# Data Mining

## [07_Classification(Part.4)]


Hilmy A.Tawakal

[hilmi.tawakal@gmail.com]

# Data Mining
# Classification: Basic Concepts, Decision Trees, and Model Evaluation

Lecture Notes for Chapter 4

Introduction to Data Mining

by

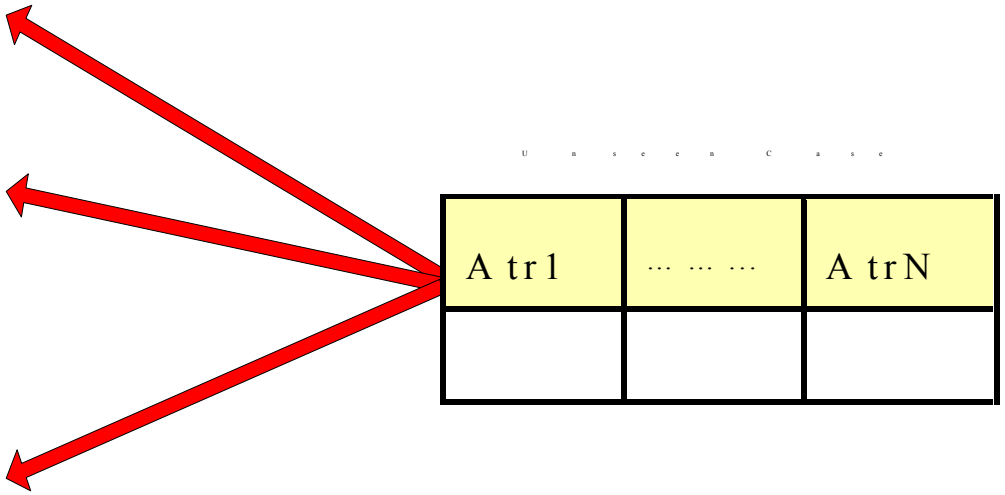
Tan, Steinbach, Kumar

# Instance-Based Classifiers

- Store the training records
- Use training records to predict the class label of unseen cases

Set of Stored Cases

| Atr1 | ... ... ... | AtrN | Class |
|---|---|---|---|
| | | | A |
| | | | B |
| | | | B |
| | | | C |
| | | | A |
| | | | C |
| | | | B |



# Instance Based Classifiers

Examples:

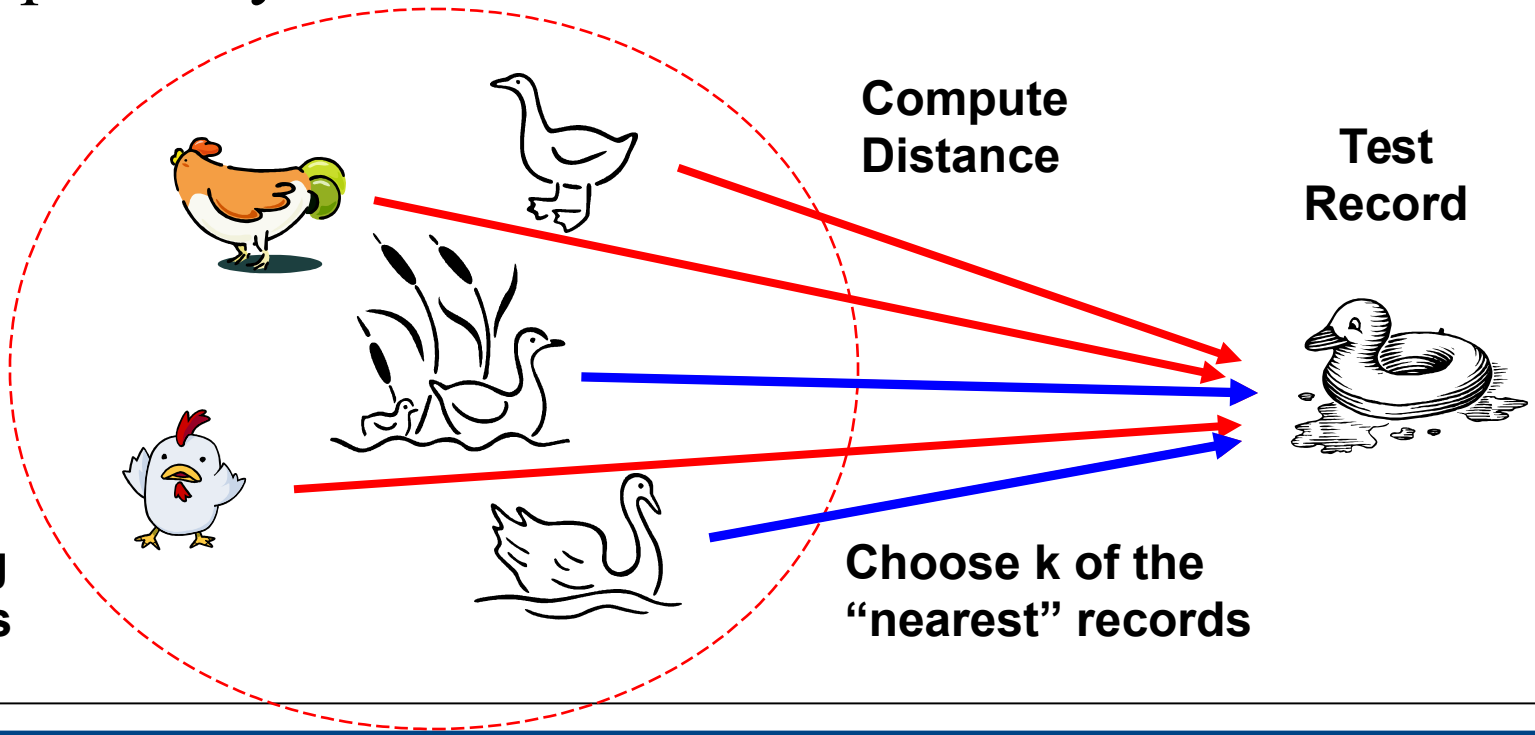
- Rote-learner
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly

- Nearest neighbor
  - Uses k “closest” points (nearest neighbors) for performing classification

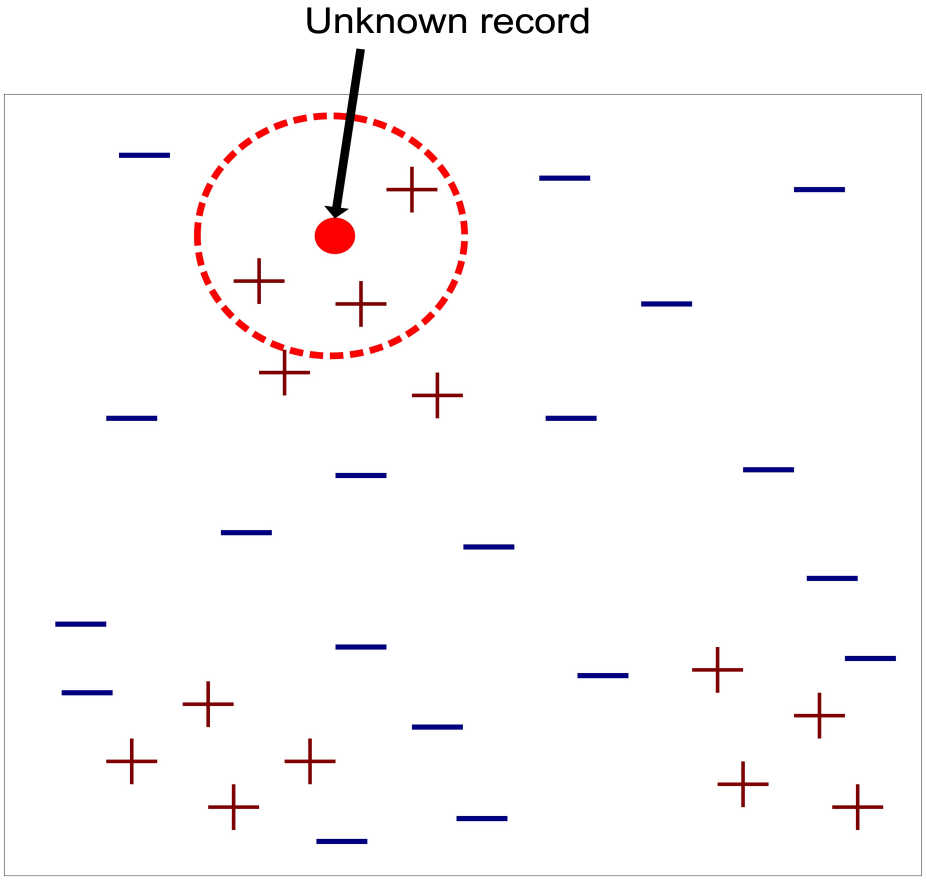
# Nearest Neighbor Classifiers

Basic idea:

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



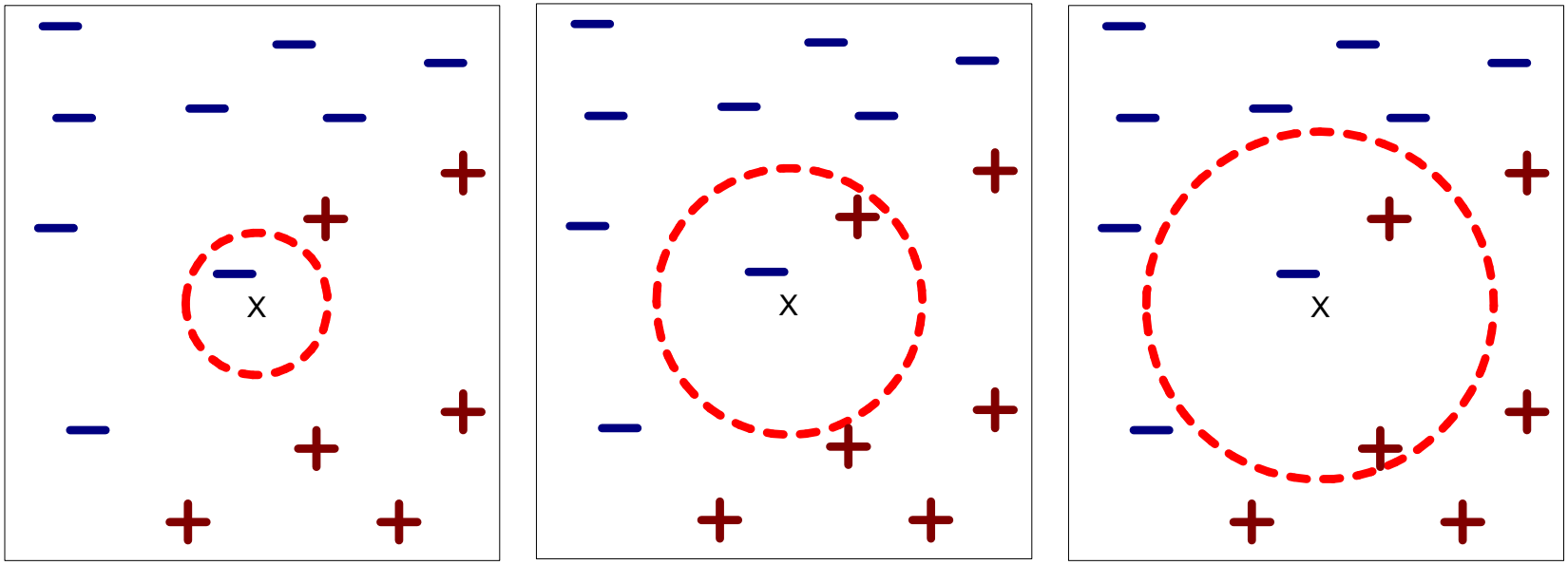
# Nearest-Neighbor Classifiers




- 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

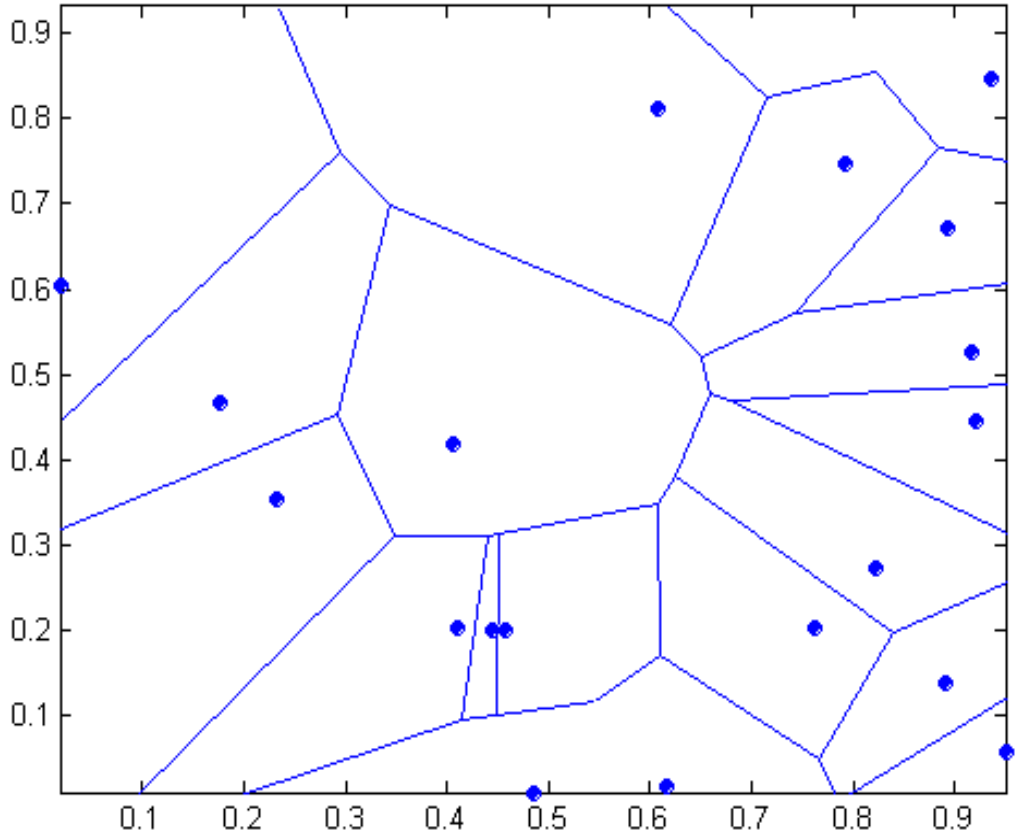



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

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

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

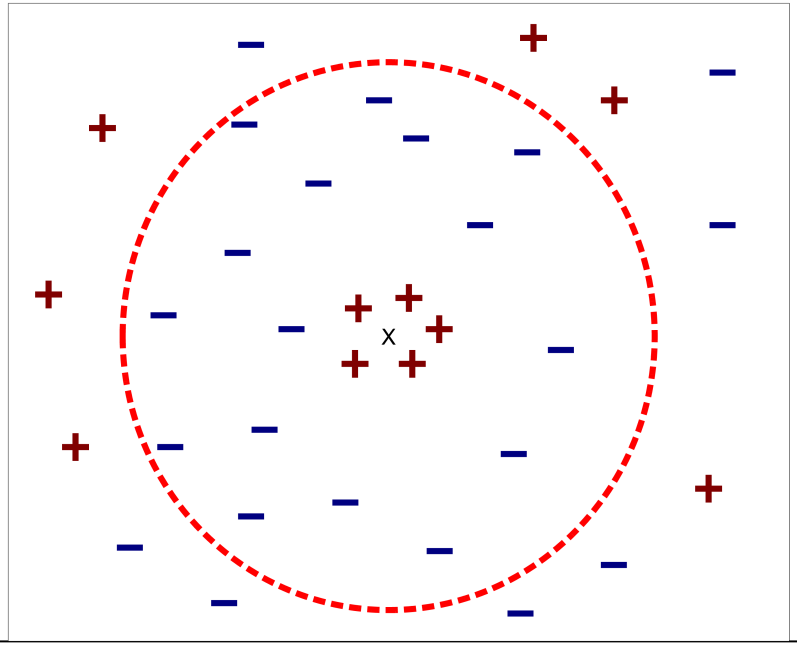
Determine the class from nearest neighbor list

- 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



# 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 | vs | 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

# Example: PEBLS

PEBLS: Parallel Examplar-Based Learning System (Cost & Salzberg)

- Works with both continuous and nominal features
  - For nominal features, distance between two nominal values is computed using modified value difference metric (MVDM)
- Each record is assigned a weight factor
- Number of nearest neighbor, k = 1

# Example: PEBLS

| 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 |

Distance between nominal attribute values:

d(Single,Married)

= | 2/4 – 0/4 | + | 2/4 – 4/4 | = 1

d(Single,Divorced)

= | 2/4 – 1/2 | + | 2/4 – 1/2 | = 0

d(Married,Divorced)

= | 0/4 – 1/2 | + | 4/4 – 1/2 | = 1

d(Refund=Yes,Refund=No)

= | 0/3 – 3/7 | + | 3/3 – 4/7 | = 6/7

| Class | Marital Status | | |
|---|---|---|---|
| | Single | Married | Divorced |
| Yes | 2 | 0 | 1 |
| No | 2 | 4 | 1 |

| Class | Refund | |
|---|---|---|
| | Yes | No |
| Yes | 0 | 3 |
| No | 3 | 4 |

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

# Example: PEBLS

| *Tid* | Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|---|
| X | Yes | Single | 125K | **No** |
| Y | No | Married | 100K | **No** |

Distance between record X and record Y:

$$\Delta(X,Y) = w_X w_Y \sum_{i=1}^{d} d(X_i, Y_i)^2$$

where:

$$w_X = \frac{\text{Number of times X is used for prediction}}{\text{Number of times X predicts correctly}}$$

$w_X \cong 1$ if X makes accurate prediction most of the time

$w_X > 1$ if X is not reliable for making predictions

# Bayes Classifier

A probabilistic framework for solving classification problems

Conditional Probability: $$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

# 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?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

# 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 $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

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

- 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 $P(C_j) \prod P(A_i | C_j)$ 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: $P(C) = N_c/N$

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

For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_c$$ k

- where $|A_{ik}|$ is number of instances having attribute $A_i$ and belongs to class $C_k$
- 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 $P(A_i|c)$

k

# 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

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

# 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)
  × 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 Classifier

If one of the conditional probability is zero, then the entire expression becomes zero

Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter

# 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**

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

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

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

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

| 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

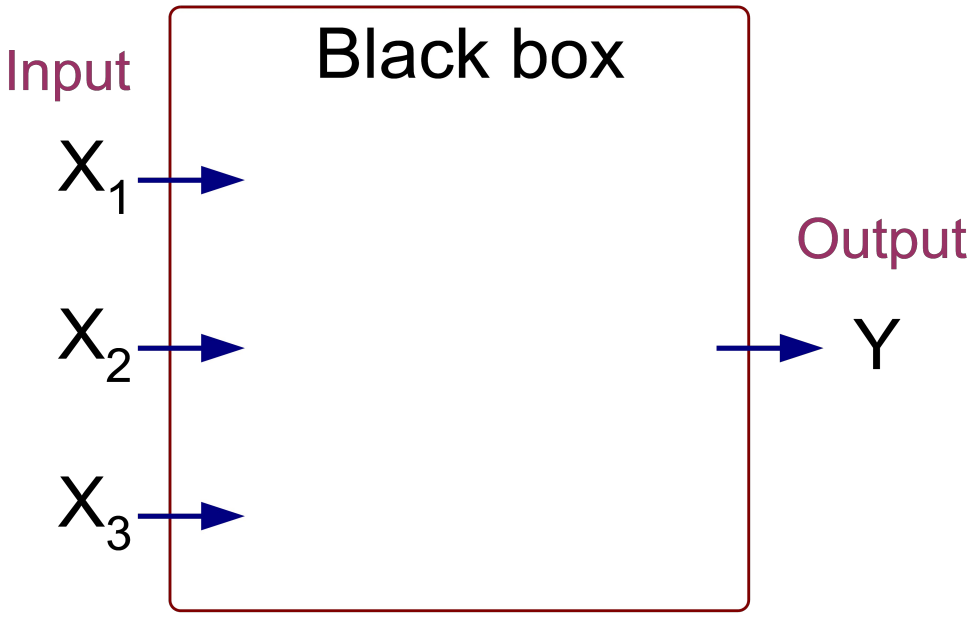
Handle missing values by ignoring the instance during probability estimate calculations

Robust to irrelevant attributes

Independence assumption may not hold for some attributes

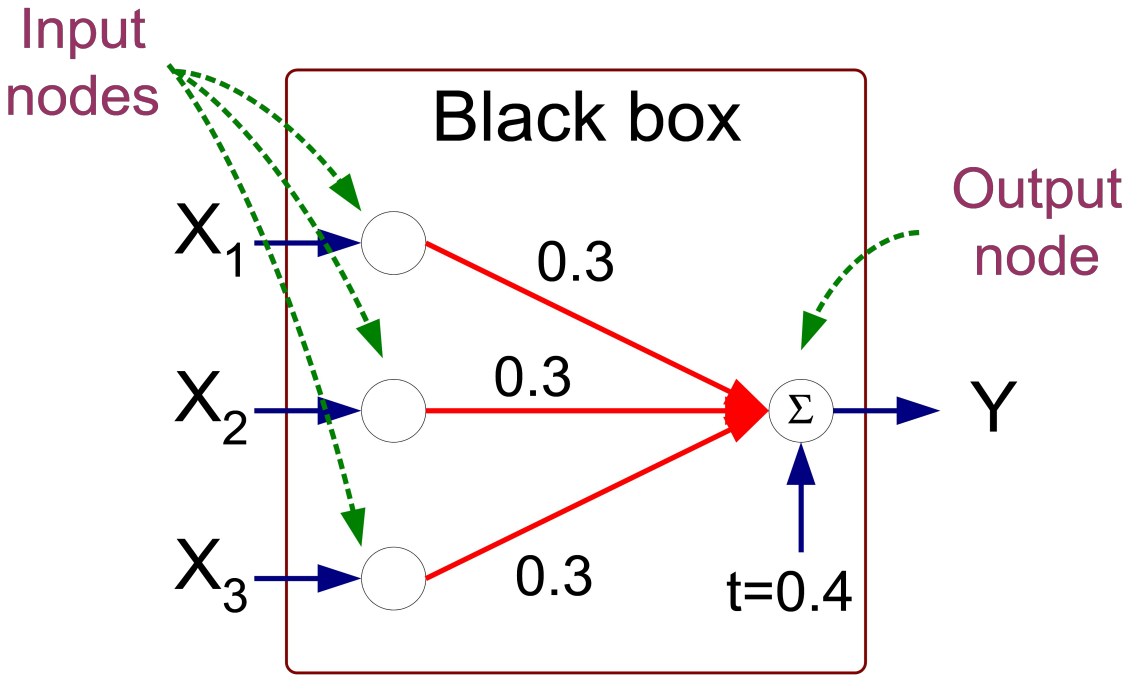
- Use other techniques such as Bayesian Belief Networks (BBN)

# Artificial Neural Networks (ANN)




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

# Artificial Neural Networks (ANN)




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

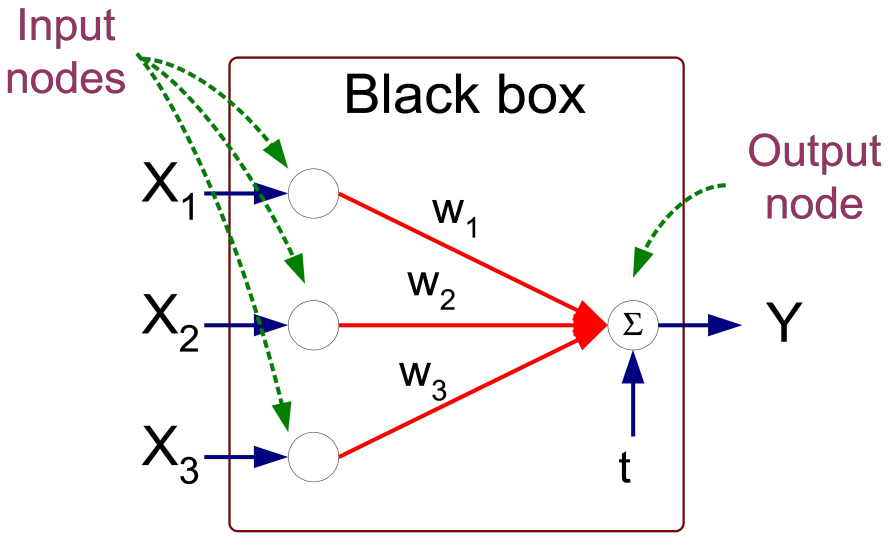
$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

# 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

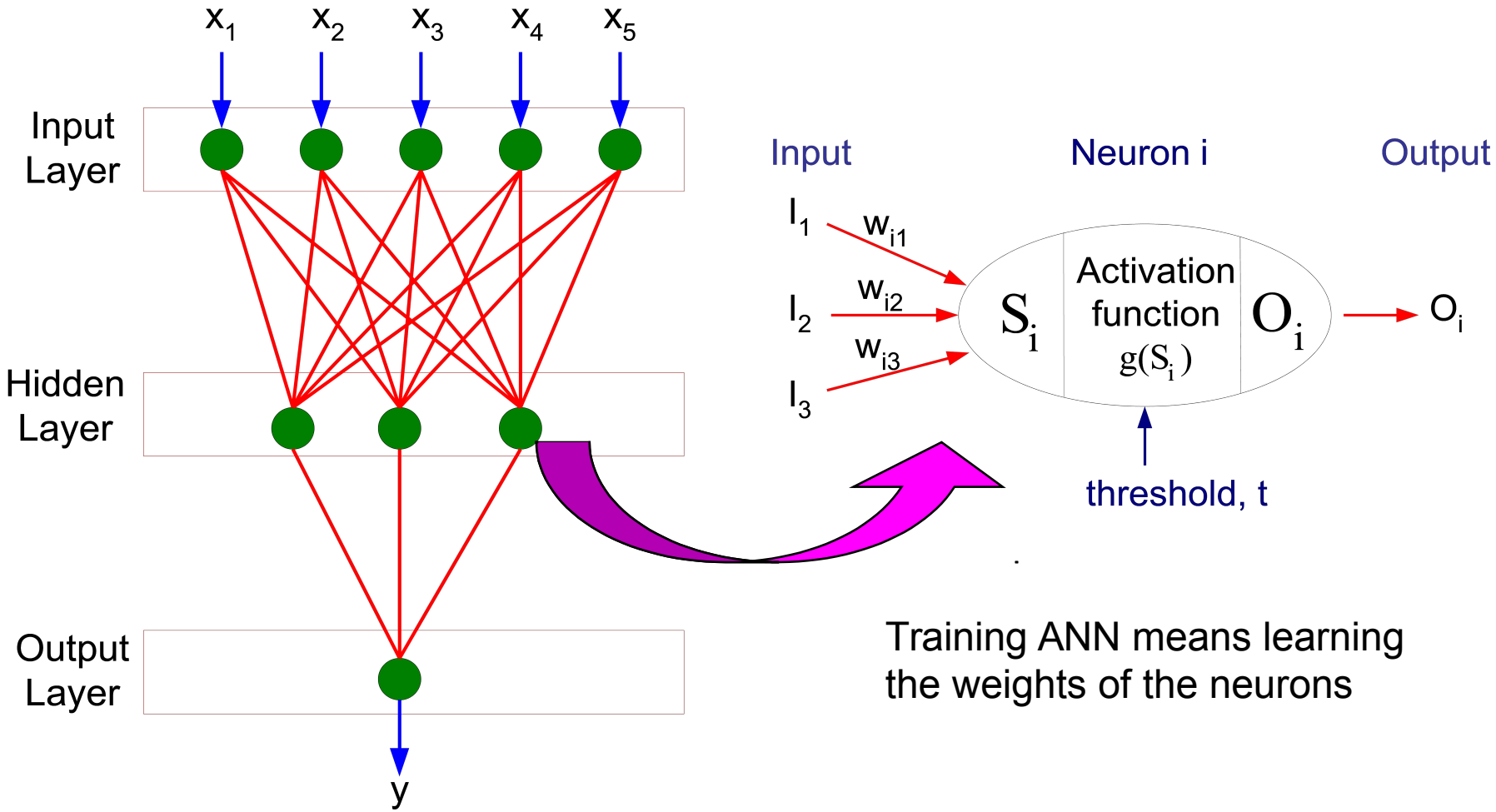



**Perceptron Model**

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

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

# General Structure of ANN




Training ANN means learning the weights of the neurons

# Algorithm for learning ANN

Initialize the weights ($w_0$, $w_1$, …, $w_k$)

Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples

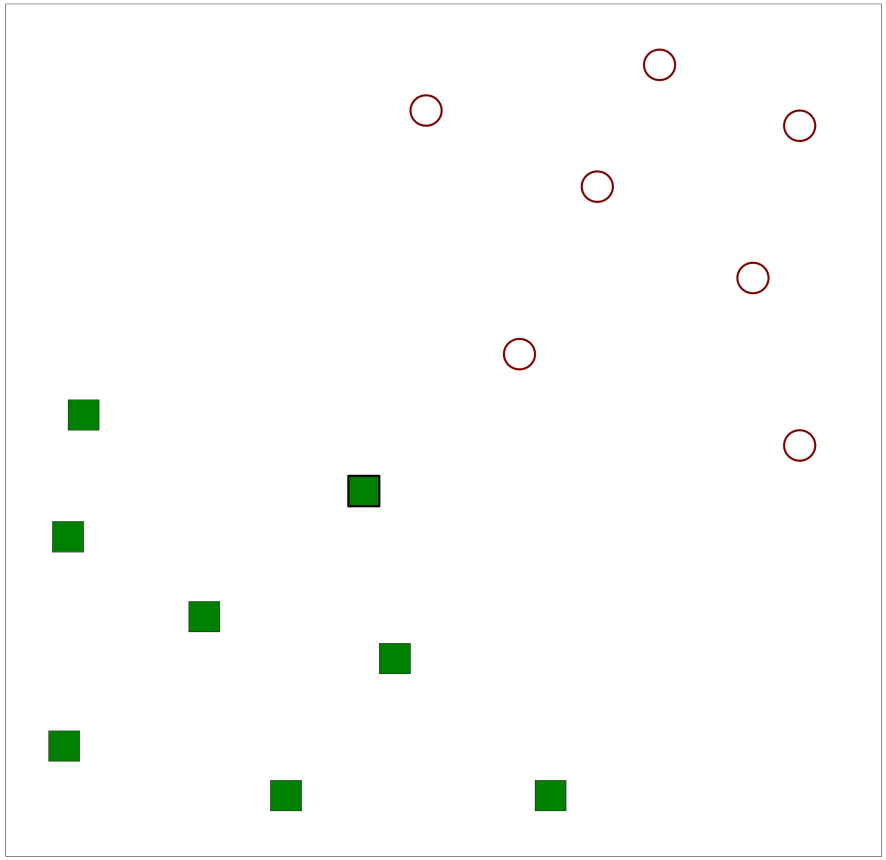
- Objective function:

$$E = \sum_i [Y_i - f(w_i, X_i)]^2$$

- Find the weights $w_i$'s that minimize the above objective function
  - e.g., backpropagation algorithm (see lecture notes)

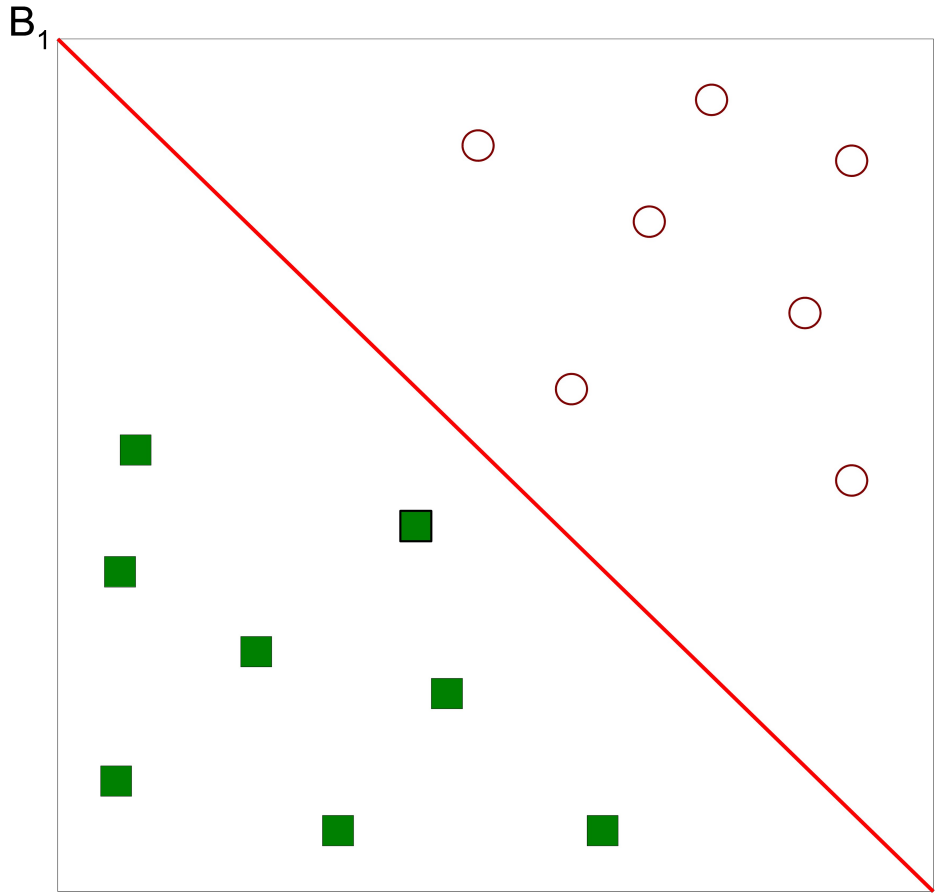
# Support Vector Machines

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



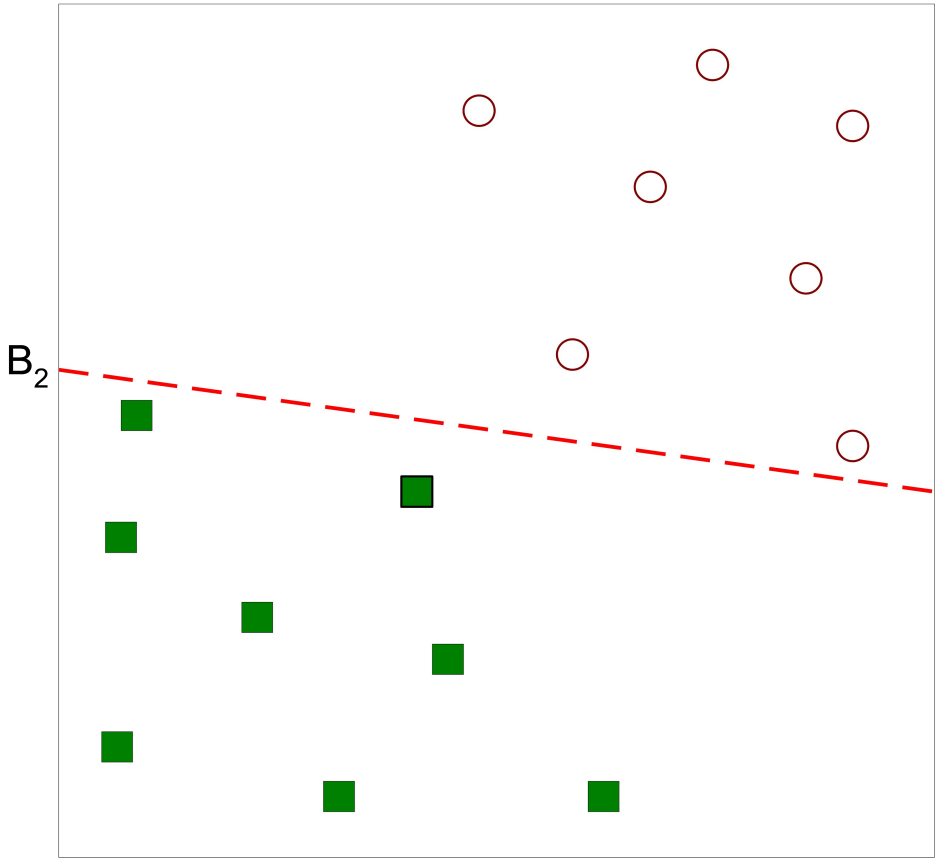
# Support Vector Machines

One Possible Solution



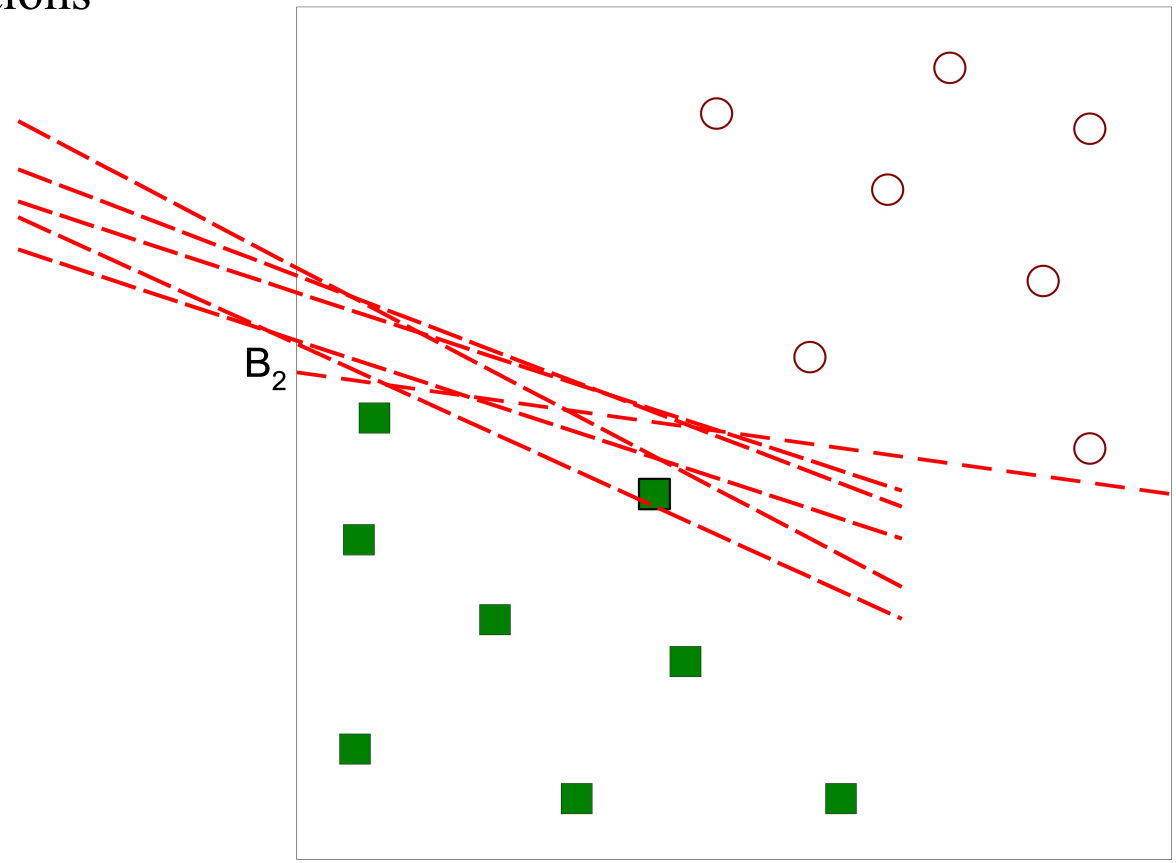
# Support Vector Machines

Another possible solution



# Support Vector Machines

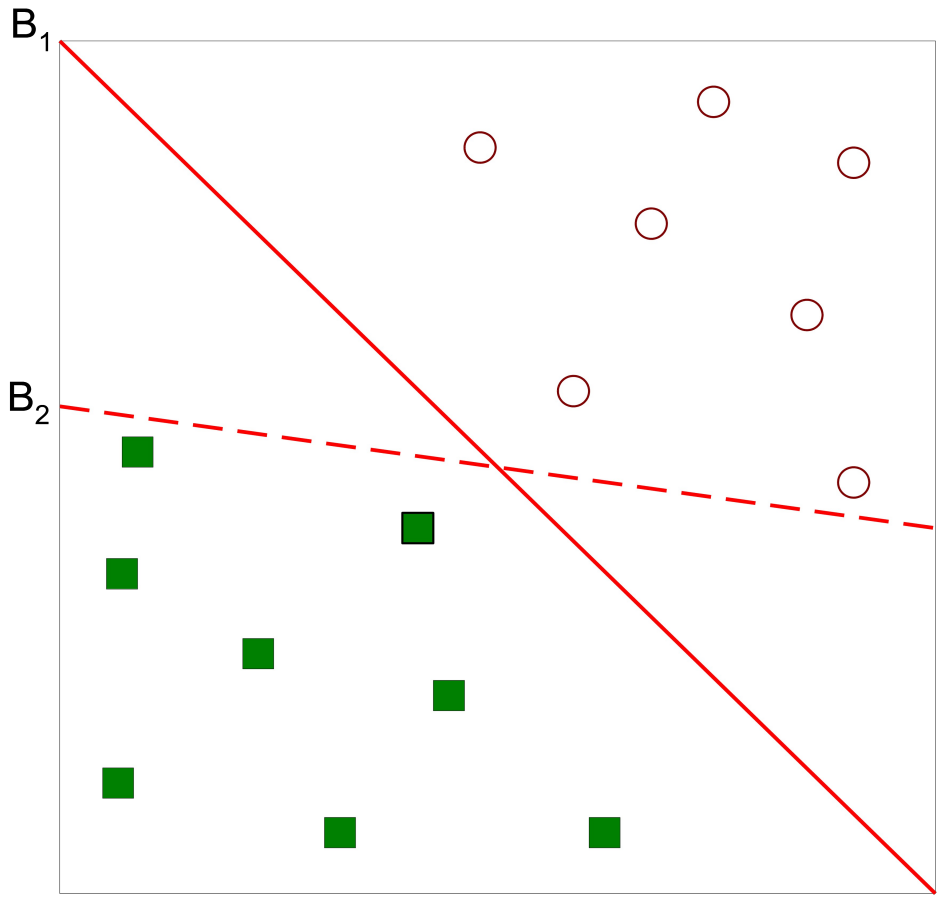
Other possible solutions



# Support Vector Machines

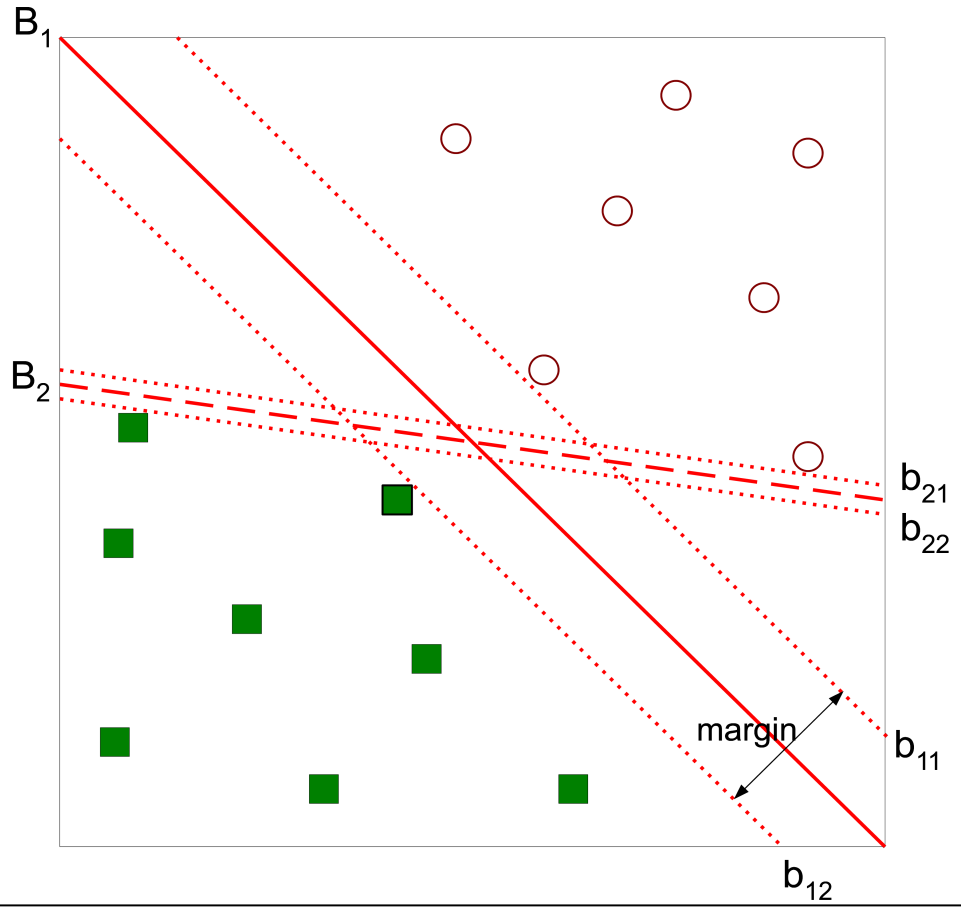
Which one is better? B1 or B2?

How do you define better?

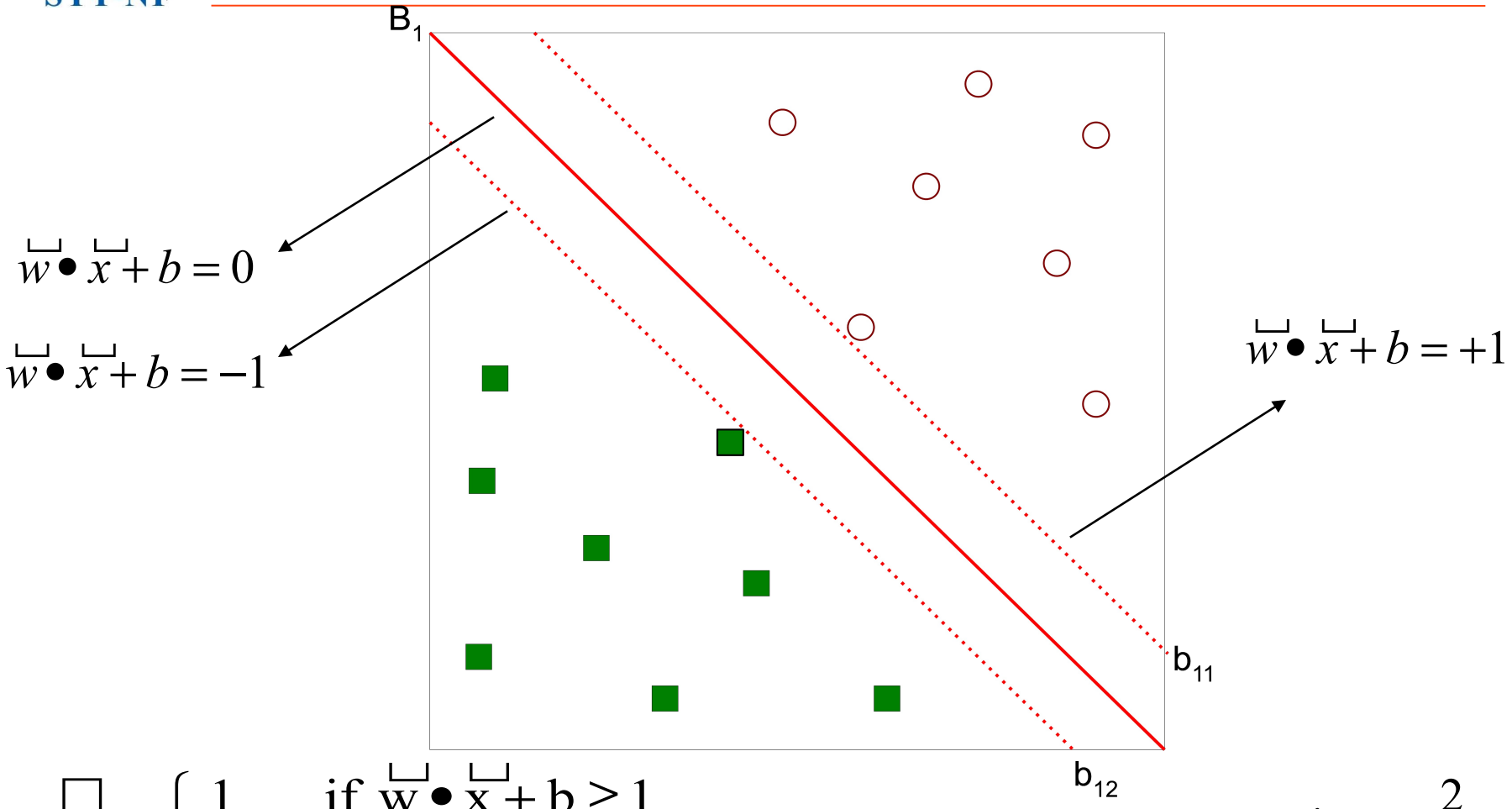


# Support Vector Machines

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:

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

- Which is equivalent to minimizing:

$$L(w) = \frac{\| w \|^2}{2}$$

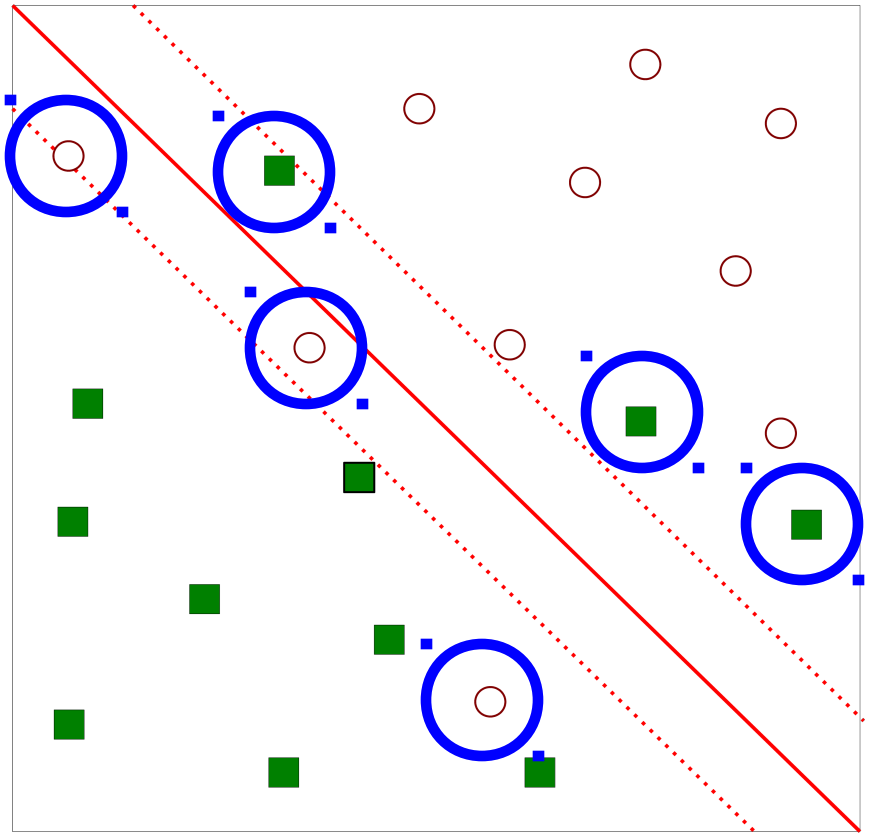
- But subjected to the following constraints:

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

  - 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

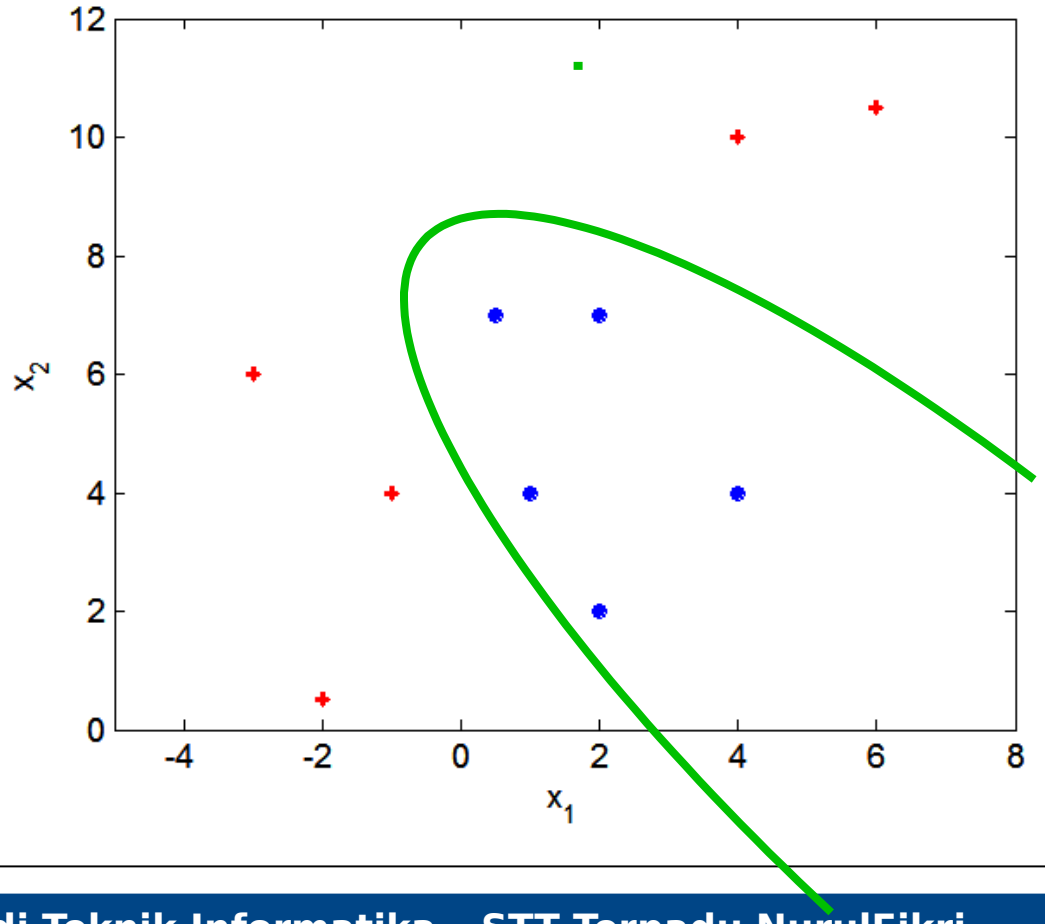
What if the problem is not linearly separable?

- 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:

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

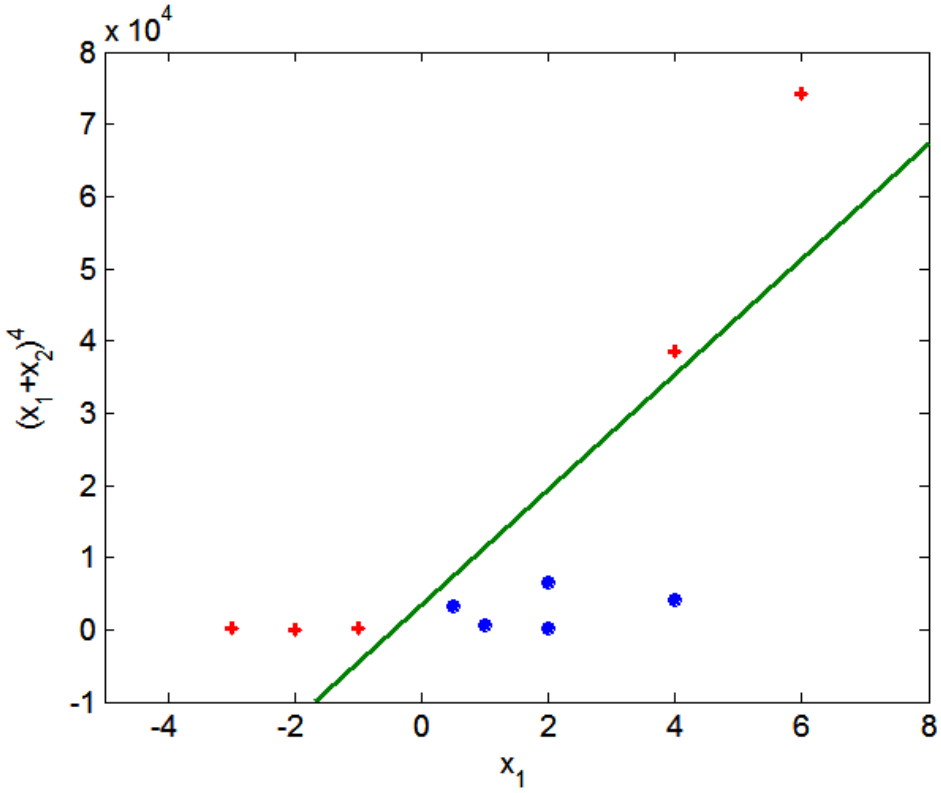
# Nonlinear Support Vector Machines

What if decision boundary is not linear?



# Nonlinear Support Vector Machines

Transform data into higher dimensional space

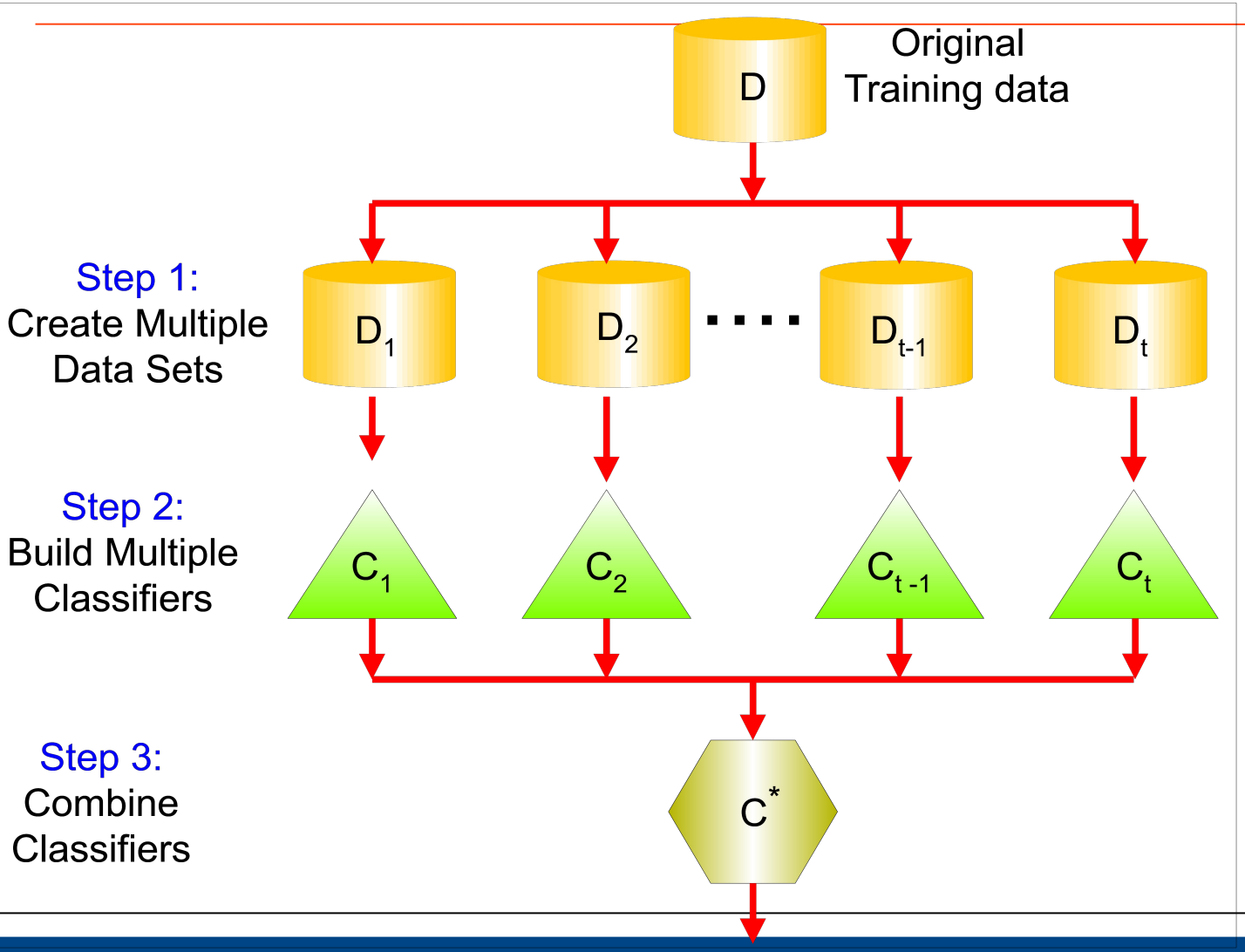


# Ensemble Methods

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:

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

# Examples of Ensemble Methods

How to generate an ensemble of classifiers?

- Bagging
- Boosting

# Bagging

Sampling with replacement

| Original Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bagging (Round 1) | 7 | 8 | 10 | 8 | 2 | 5 | 10 | 10 | 5 | 9 |
| Bagging (Round 2) | 1 | 4 | 9 | 1 | 2 | 3 | 2 | 7 | 3 | 2 |
| Bagging (Round 3) | 1 | 8 | 5 | 10 | 5 | 5 | 9 | 6 | 3 | 7 |

Build classifier on each bootstrap sample

Each sample has probability $(1 - 1/n)^n$ of being selected

# Boosting

An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records

- Initially, all N records are assigned equal weights
- Unlike bagging, weights may change at the end of boosting round

# Boosting

Records that are wrongly classified will have their weights increased

Records that are classified correctly will have their weights decreased

| Original Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Boosting (Round 1) | 7 | 3 | 2 | 8 | 7 | 9 | 4 | 10 | 6 | 3 |
| Boosting (Round 2) | 5 | 4 | 9 | 4 | 2 | 5 | 1 | 7 | 4 | 2 |
| Boosting (Round 3) | 4 | 4 | 8 | 10 | 4 | 5 | 4 | 6 | 3 | 4 |

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

# Example: AdaBoost

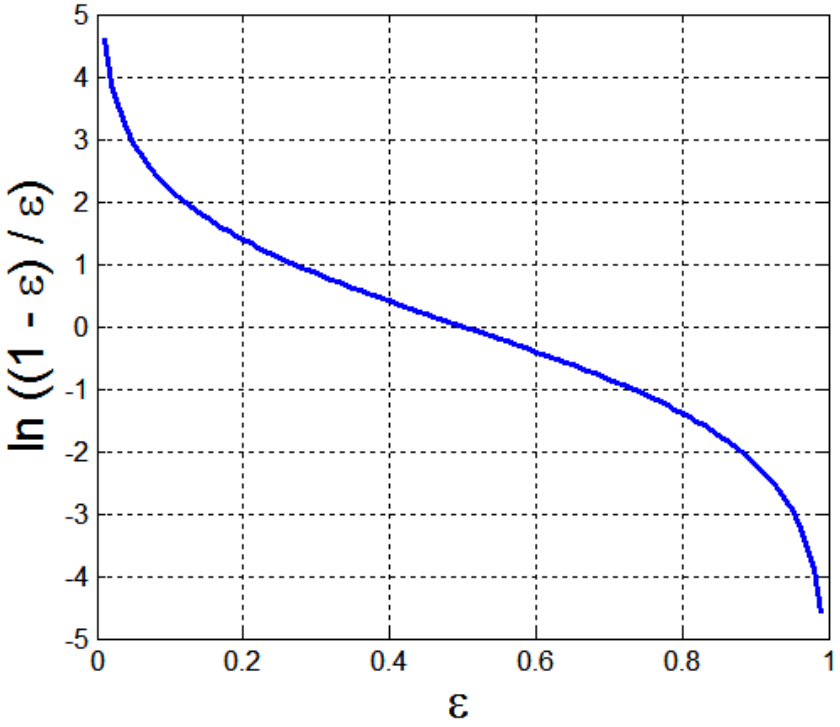
Base classifiers: $C_1, C_2, \ldots, C_T$

Error rate:

$$\varepsilon_i = \frac{1}{N}\sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

Importance of a classifier:

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$



# Example: AdaBoost

Weight update:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

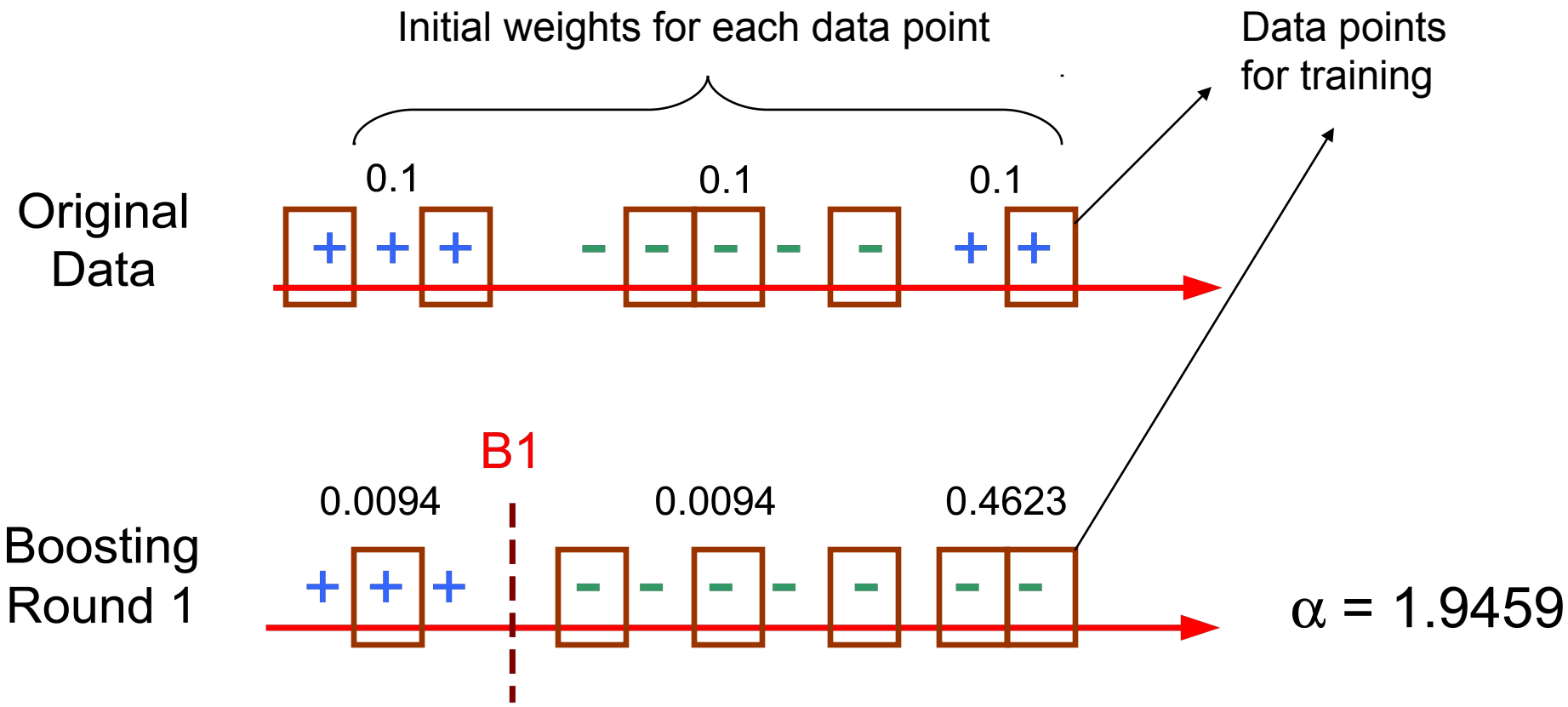
where $Z_j$ is the normalization factor

If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated

Classification:

$$C^*(x) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta\left(C_j(x) = y\right)$$

# Illustrating AdaBoost



# Illustrating AdaBoost