Data Mining
資料探勘

分類與預測
(Classification and Prediction)

1002DM03
MI4
Thu. 9,10 (16:10-18:00) B513

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. tku.edu.tw/myday/
2012-03-01

課程大綱 (Syllabus)

週次	日期	內容（Subject/Topics）
1	101/02/16	資料探勘導論 (Introduction to Data Mining)
2	101/02/23	關連分析 (Association Analysis)
3	101/03/01	分類與預測 (Classification and Prediction)
4	101/03/08	分群分析 (Cluster Analysis)
5	101/03/15	個案分析與實作一 (分群分析)： Banking Segmentation (Cluster Analysis – KMeans)
6	101/03/22	個案分析與實作二 (關連分析)： Web Site Usage Associations (Association Analysis)
7	101/03/29	個案分析與實作三 (決策樹、模型評估)： Enrollment Management Case Study (Decision Tree, Model Evaluation)

課程大綱 (Syllabus)

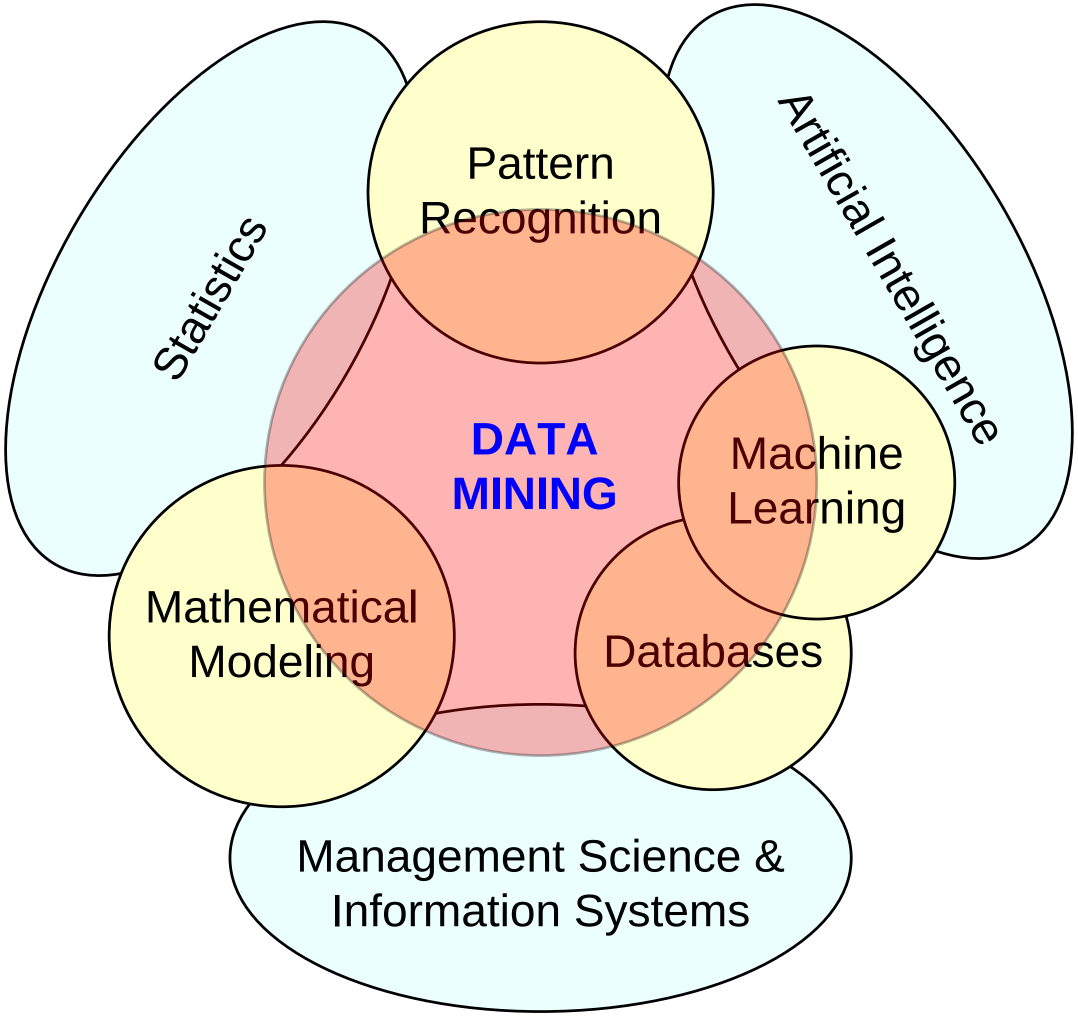
週次	日期	內容（Subject/Topics）
8	101/04/05	教學行政觀摩日 (--No Class--)
9	101/04/12	期中報告 (Midterm Presentation)
10	101/04/19	期中考試週
11	101/04/26	個案分析與實作四 (迴歸分析、類神經網路)： Credit Risk Case Study (Regression Analysis, Artificial Neural Network)
12	101/05/03	文字探勘與網頁探勘 (Text and Web Mining)
13	101/05/10	社會網路分析、意見分析 (Social Network Analysis, Opinion Mining)
14	101/05/17	期末專題報告 (Term Project Presentation)
15	101/05/24	畢業考試週

Outline

- Classification and Prediction
- Decision Tree
- Support Vector Machine (SVM)
- Evaluation (Accuracy of Classification Model)

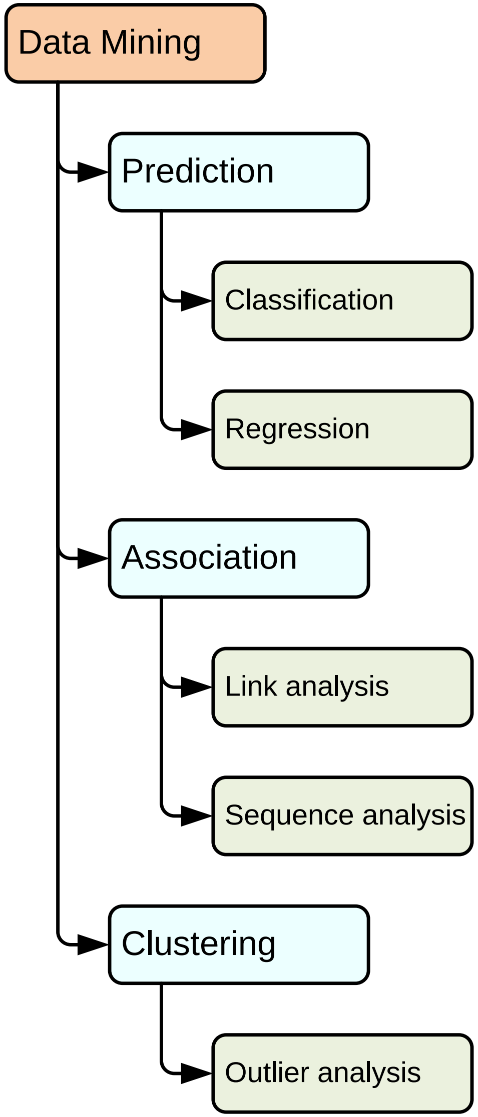
Source: Han & Kamber (2006)

Data Mining at the Intersection of Many Disciplines



Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

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)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

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

Source: Han & Kamber (2006)

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”

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

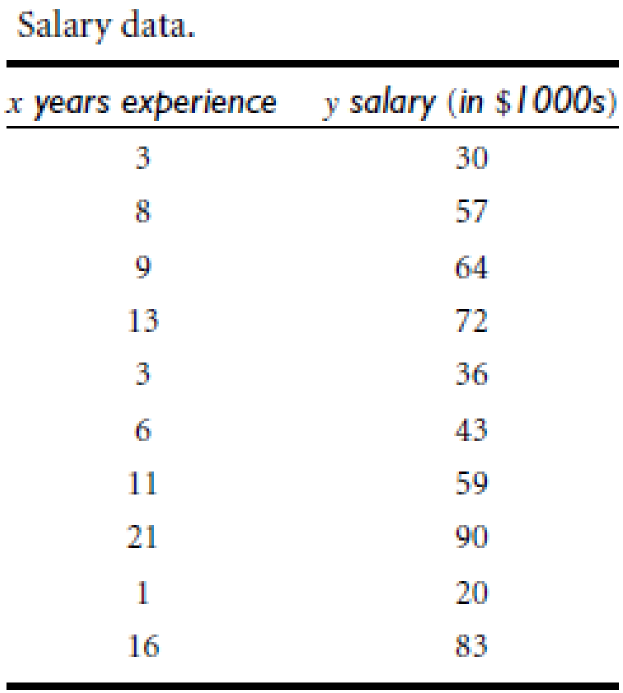
What Is Prediction?

- (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

Source: Han & Kamber (2006)

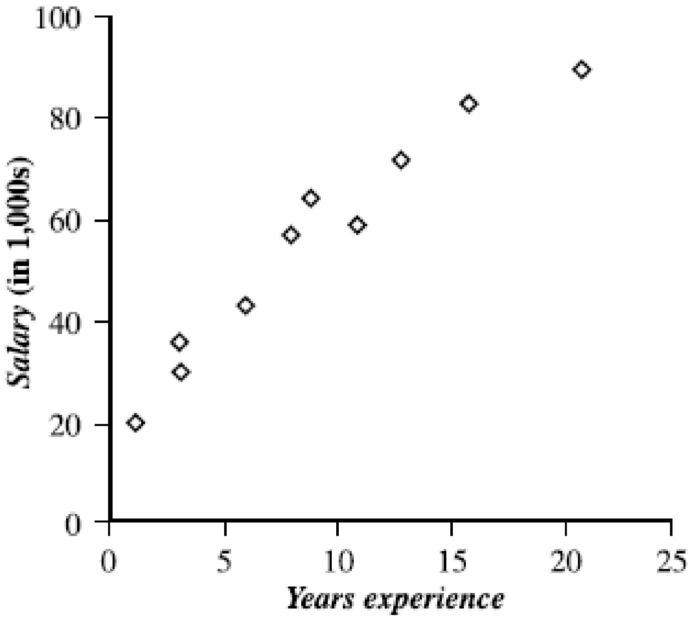
Prediction Methods

- 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



Source: Han & Kamber (2006)

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.

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

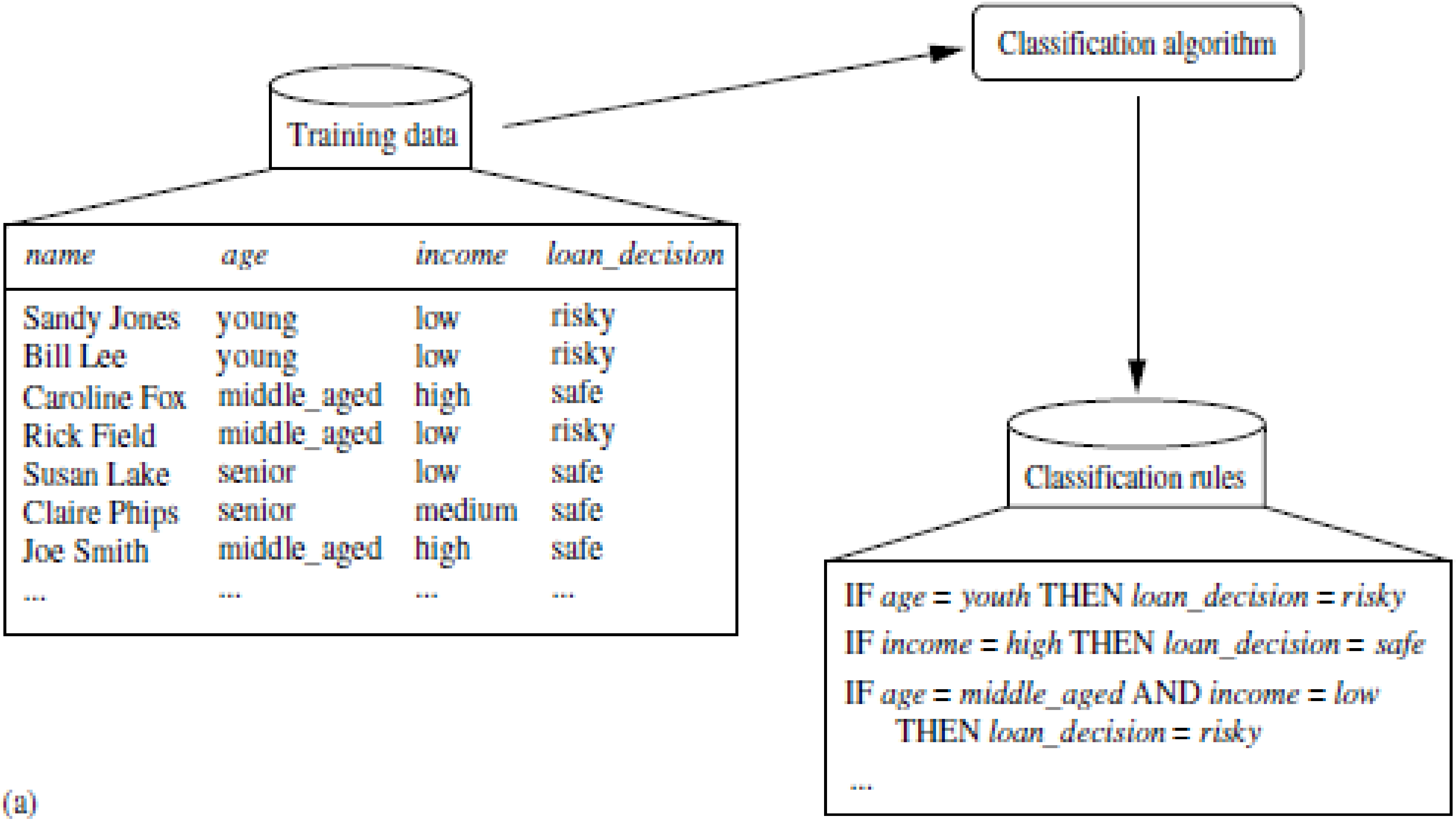
Classification—A Two-Step Process

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

Source: Han & Kamber (2006)

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
...	...	...	...

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

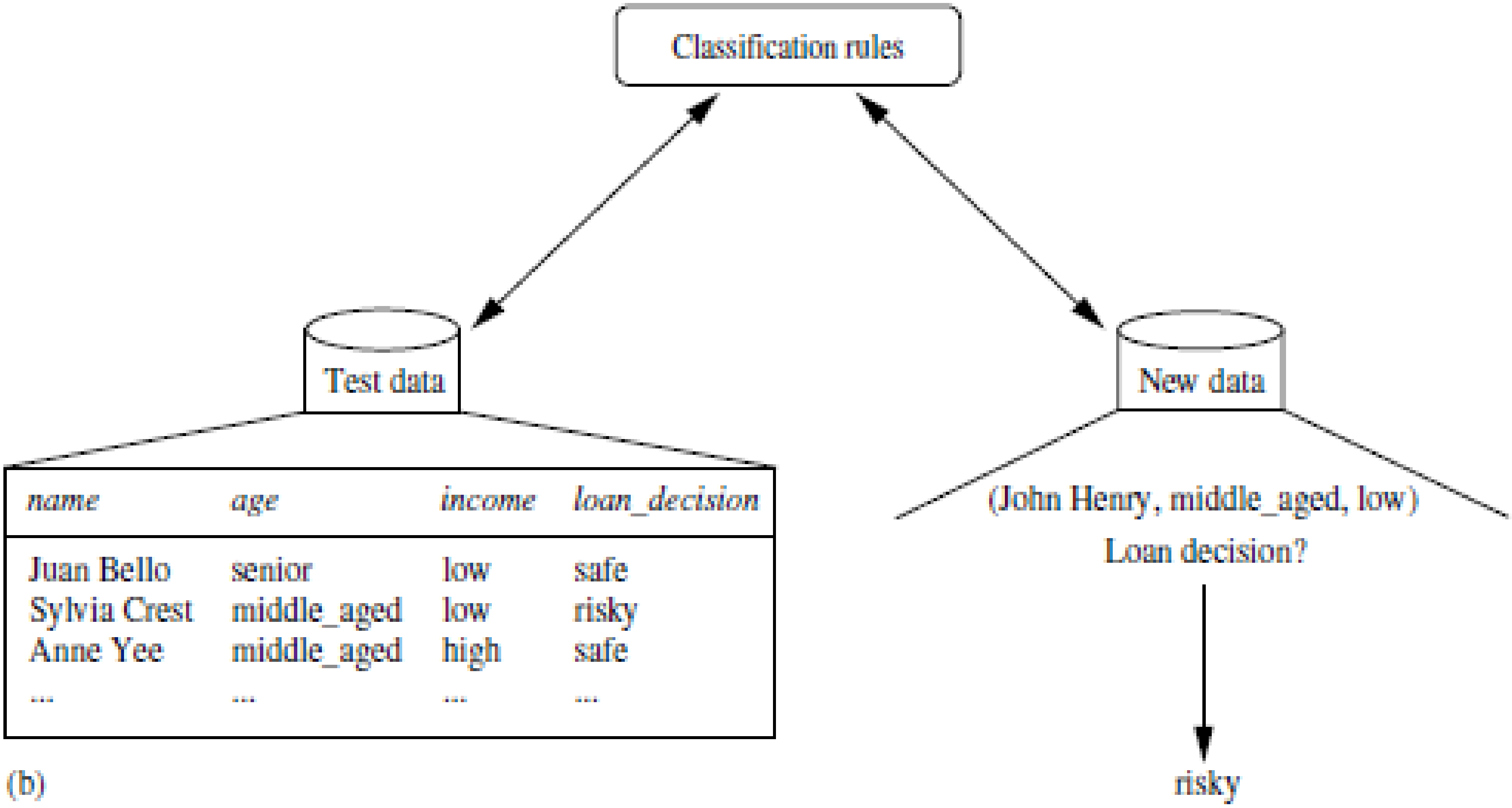
IF *income* = *high* THEN *loan_decision* = *safe*

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

...

Source: Han & Kamber (2006)

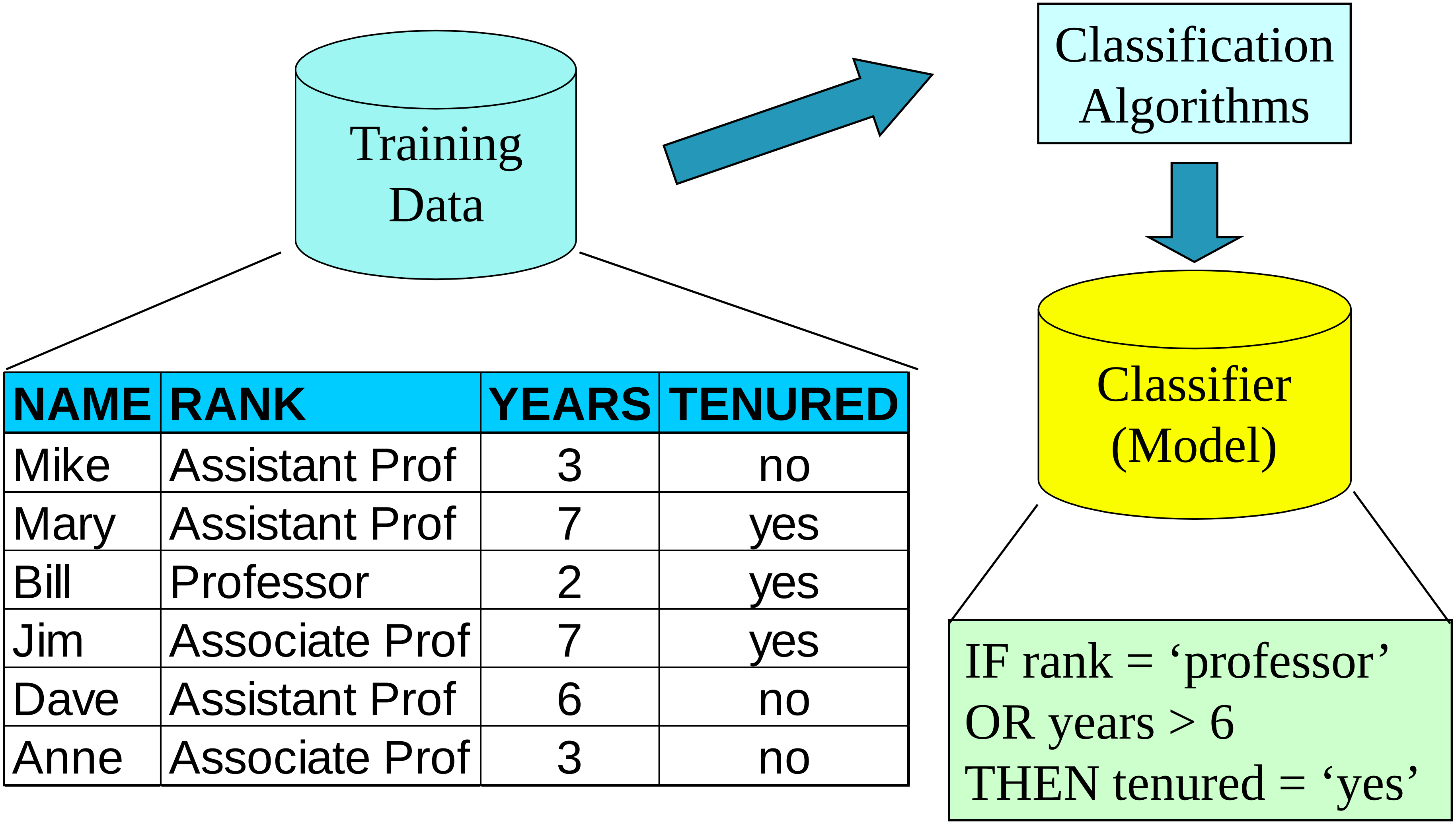
Data Classification Process 2
(b) Classification: Test data are used to estimate the accuracy of the classification rules.



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

Source: Han & Kamber (2006)

Process (1): Model Construction



Training Data

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

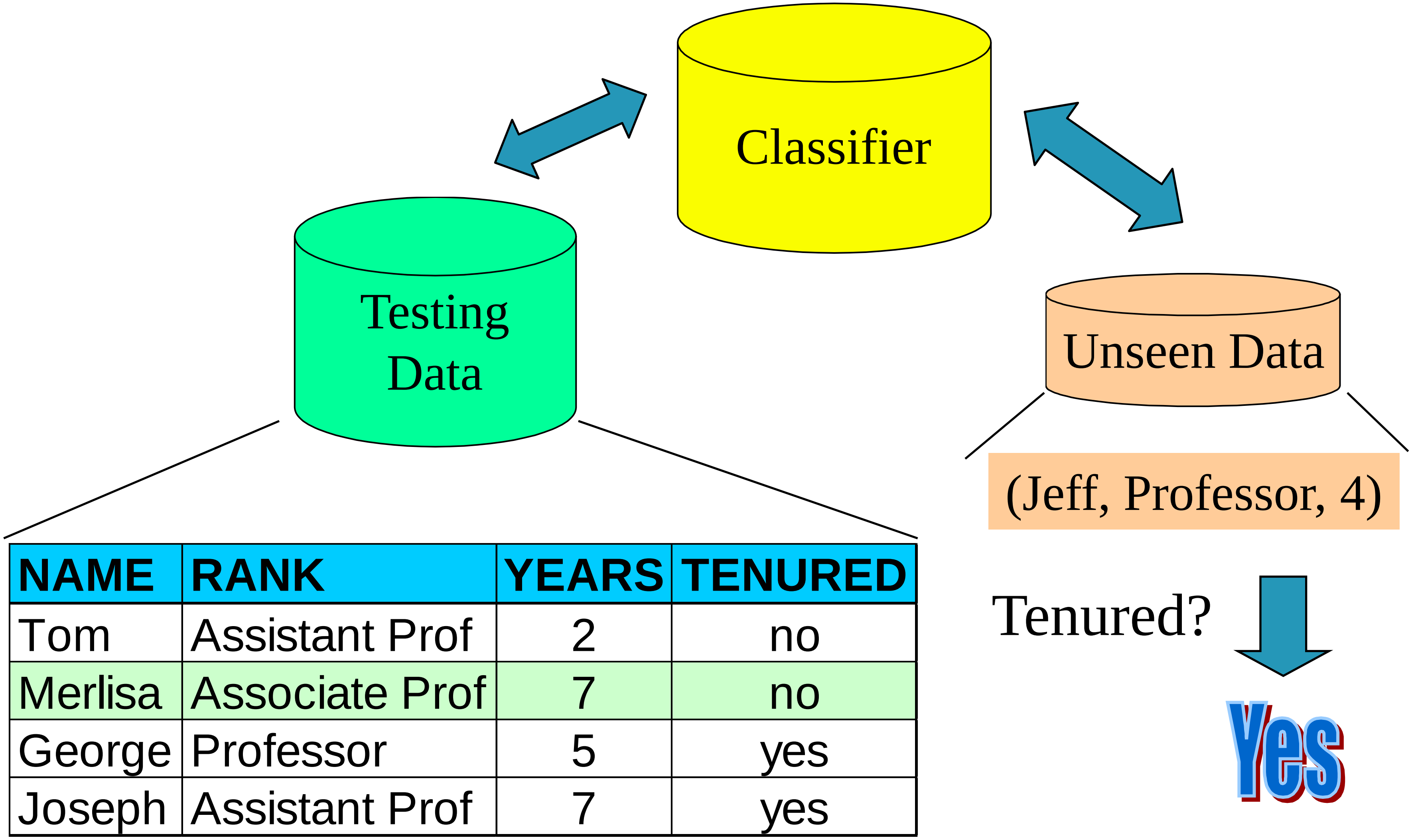
Classification Algorithms

Classifier (Model)

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Source: Han & Kamber (2006)

Process (2): Using the Model in Prediction



Classifier

Testing Data

Unseen Data

(Jeff, Professor, 4)

Tenured?

Yes

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

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

Classification by Decision Tree Induction
Training Dataset

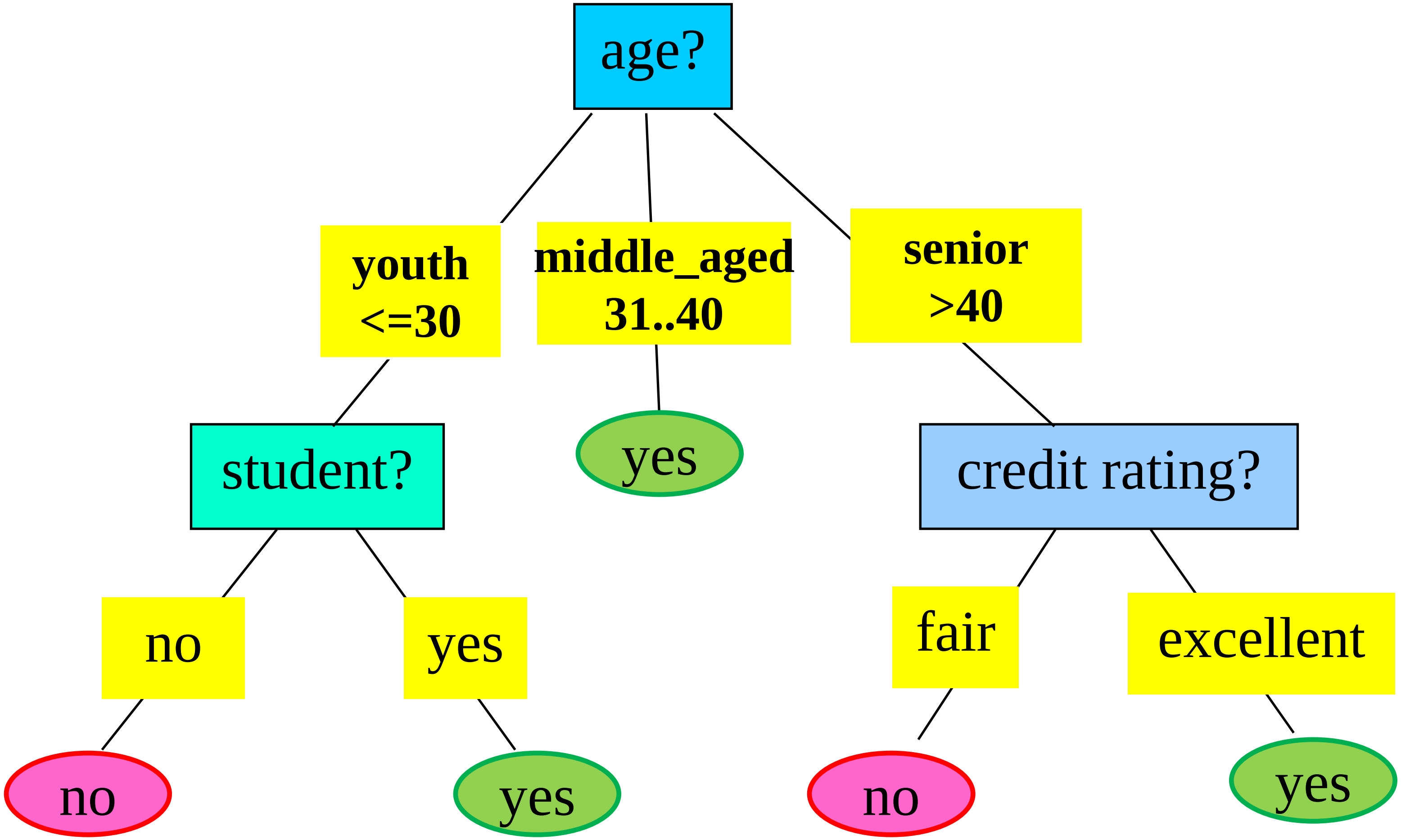
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent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

Source: Han & Kamber (2006)

Classification by Decision Tree Induction

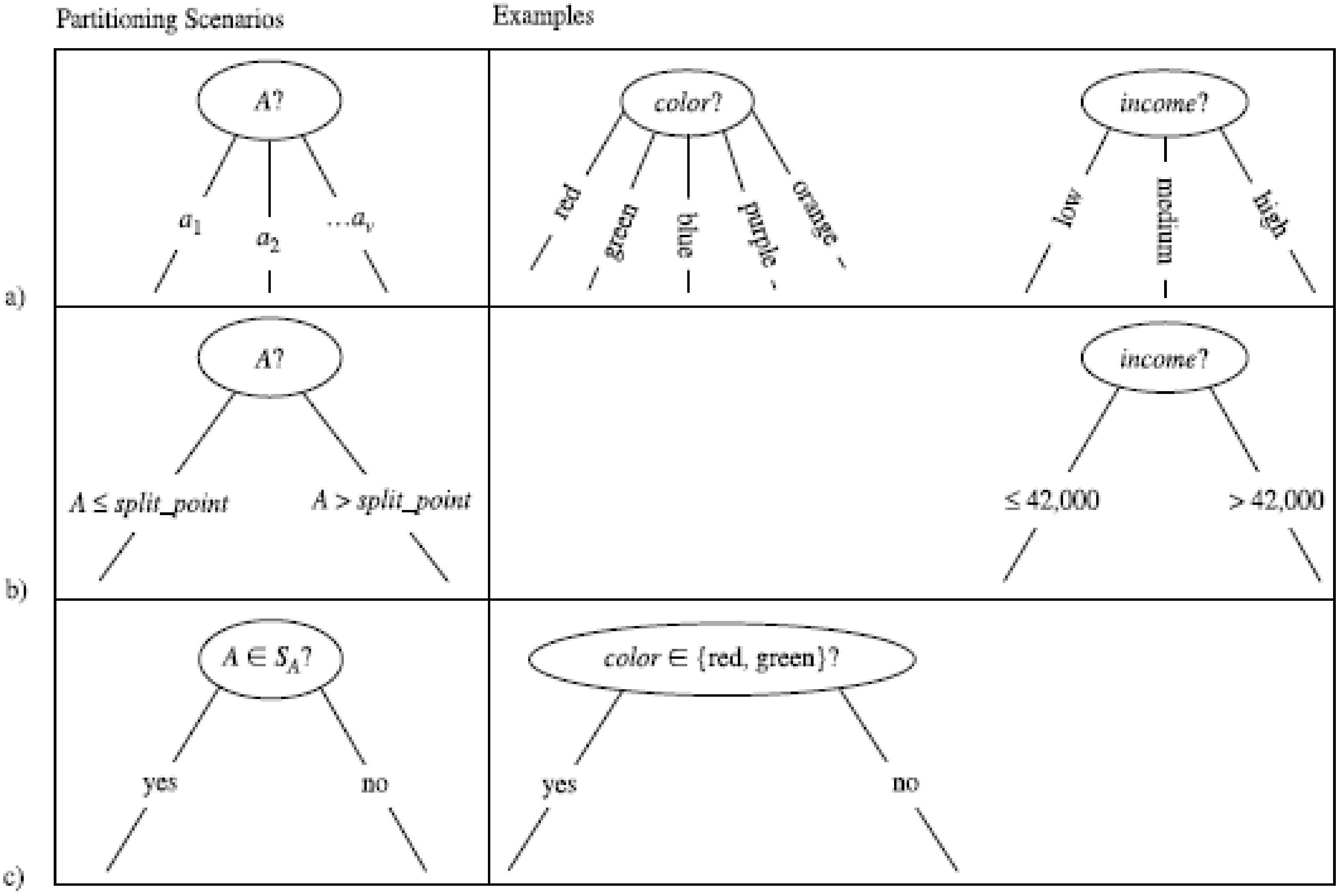
Output: A Decision Tree for “*buys_computer*”



buys_computer=“yes” or buys_computer=“no”

Source: Han & Kamber (2006)

Three possibilities for partitioning tuples based on the splitting Criterion



Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

Attribute Selection Measure

- Information Gain
- Gain Ratio
- Gini Index

Source: Han & Kamber (2006)

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"

Source: Han & Kamber (2006)

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:

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

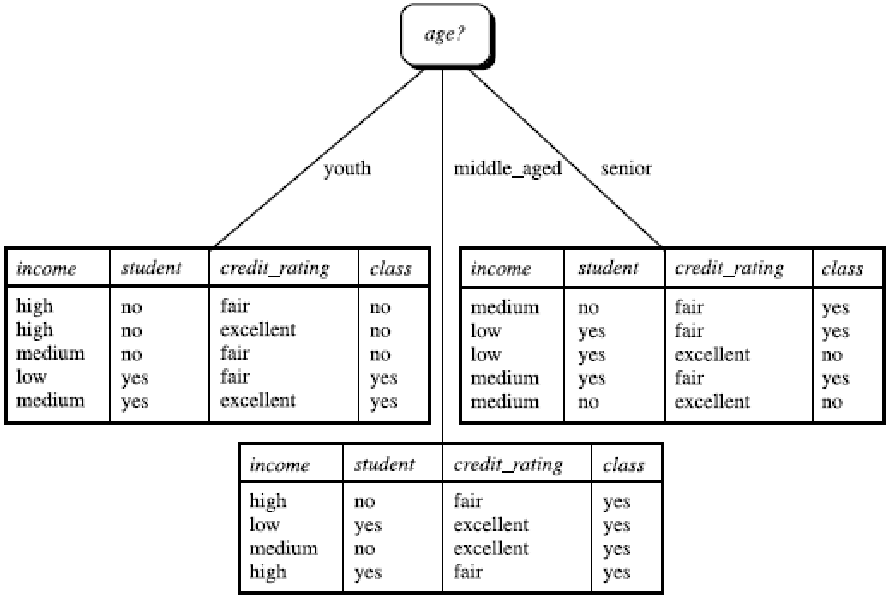
- Information gained by branching on attribute A

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

Source: Han & Kamber (2006)

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



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

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

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

Source: Han & Kamber (2006)

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

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

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

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

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

& Kamber (2006)

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)

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

  - 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

Source: Han & Kamber (2006)

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*)

Source: Han & Kamber (2006)

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

$$\begin{aligned} gini_{income \in \{low, medium\}}(D) &= \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_1) \\ &= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) \\ &= 0.450 \\ &= Gini_{income \in \{high\}}(D) \end{aligned}$$

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

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

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)

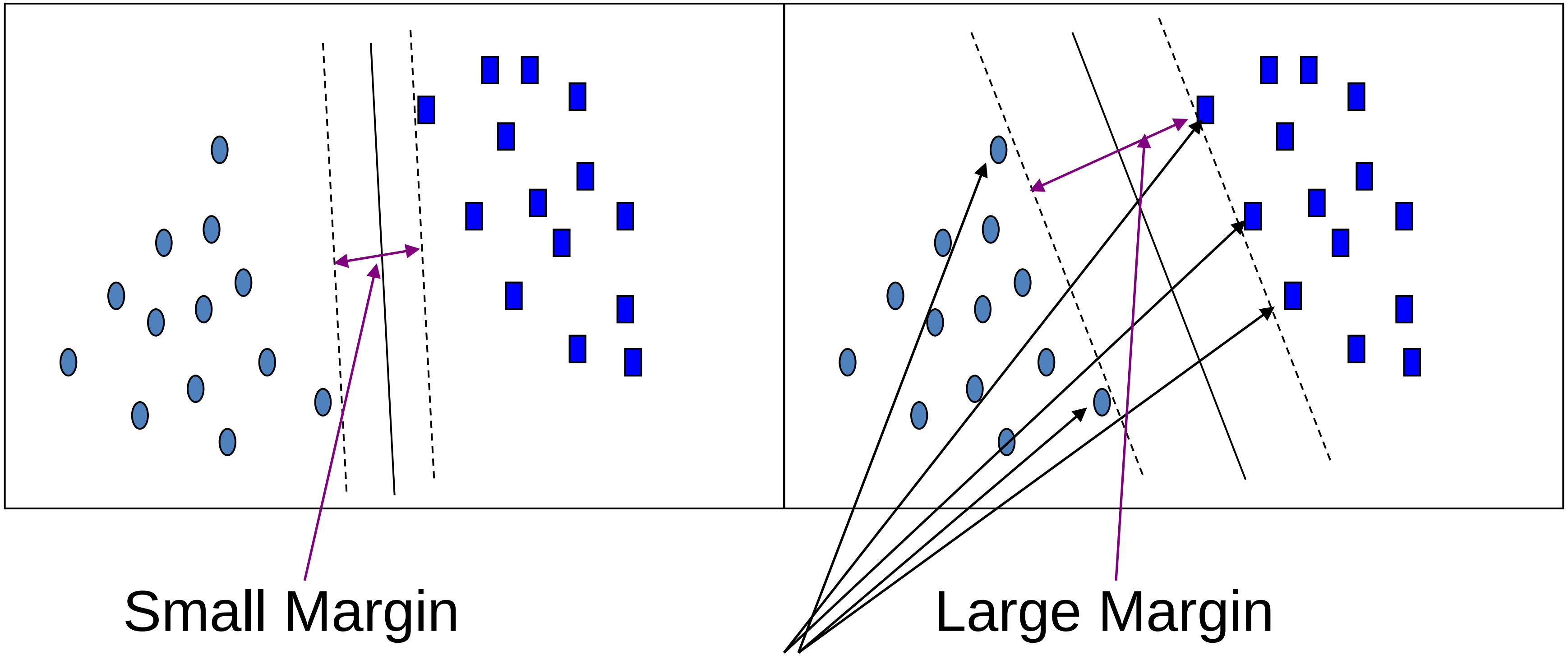
Source: Han & Kamber (2006)

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

Source: Han & Kamber (2006)

SVM—General Philosophy



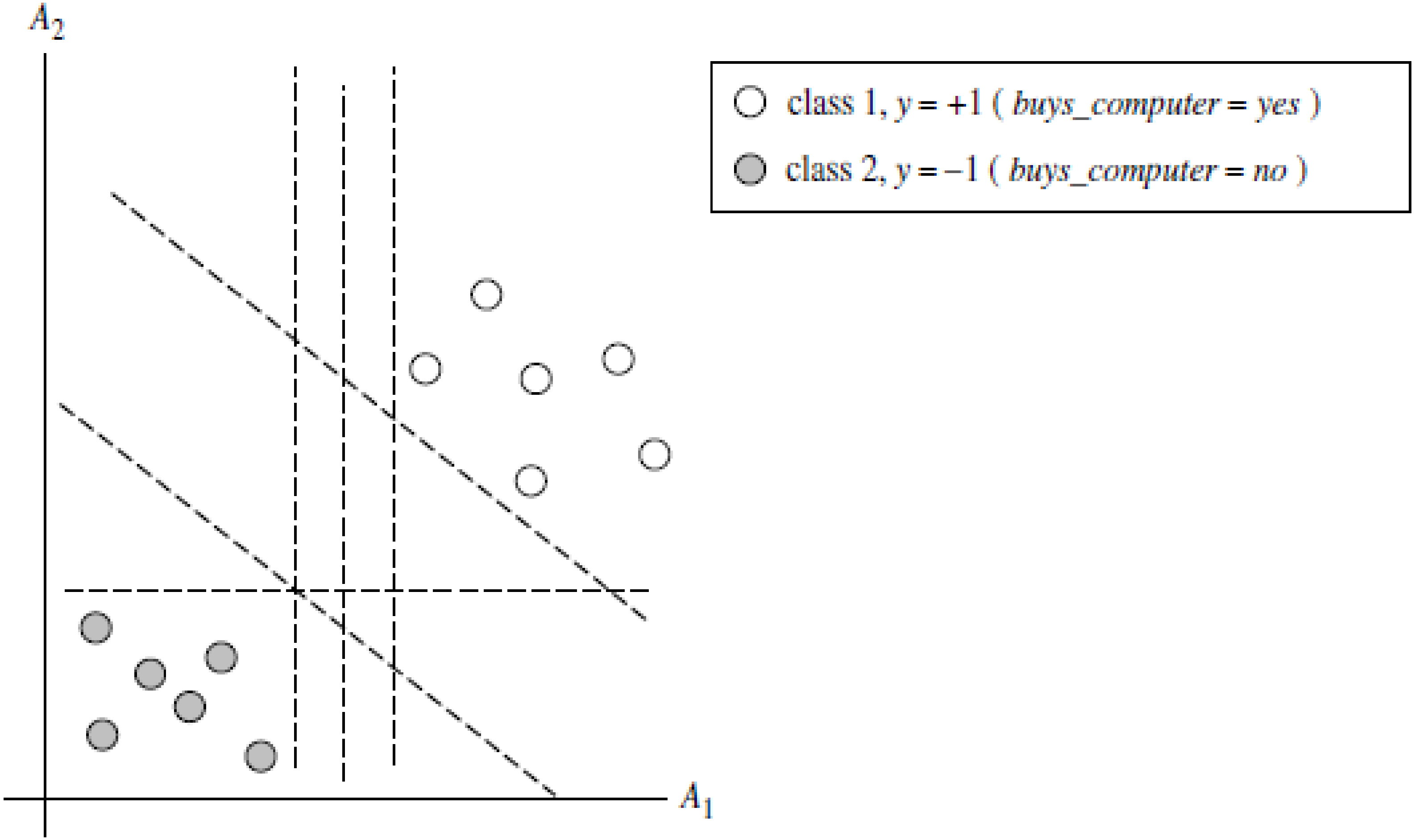
Small Margin

Large Margin

Support Vectors

Source: Han & Kamber (2006)

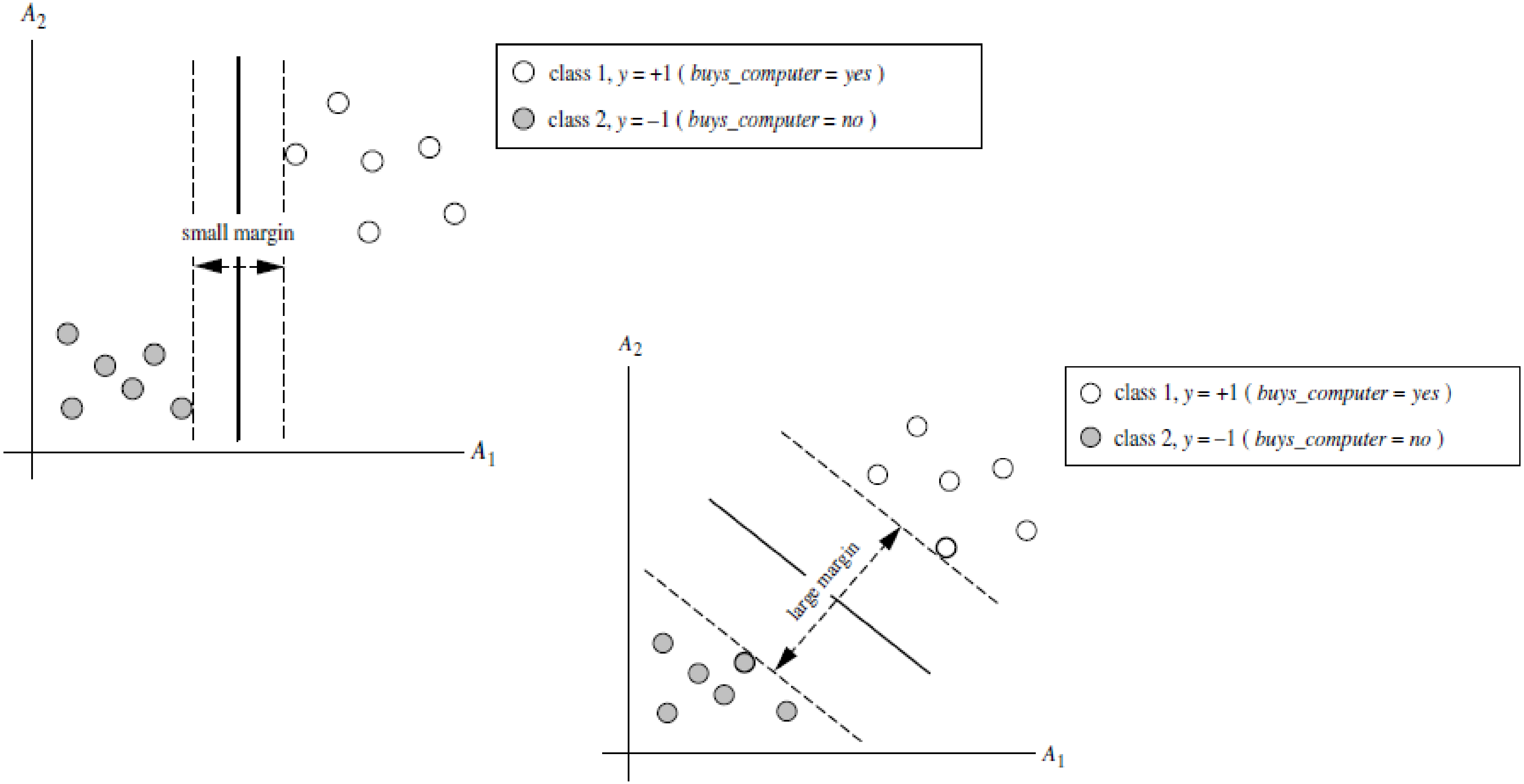
Classification (SVM)


A_2
○ class 1, $y = +1$ (*buys_computer* = *yes*)
● class 2, $y = -1$ (*buys_computer* = *no*)
A_1

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

Source: Han & Kamber (2006)

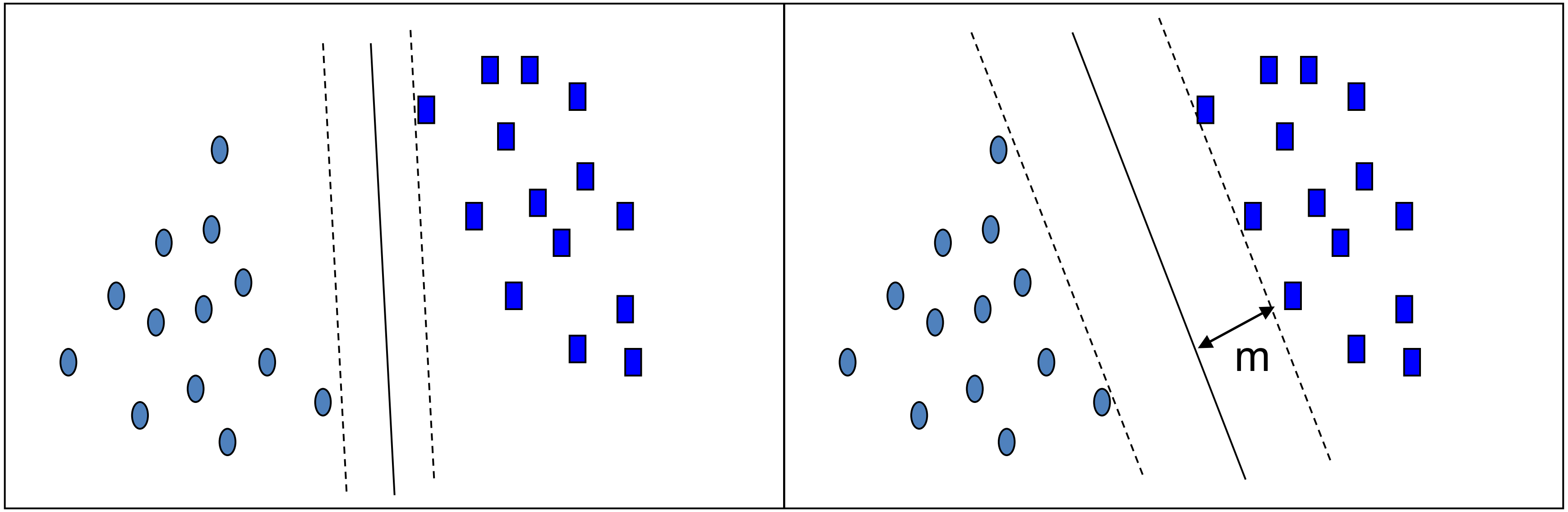
Classification (SVM)


A_2
○ class 1, $y = +1$ (*buys_computer* = *yes*)
● class 2, $y = -1$ (*buys_computer* = *no*)
small margin
A_1
A_2
○ class 1, $y = +1$ (*buys_computer* = *yes*)
● class 2, $y = -1$ (*buys_computer* = *no*)
large margin
A_1

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

Source: Han & Kamber (2006)

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)

Source: Han & Kamber (2006)

SVM—Linearly Separable

- A separating hyperplane can be written as

  $$\mathbf{W} \bullet \mathbf{X} + b = 0$$

  where **W**={w_1, w_2, …, 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

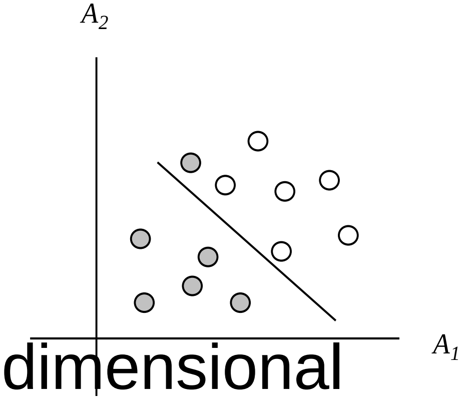
Source: Han & Kamber (2006)

Why Is SVM Effective on High Dimensional Data?

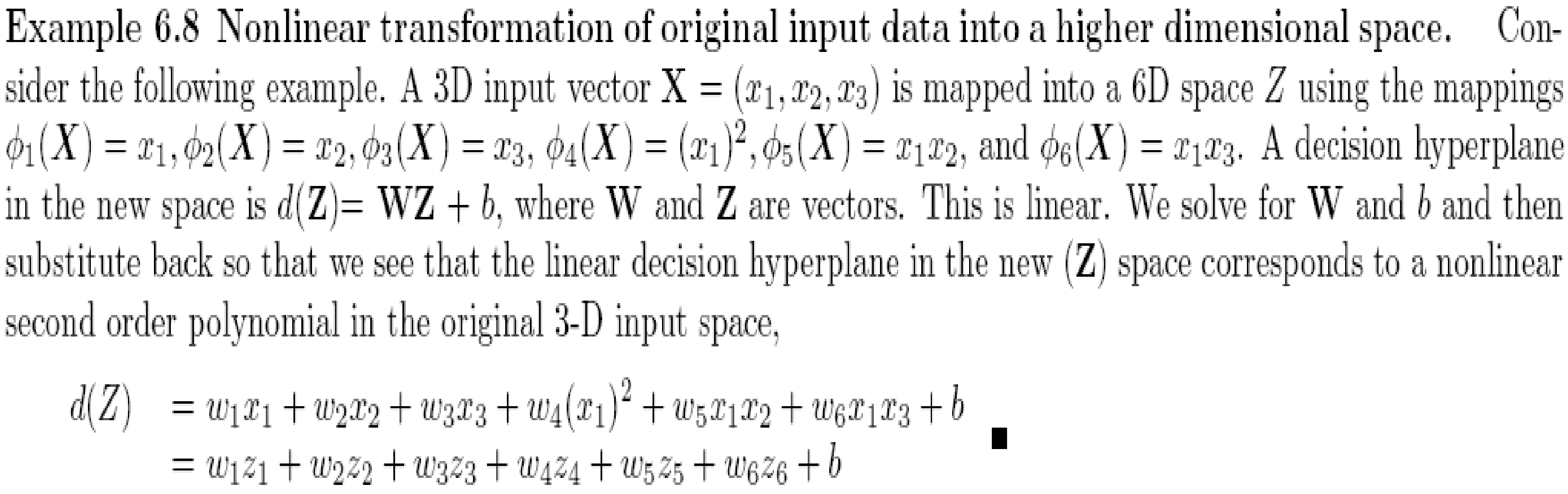
- The complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data
- The support vectors are the essential or critical training examples — they lie closest to the decision boundary (MMH)
- If all other training examples are removed and the training 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

Source: Han & Kamber (2006)

SVM—Linearly Inseparable



- Transform the original input data into a higher dimensional space



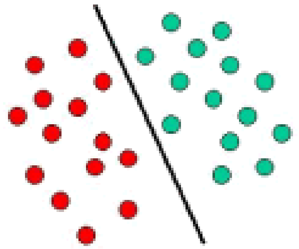
Example 6.8 Nonlinear transformation of original input data into a higher dimensional space. Consider the following example. A 3D input vector $\mathbf{X} = (x_1, x_2, x_3)$ is mapped into a 6D space Z using the mappings $\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3$, $\phi_4(X) = (x_1)^2, \phi_5(X) = x_1x_2$, and $\phi_6(X) = x_1x_3$. 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,

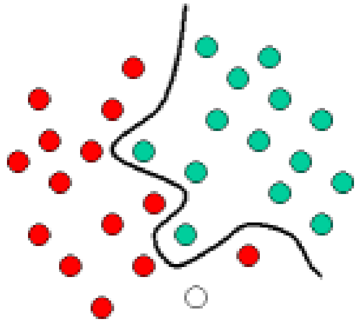
$$\begin{aligned} d(Z) &= w_1x_1 + w_2x_2 + w_3x_3 + w_4(x_1)^2 + w_5x_1x_2 + w_6x_1x_3 + b \\ &= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + b \end{aligned} \qquad \blacksquare$$

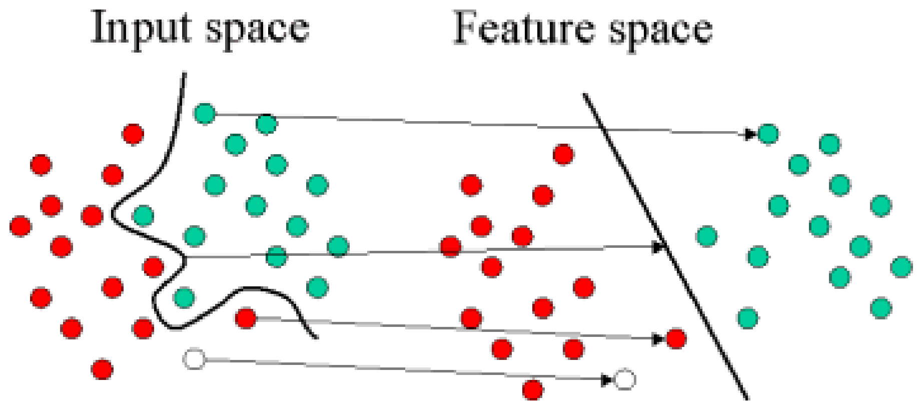
- Search for a linear separating hyperplane in the new space

Source: Han & Kamber (2006)

Mapping Input Space to 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)

Source: Han & Kamber (2006)

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)

Source: Han & Kamber (2006)

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.

Source: Han & Kamber (2006)

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)

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

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

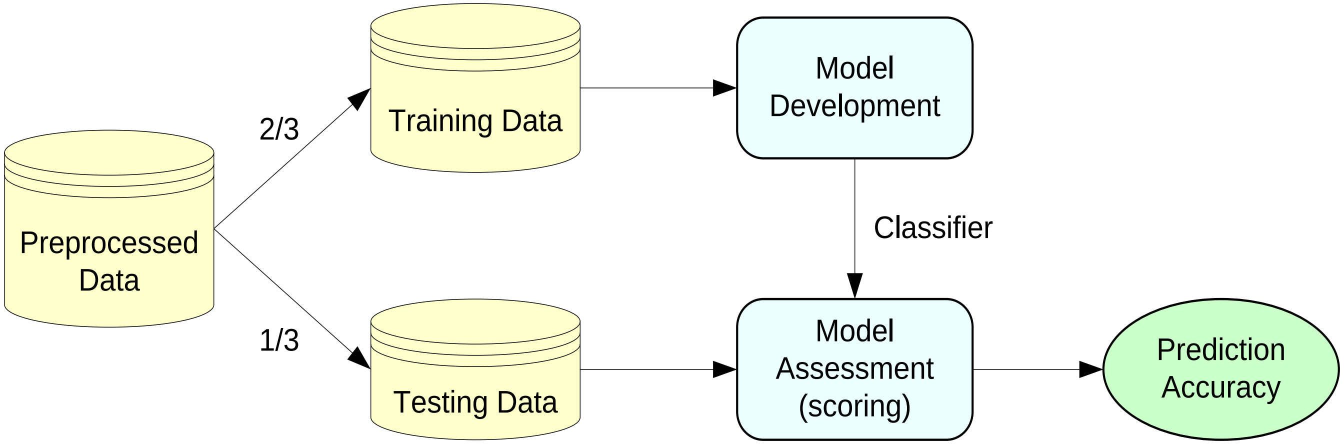
$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

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

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

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%])

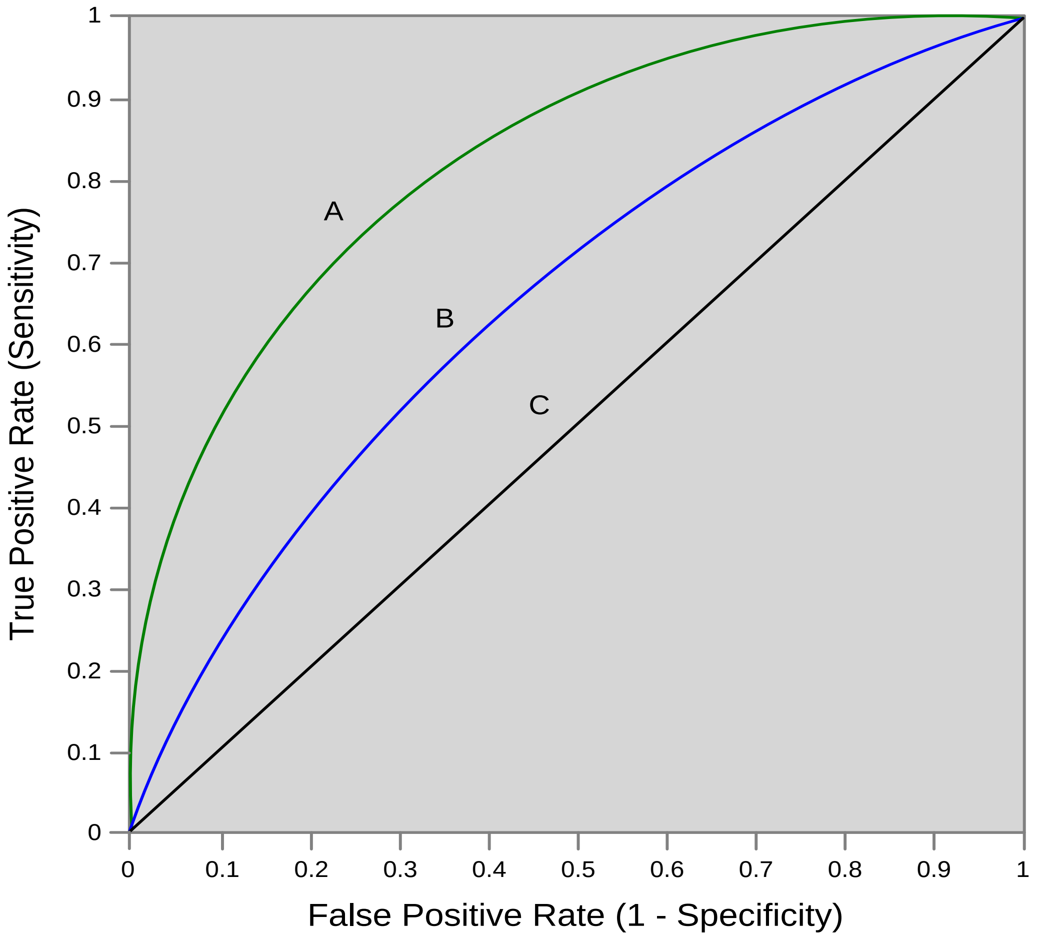
Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

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

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Estimation Methodologies for Classification – ROC Curve



Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Summary

- Classification and Prediction
- Decision Tree
- Support Vector Machine (SVM)
- Evaluation (Accuracy of Classification Model)

Source: Han & Kamber (2006)

References

- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, 2006, Elsevier
- Efraim Turban, Ramesh Sharda, Dursun Delen, Decision Support and Business Intelligence Systems, Ninth Edition, 2011, Pearson.