Machine Learning
机器学习

Lecture3：Linear classification
线性分类

李洁
nijanice@163.com

线性模型
Linear Model

easily understood and implemented, efficient and scalable

- Linear regression
- Linear classification

线性回归模型
Linear regression model

$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0$$

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

给西瓜打分

周志华. “机器学习” （西瓜书）

线性回归模型
Linear Regression Model

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1.2,\dots,N}$$

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots x_n^{(i)})^T$

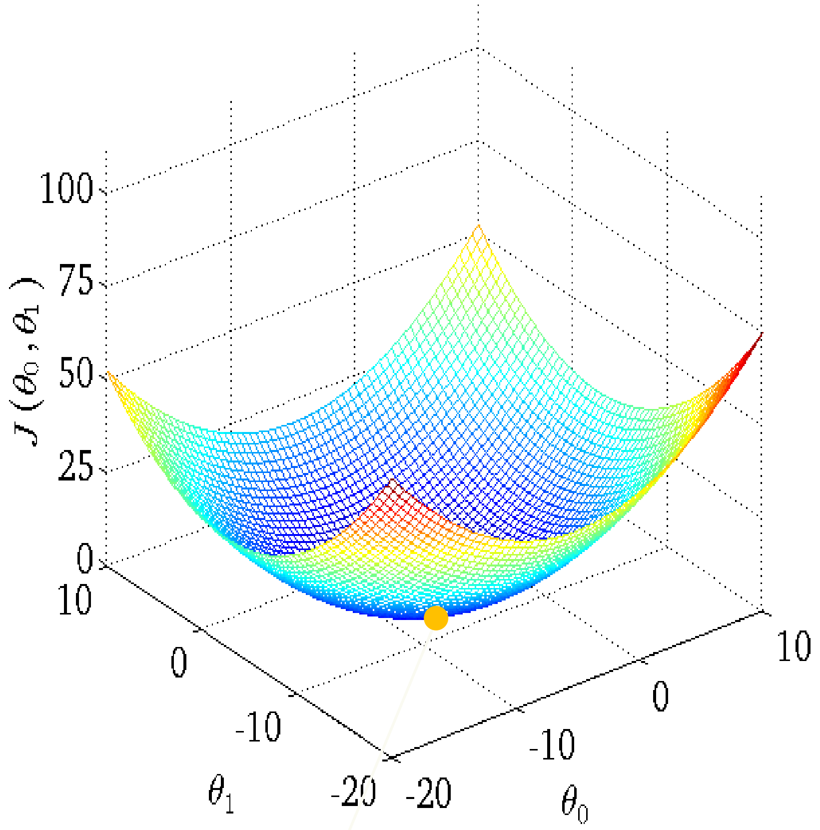
$y^{(i)}$ = label of i^{th} training example

let the machine learn a function from data to label

$$y \approx \boxed{f_\theta(x)} \Rightarrow \boxed{f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0}$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

凸函数
Bowled shape Convex Function

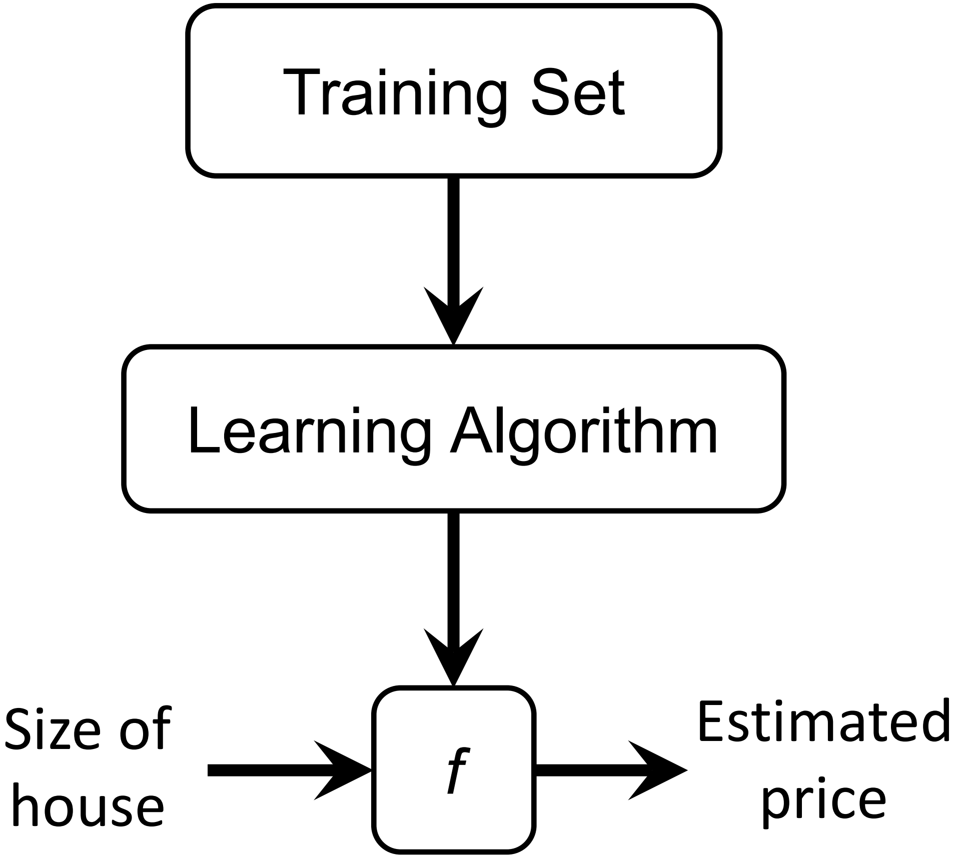


$$J(\theta_0, \theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta\left(x^{(i)}\right) - y^{(i)})^2$$

Different initial lead to the same optimum

Unique Minimum

单变量线性回归
Linear regression with one variable



- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

until we hopefully end up at a minimum

Hypothesis:

$$f_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

Goal:

$$\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$$

梯度下降法
Gradient descent algorithm

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j = 1$ and $j = 0$)

}

Linear Regression Model

$$f_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})^2$$

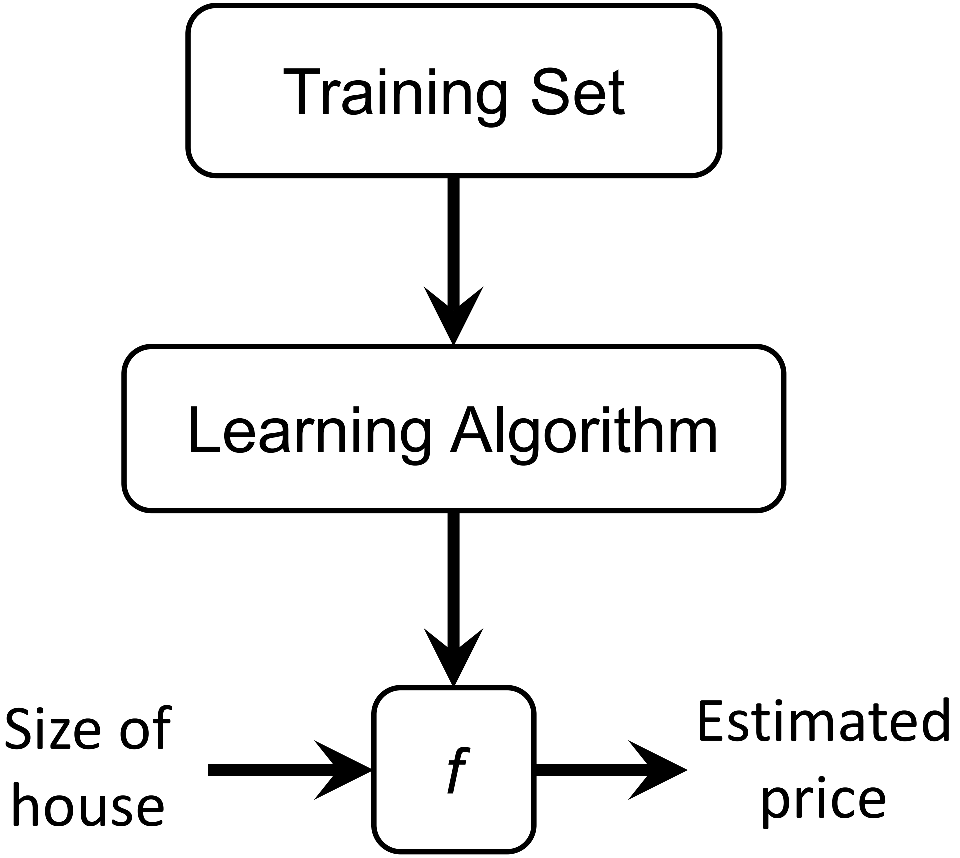
Repeat until convergece

$$\theta_0 := \theta_0 - a \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - a \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

update θ_0 and θ_1 simultaneously

单变量线性回归
Linear regression with one variable



- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Hypothesis:

$$f_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

Goal:

$$\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$$

多变量线性回归
Linear regression with multiple variable

Hypothesis: $f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function: $J(\theta_0, \theta_1, \ldots \theta_n) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

多变量线性回归
Linear regression with multiple variable

Hypothesis: $f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function: $J(\theta_0, \theta_1, \ldots \theta_n) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

线性模型
Linear Model

easily understood and implemented, efficient and scalable

- Linear regression

- Linear classification
  - Logistic regression
  - Linear discriminant analysis
  - Multi-class classification
  - Evaluation methods

线性回归模型
Linear regression model

好瓜？　坏瓜？

周志华. “机器学习” （西瓜书）

线性回归模型
Linear regression model

$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0$$

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

给西瓜打分

周志华. “机器学习” （西瓜书）

线性回归模型
Linear regression model

好瓜?　坏瓜?

Binary class labels:

好瓜：　1　　positive class (正类)

坏瓜：　0 or -1　　negative class (负类)

线性模型举例
Linear model example

$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0$$

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

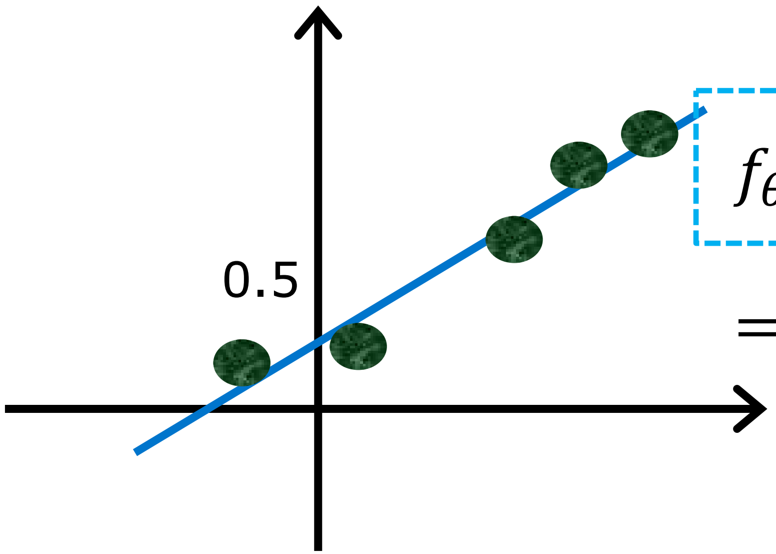
给西瓜打分

好瓜?　坏瓜?

How to relate
the output of a linear regression model to classification labels?

如何用于分类
Classification task

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)

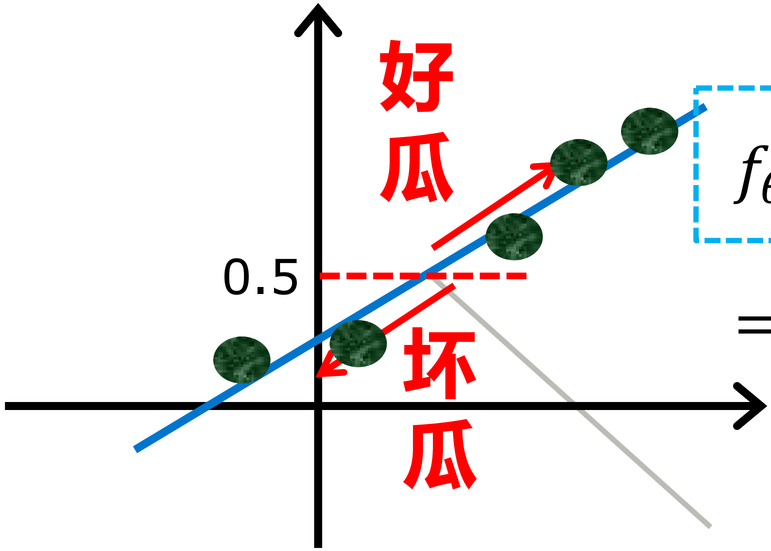


$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0$$

$$= \theta^T x$$ (suppose $x_0 = 1$)

如何用于分类
Classification task

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)



$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0$$

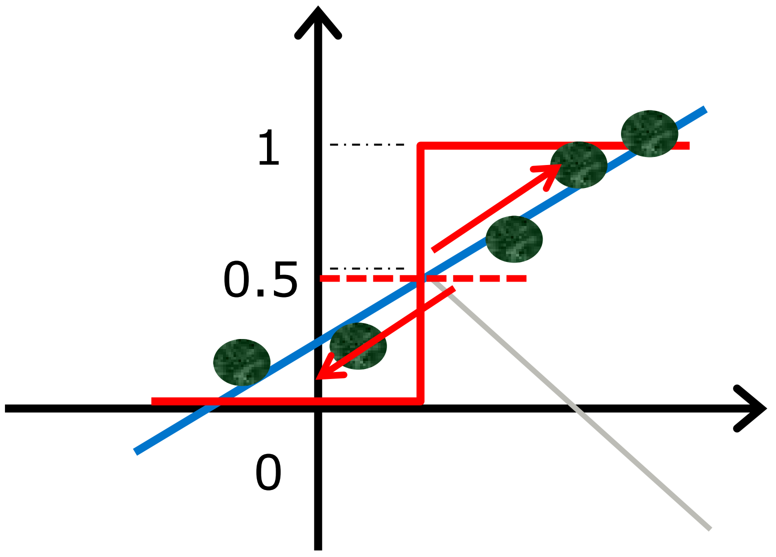
$$= \theta^T x \qquad \text{(suppose } x_0 = 1 \text{)}$$

If $\theta^T x \geq 0.5$, predict "y = 1"

If $\theta^T x < 0.5$, predict "y = 0"

如何用于分类
Classification task

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)



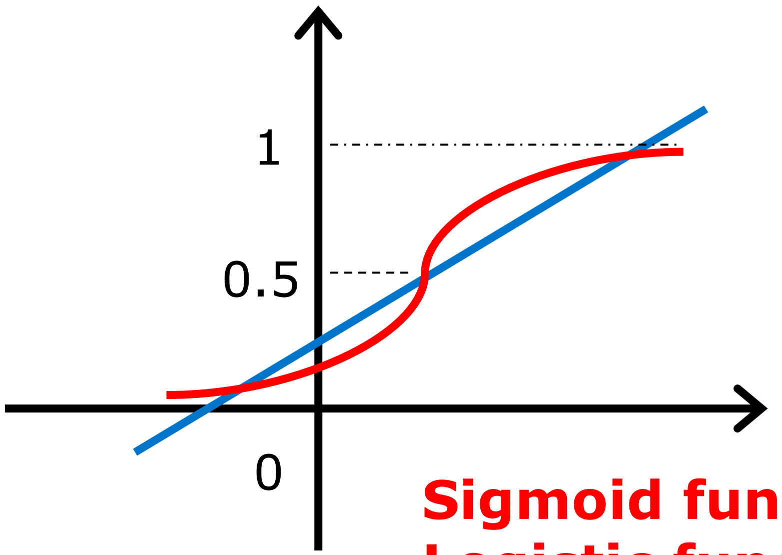
$$f_\theta(x) = \begin{cases} 1 & \theta^T x \geq 0.5 \\ 0 & \theta^T x < 0.5 \end{cases}$$

If $\theta^T x \geq 0.5$, predict "y = 1"

If $\theta^T x < 0.5$, predict "y = 0"

如何用于分类
Classification task

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)

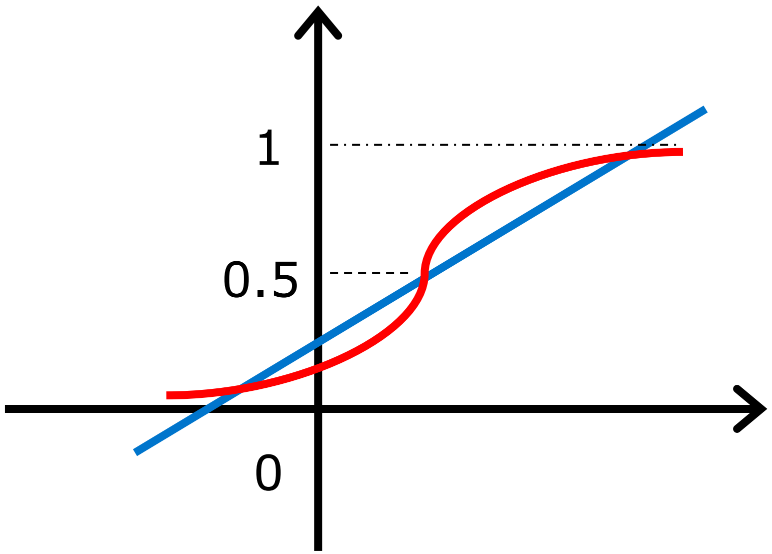


Sigmoid function
Logistic function

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

输出的解释
Interpretation of Hypothesis Output

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)



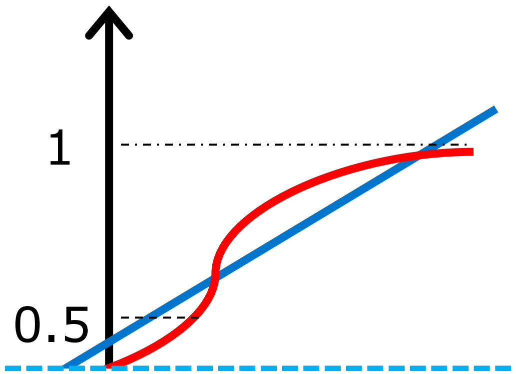
$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$0 < f < 1$

estimated probability that y = 1, given x, parameterized by θ

输出的解释
Interpretation of Hypothesis Output

$y \in \{0, 1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)



$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$P(y = 1/x) = \frac{1}{1 + e^{-\theta^T x}}$$

estimated probability that y = 1, given x, parameterized by θ

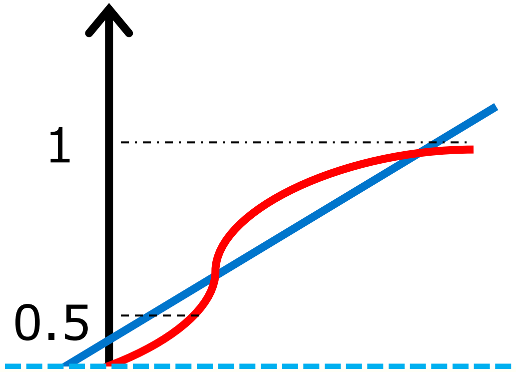
$$P(y = 0/x) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}$$

estimated probability that y = 0, given x, parameterized by θ

输出的解释
Interpretation of Hypothesis Output

$y \in \{0,1\}$ 1: "Positive Class" (好瓜)
0: "Negative Class" (坏瓜)

$f_\theta(x) \geq 0.5$



$$f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$P(y=1/x) = \frac{1}{1+e^{-\theta^T x}}$$

estimated probability that y = 1, given x, parameterized by θ

$$P(y=0/x) = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

estimated probability that y = 0, given x, parameterized by θ

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

$$\mathrm{Log}\left(\frac{P(y=1/x)}{P(y=0/x)}\right)$$

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

$$Log\left(\frac{P(y=1/x)}{P(y=0/x)}\right)$$

?

$$P(y=1/x)=\frac{1}{1+e^{-\theta^T x}}$$

$$P(y=0/x)=\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

$$\mathrm{Log}\left(\frac{P(y=1/x)}{P(y=0/x)}\right)$$

$$= Log\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}}$$

$$= \theta^T x$$

$$P(y=1/x) = \frac{1}{1+e^{-\theta^T x}}$$

$$P(y=0/x) = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

$$\mathrm{Log}\left(\frac{P(y=1/x)}{P(y=0/x)}\right)$$

$$= Log\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}}$$

$$= \theta^T x$$

$$P(y=1/x) = \frac{1}{1+e^{-\theta^T x}}$$

$$P(y=0/x) = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

对数几率回归
Logistic regression

对数几率就是指取该事件发生的概率与不发生的概率的比值的对数

用线性回归模型的预测结果去逼近真实标记的对数几率

$$\mathrm{Log}\left(\frac{P(y=1/x)}{P(y=0/x)}\right)$$

$$= Log\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}}$$

$$= \theta^T x$$

$$P(y=1/x) = \frac{1}{1+e^{-\theta^T x}}$$

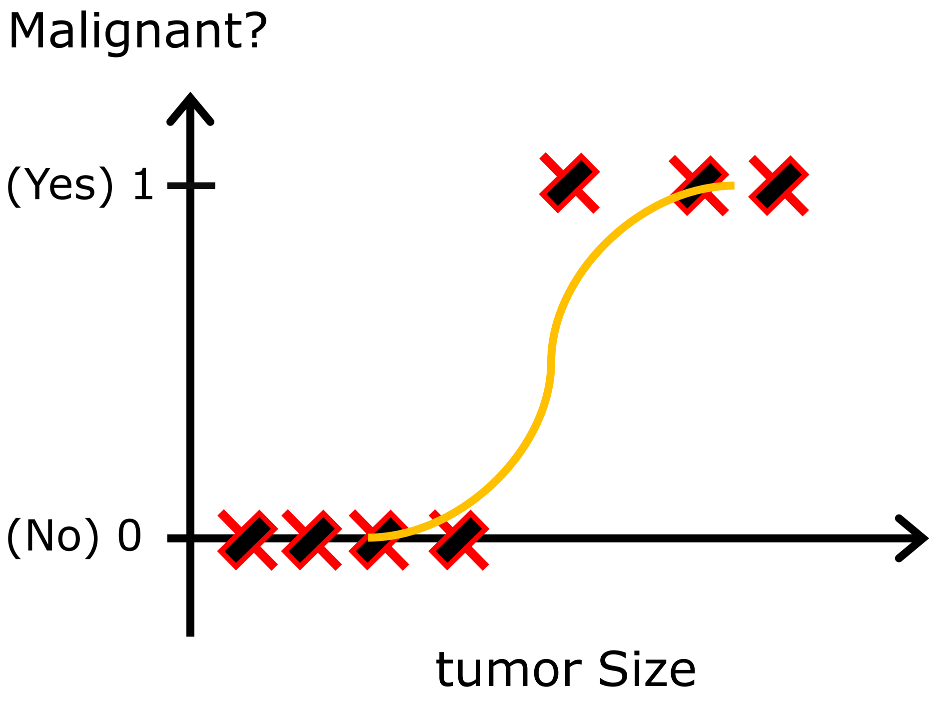
$$P(y=0/x) = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

输出的解释
Interpretation of Hypothesis Output

Example:

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$



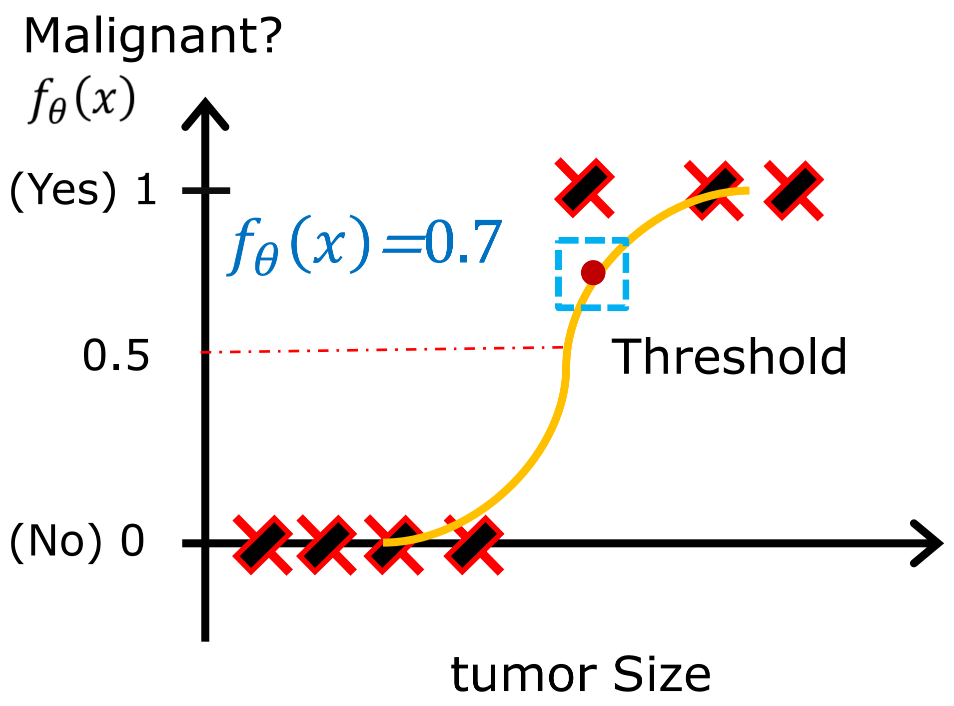
输出的解释
Interpretation of Hypothesis Output

Example:

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

?

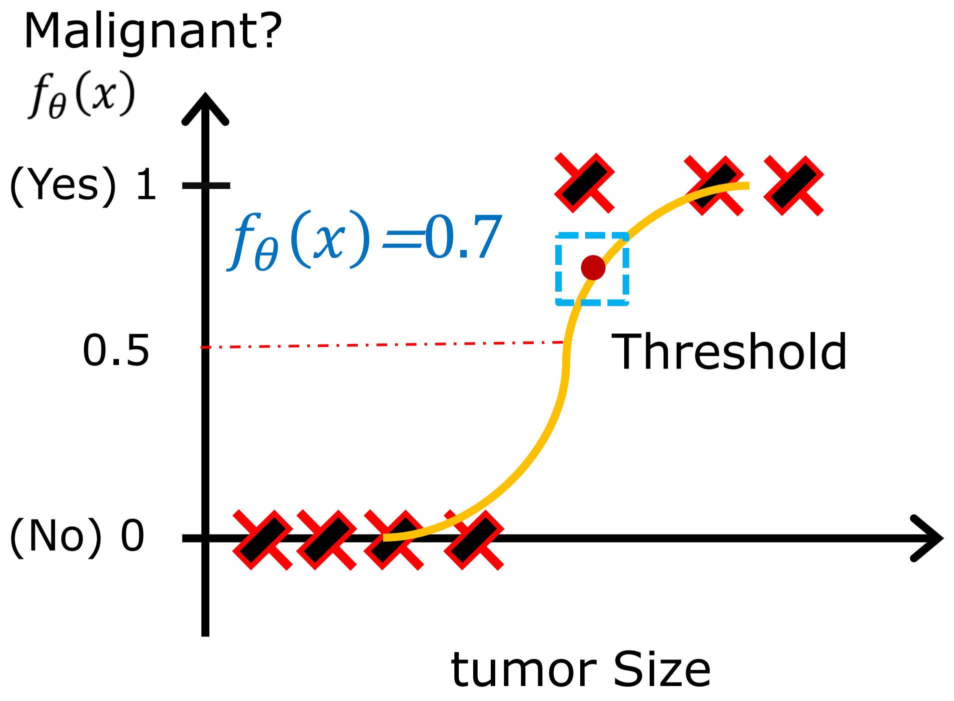


输出的解释
Interpretation of Hypothesis Output

Example:

$$f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$



- Tell patient that 70% chance of tumor being malignant
- Threshold classifier output $f_\theta(x)$ at 0.5, predict y = 1

监督学习
Supervised Learning

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1.2,\ldots,N}$$

$x^{(i)}$= input data(features) of i^{th} training example
$y^{(i)}$= output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx f_\theta(x^{(i)})$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

1. What is the learning objective?
2. How to update the parameters?

监督学习
Supervised Learning

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1.2,\ldots,N}$$

$x^{(i)}$= input data(features) of i^{th} training example
$y^{(i)}$= output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx f_\theta(x^{(i)})$$

⇨

$$f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

1. What is the learning objective?
2. How to update the parameters?

问题1：学习目标
What is the Learning Objective?

- Make the prediction closed to the corresponding label

$$\min_{\theta} \quad \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)}))$$

(Empirical Risk Minimization, ERM)

Loss functior $L(y^{(i)}, f_\theta(x^{(i)}))$ measures the error between the label and prediction for single sample.

The definition of loss function depends on the data and task

问题1：学习目标
What is the Learning Objective?

- Make the prediction closed to the corresponding label

Cost function

$$J(\theta) =_{\theta} \frac{1}{N}\sum_{i=1}^{N} L\left(y^{(i)}, f_\theta\left(x^{(i)}\right)\right)$$

Loss function $L(y^{(i)}, f_\theta(x^{(i)}))$ measures the error between the label and prediction for single sample.

Squared loss ?

$$Loss(y^{(i)}, f_\theta(x^{(i)})) = (y^{(i)} - f_\theta(x^{(i)}))^2$$

?

线性回归模型
Linear Regression Model

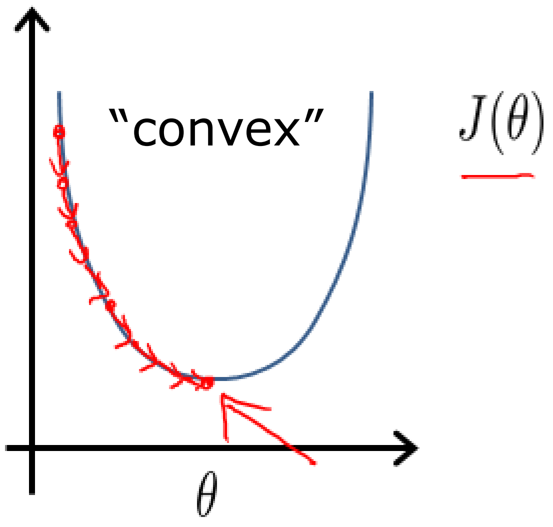
Linear regression:

Loss function $$L(y^{(i)}, f_\theta(x^{(i)})) = (y^{(i)} - f_\theta(x^{(i)}))^2$$

Cost function $$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

$$f_\theta(x) = \theta^T x$$

Gradient descent:



$$\theta_j := \theta_j - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

线性回归模型
Linear Regression Model

Linear regression:

Loss function $$L(y^{(i)}, f_\theta(x^{(i)})) = (y^{(i)} - f_\theta(x^{(i)}))^2$$

Cost function $$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Gradient descent:

线性回归模型
Linear Regression Model

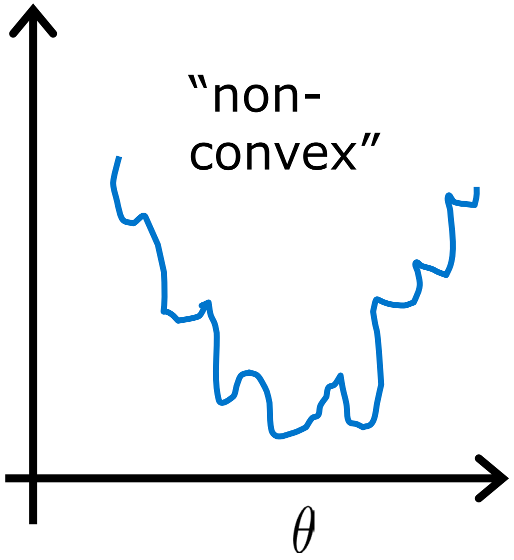
Linear regression:

Loss function $$L(y^{(i)}, f_\theta(x^{(i)})) = (y^{(i)} - f_\theta(x^{(i)}))^2$$

Cost function $$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

$$\boxed{f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}}$$

Gradient descent:



There are many local minimum that affect the efficiency of gradient descent

损失函数
Loss function

0-1 Loss Function

$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \begin{cases} 1, & \text{if } y^{(i)} \neq f\left(x^{(i)}\right) \\ 0, & \text{if } y^{(i)} = f\left(x^{(i)}\right) \end{cases}$$

Mean Squared Error, MSE

$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \left(y^{(i)} - f\left(x^{(i)}\right)\right)^2$$

Absolute Loss Function

$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \left|y^{(i)} - f\left(x^{(i)}\right)\right|$$

Logarithmic Loss Function (Cross-Entropy Loss Function)

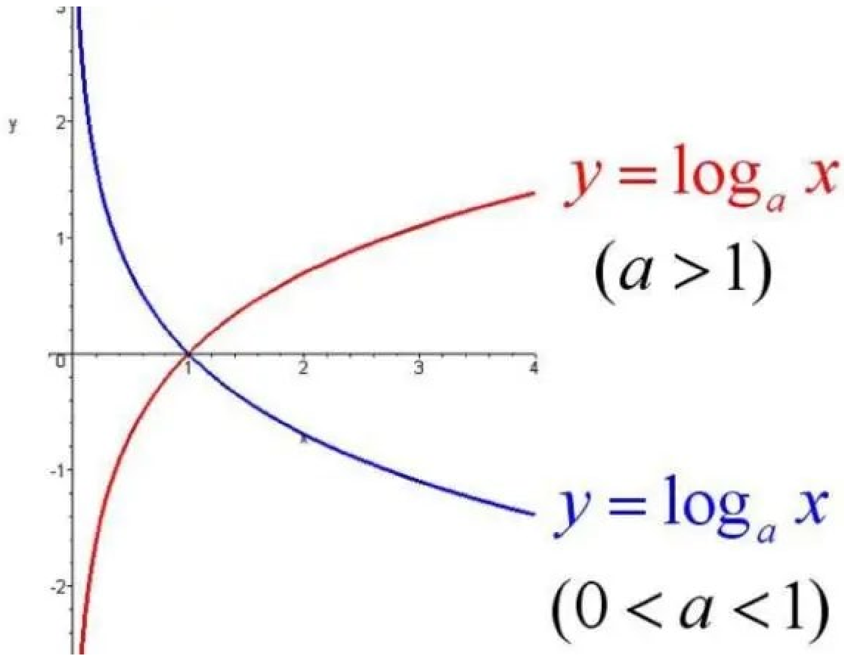
$$L\left(y^{(i)}, p^{(i)}\right) = -\left[y^{(i)} \log\left(p^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - p^{(i)}\right)\ \right]$$

$y^{(i)} \in \{0,1\}$, $p^{(i)} = f\left(x^{(i)}\right)$ is the predicted probability that the i th sample belongs to the positive class (usually denoted as class 1))

逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

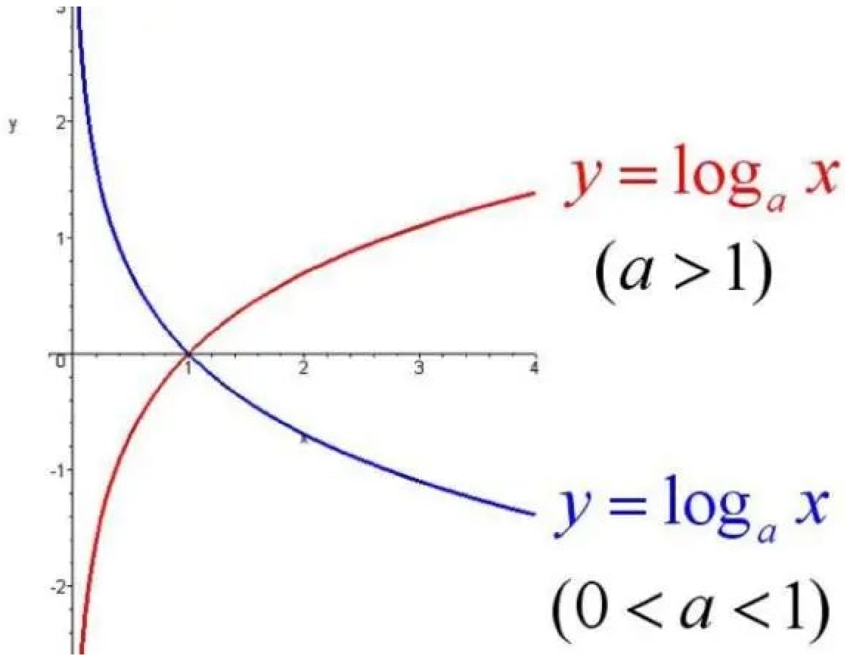
$$\mathbf{0} < f_\theta(x) = \frac{1}{1+e^{-\theta^T x}} < \mathbf{1}$$

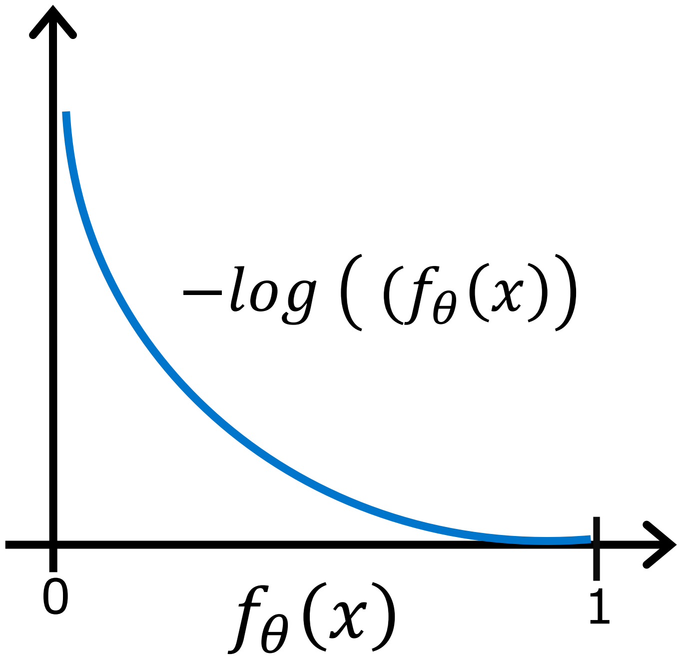


逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

$$\mathbf{0} < f_\theta(x) = \frac{1}{1+e^{-\theta^T x}} < \mathbf{1}$$

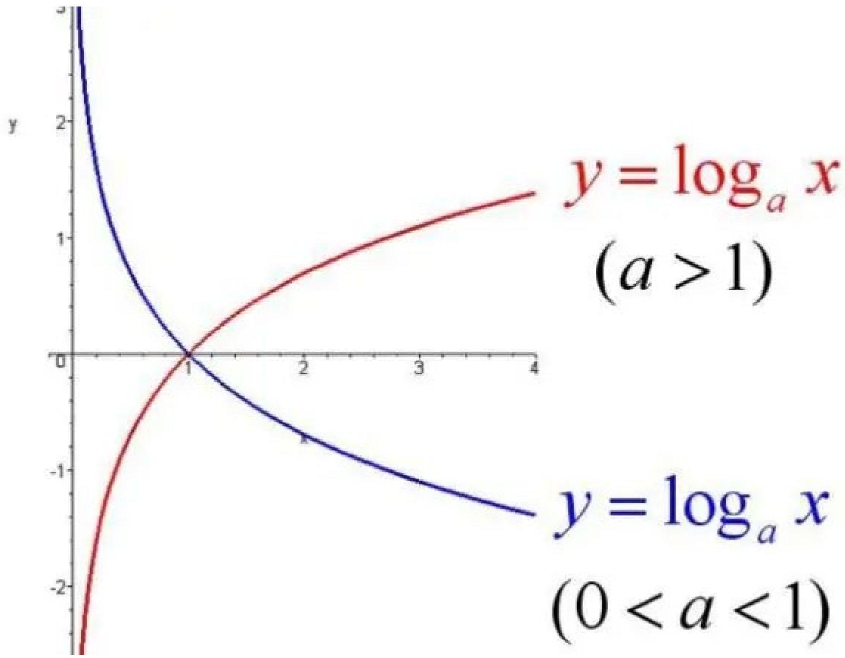


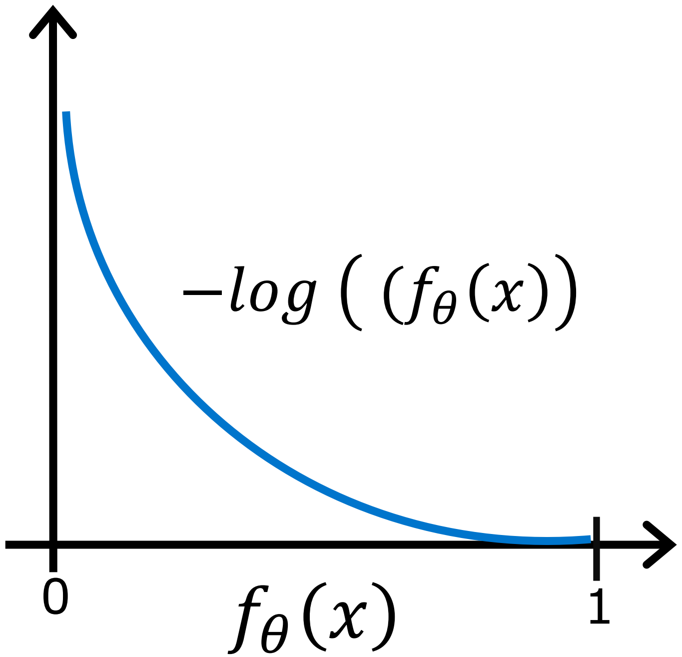


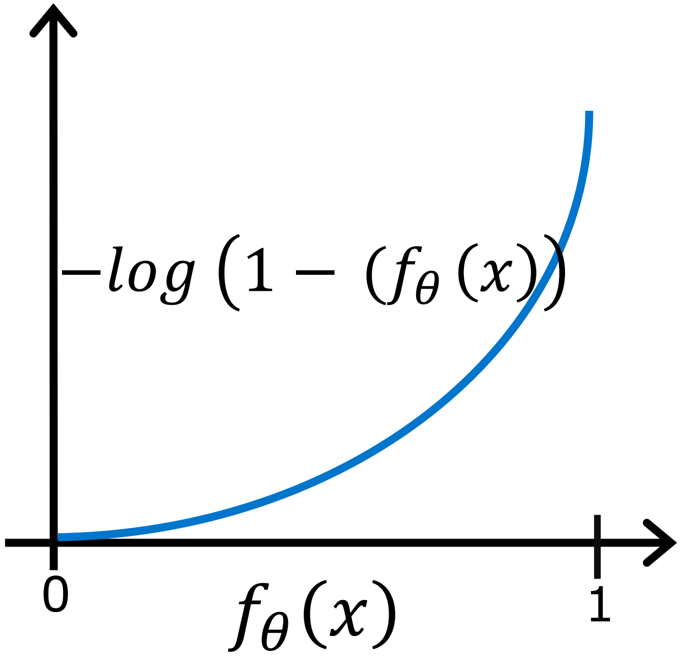
逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)) \right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)) \right) & if \quad y = 0 \end{cases}$$

$$\mathbf{0} < f_\theta(x) = \frac{1}{1+e^{-\theta^T x}} < \mathbf{1}$$







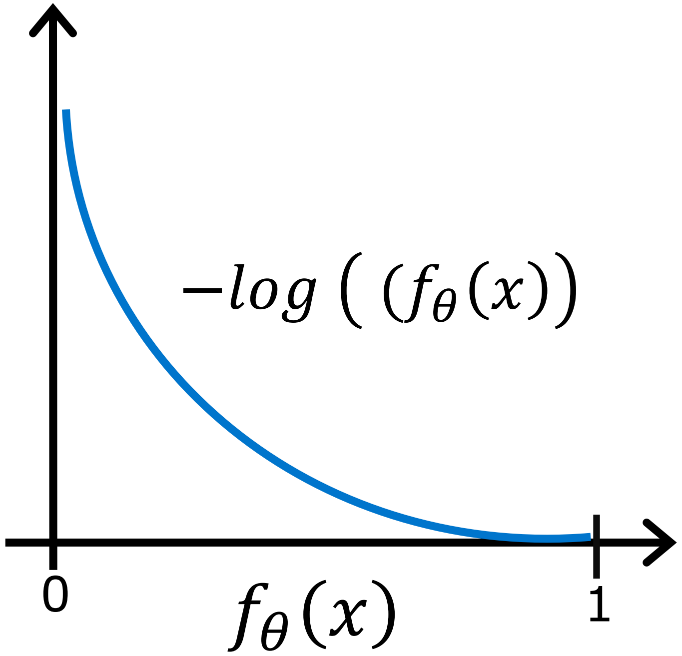
逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left(\ (f_\theta(x)\right) & if \quad y = 1 \\ -log\left(\ 1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} \boxed{-log\left((f_\theta(x)\right) \quad if \quad y = 1} \\ -log\left(1 - (f_\theta(x)\right) \quad if \quad y = 0 \end{cases}$$

Suppose y = 1



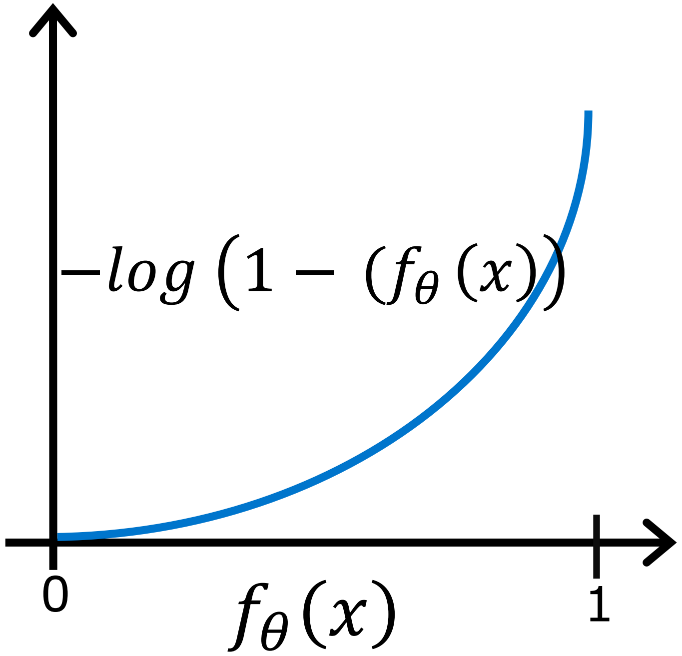
$$if \quad f_\theta(x) \to 1 \, , Loss(f_\theta(x), y) \to 0$$

$$if \quad f_\theta(x) \to 0, Loss(f_\theta(x), y) \to \infty$$

逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)) \right) & if \quad y = 1 \\ \boxed{-log\left(1 - (f_\theta(x)) \right) \quad if \quad y = 0} \end{cases}$$

Suppose y = 0



$$if \quad f_\theta(x) \to 0, Loss(f_\theta(x), y) \to 0$$

$$if \quad f_\theta(x) \to 1 , Loss(f_\theta(x), y) \to \infty$$

逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

$$= -ylog\left((f_\theta(x)\right) \; - (1 - y)\, log\left(1 - (f_\theta(x)\right)$$

(note： y=0 or 1 always)

逻辑斯蒂回归中损失函数
Loss function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

$$= \boxed{-ylog\left((f_\theta(x)\right) \; - (1-y)\, log\left(1 - (f_\theta(x)\right)}$$

(note： y=0 or 1 always)

Cross-entropy loss function

逻辑斯蒂回归中代价函数
Cost function in Logistic regression

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)) \right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)) \right) & if \quad y = 0 \end{cases}$$

$$= -ylog\left((f_\theta(x)) \right) - (1-y)\, log\left(1 - (f_\theta(x)) \right)$$

Cost Function： $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)}))$

$$= \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} -y^{(i)}log\left((f_\theta(x^{(i)})) \right) \\ -(1-y^{(i)})\, log\left(1 - (f_\theta(x^{(i)})) \right) \end{bmatrix}$$

学习目标
Learning Objective

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)}))$$

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1-(f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

$$= -ylog\left((f_\theta(x)\right) \; - (1-y)\, log\left(1-(f_\theta(x)\right)$$

(note： y=0 or 1 always)

Goal:

$$\min_\theta J(\theta) = \min_\theta \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)} log\left((f_\theta(x^{(i)})\right) \\ -(1-y^{(i)})\, log\left(1-(f_\theta(x^{(i)})\right) \end{array}\right]$$

逻辑斯蒂回归求解
Logistic regression solution

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)}log\left(\left(f_\theta\left(x^{(i)}\right)\right)\right) \\ -\left(1-y^{(i)}\right)log\left(1-\left(f_\theta\left(x^{(i)}\right)\right)\right) \end{array}\right]$$

To fit parameters θ :

$$\min_\theta J(\theta)$$

To make a prediction given new x :

Output $$f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

逻辑斯蒂回归求解
Logistic regression solution

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} -y^{(i)} log\left((f_\theta(x^{(i)}) \right) \\ -(1-y^{(i)})\, log\left(1-(f_\theta(x^{(i)}) \right) \end{bmatrix}$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all θ_j)

$\}$

逻辑斯蒂回归梯度推导

Derivation of logistic gradient

$$
\mathrm{J}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} -y^{(i)} log\left(\left(f_\theta\left(x^{(i)}\right)\right)\right) \\ -\left(1-y^{(i)}\right) log\left(1-\left(f_\theta\left(x^{(i)}\right)\right)\right) \end{bmatrix}
$$

$$\theta$$

$$
\frac{\partial \mathrm{J}(\theta)}{\partial \theta_j} = ?
$$

$$
f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}
$$

$$
\frac{dx^T A}{dx} = \mathrm{A}
$$

逻辑斯蒂回归梯度推导
Derivation of logistic gradient

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\boxed{\begin{matrix} -y^{(i)} log\left((f_\theta(x^{(i)})\right) \\ -(1-y^{(i)})\, log\left(1-(f_\theta(x^{(i)})\right)\end{matrix}}\right]$$

$$y^{(i)} lo\, g\left(\frac{1}{1+e^{-\theta^\top x^{(i)}}}\right) + (1-y^{(i)}) lo\, g\left(1-\frac{1}{1+e^{-\theta^\top x^{(i)}}}\right)$$

$$= -y^{(i)} lo\, g\left(1+e^{-\theta^\top x^{(i)}}\right) - (1-y^{(i)}) lo\, g\left(1+e^{\theta^\top x^{(i)}}\right)$$

逻辑斯蒂回归梯度推导
Logistic gradient derivation

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{\partial}{\partial\theta_j}\left(-\frac{1}{N}\sum_{i=1}^{N}\left(-y^{(i)}\log\left(1+e^{-\theta^\top x^{(i)}}\right)-\left(1-y^{(i)}\right)\log\left(1+e^{\theta^\top x^{(i)}}\right)\right)\right)$$

$$=-\frac{1}{N}\sum_{i=1}^{N}\left(-y^{(i)}\frac{-x_j^{(i)}e^{-\theta^\top x^{(i)}}}{1+e^{-\theta^\top x^{(i)}}}-\left(1-y^{(i)}\right)\frac{x_j^{(i)}e^{\theta^\top x(i)}}{1+e^{\theta^\top x(i)}}\right)$$

$$=-\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)}-f\left(x^{(i)}\right)\right)x_j^{(i)}$$

$$=\frac{1}{N}\sum_{i=1}^{N}\left(f\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)}$$

逻辑斯蒂回归求解
Logistic regression solution

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)} log\left((f_\theta(x^{(i)})\right) \\ -(1-y^{(i)})\, log\left(1-(f_\theta(x^{(i)})\right)\end{array}\right]$$

Want $\{ \min_\theta J(\theta)$:

Repeat

$$\theta_j := \theta_j - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

逻辑斯蒂回归中损失函数和代价函数
Loss function & Cost function

Cost Function： $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)}))$

$$Loss(f_\theta(x), y) = \begin{cases} -log\left((f_\theta(x)\right) & if \quad y = 1 \\ -log\left(1 - (f_\theta(x)\right) & if \quad y = 0 \end{cases}$$

$$= -ylog\left((f_\theta(x)\right) - (1-y)\, log\left(1 - (f_\theta(x)\right)$$

(note： y=0 or 1 always)

Cross-entropy loss function

$$\min_\theta J(\theta) = \min_\theta \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)}log\left((f_\theta(x^{(i)})\right) \\ -(1-y^{(i)})\, log\left(1 - (f_\theta(x^{(i)})\right) \end{array} \right]$$

(also can be acquired by Maximum likelihood method)

极大似然法
Maximum likelihood method

Assume the data is generated based on :

$$f_\theta\left(x^{(i)}\right) = P(y^{(i)}|x^{(i)})$$

The probability of generating all the training data:

$$= \prod_{i=1}^{N} (f_\theta\left(x^{(i)}\right)^{y^{(i)}} (1 - f_\theta\left(x^{(i)}\right)^{1-y^{(i)}})$$

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

estimated probability that y = 1, given x

极大似然法
Maximum likelihood method

Assume the data is generated based on :

$$f_\theta\left(x^{(i)}\right) = P(y^{(i)} \backslash x^{(i)})$$

The probability of generating all the training data:

$$= \prod_{i=1}^{N} \left(f_\theta\left(x^{(i)}\right)^{y^{(i)}} \left(1 - f_\theta\left(x^{(i)}\right)^{1-y^{(i)}} \right) \right.$$

Log Likelihood function:

$$\log L(\theta) = \sum_{i=1}^{N} \left[y^{(i)} log \left(\left(f_\theta\left(x^{(i)}\right)\right) \right) + (1 - y^{(i)})\, log \left(1 - \left(f_\theta\left(x^{(i)}\right)\right) \right) \right]$$

极大似然法
Maximum likelihood method

Assume the data is generated based on :

$$f_\theta\left(x^{(i)}\right) = P(y^{(i)} \backslash x^{(i)})$$

The probability of generating all the training data:

$$= \prod_{i=1}^{N} \left(f_\theta\left(x^{(i)}\right)^{y^{(i)}} \left(1 - f_\theta\left(x^{(i)}\right) \right)^{1-y^{(i)}} \right)$$

Log Likelihood function:

$$\sum_{i=1}^{N} \left[y^{(i)} log \left(\left(f_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \, log \left(1 - \left(f_\theta\left(x^{(i)}\right)\right)\right) \right]$$

maximize log-likelihood= minimize negative log-likelihood

$$\min_\theta \sum_{i=1}^{N} -\left[y^{(i)} log \left(\left(f_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \, log \left(1 - \left(f_\theta\left(x^{(i)}\right)\right)\right) \right]$$

逻辑斯蒂回归中损失函数和代价函数
Loss function & Cost function

Cost Function： $J(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)}))$

$$Loss(f_\theta(x), y) = \begin{cases} -log\,((f_\theta(x)) & if \quad y = 1 \\ -log\,(1 - (f_\theta(x)) & if \quad y = 0 \end{cases}$$

$$= \boxed{-ylog\,((f_\theta(x)) \; - (1-y)\,log\,(1 - (f_\theta(x))}$$

(note： y=0 or 1 always)

Cross-entropy loss function

$$\min_\theta J(\theta) = \min_\theta \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)}log\,((f_\theta(x^{(i)})) \\ -(1-y^{(i)})\,log\,(1 - (f_\theta(x^{(i)})) \end{array} \right]$$

(also can be acquired by Maximum likelihood method)

线性模型
Linear Model

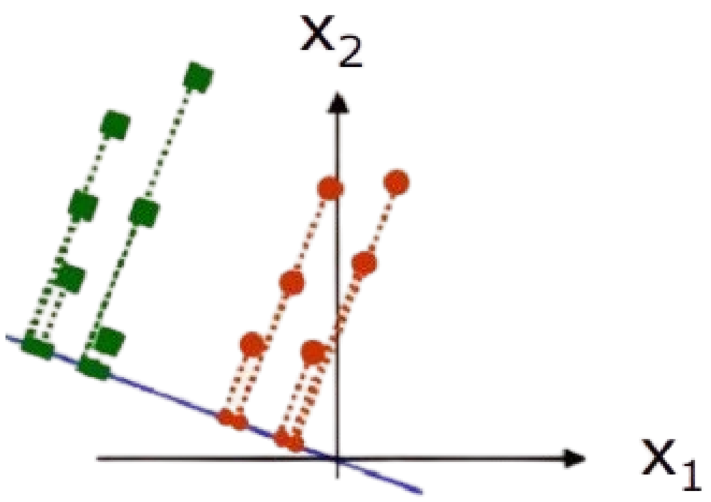
easily understood and implemented, efficient and scalable

- Linear regression

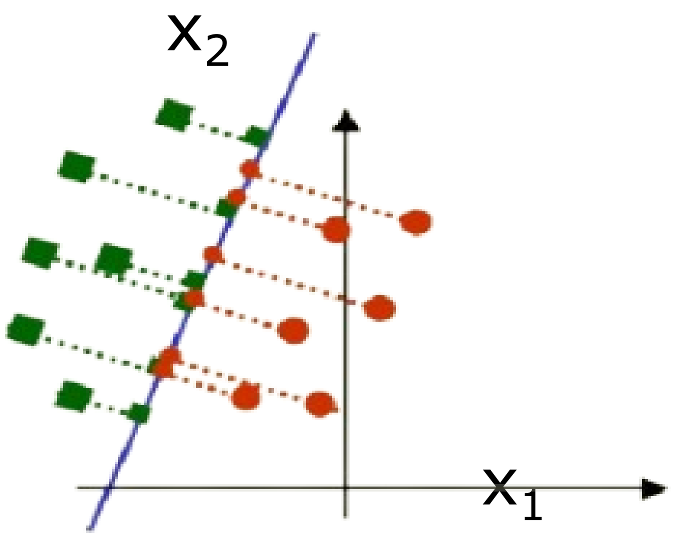
- Linear classification
  - Logistic regression
  - Linear discriminant analysis
  - Multi-class classification
  - Evaluation methods

线性判别分析
Linear Discriminant Analysis

Seeks to find directions along which the classes are best separated.



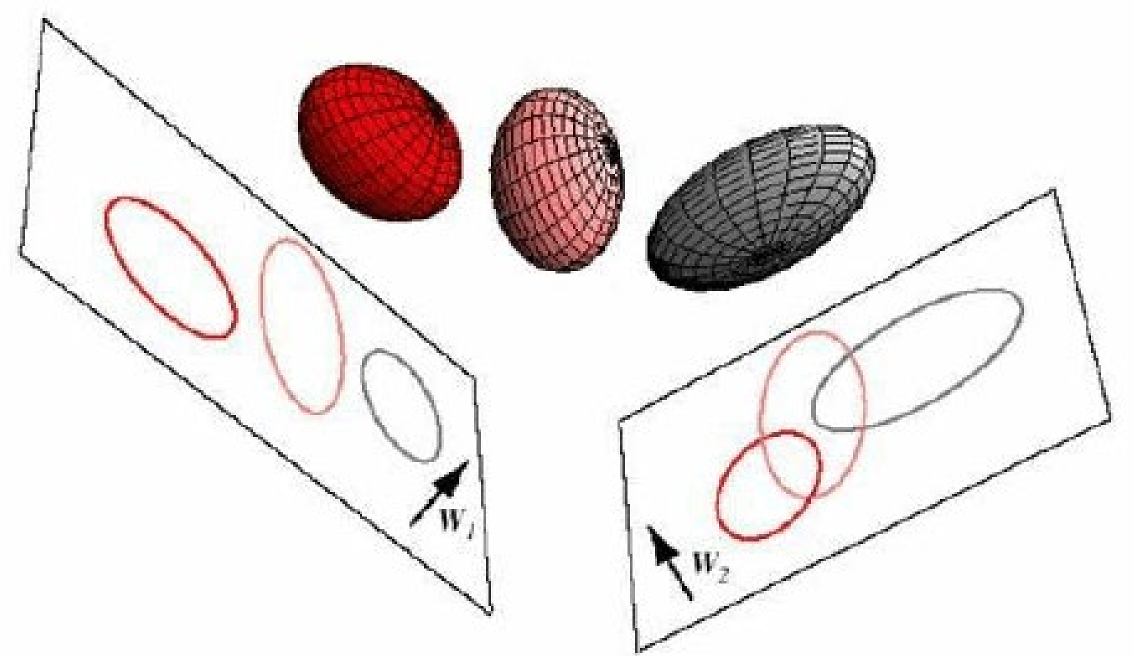
Good line to project to,
classes are well separated



Bad line to project to,
classes are mixed up

线性判别分析
Linear Discriminant Analysis

Seeks to find directions along which the classes are best separated.

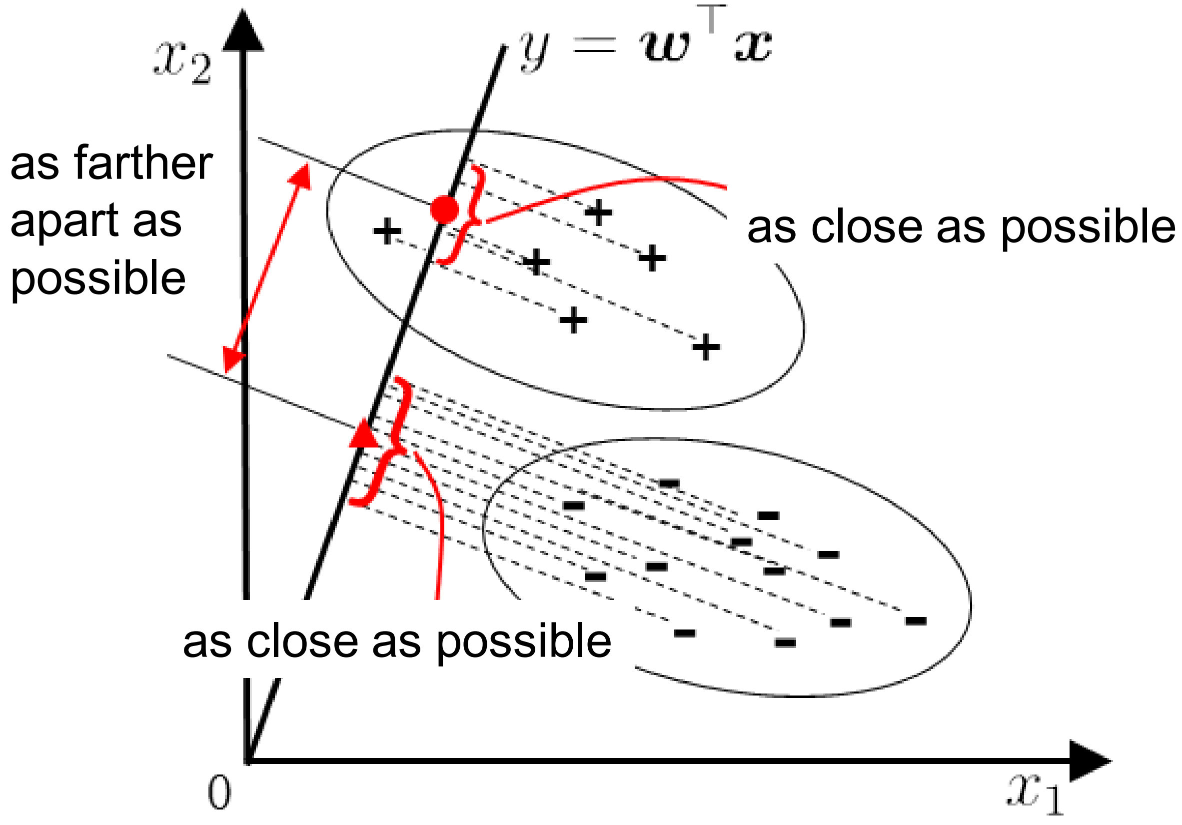


Good subspace to project to, classes are well separated

Bad subspace to project to, classes are mixed up

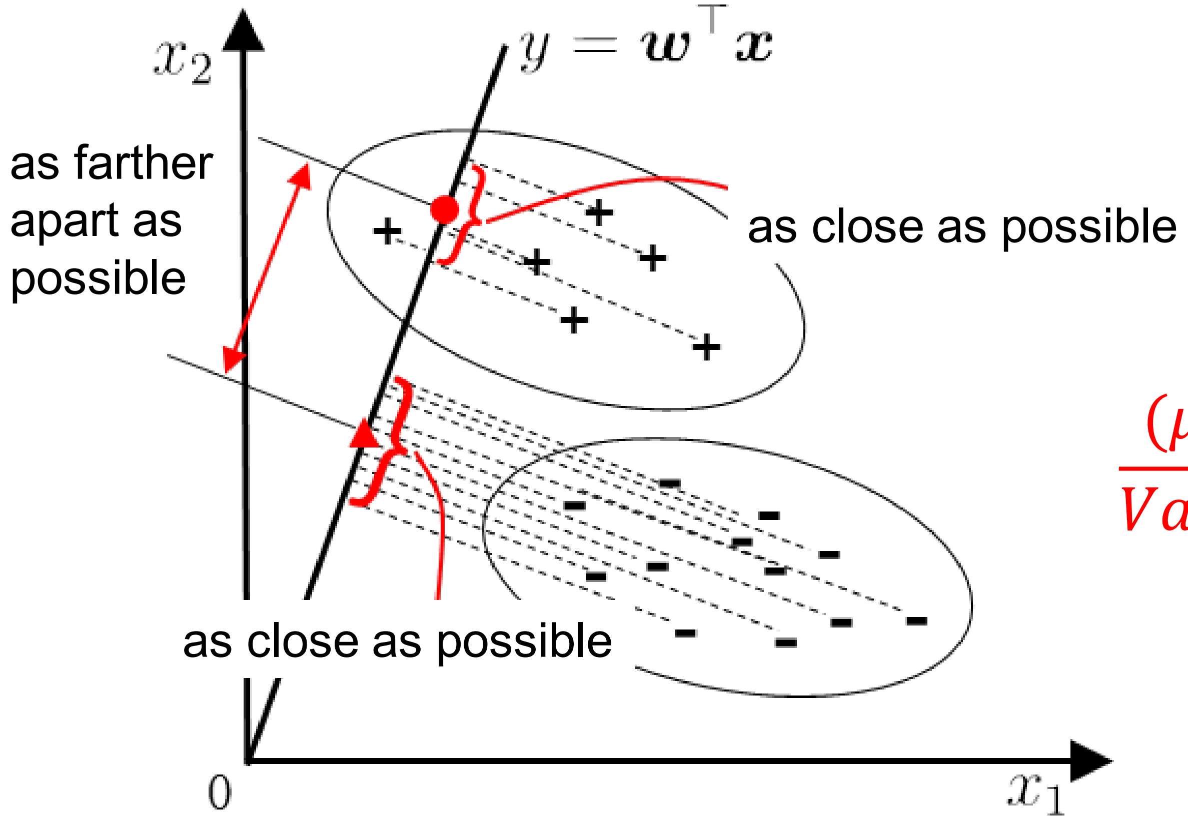
线性判别分析
Linear Discriminant Analysis

Looks for a projection where examples from the same class are projected very close to each other and, at the same time , the projected means are as farther apart as possible



线性判别分析
Linear Discriminant Analysis

Looks for a projection where examples from the same class are projected very close to each other and, at the same time , the projected means are as farther apart as possible



$$\frac{(\mu_0' - \mu_1')^2}{Var_0' + Var_1'}$$

线性判别分析
Linear Discriminant Analysis

- w is the direction we want to find, such that the projections maximize class separability after the projection (the norm of w is set to 1).

- The mean for class 0:

$$\mu_0 = \frac{1}{N_0}\sum_{x\in X_0} x$$

$$\mu_0' = \frac{1}{N_0}\sum_{x\in X_0} w^T x = w^T\mu_0$$

- The mean for class 1:

$$\mu_1 = \frac{1}{N_1}\sum_{x\in X_1} w^T x$$

$$\mu_1' = \frac{1}{N_1}\sum_{x\in X_1} w^T x = w^T\mu_1$$

- The variance for class 0:

$$\mathrm{Var}_0' = \frac{1}{N_0}\sum_{x\in X_0}(w^T x - w^T\mu_0)^2$$

$$= w^T\frac{1}{N_0}\sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T w = w^T\Sigma_0 w$$

- The variance for class 1:

$$\mathrm{Var}_1' = \frac{1}{N_1}\sum_{x\in X_1}(w^T x - w^T\mu_1)^2$$

$$= w^T\frac{1}{N_1}\sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T w = w^T\Sigma_1 w$$

$$\arg\max_{w} J(w) = \frac{(\mu_0' - \mu_1')^2}{\mathrm{Var}_0' + \mathrm{Var}_1'} = \frac{(w^T\mu_0 - w^T\mu_1)^2}{w^T\Sigma_0 w + w^T\Sigma_1 w} = \frac{\left\|w^T\mu_0 - w^T\mu_1\right\|^2}{w^T\Sigma_0 w + w^T\Sigma_1 w}$$

线性判别分析
Linear Discriminant Analysis

- generalized Rayleigh quotient

$$J = \frac{\left\| \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1 \right\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}} = \frac{\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1\right)\boldsymbol{w}} = \frac{w^T S_b w}{w^T S_w w}$$

- Between-class scatter matrix

$$\mathbf{S}_b = \left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)^{\mathrm{T}}$$

- Within-class scatter matrix

$$\begin{aligned}\mathbf{S}_w &= \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1 \\ &= \sum_{\boldsymbol{x}\in X_0}\left(\boldsymbol{x} - \boldsymbol{\mu}_0\right)\left(\boldsymbol{x} - \boldsymbol{\mu}_0\right)^{\mathrm{T}} + \sum_{\boldsymbol{x}\in X_1}\left(\boldsymbol{x} - \boldsymbol{\mu}_1\right)\left(\boldsymbol{x} - \boldsymbol{\mu}_1\right)^{\mathrm{T}}\end{aligned}$$

线性判别分析
Linear Discriminant Analysis

- To find the maximum of

$$J = \frac{\boldsymbol{w}^{\mathrm{T}} \mathbf{S}_b \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}} \mathbf{S}_w \boldsymbol{w}}$$

- Yields

$$w^* = \arg\max_w J(w) = \arg\max_w \left(\frac{w^T S_B w}{w^T S_W w} \right) = S_W^{-1} (\mu_1 - \mu_2)$$

- Usually employs SVD of $S_w = U \boldsymbol{\Sigma} V^T$ to get $S_W^{-1} = U \boldsymbol{\Sigma} V^T$

线性判别分析推导
Linear Discriminant Analysis derivation

- To find the maximum of

$$J = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

suppose

$$\max_{w} w^T S_b w$$
$$s.t.\ w^T S_w w = c,\ c \neq 0$$

lagrange method

$$L(w, \lambda) = w^T S_b w - \lambda(w^T S_w w - c)$$

$$\frac{\partial L(w,\lambda)}{\partial w} = (S_b + S_b^T)w - \lambda(S_w + S_w^T)w$$
$$= 2S_b w - 2\lambda S_w w = 0$$

$$S_w^{-1} S_b w = \lambda w$$

$$S_b w = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = \beta(\mu_1 - \mu_2)$$

$$w = \frac{\beta}{\lambda} S_w^{-1}(\mu_1 - \mu_2)$$

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

线性判别分析推导
Linear Discriminant Analysis derivation

- To find the maximum of

$$J = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

suppose

$$\max_w \; w^T S_b w$$
$$s.t.\; w^T S_w w = c,\; c \neq 0$$

lagrange method

$$L(w,\lambda) = w^T S_b w - \lambda(w^T S_w w - c)$$

$$\frac{\partial L(w,\lambda)}{\partial w} = (S_b + S_b^T)w - \lambda(S_w + S_w^T)w = 2S_b w - 2\lambda S_w w = 0$$

$$S_w^{-1} S_b w = \lambda w$$

$$\frac{dx^T A x}{dx} = (A + A^T)x$$

$$S_b w = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = \beta(\mu_1 - \mu_2)$$

$$w = \frac{\beta}{\lambda} S_w^{-1}(\mu_1 - \mu_2)$$

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

线性模型
Linear Model

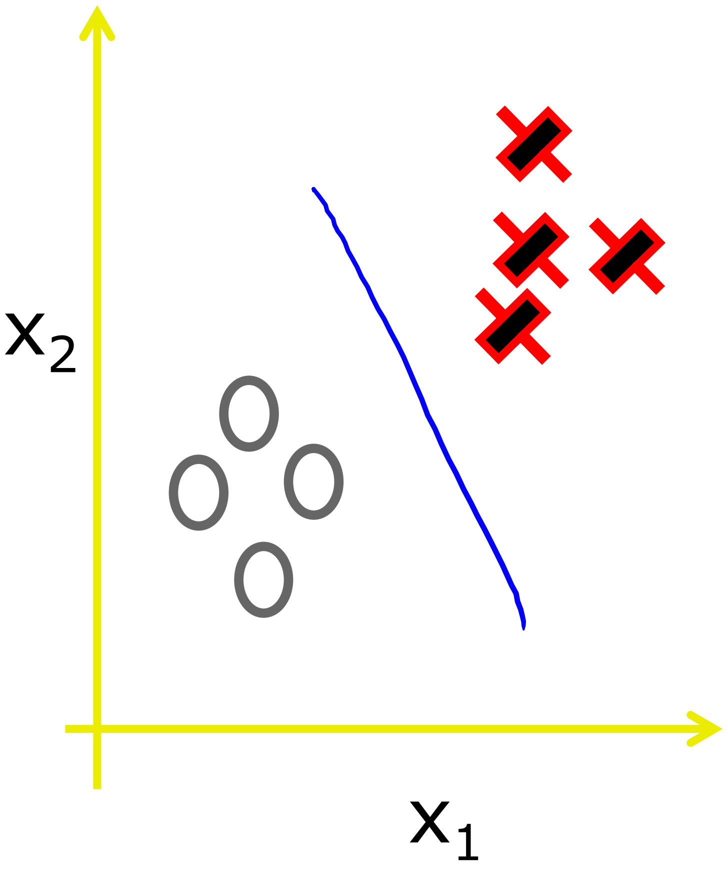
easily understood and implemented, efficient and scalable

- Linear regression

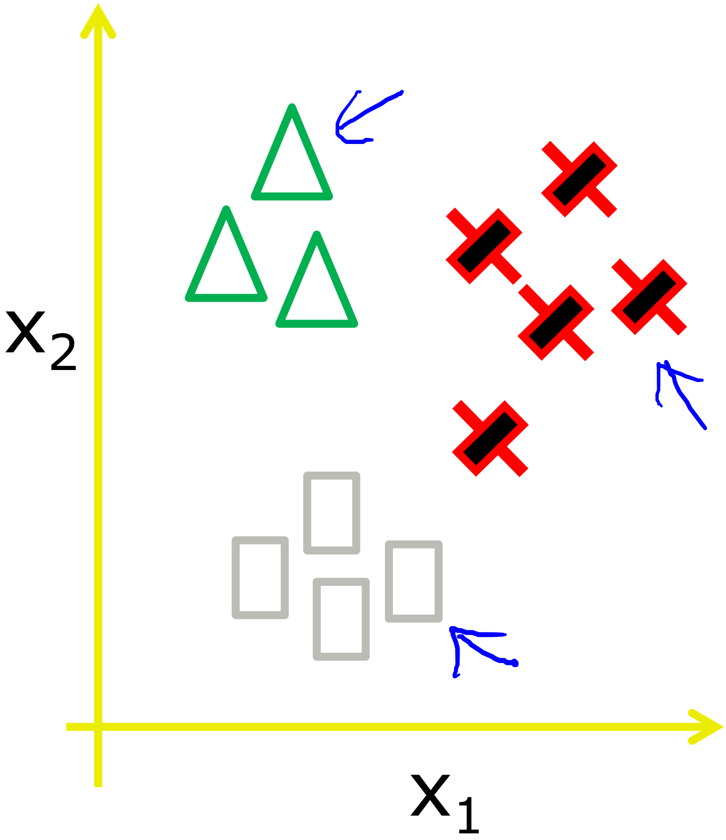
- Linear classification
  - Logistic regression
  - Linear discriminant analysis
  - Multi-class classification
  - Evaluation methods

多类分类
Multiclass classification

Binary classification:



Multi-class classification:



Email foldering/tagging: Work, Friends, Family
Medical diagrams: Not ill, Cold, Flu
Weather: Sunny, Cloudy, Rain, Snow

Softmax回归
Softmax Regression

- Input: $x = (x_1, x_2, \ldots x_n)^T$
- Output: class label $\in \{0,1,..K\}$ represented by one-hot vector

  $y = (I(1 = k)\ , I(2 = k) \ldots I(K = k))^T$

- For each class k, there is a weight vector θ_k
- For each class k, the predicted probability is : $p = \dfrac{e^{\theta_k^{\mathrm{T}} x}}{\sum_{\mathrm{j}=1}^{\mathrm{K}} e^{\theta_j^{\mathrm{T}} x}}$
- Using cross-entropy, the cost function:

  $-\sum_{i=1}^{N} \sum_{j=1}^{K} y_j^i \log\left(p_j^i\right)$

- Predicting label:

  $\hat{y} = arg \max_{\mathrm{j}=1\ldots K} p_j(x)$

Softmax回归
Softmax Regression

- Input: $x = (x_1, x_2, \dots x_n)^T$
- Output: class label $\in \{0,1,..K\}$ represented by one-hot vector

$$y = (I(1 = k)\ , I(2 = k) \dots I(K = k))^T$$

e.g. K=3,then $label1 = (1,0,0)^T$
$label2 = (0,1,0)^T$
$label3 = (0,0,1)^T$

- For each class k, there is a weight vector θ_k
- For each class k, the predicted probability is : $p = \frac{e^{\theta_k^{\mathrm{T}} x}}{\sum_{\mathrm{j=1}}^{\mathrm{K}} e^{\theta_j^{\mathrm{T}} x}}$
- Using cross-entropy, the cost function:

$$-\sum_{i=1}^{N} \sum_{j=1}^{K} y_j^i \log\left(p_j^i\right)$$

e.g. $y = (0,0,1)^T, \quad \mathrm{p} = (0.3,0.3,0.4)^T$

$\mathrm{loss} = -(0 * \log(0.3) + 0 * * \log(0.3) + 1 * \log(0.4))$
≈ 0.916

$y = (0,0,1)^T, \quad \mathrm{p} = (0.1,0.1,0.8)^T$

$\mathrm{loss} = -(0 * \log(0.1) + 0 * * \log(0.1) + 1 * \log(0.8))$
≈ 0.223

- Predicting label:

$$\hat{y} = arg \max_{\mathrm{j=1\dots}K} p_j(x)$$

Softmax回归
Softmax Regression

$$\arg\min_{\theta} \mathrm{J}(\theta) = -\sum_{i=1}^{N}\sum_{j=1}^{K} y_j^i \log\left(p_j^i\right)$$

Compute the gradient:

$$\frac{\partial \mathrm{J}(\theta)}{\partial \theta_k} = -\sum_{i=1}^{N}\sum_{j=1}^{K} y_j^i \frac{\partial \log\left(p_k^i\right)}{\partial \theta_k} = -\sum_{i=1}^{N} y_{\mathrm{k}}^i \frac{1}{p_k^i}\frac{\partial p_k^i}{\partial \theta_k}$$

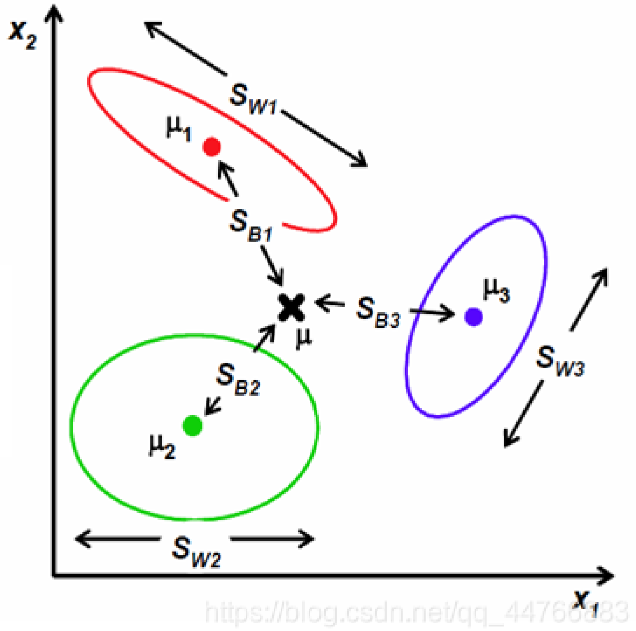
$$= -\sum_{i=1}^{N}\sum_{j=1}^{K} y_{\mathrm{j}}^i \frac{1}{p_k^i} p_k^i(\delta_{\mathrm{jk}}^i - p_k^i)x^{(i)} \quad = \sum_{i=1}^{N}(p_k^i - y_k^i)x^{(i)}$$

Update parameters: $\theta_k := \theta_{\mathrm{k}} - a\sum_{i=1}^{N}(p_k^i - y_k^i)x^{(i)}$

多类线性判别分析
Multi-Class LDA

- Between-class scatter matrix（ N_i denote the number of examples in each class）

$$S_b = \sum_{j=1}^{m} N_i(\mu_j - \mu)(\mu_j - \mu)^T$$



- Within-class scatter matrix

$$S_w = \sum_{j=1}^{m} S_{w_j} \quad , S_{w_j} = \sum_{x \in X_j} (x - \mu_j)(x - \mu_j)^T$$

- Global scatter matrix

$$S_t = S_b + S_w = \sum_{i=1}^{N} (x^i - \mu)(x^i - \mu)^T$$

多类线性判别分析
Multi-Class LDA

- To find a projection matrix W that maximizes the following ratio:

$$J = \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

- Solve the Generalized Eigenvalue $S_w^{-1} S_b W = W\Lambda$

  Λ is the diagonal matrix of eigenvalues (each eigenvalue represents the ratio of between-class scatter to within-class scatter for the corresponding direction).

the solution $\boldsymbol{W} \in \boldsymbol{R}^{d\times \mathrm{k}}$ will be the eigenvector(s) of $\mathbf{S}_w^{-1}\mathbf{S}_b$

d : the original dimensionality of the data

k : the target dimensionality after projection (usually k=m−1, where m is the number of classes).

A supervised dimensionality reduction technique $x' = W^T x$

多分类问题
multiple classification

- Binary classification methods are extended to multi- classes
- Use binary classifiers to solve multiple classification problems (commonly used)
  - Split the problem, for each of the binary classification training a classifier

    Split strategy
    - One vs. One, OvO
    - One vs. Rest, OvR
    - Many vs. Many, MvM
  - Integration of the prediction results for each classifier to obtain the final multi-classification results

一对一策略
One-vs-one

- Split stage
  - N classes split into pairs of two classes
  - N (N-1) /2 binary classifier

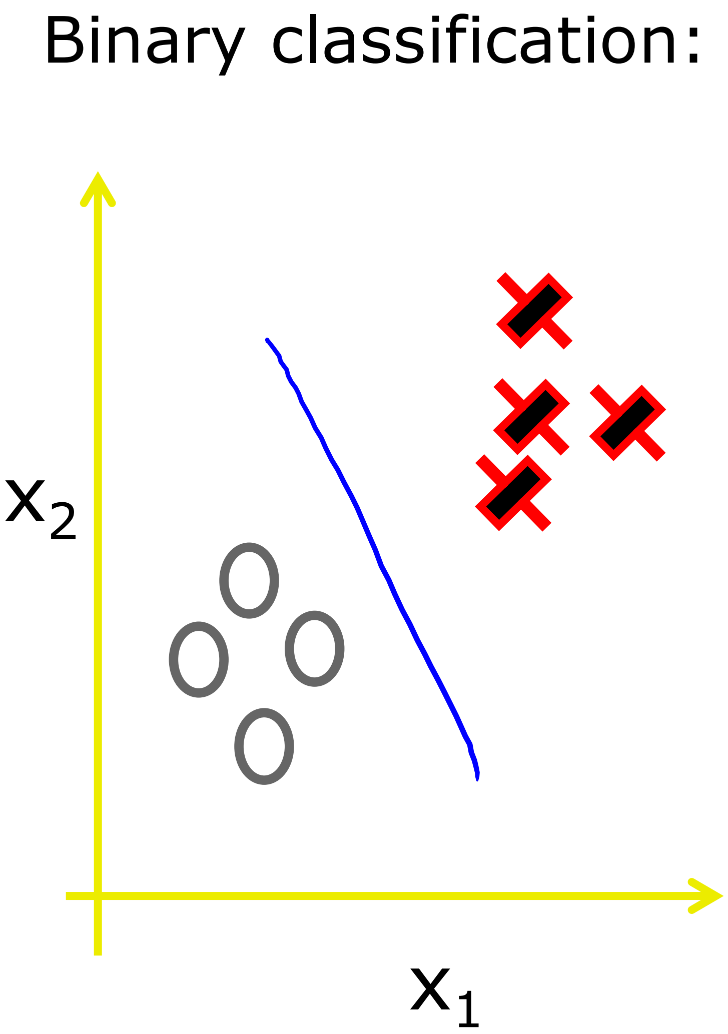
- Test stage
  - New samples are submitted to all classifiers for prediction
  - N (N-1) /2 classification results
  - Vote the final classification result

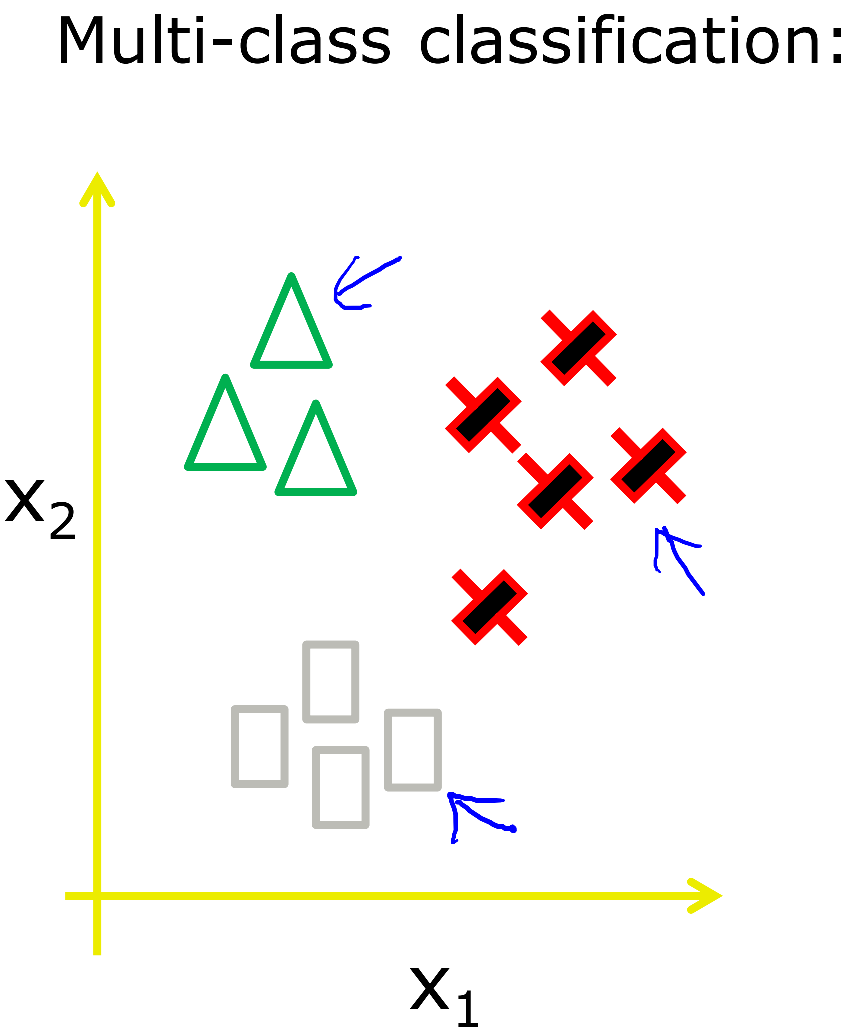
一对其余策略
One-vs-rest

- Split stage
  - Take one class as the positive example, other as negative
  - N binary classifier

- Test stage
  - New samples are submitted to all classifiers for prediction
  - N classification results
  - Compare predicate confidence of each classifier

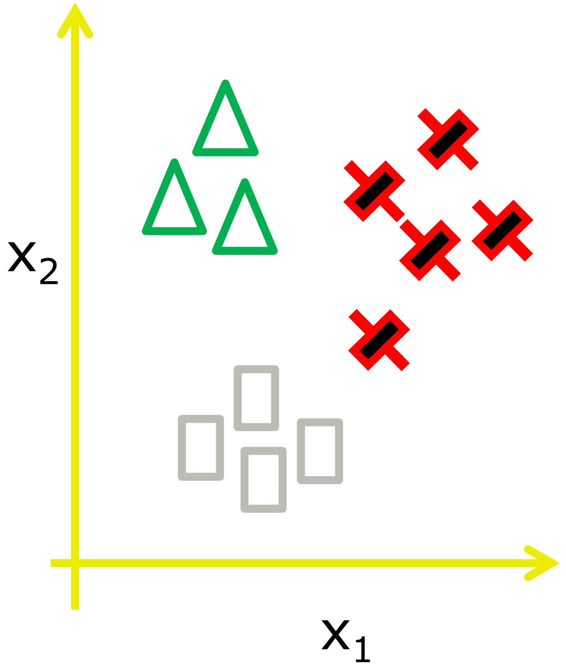
多类分类
Multiclass classification

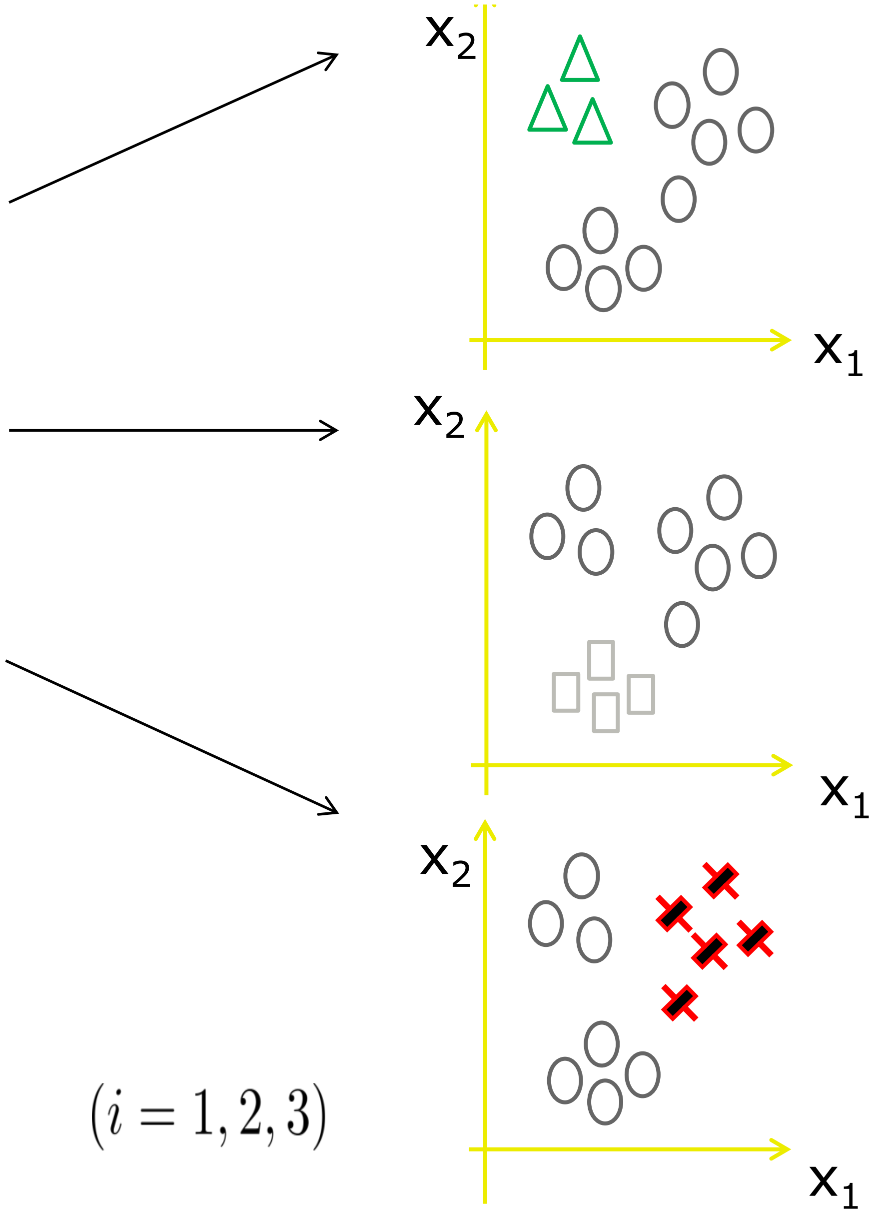




多类分类
Multiclass classification

One-vs-rest



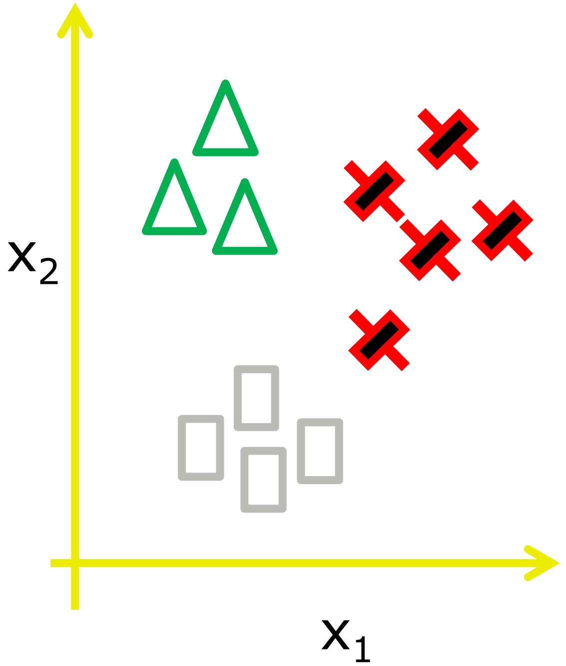


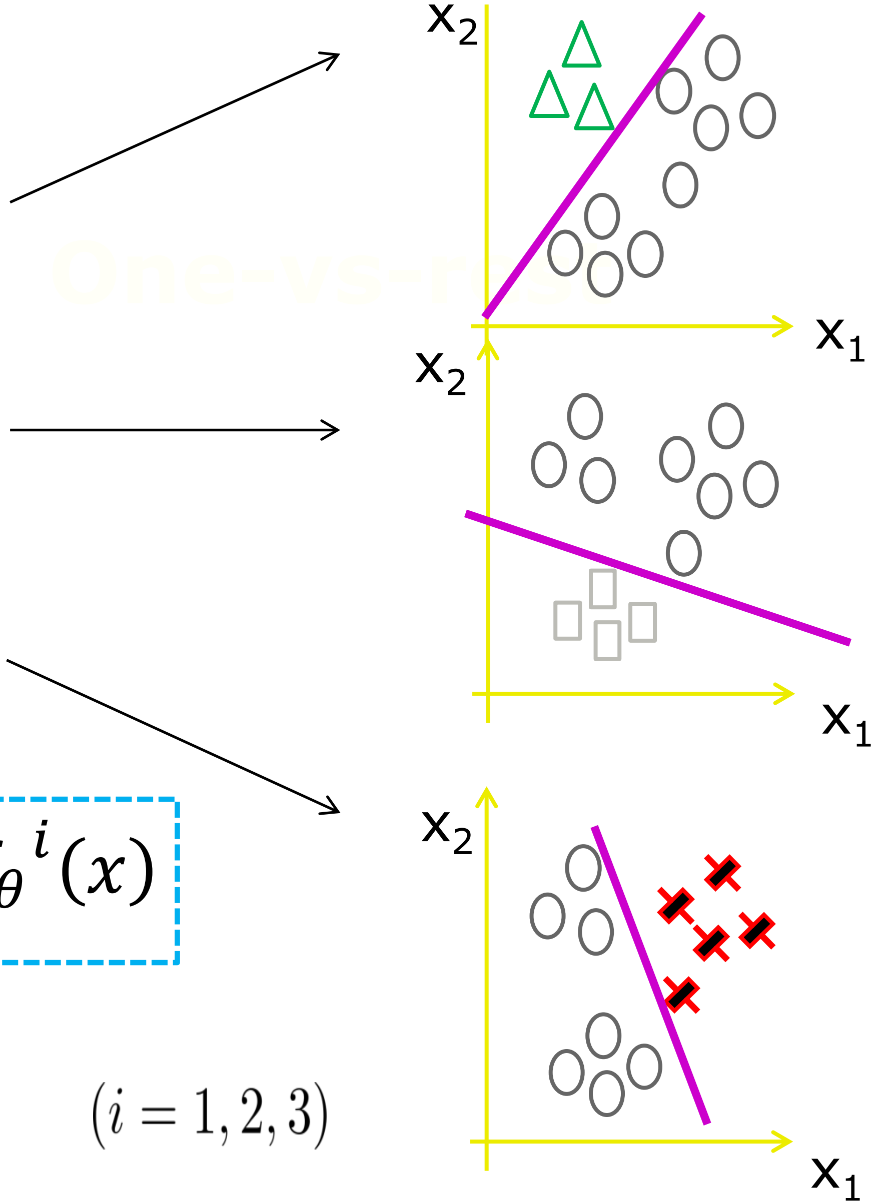
Class 1: △
Class 2: □
Class 3: ✖

$$f_\theta{}^{i}(x) = \mathrm{P}(y=i/x;\theta) \qquad (i = 1, 2, 3)$$

多类分类
Multiclass classification

One-vs-rest





Class 1:
Class 2:
Class 3:

$$\max_{i} f_{\theta}{}^{i}(x)$$

$$f_{\theta}{}^{i}(x) = \mathrm{P}(y{=}i/x;\theta) \qquad (i = 1, 2, 3)$$

类别不平衡问题
Class imbalance problem

- Class imbalance
  - The number of training samples in two classes is dramatically different

类别不平衡问题
Class imbalance problem

- Class imbalance
  - The number of training samples in two classes is dramatically different
- Rescaling
  - undersampling
    - e.g. EasyEnsemble、BalanceCascade
  - oversampling
    - e.g. SMTOE、Borderline SMOTE、ADASYN
  - Class Weighting 、threshold-moving
  - Data Augmentation
  - Ensemble Methods:…

评价标准
Evaluation metrics

- For regression tasks:
  - Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean absolute Error (MAE), R-Squared (R2)
- For classification tasks:
  - Accuracy
  - Confusion Matrix
  - True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN)
  - ROC-AUC, PR-AUC

分类性能评价
Classification Evaluation Metrics

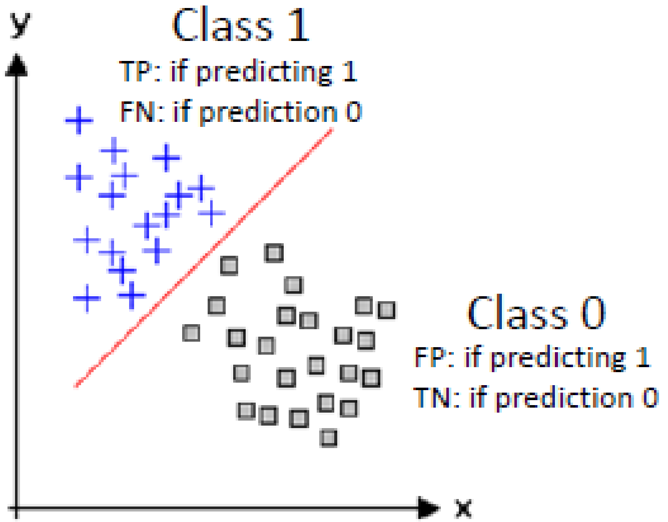
Confusion matrix

Predicted classes

Actual classes

	1	0
1	True Positive	False Negative
0	False Positive	True Negative

- True / False
  - True: prediction = label
  - False: prediction ≠ label
- Positive / Negative
  - Positive: predict y = 1
  - Negative: predict y = 0



分类性能评价：精度和错误率
Accuracy & Error rate

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- Accuracy: the ratio of cases when prediction = label

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

- Error rate: the ratio of cases when prediction ≠ label

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

分类性能评价：精度和错误率
Accuracy & Error rate

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- 问：如果正类/负类=5/95，模型把所有样本都识别成负类，请问Accuracy和Error rate？

分类性能评价：精度和错误率
Accuracy & Error

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- 问：如果正类/负类=5/95，模型把所有样本都识别成负类，请问Accuracy和Error rate？

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{95}{100} = 95\%$$

$$\text{Error rate} = \frac{FP+FN}{TP+TN+FP+FN} = \frac{5}{100} = 5\%$$

分类性能评价：精度和错误率
Accuracy & Error rate

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- It is difficult to measure actual performance in situations of imbalanced class.

分类性能评价：查准率和查全率
Precision & Recall

Prediction

		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- **Precision:** the ratio of true class 1 cases in those with prediction 1

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Prediction

		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- **Recall:** the ratio of cases with prediction 1 in all true class 1 cases

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

分类性能评价：查准率和查全率
Precision & Recall

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

		Prediction	
		1	0
Label	1	True Positive	False Negative
	0	False Positive	True Negative

- 问:如果正类/负类=50/50，模型只识别出一个正类，其余被识别成负类，请问Precision和Recall？

分类性能评价：查准率和查全率
Precision & Recall

Prediction

Label		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

Prediction

Label		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

- 问:如果正类/负类=50/50，模型只识别出一个正类，其余被识别成负类，请问Precision和Recall？

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{1}{1} = 100\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{1}{1+49} = 2\%$$

分类性能评价：查准率和查全率
Precision & Recall

Prediction

Label		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

Prediction

Label		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

- High precision and high recall indicate good model performance

分类性能评价：敏感度和特异度
Sensitivity & Specificity

Prediction

Label		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

- True / False
  - True: prediction = label
  - False: prediction ≠ label
- Positive / Negative
  - Positive: predict y = 1
  - Negative: predict y = 0

- Sensitivity（Recall）：

$$\frac{TP}{TP + FN}$$

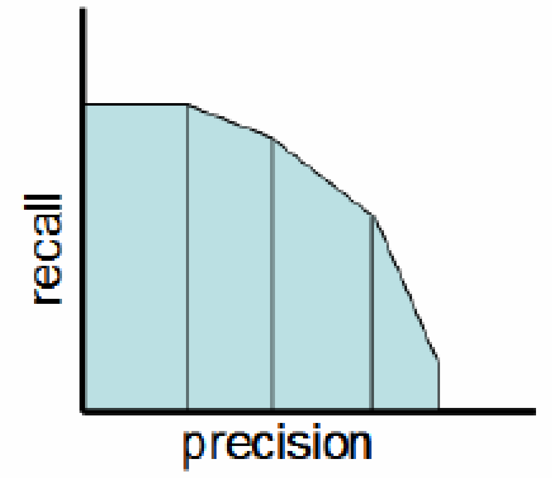
- Specificity：

$$\mathrm{FPR} = \frac{TN}{FP + TN}$$

分类性能评价：F 度量
F Score

- Precision-recall tradeoff

$$\hat{y} = \begin{cases} 1, & p_\theta(y=1|x) > h \\ 0, & \text{otherwise} \end{cases} \qquad \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$



$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

  - Higher threshold, higher precision, lower recall
    - Extreme case: threshold = 1
  - Lower threshold, lower precision, higher recall
    - Extreme case: threshold = 0

- F_1 、F_β score

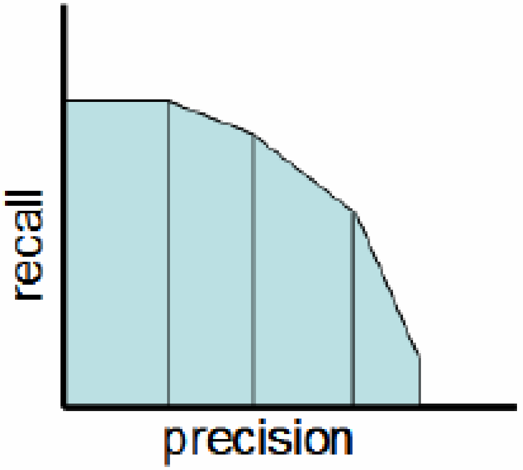
$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{(\beta^2 \times Precision) + Recall}$$

分类性能评价：PR曲线
Precision-Recall Curve

- Precision-recall tradeoff

$$\hat{y} = \begin{cases} 1, & p_\theta(y=1|x) > h \\ 0, & \text{otherwise} \end{cases} \qquad \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

  - Higher threshold, higher precision, lower recall
    - Extreme case: threshold = 1
  - Lower threshold, lower precision, higher recall
    - Extreme case: threshold = 0
- Ideal performance: ?



$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

Predicted classes

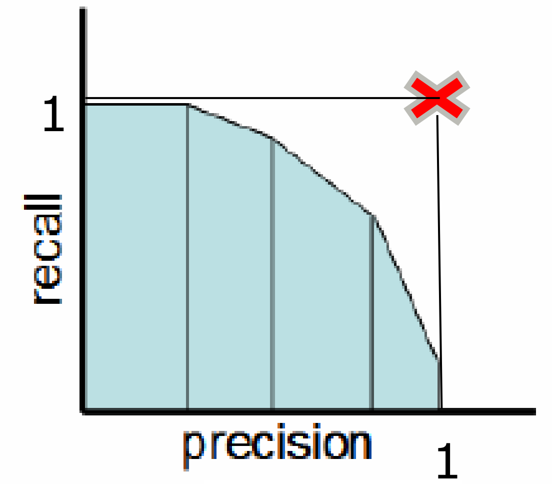
		1	0
Actual classes	1	True Positive	False Negative
	0	False Positive	True Negative

分类性能评价：PR曲线
Precision-Recall Curve

- Precision-recall tradeoff

$$\hat{y} = \begin{cases} 1, & p_\theta(y=1|x) > h \\ 0, & \text{otherwise} \end{cases} \qquad \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  - Higher threshold, higher precision, lower recall
    - Extreme case: threshold = 1
  - Lower threshold, lower precision, higher recall
    - Extreme case: threshold = 0
- Ideal performance: at coordinates (1,1)



$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

Predicted classes

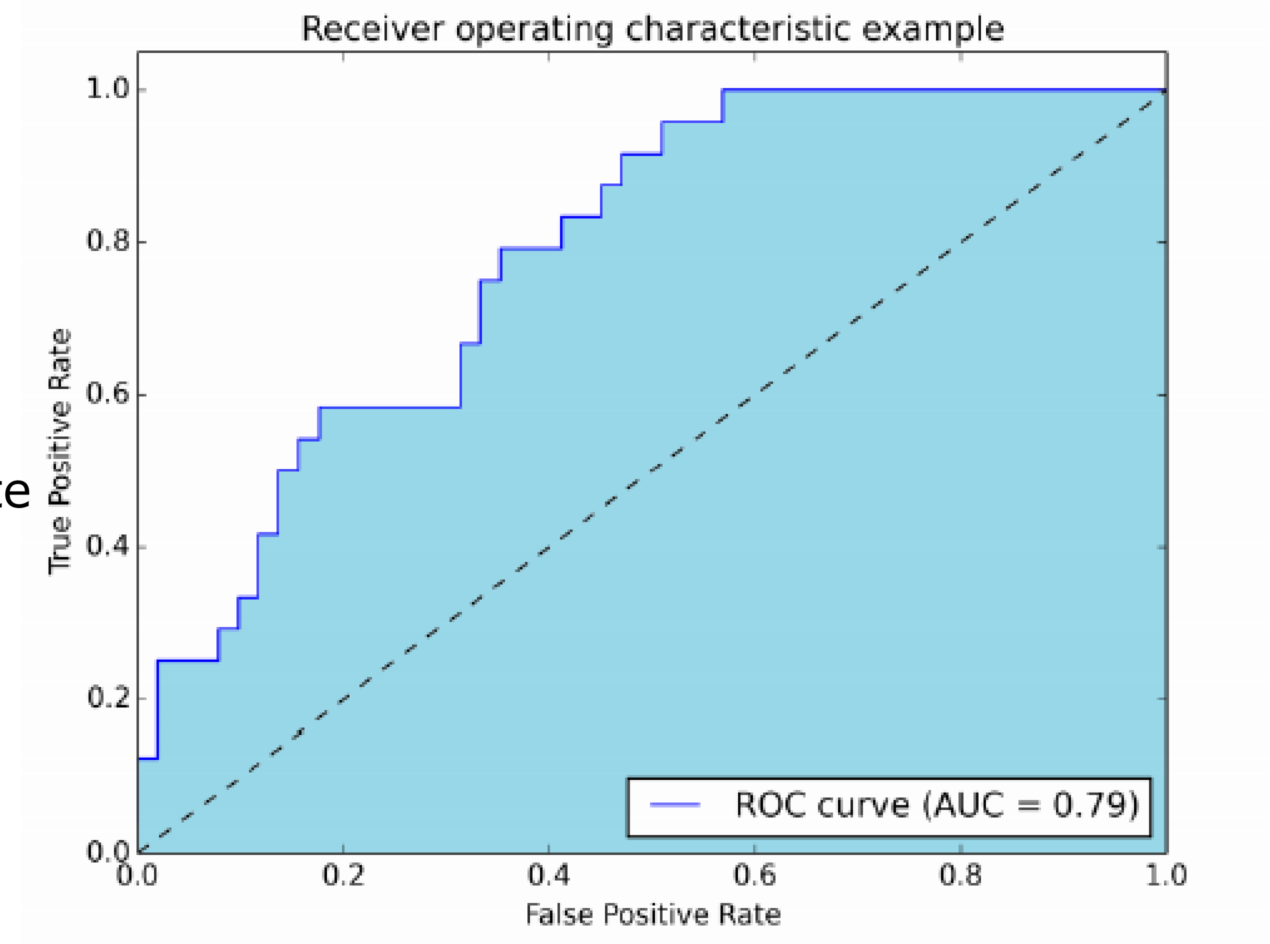
Actual classes		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

分类性能评价： ROC曲线
Receiver Operating Characteristic

- Ranking-based measure: Area Under ROC Curve (AUC)

True Positive Rate

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

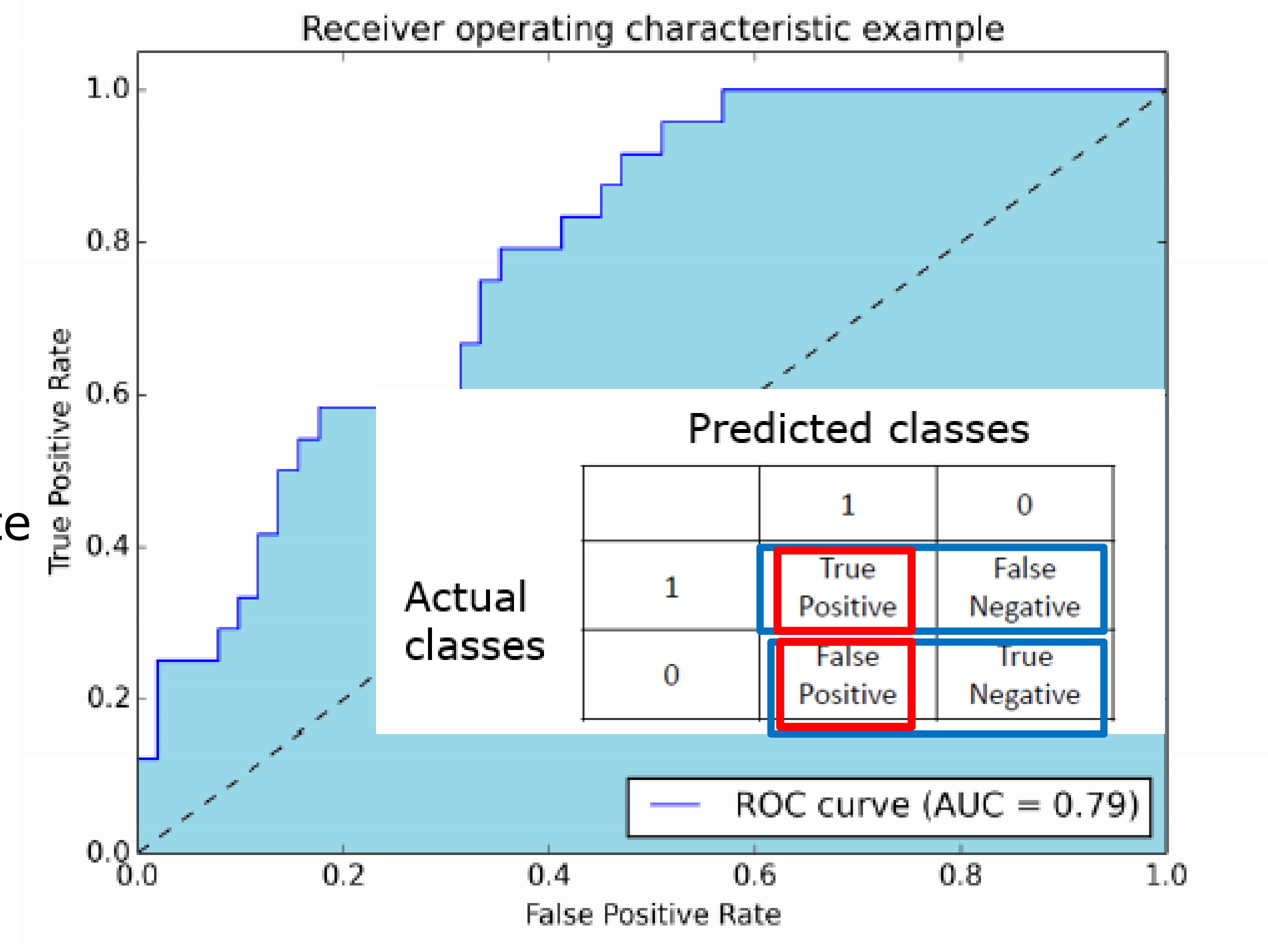


False Positive Rate

$$\mathrm{FPR} = \frac{FP}{TN + FP}$$

分类性能评价： ROC曲线
Receiver Operating Characteristic

- Ranking-based measure: Area Under ROC Curve (AUC)



True Positive Rate

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

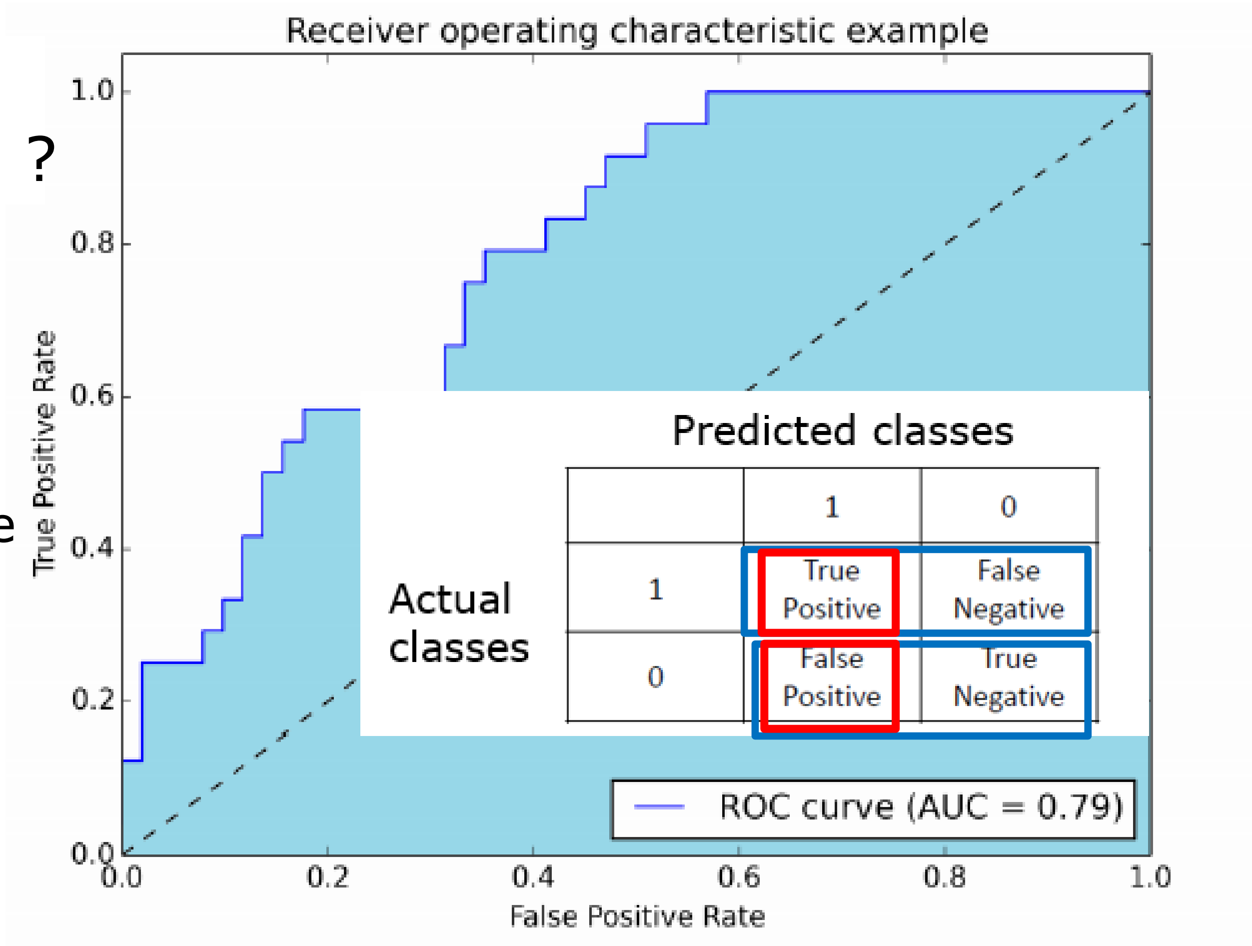
False Positive Rate

$$\mathrm{FPR} = \frac{FP}{TN + FP}$$

分类性能评价： ROC曲线
Receiver Operating Characteristic

- Ranking-based measure: Area Under ROC Curve (AUC)

Ideal performance: ?



True Positive Rate

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

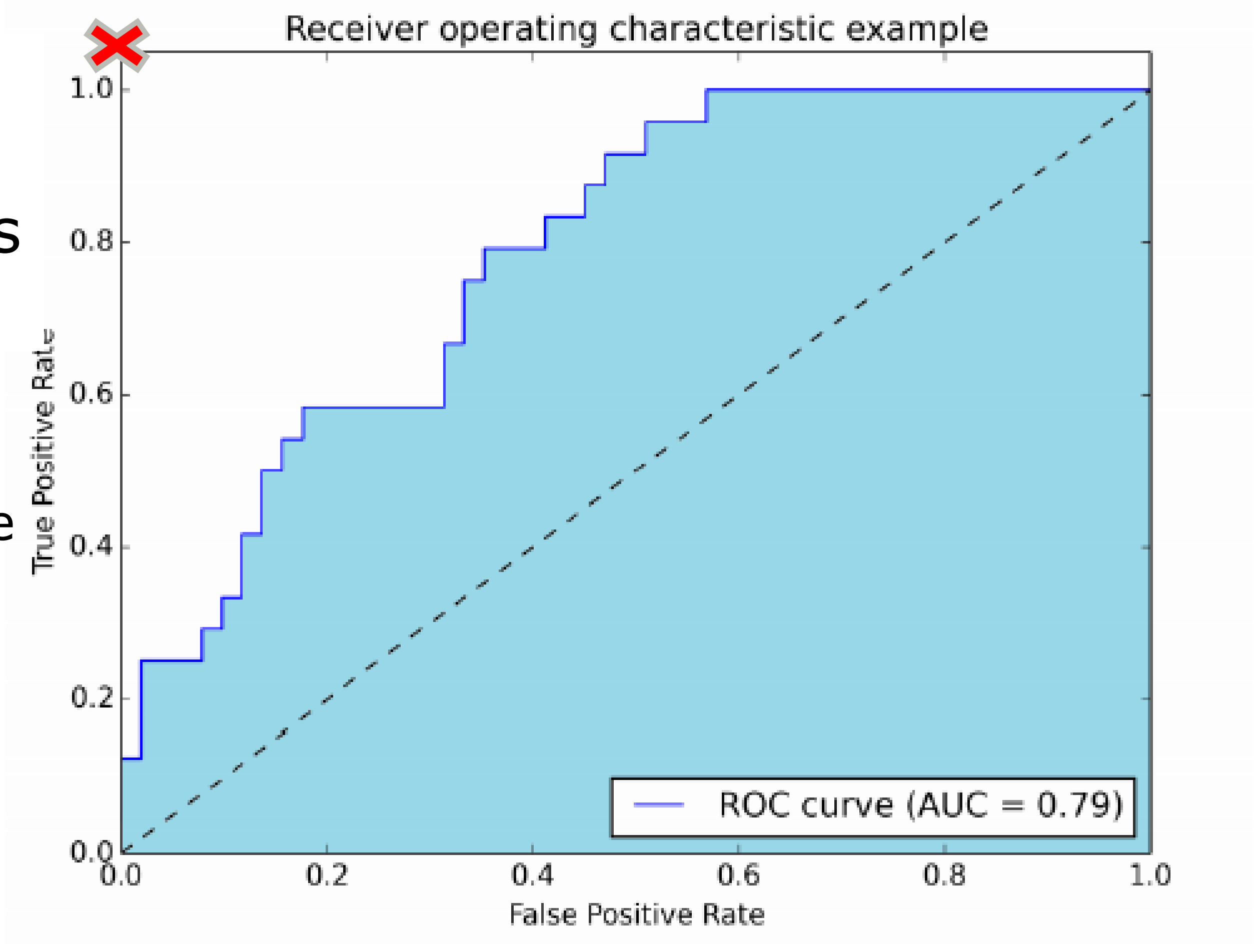
False Positive Rate

$$\mathrm{FPR} = \frac{FP}{TN + FP}$$

分类性能评价： ROC曲线
Receiver Operating Characteristic

- Ranking-based measure: Area Under ROC Curve (AUC)

Ideal performance: at coordinates (0,1)



True Positive Rate

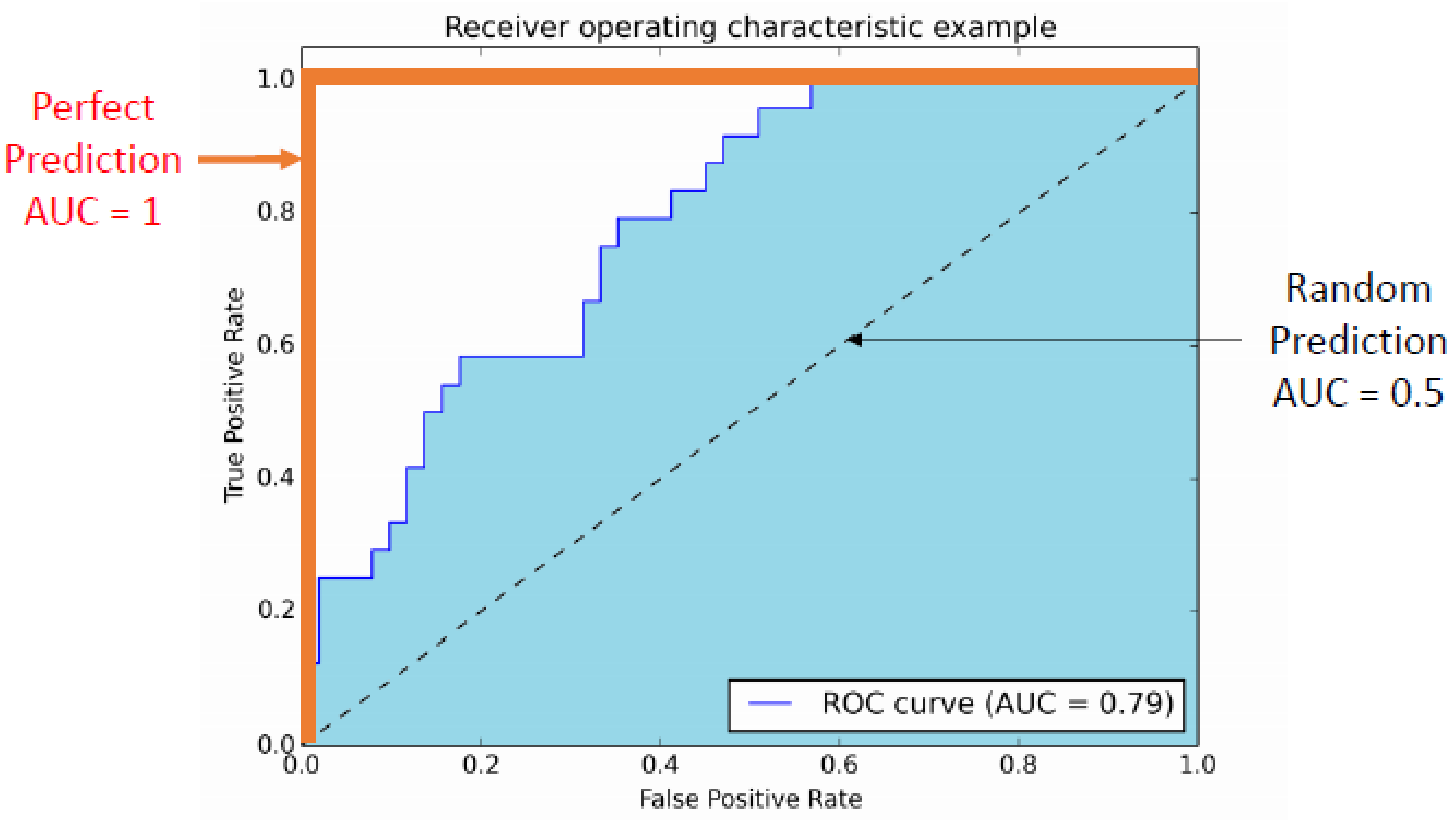
$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

False Positive Rate

$$\mathrm{FPR} = \frac{FP}{TN + FP}$$

分类性能评价： AUC 面积
Area under Curve

- Ranking-based measure: Area Under ROC Curve (AUC)



曲线绘制方法
Curve plotting method

- A simple example of Area Under ROC Curve (AUC)

- Sort the training samples by predicted values in descending order

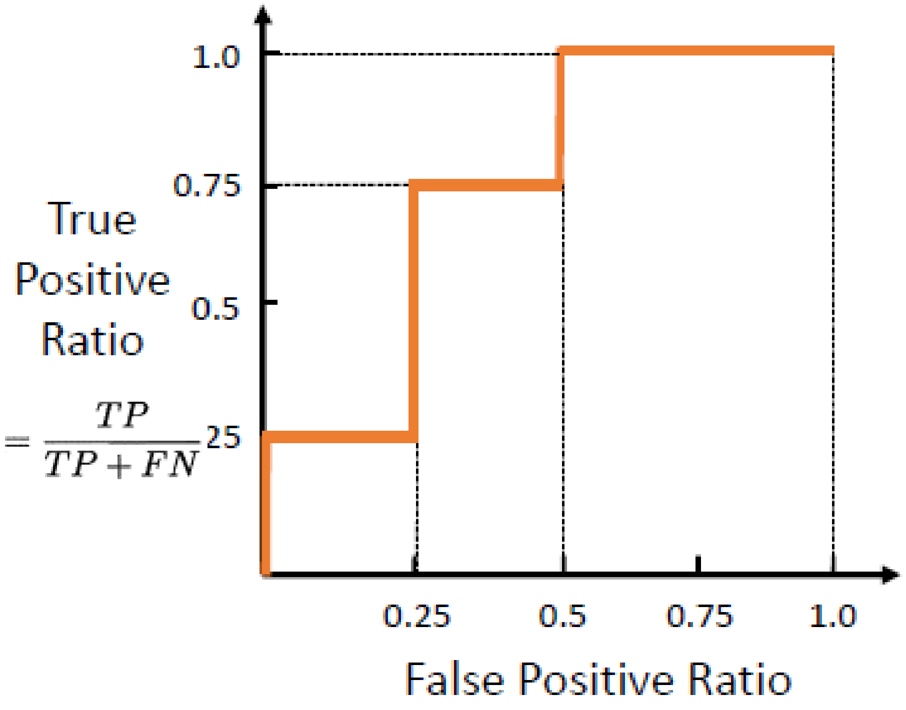
- By predicting each sample as a positive example in this order(or setting thresholds according to prediction values literately, FPR and TPR can be calculated each time

- Connect these values into a curve

Prediction	Label
0.91	1
0.85	0
0.77	1
0.72	1
0.61	0
0.48	1
0.42	0
0.33	0

曲线绘制方法
Curve plotting method

- A simple example of Area Under ROC Curve (AUC)



$$\text{TPR} = \frac{TP}{TP + FN}$$

Prediction	Label
0.91	1
0.85	0
0.77	1
0.72	1
0.61	0
0.48	1
0.42	0
0.33	0

Predicted classes

Actual classes		1	0
	1	True Positive	False Negative
	0	False Positive	True Negative

AUC = 0.75

$$\text{FPR} = \frac{FP}{TN + FP}$$

多分类任务评价指标
Multi-Class Evaluation Metrics

- Accuracy: Can be used in multi-class tasks
- Precision, Recall, F1 Score: Requires macro, micro, or weighted averaging to summarize results across all classes.
- Confusion Matrix: Expanded to an $m \times m$ matrix for multi-class tasks, showing detailed classification results.
- ROC-AUC: One-vs-Rest approach can be used, with macro or weighted averaging to summarize.
- PR-AUC: Particularly useful for imbalanced datasets, focusing more on positive classes.
- Hamming Loss: Measures the proportion of incorrectly predicted labels, useful for multi-label classification.
- Top-k Accuracy: Useful when there are multiple plausible predictions for a given task.