Data Mining 2025

Classification II

Dept. of Computer Science and Information Engineering

National Cheng Kung University

Kun-Ta Chuang

ktchuang@mail.ncku.edu.tw

Rule-Based Classifier

- Classify records by using a collection of "if…then…" rules
- Rule: (*Condition*) → *y*
  - where
    - *Condition* is a conjunctions of attributes
    - *y* is the class label
  - *LHS*: rule antecedent or condition
  - *RHS*: rule consequent
  - Examples of classification rules:
    - (Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) → Birds
    - (Taxable Income < 50K) $\wedge$ (Refund=Yes) → Evade=No

Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds

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

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

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

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

R5: (Live in Water = sometimes) → Amphibians

Application of Rule-Based Classifier

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

- R1: (Give Birth = no) ^ (Can Fly = yes) → Birds
- R2: (Give Birth = no) ^ (Live in Water = yes) → Fishes
- R3: (Give Birth = yes) ^ (Blood Type = warm) → Mammals
- R4: (Give Birth = no) ^ (Can Fly = no) → Reptiles
- R5: (Live in Water = sometimes) → Amphibians

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

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- Coverage of a rule:
  - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
  - Fraction of records that satisfy both the antecedent and consequent of a rule

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

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

How does Rule-based Classifier Work?

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

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

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

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

R5: (Live in Water = sometimes) → Amphibians

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

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

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Characteristics of Rule-Based Classifier

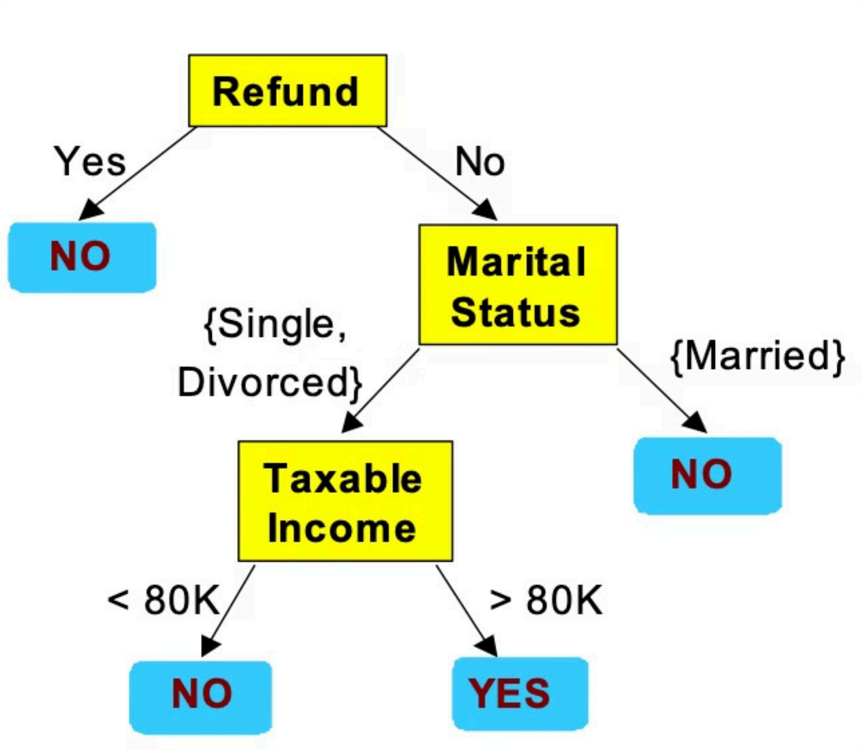
Mutually exclusive rules

- Classifier contains mutually exclusive rules if the rules are independent of each other
- Every record is covered by at most one rule

Exhaustive rules

- Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
- Each record is covered by at least one rule

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

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

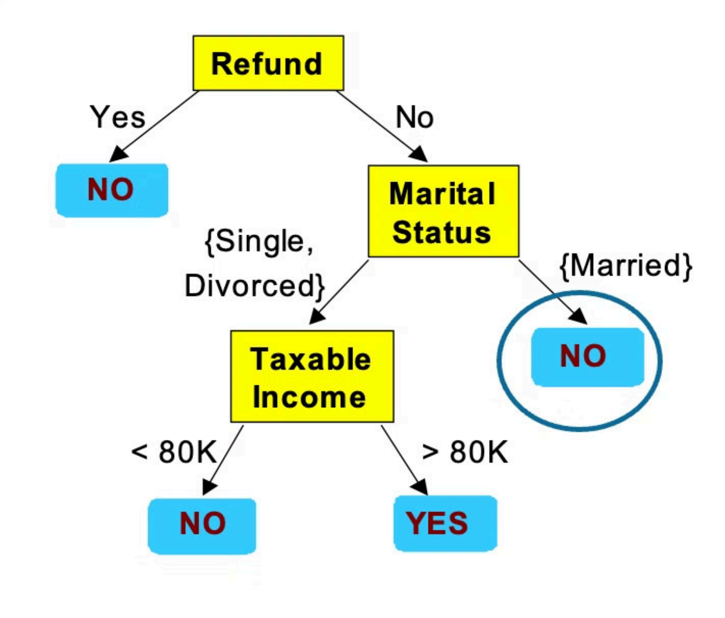
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(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

Effect of Rule Simplification

- Rules are no longer mutually exclusive
    - A record may trigger more than one rule
    - Solution?
        - Ordered rule set
            - Rules are rank ordered according to their priority
            - An ordered rule set is known as a decision list
        - Unordered rule set – use voting schemes
- Rules are no longer exhaustive
    - A record may not trigger any rules
    - Solution?
        - Use a default class

Building Classification Rules

Direct Method:

- Extract rules directly from data
- e.g.: RIPPER, CN2, Holte's 1R

Indirect Method:

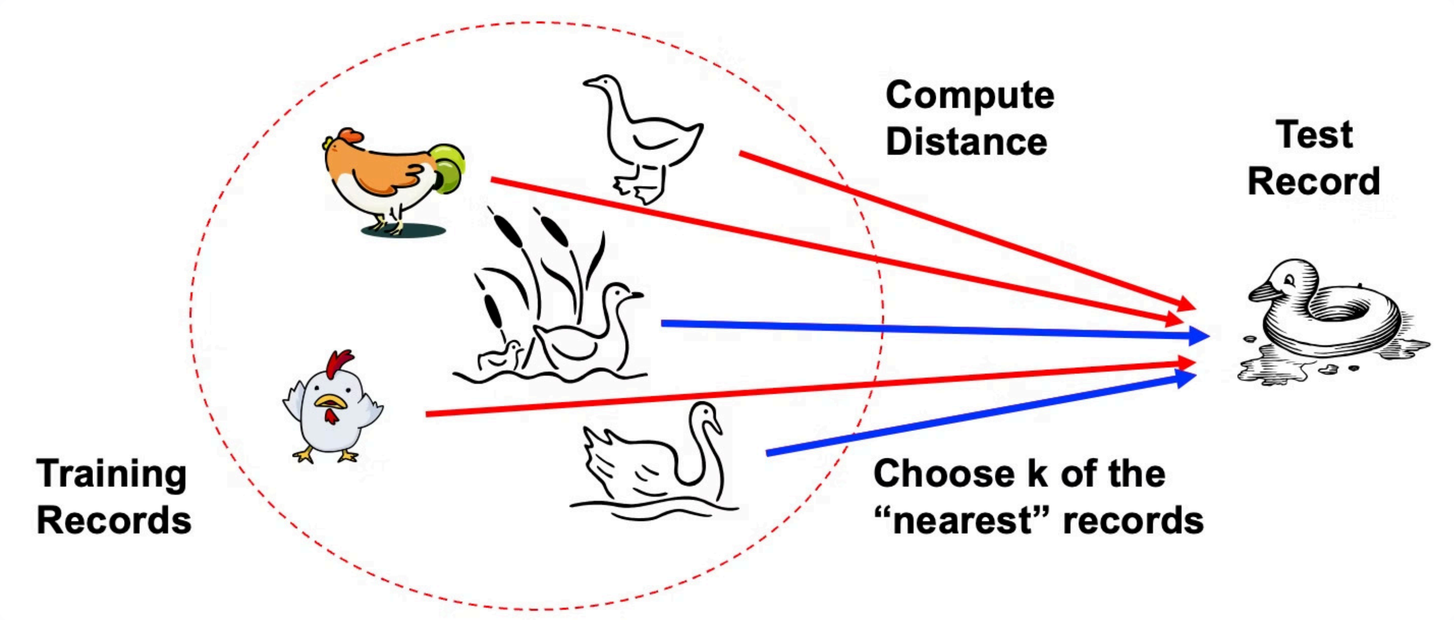
- Extract rules from other classification models (e.g. decision trees, neural networks, etc).
- e.g: C4.5rules

Advantages of Rule-Based Classifiers

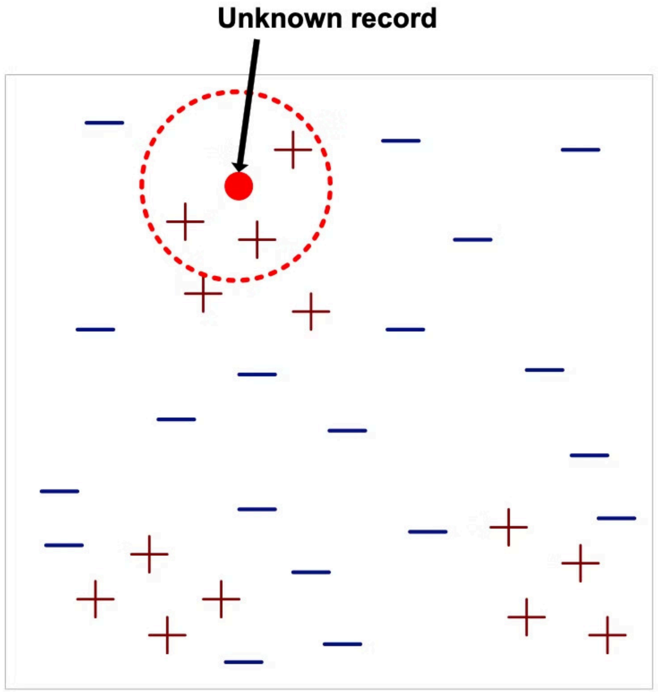
- As highly expressive as decision trees
- Easy to interpret
- Easy to generate
- Can classify new instances rapidly
- Performance comparable to decision trees

Nearest Neighbor Classifiers

- Basic idea:
- If it walks like a duck, quacks like a duck, then it's probably a duck
- Choose k of the "nearest" records

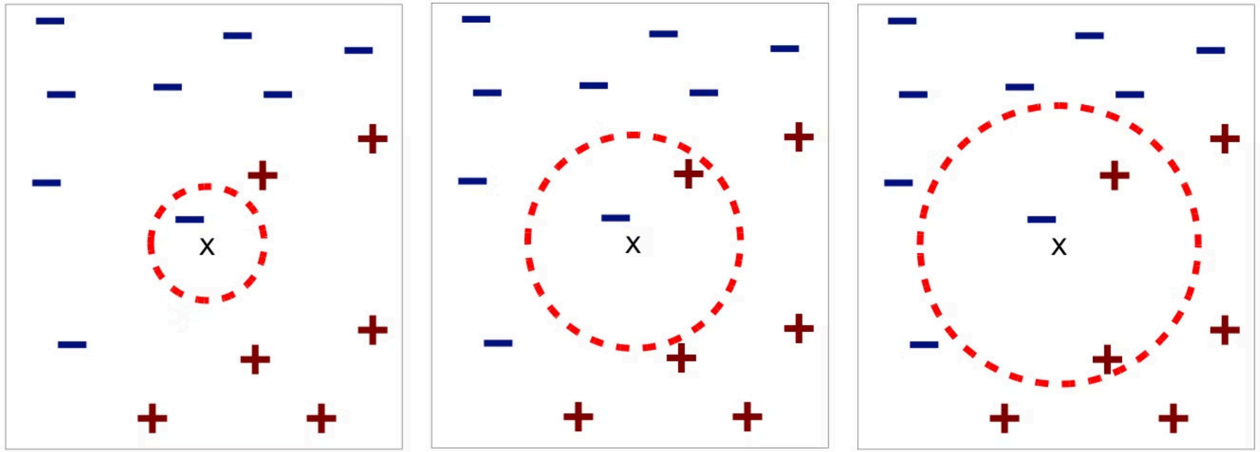


Nearest-Neighbor Classifiers



- 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

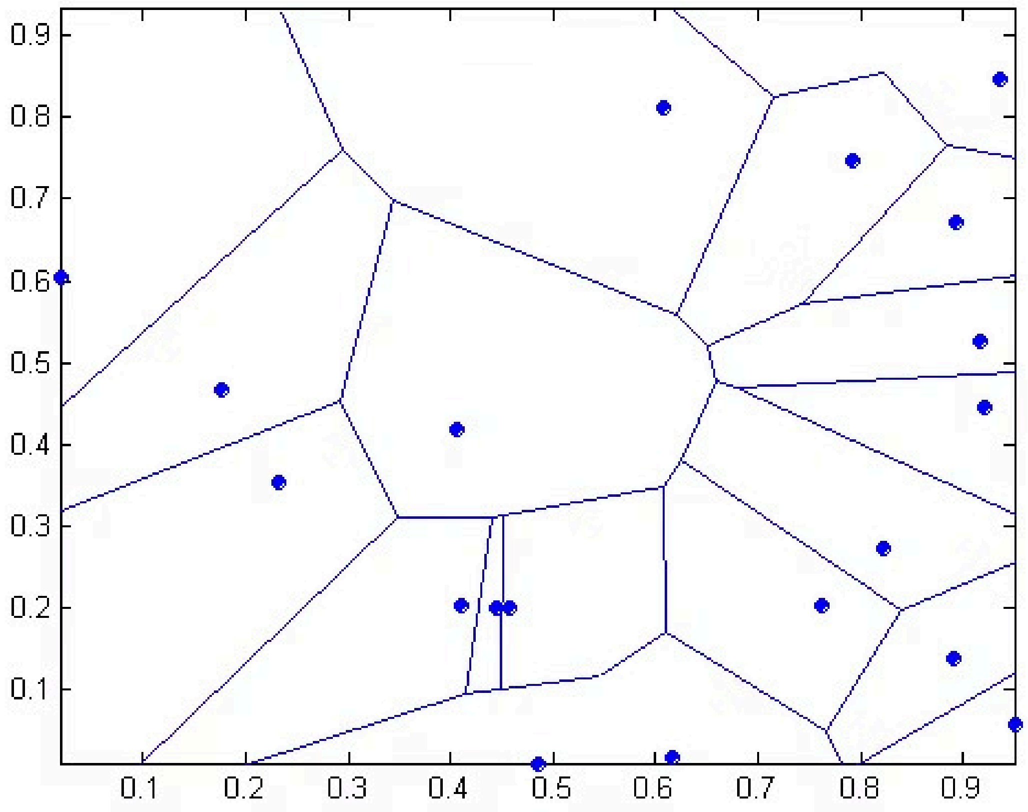


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

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

1 nearest-neighbor

Voronoi Diagram



Nearest Neighbor Classification

- Compute distance between two points
  - Euclidean distance

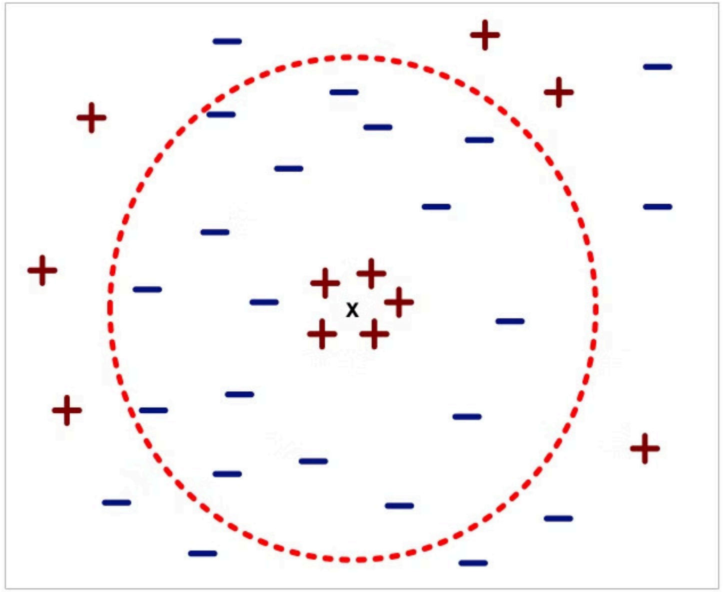
$$d(p,q) = \sqrt{\sum(pi - qi)2}$$

- 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

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



Nearest Neighbor Classification…

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example:
  - height of a person may vary from 1.5m to 1.8m
  - weight of a person may vary from 90lb to 300lb
  - income of a person may vary from $10K to $1M

Nearest Neighbor Classification…

- Problem with Euclidean measure:
- High dimensional data
- **curse of dimensionality**
- Can produce counter-intuitive results

1 1 1 1 1 1 1 1 1 1 1 0	**1 0 0 0 0 0 0 0 0 0 0 0**
0 1 1 1 1 1 1 1 1 1 1 1	**0 0 0 0 0 0 0 0 0 0 0 1**
d = 1.4142	**d = 1.4142**

◆ Solution: Normalize the vectors to unit length

Nearest neighbor Classification...

- k-NN classifiers are lazy learners
  - It does not build models explicitly
  - Unlike eager learners such as decision tree induction and rule-based systems
  - Classifying unknown records are relatively expensive

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:
  - $P(C|A) = P(A,C)/P(A)$
  - $P(A|C) = P(A,C)/P(C)$
- Bayes theorem:
  - $P(C|A) = P(A|C)P(C)/P(A)$

Example of Bayes Theorem

Given:

- A doctor knows that meningitis causes stiff neck 50% of the time
- Prior probability of any patient having meningitis is 1/50,000
- Prior probability of any patient having stiff neck is 1/20

If a patient has stiff neck, what's the probability he/she has meningitis?

$P(M|S) = P(S|M)P(M)/P(S) = 0.5\times(1/50000) / (1/20) = 0.0002$

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

Bayesian Classifiers

- Approach:
  - compute the posterior probability P(C | A1, A2, …, An) 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 | A1, A2, …, An)
  - Equivalent to choosing value of C that maximizes P(A1, A2, …, An|C) P(C)

- How to estimate P(A1, A2, …, An | 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.

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Class: P(C) = Nc/N

e.g., P(No) = 7/10, P(Yes) = 3/10

- For discrete attributes: P(Ai | Ck) = |A_ik|/ Nck

where |A_ik| is number of instances having attribute Ai and belongs to class Ck

Examples:

P(Status=Married|No) = 4/7

P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - **Discretize** the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - **Two-way split:** (A < v) or (A > v)
    - choose only one of the two splits as new attribute
  - **Probability density estimation:**
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability P(Ai|c)

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

- One for each (A_i, c_i) pair

For (Income, Class=No):

- If Class=No
  - sample mean = 110
  - sample variance = 2975

P(Income =120| No) = 1/√2π(54.54) e^(-(120-110)²/2(2975)) =0.0072

Example of Naïve Bayes Classifier

Given a Test Record:

X =(Refund =No, Married, Income =120K)

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) × P(Married| Class=No) × P(Income=120K| Class=No) = 4/7 × 4/7 × 0.0072 = 0.0024
- P(X|Class=Yes) = P(Refund=No| Class=Yes) × P(Married| Class=Yes) × P(Income=120K| Class=Yes) = 1 × 0 × 1.2 × 10^{-9} = 0

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

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

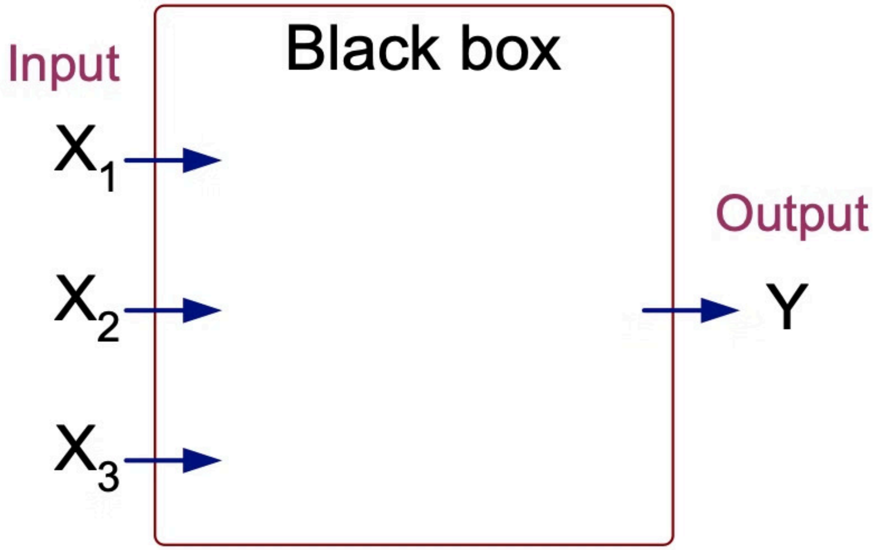
=> Class = No

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)

Artificial Neural Networks (ANN)

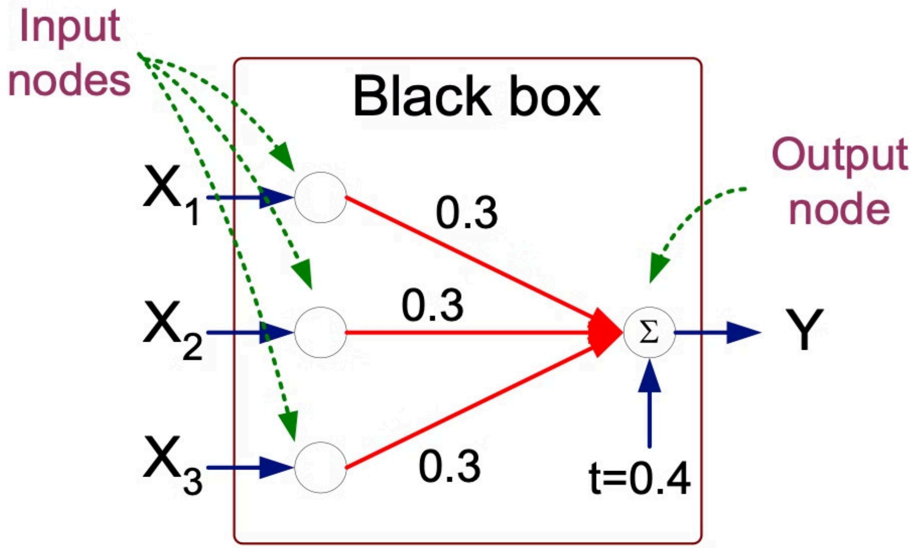
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



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

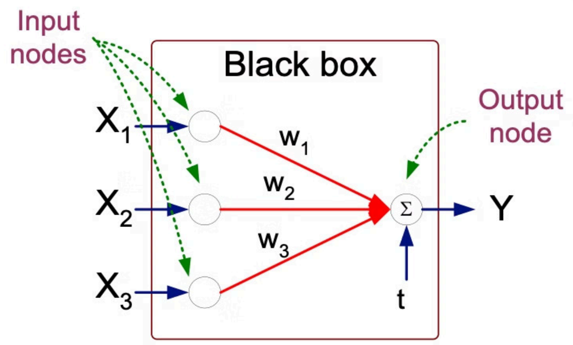


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

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

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

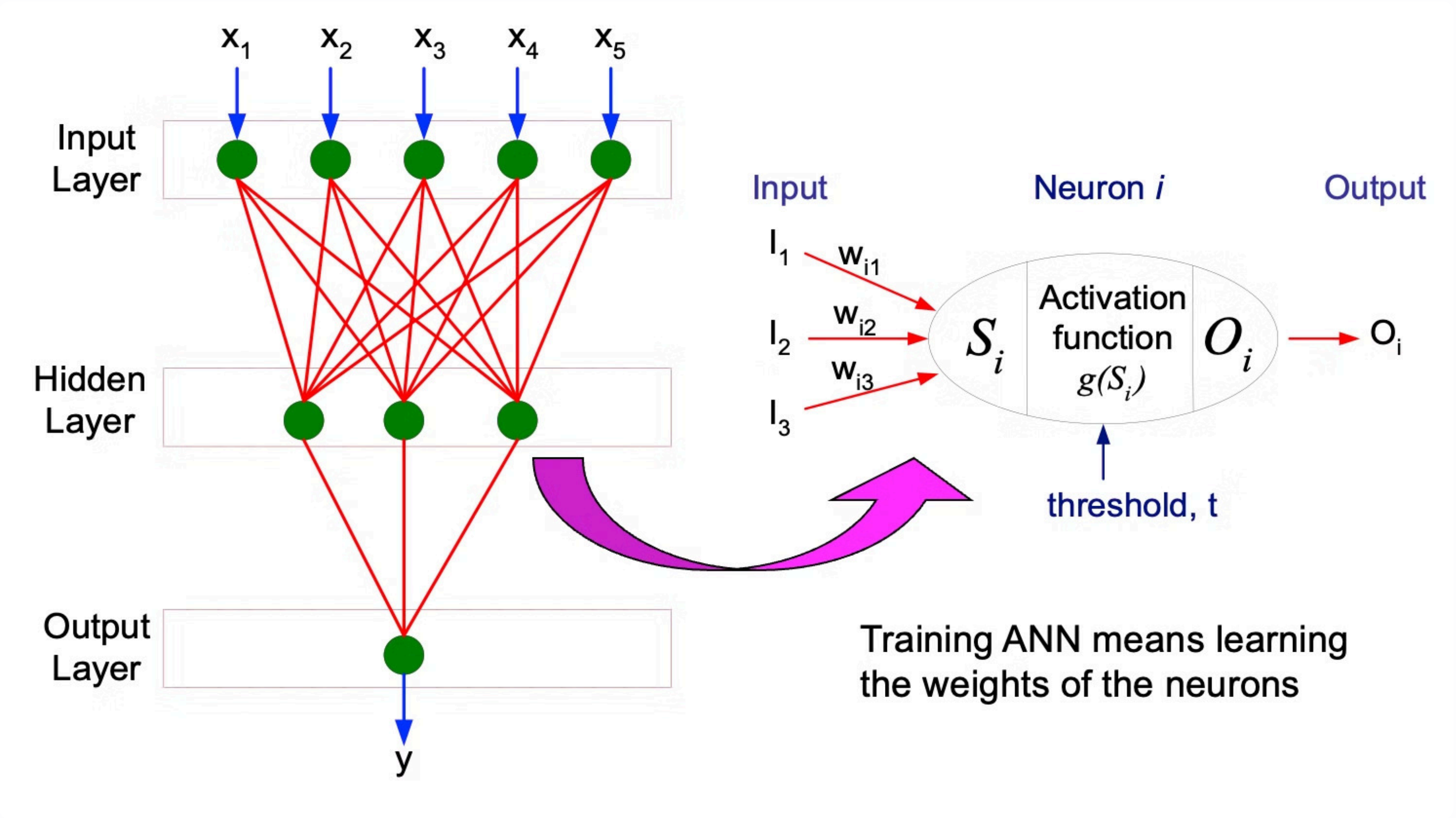


Perceptron Model

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

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

General Structure of ANN



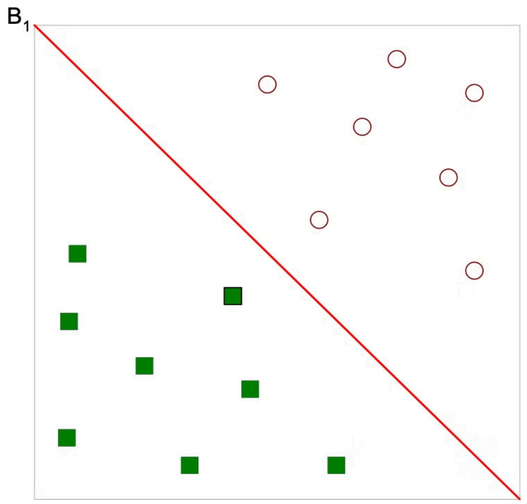
Algorithm for learning ANN

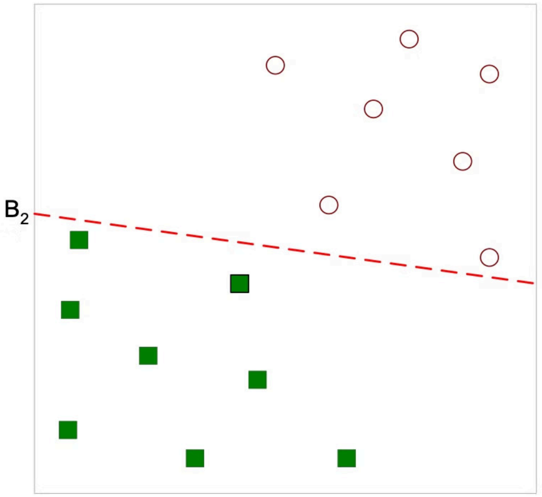
- Initialize the weights (w0, w1, …, wk)
- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples
- Objective function: E = ∑[Yi - f(wi,Xi)]2
- Find the weights wi's that minimize the above objective function
- e.g., backpropagation algorithm

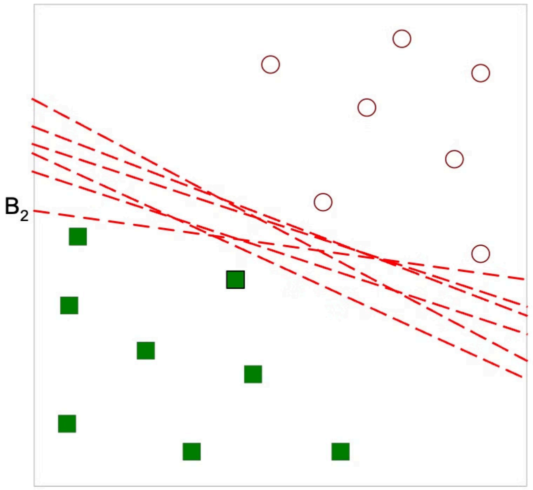
Support Vector Machines

Find a linear hyperplane (decision boundary) that will separate the data

- Which one is better? B1 or B2?
- How do you define better?

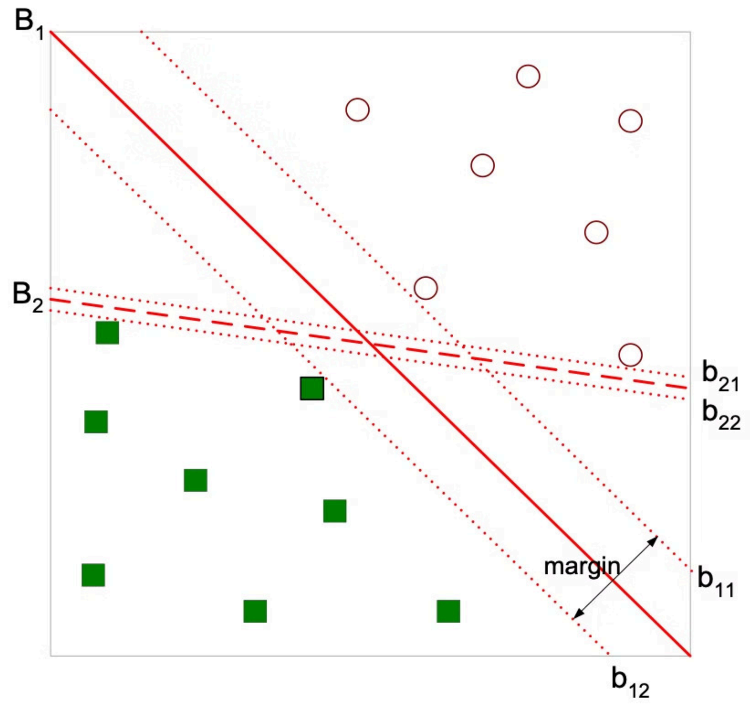




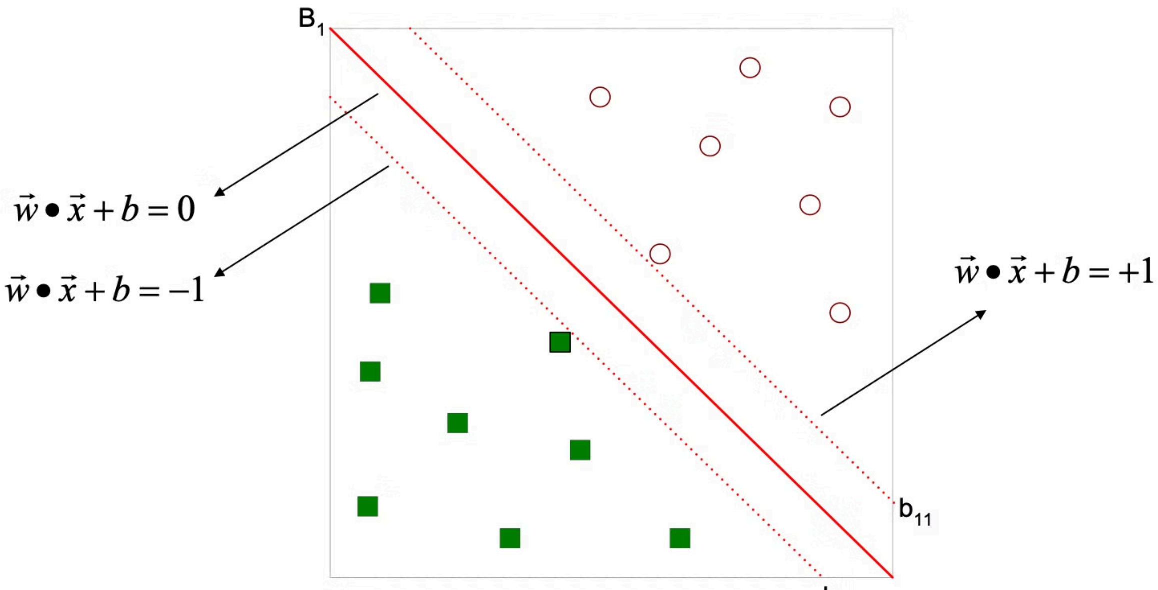


Support Vector Machines

Find hyperplane **maximizes** the margin => B1 is better than B2



Support Vector Machines



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$

Support Vector Machines

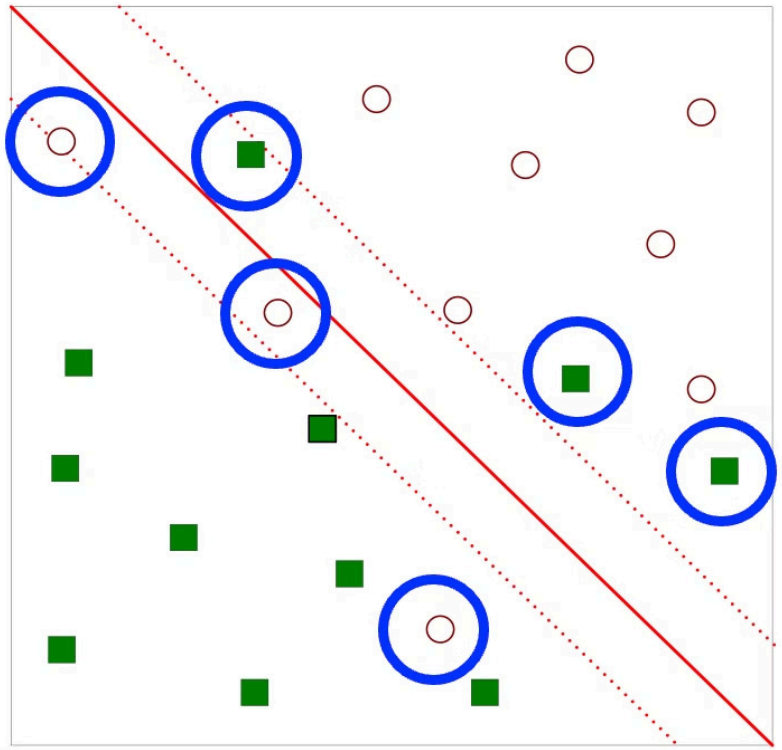
- We want to maximize: Margin = $2/||w||^2$
- Which is equivalent to minimizing: $L(w) = ||w||^2/2$
- But subjected to the following constraints:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$$

- This is a constrained optimization problem
    - Numerical approaches to solve it (e.g., quadratic programming)

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?
- Introduce slack variables
- Need to minimize:

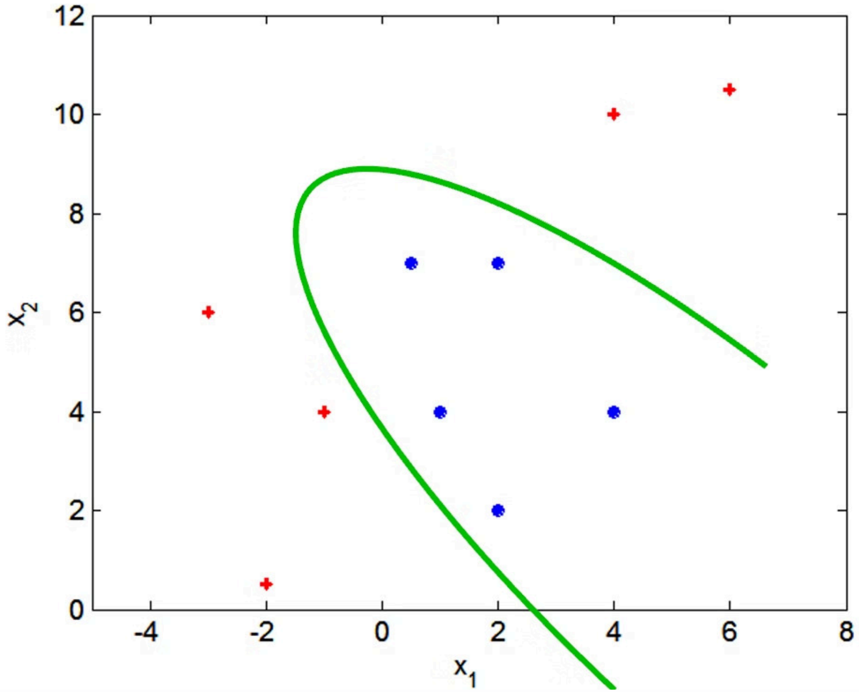
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k\right)$$

- Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

Nonlinear Support Vector Machines

- What if decision boundary is not linear?



Nonlinear Support Vector Machines

- Transform data into higher dimensional space