Data Mining

Lecture 5: Classification

Primary References

- *Michael J. A. Berry and Gordon S. Linoff (2004), **Data Mining Techniques** for Marketing, Sales, and Customer Relationship Management, 2nd ed., Wiley*
- ***Introduction to Data Mining and Knowledge Discovery**, Third Edition, ISBN: 1-892095-02-5 (Can be downloaded via website for free)*
- *Tan, P., Steinbach, M., and Kumar, V. (2006) Introduction to Data Mining, 1st edition, Addison-Wesley, ISBN: 0-321-32136-7.*
- *Vasant Dhar and Roger Stein, Prentice-Hall (1997), Seven Methods for Transforming Corporate Data Into Business Intelligence*
- *H. Witten and E. Frank (2005), Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, Morgan Kaufmann, ISBN: 0-12-088407-0, closely tied to the WEKA software.*
- *Ethem ALPAYDIN, Introduction to Machine Learning, The MIT Press, October 2004, ISBN 0-262-01211-1*
- *J. Han and M. Kamber (2000) Data Mining: Concepts and Techniques, Morgan Kaufmann. Database oriente.* — Slides for Textbook —Classification, http://www.cs.sfu.ca
- 資料探勘，丁一賢(2005)

Examples of Classification Task

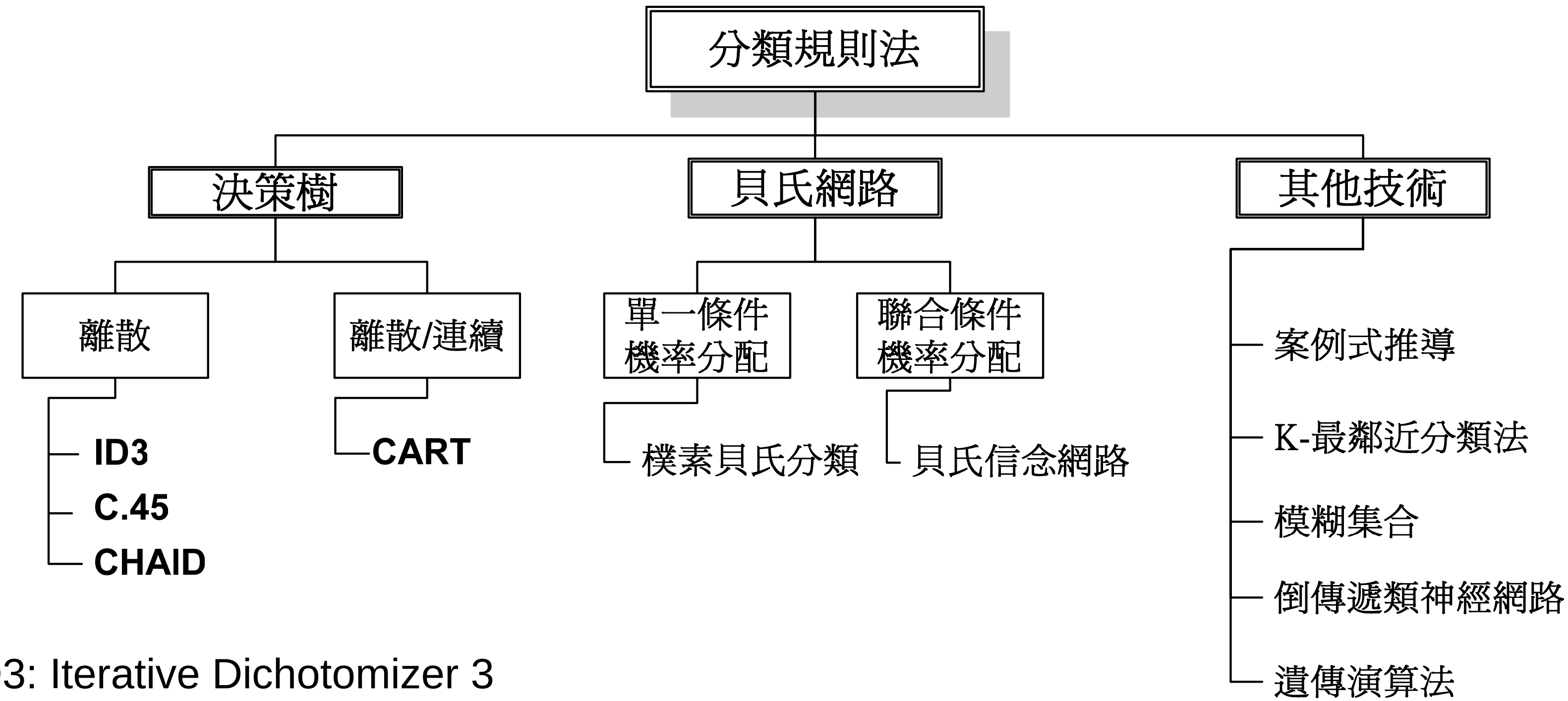
- Predicting tumor cells as benign or malignant

- Classifying credit card transactions
  as legitimate or fraudulent

- Classifying secondary structures of protein
  as alpha-helix, beta-sheet, or random
  coil

- Categorizing news stories as finance,
  weather, entertainment, sports, etc

Classification



ID3: Iterative Dichotomizer 3

CART: Classification and Regression Trees

CHAID: Chi-Square Automatic Interaction Detector

Ref: 資料採勘，丁一賢 (2005)

決策樹演算法

- ID3 (Iterative Dichotomizer 3)
  - 可處理離散型資料。
  - 兼顧高分類正確率以及降低決策樹的複雜度。
  - 必須將連續型資料作離散化的程序。
- CART (Classification and Regression Trees)
  - 是以每個節點的動態臨界值作為條件判斷式。
  - CART藉由單一輸入的變數函數，在每個節點分隔資料，並建立一個二元決策樹 。
  - CART是使用 Gini Ratio來衡量指標，如果分散的指標程度很高，表示資料中分佈許多類別，相反的，如果指標程度越低，則代表單一類別的成員居多。

決策樹演算法

- C4.5
  - 改良自ID3演算法。
  - 先建構一顆完整的決策樹，再針對每一個內部節點，依使用者定義的預估錯誤率(Predicted Error Rate)來作決策樹修剪的動作。
  - 不同的節點，特徵值離散化結果是不相同的。
- CHAID (Chi-Square Automatic Interaction Detector)
  - 利用卡方分析(Chi-Square Test)預測二個變數是否需要合併，如能夠產生最大的類別差異的預測變數，將成為節點的分隔變數。
  - 計算節點中類別的 P值 (P-Value)，以P值大小來決定決策樹是否繼續生長，所以不需像C4.5或CART要再做決策樹修剪的動作。

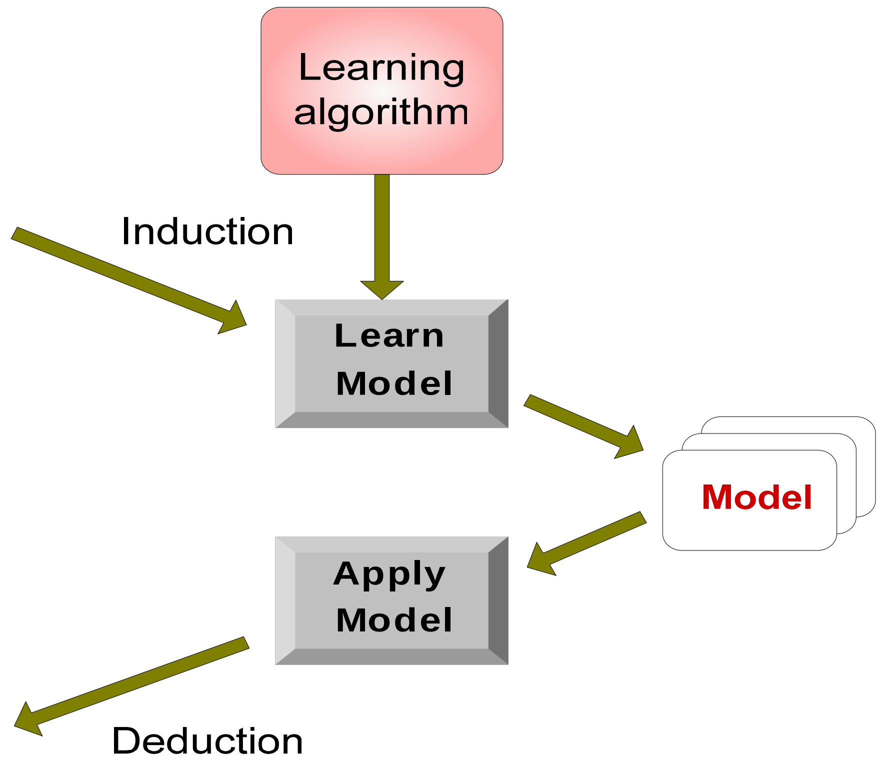
決策樹演算法之比較

	作者	資料屬性	分割規則	修剪樹規則
ID3	**Quinlan（1979）**	離散型資料	**Entropy、** **Gain Ratio**	**Predicted Error Rate**
C4.5	**Quinlan（1993）**	離散型資料	**Gain Ratio**	**Predicted Error Rate**
CHAID	**Kass（1980）**	離散型資料	**Chi-Square Test**	**No Pruning**
CART	**Briemen（1984）**	離散與 連續型資料	**Gini Index**	**Entire Error Rate**

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

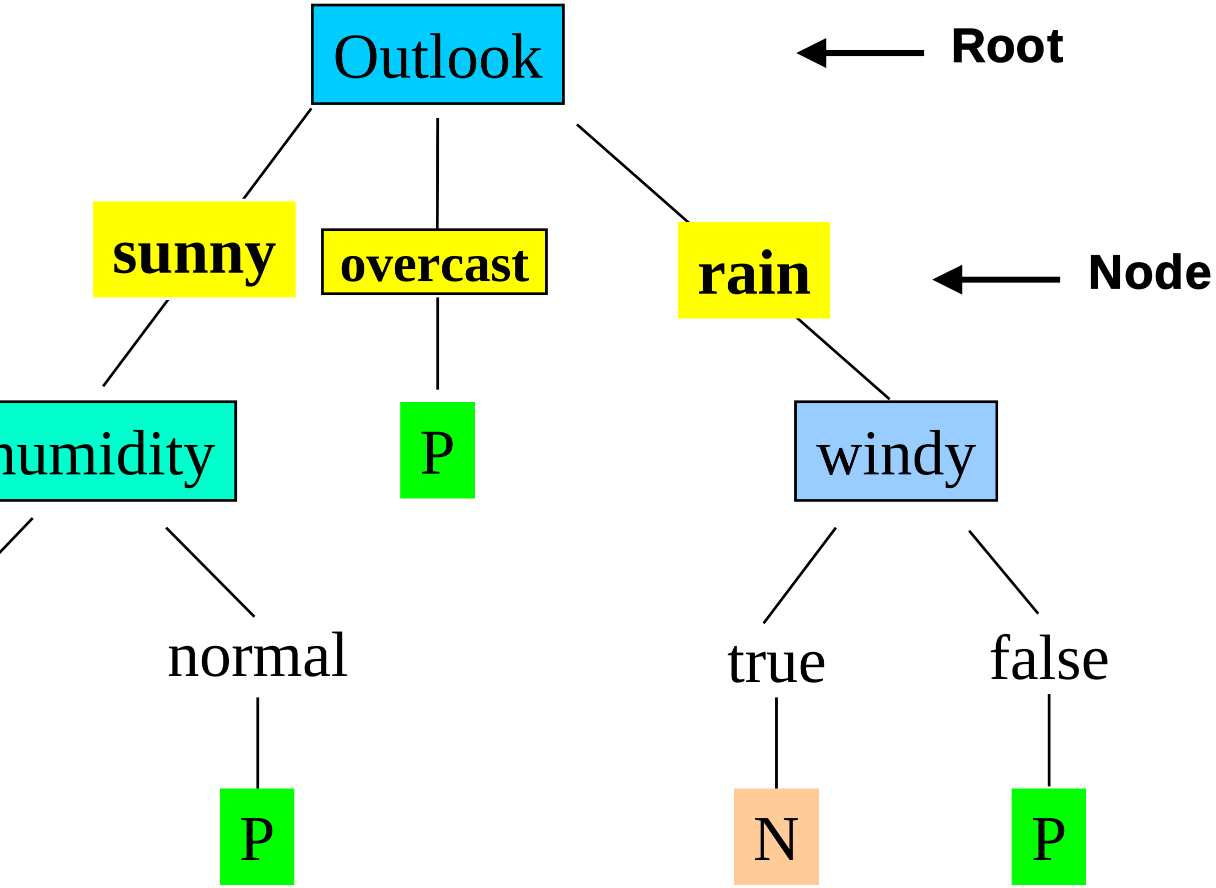
Supervised vs. Unsupervised Learning

- 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

Training Dataset

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Output: A Decision Tree for *"Play tennis or not"*



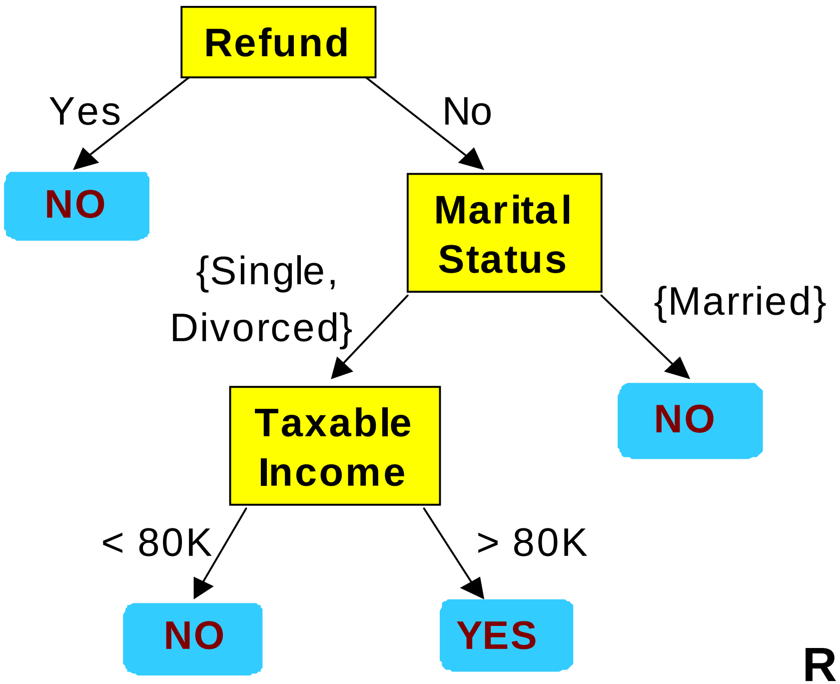
Another Example

- Rule-based Classifier:

```
If tear production rate = reduced then recommendation = none.
If age = young and astigmatic = no and tear production rate = normal
   then recommendation = soft
If age = pre-presbyopic and astigmatic = no and tear production
   rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope and
   astigmatic = no then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no and
   tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes and
   tear production rate = normal then recommendation = hard
If age = young and astigmatic = yes and tear production rate =
    normal
   then recommendation = hard
If age = pre-presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
```

Rules are mutually exclusive and exhaustive before pruning.

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

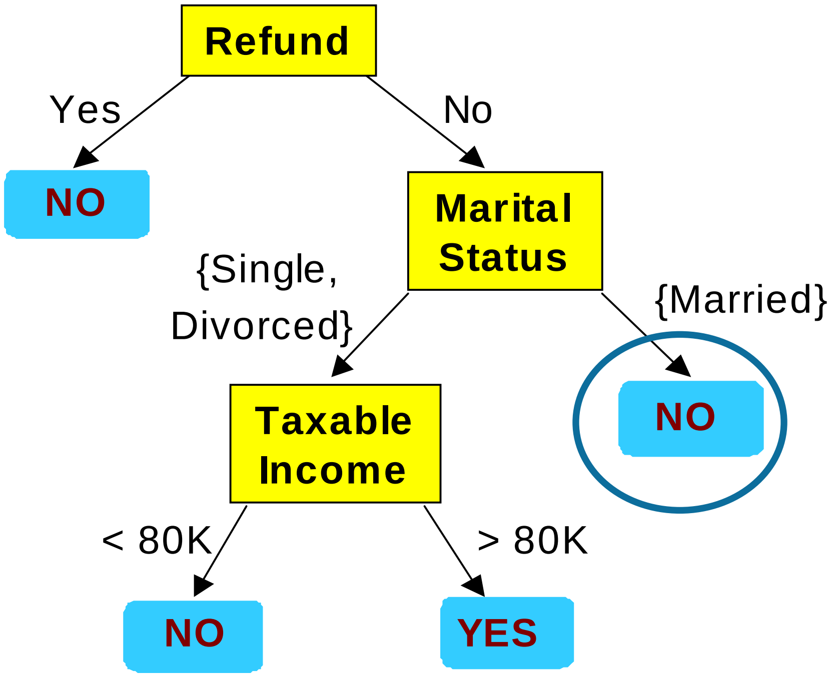
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified



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

Initial Rule: (Refund=No) $\wedge$ (Status=Married) $\rightarrow$ No

Simplified Rule: (Status=Married) $\rightarrow$ No

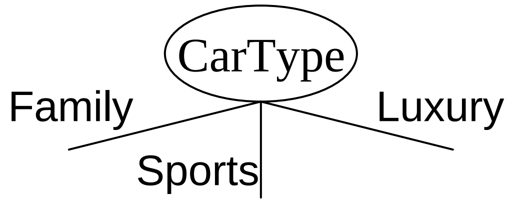
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

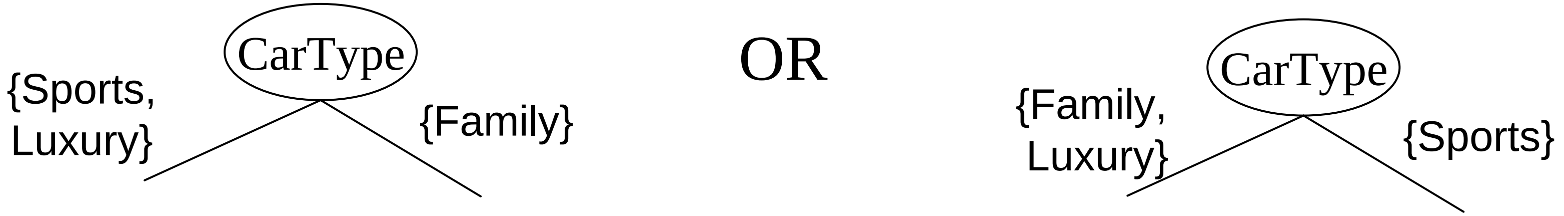
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

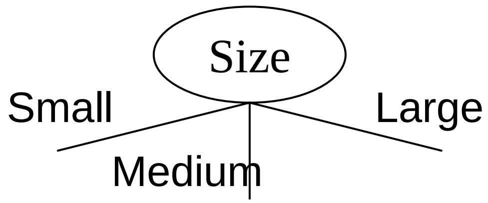


- Binary split: Divides values into two subsets. Need to find optimal partitioning.

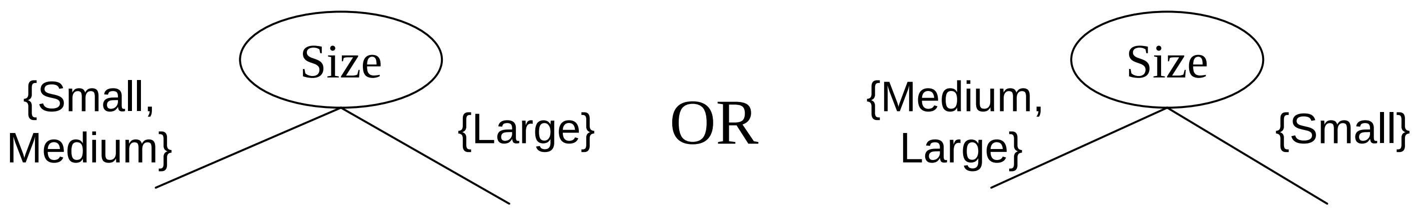


Splitting Based on Ordinal Attributes

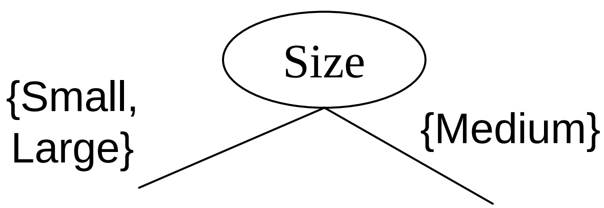
- Multi-way split: Use as many partitions as distinct values.



- Binary split: Divides values into two subsets. Need to find optimal partitioning.



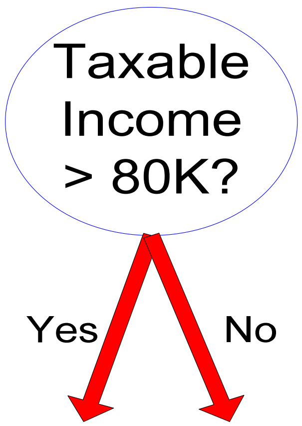
- What about this split?



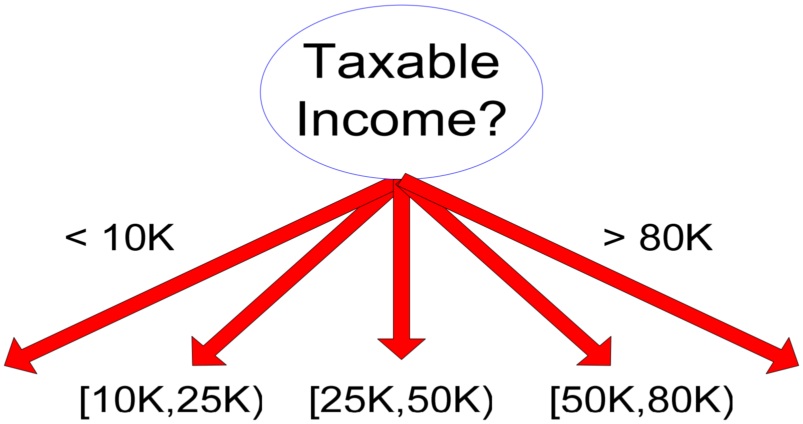
Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split

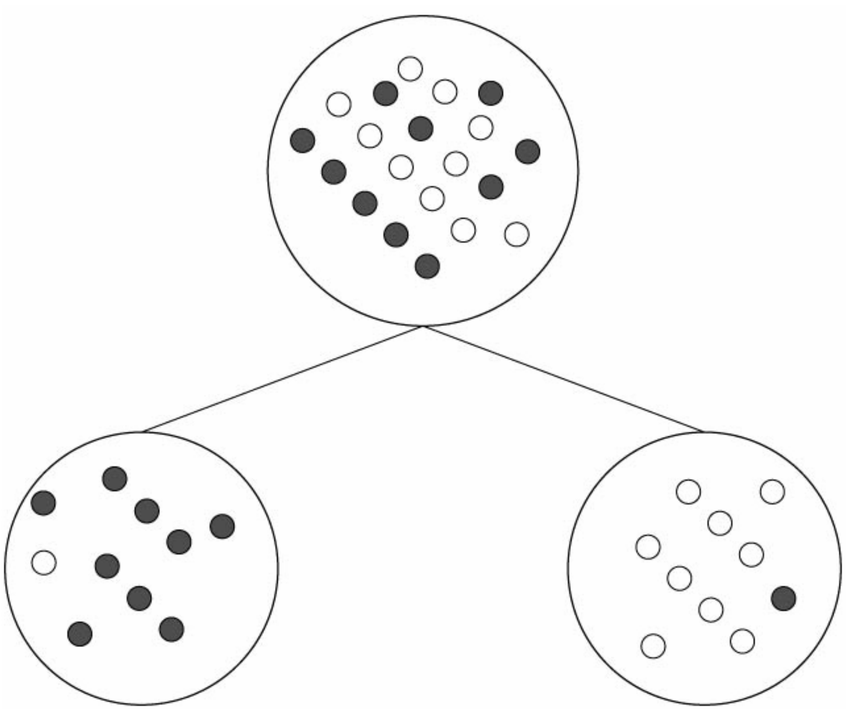


(ii) Multi-way split

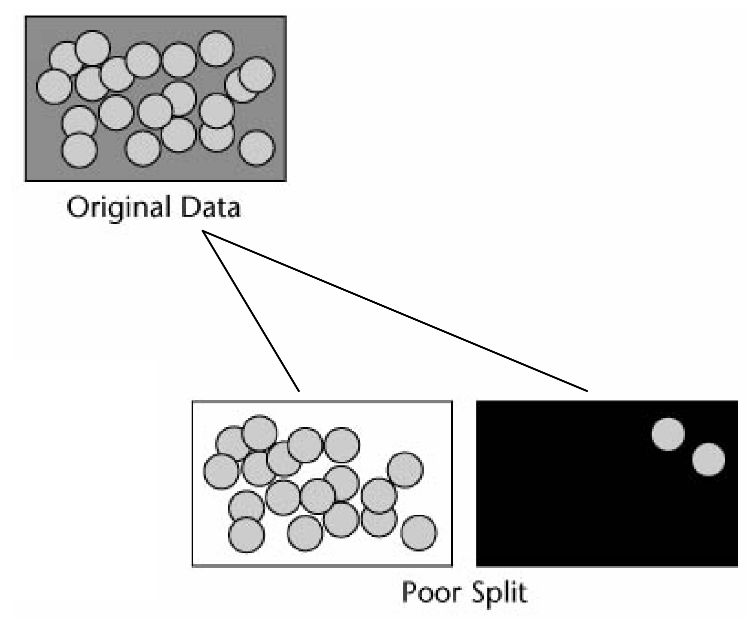
Algorithm for Decision Tree Induction

- 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

Example: Good & Poor Splits



Good Split



Tests for Choosing Best Split

- Purity (Diversity) Measures:
  - Gini (population diversity)
  - Entropy (information gain)
  - Information Gain Ratio
  - Chi-square Test

Attribute Selection Measure

- Information gain (ID3/C4.5)
  - All attributes are assumed to be categorical
  - Can be modified for continuous-valued attributes
- Gini index (IBM IntelligentMiner)
  - All attributes are assumed continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
  - Can be modified for categorical attributes

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there are two classes, *P* and *N*
  - Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

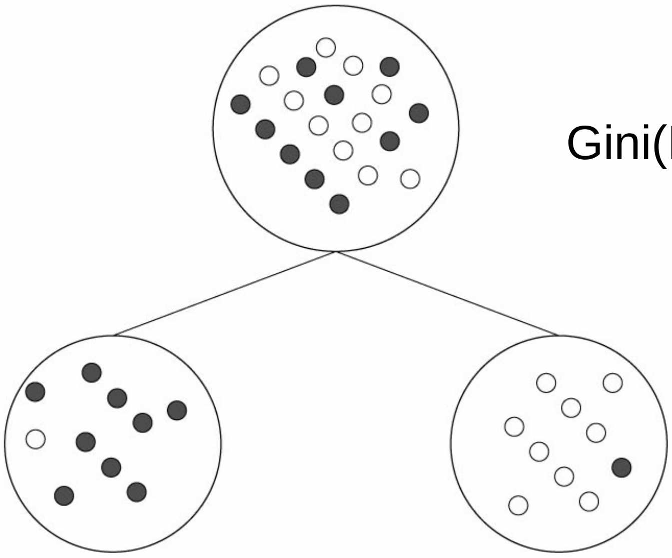
$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Gini Index (IBM Intelligent Miner)

- 樣本分佈愈平均，資訊量愈大，亂度愈大，Gini值愈大，

$$Gini(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

p(j | t) is the relative frequency of class j at node t).



Gini(Root Node) = 1- $(0.5^2 + 0.5^2)$ = 0.5

$Gini_1$ (Leaf node) = 1 – $(0.1^2 + 0.9^2)$ = 0.18

$Gini_2$ (Leaf node) = 1 – $(0.1^2 + 0.9^2)$ = 0.18

$Gini_t$ (Leaf node) = 10/20*0.18 + 10/20*0.18 = 0.18

Information Gain in Decision Tree Induction

- Assume that using attribute A a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on A $Gain(A) = I(p,n) - E(A)$

Attribute Selection by Information Gain Computation

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for *age*:

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

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

Hence

$$Gain(age) = I(p,n) - E(age)$$

= 0.94 – 0.69 = 0.25

Similarly

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

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

Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
  - Postpruning: Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees

- DEMO 1
  - http://www.cs.ualberta.ca/~aixplore/learning/DecisionTrees/InterArticle/2-DecisionTree.html
- Demo 2
  - SAS Enterprise Miner
  - Ex. ID Potential Customers

Decision Tree Advantages

1. Easy to understand
2. Map nicely to a set of business rules
3. Applied to real problems
4. Make no prior assumptions about the data
5. Able to process both numerical and categorical data

Decision Tree Disadvantages

1. Output attribute must be categorical
2. Limited to one output attribute
3. Decision tree algorithms are unstable
4. Trees created from numeric datasets can be complex

Bayesian Theorem

- 假設 $C_1, C_2, C_3 .... C_n$ 是樣本空間(sample space) S 的分割，且有一事件 A，則有兩定理存在：
  - 總機率法則(Law of Total Probability) $P(A) = \sum P(C_i) P(A|C_i)$
  - 貝氏定理(Bayes' Rule)

$$P(C_j|A) = \frac{P(C_j)P(A|C_j)}{\sum P(C_i)P(A|C_i)}$$

  - 其中
    - $P(C_i)$ ：事前機率(Prior Probability)
    - $P(A|C_i)$：樣本機率(Sample Probability)
    - $P(C_j|A)$：事後機率(Posterior Probability)

•Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Bayesian classification

- The classification problem may be formalized using a-posteriori probabilities:
- P(C|X) = prob. that the sample tuple $X=<x_1,\ldots,x_k>$ is of class C.
- E.g. P(class=N | outlook=sunny,windy=true,...)
- Idea: assign to sample X the class label C such that P(C|X) is maximal

Estimating a-posteriori probabilities

- Bayes theorem:

$$P(C|X) = P(X|C) \cdot P(C) \;/\; P(X)$$

- $P(X)$ is constant for all classes
- $P(C)$ = relative freq of class C samples
- C such that $P(C|X)$ is maximum =
  C such that $P(X|C) \cdot P(C)$ is maximum
- Problem: computing $P(X|C)$ is unfeasible!

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes $P(C| A_1, A_2,\ldots,A_n)$

- Can we estimate $P(C| A_1, A_2,\ldots,A_n)$ directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$

  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C) \, P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \ldots P(A_n | C_j)$
  - Can estimate $P(A_i | C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \Pi P(A_i | C_j)$ is maximal.

Play-tennis example: estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14	
P(n) = 5/14	

outlook	
P(sunny|p) = 2/9	**P(sunny|n) = 3/5**
P(overcast|p) = 4/9	**P(overcast|n) = 0**
P(rain|p) = 3/9	**P(rain|n) = 2/5**
temperature	
P(hot|p) = 2/9	**P(hot|n) = 2/5**
P(mild|p) = 4/9	**P(mild|n) = 2/5**
P(cool|p) = 3/9	**P(cool|n) = 1/5**
humidity	
P(high|p) = 3/9	**P(high|n) = 4/5**
P(normal|p) = 6/9	**P(normal|n) = 2/5**
windy	
P(true|p) = 3/9	**P(true|n) = 3/5**
P(false|p) = 6/9	**P(false|n) = 2/5**

Play-tennis example: classifying X

- An unseen sample X = <rain, hot, high, false>

- $P(X|p) \cdot P(p) = P(rain|p) \cdot P(hot|p) \cdot P(high|p) \cdot P(false|p) \cdot P(p) = 3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$
- $P(X|n) \cdot P(n) = P(rain|n) \cdot P(hot|n) \cdot P(high|n) \cdot P(false|n) \cdot P(n) = 2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = 0.018286$

- Sample X is classified in class n (don't play)

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No) $\times$ P(Married| Class=No) $\times$ P(Income=120K| Class=No) = 4/7 $\times$ 4/7 $\times$ 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes) $\times$ P(Married| Class=Yes) $\times$ P(Income=120K| Class=Yes) = 1 $\times$ 0 $\times$ 1.2 $\times$ 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)
=> Class = No

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

Naïve Bayes (Summary)

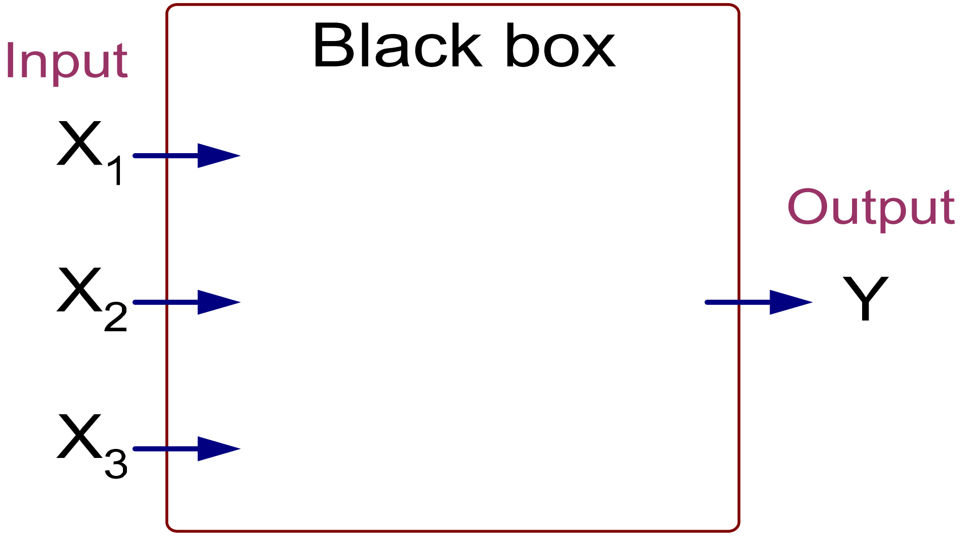
- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

Neural Networks

- Advantages
  - prediction accuracy is generally high
  - robust, works when training examples contain errors
  - output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
  - fast evaluation of the learned target function
- Criticism
  - long training time
  - difficult to understand the learned function (weights)
  - not easy to incorporate domain knowledge

Artificial Neural Networks (ANN)

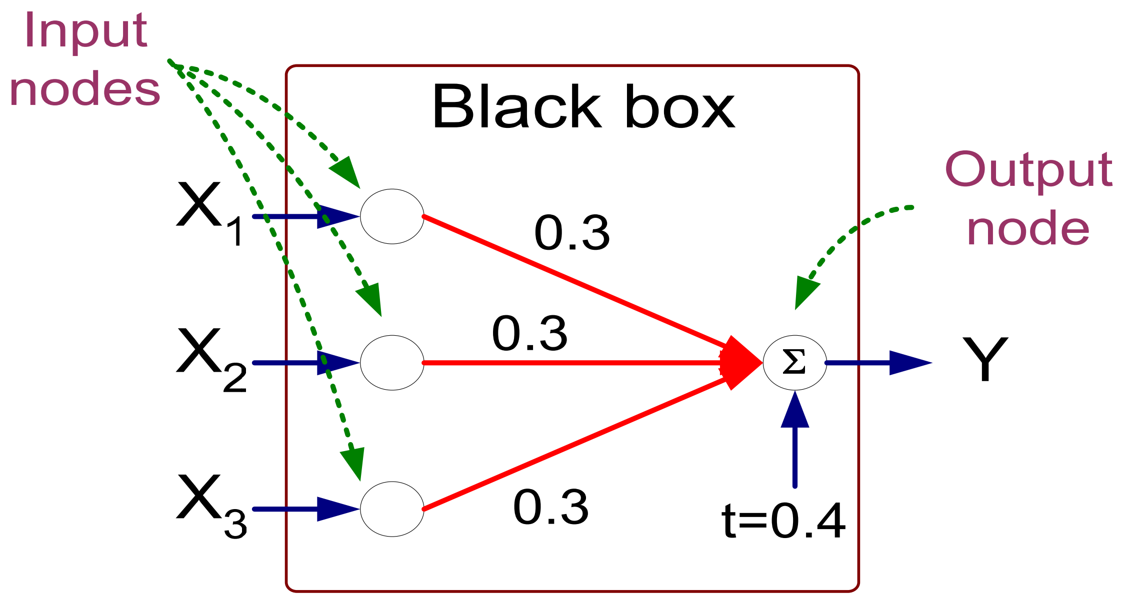
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

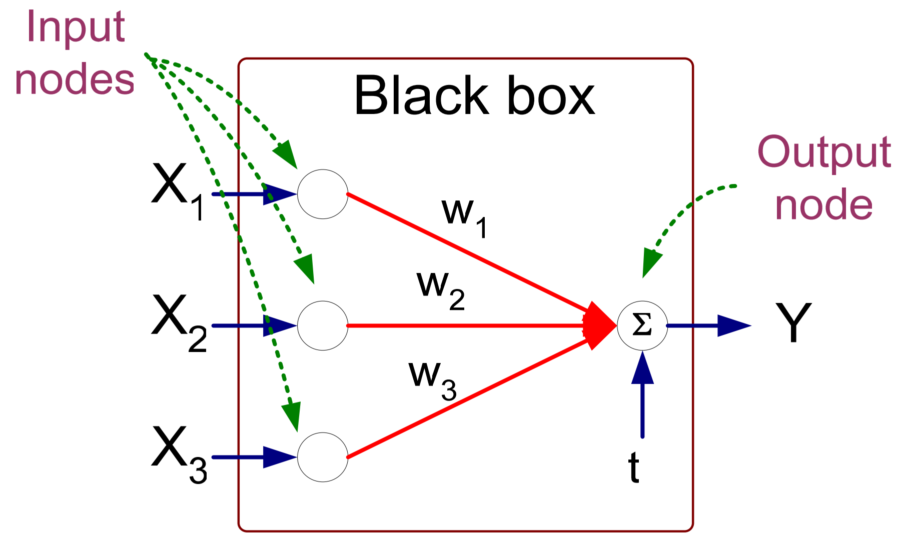
$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links

- Output node sums up each of its input value according to the weights of its links

- Compare output node against some threshold t

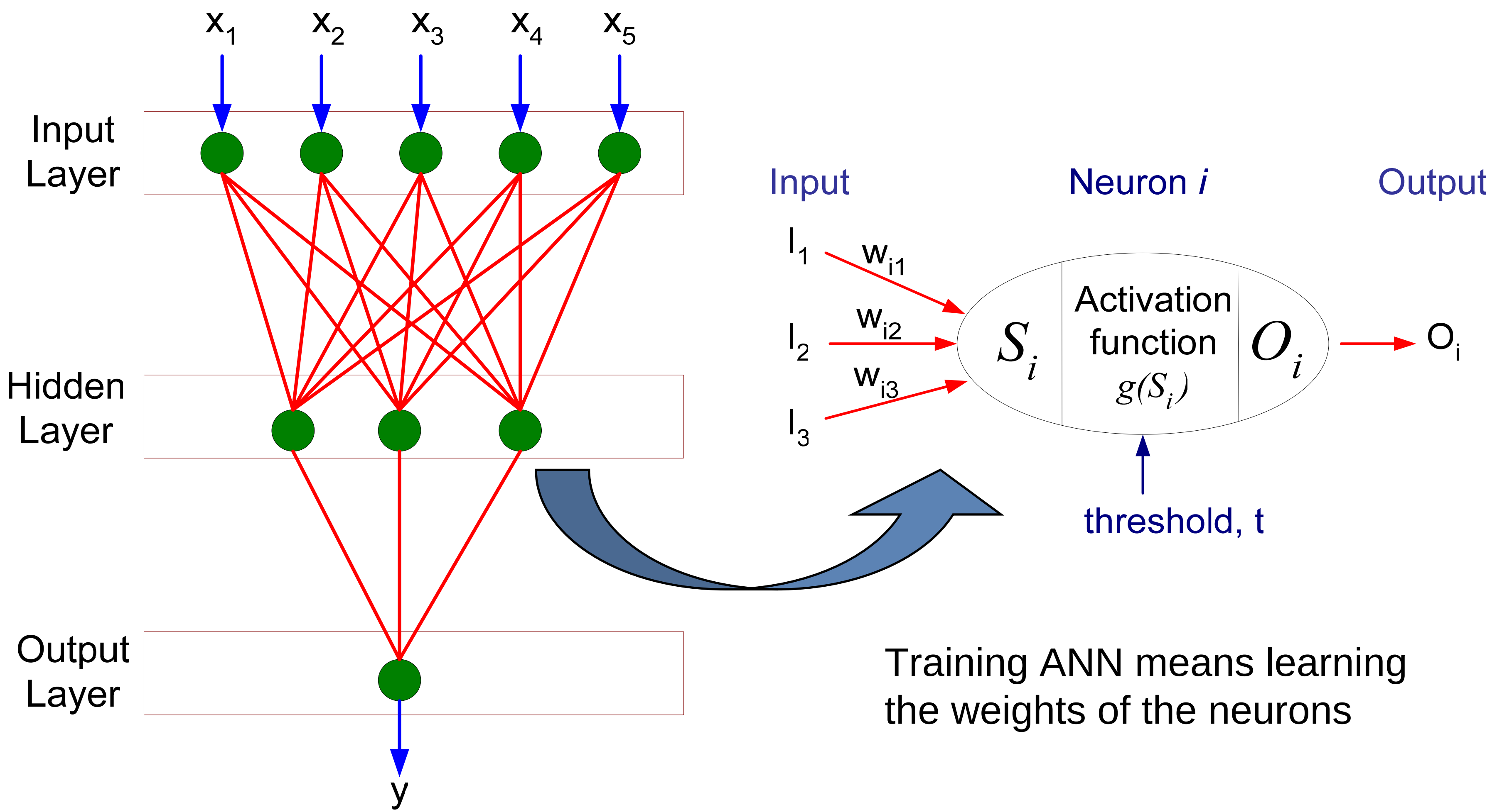


Perceptron Model

$$Y = I(\sum_i w_i X_i - t) \quad \text{or}$$

$$Y = sign(\sum_i w_i X_i - t)$$

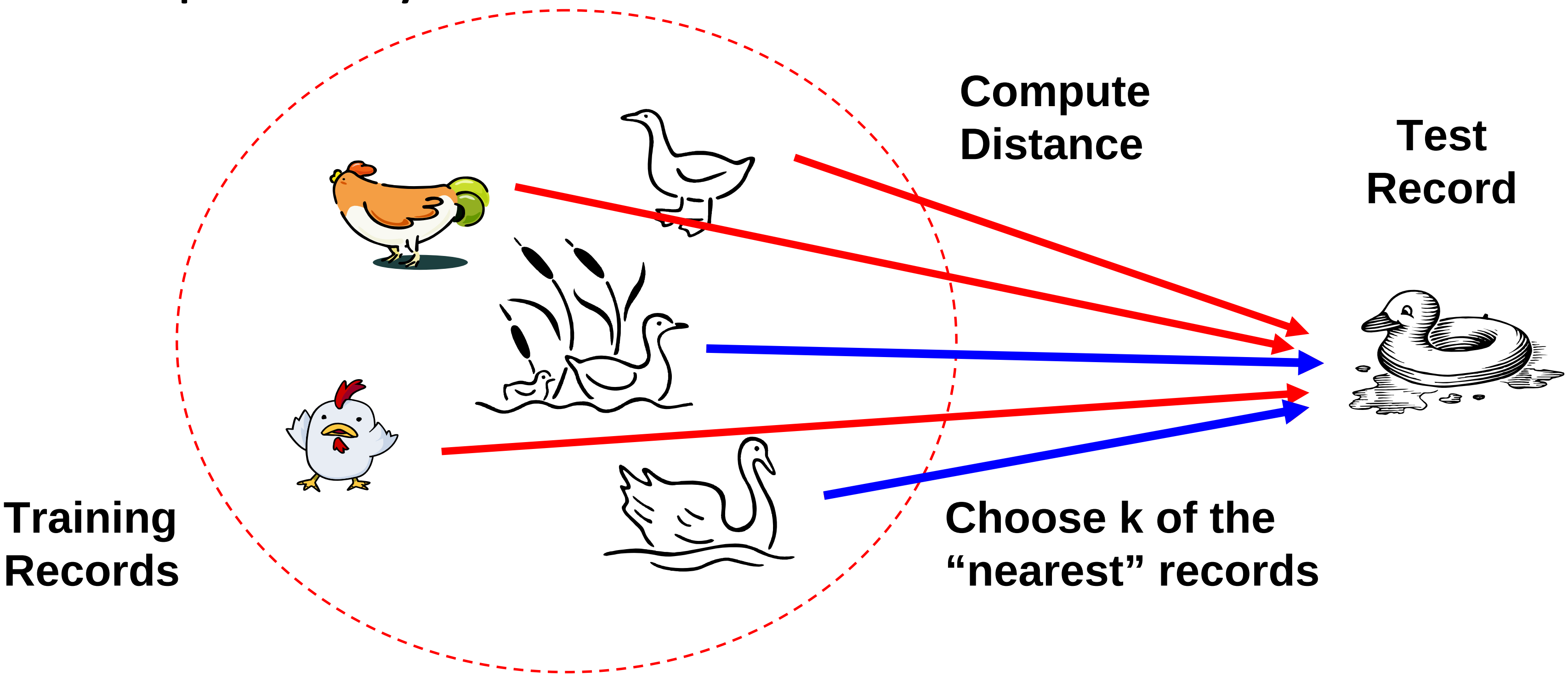
General Structure of ANN



Training ANN means learning the weights of the neurons

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



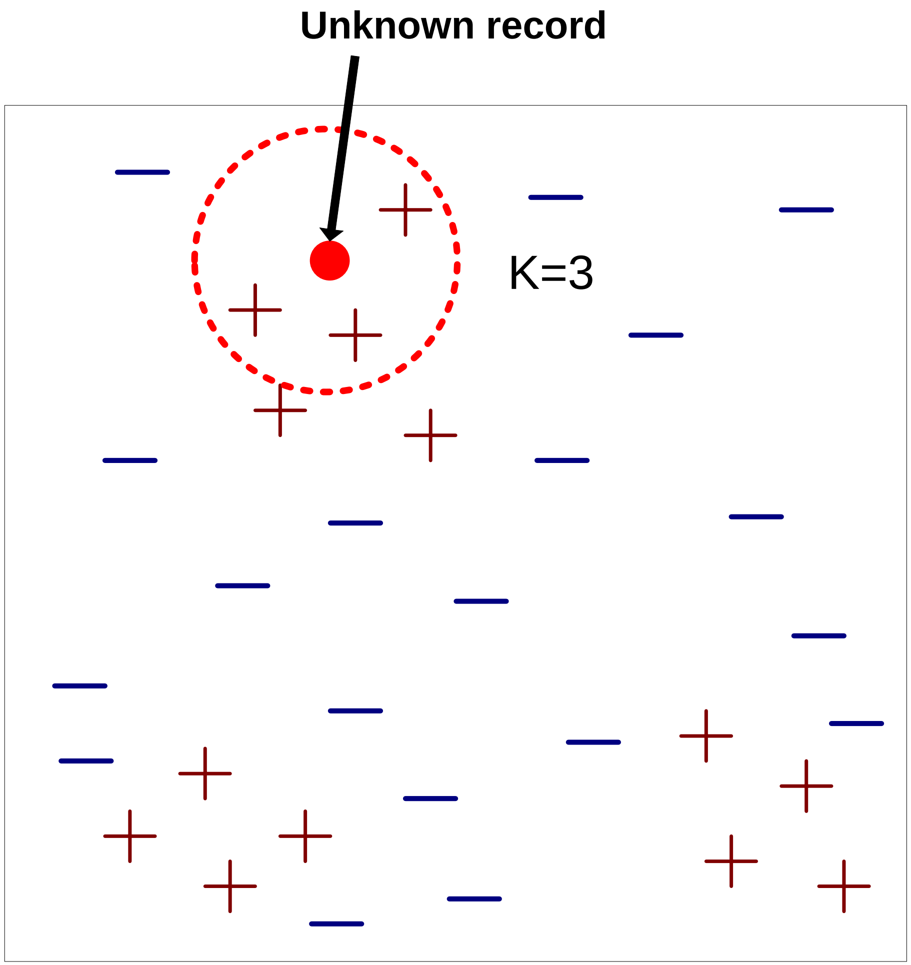
Discussion on the *k*-NN Algorithm

- The k-NN algorithm for continuous-valued target functions
  - Calculate the mean values of the *k* nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query point x_q
    - giving greater weight to closer neighbors

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- Robust to noisy data by averaging k-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes.
  - To overcome it, elimination of the least relevant attributes.

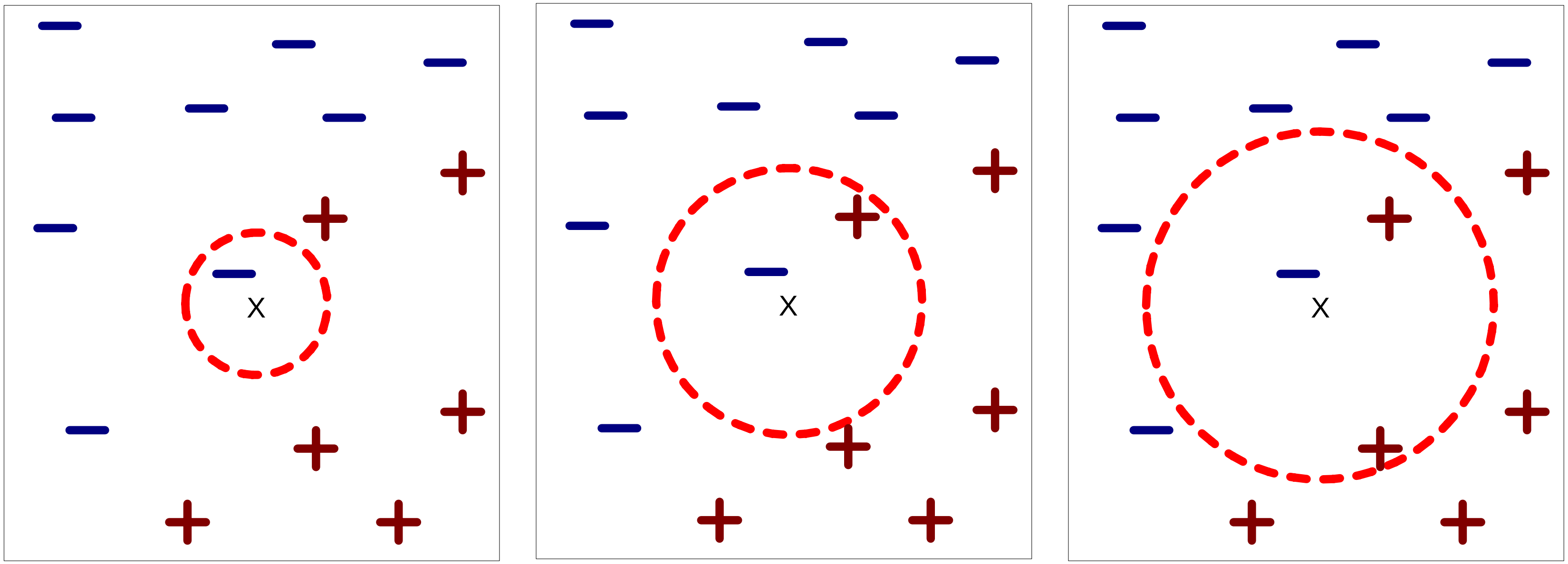
Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest Neighbor Classification

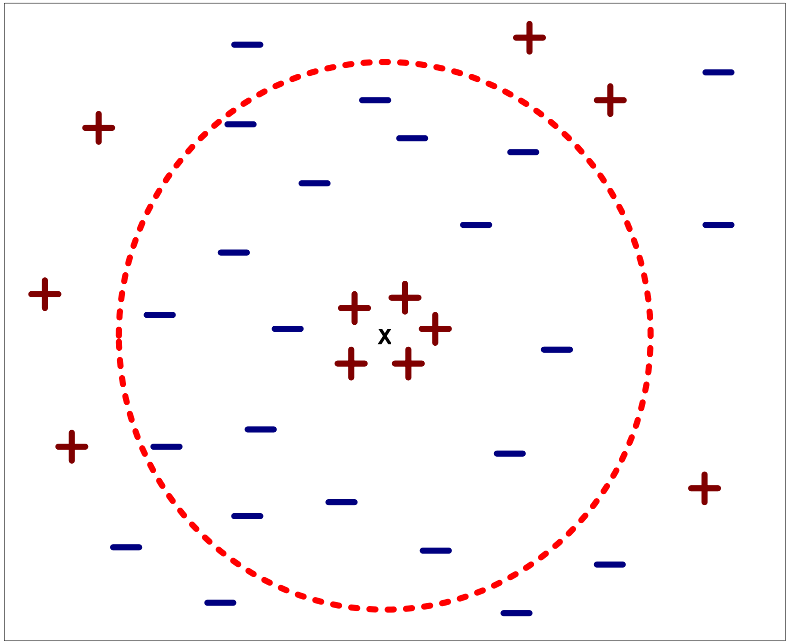
- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



What Is Prediction?

- Prediction is similar to classification
  - First, construct a model
  - Second, use model to predict unknown value
    - Major method for prediction is regression
      - Linear and multiple regression
      - Non-linear regression
      - Logit/probit model
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions

Predictive Modeling in Databases

- Predictive modeling: Predict data values or construct generalized linear models based on the database data.
- Determine the major factors which influence the prediction

迴歸分析

- 迴歸分析為統計分析的一種方法，主要在了解自變數（Independent Variable）與依變數（Dependent Variable）間之數量關係。
- 迴歸分析依自變數個數可分為不同的類型
  - 單一自變數時，則稱為簡單迴歸（Simple Regression）
  - 自變數不只一個時，稱為複迴歸（Multiple Regression）
- 迴歸方程式又可分為
  - 直線迴歸（Linear Regression）
  - 非直線性迴歸

Regress Analysis and Log-Linear Models in Prediction

- <u>Linear regression</u>: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of Y_1, Y_2, ..., X_1, X_2, ...
- <u>Multiple regression</u>: Y = b0 + b1 X1 + b2 X2.
  - Many nonlinear functions can be transformed into the above.
- <u>Log-linear models</u>:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: $p(a, b, c, d) = \alpha_{ab}\, \beta_{ac} \chi_{ad}\, \delta_{bcd}$

Linear Regression

- 決定係數（Coefficient of Determination）則可估以直線迴歸方程式預測依變數的準確度，為相關係數r 的平方。
- 公式 $$r^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}\left(\widehat{Y}_i - \overline{Y}\right)^2}{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}$$
- 決定係數永遠介於0 到1 之間
- 判定係數愈接近1，表示對依變數的解釋能力愈好，迴歸分析預測的結果則愈可靠。
- 判定係數愈接近0，表示預測結果的可信度愈低。

Logit Model

- Logit Model 的特性
  - 當研究結果的依變數是離散型。
  - 用於處理類別資料的問題。
  - 適用於依變數是屬於質化變數(非量化)的迴歸模型。
  - 可克服自變數須服從常態分配的假設 。

e

Logit/Probit Model

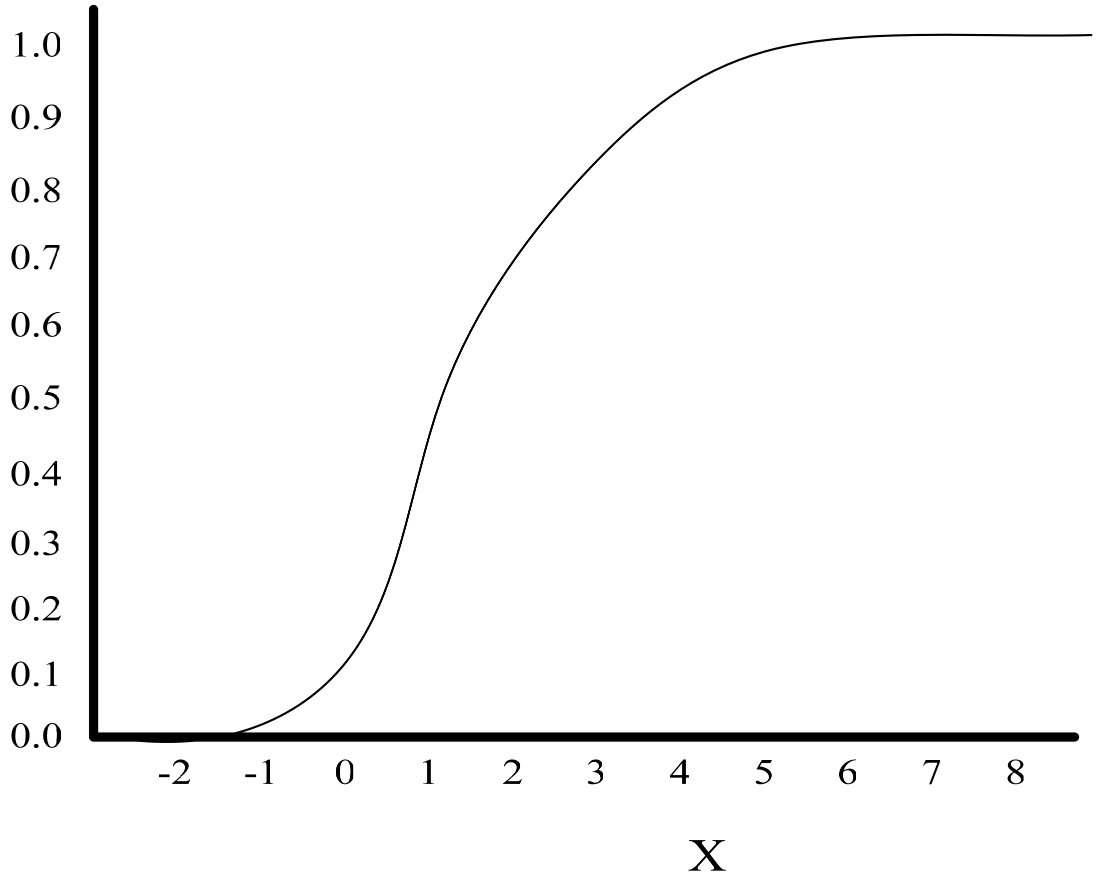
- 一個簡單的範例 (dep. var. is binary output)
  - 資料變數呈現只有成功或失敗兩種可能情況，故令 $P(X)$ 表示某種事件發生的機率，它受因素X的影響，若 X 與 $P(X)$ 關係滿足：

$$P(X) = \frac{e^{f(x)}}{1+e^{f(x)}} \quad , \quad 0 \le P(x) \le 1$$

其中，e 為常數， $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots.. + \beta_k x_k$

Odds ratio = p/(1-p) = $e^{f(x)}$, logit = ln(p/(1-p))=f(x)

Logistic Regression函數圖形



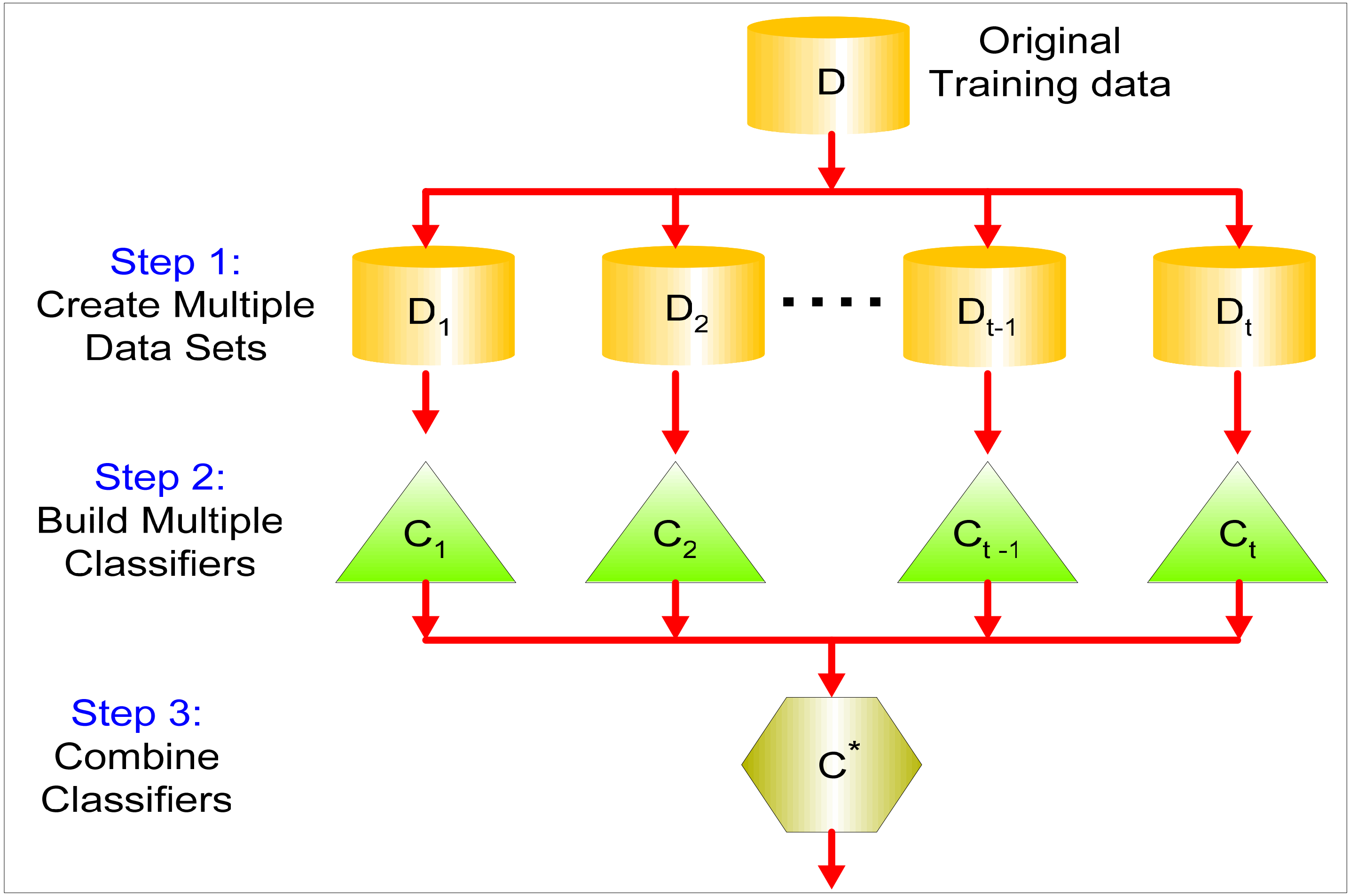
Classification Accuracy: Estimating Error Rates

- Partition: Training-and-testing
  - use two independent data sets, e.g., training set (2/3), test set(1/3)
  - used for data set with large number of samples
- Cross-validation
  - divide the data set into *k* subsamples
  - use *k-1* subsamples as training data and one sub-sample as test data --- *k*-fold cross-validation
  - for data set with moderate size
- Bootstrapping (leave-one-out)
  - for small size data

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea



Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Summary

- Classification is an extensively studied [kdnugget.com] problem (mainly in statistics, machine learning & neural networks)
- Classification is probably one of the most widely used data mining techniques with a lot of extensions
- Scalability is still an important issue for database applications: thus combining classification with database techniques should be a promising topic
- Research directions: classification of non-relational data, e.g., text, spatial, multimedia, etc..