数据挖掘
第4-3章 分类-贝叶斯分类器

教师：王东京
学院：计算机学院
邮箱：dongjing.wang@hdu.edu.cn

贝叶斯分类器 Bayes Classifier

- 解决分类问题的概率框架
- 条件概率 Conditional Probability:

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)}$$

$$P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$$

- 贝叶斯定理 Bayes theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

贝叶斯分类器 Bayes Classifier

考虑两队之间的足球比赛：队 0 和队 1。假设 65%的比赛队 0 胜出，剩余的比赛队 1 获胜。队 0 获胜的比赛中只有 30%是在队 1 的主场，而队 1 取胜的比赛中 75%是主场获胜。如果下一场比赛在队 1 的主场进行，哪一支球队最有可能胜出呢？

- 贝叶斯定理 Bayes theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

- X代表东道主，Y代表比赛的胜利者。

队 0 取胜的概率是 $P(Y=0)=0.65$，

队 1 取胜的概率是 $P(Y=1)=1-P(Y=0)=0.35$，

队 1 取胜时作为东道主的概率是 $P(X=1|Y=1)=0.75$，

队 0 取胜时队 1 作为东道主的概率是 $P(X=1|Y=0)=0.3$。

- 哪支球队更可能胜出？ $P(Y=1 \mid X=1)$ $P(Y=0 \mid X=1)$

$$P(Y=1 \mid X=1)=0.5738 \qquad P(Y=0 \mid X=1)=0.4262$$

Bayes Theorem 在分类中的应用

- 将属性（attribute）和类别标签（class label）看做随机变量

- 给定记录的属性集(X_1, X_2,…, X_d)
  - 目标是预测类别 Y
  - 我们希望找到能够最大化概率P(Y| X_1, X_2,…, X_d) 的Y

- 是否能够根据数据直接估计P(Y| X_1, X_2,…, X_d)?

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

Example Data

给定一个测试记录:

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

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

- 是否能够直接估计
  P(Evade = Yes | X) 和 P(Evade = No | X)?

  简化：接下来我们将
  用Yes替换Evade =Yes
  用No替换Evade = No

用Bayes Theorem 进行分类

- P(Y|X)以概率的方式捕获变量X和Y之间的关系，是Y的后验概率，P(Y)是Y的先验概率。
  - 使用贝叶斯定理计算后验概率（posterior probability）

$$P(Y \mid X_1 X_2 \ldots X_n) = \frac{P(X_1 X_2 \ldots X_d \mid Y) P(Y)}{P(X_1 X_2 \ldots X_d)}$$

  - 最大后验概率（Maximum a-posteriori）：选择能够最大化下述概率的类别Y
    - $P(Y \mid X_1, X_2, \ldots, X_d)$
  - 等价于选择一个能够最大化下述概率的类别Y
    - $P(X_1, X_2, \ldots, X_d|Y)\ P(Y)$
- 如何计算类条件概率 $P(X_1, X_2, \ldots, X_d \mid Y)$?
  - 朴素贝叶斯和贝叶斯信念网络

朴素贝叶斯分类器 Naïve Bayes Classifier

- 给定类标号Y，朴素贝叶斯分类器在估计类条件概率时假设属性 X_i 之间条件独立：
  - $P(X_1, X_2, \ldots, X_d \mid Y_j) = P(X_1 \mid Y_j)\, P(X_2 \mid Y_j) \ldots P(X_d \mid Y_j)$
  - 现在可以根据数据集中X_i 和 Y_j所有的组合估计$P(X_i \mid Y_j)$
  - 对于一个新的测试数据，如果$P(Y_j) \prod P(X_i \mid Y_j)$是最大的，那么这条数据会分类为$Y_j$

$$P(Y \mid X_1 X_2 \ldots X_n) = \frac{P(X_1 X_2 \ldots X_d \mid Y) P(Y)}{P(X_1 X_2 \ldots X_d)}$$

条件独立Conditional Independence

- **给定Z，X** 条件独立于 **Y** ， 如果下述条件成立:
  - P(**X**|**YZ**) = P(**X**|**Z**)

- Example: Arm length and reading skills
  - Young child has shorter arm length and limited reading skills, compared to adults
  - If age is fixed, no apparent relationship between arm length and reading skills
  - Arm length and reading skills are conditionally independent given age

Naïve Bayes on Example Data

Given a Test Record:

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

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

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

P(X | Yes) =

P(Refund = No | Yes) x

P(Divorced | Yes) x

P(Income = 120K | Yes)

P(X | No) =

P(Refund = No | No) x

P(Divorced | No) x

P(Income = 120K | No)

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

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

- P(y) = fraction of instances of class y
  - e.g., P(No) = 7/10,
    P(Yes) = 3/10
- For categorical attributes（离散属性）:

  $P(X_i = c \mid y) = n_c / n$

  - where $|X_i = c|$ is number of instances having attribute value $X_i = c$ and belonging to class y
  - Examples:

    P(Status=Married|No) = 4/7
    P(Refund=Yes|Yes)=0

Estimate Probabilities from Data

- For continuous attributes(连续属性):
  - Discretization: Partition the range into bins:
    - Replace continuous value with bin value
      - Attribute changed from continuous to ordinal
  - 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, use it to estimate the conditional probability $P(X_i|Y)$

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:

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

  - One for each (X_i,Y_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:

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

Naïve 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) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
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 | No) = P(Refund=No | No)
  × P(Divorced | No)
  × P(Income=120K | No)
  = 4/7 × 1/7 × 0.0072 = 0.0006

- P(X | Yes) = P(Refund=No | Yes)
  × P(Divorced | Yes)
  × P(Income=120K | Yes)
  = 1 × 1/3 × 1.2 × 10^{-9} = 4 × 10^{-10}

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

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

Naïve Bayes Classifier can make decisions with partial information about attributes in the test record

Even in absence of information about any attributes, we can use Apriori Probabilities of Class Variable:

Naïve 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) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
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(Yes) = 3/10

P(No) = 7/10

If we only know that marital status is Divorced, then:

P(Yes | Divorced) = 1/3 x 3/10 / P(Divorced)

P(No | Divorced) = 1/7 x 7/10 / P(Divorced)

If we also know that Refund = No, then

P(Yes | Refund = No, Divorced) = 1 x 1/3 x 3/10 / P(Divorced, Refund = No)

P(No | Refund = No, Divorced) = 4/7 x 1/7 x 7/10 / P(Divorced, Refund = No)

If we also know that Taxable Income = 120, then

P(Yes | Refund = No, Divorced, Income = 120) = 1.2×10^{-9} x 1 x 1/3 x 3/10 / P(Divorced, Refund = No, Income = 120)

P(No | Refund = No, Divorced Income = 120) = 0.0072 x 4/7 x 1/7 x 7/10 / P(Divorced, Refund = No, Income = 120)

Example of Naïve Bayes Classifier

Given a Test Record:

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

Naïve 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) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
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(Yes) = 3/10

P(No) = 7/10

P(Yes | Divorced) = 1/3 x 3/10 / P(Divorced)

P(No | Divorced) = 1/7 x 7/10 / P(Divorced)

P(Yes | Refund = No, Divorced) = 1 x 1/3 x 3/10 / P(Divorced, Refund = No)

P(No | Refund = No, Divorced) = 4/7 x 1/7 x 7/10 / P(Divorced, Refund = No)

Issues with Naïve Bayes Classifier

Naïve 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) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
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(Yes) = 3/10

P(No) = 7/10

P(Yes | Married) = 0 x 3/10 / P(Married)

P(No | Married) = 4/7 x 7/10 / P(Married)

Issues with Naïve Bayes Classifier

Consider the table with Tid = 7 deleted

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
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier:

P(Refund = Yes | No) = 2/6
P(Refund = No | No) = 4/6
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/6
P(Marital Status = Divorced | No) = 0
P(Marital Status = Married | No) = 4/6
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0/3
For Taxable Income:
If class = No: sample mean = 91
sample variance = 685
If class = No: sample mean = 90
sample variance = 25

Given X = (Refund = Yes, Divorced, 120K)

P(X | No) = 2/6 X 0 X 0.0083 = 0

P(X | Yes) = 0 X 1/3 X 1.2 X 10^{-9} = 0

Naïve Bayes will not be able to classify X as Yes or No!

Issues with Naïve Bayes Classifier

- If one of the conditional probabilities is zero, then the entire expression becomes zero
- Need to use other estimates of conditional probabilities than simple fractions
- Probability estimation:

$$\text{original: } P(X_i = c|y) = \frac{n_c}{n}$$

$$\text{Laplace Estimate: } P(X_i = c|y) = \frac{n_c + 1}{n + v}$$

$$\text{m} - \text{estimate: } P(X_i = c|y) = \frac{n_c + mp}{n + m}$$

n: number of training instances belonging to class y

n_c: number of instances with $X_i = c$ and $Y = y$

v: total number of attribute values that X_i can take

p: initial estimate of ($P(X_i = c|y)$ known apriori

m: hyper-parameter for our confidence in p

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

朴素贝叶斯的特征

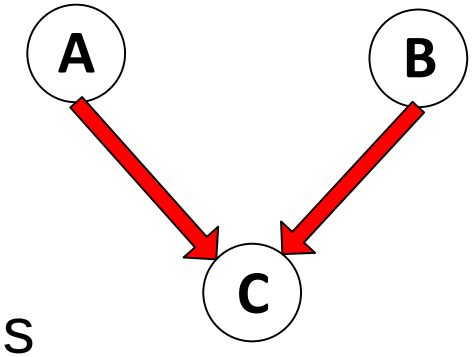
- 面对孤立的噪声点，朴素贝叶斯分类器是健壮的
- 通过在概率估算过程中忽略实例来处理缺失值
- 对不相关的属性不敏感
- 冗余和相关属性会违反类条件（条件独立）假设
  - 使用其他技术，例如贝叶斯信念网络（BBN）

贝叶斯信念网络 Bayesian Belief Networks

- BBN模型的一般特点。
  - (1)BBN提供了一种用图形模型来捕获特定领域的先验知识的方法。网络还可以用来对变量间的因果依赖关系进行编码。
  - (2)构造网络可能既费时又费力。然而，一旦网络结构确定下来，添加新变量就十分容易。
  - (3)贝叶斯网络很适合处理不完整的数据。对有属性遗漏的实例可以通过对该属性的所有可能取值的概率求和或求积分来加以处理。
  - (4)因为数据和先验知识以概率的方式结合起来了，所以该方法对模型的过分拟合问题是非常鲁棒的。

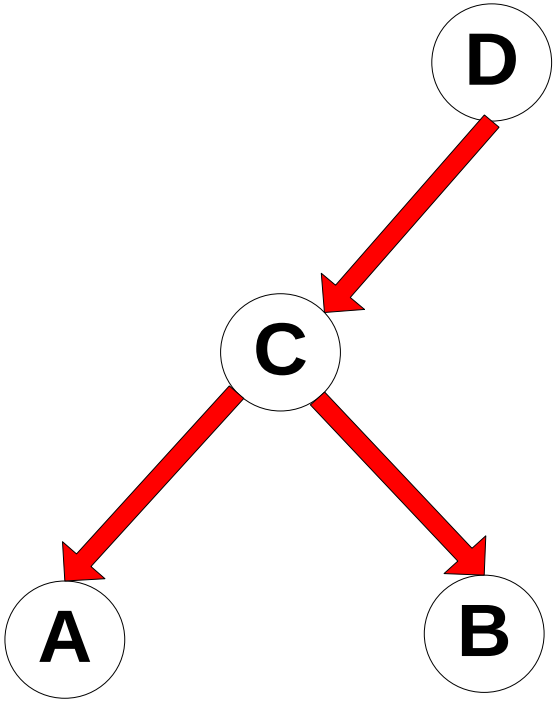
贝叶斯信念网络 Bayesian Belief Networks

- Provides graphical representation of probabilistic relationships among a set of random variables
- Consists of:
  - A directed acyclic graph (dag)
    - Node corresponds to a variable
    - Arc corresponds to dependence relationship between a pair of variables



  - A probability table associating each node to its immediate parent

Conditional Independence



D is parent of C

A is child of C

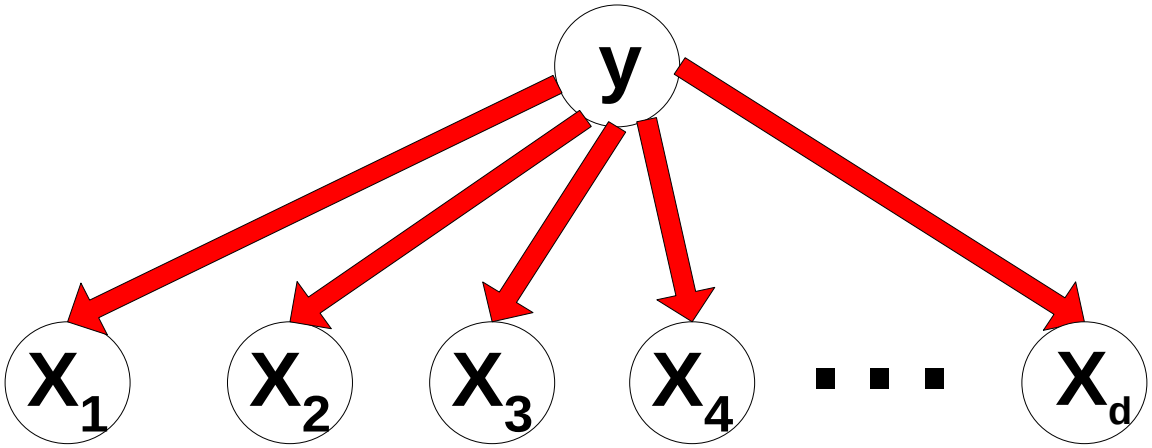
B is descendant of D

D is ancestor of A

- A node in a Bayesian network is conditionally independent of all of its nondescendants, if its parents are known

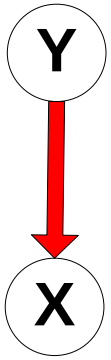
Conditional Independence

- Naïve Bayes assumption:



Probability Tables

- If X does not have any parents, table contains prior probability P(X)



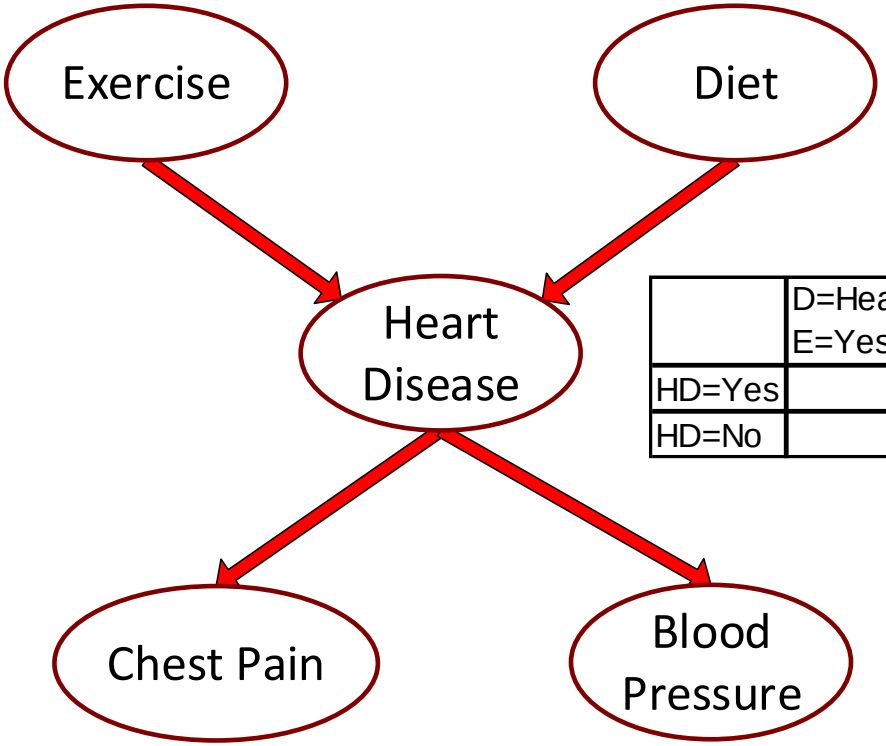
- If X has only one parent (Y), table contains conditional probability P(X|Y)

- If X has multiple parents (Y_1, Y_2,…, Y_k), table contains conditional probability P(X|Y_1, Y_2,…, Y_k)

Example of Bayesian Belief Network

Exercise=Yes	0.7
Exercise=No	0.3

Diet=Healthy	0.25
Diet=Unhealthy	0.75



	D=Healthy E=Yes	D=Healthy E=No	D=Unhealthy E=Yes	D=Unhealthy E=No
HD=Yes	0.25	0.45	0.55	0.75
HD=No	0.75	0.55	0.45	0.25

	HD=Yes	HD=No
CP=Yes	0.8	0.01
CP=No	0.2	0.99

	HD=Yes	HD=No
BP=High	0.85	0.2
BP=Low	0.15	0.8

Example of Inferencing using BBN

- Given: X = (E=No, D=Yes, CP=Yes, BP=High)
  - Compute P(HD|E,D,CP,BP)?

- P(HD=Yes| E=No,D=Yes) = 0.55
  P(CP=Yes| HD=Yes) = 0.8
  P(BP=High| HD=Yes) = 0.85
  - P(HD=Yes|E=No,D=Yes,CP=Yes,BP=High)
    $\propto 0.55 \times 0.8 \times 0.85 = 0.374$

- P(HD=No| E=No,D=Yes) = 0.45
  P(CP=Yes| HD=No) = 0.01
  P(BP=High| HD=No) = 0.2
  - P(HD=No|E=No,D=Yes,CP=Yes,BP=High)
    $\propto 0.45 \times 0.01 \times 0.2 = 0.0009$

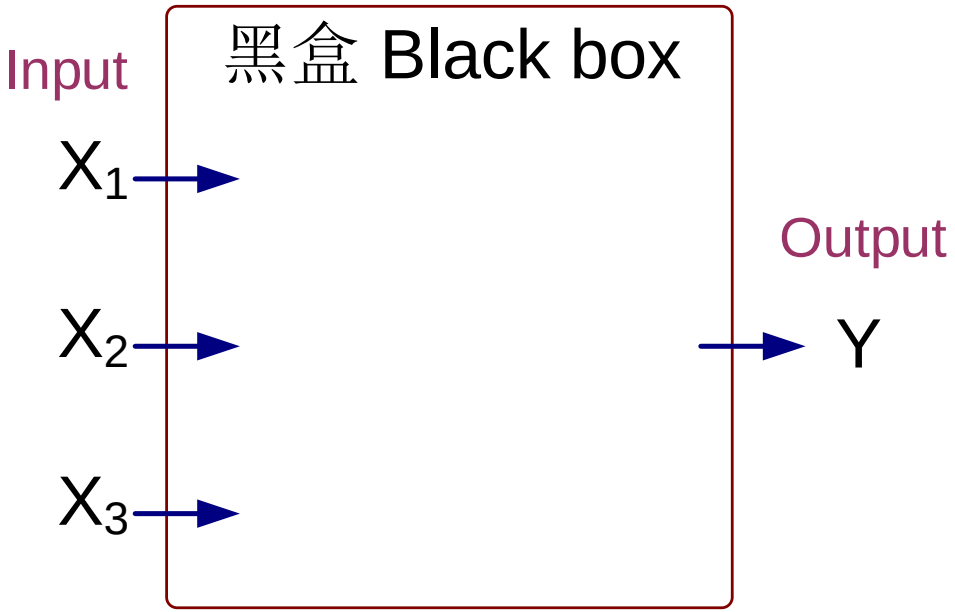
Classify X as Yes

数据挖掘
第4-4章 分类-人工神经网络

教师：王东京
学院：计算机学院
邮箱：dongjing.wang@hdu.edu.cn

人工神经网络 Artificial Neural Networks (ANN)

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



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

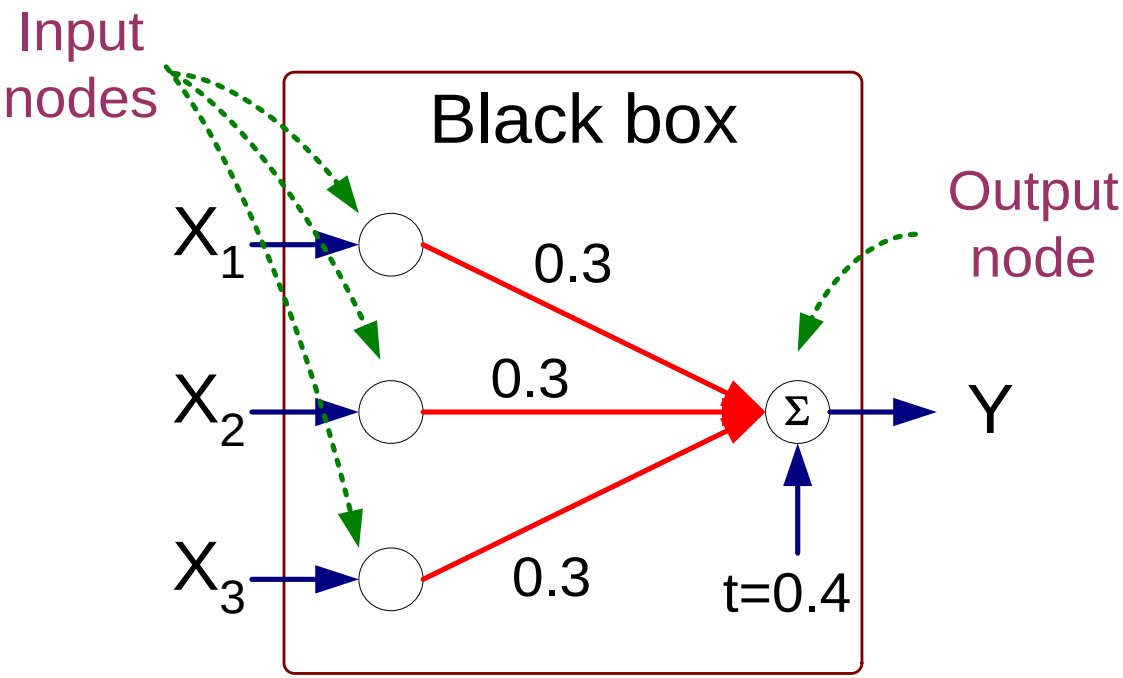
投票器

Artificial Neural Networks (ANN)

$$Y = sign(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4)$$

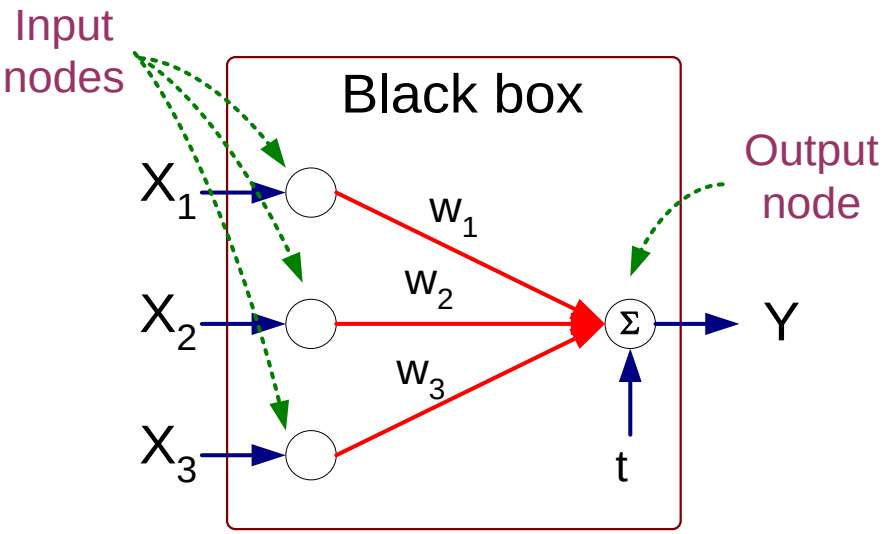
$$\text{其中 } sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

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



Artificial Neural Networks (ANN)

- 模型是相互连接的节点和加权链接的组合
- 输出节点根据其链接的权重对其每个输入值求和
- 将输出节点与某个阈值 t 比较



感知机模型Perceptron Model

$$Y = sign(\sum_{i=1}^{d} w_i X_i - t)$$

$$= sign(\sum_{i=0}^{d} w_i X_i)$$

感知机 Perceptron

- 单层网络 Single layer network
  - Contains only input and output nodes
- 激活函数 Activation function: f = sign(w•x)
- 模型的使用非常直观简单

$$Y = sign(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4)$$

$$\text{where } sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

  - $X_1 = 1, X_2 = 0, X_3 = 1 \Rightarrow y = sign(0.2) = 1$

感知机学习规则 Perceptron Learning Rule

- 初始化权重 (w_0, w_1, …, w_d)
- 重复
  - 对于每个训练样例(x_i, y_i)
    - 计算 f(w, x_i)
    - 更新权重:

$$w^{(k+1)} = w^{(k)} + \lambda\left[y_i - f(w^{(k)}, x_i)\right]x_i$$

- 条件满足则停止训练

Perceptron Learning Rule

- 权重更新公式:

$$w^{(k+1)} = w^{(k)} + \lambda\left[y_i - f(w^{(k)}, x_i)\right]x_i \ ;\ \lambda\text{: 学习率}$$

- 直觉 Intuition:
  - Update weight based on error: $e = \left[y_i - f(w^{(k)}, x_i)\right]$
  - If y=f(x,w), e=0: no update needed
  - If y>f(x,w), e=2: weight must be increased so that f(x,w) will increase
  - If y<f(x,w), e=-2: weight must be decreased so that f(x,w) will decrease

Example of Perceptron Learning

$$w^{(k+1)} = w^{(k)} + \lambda\left[y_i - f(w^{(k)}, x_i)\right]x_i$$

$$Y = sign(\sum_{i=0}^{d} w_i X_i)$$

$$\lambda = 0.1$$

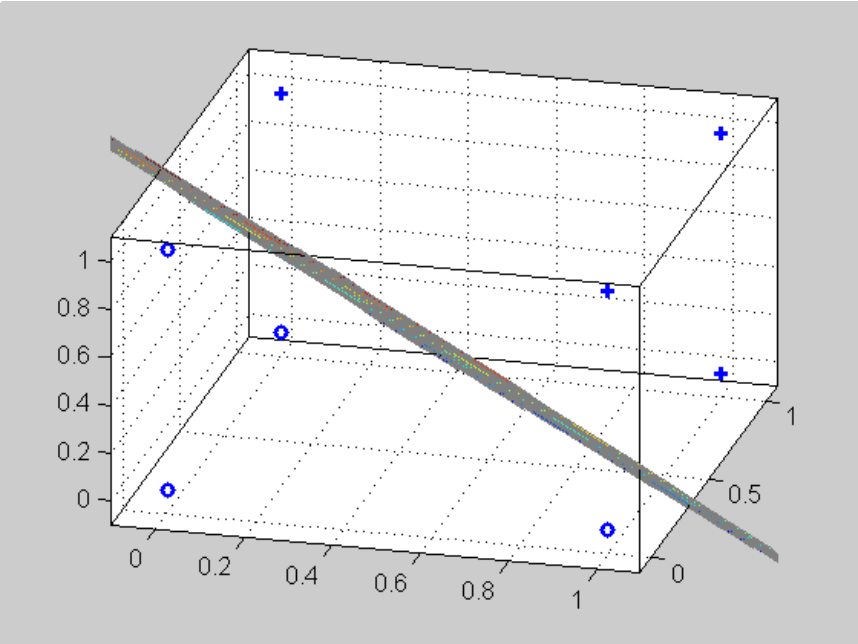
X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	-0.2	0	0
2	0	0	0	0.2
3	0	0	0	0.2
4	0	0	0	0.2
5	-0.2	0	0	0
6	-0.2	0	0	0
7	0	0	0.2	0.2
8	-0.2	0	0.2	0.2

Epoch	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	0	0.2	0.2
2	-0.2	0	0.4	0.2
3	-0.4	0	0.4	0.2
4	-0.4	0.2	0.4	0.4
5	-0.6	0.2	0.4	0.2
6	-0.6	0.4	0.4	0.2

Perceptron Learning Rule

- 由于 f(w,x) 是输入变量的线性组合（linear combination），因此决策边界是线

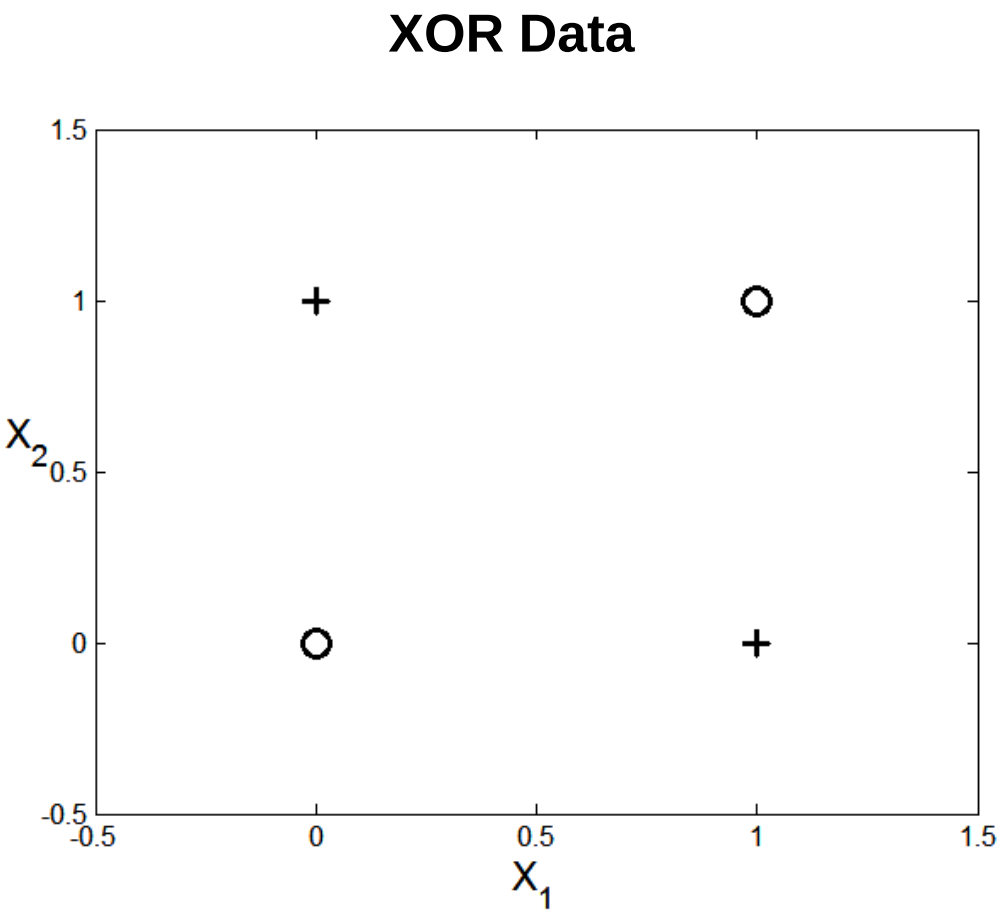


- 对于线性不可分（nonlinearly separable）问题，感知器学习算法将失败，因为没有线性超平面可以完美地分离数据

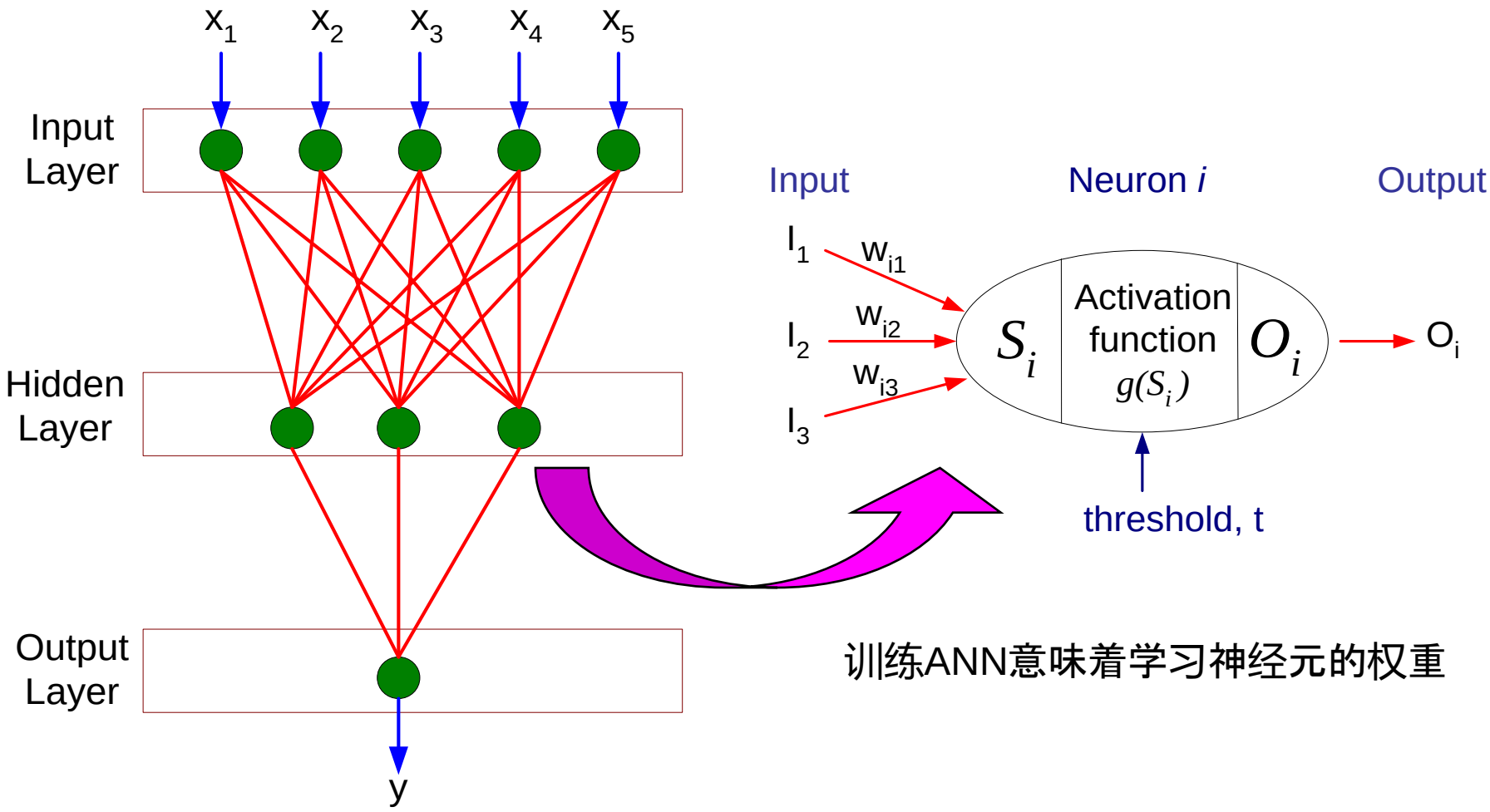
线性不可分数据 Nonlinearly Separable Data

$$y = x_1 \oplus x_2$$

x_1	x_2	y
0	0	-1
1	0	1
0	1	1
1	1	-1



多层神经网络的一般结构



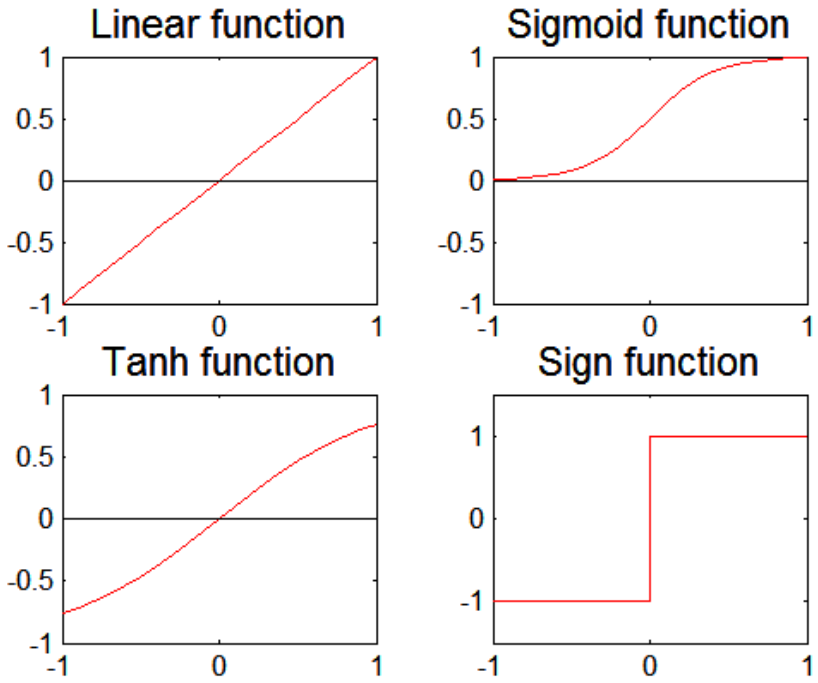
训练ANN意味着学习神经元的权重

Artificial Neural Networks (ANN)

- 各种类型的神经网络拓扑
  - 单层网络（感知器）与多层网络
  - 前馈与递归网络

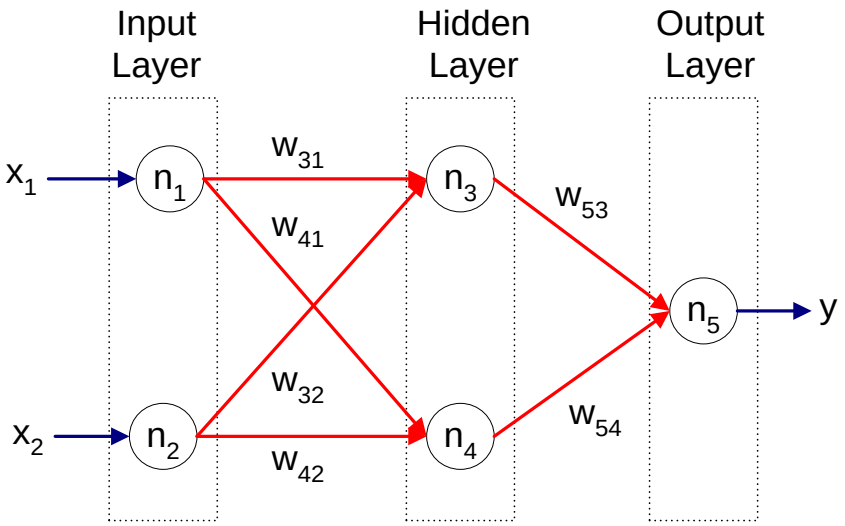
- 各种类型的激活函数（f）

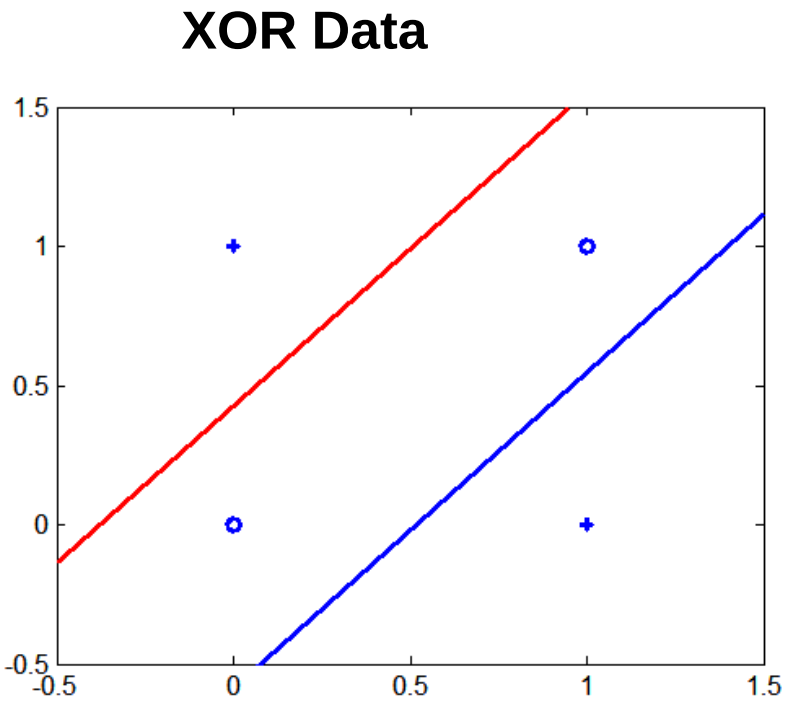
$$Y = f(\sum_i w_i X_i)$$



多层神经网络 Multi-layer Neural Network

- 多层神经网络可以解决涉及非线性决策面的任何类型的分类任务





多层神经网络的学习

- 我们可以将感知器学习规则应用于每个节点，包括隐藏节点吗?
  - 感知器学习规则计算误差项 e = y-f(w,x) 并相应地更新权重
    - 问题：如何确定隐藏节点的y的真值?
  - 如果根据输出节点中的错误近似估计隐藏节点中的错误
    - 问题：
      - 不清楚隐藏节点中的调整如何影响整体错误
      - 无法保证收敛到最优解

梯度下降（gradient descent ）

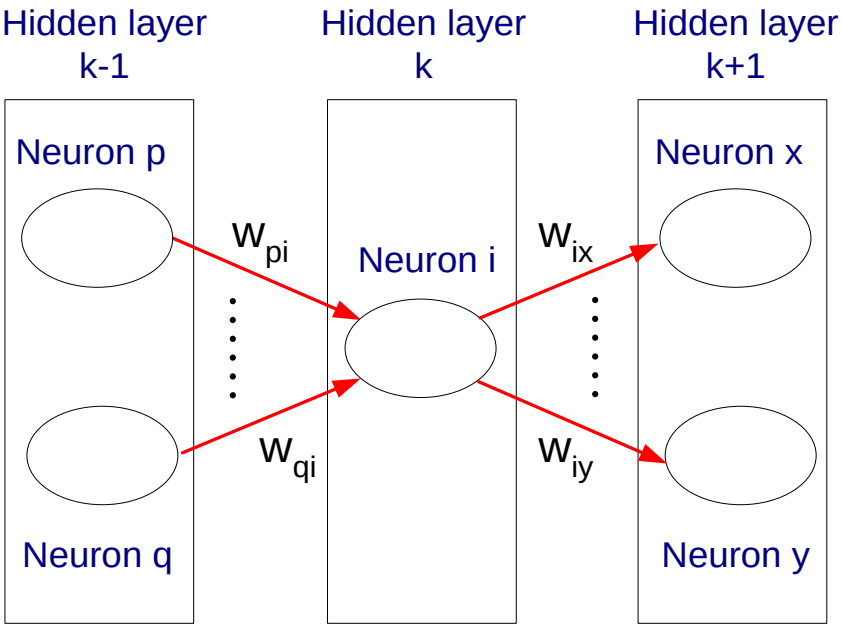
- 权重更新: $$w_j^{(k+1)} = w_j^{(k)} - \lambda \frac{\partial E}{\partial w_j}$$
- 误差函数: $$E = \frac{1}{2}\sum_{i=1}^{N}\left(t_i - f(\sum_j w_j x_{ij})\right)$$
- 激活函数必须是可微的（differentiable）
- For sigmoid function:

$$w_j^{(k+1)} = w_j^{(k)} + \lambda \sum_i (t_i - o_i) o_i (1 - o_i) x_{ij}$$

- Stochastic gradient descent (update the weight immediately)

梯度下降（gradient descent ）

- 对于输出神经元，权重更新公式与之前相同（感知器的梯度下降）



- 对于隐藏的神经元：

$$w_{pi}^{(k+1)} = w_{pi}^{(k)} + \lambda o_i (1 - o_i) \sum_{j \in \Phi_i} \delta_j w_{ij} x_{pi}$$

$$\text{Output neurons}: \delta_j = o_j (1 - o_j)(t_j - o_j)$$

$$\text{Hidden neurons}: \delta_j = o_j (1 - o_j) \sum_{k \in \Phi_j} \delta_k w_{jk}$$

Design Issues in ANN

- 输入层中的节点数
  - 每个二值/连续属性对应一个输入节点
  - 每个具有k个值的类别属性需要 k 个或 $\log_2 k$ 个节点
- 输出层中的节点数
  - 二分类问题需要一个输出节点
  - K分类问题需要 k 个或 $\log_2 k$ 个节点
- 隐藏层中的节点数
- 初始权重和偏差

Characteristics of ANN

- 多层人工神经网络是通用逼近器（approximators），但如果网络太大，可能会出现过拟合（overfitting）的情况
- 梯度下降可能会收敛到局部最小值（local minimum）
- 建立模型（训练）非常耗时，但是测试可能非常快
- 可以处理冗余属性，因为权重是自动学习的
- 对训练数据中的噪声敏感
- 难以处理缺失属性

深度神经网络 Deep Neural Networks

- 涉及大量隐藏层（hidden layers）
- 可以在多个抽象级别上表示要素
- 通常，每层需要较少的节点就能实现类似于浅层网络的泛化性能
- 深度网络已成为视觉和自然语言处理等复杂问题的首选技术

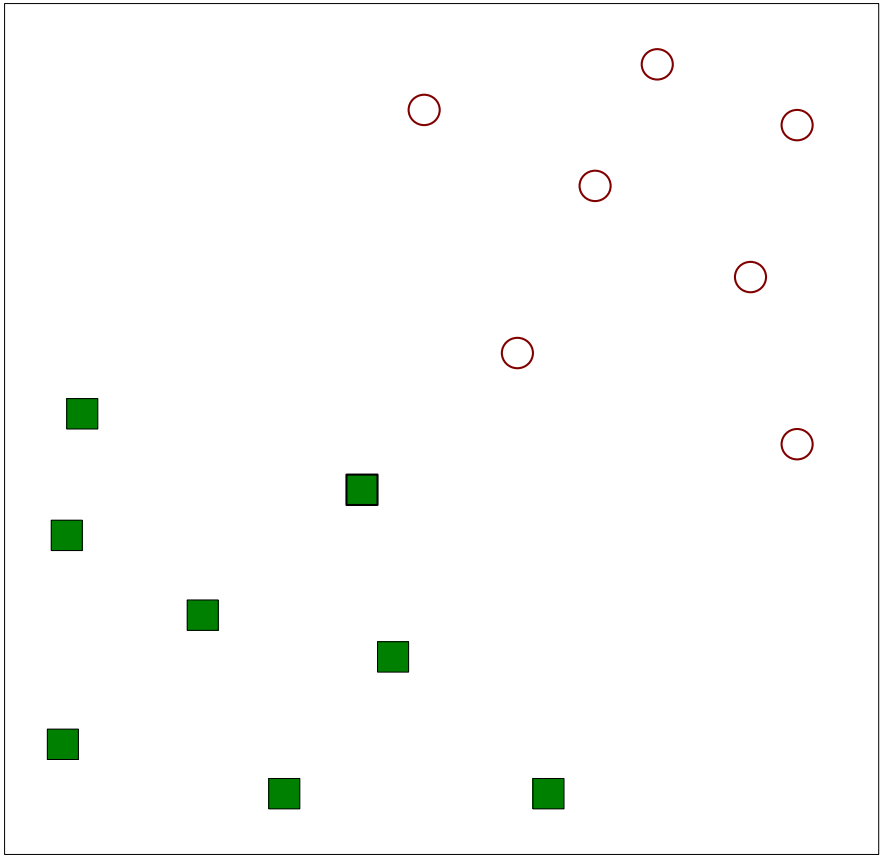
Deep Nets: Challenges and Solutions

- **挑战**
  - 收敛缓慢（Slow convergence）
  - 对模型参数初始值的敏感性
  - 大量节点使深层网络易于过拟合（overfitting）
- **解决方案**
  - 更大型的训练数据集
  - 先进的计算能力，例如GPU
  - 算法进步
    - 新架构和激活单元
    - 更好的参数和超参数选择策略
    - 正则化

数据挖掘
第4-5章 分类-支持向量机

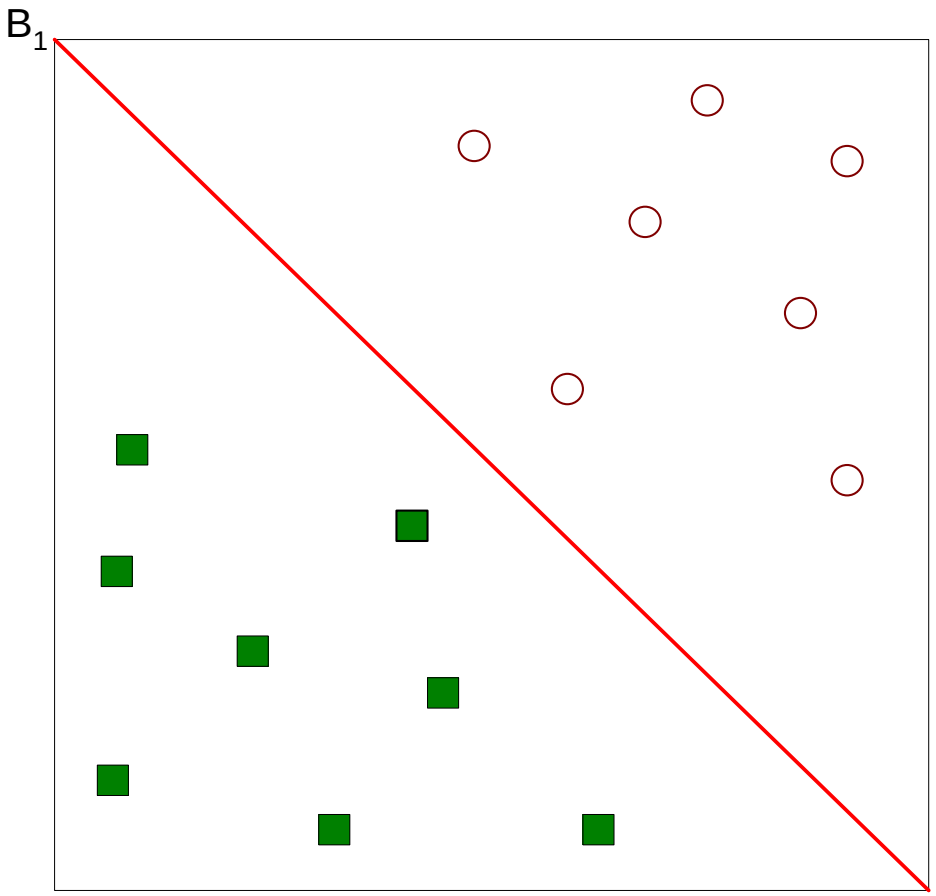
教师：王东京
学院：计算机学院
邮箱：dongjing.wang@hdu.edu.cn

支持向量机 Support Vector Machines



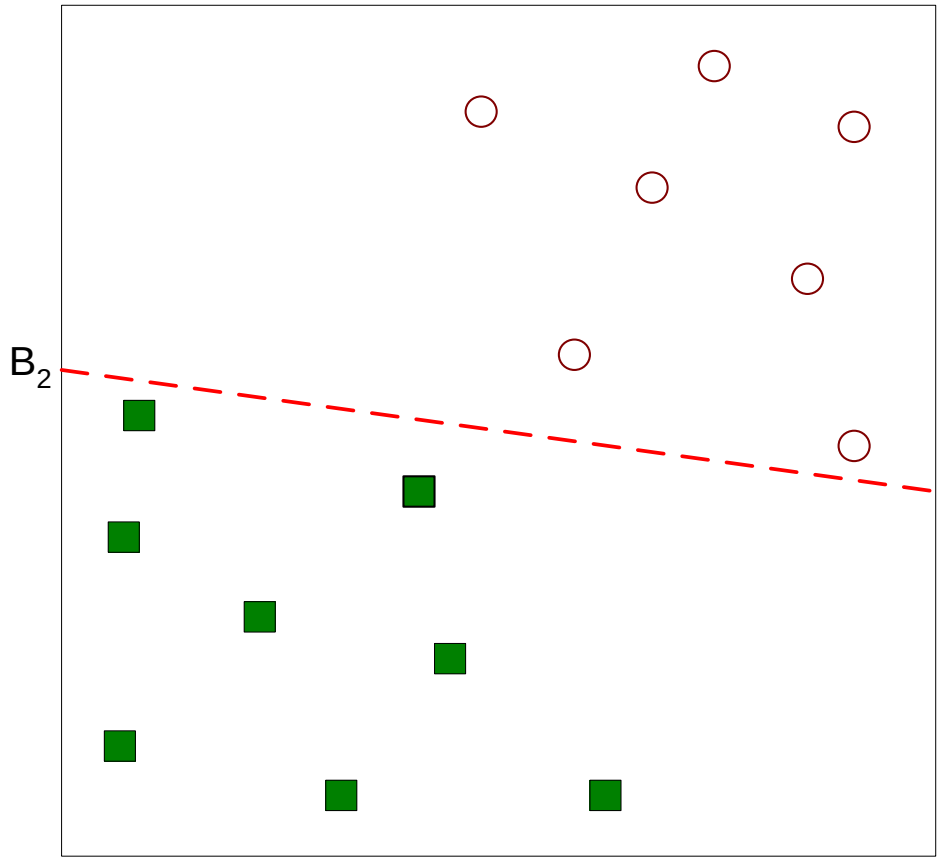
- 查找将数据分开的线性超平面（linear hyperplane ） （决策边界）

Support Vector Machines



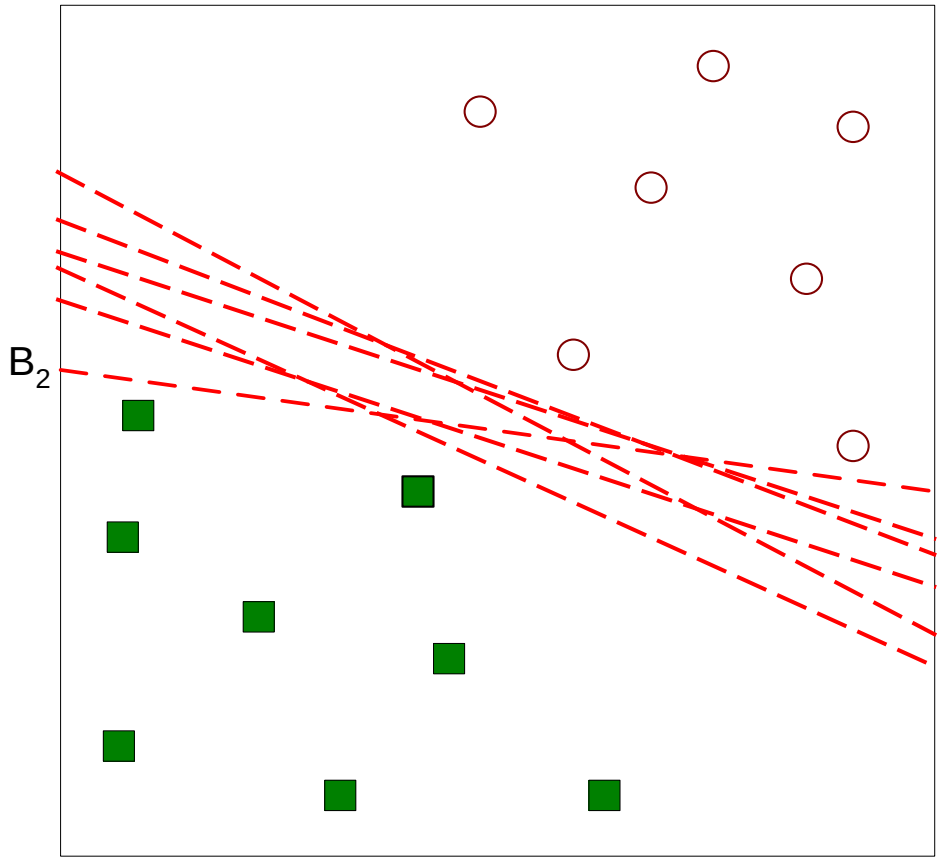
- One Possible Solution

Support Vector Machines



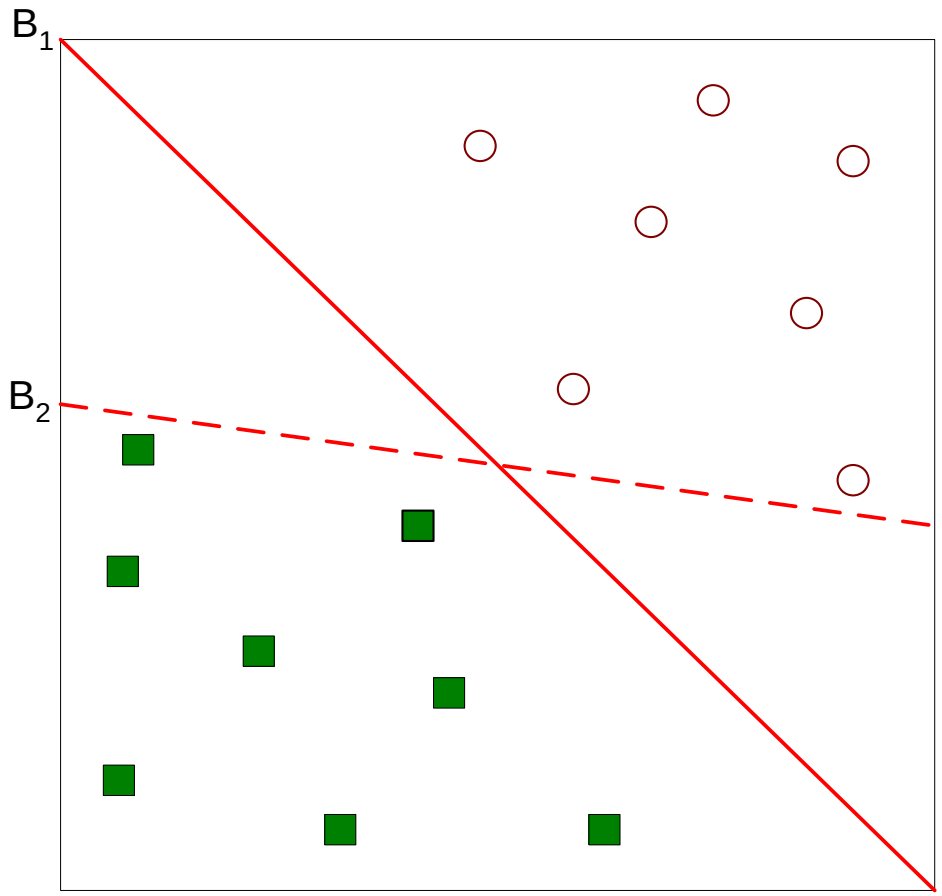
- Another possible solution

Support Vector Machines



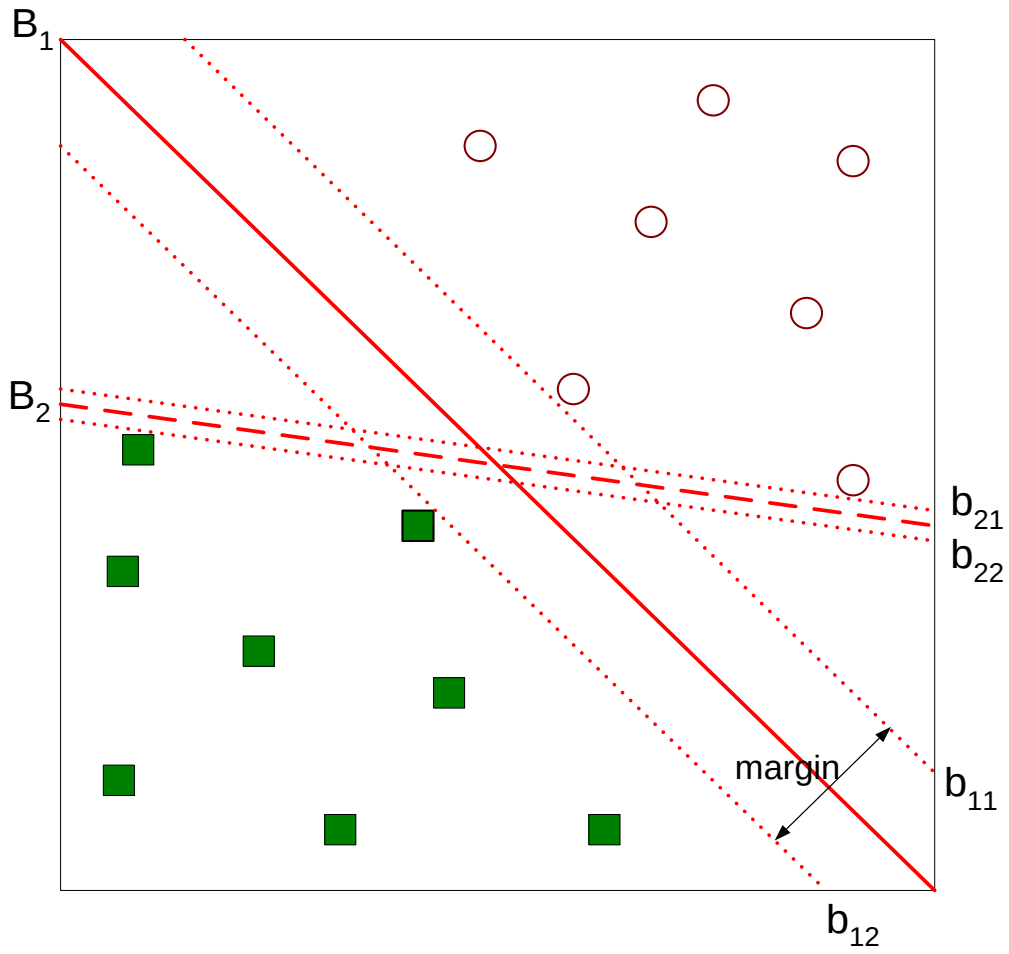
- Other possible solutions

Support Vector Machines



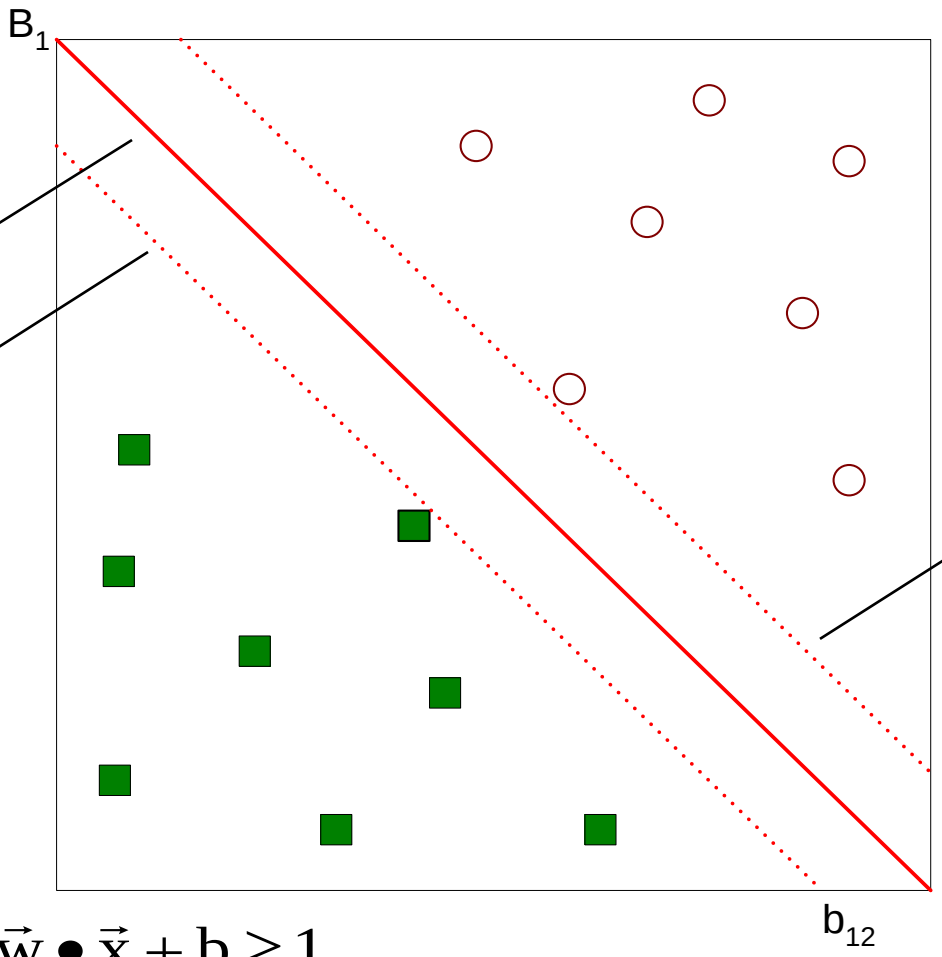
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



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

Support Vector Machines



$\vec{w} \bullet \vec{x} + b = 0$

$\vec{w} \bullet \vec{x} + b = -1$

$\vec{w} \bullet \vec{x} + b = +1$

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

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

线性SVM-Linear SVM

- Linear model:

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

- Learning the model is equivalent to determining the values of $\vec{w}$ and b
  - How to find $\vec{w}$ and b from training data?

Learning Linear SVM

- Objective is to maximize: $\text{Margin} = \frac{2}{\| \vec{w} \|}$
  - Which is equivalent to minimizing: $L(\vec{w}) = \frac{\| \vec{w} \|^2}{2}$
  - Subject to the following constraints:

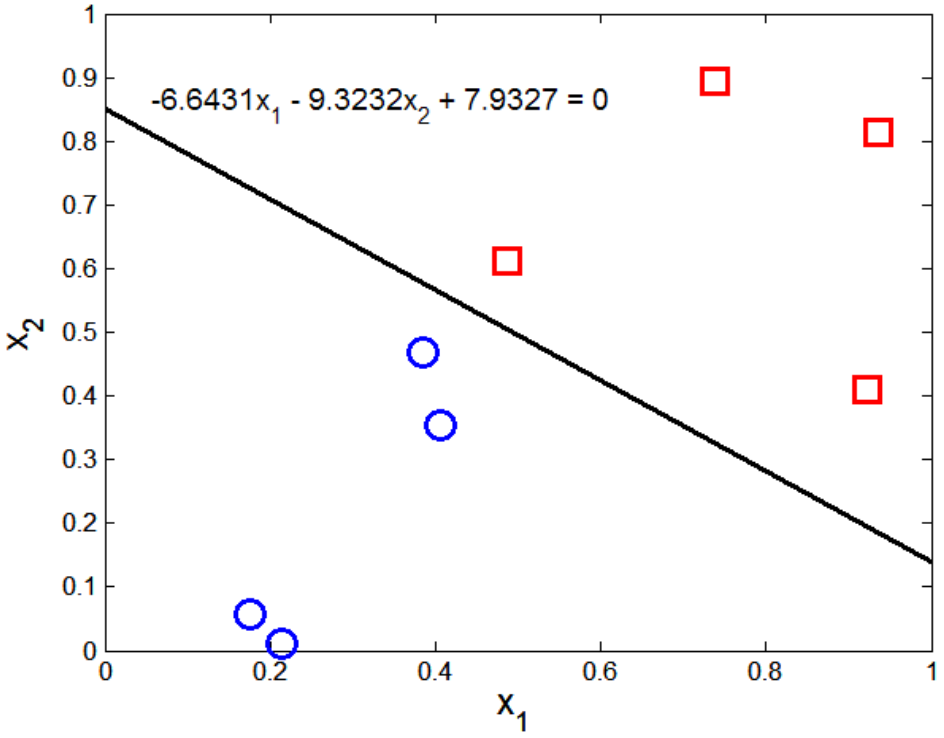
$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \leq -1 \end{cases}$$

or

$$y_i(\mathrm{w} \bullet \mathrm{x}_i + b) \geq 1, \qquad i = 1,2,\ldots,N$$

  - This is a constrained optimization problem
    - Solve it using Lagrange multiplier method

Example of Linear SVM



Support vectors

x1	x2	y	λ
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

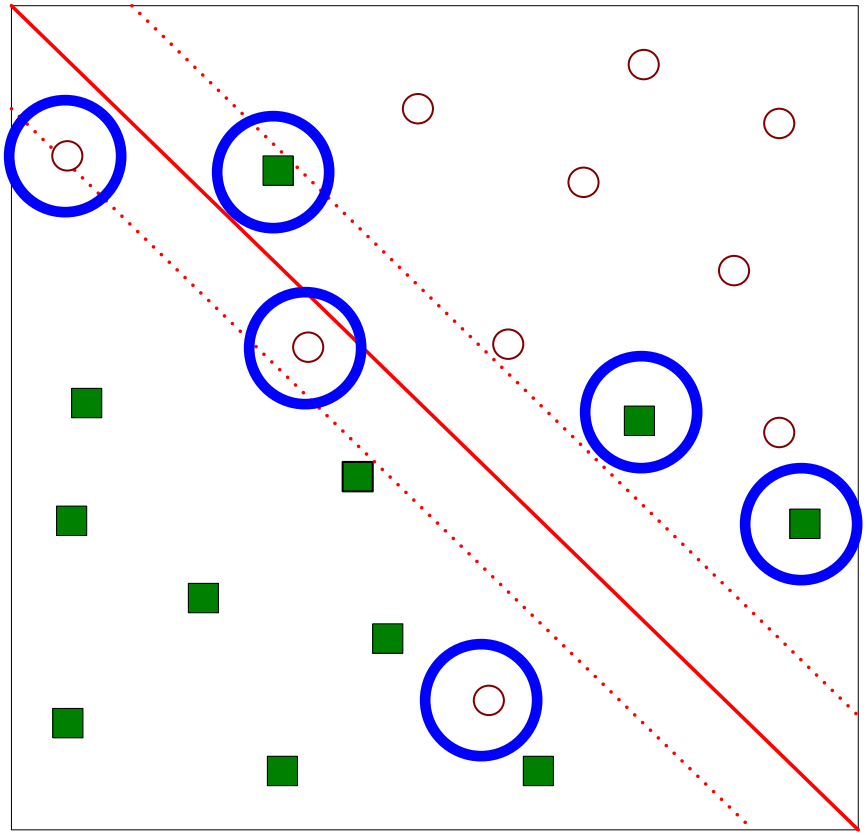
Learning Linear SVM

- Decision boundary depends only on support vectors
  - If you have data set with same support vectors, decision boundary will not change
  - How to classify using SVM once **w** and *b* are found? Given a test record, x_i

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

Support Vector Machines

- 线性不可分问题？



Support Vector Machines

- 线性不可分问题？
  - 引入松弛变量 slack variables
    - Need to minimize:

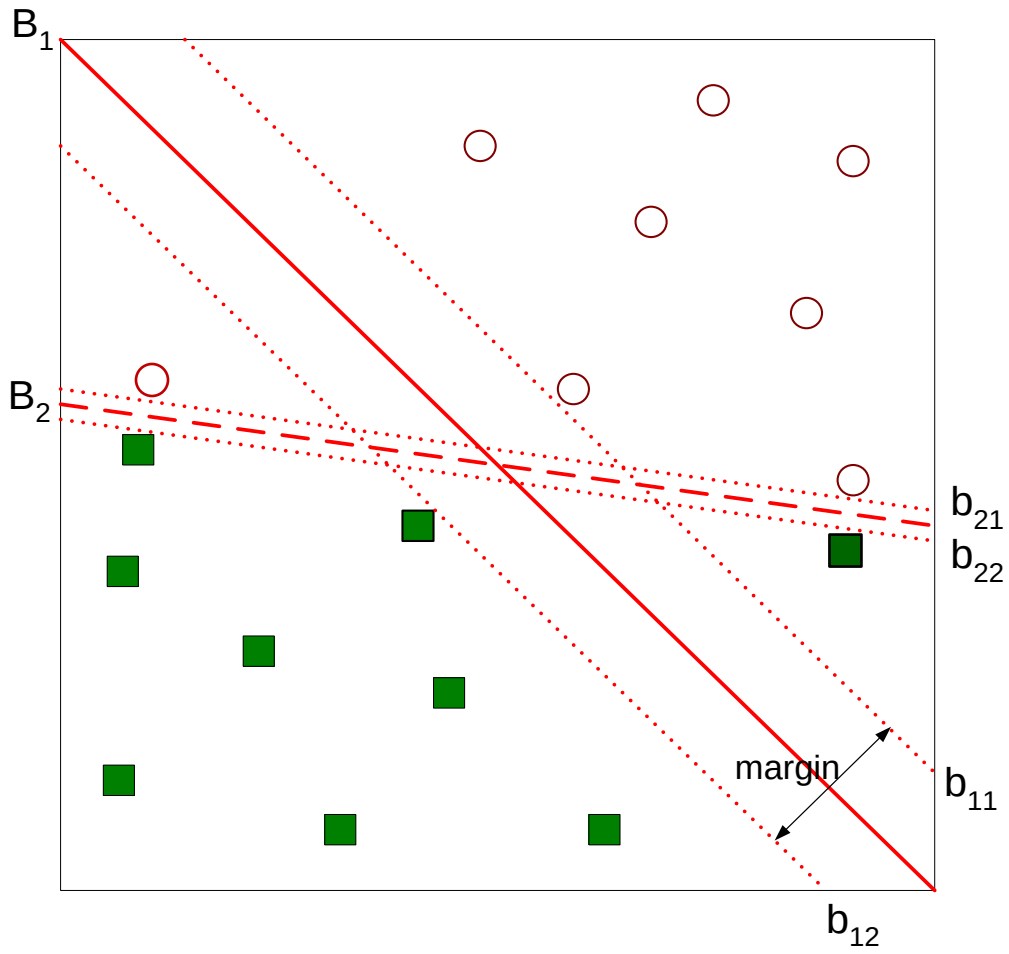
$$L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k \right)$$

    - Subject to:

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

    - If k is 1 or 2, this leads to similar objective function as linear SVM but with different constraints (see textbook)

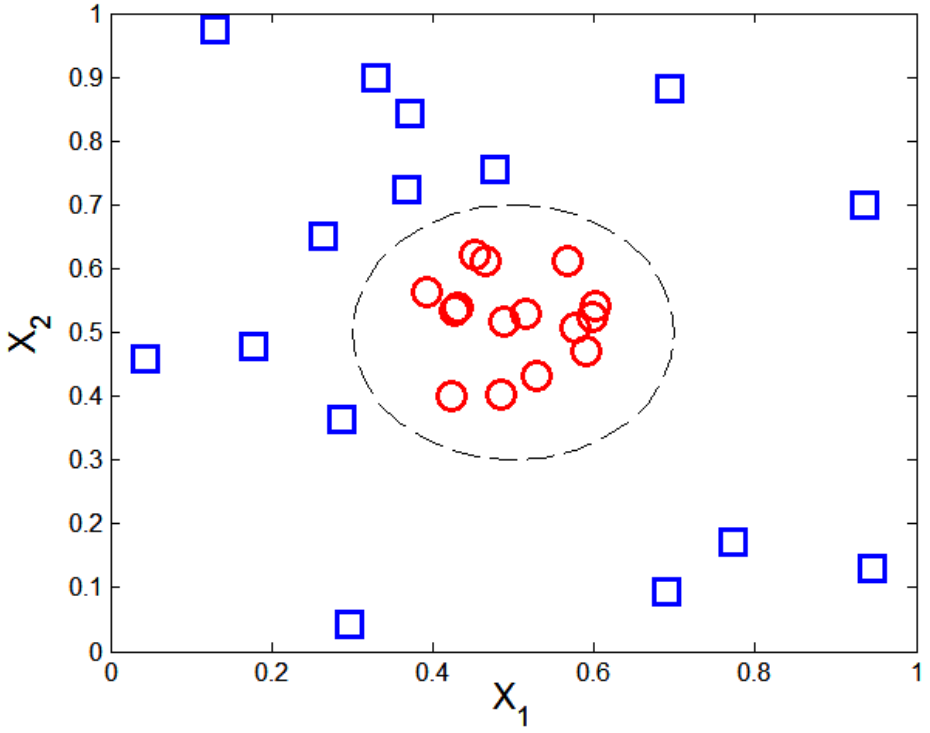
Support Vector Machines



- Find the hyperplane that optimizes both factors

Nonlinear Support Vector Machines

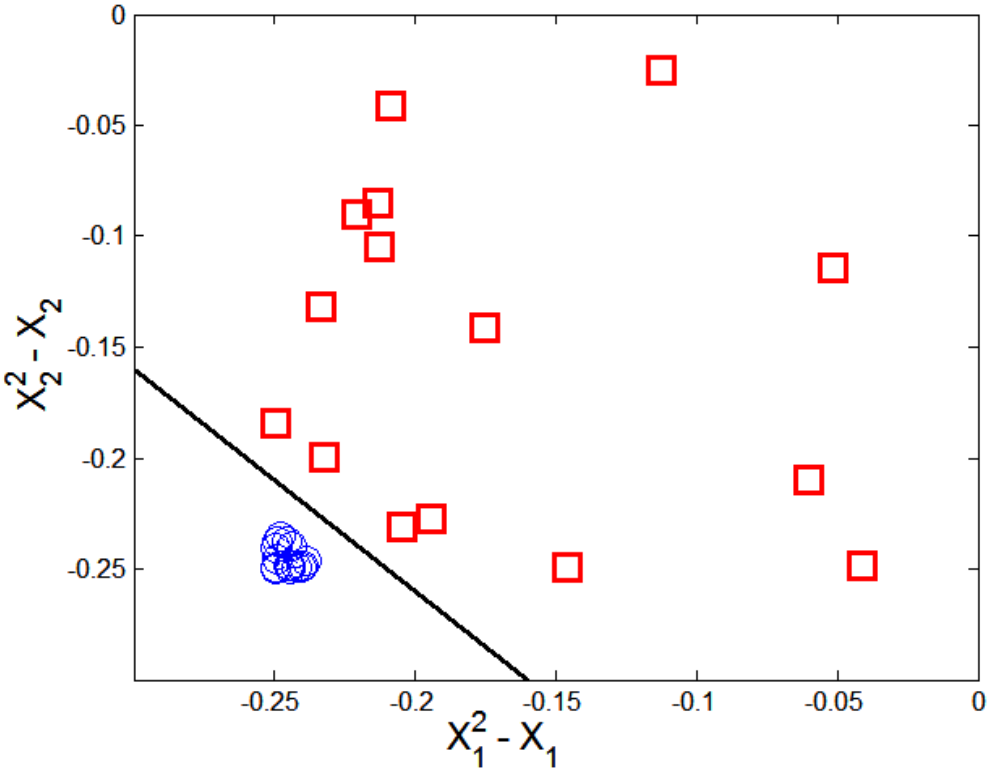
- What if decision boundary is not linear?



$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

Nonlinear Support Vector Machines

- Transform data into higher dimensional space



$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Decision boundary:

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

Learning Nonlinear SVM

- Optimization problem:

$$\min_{w} \frac{\|\mathbf{w}\|^2}{2}$$
$$subject\ to \quad y_i(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b) \geq 1,\ \forall\{(\boldsymbol{x}_i, y_i)\}$$

- Which leads to the same set of equations (but involve Φ(x) instead of x)

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

$$\mathbf{w} = \sum_i \lambda_i y_i \Phi(\mathbf{x}_i)$$

$$\lambda_i \{ y_i (\sum_j \lambda_j y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) + b) - 1 \} = 0,$$

$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \Phi(\mathbf{z}) + b) = sign(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b).$$

Learning NonLinear SVM

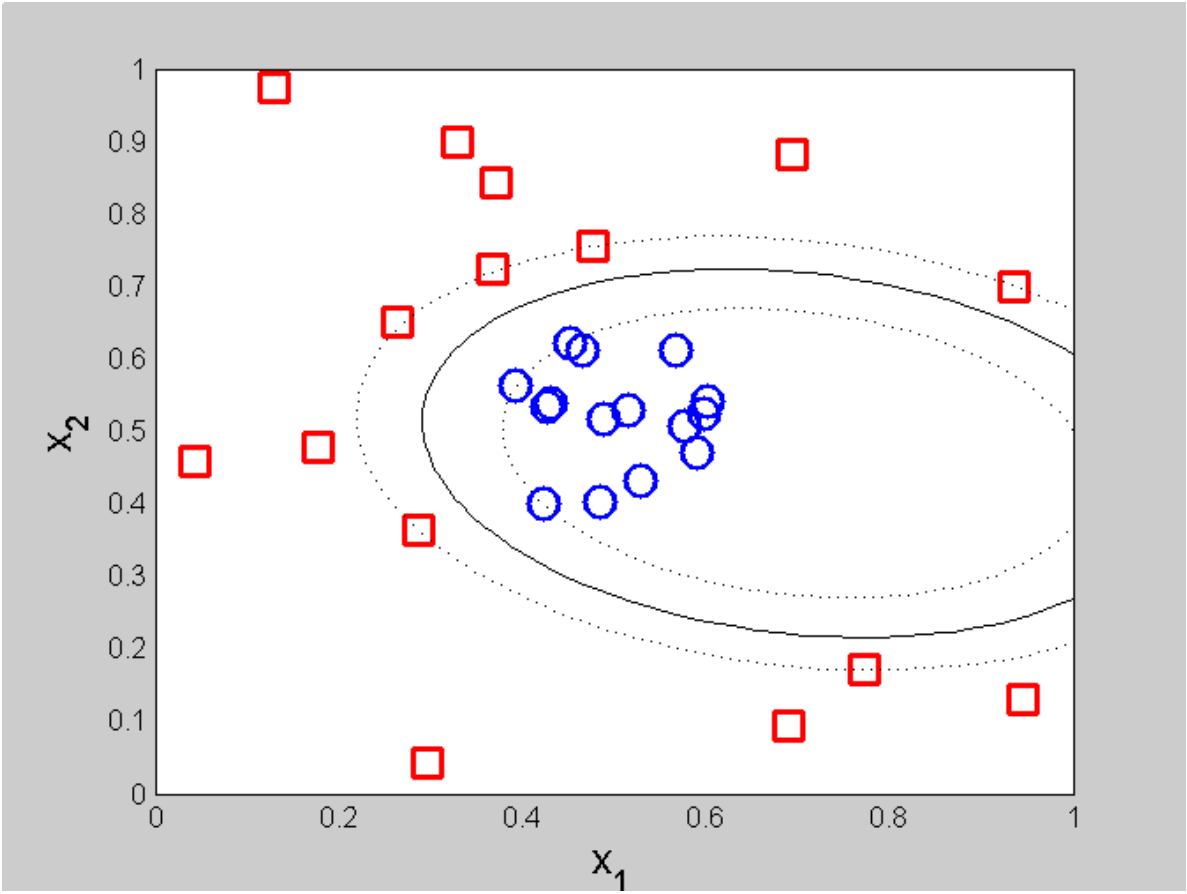
- Issues:
  - What type of mapping function Φ should be used?
  - How to do the computation in high dimensional space?
    - Most computations involve dot product $\Phi(x_i) \bullet \Phi(x_j)$
    - Curse of dimensionality?

Learning Nonlinear SVM

- Kernel Trick:
  - $\Phi(x_i) \bullet \Phi(x_j) = K(x_i, x_j)$
  - $K(x_i, x_j)$ is a kernel function (expressed in terms of the coordinates in the original space)
    - Examples:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$
$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$
$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$

Example of Nonlinear SVM



SVM with polynomial degree 2 kernel

Learning Nonlinear SVM

- Advantages of using kernel:
  - Don't have to know the mapping function Φ
  - Computing dot product $\Phi(x_i) \bullet \Phi(x_j)$ in the original space avoids curse of dimensionality

- Not all functions can be kernels
  - Must make sure there is a corresponding Φ in some high-dimensional space
  - Mercer's theorem (see textbook)

Characteristics of SVM

- 学习问题被公式化为凸优化（convex optimization）问题
  - 可以使用有效算法找到全局最小值（global minima）
  - 许多其他方法使用贪婪方法并找到局部最优解
  - 建立模型的计算复杂度很高

- 对噪音更鲁棒（Robust to noise）
- 通过最大化决策边界的边缘来避免过拟合
- 与许多其他技术相比，SVM可以更好地处理无关和冗余的属性
- 用户需要提供核函数（kernel function）和目标函数（cost function）
- 难以处理缺失值

数据挖掘
第4-6章 分类-集成学习

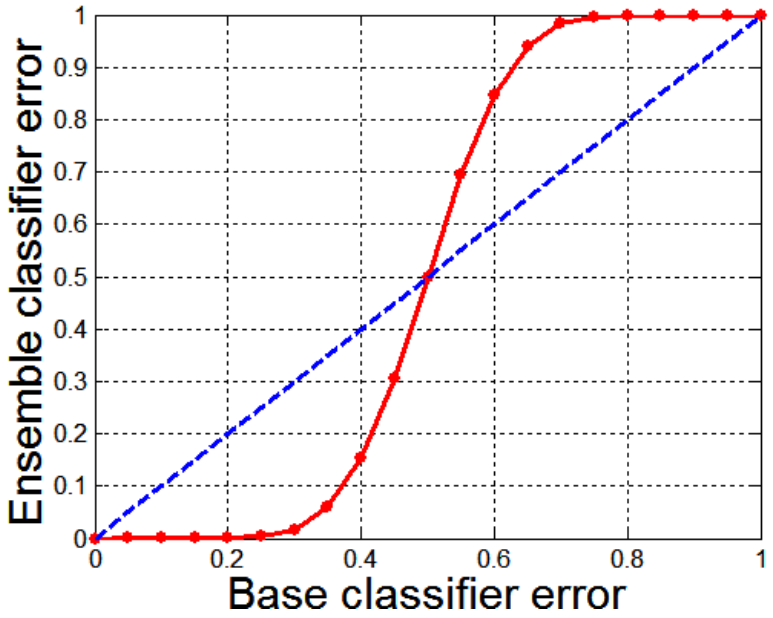
教师：王东京
学院：计算机学院
邮箱：dongjing.wang@hdu.edu.cn

集成学习方法 Ensemble Methods

- 根据训练数据构造一组分类器
- 通过组合多个分类器的预测结果，来预测测试记录的类别标签

为什么集成学习有效果?

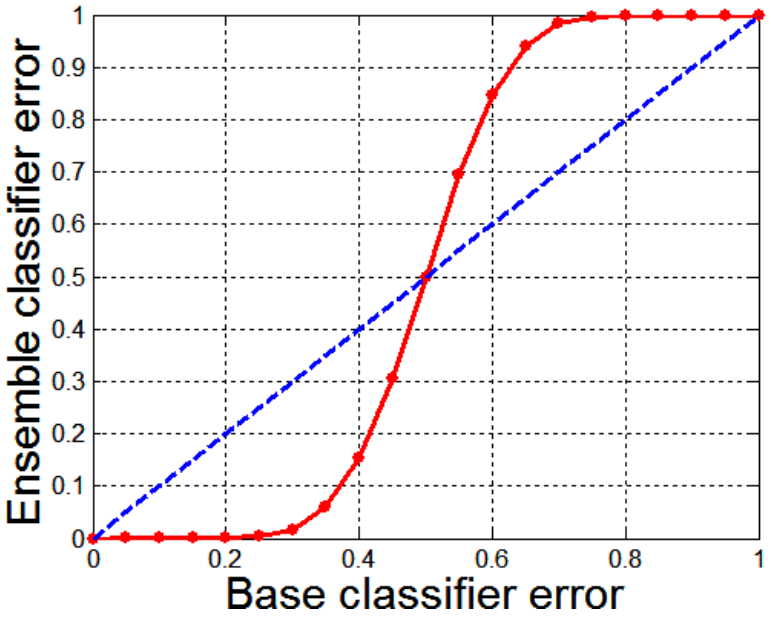
- 假设有25个基本分类器
  - 每个分类器的错误率是0.35
  - 假设不同分类器所犯的错误是不相关的
  - 集成分类器做出错误预测的概率:



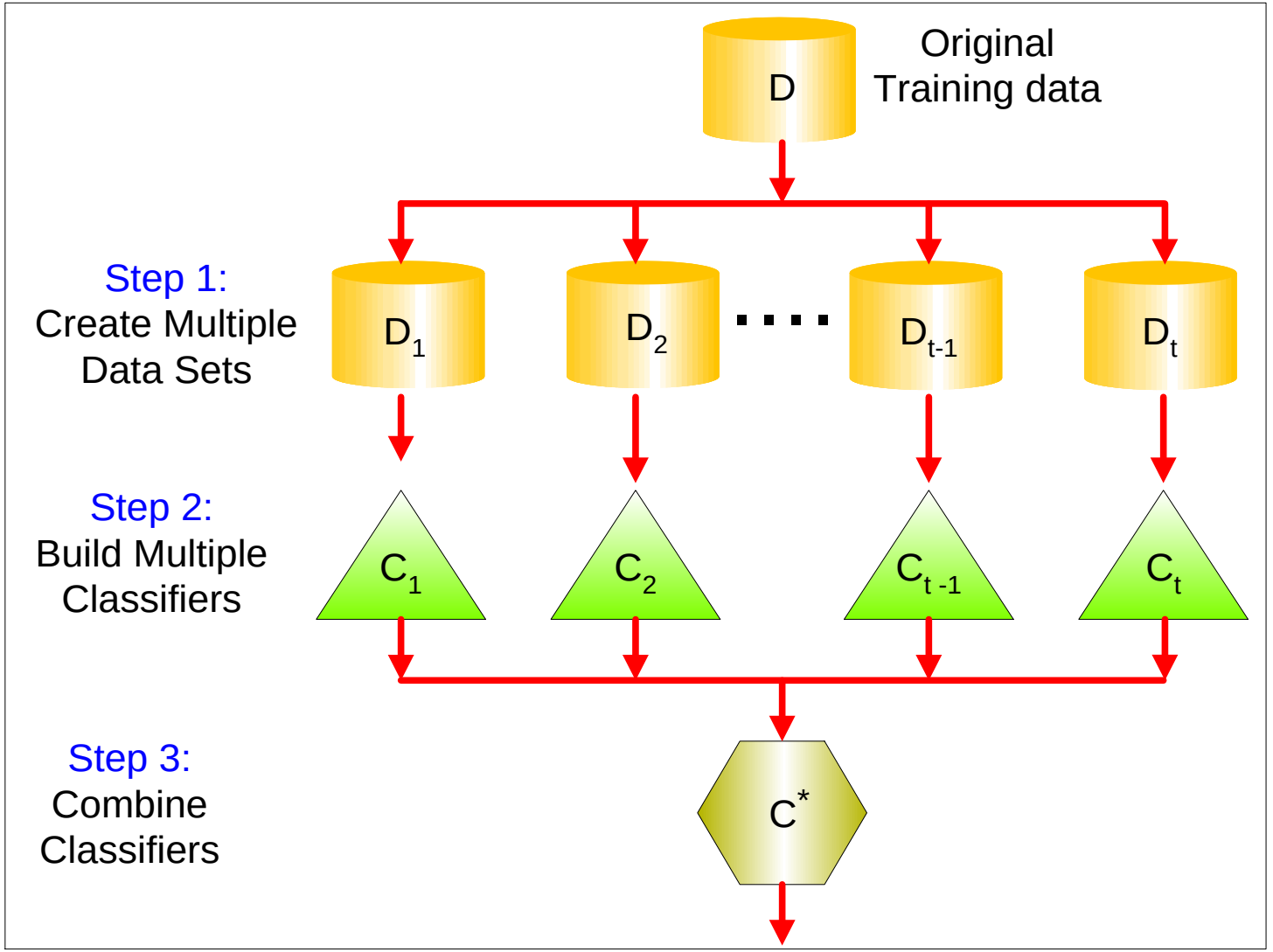
$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

为什么集成学习有效果?

- （1）基分类器之间应该是相互独立的；
- （2）基分类器应当好于随机猜测分类器。



General Approach



Types of Ensemble Methods

- 处理训练数据分布
  - Example: bagging, boosting
- 处理输入特征
  - Example: random forests
- 处理类别标签
  - Example: error-correcting output coding

- 处理学习算法

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 data instance has probability $1- (1 – 1/n)^n$ of being selected as part of the bootstrap sample

Bagging Algorithm

Algorithm 5.6 Bagging Algorithm

1: Let k be the number of bootstrap samples.
2: **for** $i = 1$ to k **do**
3: Create a bootstrap sample of size n, D_i.
4: Train a base classifier C_i on the bootstrap sample D_i.
5: **end for**
6: $C^*(x) = \arg\max_y \sum_i \delta\big(C_i(x) = y\big)$, $\{\delta(\cdot) = 1$ if its argument is true, and 0 otherwise.$\}$

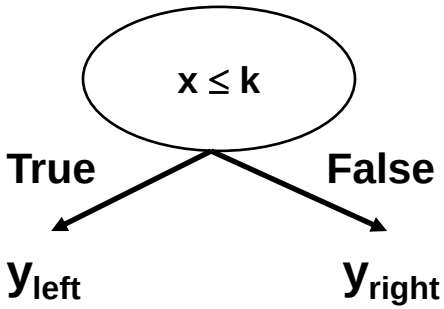
Bagging Example

- Consider 1-dimensional data set:

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Classifier is a decision stump
  - Decision rule: $x \leq k$ versus $x > k$
  - Split point k is chosen based on entropy



Bagging Example

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Example

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Round 2:

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

x <= 0.7 ➔ y = 1
x > 0.7 ➔ y = 1

Bagging Round 3:

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Round 4:

x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x <= 0.3 ➔ y = 1
x > 0.3 ➔ y = -1

Bagging Round 5:

x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1
y	1	1	1	-1	-1	-1	-1	1	1	1

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Example

Bagging Round 6:

x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1
y	1	-1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 ➔ y = -1
x > 0.75 ➔ y = 1

Bagging Round 7:

x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1
y	1	-1	-1	-1	-1	1	1	1	1	1

x <= 0.75 ➔ y = -1
x > 0.75 ➔ y = 1

Bagging Round 8:

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 ➔ y = -1
x > 0.75 ➔ y = 1

Bagging Round 9:

x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 ➔ y = -1
x > 0.75 ➔ y = 1

Bagging Round 10:

x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
y	1	1	1	1	1	1	1	1	1	1

x <= 0.05 ➔ y = 1
x > 0.05 ➔ y = 1

Bagging Example

- Summary of Training sets:

Round	Split Point	Left Class	Right Class
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

Bagging Example

- Assume test set is the same as the original data
- Use majority vote to determine class of ensemble classifier

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1

Predicted Class

Boosting

- 通过更多地关注先前错误分类的记录来自适应地更改训练数据分布的迭代过程
  - 最初，所有N条记录均分配有相等的权重
  - 与bagging不同，样本权重可能会在每个迭代结束时发生变化

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

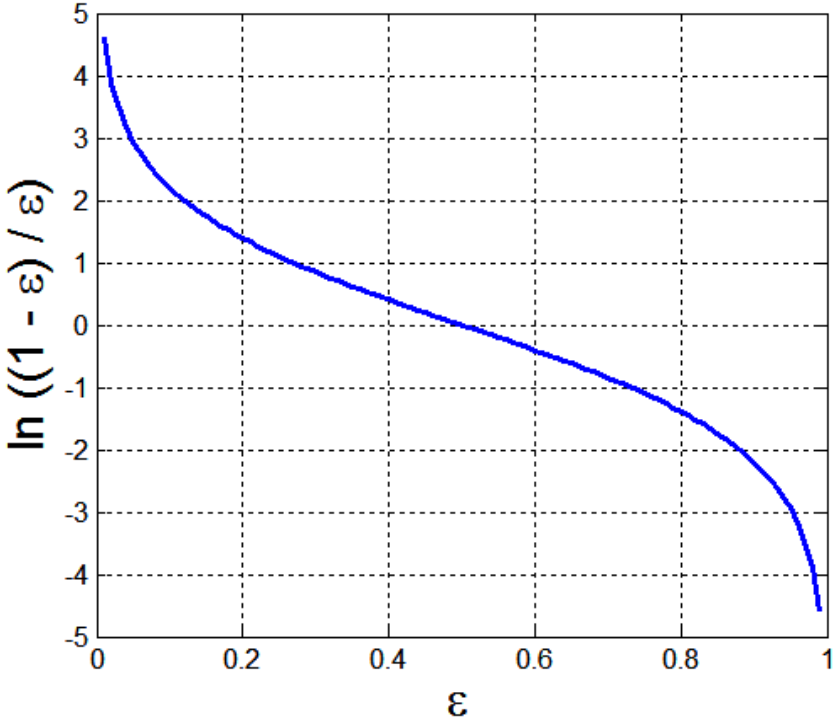
AdaBoost

- Base classifiers: C_1, C_2, …, 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:

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



AdaBoost Algorithm

- Weight update:

$$w_j^{(i+1)} = \frac{w_j^{(i)}}{Z_i} \begin{cases} \exp^{-\alpha_i} & \text{if } C_i(x_j) = y_j \\ \exp^{\alpha_i} & \text{if } C_i(x_j) \neq y_j \end{cases}$$

where Z_i 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:

$$C^*(x) = \underset{y}{\operatorname{argmax}} \sum_{i=1}^{T} \alpha_i \delta(C_i(x) = y)$$

AdaBoost Algorithm

Algorithm 5.7 AdaBoost Algorithm

1: $\mathbf{w} = \{w_j = 1/n \mid j = 1, 2, \cdots, n\}$. {Initialize the weights for all n instances.}
2: Let k be the number of boosting rounds.
3: **for** $i = 1$ to k **do**
4: Create training set D_i by sampling (with replacement) from D according to w.
5: Train a base classifier C_i on D_i.
6: Apply C_i to all instances in the original training set, D.
7: $\epsilon_i = \frac{1}{n}\left[\sum_j w_j \, \delta\left(C_i(x_j) \neq y_j\right)\right]$ {Calculate the weighted error}
8: **if** $\epsilon_i > 0.5$ **then**
9: $\mathbf{w} = \{w_j = 1/n \mid j = 1, 2, \cdots, n\}$. {Reset the weights for all n instances.}
10: Go back to Step 4.
11: **end if**
12: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$.
13: Update the weight of each instance according to equation (5.88).
14: **end for**
15: $C^*(\mathbf{x}) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y)\big)$.

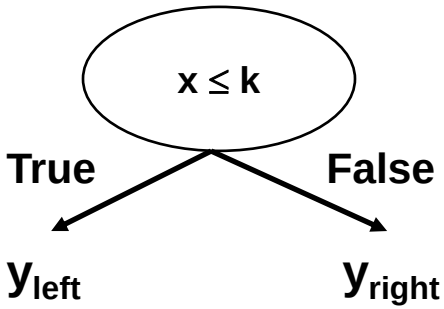
AdaBoost Example

- Consider 1-dimensional data set:

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Classifier is a decision stump
  - Decision rule: $x \leq k$ versus $x > k$
  - Split point k is chosen based on entropy



AdaBoost Example

- Training sets for the first 3 boosting rounds:

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

- Summary:

Round	Split Point	Left Class	Right Class	alpha
1	0.75	-1	1	1.738
2	0.05	1	1	2.7784
3	0.3	1	-1	4.1195

AdaBoost Example

- Weights

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

- Classification

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	-1	-1	-1	-1	-1	-1	-1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
Sum	5.16	5.16	5.16	-3.08	-3.08	-3.08	-3.08	0.397	0.397	0.397
Sign	1	1	1	-1	-1	-1	-1	1	1	1

Predicted Class

作业

杭电网络教学平台上

1. 注意截止时间
2. 需要提交（而非仅保存）

谢谢！

数据挖掘
教师：王东京
学院：计算机学院
邮箱：dongjing.wang@hdu.edu.cn