Business Intelligence Trends
商業智慧趨勢

商業智慧的資料探勘
(Data Mining for Business Intelligence)

1012BIT05
MIS MBA
Mon 6, 7 (13:10-15:00) Q407

Min-Yuh Day
戴敏育
Assistant Professor
專任助理教授
Dept. of Information Management, Tamkang University
淡江大學 資訊管理學系

1

課程大綱 (Syllabus)

週次 日期　　　內容（Subject/Topics）

1　102/02/18　商業智慧趨勢課程介紹
　　　　　　　(Course Orientation for Business Intelligence Trends)

2　102/02/25　管理決策支援系統與商業智慧
　　　　　　　(Management Decision Support System and Business Intelligence)

3　102/03/04　企業績效管理 (Business Performance Management)

4　102/03/11　資料倉儲 (Data Warehousing)

5　102/03/18　商業智慧的資料探勘 (Data Mining for Business Intelligence)

6　102/03/25　商業智慧的資料探勘 (Data Mining for Business Intelligence)

7　102/04/01　教學行政觀摩日 (Off-campus study)

8　102/04/08　個案分析一 (SAS EM 分群分析)： Banking Segmentation
　　　　　　　(Cluster Analysis – KMeans using SAS EM)

9　102/04/15　個案分析二 (SAS EM 關連分析)： Web Site Usage Associations
　　　　　　　(Association Analysis using SAS EM)

課程大綱 (Syllabus)

週次 日期　　　內容（Subject/Topics）

10　102/04/22　期中報告 (Midterm Presentation)

11　102/04/29　個案分析三 (SAS EM 決策樹、模型評估)：
　　　　　　　　Enrollment Management Case Study
　　　　　　　　(Decision Tree, Model Evaluation using SAS EM)

12　102/05/06　個案分析四 (SAS EM 迴歸分析、類神經網路)：
　　　　　　　　Credit Risk Case Study
　　　　　　　　(Regression Analysis, Artificial Neural Network using SAS EM)

13　102/05/13　文字探勘與網路探勘 (Text and Web Mining)

14　102/05/20　意見探勘與情感分析 (Opinion Mining and Sentiment Analysis)

15　102/05/27　商業智慧導入與趨勢
　　　　　　　　(Business Intelligence Implementation and Trends)

16　102/06/03　商業智慧導入與趨勢
　　　　　　　　(Business Intelligence Implementation and Trends)

17　102/06/10　期末報告1 (Term Project Presentation 1)

18　102/06/17　期末報告2 (Term Project Presentation 2)

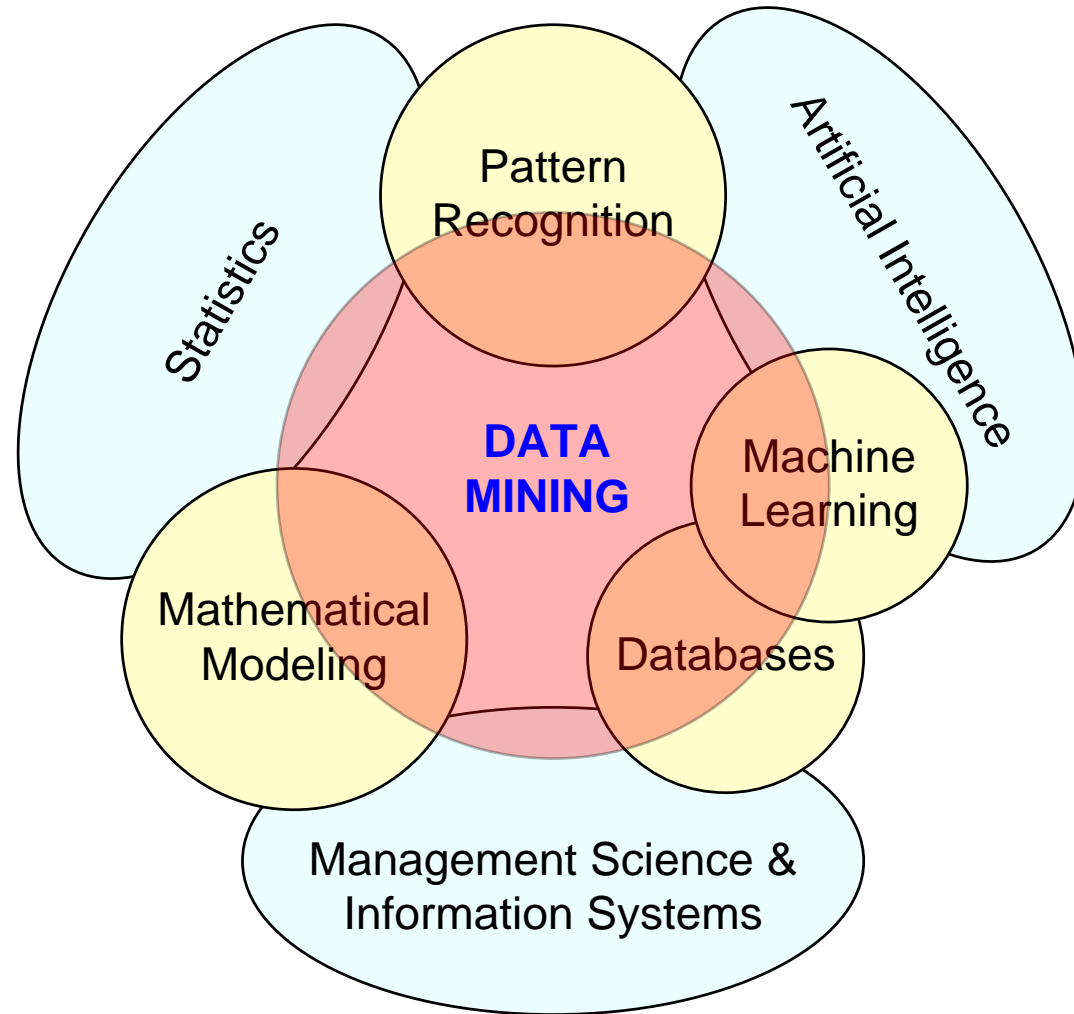# Decision Support and Business Intelligence Systems
(9th Ed., Prentice Hall)

Chapter 5:

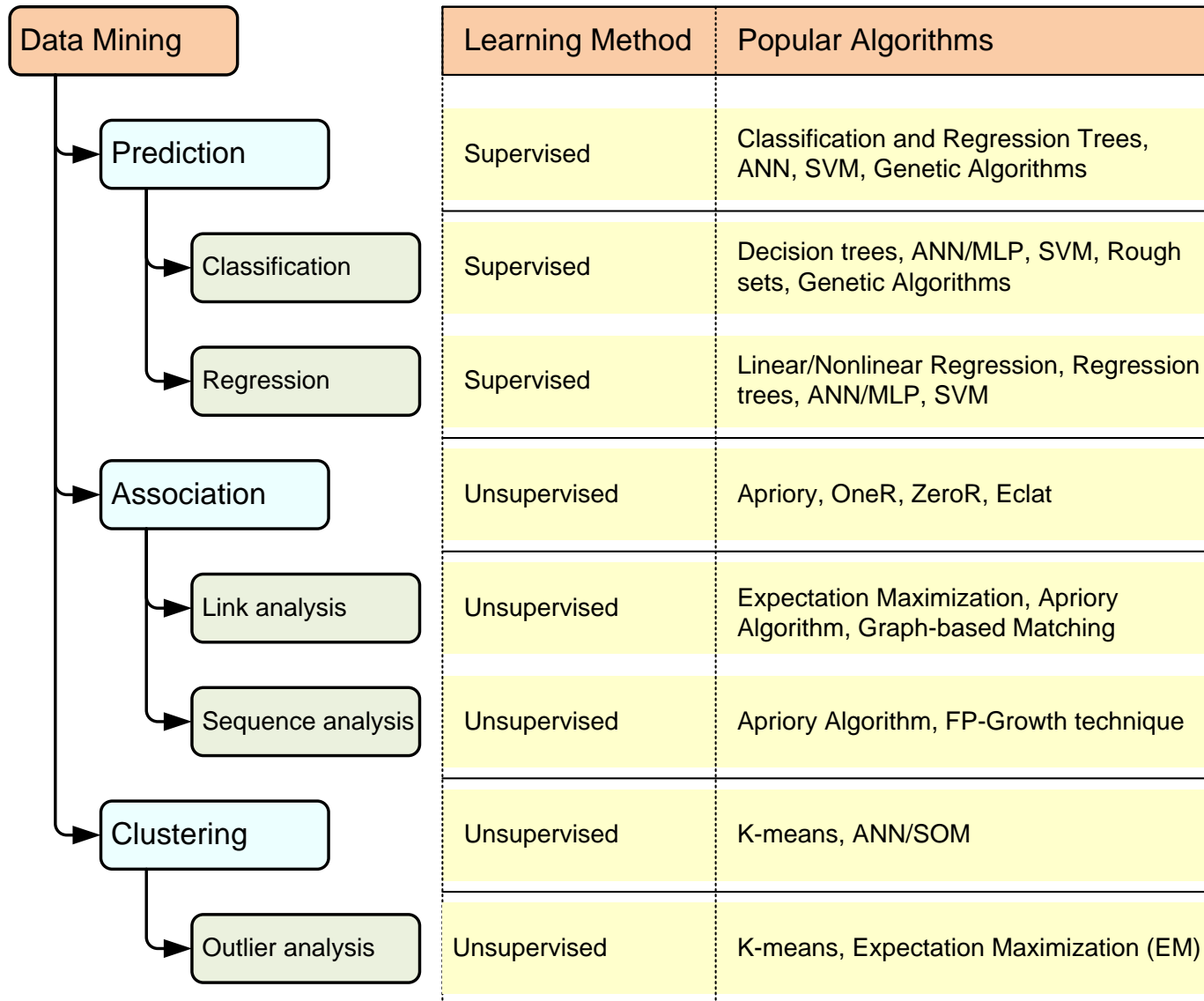### Data Mining for Business Intelligence

Learning Objectives

- Define data mining as an enabling technology for business intelligence
- Standardized data mining processes
  - CRISP-DM
  - SEMMA
- Association Analysis
  - Association Rule Mining (Apriori Algorithm)
- Classification
  - Decision Tree
- Cluster Analysis
  - *K-Means* Clustering

# Data Mining at the Intersection of Many Disciplines

A Taxonomy for Data Mining Tasks

Data Mining		Learning Method	Popular Algorithms
Prediction		Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
	Classification	Supervised	Decision trees, ANN/MLP, SVM, Rough sets, Genetic Algorithms
	Regression	Supervised	Linear/Nonlinear Regression, Regression trees, ANN/MLP, SVM
Association		Unsupervised	Apriory, OneR, ZeroR, Eclat
	Link analysis	Unsupervised	Expectation Maximization, Apriory Algorithm, Graph-based Matching
	Sequence analysis	Unsupervised	Apriory Algorithm, FP-Growth technique
Clustering		Unsupervised	K-means, ANN/SOM
	Outlier analysis	Unsupervised	K-means, Expectation Maximization (EM)

7

Data Mining Software

- Commercial
  - SPSS - PASW (formerly Clementine)
  - SAS - Enterprise Miner
  - IBM - Intelligent Miner
  - StatSoft – Statistical Data Miner
  - … many more

- Free and/or Open Source
  - Weka
  - RapidMiner…
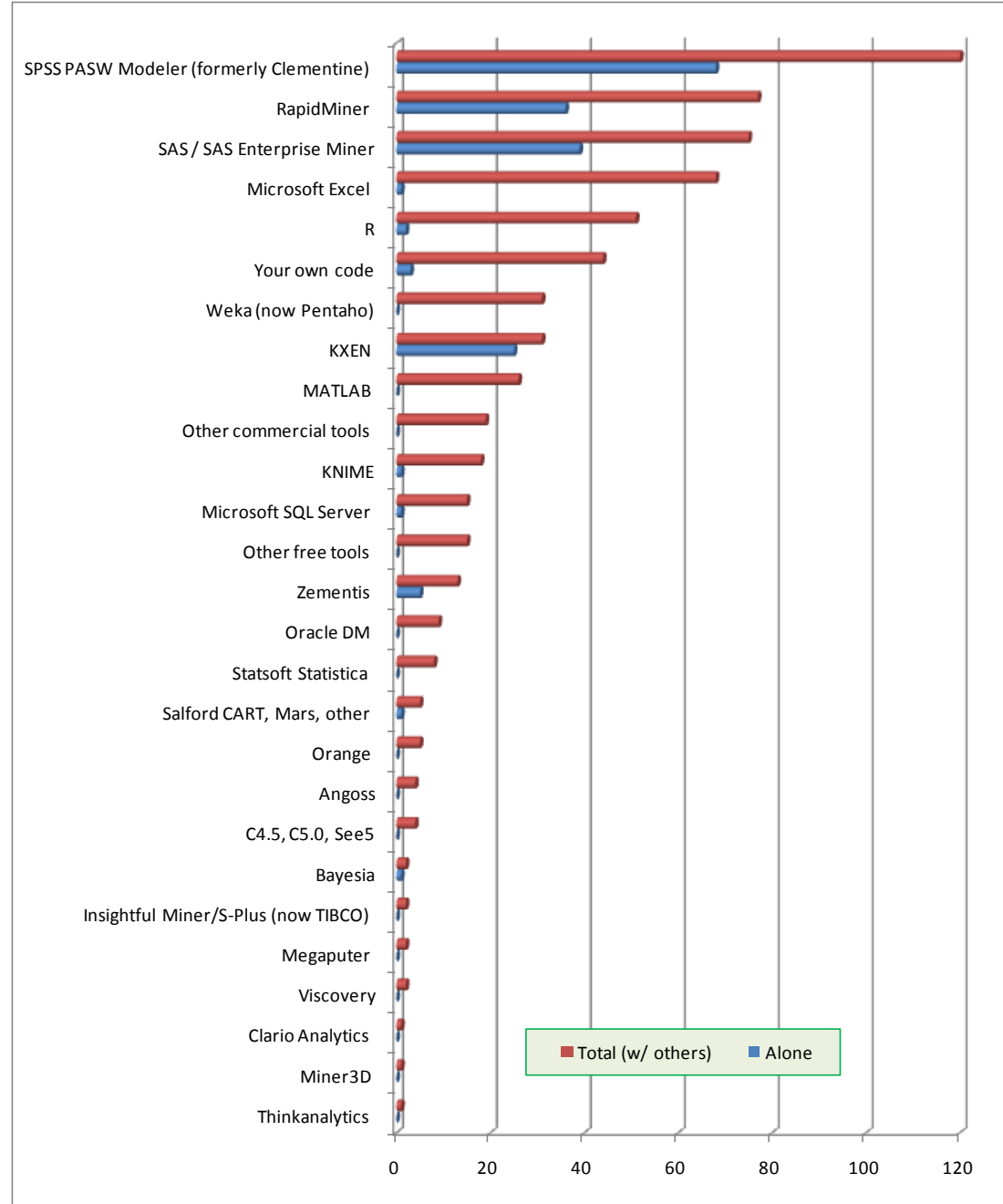
Source: KDNuggets.com, May 2009

Why Data Mining?

- More intense competition at the global scale

- Recognition of the value in data sources

- Availability of quality data on customers, vendors, transactions, Web, etc.

- Consolidation and integration of data repositories into data warehouses

- The exponential increase in data processing and storage capabilities; and decrease in cost

- Movement toward conversion of information resources into nonphysical form

Definition of Data Mining

- The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data stored in structured databases.
  - Fayyad et al., (1996)

- Keywords in this definition: Process, nontrivial, valid, novel, potentially useful, understandable.

- Data mining: a misnomer?

- Other names:
  - knowledge extraction, pattern analysis, knowledge discovery, information harvesting, pattern searching, data dredging,...
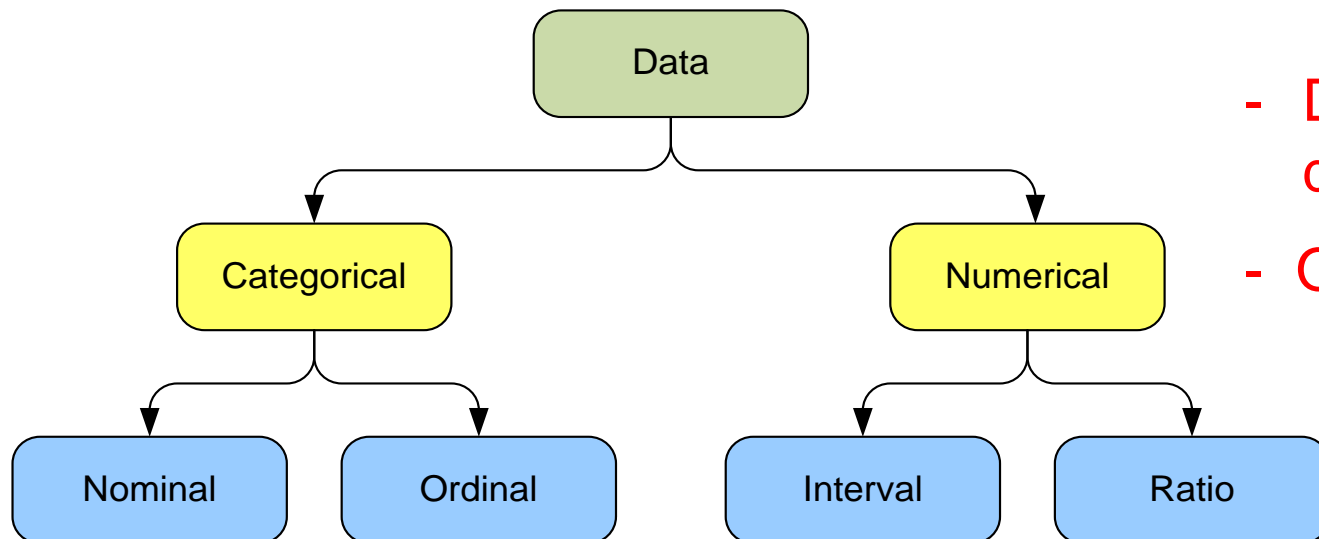
Data Mining Characteristics/Objectives

- Source of data for DM is often a consolidated data warehouse (not always!)

- DM environment is usually a client-server or a Web-based information systems architecture

- Data is the most critical ingredient for DM which may include soft/unstructured data

- The miner is often an end user

- Striking it rich requires creative thinking

- Data mining tools' capabilities and ease of use are essential (Web, Parallel processing, etc.)

Data in Data Mining

- Data: a collection of facts usually obtained as the result of experiences, observations, or experiments

- Data may consist of numbers, words, images, …

- Data: lowest level of abstraction (from which information and knowledge are derived)

```
                    ┌──────────┐
                    │   Data   │
                    └──────────┘
              ┌───────────┴───────────┐
        ┌───────────┐           ┌───────────┐
        │Categorical│           │ Numerical │
        └───────────┘           └───────────┘
         ┌─────┴─────┐           ┌─────┴─────┐
    ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐
    │ Nominal │ │ Ordinal │ │Interval │ │  Ratio  │
    └─────────┘ └─────────┘ └─────────┘ └─────────┘
```

- DM with different data types?

- Other data types?

What Does DM Do?

- DM extract patterns from data
  - Pattern?
    A mathematical (numeric and/or symbolic) relationship among data items

- Types of patterns
  - Association
  - Prediction
  - Cluster (segmentation)
  - Sequential (or time series) relationships

Data Mining Applications

- Customer Relationship Management
  - Maximize return on marketing campaigns
  - Improve customer retention (churn analysis)
  - Maximize customer value (cross-, up-selling)
  - Identify and treat most valued customers

- Banking and Other Financial
  - Automate the loan application process
  - Detecting fraudulent transactions
  - Optimizing cash reserves with forecasting

Data Mining Applications (cont.)

- Retailing and Logistics
  - Optimize inventory levels at different locations
  - Improve the store layout and sales promotions
  - Optimize logistics by predicting seasonal effects
  - Minimize losses due to limited shelf life

- Manufacturing and Maintenance
  - Predict/prevent machinery failures
  - Identify anomalies in production systems to optimize the use manufacturing capacity
  - Discover novel patterns to improve product quality

Data Mining Applications (cont.)

- Brokerage and Securities Trading
  - Predict changes on certain bond prices
  - Forecast the direction of stock fluctuations
  - Assess the effect of events on market movements
  - Identify and prevent fraudulent activities in trading

- Insurance
  - Forecast claim costs for better business planning
  - Determine optimal rate plans
  - Optimize marketing to specific customers
 - Identify and prevent fraudulent claim activities

Data Mining Applications (cont.)

- Computer hardware and software

- Science and engineering

- Government and defense

- Homeland security and law enforcement

- Travel industry

- Healthcare

- Medicine

Highly popular application areas for data mining

- Entertainment industry

- Sports

- Etc.

Data Mining Process

- A manifestation of best practices

- A systematic way to conduct DM projects

- Different groups has different versions

- Most common standard processes:
  - CRISP-DM
    (Cross-Industry Standard Process for Data Mining)
  - SEMMA
    (Sample, Explore, Modify, Model, and Assess)
  - KDD
    (Knowledge Discovery in Databases)

Data Mining Process: CRISP-DM
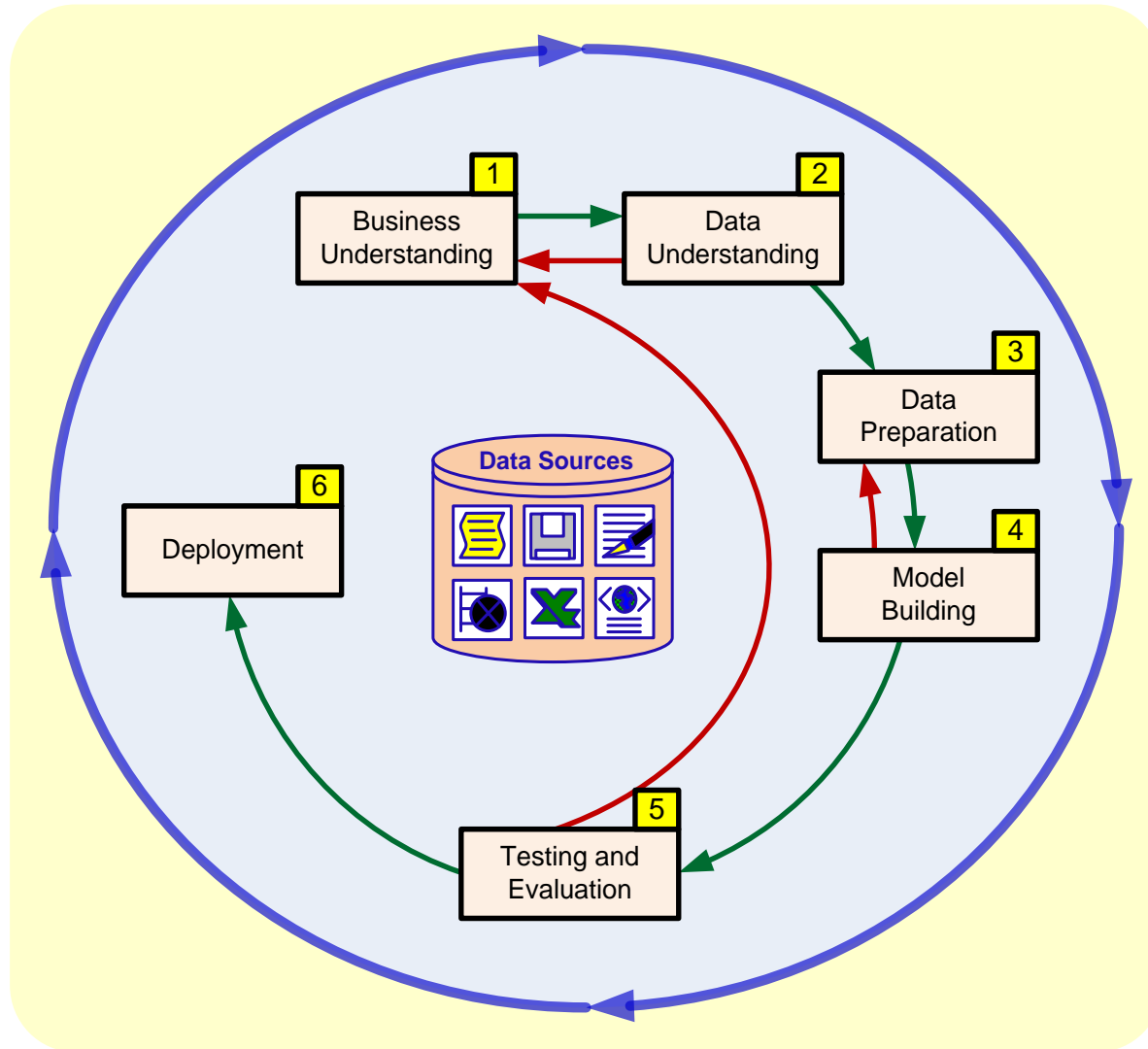
Data Mining Process: CRISP-DM

Step 1: Business Understanding

Step 2: Data Understanding

Step 3: Data Preparation (!)

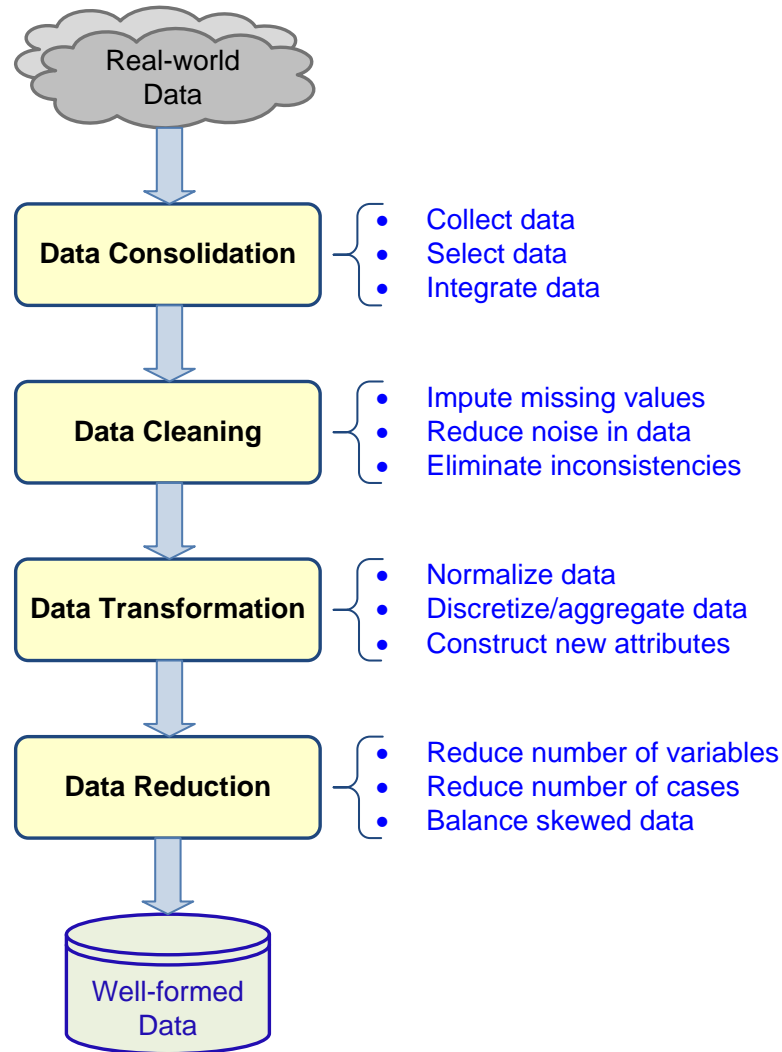Accounts for ~85% of total project time
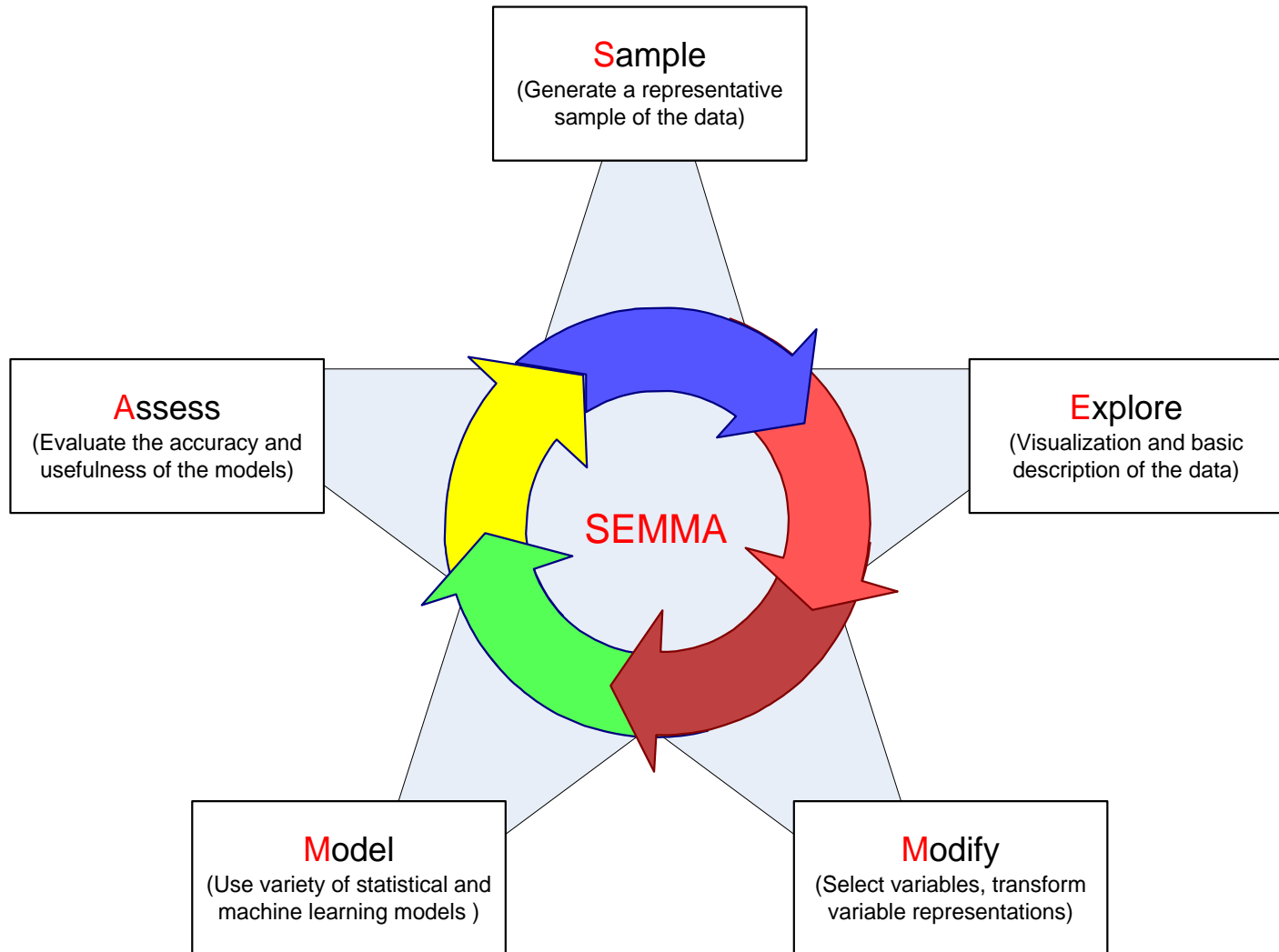
Step 4: Model Building

Step 5: Testing and Evaluation

Step 6: Deployment

- The process is highly repetitive and experimental (DM: art versus science?)

Data Preparation – A Critical DM Task

Real-world Data

Data Consolidation
- Collect data
- Select data
- Integrate data

Data Cleaning
- Impute missing values
- Reduce noise in data
- Eliminate inconsistencies

Data Transformation
- Normalize data
- Discretize/aggregate data
- Construct new attributes

Data Reduction
- Reduce number of variables
- Reduce number of cases
- Balance skewed data

Well-formed Data

Data Mining Process: SEMMA

Data Mining Methods: Classification

- Most frequently used DM method
- Part of the machine-learning family
- Employ supervised learning
- Learn from past data, classify new data
- The output variable is categorical (nominal or ordinal) in nature
- Classification versus regression?
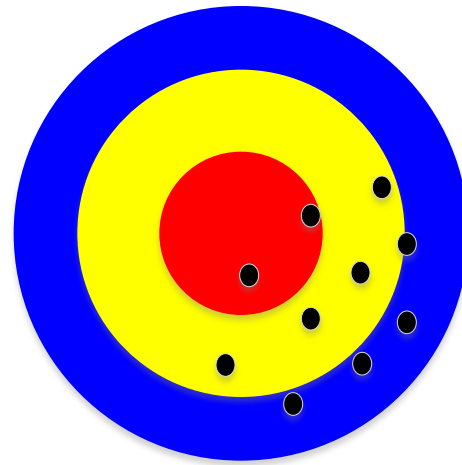- Classification versus clustering?
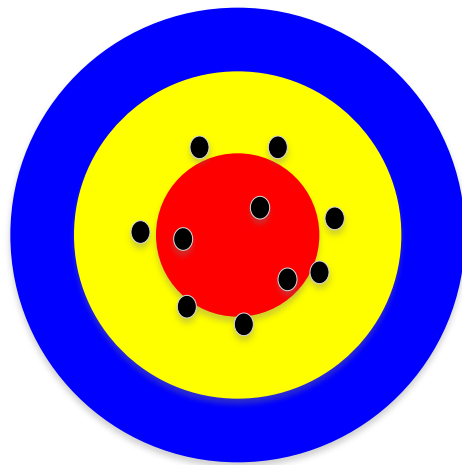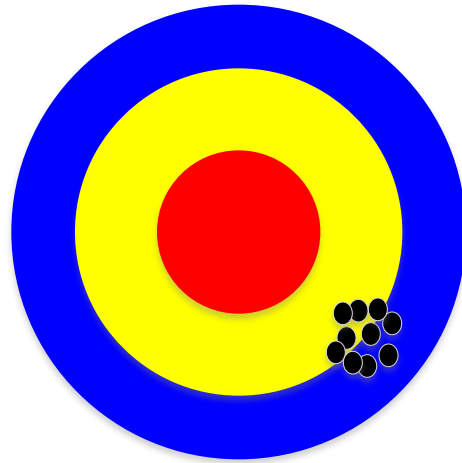
Assessment Methods for Classification
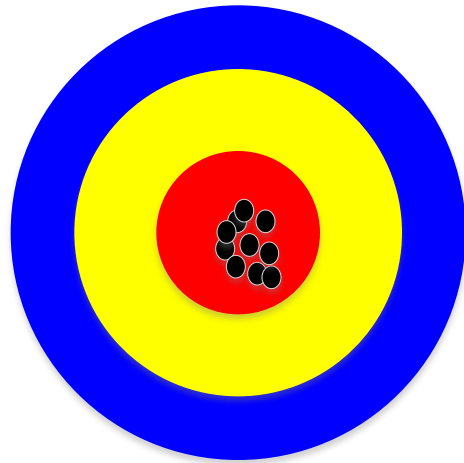
- Predictive accuracy
    - Hit rate

- Speed
    - Model building; predicting

- Robustness

- Scalability

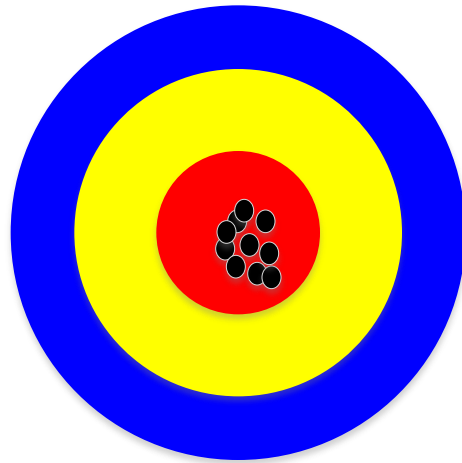- Interpretability
    - Transparency, explainability
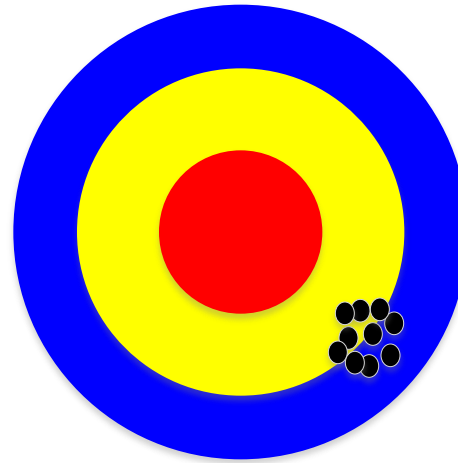
Accuracy   **Validity**

Precision   **Reliability**

Accuracy vs. Precision

A

**High Accuracy
High Precision**

B

**Low Accuracy
High Precision**

C
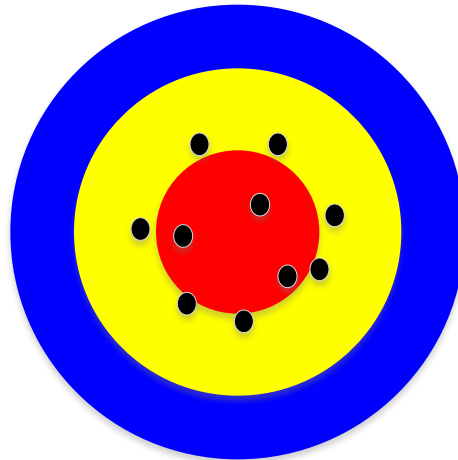
**High Accuracy
Low Precision**

D
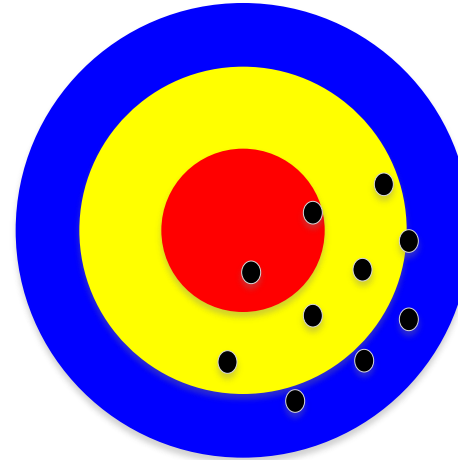
**Low Accuracy
Low Precision**

Accuracy vs. Precision
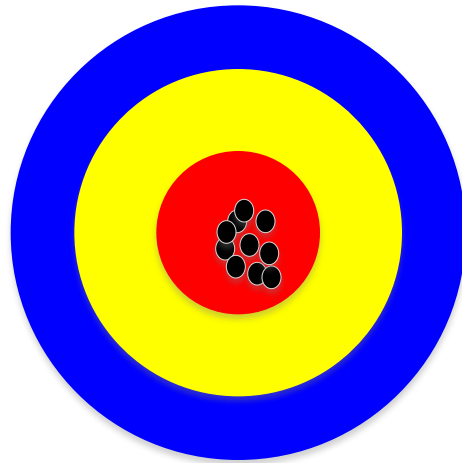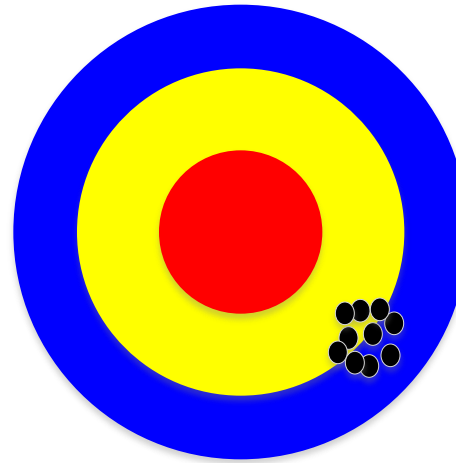
A

High Accuracy
High Precision

High Validity
High Reliability

B

Low Accuracy
High Precision

Low Validity
High Reliability

C

High Accuracy
Low Precision

High Validity
Low Reliability

D

Low Accuracy
Low Precision

Low Validity
Low Reliability

Accuracy vs. Precision

A

**High Accuracy
High Precision**

**High Validity
High Reliability**

B

**Low Accuracy
High Precision**

**Low Validity
High Reliability**

C

**High Accuracy
Low Precision**

**High Validity
Low Reliability**

D
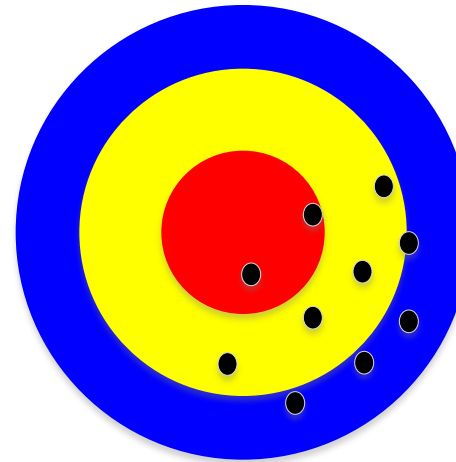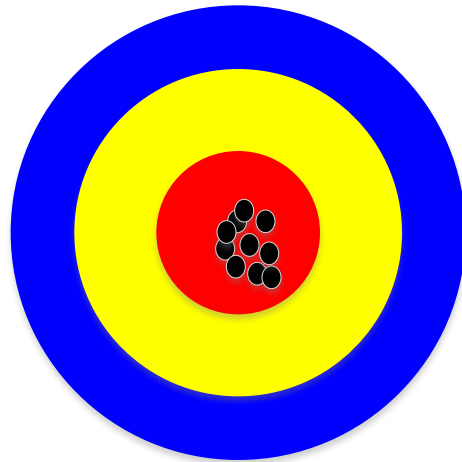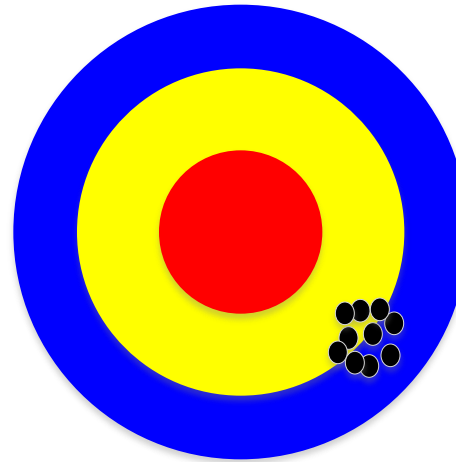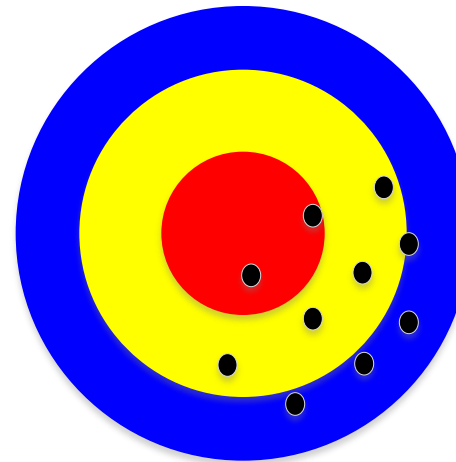
**Low Accuracy
Low Precision**

**Low Validity
Low Reliability**

Accuracy of Classification Models

- In classification problems, the primary source for accuracy estimation is the confusion matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Estimation Methodologies for Classification

- Simple split (or holdout or test sample estimation)
  - Split the data into 2 mutually exclusive sets training (~70%) and testing (30%)
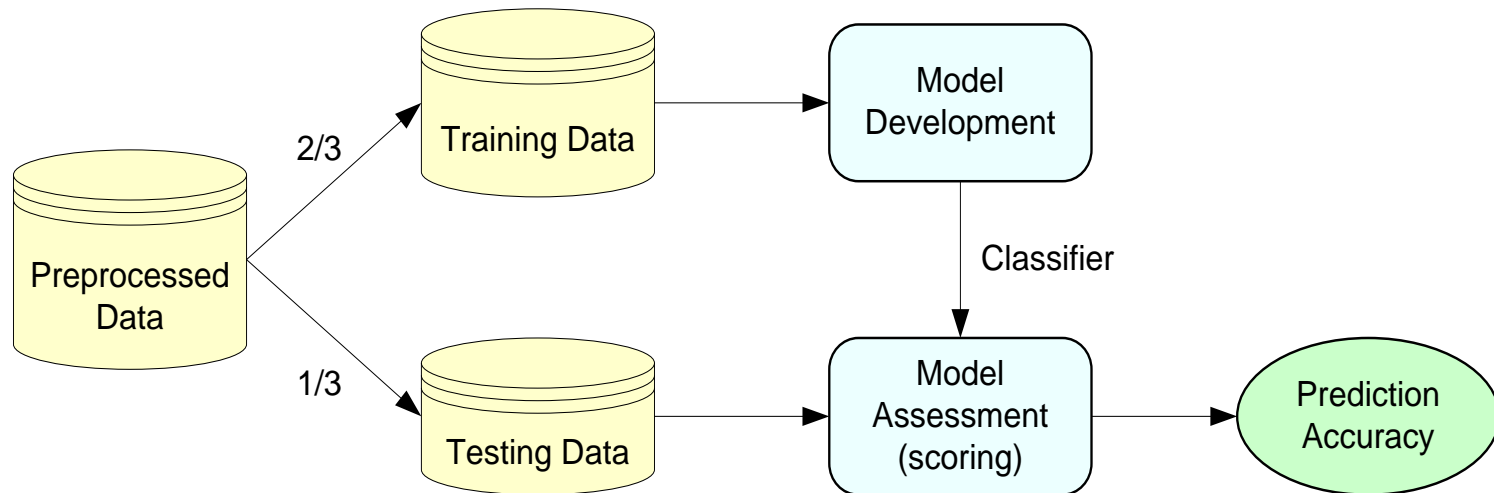
  - For ANN, the data is split into three sub-sets (training [~60%], validation [~20%], testing [~20%])

Estimation Methodologies for Classification

- *k*-Fold Cross Validation (rotation estimation)
  - Split the data into *k* mutually exclusive subsets
  - Use each subset as testing while using the rest of the subsets as training
  - Repeat the experimentation for *k* times
  - Aggregate the test results for true estimation of prediction accuracy training
- Other estimation methodologies
  - Leave-one-out, bootstrapping, jackknifing
  - Area under the ROC curve

Estimation Methodologies for Classification – ROC Curve
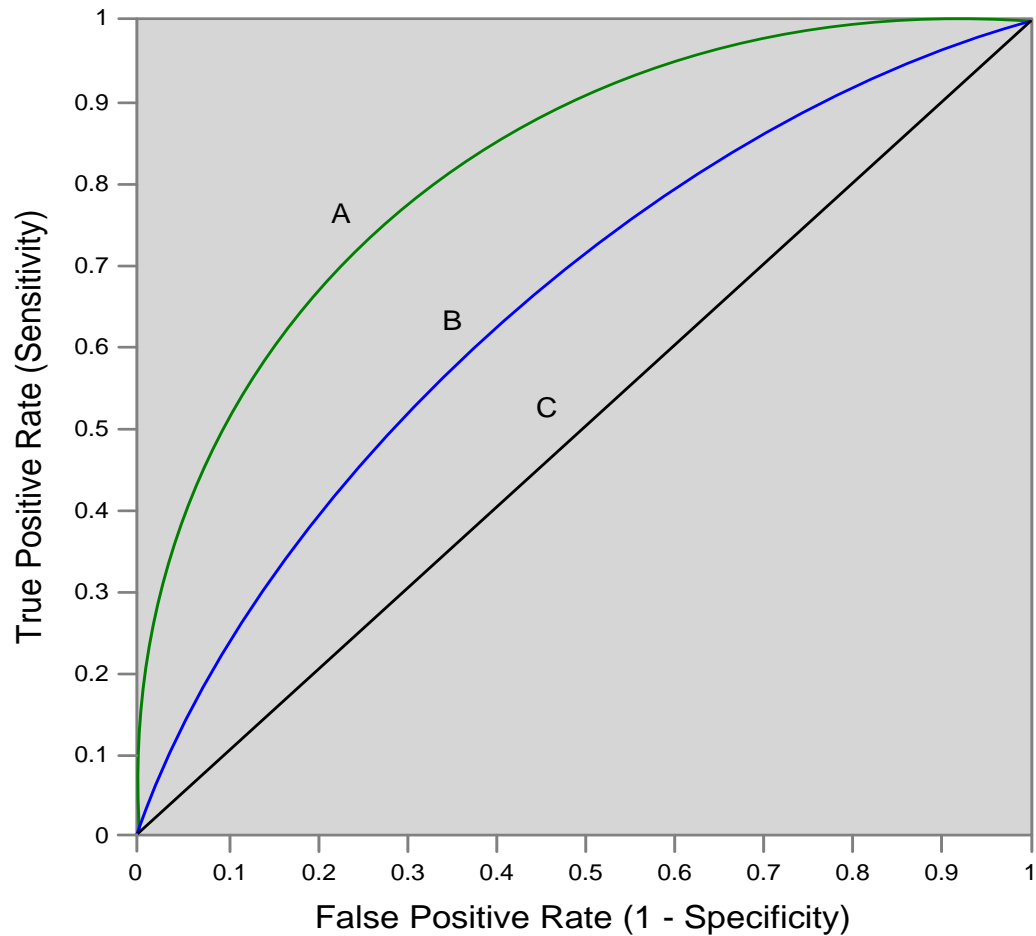
Sensitivity =True Positive Rate

Specificity =True Negative Rate

True Class (actual value)

		Positive	Negative	total
Predictive Class (prediction outcome)	Positive	True Positive (TP)	False Positive (FP)	P'
	Negative	False Negative (FN)	True Negative (TN)	N'
total		P	N	

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$True\ Positive\ Rate\ (\text{Sensitivity}) = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate\ (\text{Specificity}) = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN}$$

$$False\ Positive\ Rate\ (1 - \text{Specificity}) = \frac{FP}{FP + TN}$$

		True Class (actual value)		total
		Positive	Negative	
Predictive Class (prediction outcome)	Positive	True Positive (TP)	False Positive (FP)	P'
	Negative	False Negative (FN)	True Negative (TN)	N'
total		P	N	

$$True\ Positive\ Rate\ (Sensitivity) = \frac{TP}{TP + FN}$$

Sensitivity
= True Positive Rate
= Recall
= Hit rate
= TP / (TP + FN)

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

False Positive Rate (1 - Specificity)

True Positive Rate (Sensitivity)

	True Class (actual value)		total
	Positive	Negative	
Predictive Class (prediction outcome) — Positive	True Positive (TP)	False Positive (FP)	P'
Negative	False Negative (FN)	True Negative (TN)	N'
total	P	N	
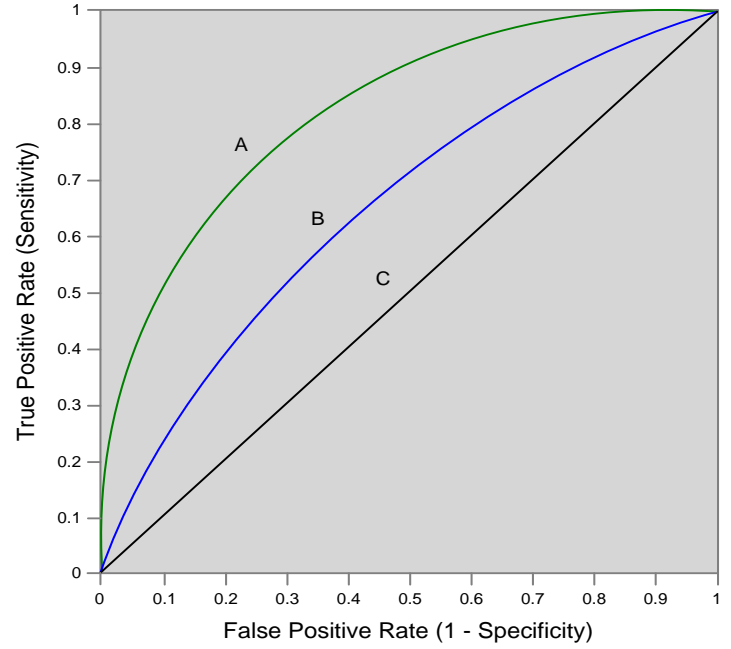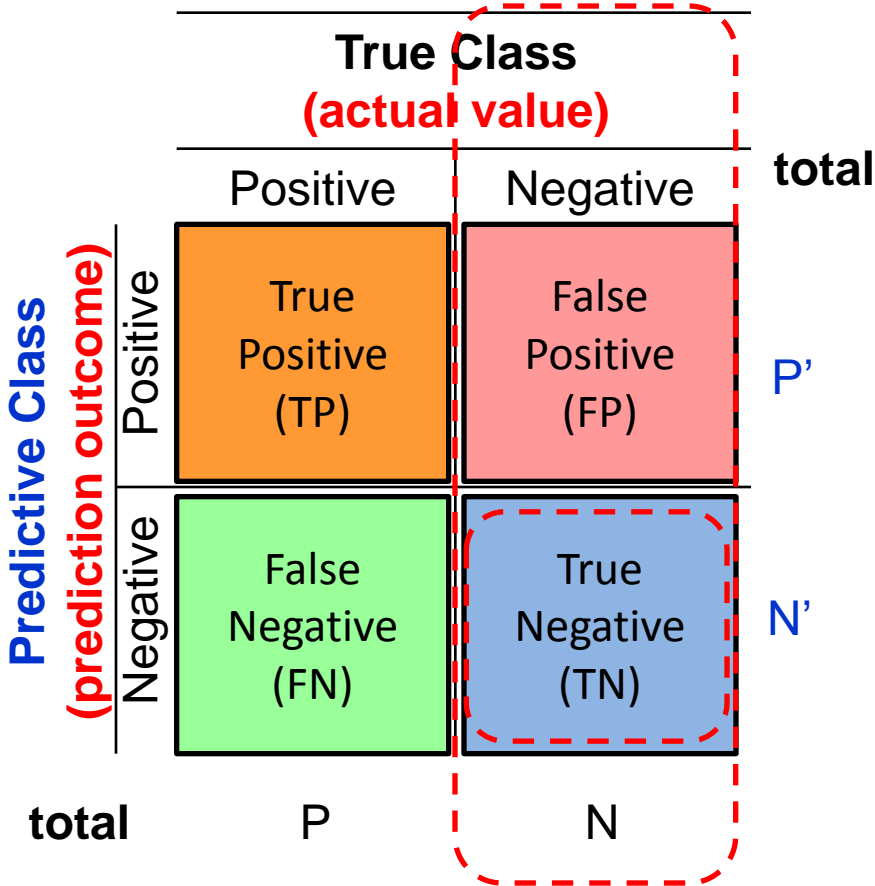
$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

Specificity
= True Negative Rate
= TN / N
= TN / (TN+ FP)

$$True\ Negative\ Rate\ (\text{Specifici ty}) = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate\ (1\text{-Specificit y}) = \frac{FP}{FP + TN}$$

True Positive Rate (Sensitivity) vs False Positive Rate (1 - Specificity)

	True Class (actual value)		total
	Positive	Negative	
Predictive Class (prediction outcome) Positive	True Positive (TP)	False Positive (FP)	P'
Negative	False Negative (FN)	True Negative (TN)	N'
total	P	N	

Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

$$Recall = \frac{TP}{TP + FN}$$

F1 score (F-score)(F-measure)
is the harmonic mean of
precision and recall
= 2TP / (P + P')
= 2TP / (2TP + FP + FN)

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic   38

A

63 (TP)	28 (FP)	91
37 (FN)	72 (TN)	109
100	100	200

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate
= TP / (TP + FN)

Specificity
= True Negative Rate
= TN / N
= TN / (TN + FP)

TPR = 0.63

$$Recall = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate\ (\text{Specifici ty}) = \frac{TN}{TN + FP}$$

FPR = 0.28

$$False\ Positive\ Rate\ (1 - \text{Specificit y}) = \frac{FP}{FP + TN}$$

PPV = 0.69
=63/(63+28)
=63/91

$$Precision = \frac{TP}{TP + FP}$$

Precision
= Positive Predictive Value (PPV)

F1 = 0.66
= 2*(0.63*0.69)/(0.63+0.69)
= (2 * 63) /(100 + 91)
= (0.63 + 0.69) / 2 =1.32 / 2 =0.66

$$F = 2 * \frac{precision * recall}{precision + recall}$$

F1 score (F-score) (F-measure)
is the harmonic mean of precision and recall
= 2TP / (P + P')
= 2TP / (2TP + FP + FN)

ACC = 0.68
= (63 + 72) / 200
= 135/200 = 67.5

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
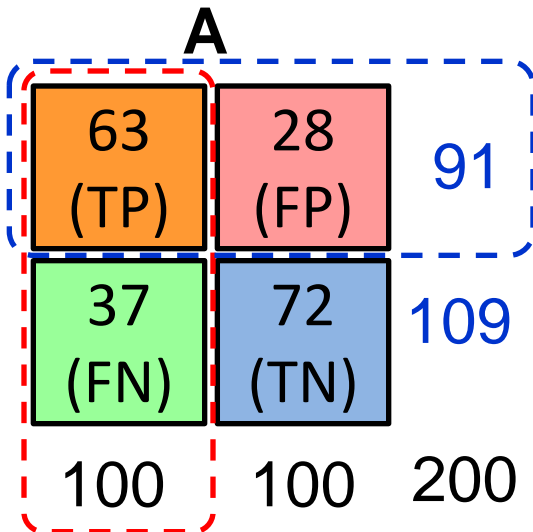
A

63 (TP)	28 (FP)	91
37 (FN)	72 (TN)	109
100	100	200

TPR = 0.63

FPR = 0.28

PPV = 0.69
    =63/(63+28)
    =63/91

F1 = 0.66
= 2*(0.63*0.69)/(0.63+0.69)
= (2 * 63) /(100 + 91)
= (0.63 + 0.69) / 2 =1.32 / 2 =0.66

ACC = 0.68
= (63 + 72) / 200
= 135/200 = 67.5

B

77 (TP)	77 (FP)	154
23 (FN)	23 (TN)	46
100	100	200
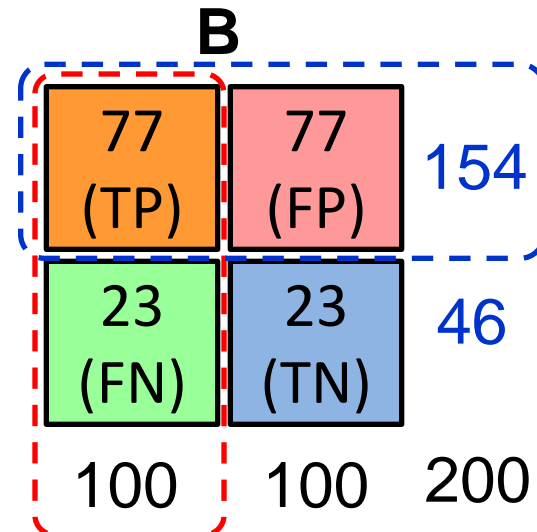
TPR = 0.77
FPR = 0.77
PPV = 0.50
F1 = 0.61
ACC = 0.50

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

$$Recall = \frac{TP}{TP + FN}$$

Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

C

24 (TP)	88 (FP)	112
76 (FN)	12 (TN)	88
100	100	200

TPR = 0.24
FPR = 0.88
PPV = 0.21
F1 = 0.22
ACC = 0.18

C'

76 (TP)	12 (FP)	88
24 (FN)	88 (TN)	112
100	100	200

TPR = 0.76
FPR = 0.12
PPV = 0.86
F1 = 0.81
ACC = 0.82

Recall
= True Positive Rate (TPR)
= Sensitivity
= Hit Rate

$$Recall = \frac{TP}{TP + FN}$$

Precision
= Positive Predictive Value (PPV)

$$Precision = \frac{TP}{TP + FP}$$

Market Basket Analysis

Which items are frequently purchased together by my customers?
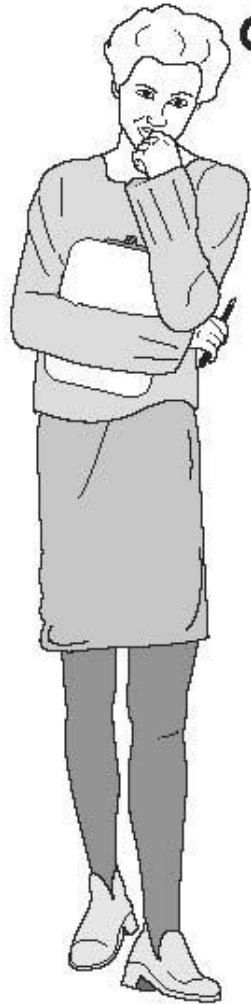
Shopping Baskets

milk bread cereal — Customer 1

milk bread sugar eggs — Customer 2

milk bread butter — Customer 3

sugar eggs — Customer n

Market Analyst
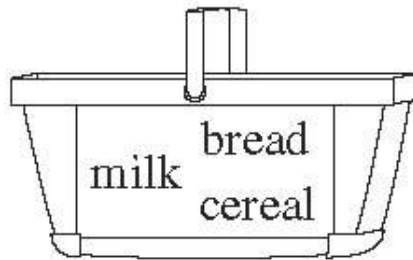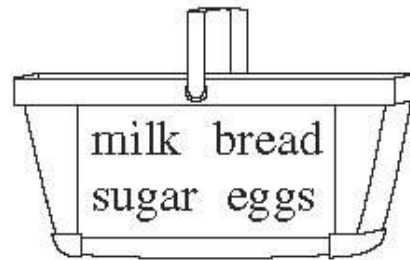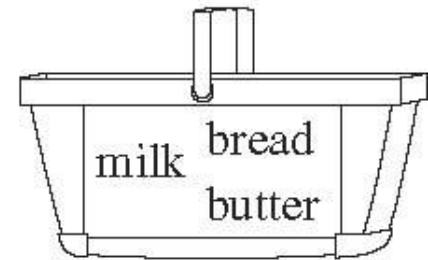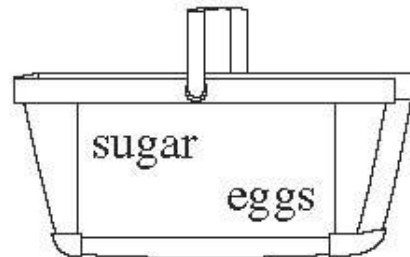
Association Rule Mining

- Apriori Algorithm

Raw Transaction Data

Transaction No	SKUs (Item No)
1	1, 2, 3, 4
1	2, 3, 4
1	2, 3
1	1, 2, 4
1	1, 2, 3, 4
1	2, 4

One-item Itemsets

Itemset (SKUs)	Support
1	3
2	6
3	4
4	5

Two-item Itemsets

Itemset (SKUs)	Support
1, 2	3
1, 3	2
1, 4	3
2, 3	4
2, 4	5
3, 4	3

Three-item Itemsets

Itemset (SKUs)	Support
1, 2, 4	3
2, 3, 4	3

Association Rule Mining

- A very popular DM method in business
- Finds interesting relationships (affinities) between variables (items or events)
- Part of machine learning family
- Employs unsupervised learning
- There is no output variable
- Also known as market basket analysis
- Often used as an example to describe DM to ordinary people, such as the famous "relationship between diapers and beers!"

Association Rule Mining

- **Input:** the simple point-of-sale transaction data
- **Output:** Most frequent affinities among items
- <u>Example:</u> according to the transaction data…

  "Customer who bought a laptop computer and a virus protection software, also bought extended service plan 70 percent of the time."

- How do you use such a pattern/knowledge?
  - Put the items next to each other for ease of finding
  - Promote the items as a package (do not put one on sale if the other(s) are on sale)
  - Place items far apart from each other so that the customer has to walk the aisles to search for it, and by doing so potentially seeing and buying other items

Association Rule Mining

- A representative applications of association rule mining include

  - In business: cross-marketing, cross-selling, store design, catalog design, e-commerce site design, optimization of online advertising, product pricing, and sales/promotion configuration

  - In medicine: relationships between symptoms and illnesses; diagnosis and patient characteristics and treatments (to be used in medical DSS); and genes and their functions (to be used in genomics projects)...

Association Rule Mining

- Are all association rules interesting and useful?

A Generic Rule: $X \Rightarrow Y$ [S%, C%]

X, Y: products and/or services

X: Left-hand-side (LHS)

Y: Right-hand-side (RHS)

S: Support: how often **X** and **Y** go together

C: Confidence: how often **Y** go together with the **X**

Example: {Laptop Computer, Antivirus Software} $\Rightarrow$ {Extended Service Plan} [30%, 70%]

Association Rule Mining

- Algorithms are available for generating association rules
  - Apriori
  - Eclat
  - FP-Growth
  - + Derivatives and hybrids of the three
- The algorithms help identify the frequent item sets, which are, then converted to association rules

Association Rule Mining

- Apriori Algorithm
  - Finds subsets that are common to at least a minimum number of the itemsets
  - uses a bottom-up approach
    - frequent subsets are extended one item at a time (the size of frequent subsets increases from one-item subsets to two-item subsets, then three-item subsets, and so on), and
    - groups of candidates at each level are tested against the data for minimum

Basic Concepts: Frequent Patterns and Association Rules

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F

Customer buys both

Customer buys diaper

Customer buys beer

- Itemset X = $\{x_1, ..., x_k\}$

- Find all the rules $X \rightarrow Y$ with minimum support and confidence

  – support, *s*, probability that a transaction contains $X \cup Y$

  – confidence, *c,* conditional probability that a transaction having X also contains *Y*
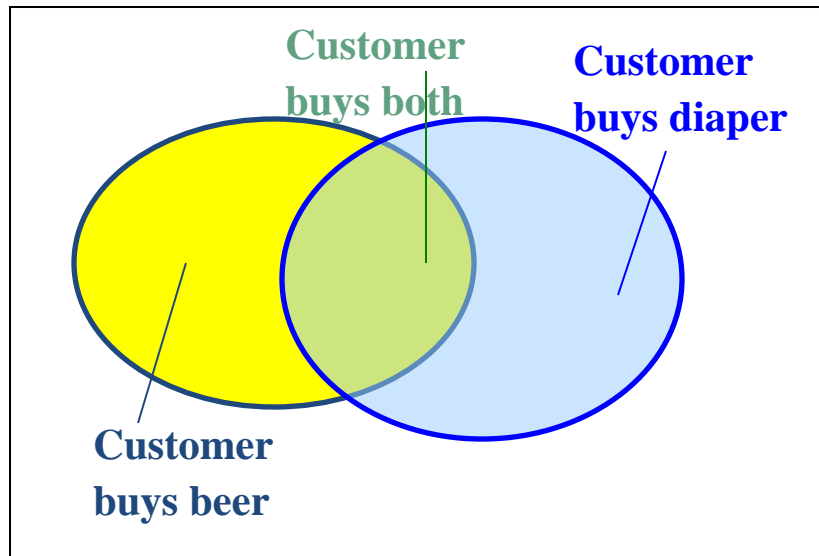
Let sup$_{min}$ = 50%, conf$_{min}$ = 50%
Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}
Association rules:
    A $\rightarrow$ D (60%, 100%)
    D $\rightarrow$ A (60%, 75%)

A $\rightarrow$ D (support = 3/5 = 60%, confidence = 3/3 =100%)
D $\rightarrow$ A (support = 3/5 = 60%, confidence = 3/4 = 75%)

Market basket analysis

- Example
  - Which groups or sets of items are customers likely to purchase on a given trip to the store?

- Association Rule
  - *Computer → antivirus_software [support = 2%; confidence = 60%]*
    - A support of 2% means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.
    - A confidence of 60% means that 60% of the customers who purchased a computer also bought the software.

Association rules

- Association rules are considered interesting if they satisfy both
  - a minimum support threshold and
  - a minimum confidence threshold.

Frequent Itemsets, Closed Itemsets, and Association Rules

Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of items. Let D, the task-relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \phi$. The rule $A \Rightarrow B$ holds in the transaction set D with **support** s, where s is the percentage of transactions in D that contain $A \cup B$ (i.e., the *union* of sets A and B, or say, both A and B). This is taken to be the probability, $P(A \cup B)$.[1] The rule $A \Rightarrow B$ has **confidence** c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, $P(B|A)$. That is,

$$\text{Support } (A \rightarrow B) \quad = P(A \cup B)$$

$$\text{Confidence } (A \rightarrow B) = P(B|A)$$

Support (A→B) = P(A ∪ B)
Confidence (A→B) = P(B|A)

- The notation *P(A ∪ B) indicates the probability that a transaction contains the union of set A and set B*
  - *(i.e., it contains every item in A and in B).*
- *This should not be confused with P(A or B),* which indicates the probability that a transaction contains either *A or B.*

Does diaper purchase predict beer purchase?

- Contingency tables

Beer

	Yes	No	
No diapers	6	94	100
diapers	40	60	100

DEPENDENT (yes)

Beer

	Yes	No
	23	77
	23	77

INDEPENDENT (no predictability)

$$\textbf{\textit{Support (A} \rightarrow \textit{B) = P(A} \cup \textit{B)}}$$

$$\textbf{\textit{Confidence (A} \rightarrow \textit{B) = P(B|A)}}$$

$$\textbf{\textit{Conf (A} \rightarrow \textit{B) = Supp (A} \cup \textit{B)/ Supp (A)}}$$

$$\textbf{\textit{Lift (A} \rightarrow \textit{B) = Supp (A} \cup \textit{B) / (Supp (A) x Supp (B))}}$$

Lift (Correlation)

$$\textbf{\textit{Lift (A} \rightarrow \textit{B) = Confidence (A} \rightarrow \textit{B) / Support(B)}}$$

Lift

Lift = Confidence / Expected Confidence if Independent

Checking ➡ Saving ⬇	No (1500)	Yes (8500)	(10000)
No	500	3500	4000
Yes	1000	5000	6000

SVG=>CHKG Expect   8500/10000 = 85% if independent
Observed Confidence is 5000/6000 = 83%
Lift = 83/85 < 1.
Savings account holders actually LESS likely than others to have checking account !!!

- Rules that satisfy both a minimum support threshold (*min_sup*) and a minimum confidence threshold (*min_conf*) are called **strong**.

- By convention, we write support and confidence values so as to occur between 0% and 100%, rather than 0 to 1.0.

- itemset
  - A set of items is referred to as an itemset.
- K-itemset
  - An itemset that contains *k items is a k*-itemset.
- Example:
  - The set {*computer, antivirus software*} is a 2-itemset.

Absolute Support and Relative Support

- Absolute Support
  - The occurrence frequency of an itemset is the number of transactions that contain the itemset
    - frequency, support count, or count of the itemset
  - Ex: 3
- Relative support
  - Ex: 60%

- If the relative support of an itemset *I satisfies a prespecified minimum support threshold, then I is a* frequent itemset.

  - *i.e., the absolute support of I satisfies the corresponding minimum support count threshold*

- The set of frequent *k-itemsets is commonly denoted by* L_K

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}$$

- the confidence of rule *A → B can be easily derived from the* support counts of *A and A ∪ B.*

- once the support counts of *A, B, and A ∪ B are* found, it is straightforward to derive the corresponding association rules *A →B and B →A* and check whether they are strong.

- Thus the problem of mining association rules can be reduced to that of mining frequent itemsets.

Association rule mining: Two-step process

1. Find all frequent itemsets
   - By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min_sup.*

2. Generate strong association rules from the frequent itemsets
   - By definition, these rules must satisfy minimum support and minimum confidence.

Efficient and Scalable Frequent Itemset Mining Methods

- The Apriori Algorithm
  - Finding Frequent Itemsets Using Candidate Generation

Apriori Algorithm

- **Apriori** is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.

- The name of the algorithm is based on the fact that the algorithm uses *prior knowledge of frequent itemset properties, as we shall* see following.

Apriori Algorithm

- Apriori employs an iterative approach known as a *level-wise search, where k-itemsets are used to explore (k+1)-itemsets.*

- *First, the set of frequent 1-itemsets is found* by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1.

- *Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found.*

- *The finding of each L_k requires one full scan of the database.*

Apriori Algorithm

- To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property.

- Apriori property

  - *All nonempty subsets of a frequent itemset must also be frequent.*

- *How is the Apriori property used in the algorithm?*
  - How L_{k-1} *is used to find* L_k *for k >= 2.*
  - A two-step process is followed, consisting of join and prune actions.

Apriori property used in algorithm
1. The join step

1. **The join step:** To find L_k, a set of **candidate** k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k. Let l_1 and l_2 be itemsets in L_{k-1}. The notation $l_i[j]$ refers to the jth item in l_i (e.g., $l_1[k-2]$ refers to the second to the last item in l_1). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the $(k-1)$-itemset, l_i, this means that the items are sorted such that $l_i[1] < l_i[2] < \ldots < l_i[k-1]$. The join, $L_{k-1} \bowtie L_{k-1}$, is performed, where members of L_{k-1} are joinable if their first $(k-2)$ items are in common. That is, members l_1 and l_2 of L_{k-1} are joined if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$. The condition $l_1[k-1] < l_2[k-1]$ simply ensures that no duplicates are generated. The resulting itemset formed by joining l_1 and l_2 is $l_1[1], l_1[2], \ldots, l_1[k-2], l_1[k-1], l_2[k-1]$.

Apriori property used in algorithm
2. The prune step

2. The prune step: C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_k). C_k, however, can be huge, and so this could involve heavy computation. To reduce the size of C_k, the Apriori property is used as follows. Any $(k-1)$-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any $(k-1)$-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k. This **subset testing** can be done quickly by maintaining a hash tree of all frequent itemsets.

Transactional data for an *AllElectronics* branch

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Example: Apriori

- Let's look at a concrete example, based on the *AllElectronics transaction database, D.*

- *There are nine transactions in this database, that is, |D| = 9.*

- Apriori algorithm for finding frequent itemsets in *D*

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Example: Apriori Algorithm

Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Example: Apriori Algorithm
$C_2 \rightarrow L_2$

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Example: Apriori Algorithm
$C_3 \rightarrow L_3$

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

The Apriori algorithm for discovering frequent itemsets for mining Boolean association rules.

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D, a database of transactions;
- min_sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)      L₁ = find_frequent_1-itemsets(D);
(2)      for (k = 2; Lₖ₋₁ ≠ φ; k++) {
(3)          Cₖ = apriori_gen(Lₖ₋₁);
(4)          for each transaction t ∈ D { // scan D for counts
(5)              Cₜ = subset(Cₖ, t); // get the subsets of t that are candidates
(6)              for each candidate c ∈ Cₜ
(7)                  c.count++;
(8)          }
(9)          Lₖ = {c ∈ Cₖ | c.count ≥ min_sup}
(10)     }
(11)     return L = ∪ₖ Lₖ;
```

procedure apriori_gen(L_{k-1}:frequent $(k-1)$-itemsets)

```
(1)      for each itemset l₁ ∈ Lₖ₋₁
(2)          for each itemset l₂ ∈ Lₖ₋₁
(3)              if (l₁[1] = l₂[1]) ∧ (l₁[2] = l₂[2]) ∧ ... ∧ (l₁[k−2] = l₂[k−2]) ∧ (l₁[k−1] < l₂[k−1]) then {
(4)                  c = l₁ ⋈ l₂; // join step: generate candidates
(5)                  if has_infrequent_subset(c, Lₖ₋₁) then
(6)                      delete c; // prune step: remove unfruitful candidate
(7)                  else add c to Cₖ;
(8)              }
(9)      return Cₖ;
```

procedure has_infrequent_subset(c: candidate k-itemset;
            L_{k-1}: frequent $(k-1)$-itemsets); // use prior knowledge

```
(1)      for each (k−1)-subset s of c
(2)          if s ∉ Lₖ₋₁ then
(3)              return TRUE;
(4)      return FALSE;
```

The Apriori Algorithm—An Example

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

2nd scan

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2
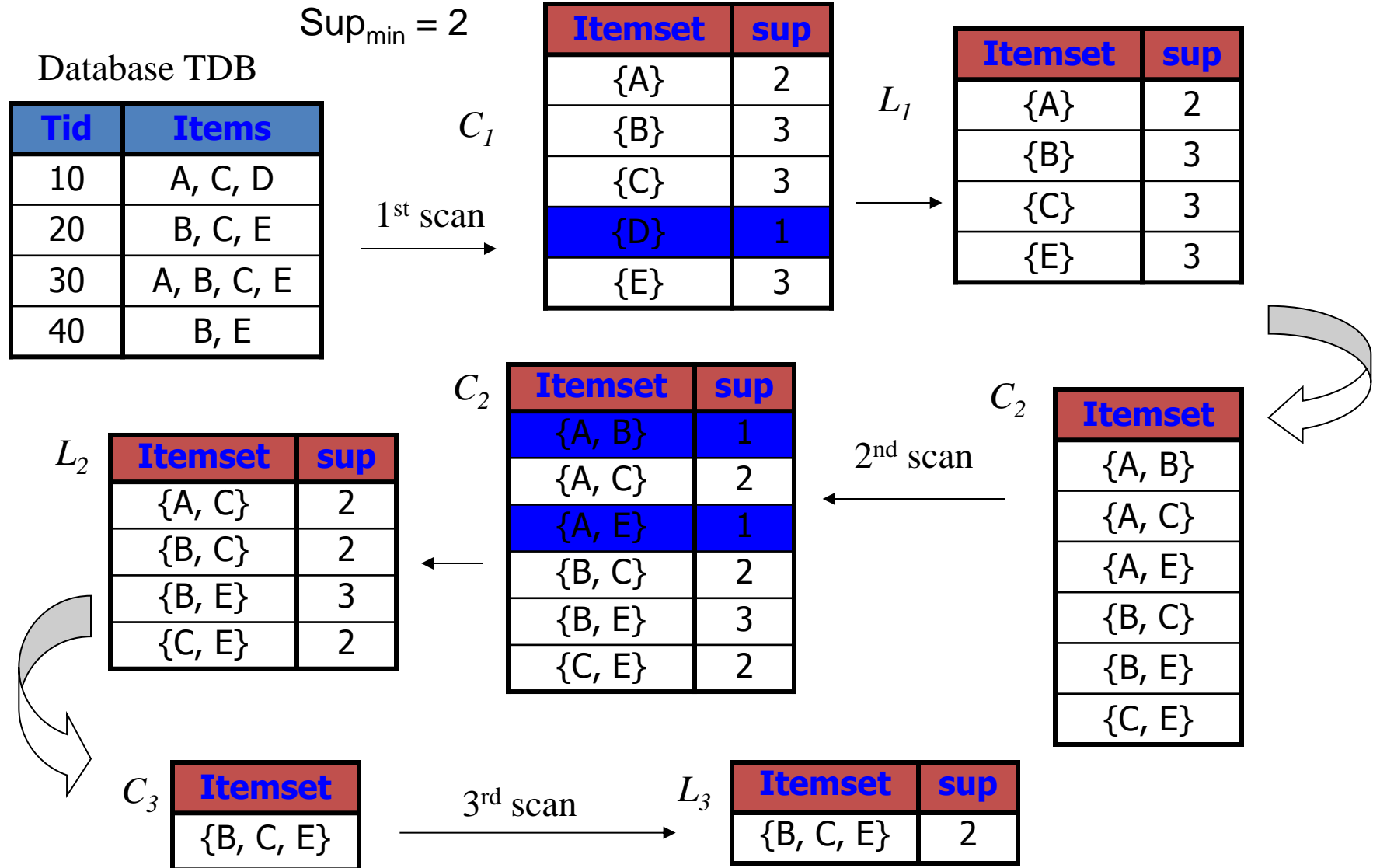
The Apriori Algorithm

- Pseudo-code:

  C_k: Candidate itemset of size k

  L_k : frequent itemset of size k

  L_1 = {frequent items};

  for (k = 1; L_k !=$\varnothing$; k++) **do begin**

     C_{k+1} = candidates generated from L_k;

      for each transaction t in database do

          increment the count of all candidates in C_{k+1}
  that are contained in t

     L_{k+1} = candidates in C_{k+1} with min_support

     end

  return $\cup_k L_k$;

Generating Association Rules from Frequent Itemsets

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}$$

- For each frequent itemset l, generate all nonempty subsets of l.

- For every nonempty subset s of l, output the rule "$s \Rightarrow (l - s)$" if $\frac{support\_count(l)}{support\_count(s)} \geq$ min_conf, where min_conf is the minimum confidence threshold.

Example:
Generating association rules

- frequent itemset $l = \{I1, I2, I5\}$

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

$$I1 \wedge I2 \Rightarrow I5, \qquad confidence = 2/4 = 50\%$$
$$I1 \wedge I5 \Rightarrow I2, \qquad confidence = 2/2 = 100\%$$
$$I2 \wedge I5 \Rightarrow I1, \qquad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow I2 \wedge I5, \qquad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow I1 \wedge I5, \qquad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow I1 \wedge I2, \qquad confidence = 2/2 = 100\%$$

- If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong.

Classification Techniques

- Decision tree analysis
- Statistical analysis
- Neural networks
- Support vector machines
- Case-based reasoning
- Bayesian classifiers
- Genetic algorithms
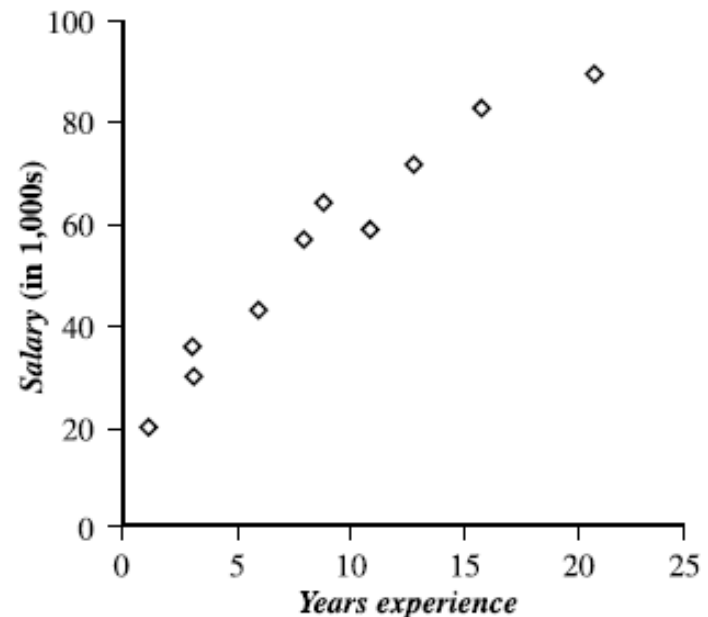- Rough sets

Example of Classification

- Loan Application Data
  - Which loan applicants are "safe" and which are "risky" for the bank?
  - "Safe" or "risky" for load application data
- Marketing Data
  - Whether a customer with a given profile will buy a new computer?
  - "yes" or "no" for marketing data
- **Classification**
  - Data analysis task
  - A model or **Classifier** is constructed to predict categorical labels
    - Labels: "safe" or "risky"; "yes" or "no"; "treatment A", "treatment B", "treatment C"

Prediction Methods

- Linear Regression

- Nonlinear Regression

- Other Regression Methods

Salary data.

x years experience	y salary (in $1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Classification and Prediction

- Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.

- Classification

  – Effective and scalable methods have been developed for decision trees induction, Naive Bayesian classification, Bayesian belief network, rule-based classifier, Backpropagation, Support Vector Machine (SVM), associative classification, nearest neighbor classifiers, and case-based reasoning, and other classification methods such as genetic algorithms, rough set and fuzzy set approaches.

- Prediction

  – Linear, nonlinear, and generalized linear models of regression can be used for prediction. Many nonlinear problems can be converted to linear problems by performing transformations on the predictor variables. Regression trees and model trees are also used for prediction.

Classification—A Two-Step Process

1. Model construction: describing a set of predetermined classes
   - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
   - The set of tuples used for model construction is training set
   - The model is represented as classification rules, decision trees, or mathematical formulae
2. Model usage: for classifying future or unknown objects
   - Estimate accuracy of the model
     - The known label of test sample is compared with the classified result from the model
     - Accuracy rate is the percentage of test set samples that are correctly classified by the model
     - Test set is independent of training set, otherwise over-fitting will occur
   - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**

  – Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations

  – New data is classified based on the training set

- **Unsupervised learning (clustering)**

  – The class labels of training data is unknown

  – Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Issues Regarding Classification and Prediction: Data Preparation

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
  - Attribute subset selection
    - Feature Selection in machine learning
- Data transformation
  - Generalize and/or normalize data
  - Example
    - Income: low, medium, high
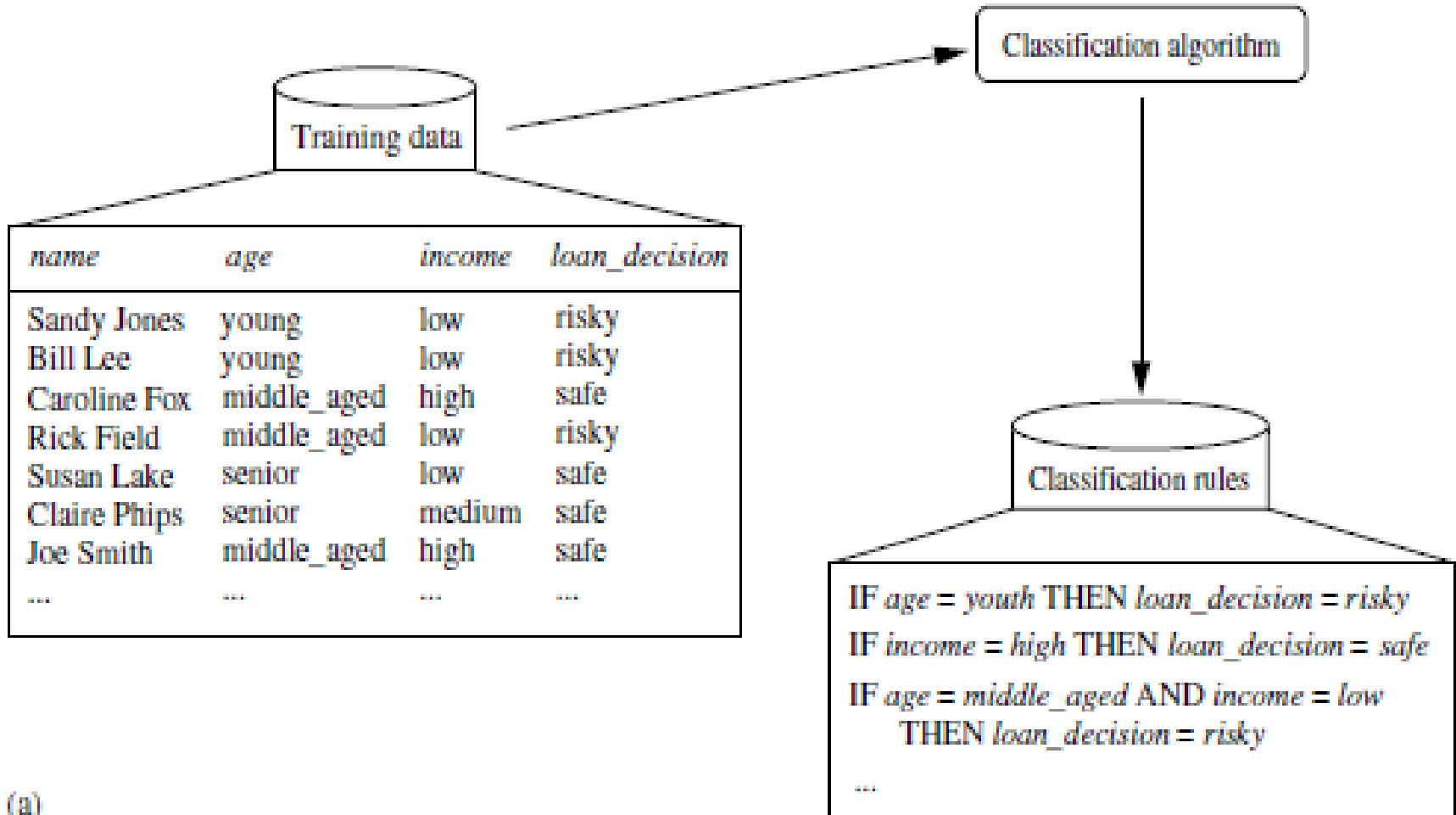
Issues:
Evaluating Classification and Prediction Methods

- **Accuracy**
  - classifier accuracy: predicting class label
  - predictor accuracy: guessing value of predicted attributes
  - estimation techniques: cross-validation and bootstrapping
- Speed
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- Robustness
  - handling noise and missing values
- Scalability
  - ability to construct the classifier or predictor efficiently given large amounts of data
- Interpretability
  - understanding and insight provided by the model

Data Classification Process 1: Learning (Training) Step (a) Learning: Training data are analyzed by classification algorithm
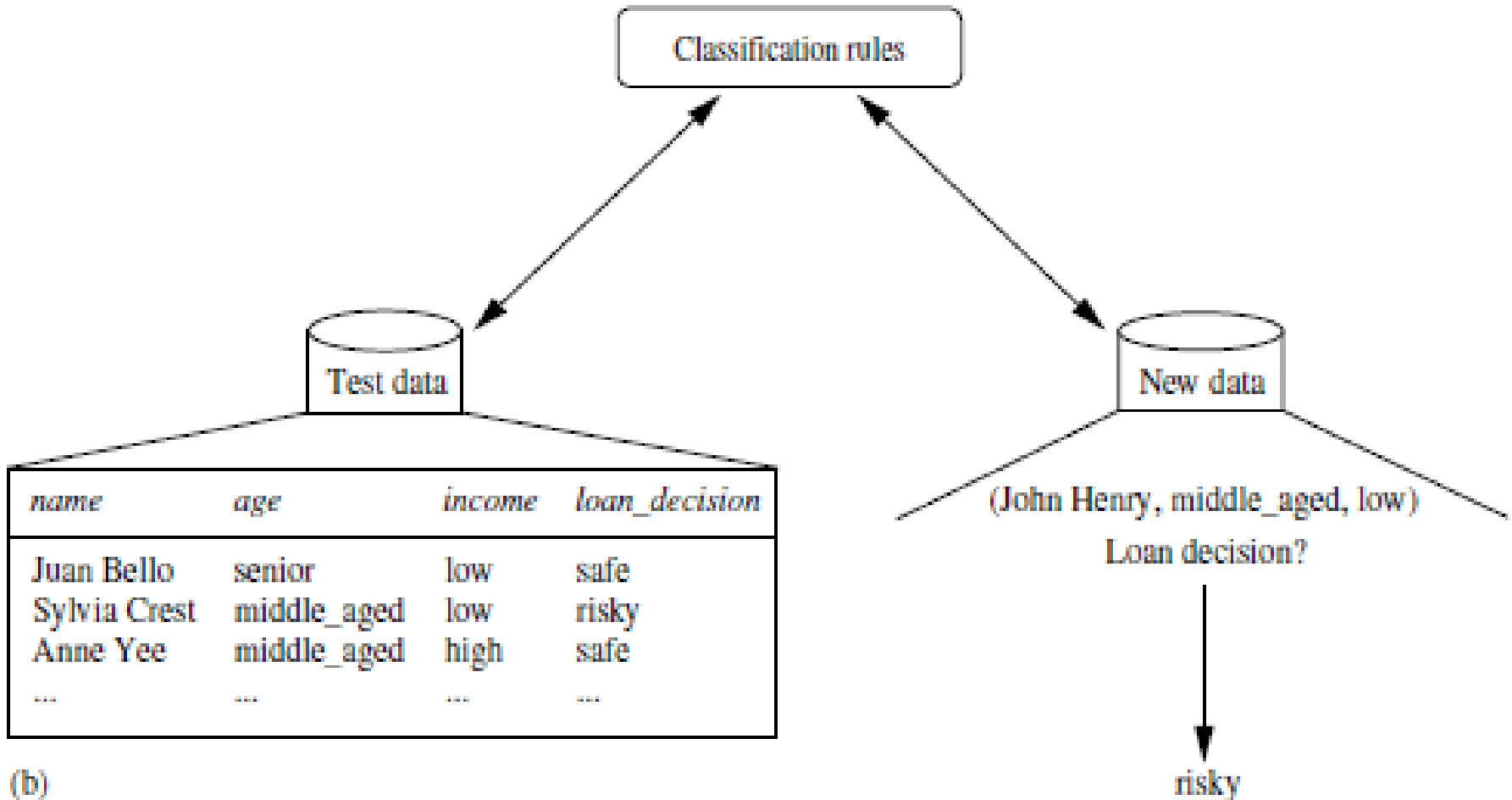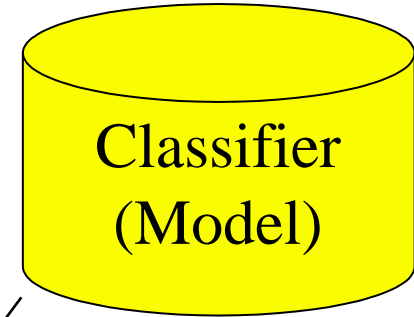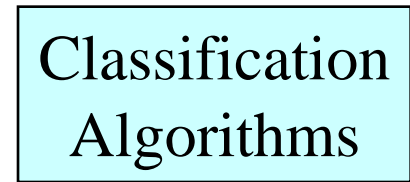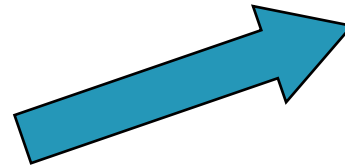
$$y = f(X)$$

name	age	income	loan_decision
Sandy Jones	young	low	risky
Bill Lee	young	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

Classification rules

IF *age* = *youth* THEN *loan_decision* = *risky*

IF *income* = *high* THEN *loan_decision* = *safe*

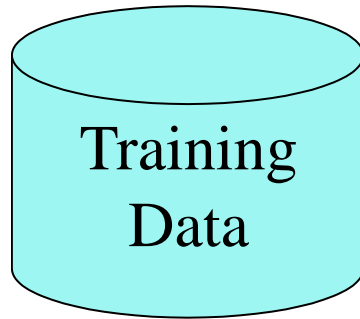IF *age* = *middle_aged* AND *income* = *low*
    THEN *loan_decision* = *risky*

...

(a)

Data Classification Process 2
(b) Classification: Test data are used to estimate the accuracy of the classification rules.

Classification rules
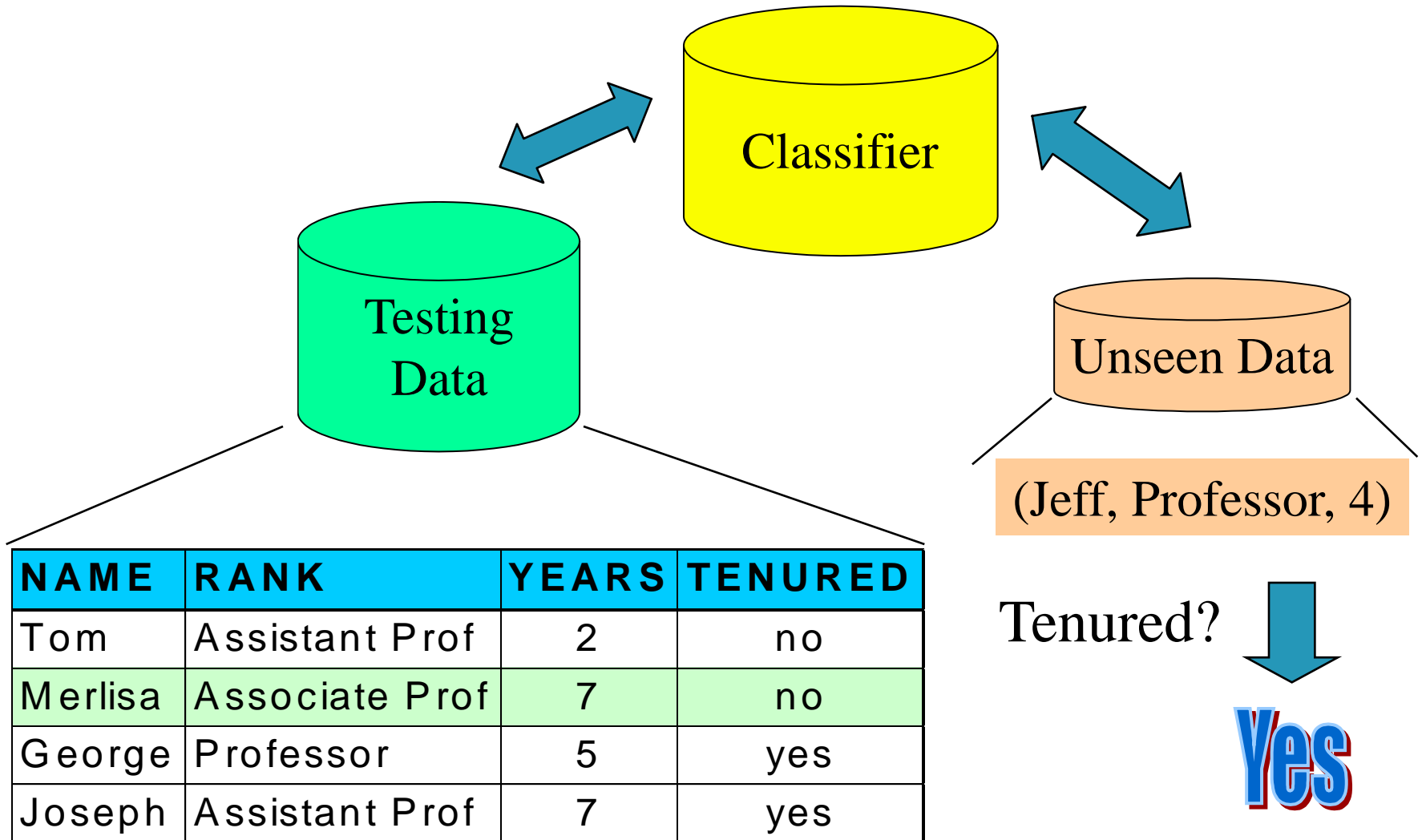
Test data

New data

name	age	income	loan_decision
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

(b)

(John Henry, middle_aged, low)

Loan decision?

risky

Process (1): Model Construction

Training Data

Classification Algorithms

Classifier (Model)

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Process (2): Using the Model in Prediction

Classifier

Testing Data

Unseen Data

(Jeff, Professor, 4)

NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

Yes

Decision Trees

A general algorithm for decision tree building

- Employs the divide and conquer method
- Recursively divides a training set until each division consists of examples from one class
  1. Create a root node and assign all of the training data to it
  2. Select the best splitting attribute
  3. Add a branch to the root node for each value of the split. Split the data into mutually exclusive subsets along the lines of the specific split
  4. Repeat the steps 2 and 3 for each and every leaf node until the stopping criteria is reached

Decision Trees

- DT algorithms mainly differ on
  - Splitting criteria
    - Which variable to split first?
    - What values to use to split?
    - How many splits to form for each node?
  - Stopping criteria
    - When to stop building the tree
  - Pruning (generalization method)
    - Pre-pruning versus post-pruning

- Most popular DT algorithms include
  - ID3, C4.5, C5; CART; CHAID; M5

Decision Trees

- Alternative splitting criteria
  - Gini index determines the purity of a specific class as a result of a decision to branch along a particular attribute/value
    - Used in CART
  - Information gain uses entropy to measure the extent of uncertainty or randomness of a particular attribute/value split
    - Used in ID3, C4.5, C5
  - Chi-square statistics (used in CHAID)

Classification by Decision Tree Induction Training Dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

This follows an example of Quinlan's ID3 (Playing Tennis)

Classification by Decision Tree Induction

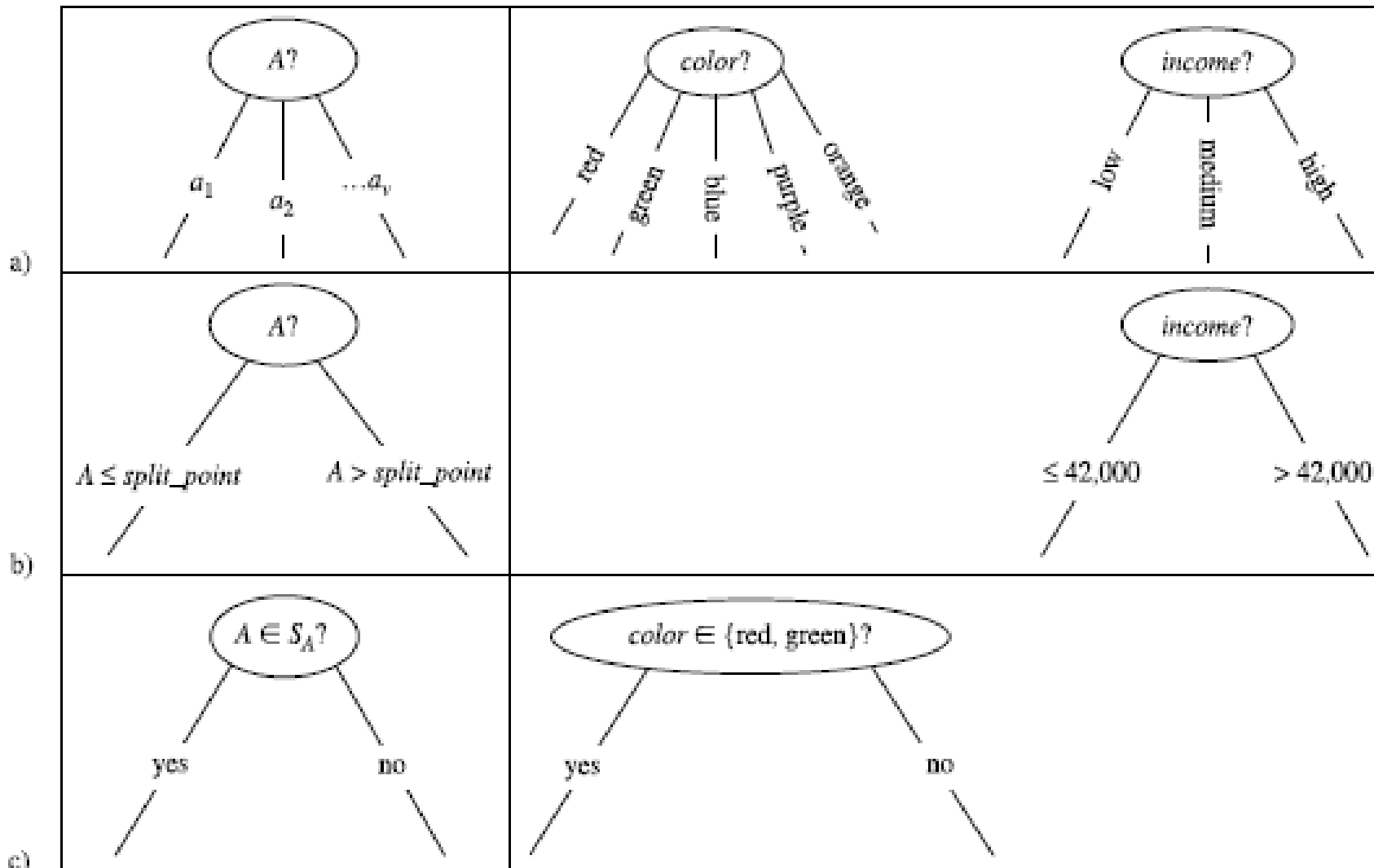Output: A Decision Tree for *"buys_computer"*
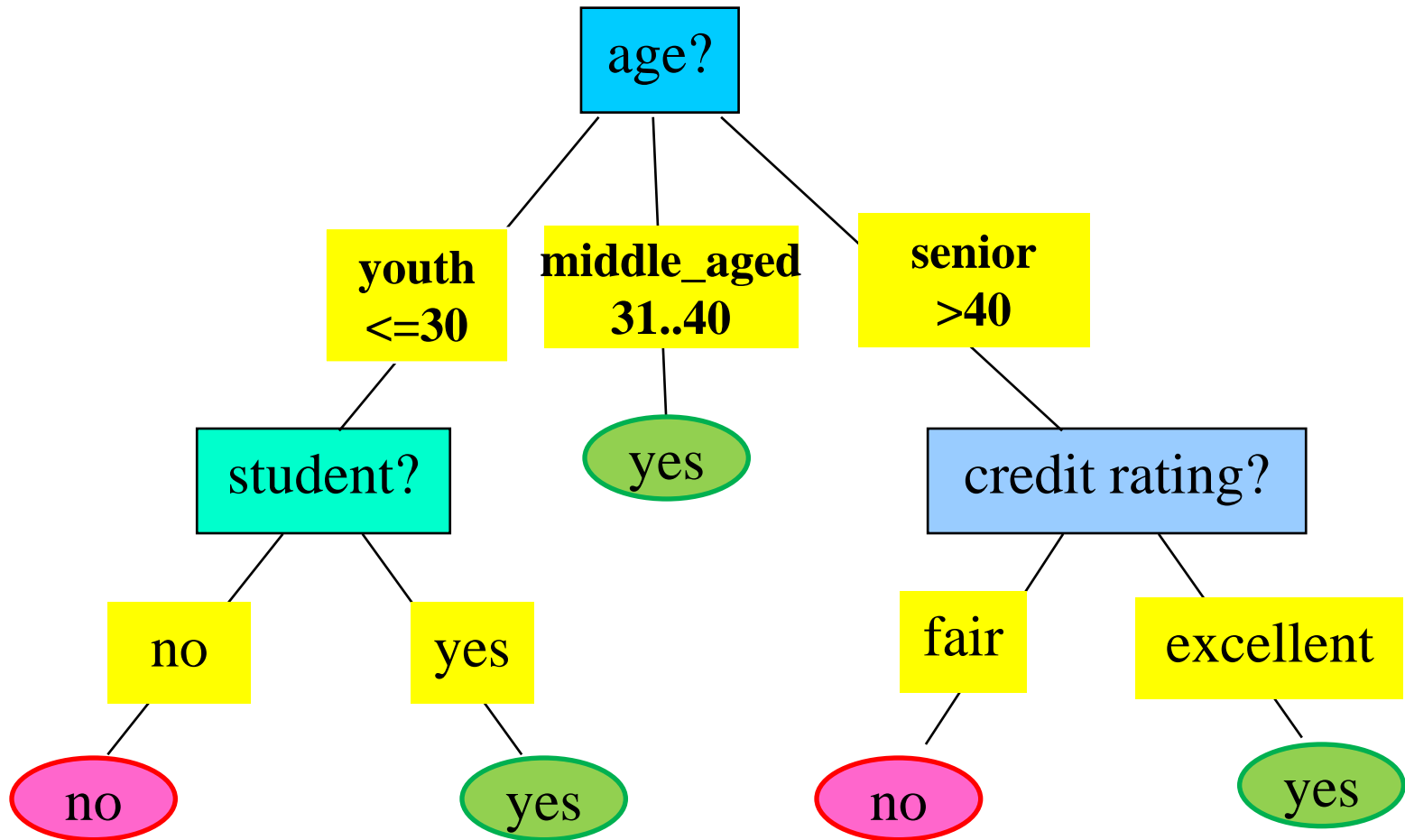
buys_computer="yes" or buys_computer="no"

Three possibilities for partitioning tuples based on the splitting Criterion

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
    - Tree is constructed in a top-down recursive divide-and-conquer manner
    - At start, all the training examples are at the root
    - Attributes are categorical (if continuous-valued, they are discretized in advance)
    - Examples are partitioned recursively based on selected attributes
    - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
    - All samples for a given node belong to the same class
    - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
    - There are no samples left

Attribute Selection Measure

- Notation: Let *D, the data partition, be a training set of* class-labeled tuples.
  Suppose the class label attribute has m distinct values defining m distinct classes, C_i *(for i = 1, ... , m).*
  Let $C_{i,D}$ be the set of tuples of class C_i in D.
  Let |D| and | $C_{i,D}$ | denote the number of tuples in D and $C_{i,D}$, respectively.

- *Example:*
  - *Class: buys_computer= "yes" or "no"*
  - *Two distinct classes (m=2)*
    - *Class C_i (i=1,2):*
      C_1 = "yes",
      C_2 = "no"

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain

- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$

- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

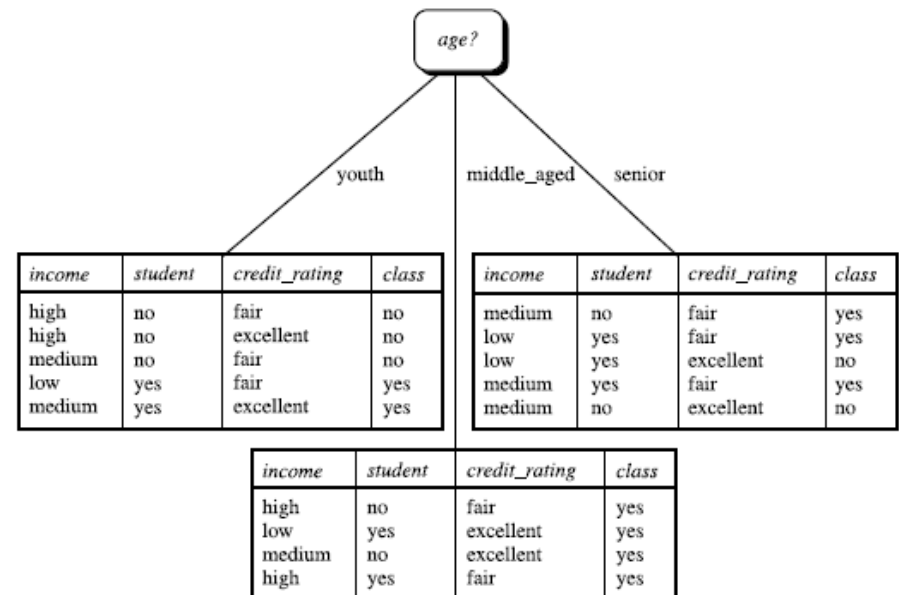- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Class-labeled training tuples from the *AllElectronics customer database*

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

age?

youth:

income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged:

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior:

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

The attribute age has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0)$$

$$+ \frac{5}{14}I(3,2) = 0.694$$

age	p_i	n_i	I(p_i, n_i)
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values

- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$$

  – GainRatio(A) = Gain(A)/SplitInfo(A)

- Ex. $$SplitInfo_A(D) = -\frac{4}{14} \times \log_2(\frac{4}{14}) - \frac{6}{14} \times \log_2(\frac{6}{14}) - \frac{4}{14} \times \log_2(\frac{4}{14}) = 0.926$$

  – gain_ratio(income) = 0.029/0.926 = 0.031

- The attribute with the maximum gain ratio is selected as the splitting attribute

Trees

- A "divisive" method (splits)
- Start with "root node" – all in one group
- Get splitting rules
- Response often binary
- Result is a "tree"
- Example: Loan Defaults
- Example: Framingham Heart Study
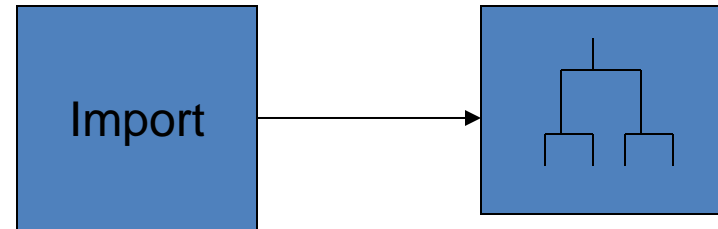- Example: Automobile fatalities

Recursive Splitting

X1=Debt To Income Ratio

Pr{default} =0.008

Pr{default} =0.012

Pr{default} =0.006

Pr{default} =0.0001

Pr{default} =0.003
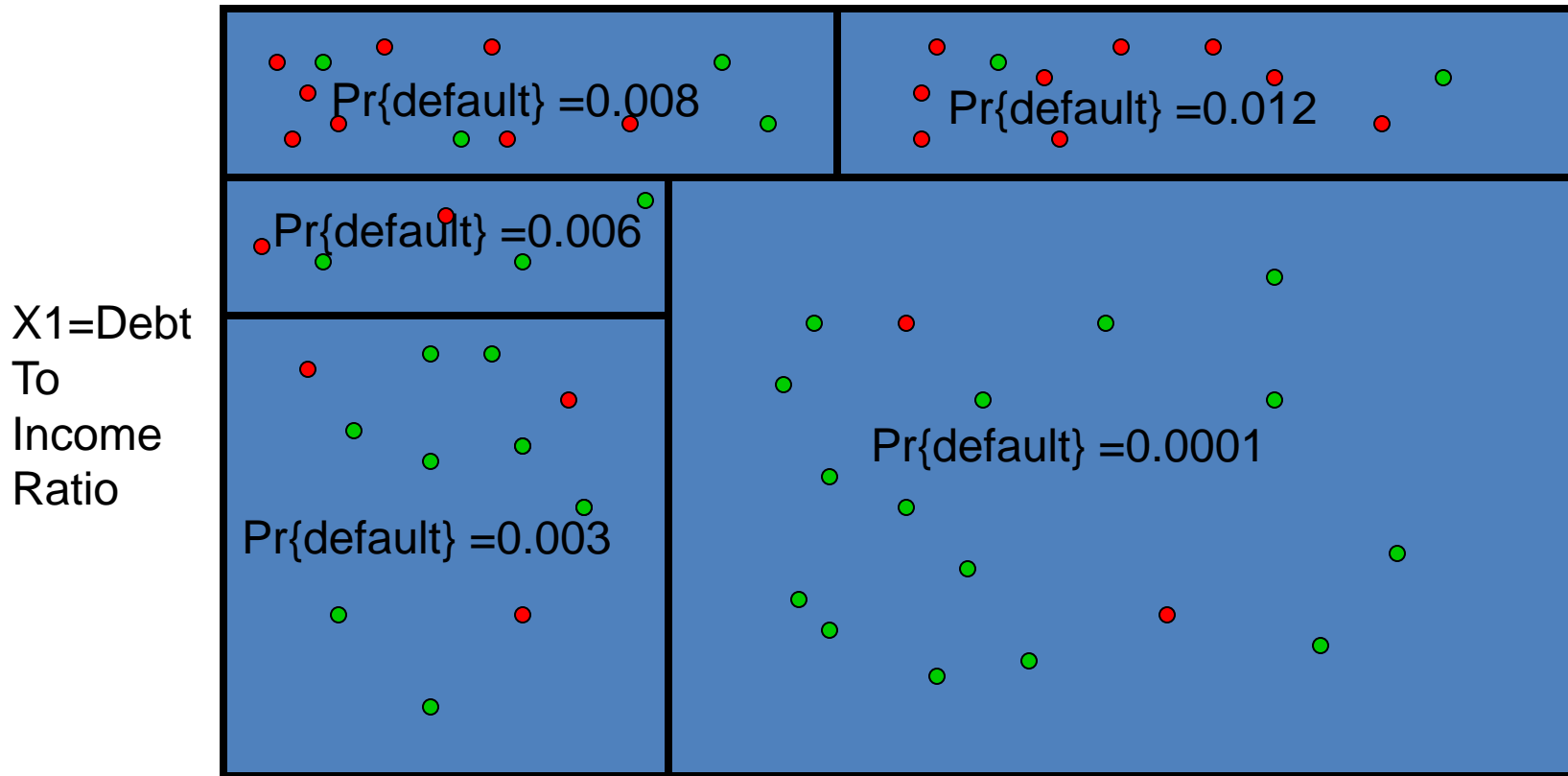
● No default

● Default

X2 = Age

Some Actual Data

- Framingham Heart Study

- First Stage Coronary Heart Disease
  - P{CHD} = Function of:
    - Age - no drug yet! ☹
    - Cholesterol
    - Systolic BP
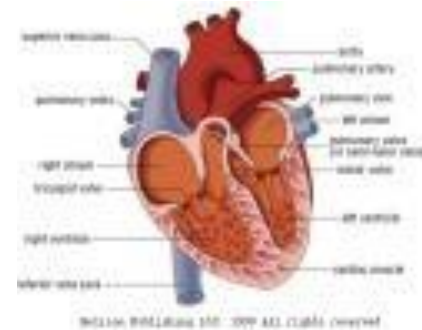
Example of a "tree"

All 1615 patients

Split # 1: Age

"terminal node"
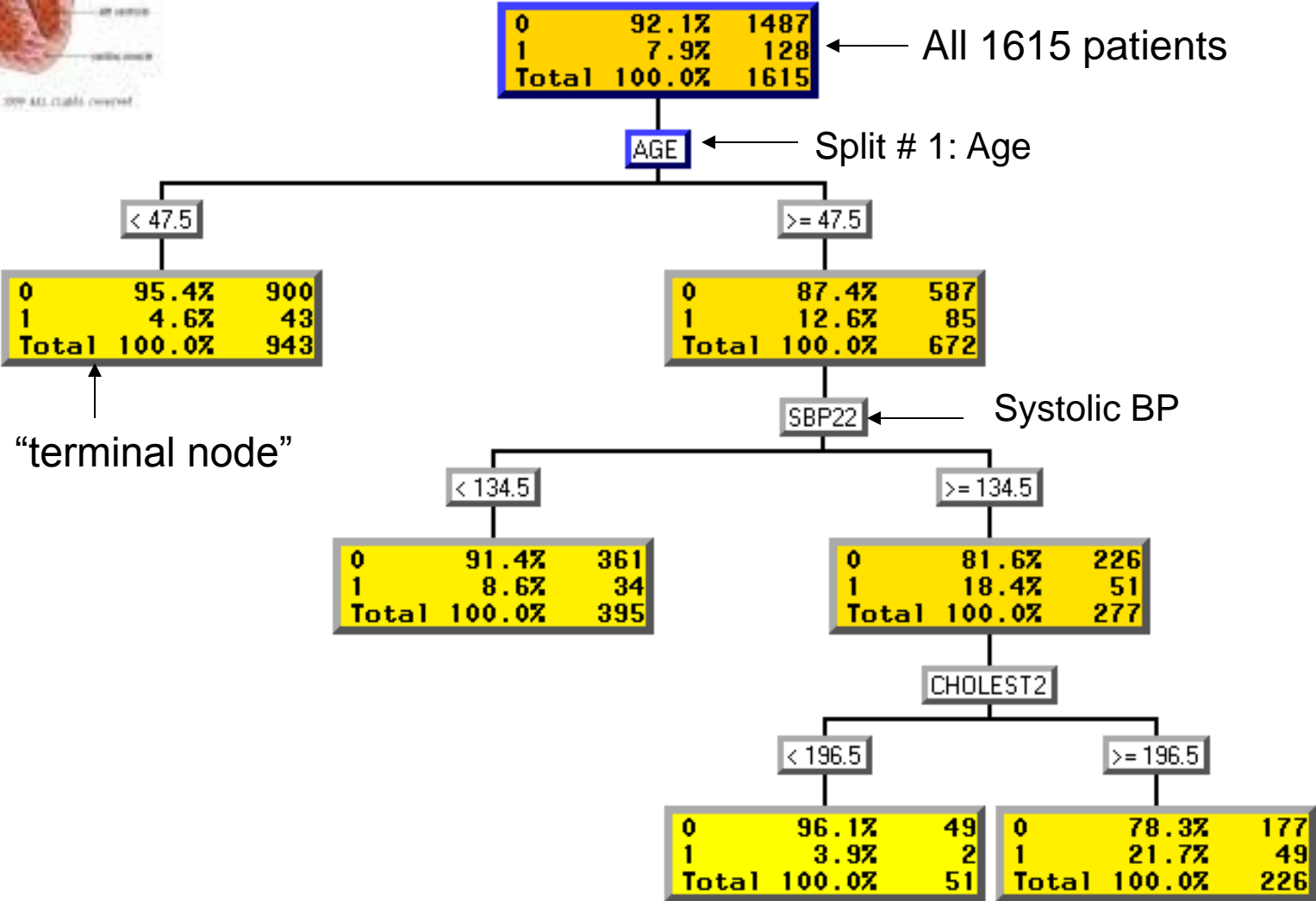
Systolic BP

How to make splits?

- Which variable to use?

- Where to split?
  - Cholesterol > _____
  - Systolic BP > _____

- Goal: Pure "leaves" or "terminal nodes"

- Ideal split: Everyone with BP>x has problems, nobody with BP<x has problems

First review Chi Square test

- Contingency tables

Heart Disease

	No	Yes	
Low BP	95	5	100
High BP	55	45	100

DEPENDENT (yes)

Heart Disease

	No	Yes
	75	25
	75	25

INDEPENDENT (no)

χ^2 Test Statistic

- Expect 100(150/200)=75 in upper left
  if independent (etc. e.g. 100(50/200)=25)

Heart Disease

	No	Yes	
Low BP	95 (75)	5 (25)	100
High BP	55 (75)	45 (25)	100
	150	50	200

$$\chi^2 = \sum_{allcells} \frac{(observed - \exp ected)^2}{\exp ected}$$

2(400/75)+
2(400/25) =
42.67

Compare to
Tables –
Significant!
(Significant ???)

WHERE IS HIGH BP CUTOFF???

Conclusion: Sufficient evidence *against* the hypothesis of no relationship.

H_0: ♥
H_1: ✕

H_0: Innocence
H_1: Guilt

Beyond reasonable doubt
$P < 0.05$

95	5
(75)	(25)
55	45
(75)	(25)

H_0: No association
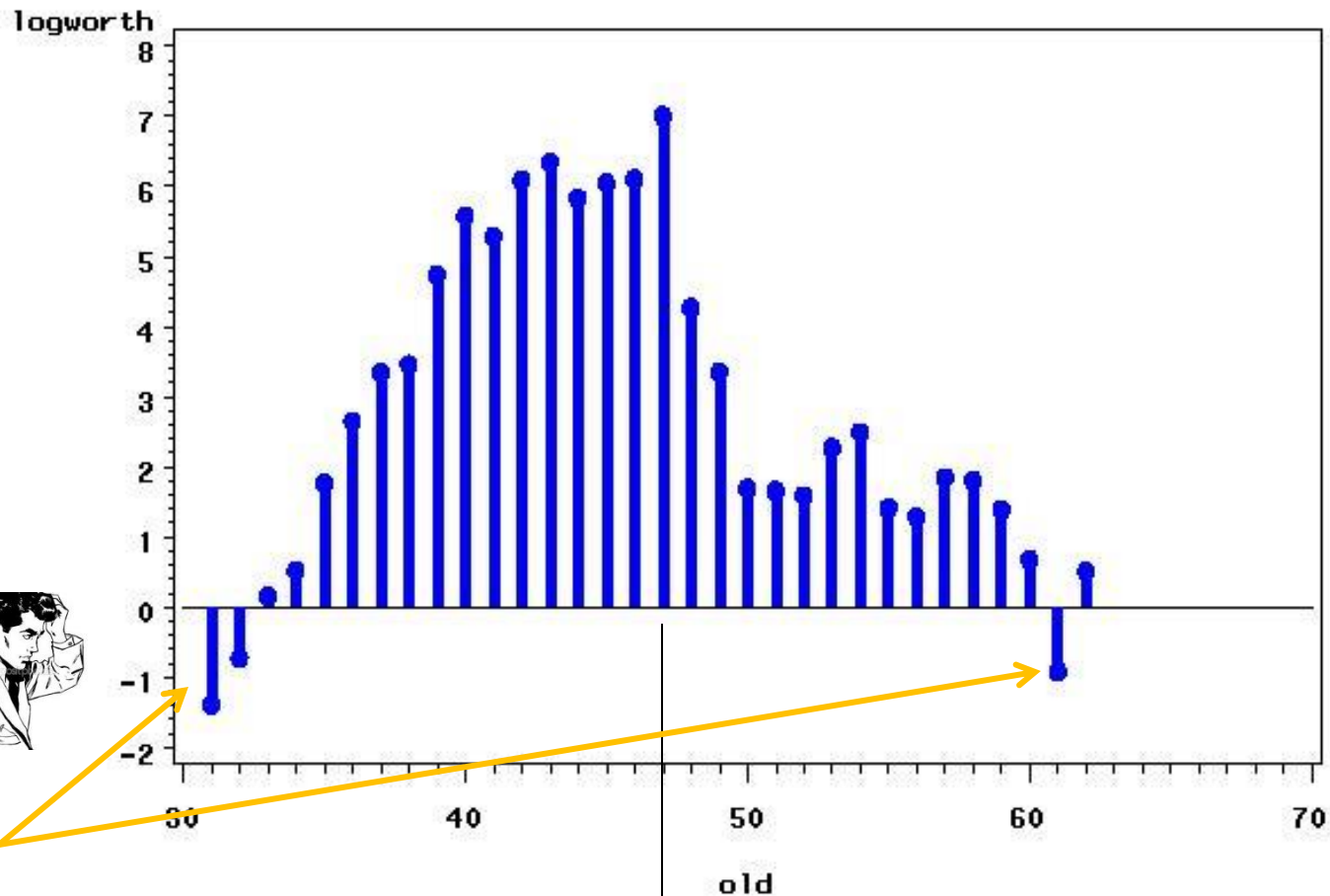H_1: BP and heart disease are associated

$P = 0.00000000064$

Measuring "Worth" of a Split

- P-value is probability of Chi-square as great as that observed if independence is true. (Pr $\{\chi^2 > 42.67\}$ is 6.4E-11)

- P-values all too small.

- Logworth = $-\log_{10}$(p-value) = 10.19

- Best Chi-square ←→ max logworth.

Logworth for Age Splits

Framingham Data

Age 47 maximizes logworth

How to make splits?

- Which variable to use?
- Where to split?
  - Cholesterol > _____
  - Systolic BP > _____
- Idea – <u>*Pick*</u> BP cutoff to minimize p-value for χ^2
- What does "signifiance" mean now?

Cluster Analysis

- Used for automatic identification of natural groupings of things

- Part of the machine-learning family

- Employ unsupervised learning

- Learns the clusters of things from past data, then assigns new instances

- There is not an output variable

- Also known as segmentation

Cluster Analysis
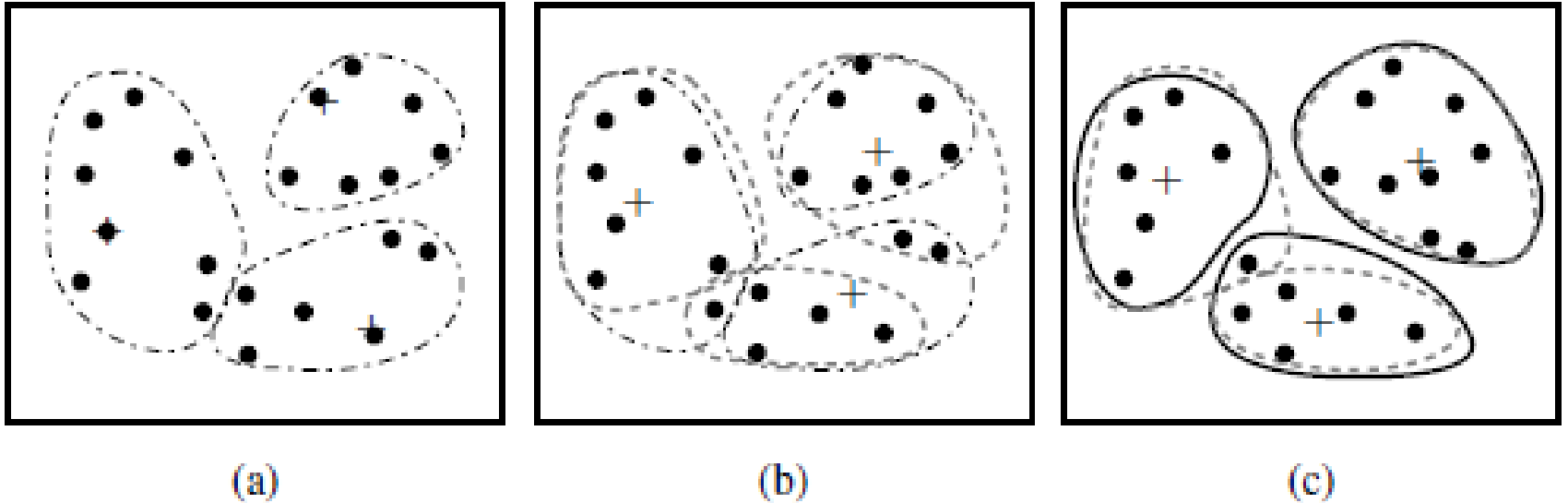
Clustering of a set of objects based on the *k-means method.*
(The mean of each cluster is marked by a "+".)

Cluster Analysis

- Clustering results may be used to
  - Identify natural groupings of customers
  - Identify rules for assigning new cases to classes for targeting/diagnostic purposes
  - Provide characterization, definition, labeling of populations
  - Decrease the size and complexity of problems for other data mining methods
  - Identify outliers in a specific domain (e.g., rare-event detection)

Example of Cluster Analysis
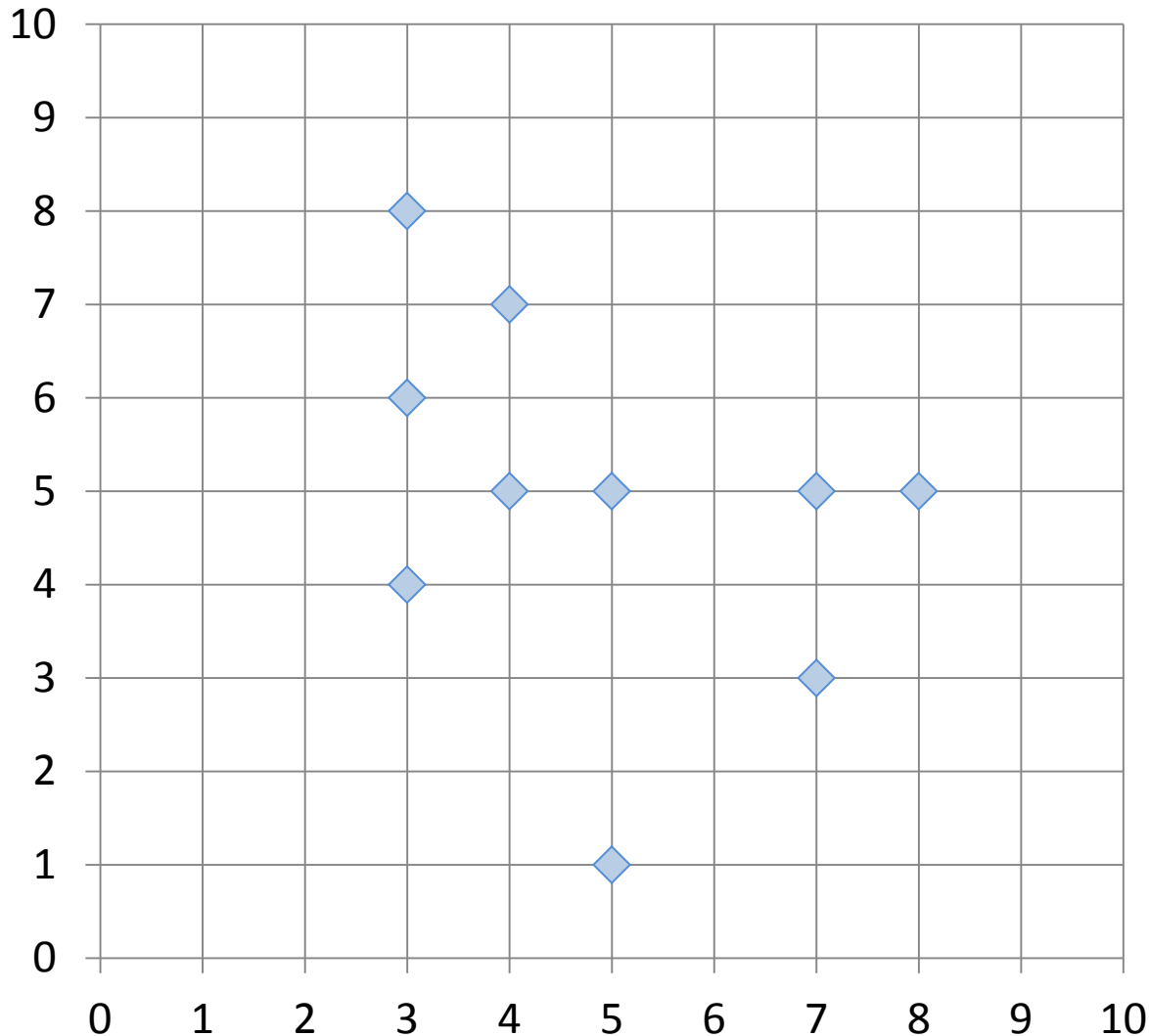
Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Cluster Analysis for Data Mining

- Analysis methods
  - Statistical methods
    (including both hierarchical and nonhierarchical),
    such as *k*-means, *k*-modes, and so on
  - Neural networks
    (adaptive resonance theory [ART],
    self-organizing map [SOM])
  - Fuzzy logic (e.g., fuzzy c-means algorithm)
  - Genetic algorithms

- Divisive versus Agglomerative methods

Cluster Analysis for Data Mining

- ## How many clusters?
  - There is not a "truly optimal" way to calculate it
  - Heuristics are often used
    1. Look at the sparseness of clusters
    2. Number of clusters = $(n/2)^{1/2}$ (n: no of data points)
    3. Use Akaike information criterion (AIC)
    4. Use Bayesian information criterion (BIC)

- Most cluster analysis methods involve the use of a distance measure to calculate the closeness between pairs of items
  - Euclidian versus Manhattan (rectilinear) distance

k-Means Clustering Algorithm

- *k* : pre-determined number of clusters
- Algorithm (Step 0: determine value of *k*)

Step 1: Randomly generate *k* random points as initial cluster centers

Step 2: Assign each point to the nearest cluster center

Step 3: Re-compute the new cluster centers

Repetition step: Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable)

Cluster Analysis for Data Mining - *k*-Means Clustering Algorithm

Step 1

Step 2

Step 3
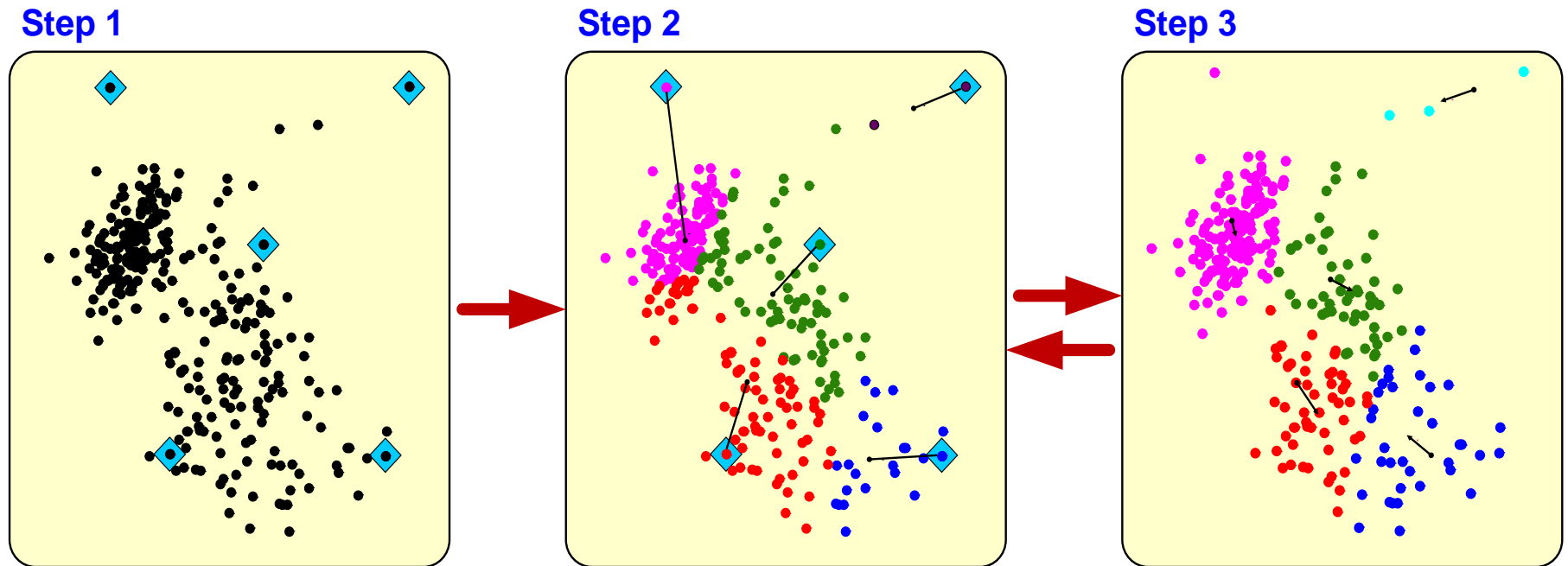
Similarity and Dissimilarity Between Objects

- <u>Distances</u> are normally used to measure the <u>similarity</u> or <u>dissimilarity</u> between two data objects

- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q)}$$

  where $i = (x_{i1}, x_{i2}, ..., x_{ip})$ and $j = (x_{j1}, x_{j2}, ..., x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If *q = 1, d* is Manhattan distance

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}|$$

Similarity and Dissimilarity Between Objects (Cont.)

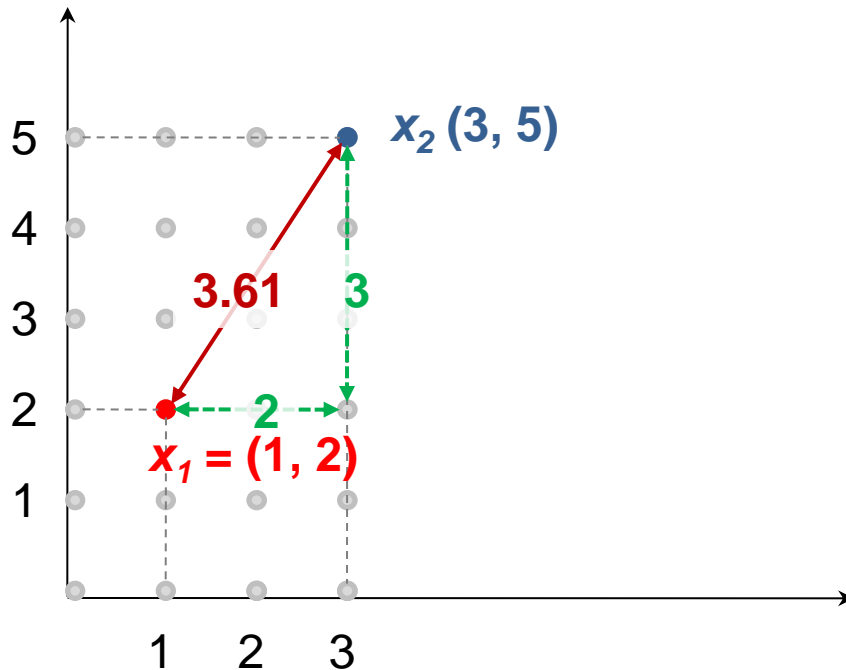- *If q = 2, d* is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + ... + |x_{ip} - x_{jp}|^2)}$$

  – Properties
    - *d(i,j)* $\geq$ 0
    - *d(i,i)* = 0
    - *d(i,j)* = *d(j,i)*
    - *d(i,j)* $\leq$ *d(i,k)* + *d(k,j)*

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures

Euclidean distance vs Manhattan distance

- Distance of two point $x_1 = (1, 2)$ and x_2 $(3, 5)$
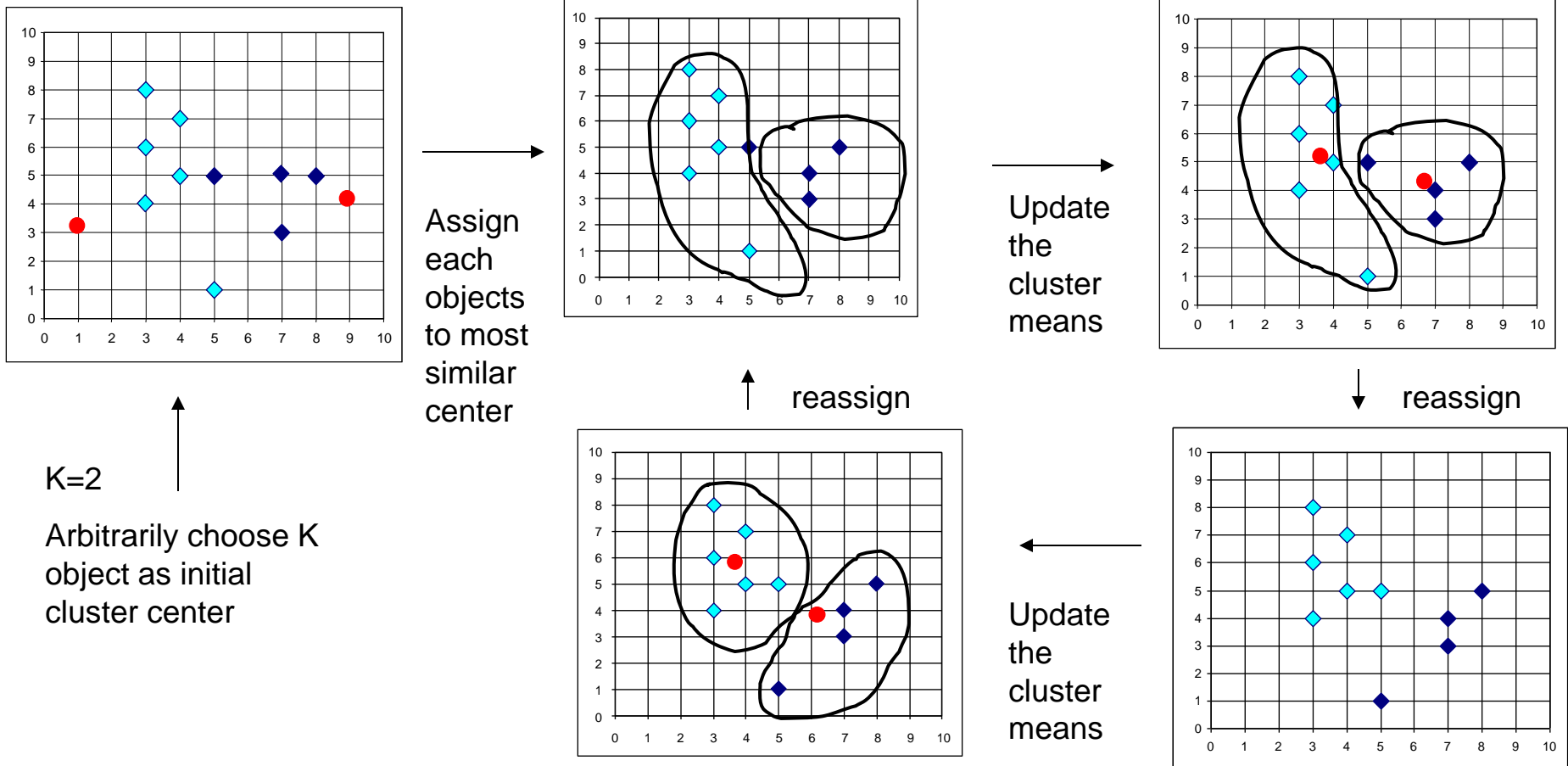

Euclidean distance:
$$= ((3-1)^2 + (5-2)^2)^{1/2}$$
$$= (2^2 + 3^2)^{1/2}$$
$$= (4 + 9)^{1/2}$$
$$= (13)^{1/2}$$
$$= 3.61$$

Manhattan distance:
$$= (3-1) + (5-2)$$
$$= 2 + 3$$
$$= 5$$

The *K-Means* Clustering Method

- Example

K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

Update the cluster means

reassign

reassign

K-Means Clustering
Step by Step
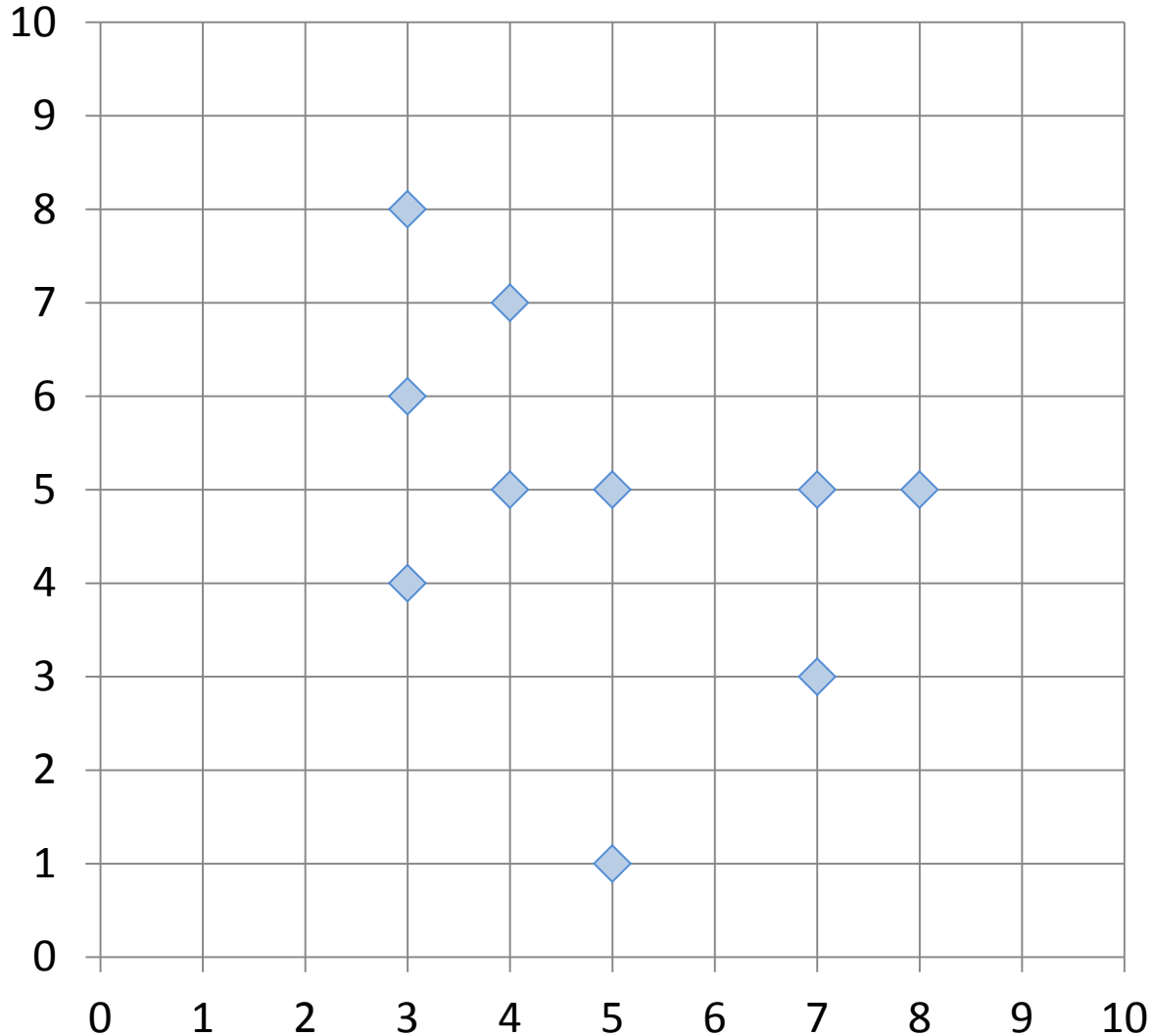
Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

Step 1: K=2, Arbitrarily choose K object as initial cluster center
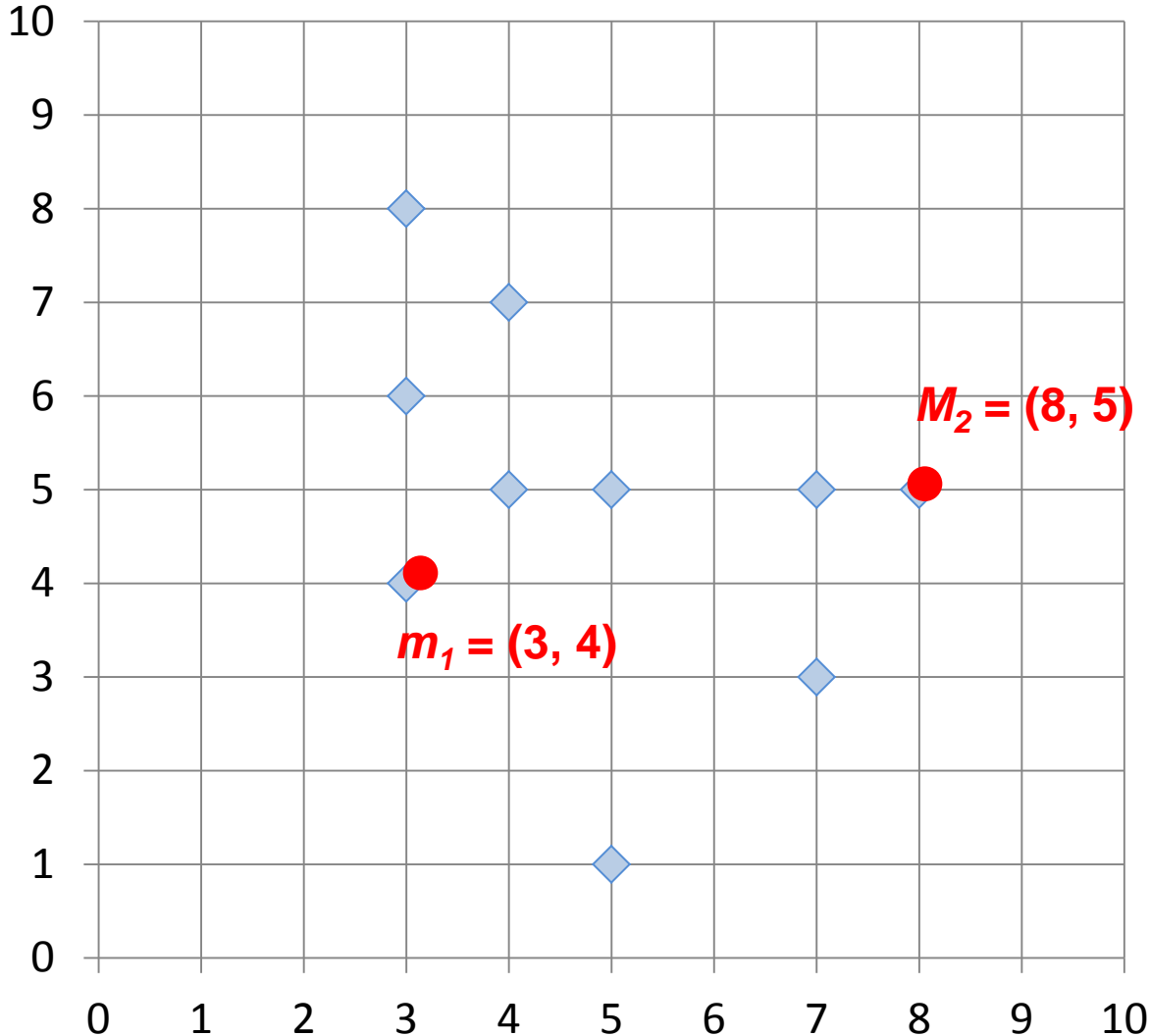
Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Initial	m1	(3, 4)
Initial	m2	(8, 5)

On the graph: $M_2 = (8, 5)$ and $m_1 = (3, 4)$

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center

$M_2 = (8, 5)$

$m_1 = (3, 4)$

K-Means Clustering
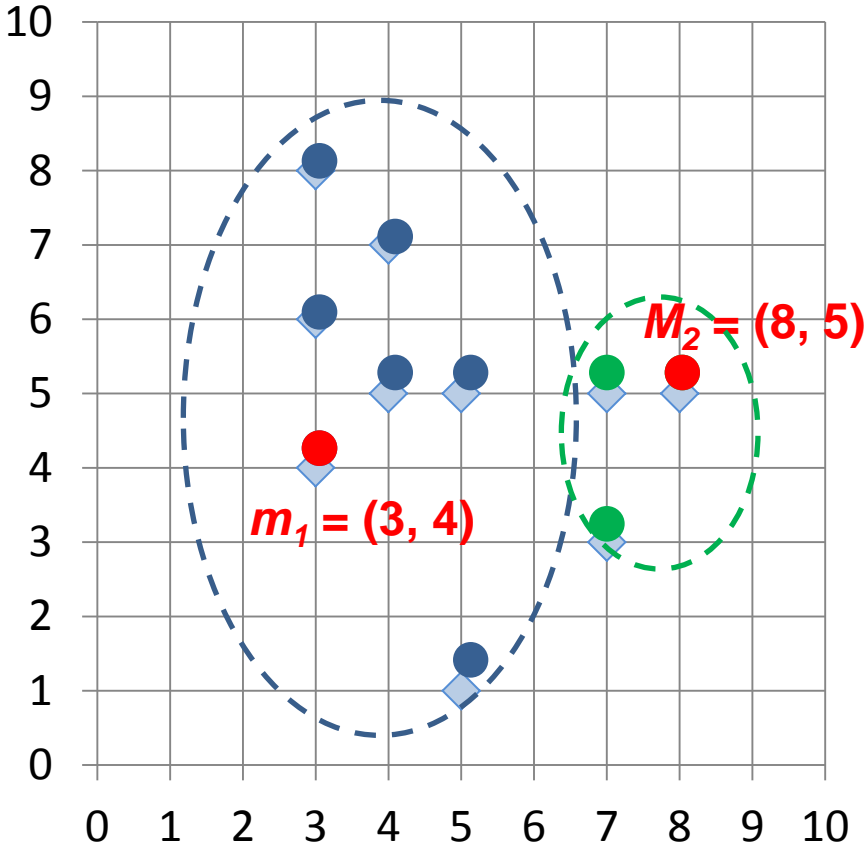
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05	e	(4, 7)	3.16	4.47	Cluster1
p06	f	(5, 1)	3.61	5.00	Cluster1
p07	g	(5, 5)	2.24	3.00	Cluster1
p08	h	(7, 3)	4.12	2.24	Cluster2
p09	i	(7, 5)	4.12	1.00	Cluster2
p10	j	(8, 5)	5.10	0.00	Cluster2

Step 2: Compute seed points as the centroids of the clusters of the current partition
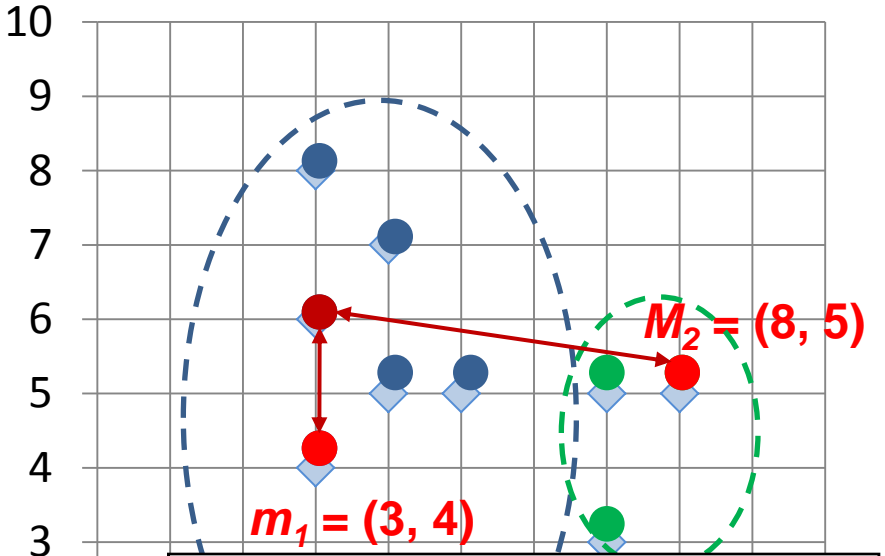
Step 3: Assign each objects to most similar center

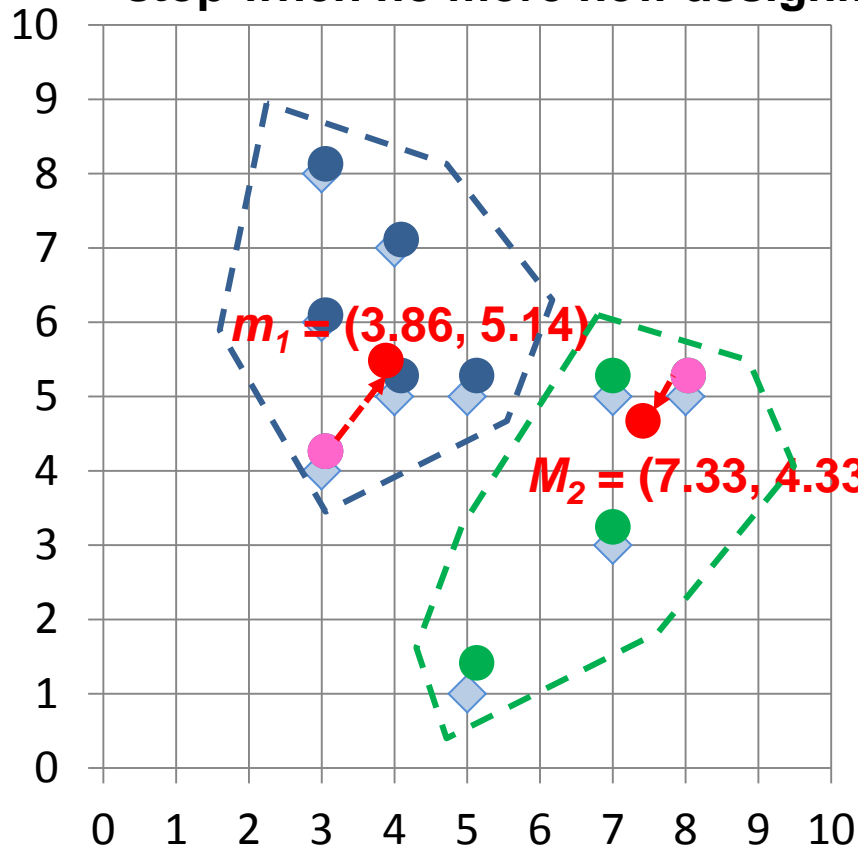| Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p08	h	(7, 3)	4.12	2.24	Cluster2
p09	i	(7, 5)	4.12	1.00	Cluster2
p10	j	(8, 5)	5.10	0.00	Cluster2

$M_2 = (8, 5)$

$m_1 = (3, 4)$

Euclidean distance
b(3,6) ⬅➡m1(3,4)
$= ((3-3)^2 + (4-6)^2)^{1/2}$
$= (0^2 + (-2)^2)^{1/2}$
$= (0 + 4)^{1/2}$
$= (4)^{1/2}$
$= 2.00$

K-

Euclidean distance
b(3,6) ⬅➡m2(8,5)
$= ((8-3)^2 + (5-6)^2)^{1/2}$
$= (5^2 + (-1)^2)^{1/2}$
$= (25 + 1)^{1/2}$
$= (26)^{1/2}$
$= 5.10$

132

Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.43	4.34	Cluster1
p02	b	(3, 6)	1.22	4.64	Cluster1
p03	c	(3, 8)	2.99	5.68	Cluster1
p04	d	(4, 5)	0.20	3.40	Cluster1
p05	e	(4, 7)	1.87	4.27	Cluster1
p06	f	(5, 1)	4.29	4.06	Cluster2
p07	g	(5, 5)	1.15	2.42	Cluster1
p08	h	(7, 3)	3.80	1.37	Cluster2
p09	i	(7, 5)	3.14	0.75	Cluster2
p10	j	(8, 5)	4.14	0.95	Cluster2

$m_1 = (3.86, 5.14)$

$M_2 = (7.33, 4.33)$

K-Means **Clustering**

m1 (3.86, 5.14)

m2 (7.33, 4.33)

**Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment**

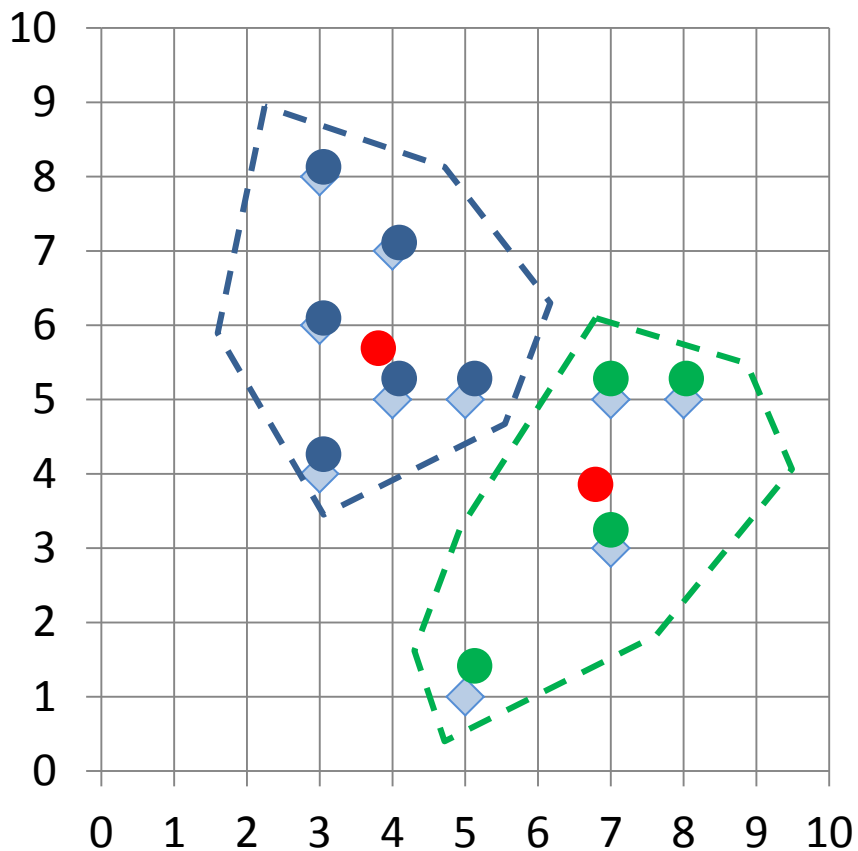| Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

K-Means Clustering

m1 (3.67, 5.83)

m2 (6.75, 3.50)

stop when no more new assignment

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

K-Means Clustering

m1 (3.67, 5.83)

m2 (6.75, 3.50)

135

stop when no more new assignment

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

m2 (6.75, 3.50)

Summary

- Define data mining as an enabling technology for business intelligence
- Standardized data mining processes
  - CRISP-DM
  - SEMMA
- Association Analysis
  - Association Rule Mining (Apriori Algorithm)
- Classification
  - Decision Tree
- Cluster Analysis
  - *K-Means* Clustering

References

- Efraim Turban, Ramesh Sharda, Dursun Delen, Decision Support and Business Intelligence Systems, Ninth Edition, 2011, Pearson.

- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, 2006, Elsevier