Data Warehousing
資料倉儲

Classification and Prediction, Cluster Analysis

992DW07
MI4
Tue. 8,9 (15:10-17:00) L413

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

http://mail.im.tku.edu.tw/~myday/

2011-05-03

Syllabus

1   100/02/15   Introduction to Data Warehousing

2   100/02/22   Data Warehousing, Data Mining, and Business Intelligence

3   100/03/01   Data Preprocessing: Integration and the ETL process

4   100/03/08   Data Warehouse and OLAP Technology

5   100/03/15   Data Warehouse and OLAP Technology

6   100/03/22   Data Warehouse and OLAP Technology

7   100/03/29   Data Warehouse and OLAP Technology

8   100/04/05   (放假一天) (民族掃墓節)

9   100/04/12   Data Cube Computation and Data Generation

10   100/04/19   Mid-Term Exam (期中考試週)

11   100/04/26   Association Analysis

12   100/05/03   Classification and Prediction, Cluster Analysis

13   100/05/10   Social Network Analysis, Link Mining, Text and Web Mining

14   100/05/17   Project Presentation

15   100/05/24   Final Exam (畢業班考試)

Classification and Prediction, Cluster Analysis

- Classification and Prediction
- Cluster Analysis

Classification vs. Prediction

- Classification
    - predicts categorical class labels (discrete or nominal)
    - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
    - models continuous-valued functions
        - i.e., predicts unknown or missing values
- Typical applications
    - Credit approval
    - Target marketing
    - Medical diagnosis
    - Fraud detection

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"

Classification Methods

- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Associative classification
- Lazy learners (or learning from your neighbors)
  - nearest neighbor classifiers
  - case-based reasoning
- Genetic Algorithms
- Rough Set Approaches
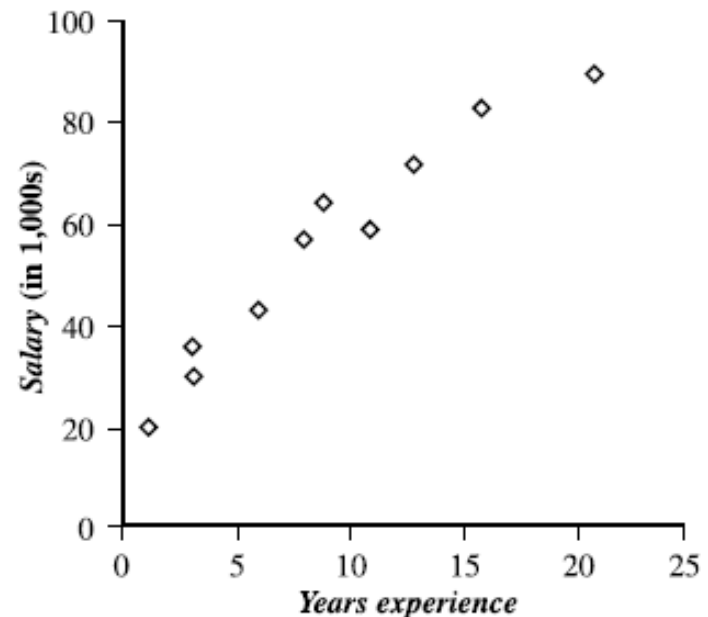- Fuzzy Set Approaches

What Is Prediction?

- (Numerical) prediction is similar to classification
  - construct a model
  - use model to predict continuous or ordered value for a given input
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or **predictor** variables and a *dependent* or **response** variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

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

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

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 and Prediction

- Stratified k-fold cross-validation is a recommended method for accuracy estimation. Bagging and boosting can be used to increase overall accuracy by learning and combining a series of individual models.

- Significance tests and ROC curves are useful for model selection

- There have been numerous comparisons of the different classification and prediction methods, and the matter remains a research topic

- No single method has been found to be superior over all others for all data sets

- Issues such as accuracy, training time, robustness, interpretability, and scalability must be considered and can involve trade-offs, further complicating the quest for an overall superior method
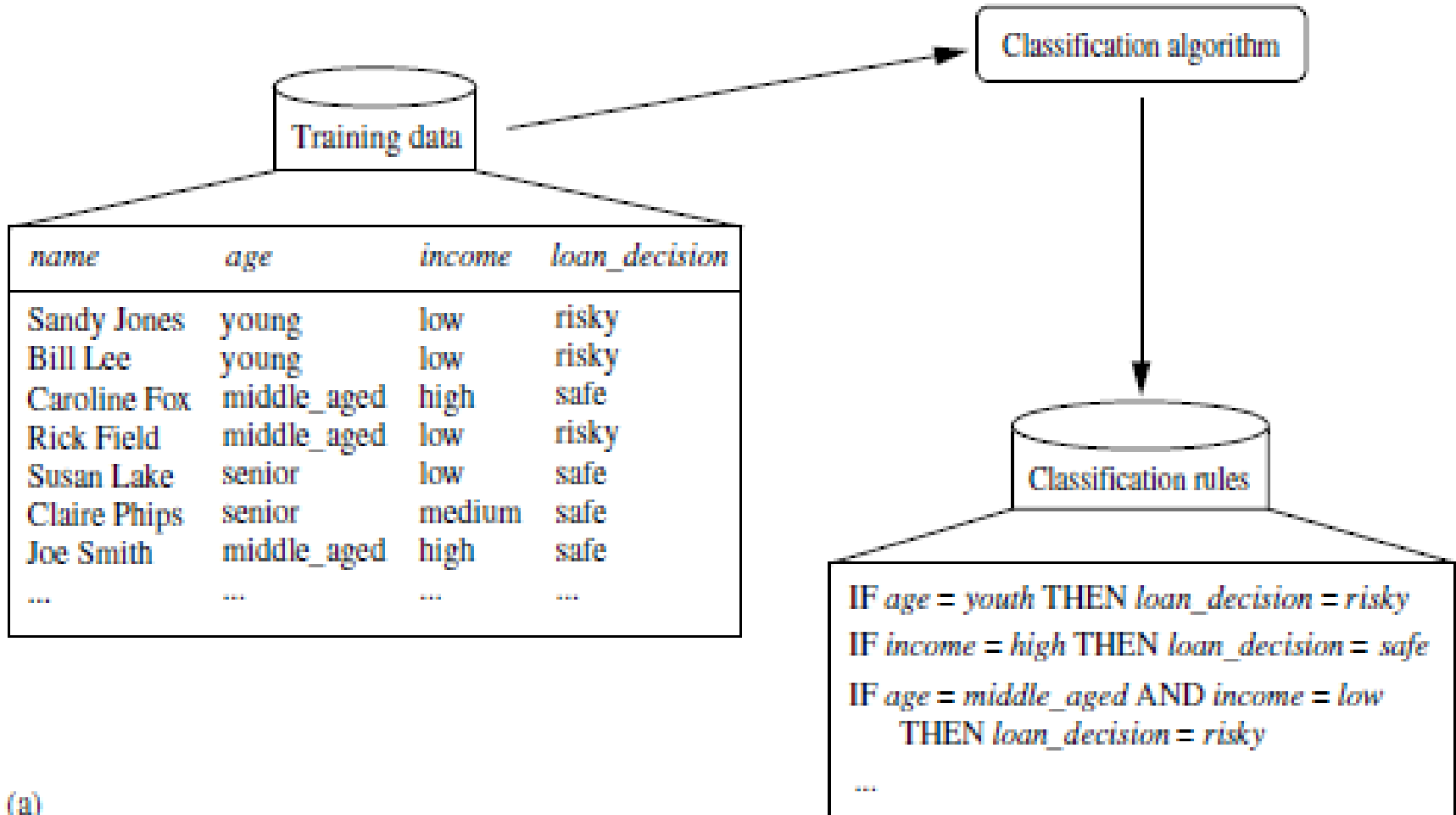
Classification—A Two-Step Process

- 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
- 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

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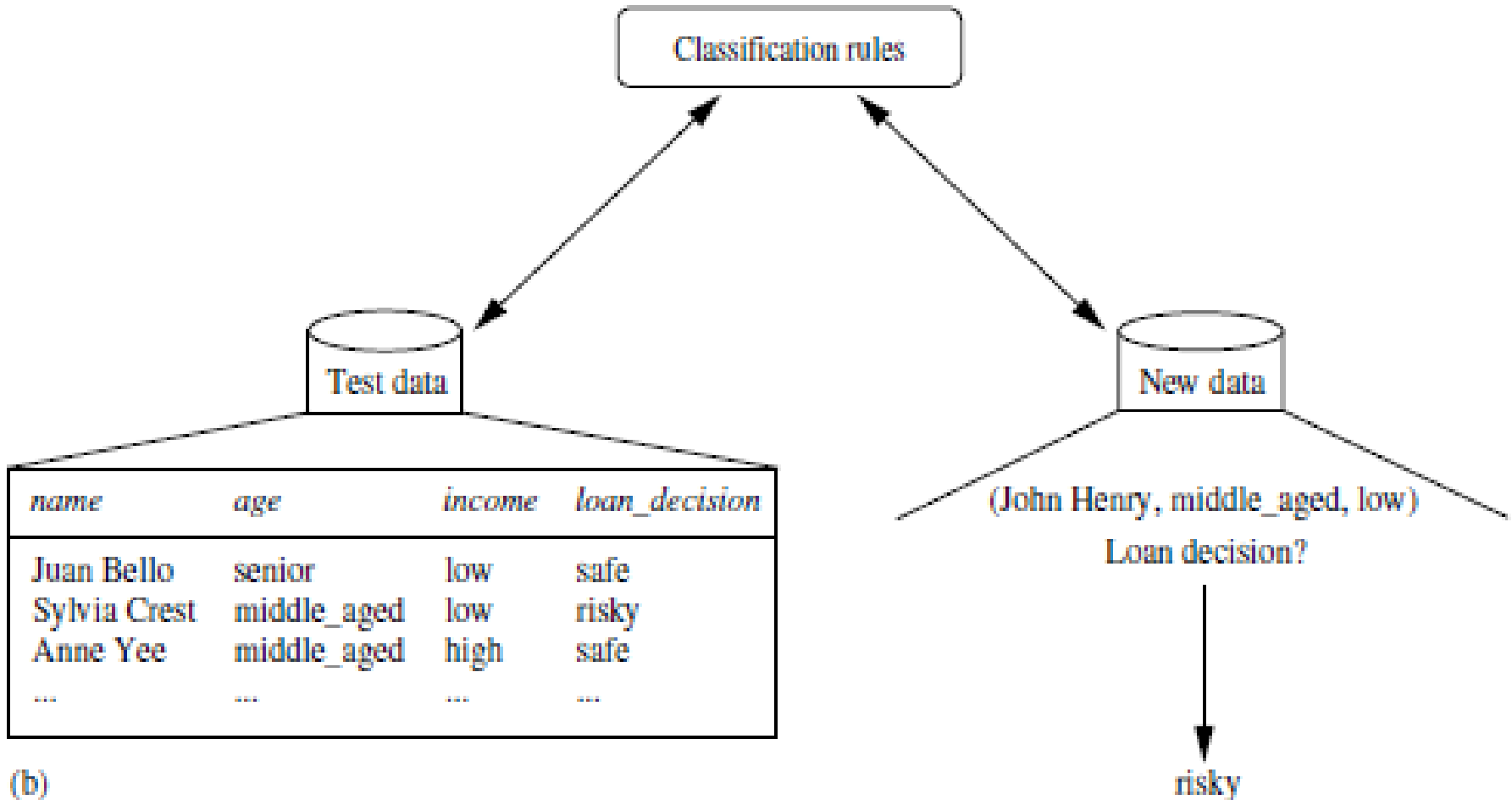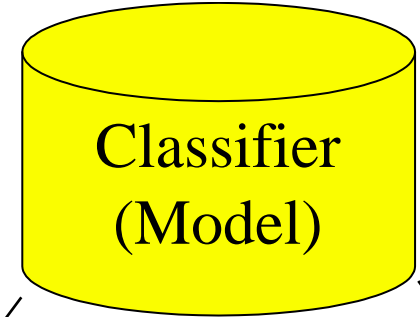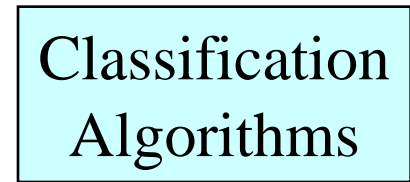
$$y = f(X)$$

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*

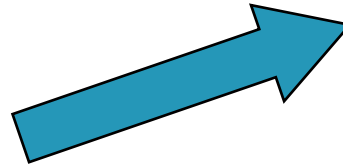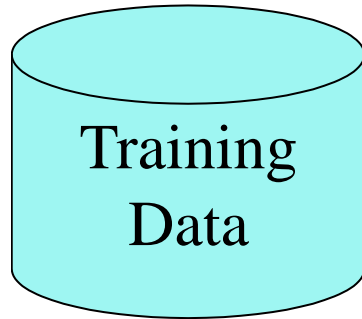IF *age* = *middle_aged* AND *income* = *low*
    THEN *loan_decision* = *risky*

...

(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
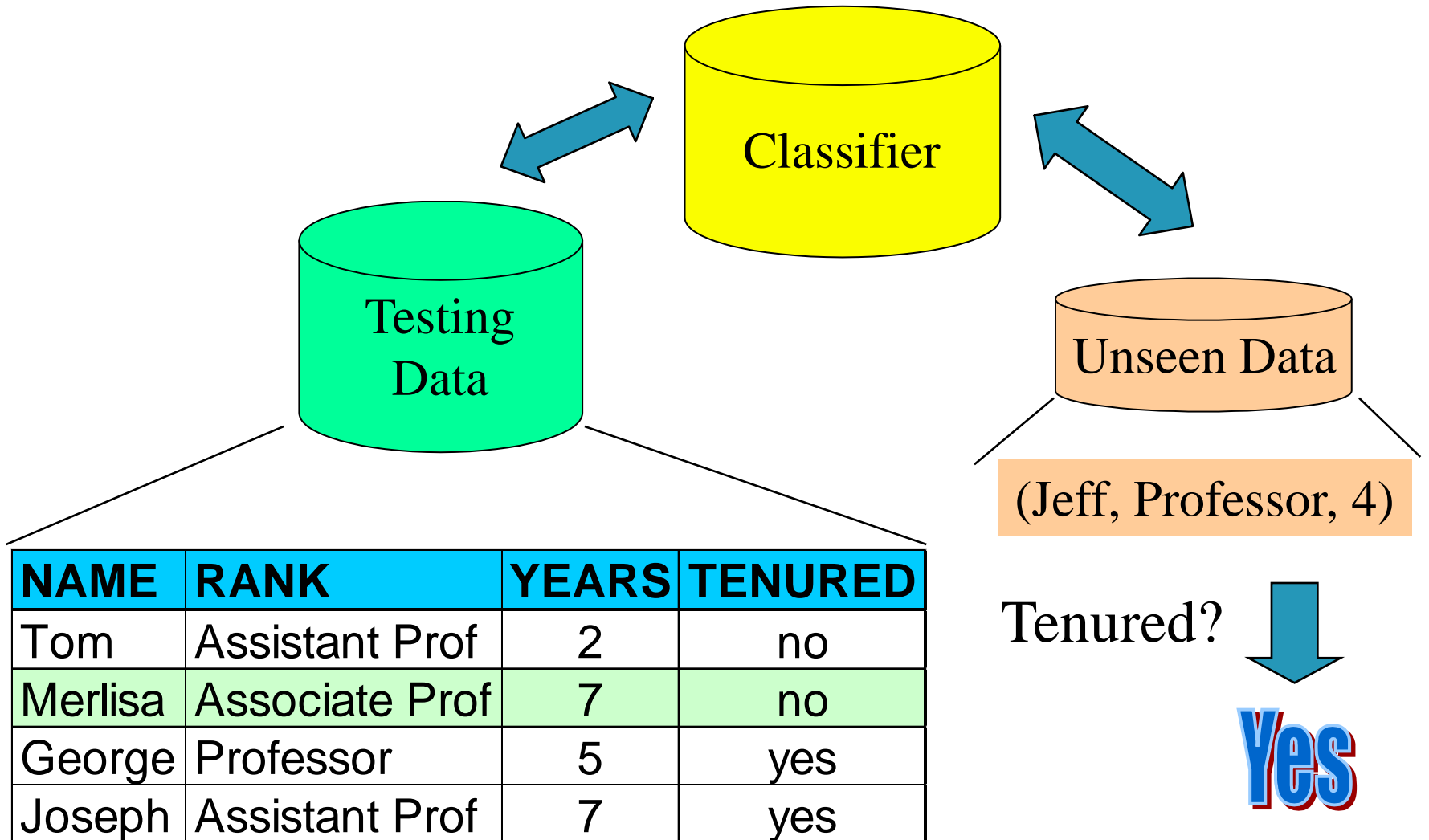
Supervised vs. Unsupervised Learning

- **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

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

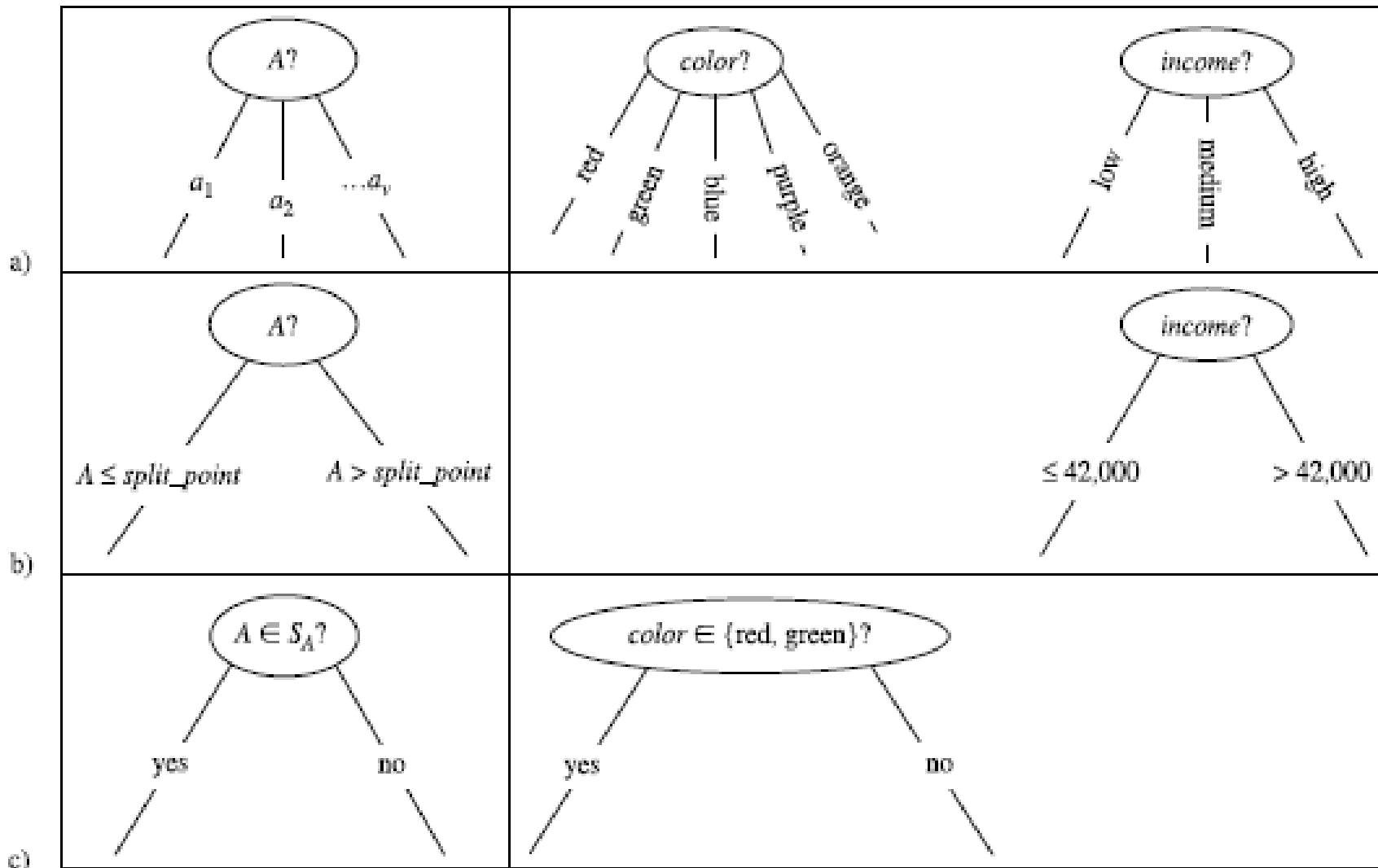Output: A Decision Tree for *"buys_computer"*
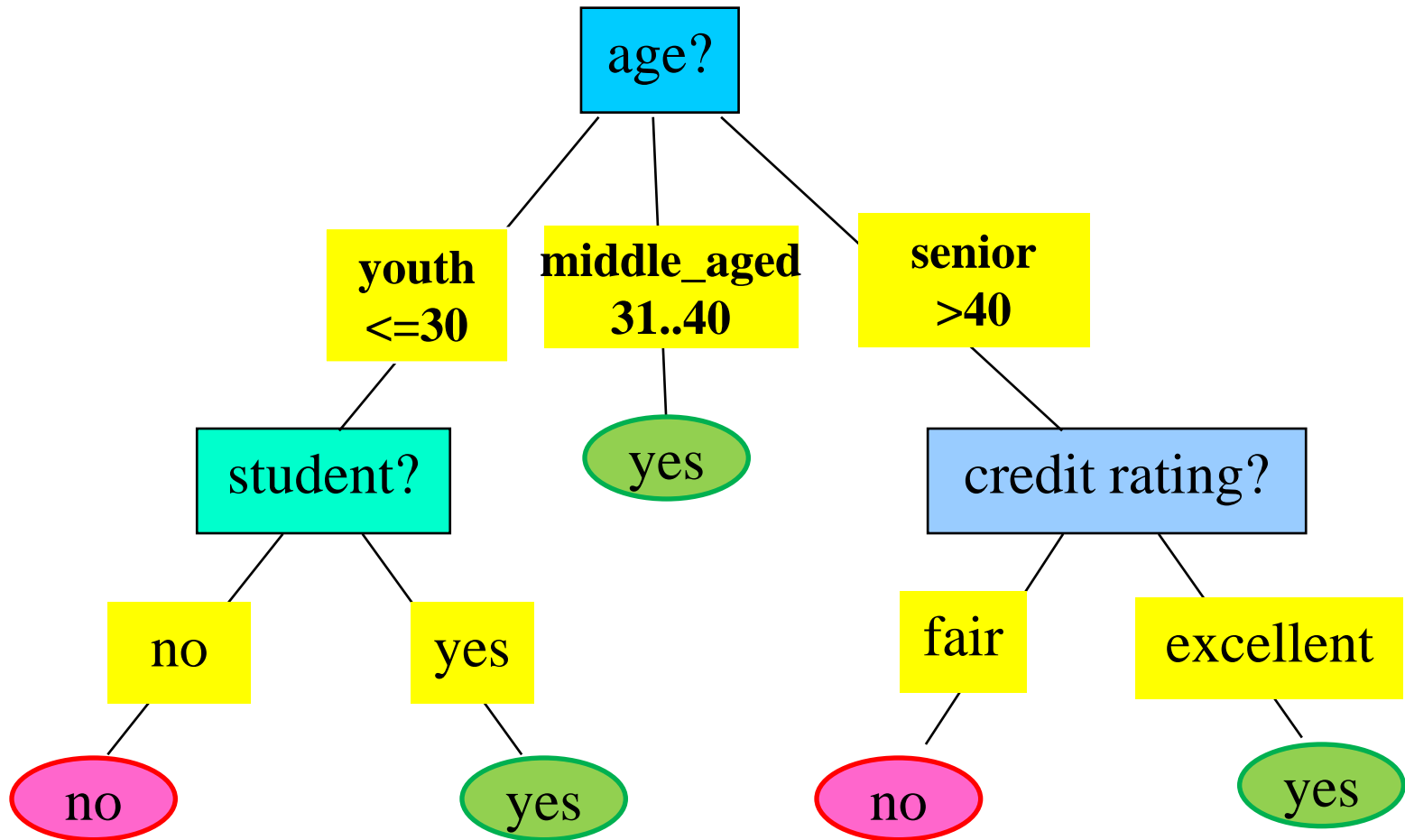
buys_computer="yes" or buys_computer="no"

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

- Information Gain

- Gain Ratio

- Gini Index

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:

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

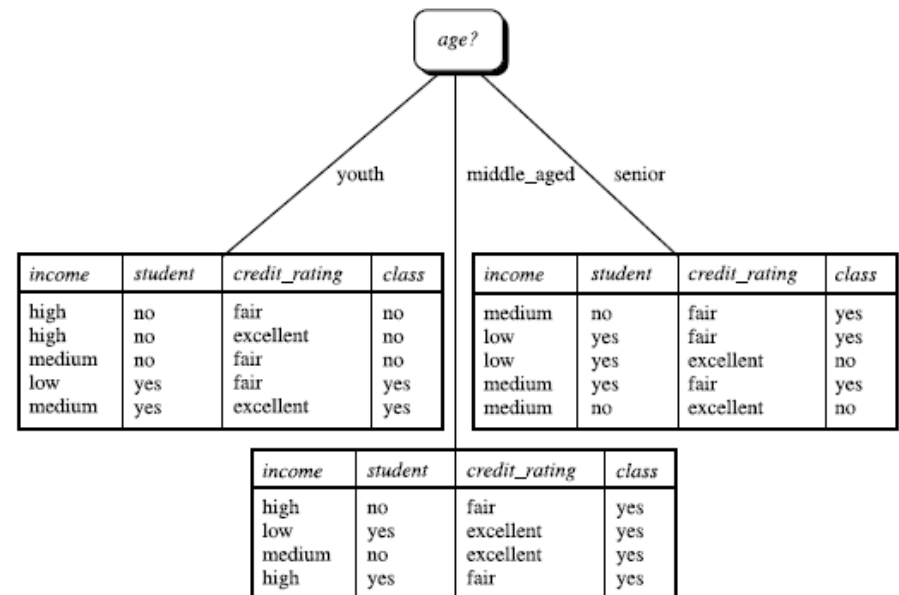- Information needed (after using A to split D into v partitions) to classify D:

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

- 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

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

middle_aged:

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

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

Data Mining: Concepts and Techniques

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

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

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

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

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

cepts and Techniques

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

Gini index (CART, IBM IntelligentMiner)

- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

  where p_j is the relative frequency of class j in D

- If a data set D is split on A into two subsets D_1 and D_2, the *gini* index $gini(D)$ is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Gini index (CART, IBM IntelligentMiner)

- Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right)Gini(D_1) + \left(\frac{4}{14}\right)Gini(D_1)$$

$$= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right)$$

$$= 0.450$$

$$= Gini_{income \in \{high\}}(D)$$

   but gini$_{\{medium,high\}}$ is 0.30 and thus the best since it is the lowest

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - Information gain:
    - biased towards multivalued attributes
  - Gain ratio:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers

- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed

- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

SVM—Support Vector Machines

- A new classification method for both linear and nonlinear data

- It uses a nonlinear mapping to transform the original training data into a higher dimension

- With the new dimension, it searches for the linear optimal separating hyperplane (i.e., "decision boundary")

- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane

- SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors)

SVM—History and Applications

- Vapnik and colleagues (1992)—groundwork from Vapnik & Chervonenkis' statistical learning theory in 1960s

- Features: training can be slow but accuracy is high owing to their ability to model complex nonlinear decision boundaries (margin maximization)

- Used both for classification and prediction

- Applications:

  - handwritten digit recognition, object recognition, speaker identification, benchmarking time-series prediction tests, document classification

SVM—General Philosophy

Small Margin

Large Margin

Support Vectors
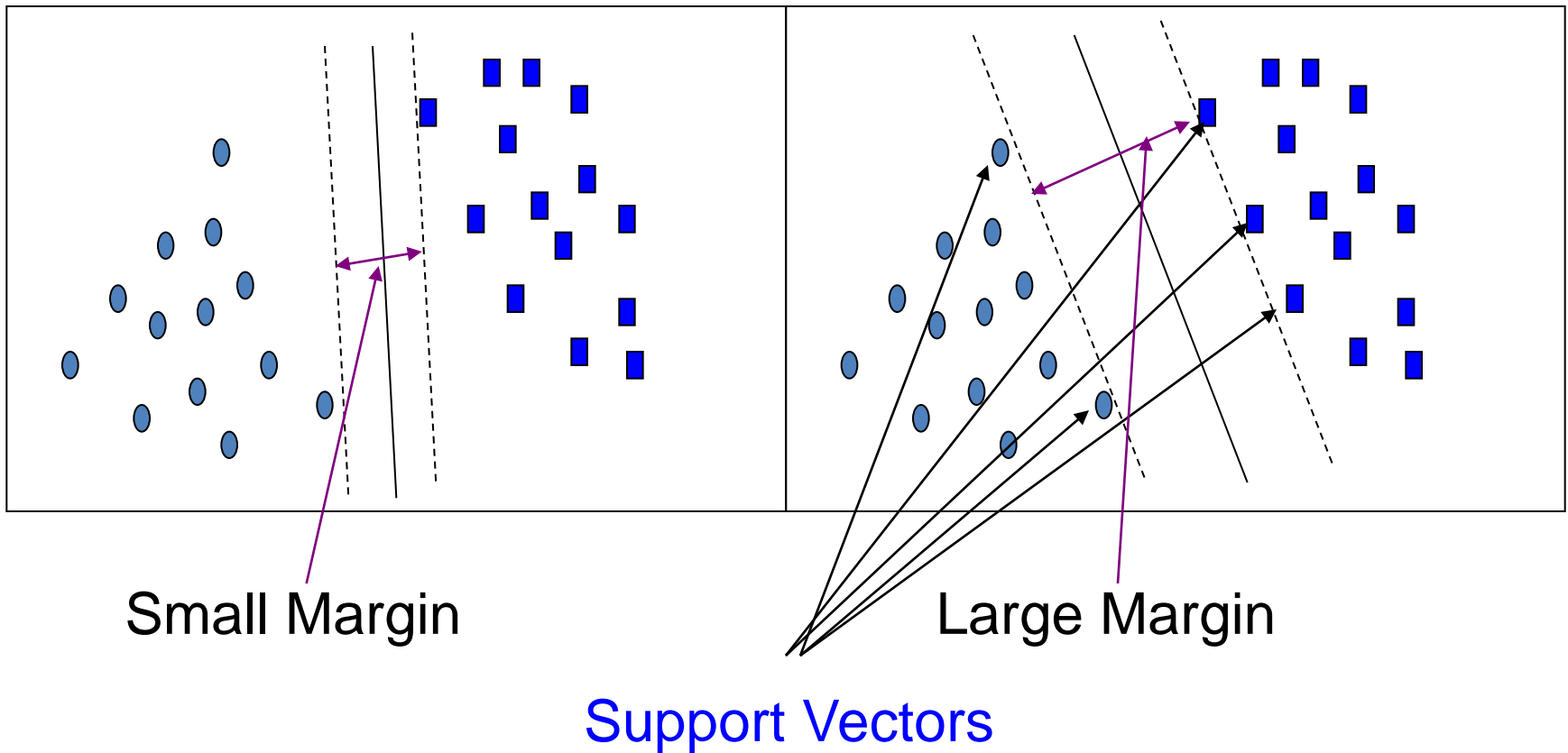
Classification (SVM)
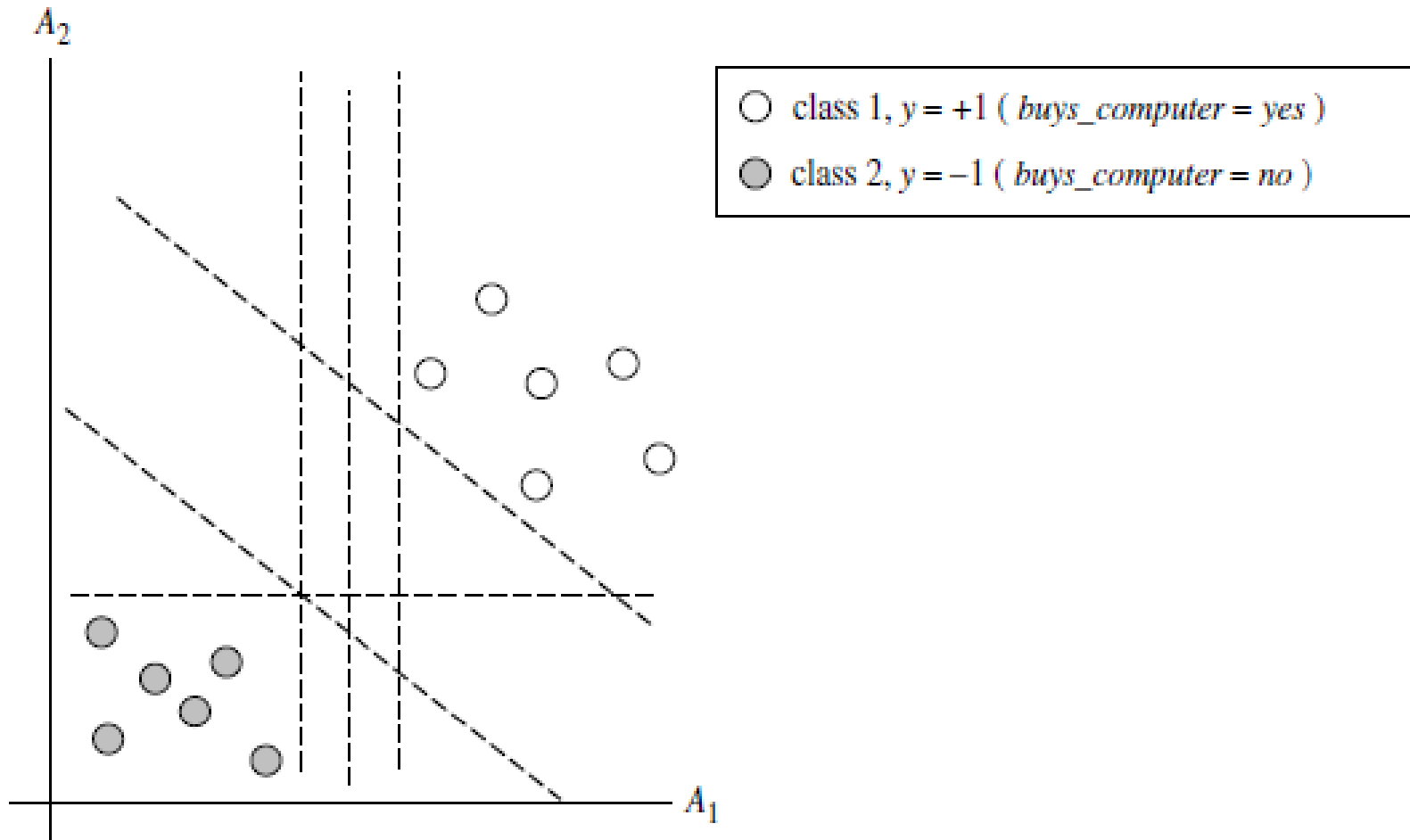
The 2-D training data are linearly separable. There are an infinite number of (possible) separating hyperplanes or "decision boundaries."Which one is best?

Classification (SVM)

class 1, $y = +1$ (*buys_computer = yes*)
class 2, $y = -1$ (*buys_computer = no*)

small margin

class 1, $y = +1$ (*buys_computer = yes*)
class 2, $y = -1$ (*buys_computer = no*)
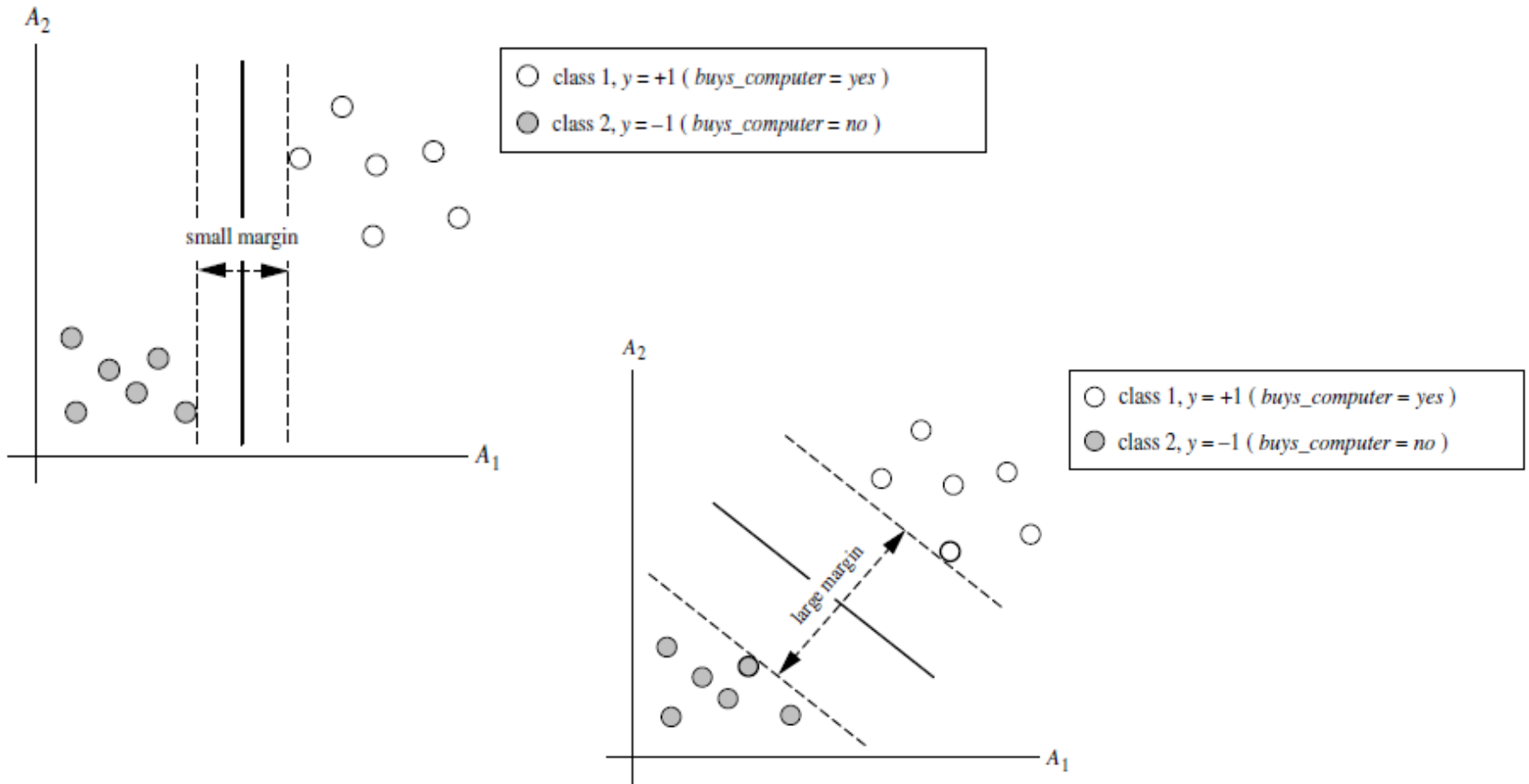
large margin

Which one is better? The one with the larger margin should have greater generalization accuracy.

SVM—When Data Is Linearly Separable
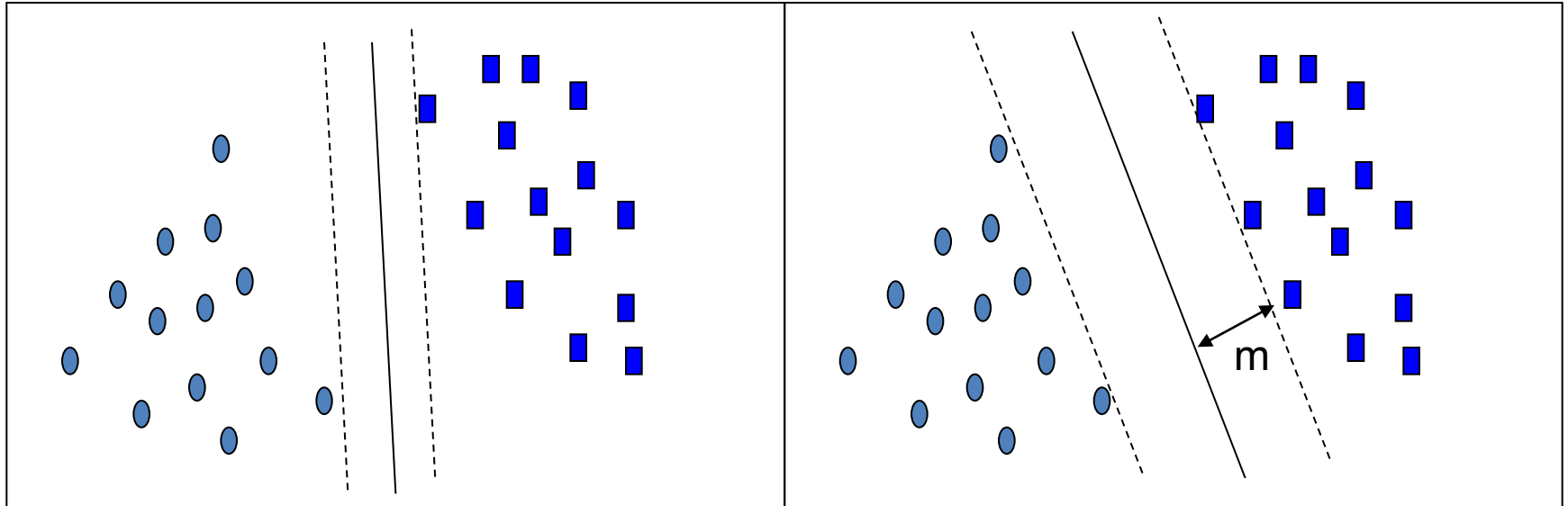
Let data D be ($\mathbf{X}_1$, y_1), …, ($\mathbf{X}_{|D|}$, $y_{|D|}$), where $\mathbf{X}_i$ is the set of training tuples associated with the class labels y_i

There are infinite lines (hyperplanes) separating the two classes but we want to find the best one (the one that minimizes classification error on unseen data)

SVM searches for the hyperplane with the largest margin, i.e., **maximum marginal hyperplane** (MMH)

SVM—Linearly Separable

- A separating hyperplane can be written as

    W ● **X** + b = 0

    where **W**=$\{w_1, w_2, \ldots, w_n\}$ is a weight vector and b a scalar (bias)

- For 2-D it can be written as

    $w_0 + w_1 x_1 + w_2 x_2 = 0$

- The hyperplane defining the sides of the margin:

    H_1: $w_0 + w_1 x_1 + w_2 x_2 \geq 1$ for $y_i = +1$, and

    H_2: $w_0 + w_1 x_1 + w_2 x_2 \leq -1$ for $y_i = -1$

- Any training tuples that fall on hyperplanes H_1 or H_2 (i.e., the sides defining the margin) are **support vectors**

- This becomes a **constrained (convex) quadratic optimization** problem: Quadratic objective function and linear constraints → *Quadratic Programming (QP)* → Lagrangian multipliers

Why Is SVM Effective on High Dimensional Data?

- The complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data

- The support vectors are the essential or critical training examples — they lie closest to the decision boundary (MMH)

- If all other training examples are removed and the training is repeated, the same separating hyperplane would be found

- The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVM classifier, which is independent of the data dimensionality

- Thus, an SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high

SVM—Linearly Inseparable
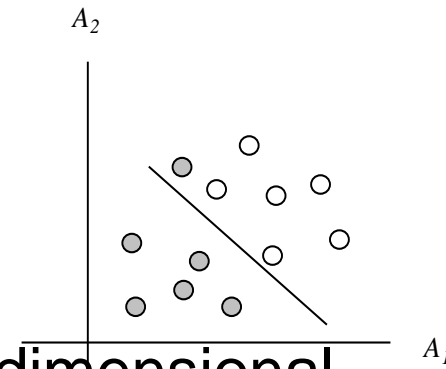
- Transform the original input data into a higher dimensional space

Example 6.8 Nonlinear transformation of original input data into a higher dimensional space. Consider the following example. A 3D input vector $\mathbf{X} = (x_1, x_2, x_3)$ is mapped into a 6D space Z using the mappings $\phi_1(X) = x_1$, $\phi_2(X) = x_2$, $\phi_3(X) = x_3$, $\phi_4(X) = (x_1)^2$, $\phi_5(X) = x_1 x_2$, and $\phi_6(X) = x_1 x_3$. A decision hyperplane in the new space is $d(\mathbf{Z}) = \mathbf{WZ} + b$, where $\mathbf{W}$ and $\mathbf{Z}$ are vectors. This is linear. We solve for $\mathbf{W}$ and b and then substitute back so that we see that the linear decision hyperplane in the new ($\mathbf{Z}$) space corresponds to a nonlinear second order polynomial in the original 3-D input space,

$$d(Z) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 (x_1)^2 + w_5 x_1 x_2 + w_6 x_1 x_3 + b$$
$$= w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + b$$

- Search for a linear separating hyperplane in the new space

Mapping Input Space to Feature Space

Input space   Feature space

Source: http://www.statsoft.com/textbook/support-vector-machines/
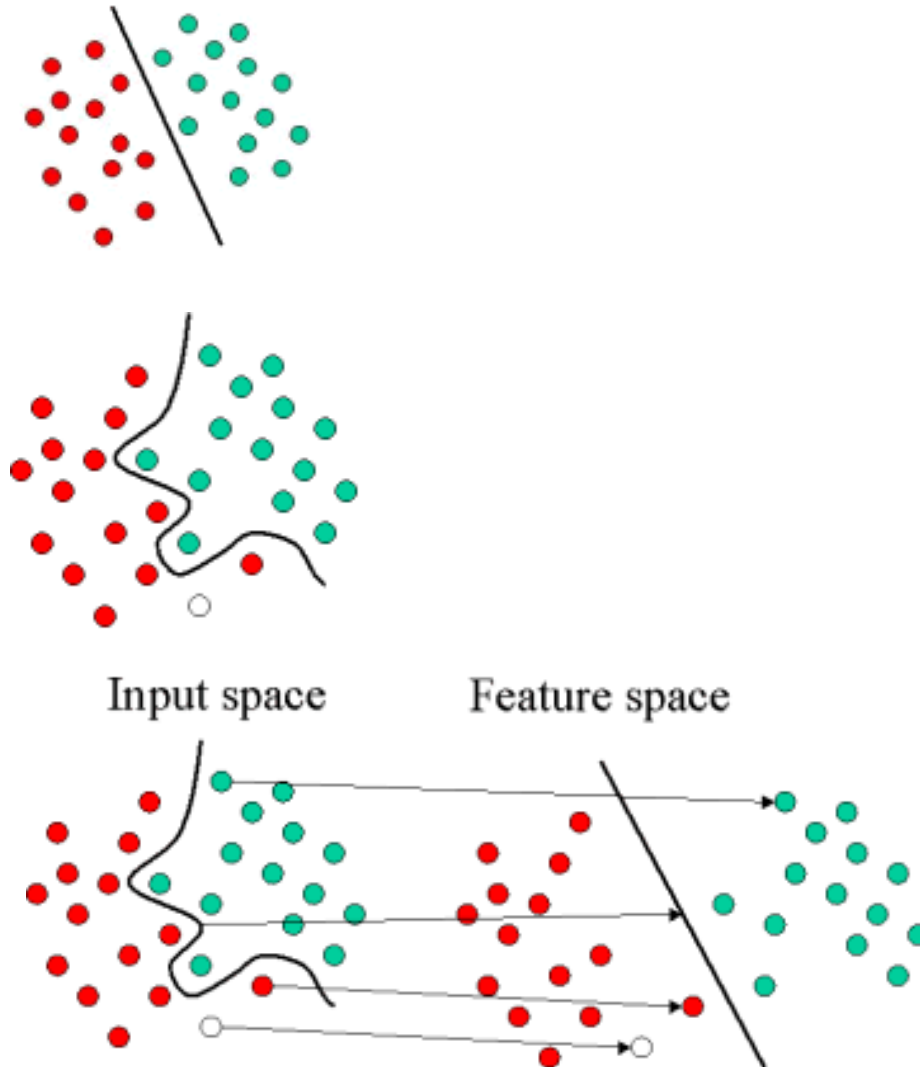
SVM—Kernel functions

- Instead of computing the dot product on the transformed data tuples, it is mathematically equivalent to instead applying a kernel function K($\mathbf{X_i}$, $\mathbf{X_j}$) to the original data, i.e., K($\mathbf{X_i}$, $\mathbf{X_j}$) = Φ($\mathbf{X_i}$) Φ($\mathbf{X_j}$)
- Typical Kernel Functions

$$\text{Polynomial kernel of degree } h: \quad K(X_i, X_j) = (X_i \cdot X_j + 1)^h$$

$$\text{Gaussian radial basis function kernel}: \quad K(X_i, X_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$$

$$\text{Sigmoid kernel}: \quad K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$$

- SVM can also be used for classifying multiple (> 2) classes and for regression analysis (with additional user parameters)

SVM vs. Neural Network

- SVM
  - Relatively new concept
  - Deterministic algorithm
  - Nice Generalization properties
  - Hard to learn – learned in batch mode using quadratic programming techniques
  - Using kernels can learn very complex functions

- Neural Network
  - Relatively old
  - Nondeterministic algorithm
  - Generalizes well but doesn't have strong mathematical foundation
  - Can easily be learned in incremental fashion
  - To learn complex functions—use multilayer perceptron (not that trivial)

SVM Related Links

- SVM Website
  - http://www.kernel-machines.org/
- Representative implementations
  - LIBSVM
    - an efficient implementation of SVM, multi-class classifications, nu-SVM, one-class SVM, including also various interfaces with java, python, etc.
  - SVM-light
    - simpler but performance is not better than LIBSVM, support only binary classification and only C language
  - SVM-torch
    - another recent implementation also written in C.

Classifier Accuracy Measures

	C$_1$	C$_2$
C$_1$	True positive	False negative
C$_2$	False positive	True negative

classes	buy_computer = yes	buy_computer = no	total	recognition(%)
buy_computer = yes	6954	46	7000	99.34
buy_computer = no	412	2588	3000	86.27
total	7366	2634	10000	95.52

- Accuracy of a classifier M, acc(M): percentage of test set tuples that are correctly classified by the model M
  - Error rate (misclassification rate) of M = 1 – acc(M)
  - Given m classes, $CM_{i,j}$, an entry in a **confusion matrix**, indicates # of tuples in class i that are labeled by the classifier as class j
- Alternative accuracy measures (e.g., for cancer diagnosis)

  sensitivity = t-pos/pos   /* true positive recognition rate */

  specificity = t-neg/neg   /* true negative recognition rate */

  precision = t-pos/(t-pos + f-pos)

  accuracy = sensitivity * pos/(pos + neg) + specificity * neg/(pos + neg)

  - This model can also be used for cost-benefit analysis

Predictor Error Measures

- Measure predictor accuracy: measure how far off the predicted value is from the actual known value

- **Loss function**: measures the error betw. y_i and the predicted value y_i'
  - Absolute error: $| y_i - y_i'|$
  - Squared error: $(y_i - y_i')^2$

- Test error (generalization error): the average loss over the test set
  - Mean absolute error: $$\dfrac{\sum_{i=1}^{d} | y_i - y_i'|}{d}$$ Mean squared error: $$\dfrac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$$

  - Relative absolute error: $$\dfrac{\sum_{i=1}^{d} | y_i - y_i'|}{\sum_{i=1}^{d} | y_i - \overline{y} |}$$ Relative squared error: $$\dfrac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \overline{y})^2}$$

  The mean squared-error exaggerates the presence of outliers

  Popularly use (square) root mean-square error, similarly, root relative squared error

Evaluating the Accuracy of a Classifier or Predictor (I)

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: k folds where k = # of tuples, for small sized data
  - Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

Evaluating the Accuracy of a Classifier or Predictor (II)

- Bootstrap
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several boostrap methods, and a common one is **.632 boostrap**
  - Suppose we are given a data set of d tuples. The data set is sampled d times, with replacement, resulting in a training set of d samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data will end up in the bootstrap, and the remaining 36.8% will form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
  - Repeat the sampling procedue k times, overall accuracy of the model:

$$acc(M) = \sum_{i=1}^{k}(0.632 \times acc(M_i)_{test\_set} + 0.368 \times acc(M_i)_{train\_set})$$

Ensemble Methods: Increasing the Accuracy
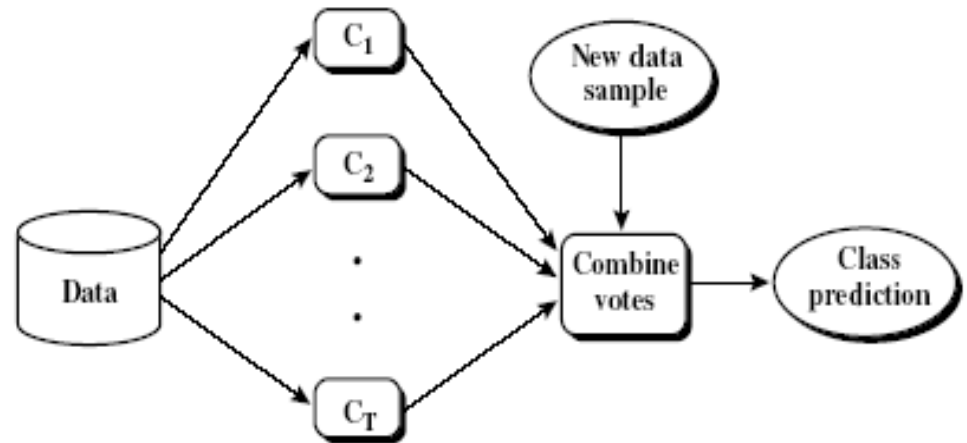
- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, ..., M_k, with the aim of creating an improved model M*
- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
 - Ensemble: combining a set of heterogeneous classifiers

Model Selection: ROC Curves

- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models

- Originated from signal detection theory

- Shows the trade-off between the true positive rate and the false positive rate

- The area under the ROC curve is a measure of the accuracy of the model

- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list

- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model
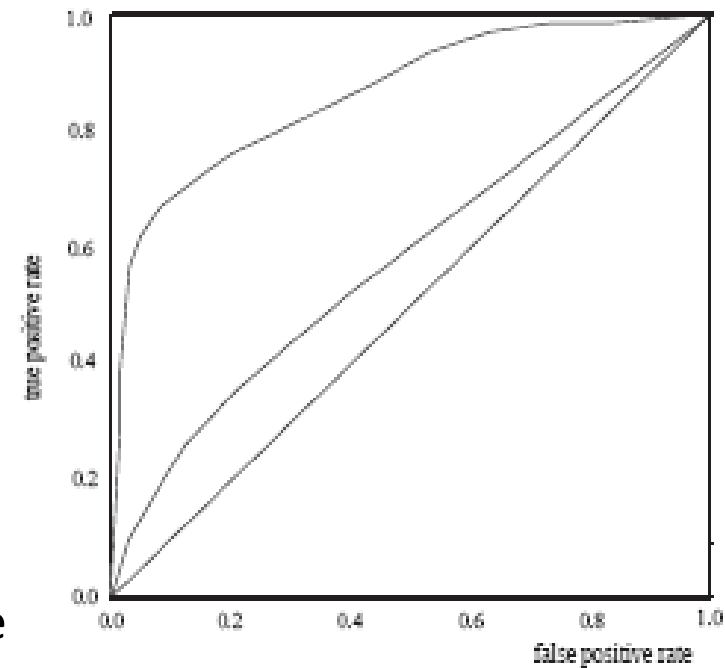
- Vertical axis represents the true positive rate

- Horizontal axis rep. the false positive rate

- The plot also shows a diagonal line

- A model with perfect accuracy will have an area of 1.0

Cluster Analysis
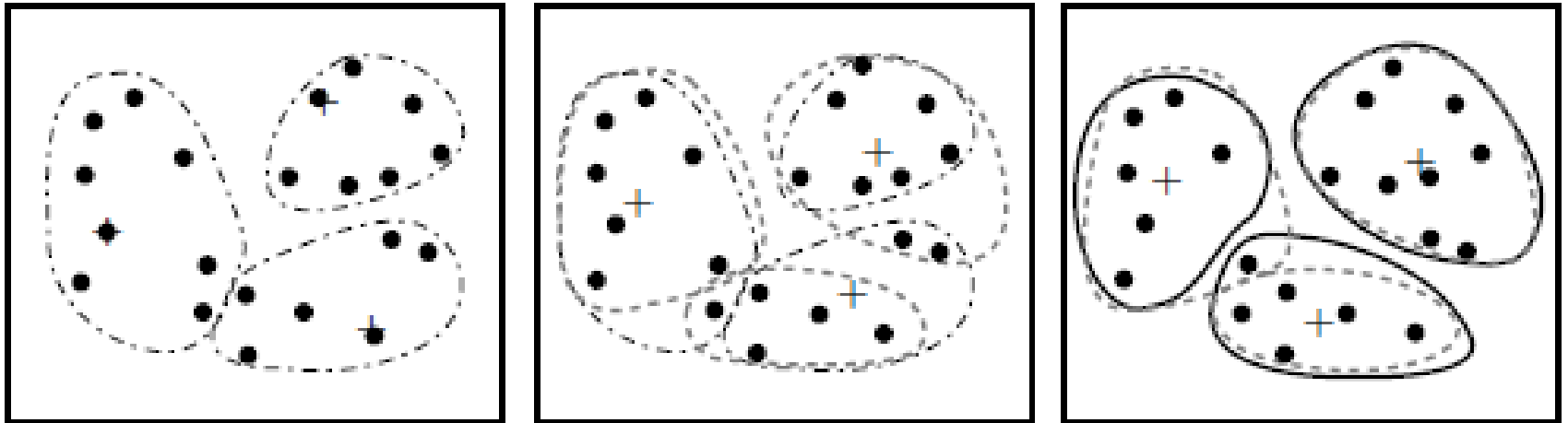
(a)   (b)   (c)

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

Clustering: Rich Applications and Multidisciplinary Efforts

- Pattern Recognition

- Spatial Data Analysis
  - Create thematic maps in GIS by clustering feature spaces
  - Detect spatial clusters or for other spatial mining tasks

- Image Processing

- Economic Science (especially market research)

- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

- Land use: Identification of areas of similar land use in an earth observation database

- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost

- City-planning: Identifying groups of houses according to their house type, value, and geographical location

- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with

    - high intra-class similarity

    - low inter-class similarity

- The quality of a clustering result depends on both the similarity measure used by the method and its implementation

- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$

- There is a separate "quality" function that measures the "goodness" of a cluster.

- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.

- Weights should be associated with different variables based on applications and data semantics.

- It is hard to define "similar enough" or "good enough"

  - the answer is typically highly subjective.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
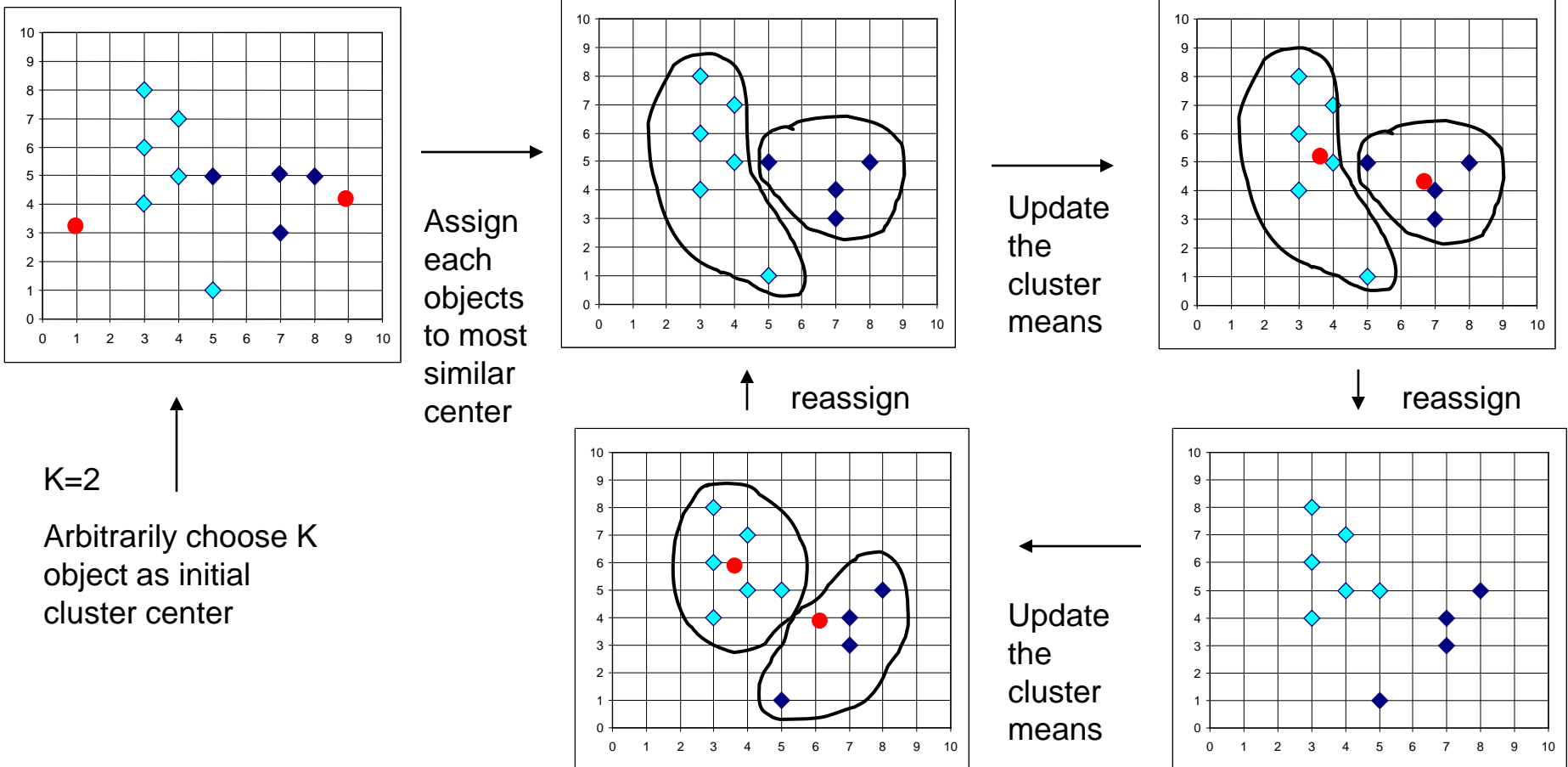- Interpretability and usability

Type of data in clustering analysis

- Interval-scaled variables

- Binary variables

- Nominal, ordinal, and ratio variables

- Variables of mixed types

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment

The *K-Means* Clustering Method
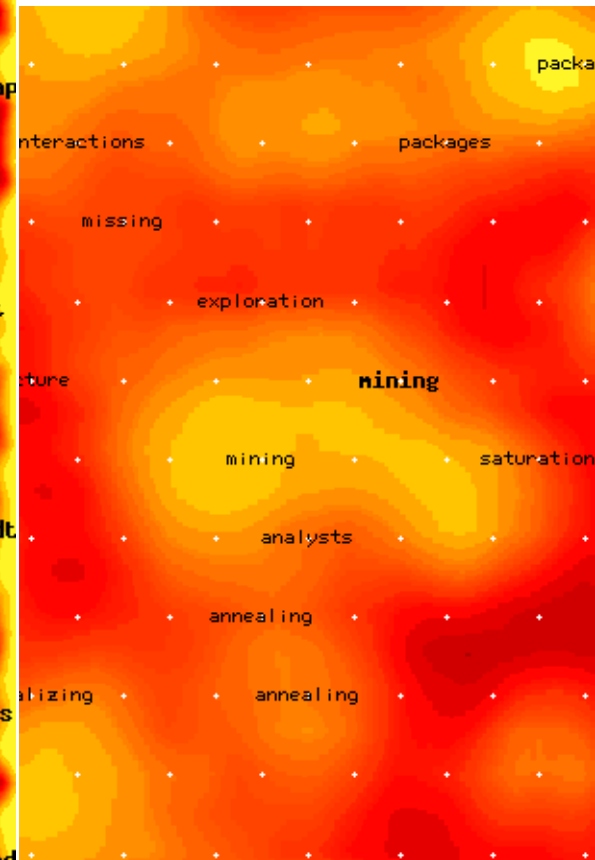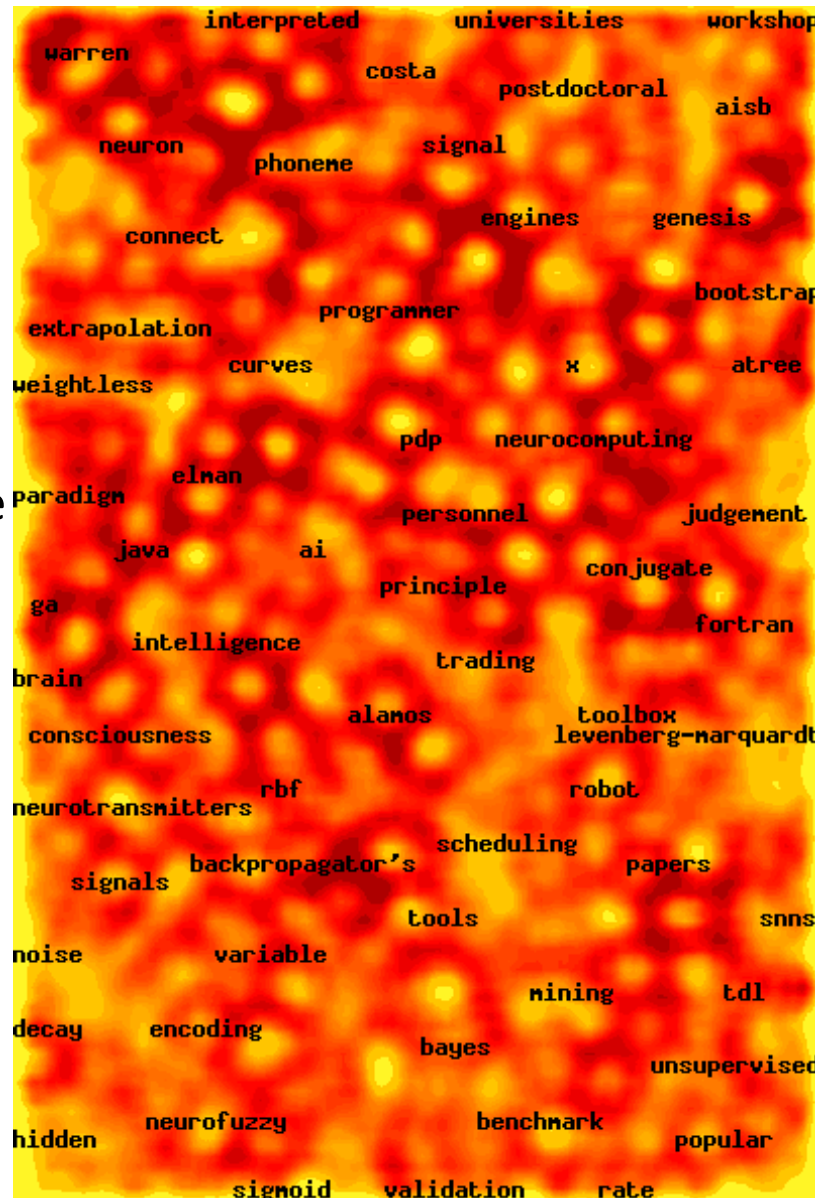
- ## Example

K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

reassign

Update the cluster means

reassign

Self-Organizing Feature Map (SOM)

- SOMs, also called topological ordered maps, or Kohonen Self-Organizing Feature Map (KSOMs)

- It maps all the points in a high-dimensional source space into a 2 to 3-d target space, s.t., the distance and proximity relationship (i.e., topology) are preserved as much as possible

- Similar to k-means: cluster centers tend to lie in a low-dimensional manifold in the feature space

- Clustering is performed by having several units competing for the current object
  - The unit whose weight vector is closest to the current object wins
  - The winner and its neighbors learn by having their weights adjusted

- SOMs are believed to resemble processing that can occur in the brain

- Useful for visualizing high-dimensional data in 2- or 3-D space

Web Document Clustering Using SOM

- The result of SOM clustering of 12088 Web articles

- The picture on the right: drilling down on the keyword "mining"

- Based on websom.hut.fi Web page

What Is Outlier Discovery?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordon, Wayne Gretzky, …
- Problem: Define and find outliers in large data sets
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

Outlier Discovery: Statistical Approaches
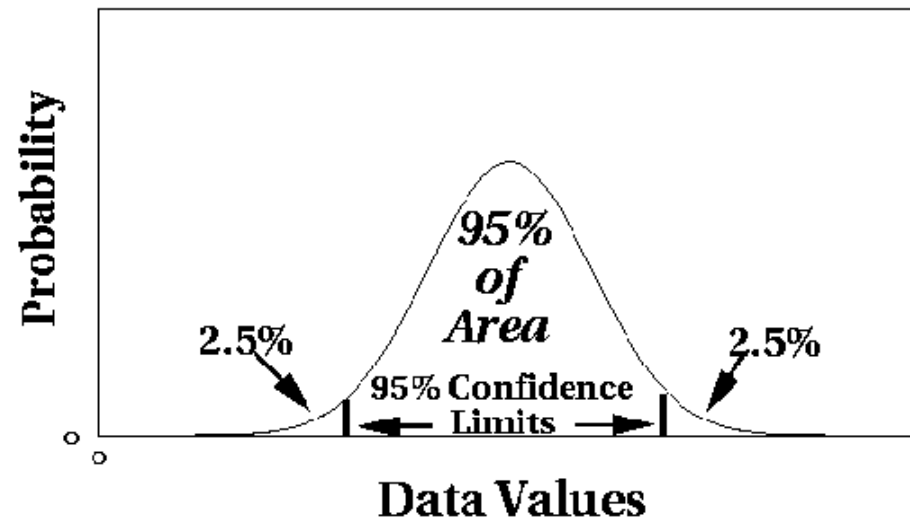
↗ Assume a model underlying distribution that generates data set (e.g. normal distribution)

- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers

- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

Cluster Analysis

- Cluster analysis groups objects based on their similarity and has wide applications

- Measure of similarity can be computed for various types of data

- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods

- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches

- There are still lots of research issues on cluster analysis

Summary

- Classification and Prediction
- Cluster Analysis

References

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