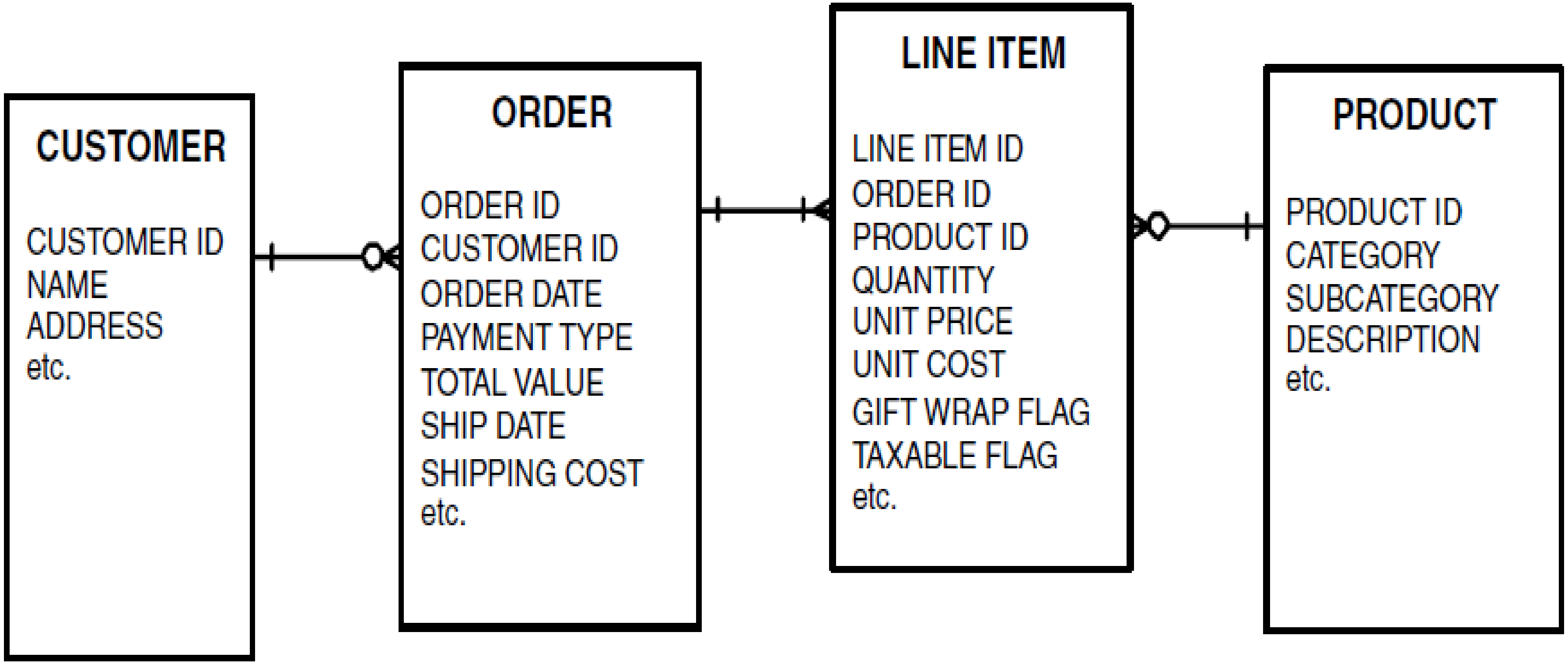
It is not just for retail and baskets

- Unusual combinations of **insurance claims** can be a sign of fraud and can spark further investigation.

- Medical patient histories can give indications of likely complications based on certain **combinations** of **treatments**.

- Supervised or Unsupervised?

Let us hand work some rules



CUSTOMER

CUSTOMER ID
NAME
ADDRESS
etc.

ORDER

ORDER ID
CUSTOMER ID
ORDER DATE
PAYMENT TYPE
TOTAL VALUE
SHIP DATE
SHIPPING COST
etc.

LINE ITEM

LINE ITEM ID
ORDER ID
PRODUCT ID
QUANTITY
UNIT PRICE
UNIT COST
GIFT WRAP FLAG
TAXABLE FLAG
etc.

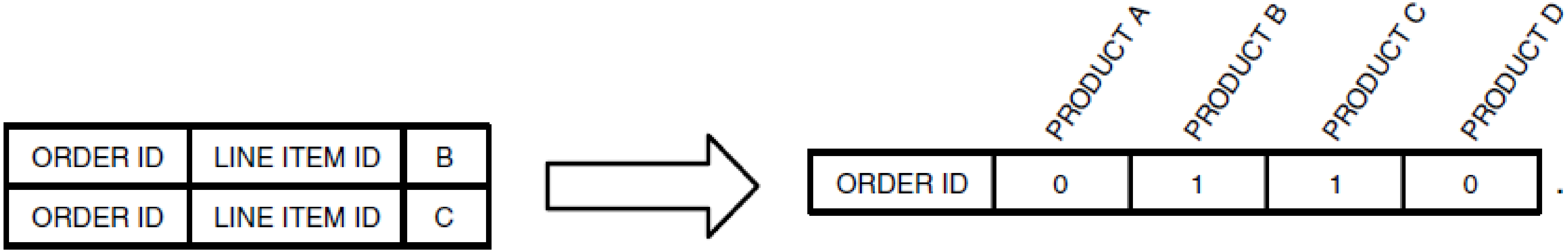
PRODUCT

PRODUCT ID
CATEGORY
SUBCATEGORY
DESCRIPTION
etc.

Some Straightforward Market Basket Insights – Descriptive Analytics

- What is the average number of orders per customer?
- What is the most common item found in a one-item order?
- What is the average number of unique items per order?
- What is the average number of items per order?

The Process – Step 1: Transform the data

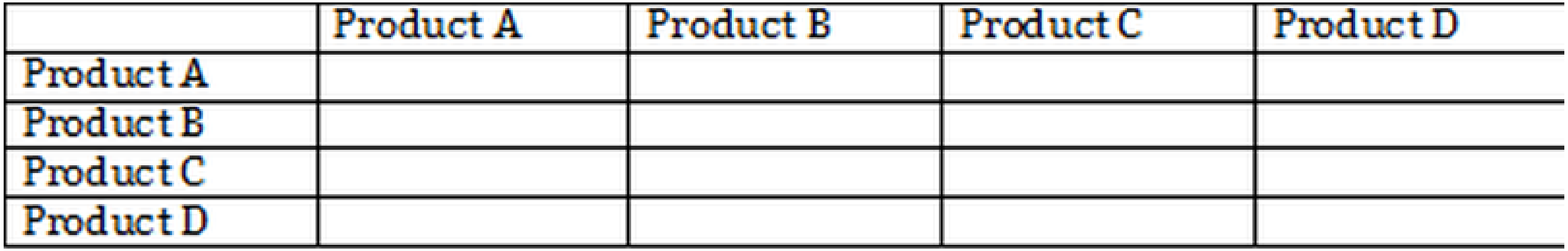


ORDER ID	LINE ITEM ID	B
ORDER ID	LINE ITEM ID	C

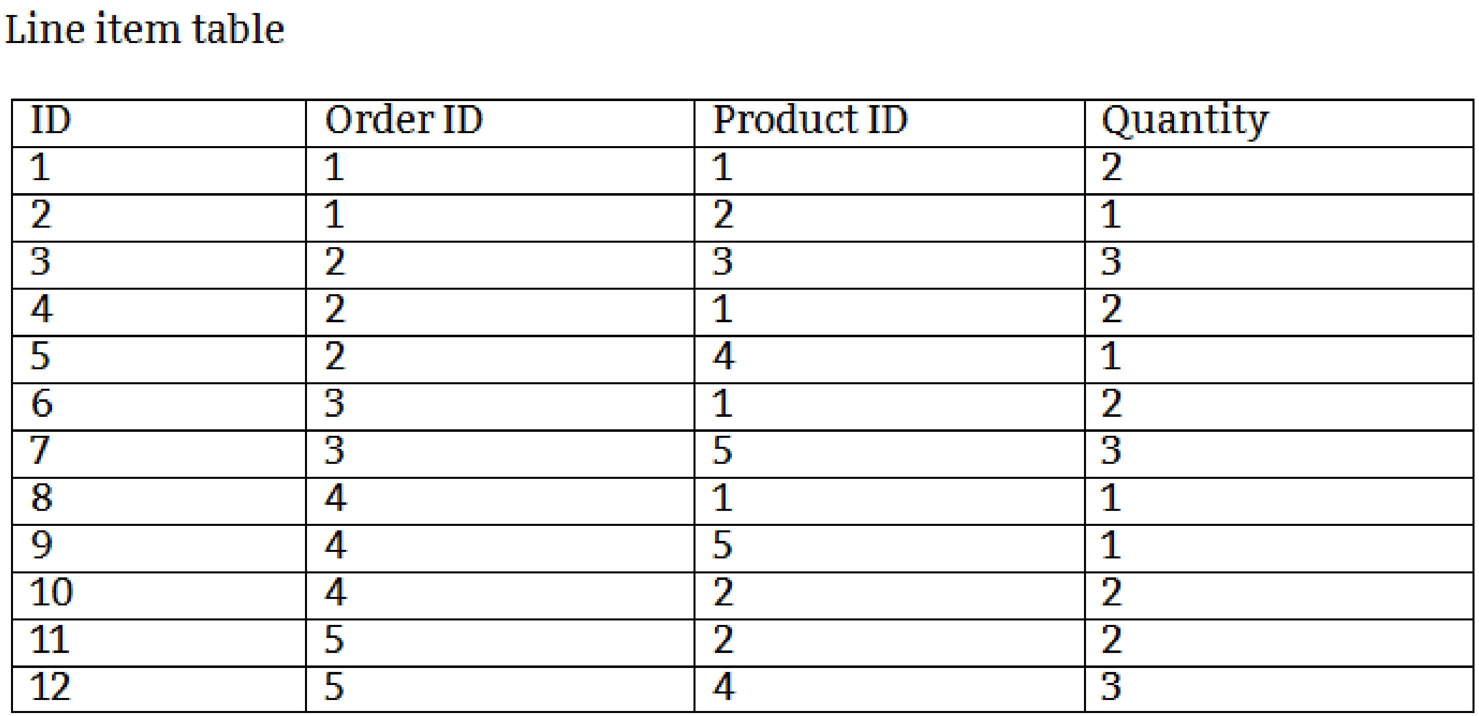
	PRODUCT A	PRODUCT B	PRODUCT C	PRODUCT D
ORDER ID	0	1	1	0

.

The Process – Step 2: Co-occurrence table



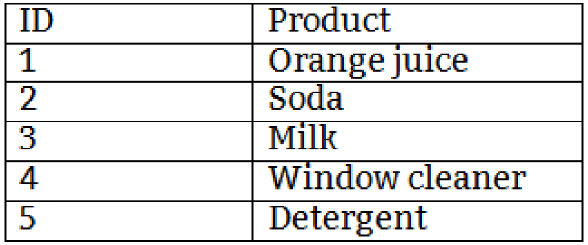
	Product A	Product B	Product C	Product D
Product A				
Product B				
Product C				
Product D				



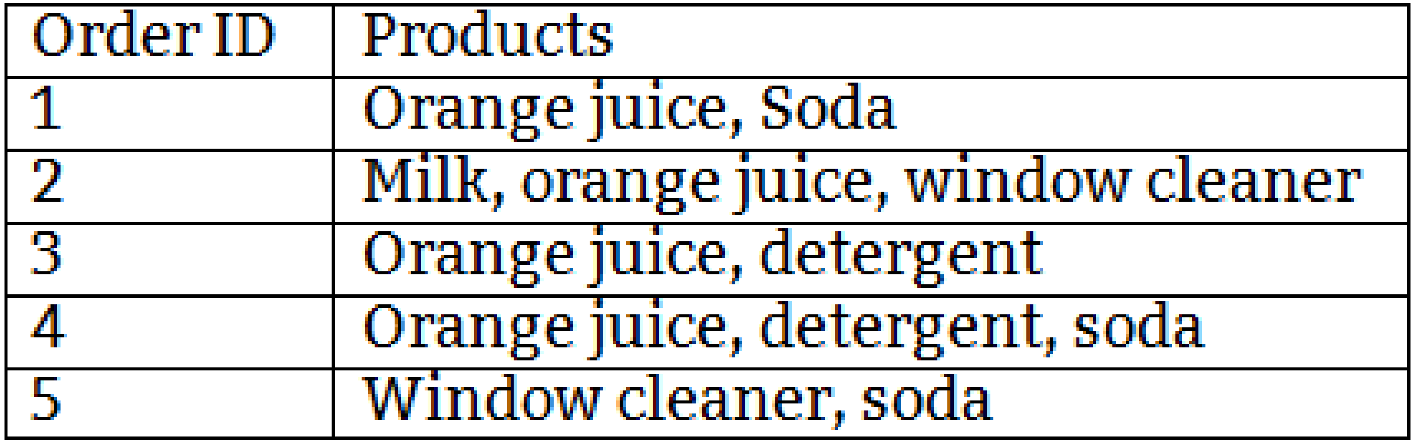
Line item table

ID	Order ID	Product ID	Quantity
1	1	1	2
2	1	2	1
3	2	3	3
4	2	1	2
5	2	4	1
6	3	1	2
7	3	5	3
8	4	1	1
9	4	5	1
10	4	2	2
11	5	2	2
12	5	4	3

Product table



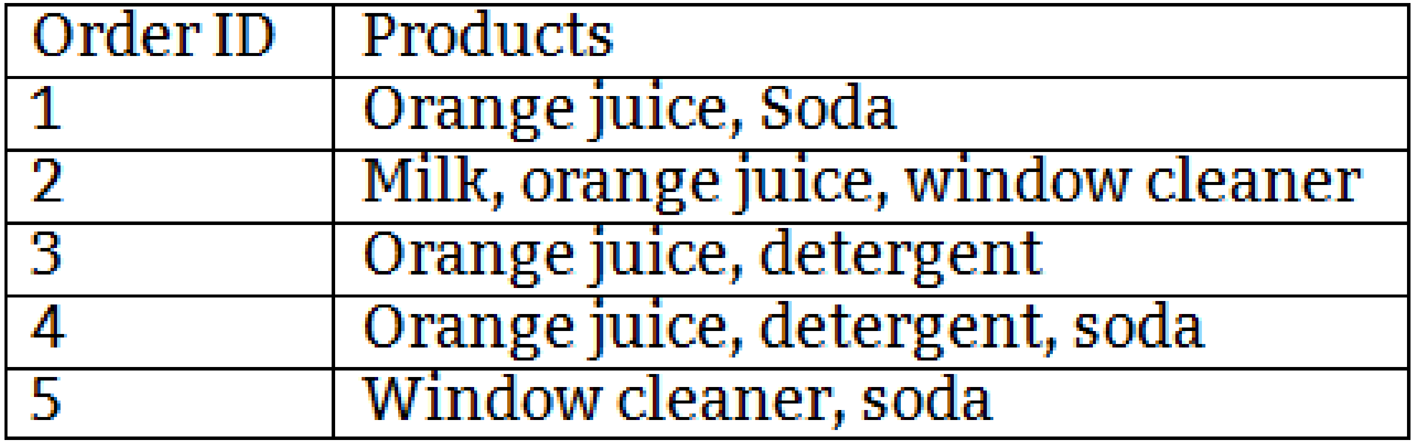
ID	Product
1	Orange juice
2	Soda
3	Milk
4	Window cleaner
5	Detergent



Order ID	Products
1	Orange juice, Soda
2	Milk, orange juice, window cleaner
3	Orange juice, detergent
4	Orange juice, detergent, soda
5	Window cleaner, soda

Co-occurrence Table

Product	OJ	Window Cleaner	Milk	Soda	Detergent
OJ	4	1	1	2	2
Window cleaner	1	2	1	1	0
Milk	1	1	1	0	0
Soda	2	1	0	3	1
Detergent	2	0	0	1	2



Order ID	Products
1	Orange juice, Soda
2	Milk, orange juice, window cleaner
3	Orange juice, detergent
4	Orange juice, detergent, soda
5	Window cleaner, soda

Insights

Product	OJ	Window Cleaner	Milk	Soda	Detergent
OJ	4	1	1	2	2
Window cleaner	1	2	1	1	0
Milk	1	1	1	0	0
Soda	2	1	0	3	1
Detergent	2	0	0	1	2

- Orange juice and soda are more likely to be purchased together than any other two items.
- Detergent is never purchased with window cleaner or milk.
- Milk is never purchased with soda or detergent.

Important considerations in building rules

- Choosing the right set of items
- The co-occurrence tables can be huge
  - Overcoming the practical limits imposed by thousands or tens of thousands of items

Summary: Best Use of Market Basket analysis

- Best results obtained when items occur in roughly the same number of transactions to prevent common items from dominating the rules
- Use hierarchies to roll up rare items to more general items, and leave more common items as is
- Data quality is extremely important for Association Rules

APRIORI ALGORITHM

Association Rules

- There are a large number of association rules algorithms.
- They all use different strategies and data structures.
- We will study the Apriori algorithm and its variants as they are the widely used.

Other Algorithms

- Eclat algorithm
- FP-growth algorithm
- AprioriDP
- Context Based Association Rule Mining Algorithm
- Node-set-based algorithm
- GUHA procedure ASSOC
- OPUS search

http://serialsjournals.com/serialjournalmanager/pdf/1482729114.pdf

Two-step process

- Find all itemsets that have **minimum support**. These are called ***frequent itemsets***.

- Use ***frequent itemsets*** to generate **rules**.

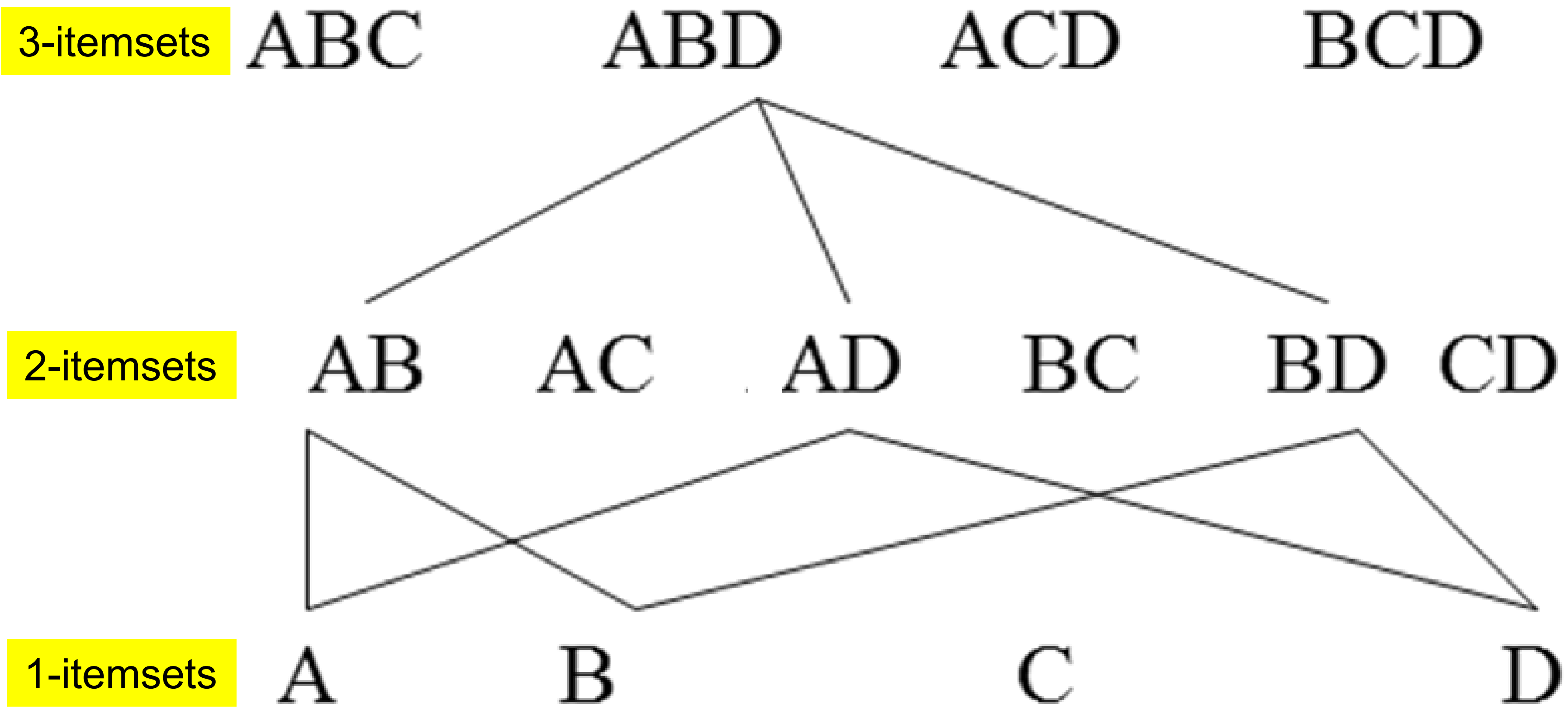
Apriori Property

- The **key idea** behind the algorithm is called the **apriori** property or **downward closure** property.
- Downward closure property: Any **subset** of a frequent itemset is **also** a frequent itemset.

Closed

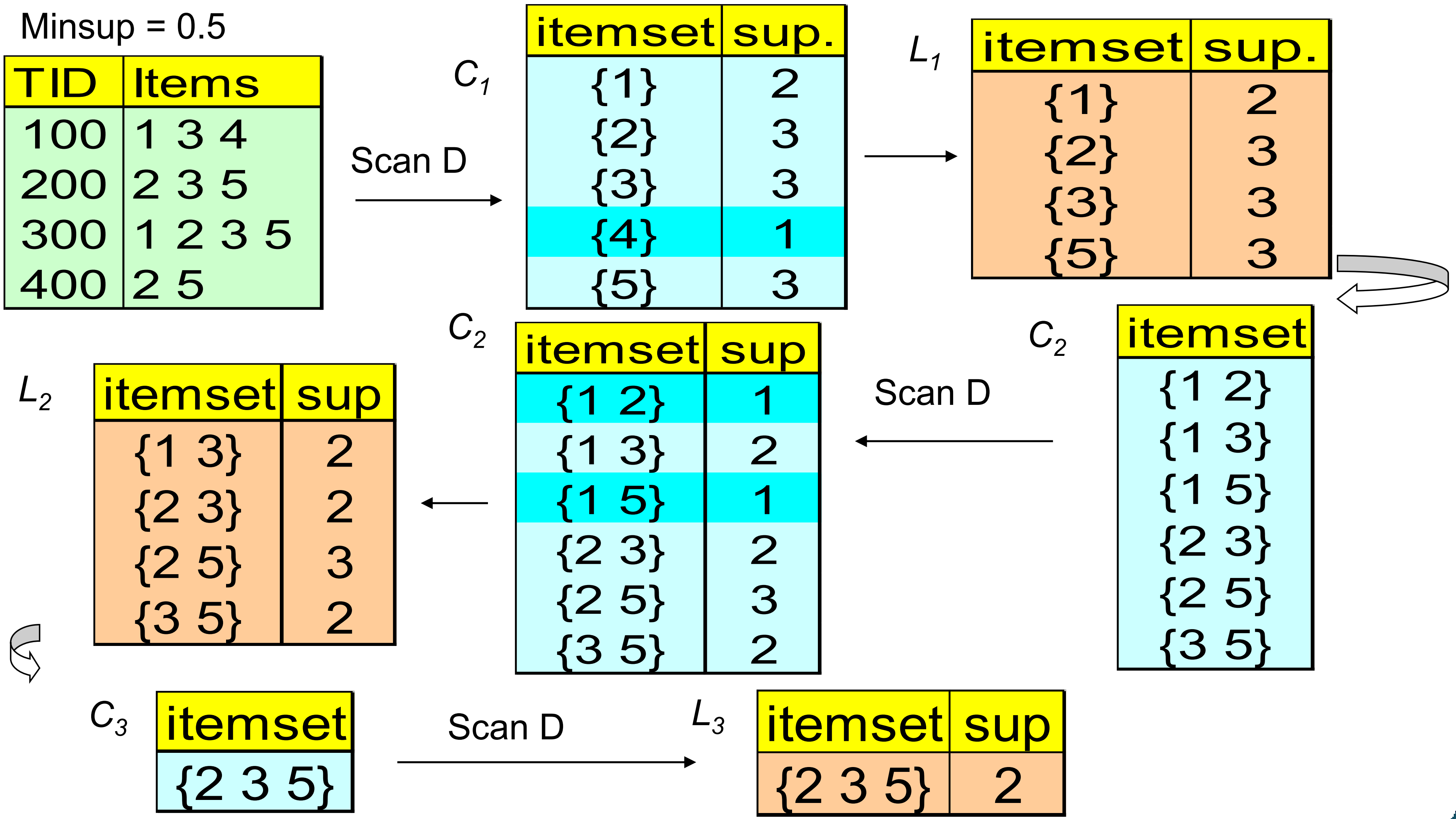
- A **set** is said to be closed under an **operation**, if the **operation** produces **another member** of the **set**.

Downward closure



- Suppose {A,B} is frequent. Since each occurrence of A, B includes both A and B, then both A and B must also be frequent
- Similar argument for larger item sets
- So, if a *k*-item set is frequent all its subsets (*k-1*, *k-2* itemsets) are also frequent

The Apriori Algorithm — Example



Database D

Minsup = 0.5

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Finalizing Rules from Apriori

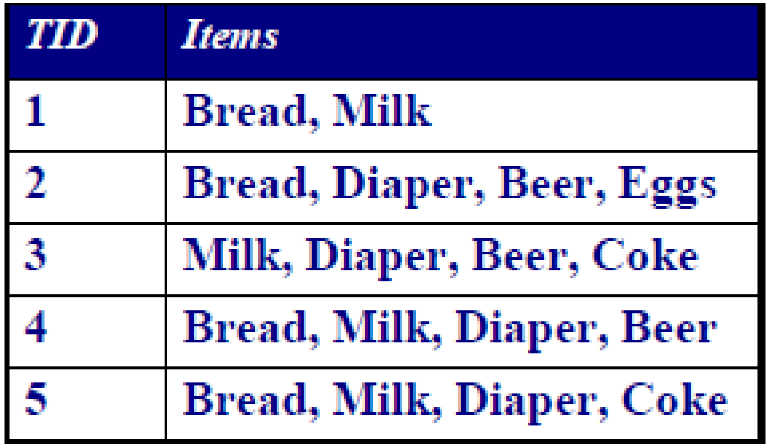
Calculate confidence for each of the finalized *k*-itemsets, formulate rules and finalize those meeting the minimum confidence required

- Example itemset

  {Milk, Diaper, Beer}

- Rules (Calculate Support and Confidence for each of the below)

{Milk, Diaper} → {Beer}	s = 0.4, c = 0.67
{Milk, Beer} → {Diaper}	s = 0.4, c = 1.00
{Diaper, Beer} → {Milk}	s = 0.4, c = 0.67
{Beer} → {Milk, Diaper}	s = 0.4, c = 0.67
{Diaper} → {Milk, Beer}	s = 0.4, c = 0.50
{Milk} → {Diaper, Beer}	s = 0.4, c = 0.50



TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

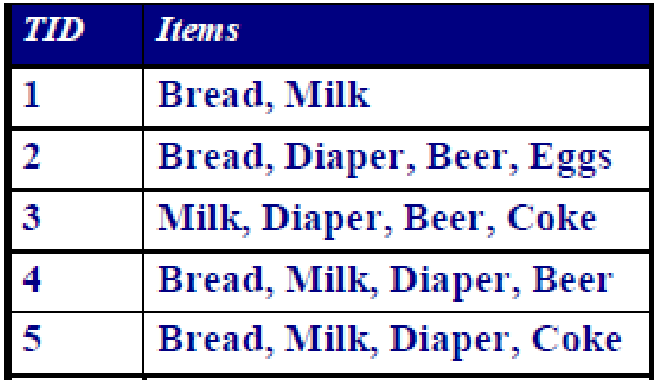
- Observation: Rules originating from the same itemset have identical support but can have different confidence *(once again recall joint and conditional probabilities)*

Apriori Recap

Definition

- An expression of the form X → Y is a rule, where X and Y form the itemset
- X is the rule's antecedent and Y is the rule's consequent

Example: $\{Milk, Diaper\} \Rightarrow Beer$



TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Rule Evaluation Metrics

- Support (s)
  - Fraction of transactions that contain both X and Y

$$s = \frac{\sum(Milk, Diaper, Beer)}{|T|} = \frac{2}{5} = 0.4$$

- Confidence (c)
  - Measures how often Y appears in transactions that contain X

$$c = \frac{\sum(Milk, Diaper, Beer)}{\sum(Milk, Diaper)} = \frac{2}{3} = 0.67$$

Step 2 of Apriori: Generating rules from frequent itemsets

- Frequent itemsets ≠ association rules. One more step is needed to generate association rules
- For each frequent itemset *X*,

  For each proper nonempty subset *A* of *X*,
  - Let $B = X - A$
  - $A \rightarrow B$ is an association rule if
    - Confidence($A \rightarrow B$) $\geq$ minconf,

      support($A \rightarrow B$) = support($A \cup B$) = support(X)

      confidence($A \rightarrow B$) = support($A \cup B$) / support(A)

Generating rules: an example

- Suppose {2,3,4} is frequent, with sup=50%
  - Proper nonempty subsets: {2,3}, {2,4}, {3,4}, {2}, {3}, {4}, with sup=50%, 50%, 75%, 75%, 75%, 75% respectively
  - These generate the following association rules:
    - 2,3 → 4, confidence=100%
    - 2,4 → 3, confidence=100%
    - 3,4 → 2, confidence=67%
    - 2 → 3,4, confidence=67%
    - 3 → 2,4, confidence=67%
    - 4 → 2,3, confidence=67%
    - All rules have support = 50%

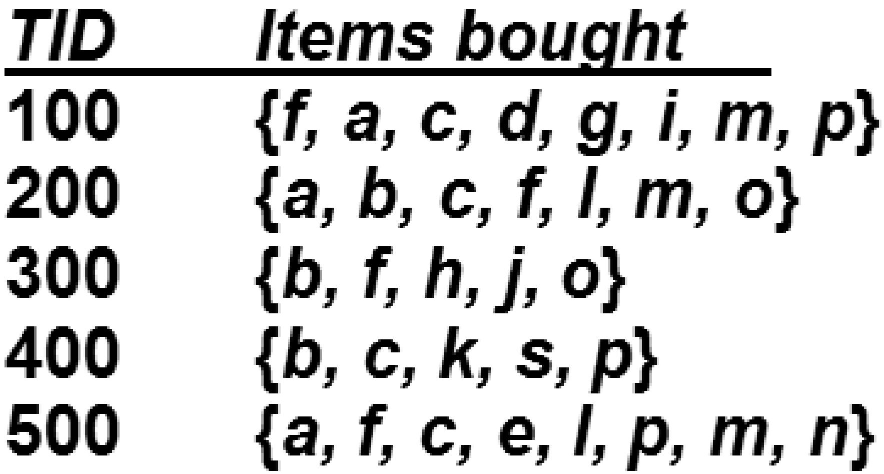
Limitations

- *Apriori* algorithm can be very slow and the bottleneck is candidate generation.
  - For example, if the transaction DB has 10^4 frequent 1-itemsets, they will generate 10^7 candidate 2-itemsets even after employing the downward closure.
  - To compute those with sup more than minsup, the database need to be scanned at every level. It needs (n +*1*) scans, where n is the length of the longest pattern.

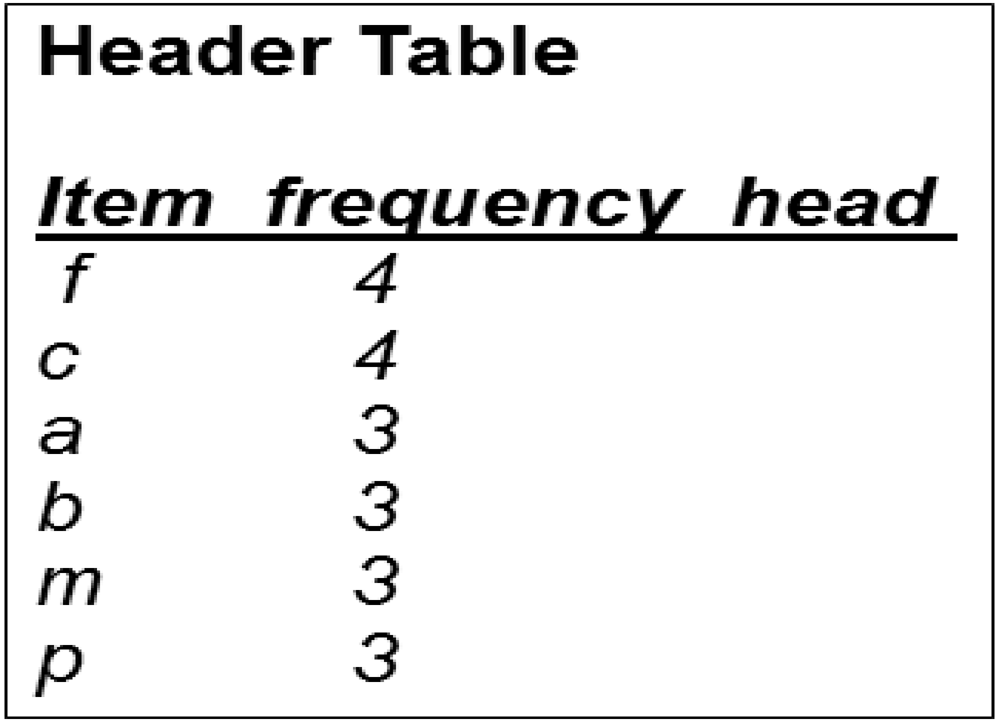
Mining Frequent Patterns *Without Candidate Generation*

- Compress a large database into a compact, Frequent-Pattern tree (FP-tree) structure
  - highly condensed, but complete for frequent pattern mining
  - avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method
  - A divide-and-conquer methodology: decompose mining tasks into smaller ones
  - Avoid candidate generation: sub-database test only!

Speeding up: FP Tree



TID	*Items bought*
100	{*f, a, c, d, g, i, m, p*}
200	{*a, b, c, f, l, m, o*}
300	{*b, f, h, j, o*}
400	{*b, c, k, s, p*}
500	{*a, f, c, e, l, p, m, n*}

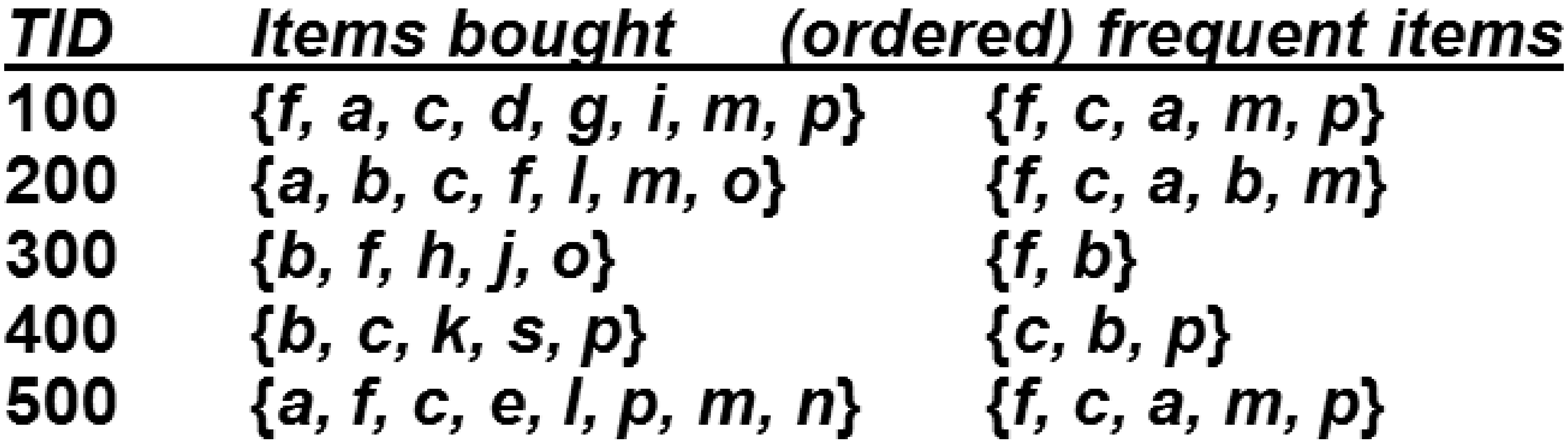


Header Table

Item	*frequency*	*head*
f	*4*	
c	*4*	
a	*3*	
b	*3*	
m	*3*	
p	*3*	

We avoided all those items that do not have the minimum support of 50% (so, a count of 3 in 5 transactions). So, d, e, g, h, I, j, k, l and n are dropped as their count is lower than 2.

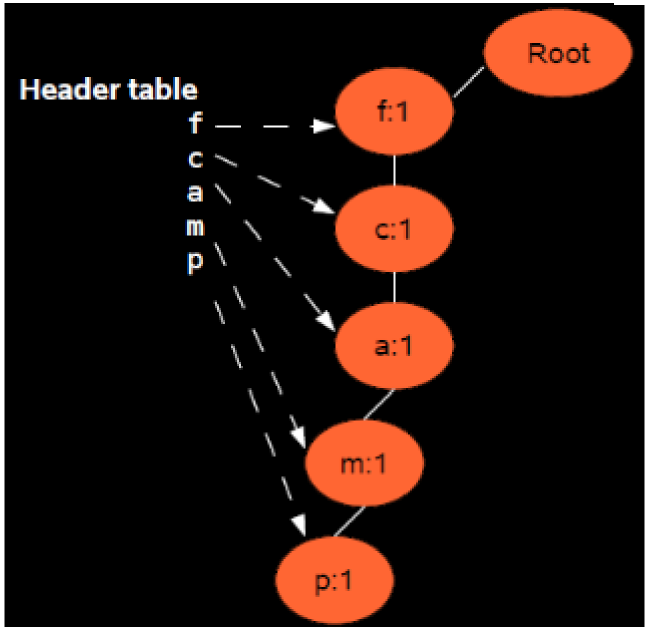
Descending order

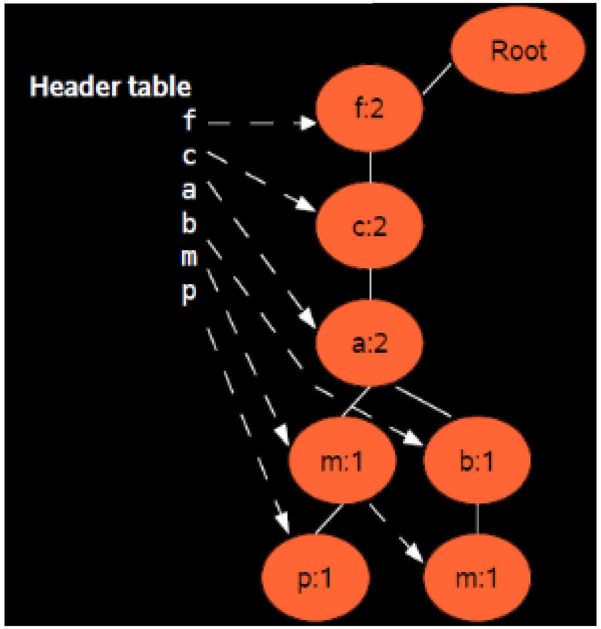


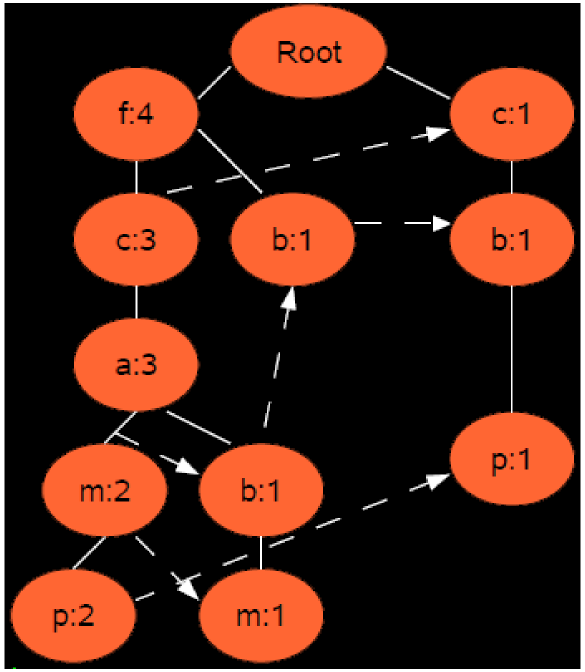
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

FP TREE

{f, c, a, m, p}
{f, c, a, b, m}
{f, b}
{c, b, p}
{f, c, a, m, p}







Benefits of the FP-tree Structure

- Completeness:
  - never breaks a long pattern of any transaction
  - preserves complete information for frequent pattern mining
- Compactness
  - reduce irrelevant information—infrequent items are gone
  - frequency descending ordering: more frequent items are more likely to be shared
  - never be larger than the original database (if not count node-links and counts)

ADDITIONAL SLIDES ON APRIORI

Details: the algorithm

```
Algorithm Apriori($T$)
    $C_1 \leftarrow$ init-pass($T$);
    $F_1 \leftarrow \{f \mid f \in C_1, f.\text{count}/n \geq minsup\}$;    // n: no. of transactions in T
    for ($k = 2$; $F_{k-1} \neq \emptyset$; $k$++) do
        $C_k \leftarrow$ candidate-gen($F_{k-1}$);
        for each transaction $t \in T$ do
            for each candidate $c \in C_k$ do
                if $c$ is contained in $t$ then
                    $c.count$++;
            end
        end
        $F_k \leftarrow \{c \in C_k \mid c.count/n \geq minsup\}$
    end
return $F \leftarrow \bigcup_k F_k$;
```

Apriori candidate generation

- The candidate-gen function takes $F_{k\text{-}1}$ and returns a superset (called the candidates) of the set of all frequent *k*-itemsets. It has two steps
  - *join* step: Generate all possible candidate itemsets C_k of length *k*
  - *prune* step: Remove those candidates in C_k that cannot be frequent.

Candidate-gen function

```
Function candidate-gen($F_{k-1}$)
    $C_k \leftarrow \emptyset$;
    for all $f_1, f_2 \in F_{k-1}$
            with $f_1 = \{i_1, \ldots, i_{k-2}, i_{k-1}\}$
            and $f_2 = \{i_1, \ldots, i_{k-2}, i'_{k-1}\}$
            and $i_{k-1} < i'_{k-1}$ do
        $c \leftarrow \{i_1, \ldots, i_{k-1}, i'_{k-1}\}$;                // join $f_1$ and $f_2$
        $C_k \leftarrow C_k \cup \{c\}$;
        for each ($k$-1)-subset $s$ of $c$ do
            if ($s \notin F_{k-1}$) then
                delete $c$ from $C_k$;                     // prune
        end
    end
    return $C_k$;
```

Details: Ordering of Items

- The items in I are sorted in lexicographic order (which is a total order).
- The order is used throughout the algorithm in each itemset.
- $\{w[1], w[2], \ldots, w[k]\}$ represents a k-itemset w consisting of items $w[1], w[2], \ldots, w[k]$, where $w[1] < w[2] < \ldots < w[k]$ according to the total order.

An example

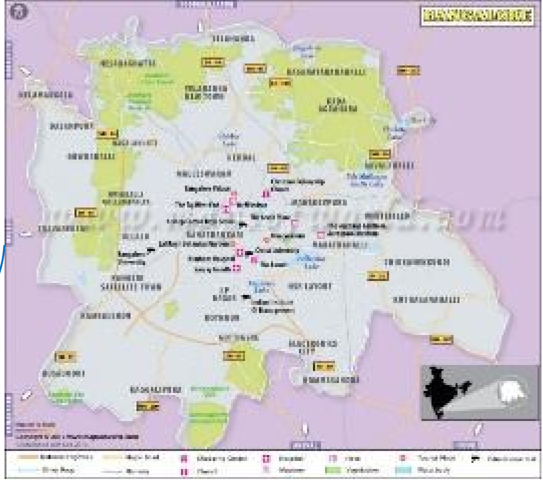
- $F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\},$
  $\{1, 3, 5\}, \{2, 3, 4\}\}$

- After join
  - $C_4 = \{\{1, 2, 3, 4\}, \{1, 3, 4, 5\}\}$
- After pruning:
  - $C_4 = \{\{1, 2, 3, 4\}\}$

  because $\{1, 4, 5\}$ is not in F_3 ($\{1, 3, 4, 5\}$ is removed)

INSOFE

Inspire...Educate...Transform.



HYDERABAD

Office and Classrooms

Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering

BENGALURU

Office

Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom

KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.