

# 逻辑回归

机器学习研究室

计算机科学与技术学院

吉林大学

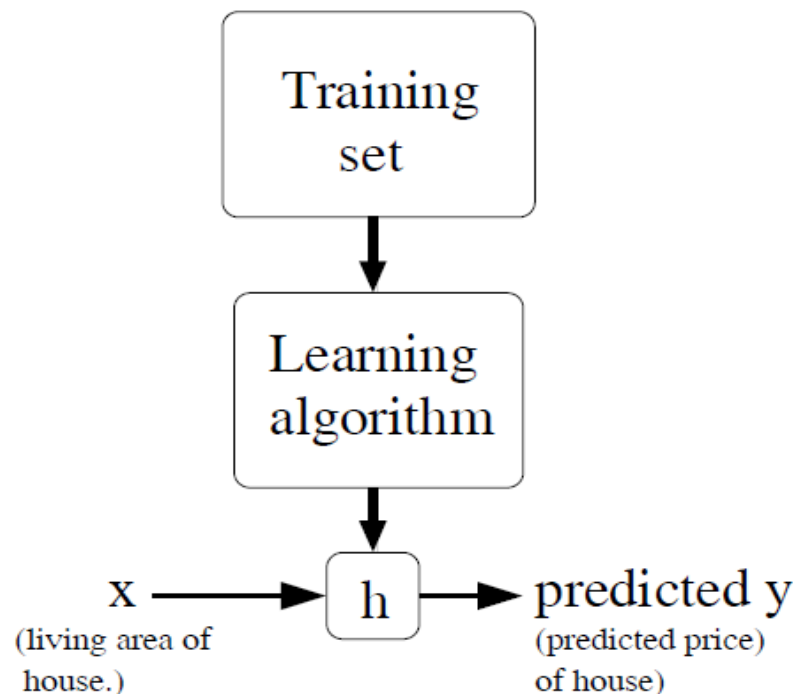
# 大纲

- 分类问题
- Logistics回归模型
- 牛顿法求解极大似然目标函数
- 模型评估方法和性能评价指标

# 分类问题

# 分类问题

- 特征  $x^{(i)}$
- 目标  $y^{(i)}$
- 训练样本  $(x^{(i)}, y^{(i)})$
- 训练集  
 $\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$
- 假设  $h(x)$



$y$  取离散值，是分类问题

# 分类问题

- 若  $y=\{1, 2, 3, \dots, M\}$ , 则称此问题为 **M分类问题**
- 通常处理的都是二分类问题
- 多分类问题常被转化为多个二分类问题
- 生活中常见的分类应用:
  - 人脸识别
  - 指纹识别
  - 手写体数字识别
  - 垃圾邮件检测

# 分类问题

## • 大型多类图像数据集—ImageNet

IMAGENET

14,197,122 images, 21841 synsets indexed

[Explore](#) [Download](#) [Challenges](#) [Publications](#) [Updates](#) [About](#)

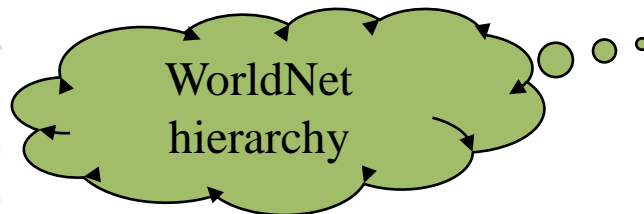
Not logged in. [Login](#) | [Signup](#)

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

[Click here](#) to learn more about ImageNet, [Click here](#) to join the ImageNet mailing list.

✓ 14M images

✓ 21k synsets indexed



<https://wordnet.princeton.edu/>



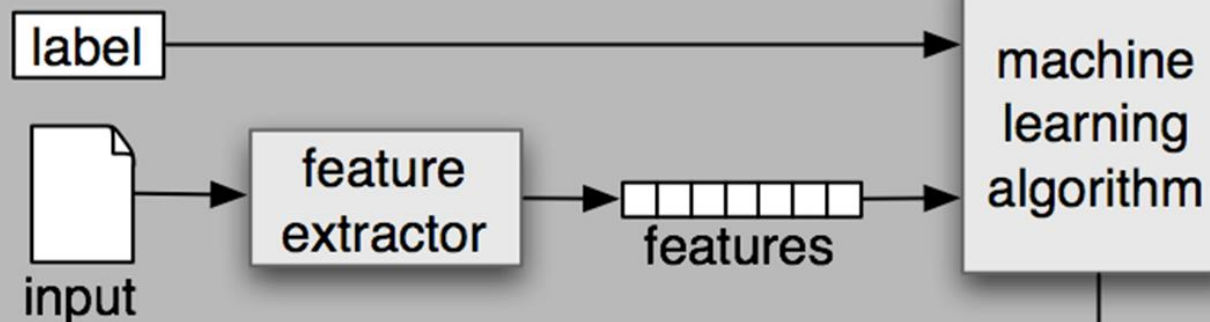
mite	container ship	motor scooter	leopard
mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat

<http://www.image-net.org/>

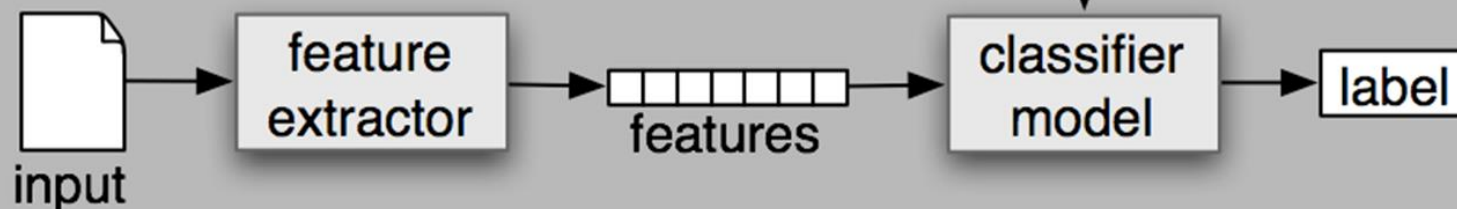
# 分类问题

## 分类的训练及预测过程

### (a) Training



### (b) Prediction





# 分类问题

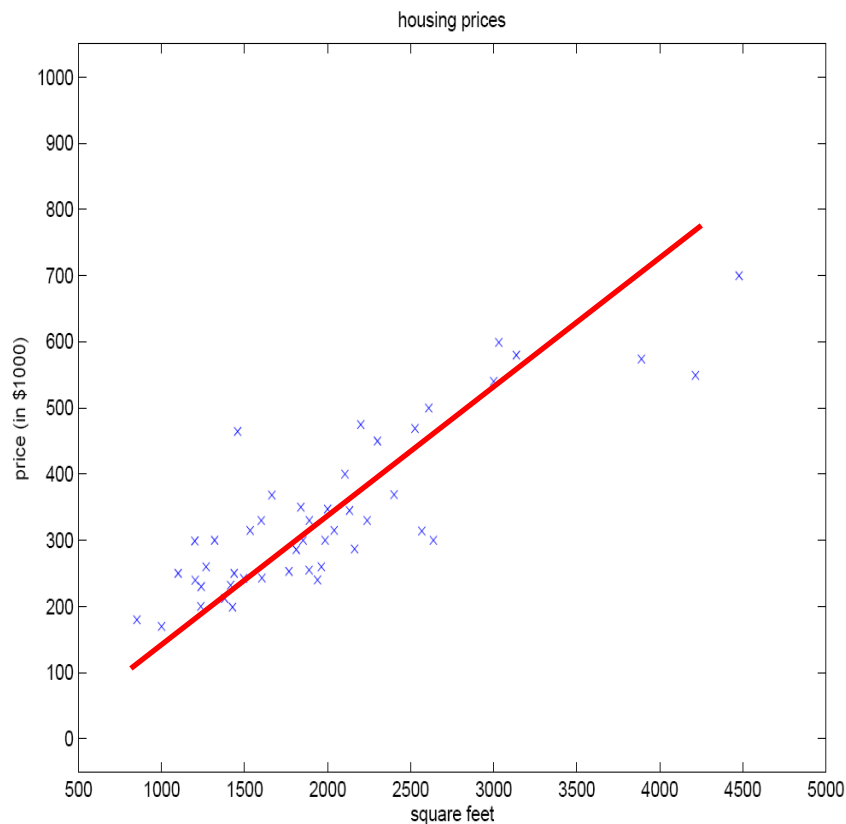
- Classification problems
  - $y \in \{c_1, c_2, \dots, c_K\}$
- Binary-classification
  - $y \in \{0, 1\}$
- An example
  - Spam filter
  - $x$ :  $\rightarrow$  features
  - $y$ :  $\rightarrow$  label
  - $y = 1$ :  $\rightarrow$  positive label
  - $y = 0$ :  $\rightarrow$  negative label





# 分类问题

- 线性分类器



线性回归  
的目标

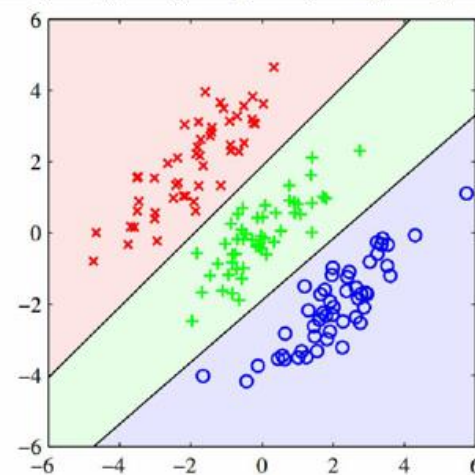
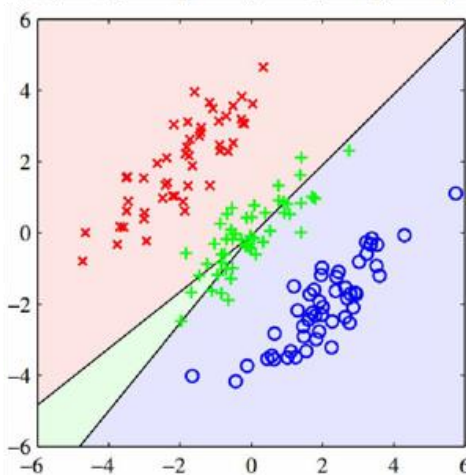
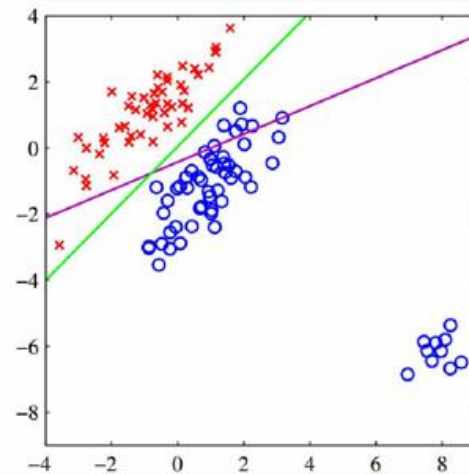
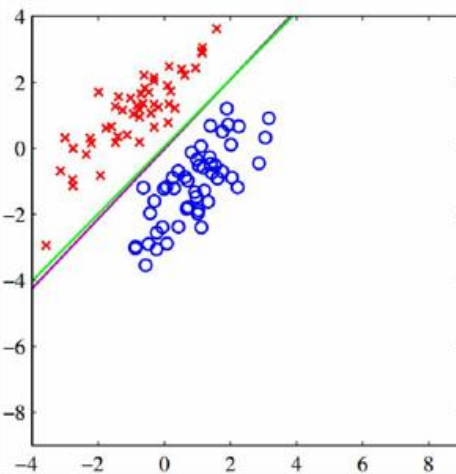
$$y = \theta^T x = \theta_0 + \sum_{j=1}^n \theta_j x_j$$

# 分类问题

- 是否可以直接用回归解决分类问题？

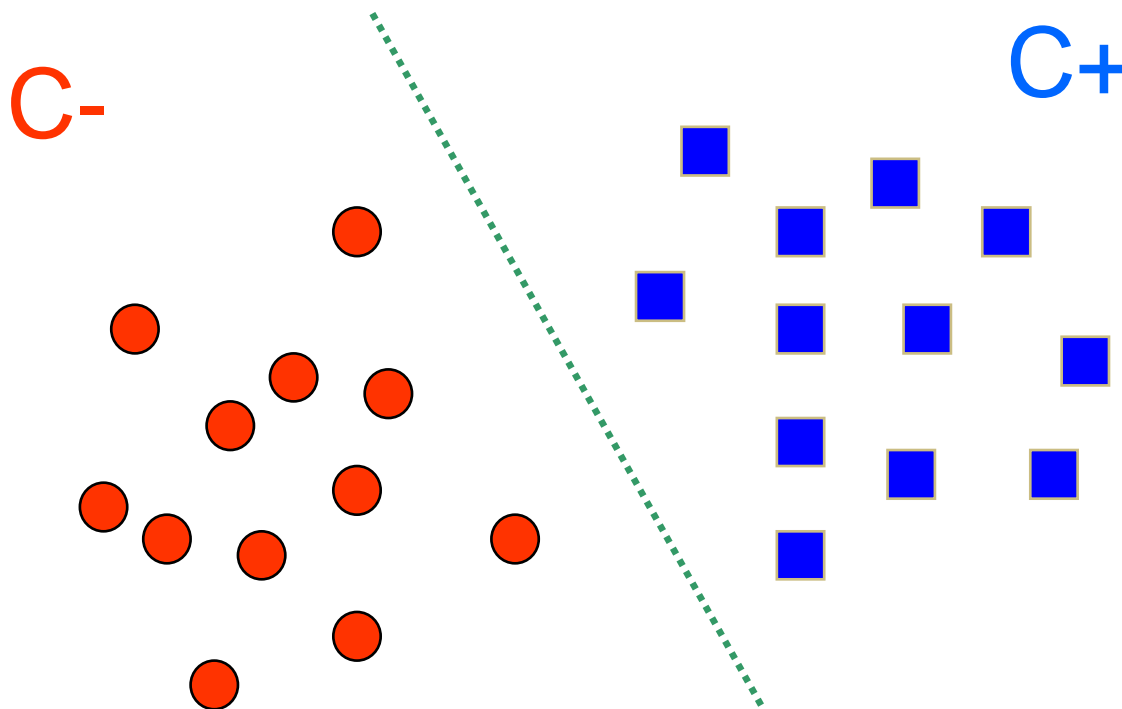
- 紫色：
  - 线性回归
- 绿色：
  - Logistic回归

- 左侧：
  - 线性回归
- 右侧：
  - Softmax回归



# Logistics regression

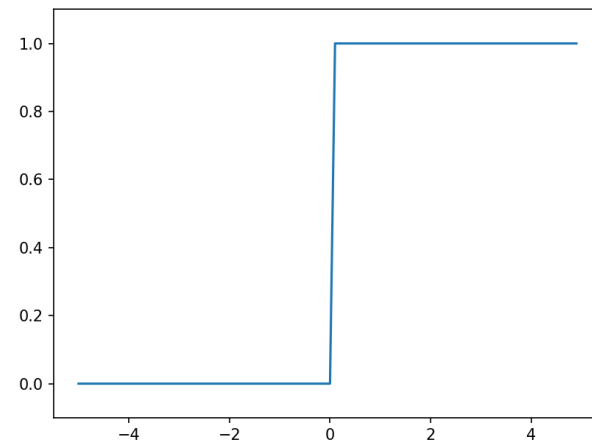
- 决策边界  $y = \theta^T x = \theta_0 + \sum_{j=1}^n \theta_j x_j = 0$



- $y > 0$ : → positive label
- $y < 0$ : → negative label

# Logistics regression

- 问题：能否架起连接线性回归问题和二分类问题的桥梁？
- 最理想的函数—单位阶跃函数
$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$
  - 预测值大于零，判为正例
  - 预测值小于零，判为反例
  - 预测值为零，则无法判别（可任意判别，小概率）
- 单位阶跃函数的缺点
  - 不连续，确定其模型参数 $\theta$ 困难



# 分类问题

- 硬分类与软分类

- Spam filter

- $\mathbf{x}$ :  $\rightarrow$  features

- $y$ :  $\rightarrow$  label

- $y = 1$ :  $\rightarrow$  positive label

- $y = 0$ :  $\rightarrow$  negative label



- Spam filter

- $\mathbf{x}$ :  $\rightarrow$  features

- $y$ :  $\rightarrow$  the probability of (say) positive class that is,  $y \in [0, 1]$

- 这两类分类方法有何差异

硬分类：离散、  
类别标签

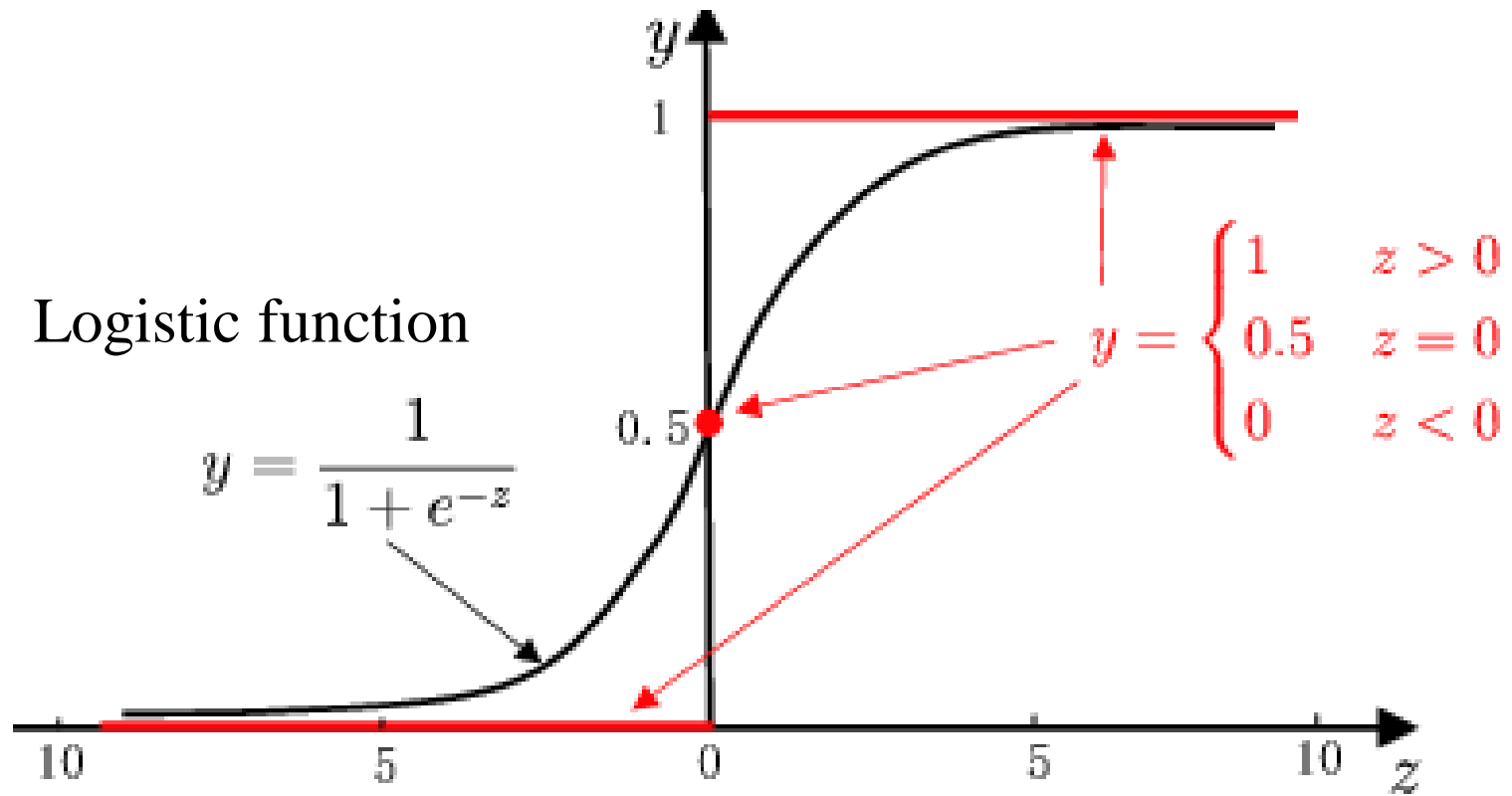
软分类：连续、  
概率值( $P(Y|X)$ )



# LOGISTIC REGRESSION

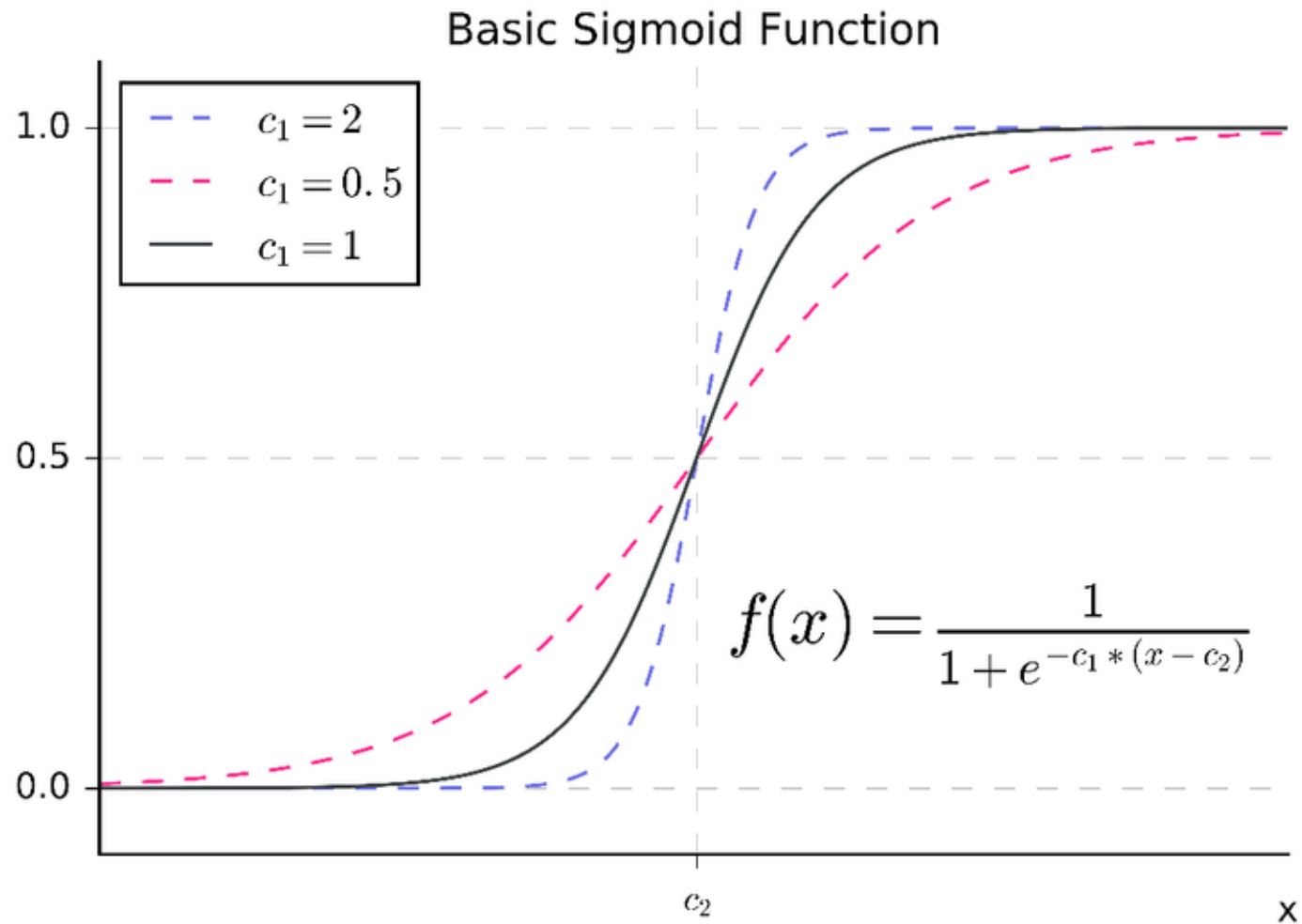
# Logistic regression

- 单位阶跃函数 vs. Sigmoid function





# Logistic regression



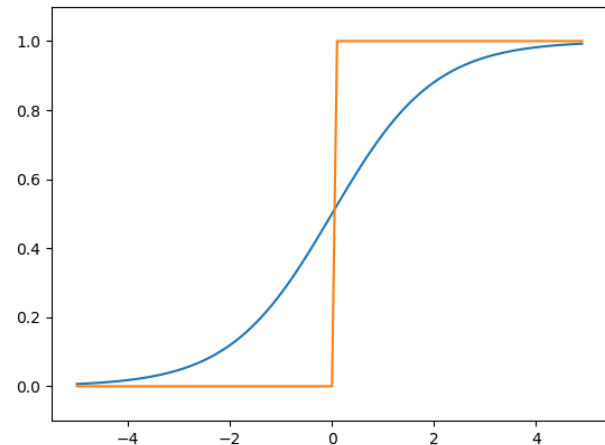
# Logistic regression

## Logistic函数(分布)的性质

$$g(z) = \frac{1}{1 + e^{-z}}$$

- 主要性质

- $\lim_{z \rightarrow \infty} g(z) = 1$
- $\lim_{z \rightarrow -\infty} g(z) = 0$
- $0 < g(z), h(z) < 1$
- $g(z)$ 的导数形式良好

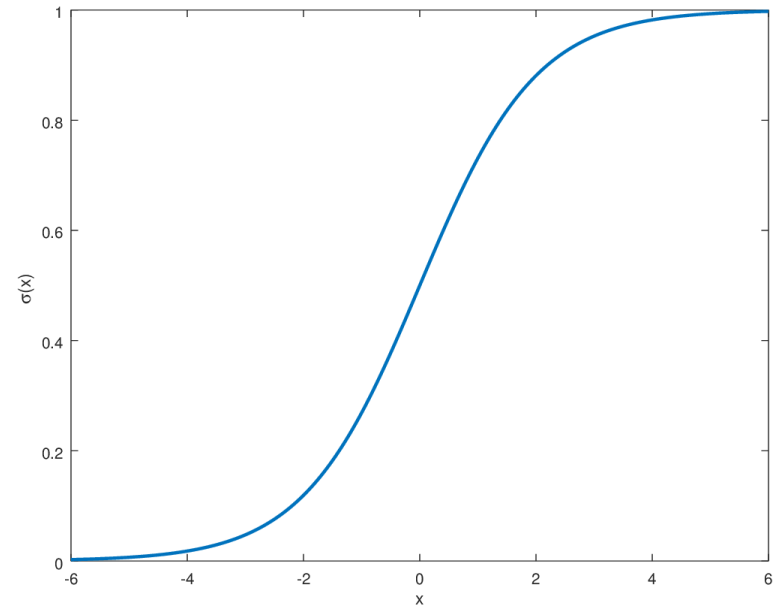


$$\begin{aligned} g'(z) &= \frac{d}{dz} \frac{1}{1 + e^{-z}} \\ &= \frac{1}{(1 + e^{-z})^2} (e^{-z}) \\ &= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{(1 + e^{-z})}\right) \\ &= g(z)(1 - g(z)) \end{aligned}$$

# Logistic regression

$$g(z) = \frac{1}{1 + e^{-z}}$$

logistic function or  
sigmoid function



$$z = \theta^T x = \theta_0 + \sum_{j=1}^n \theta_j x_j$$

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

# Logistic regression

- Probabilistic assumption of binary classification

$$P(y = 1|x; \theta) = h_{\theta}(x)$$

$$P(y = 0|x; \theta) = 1 - h_{\theta}(x)$$

- Compact form of binary classification:  $y \in \{0, 1\}$

$$P(y|x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y_i}$$

# Logistic regression

- Output with sigmoid function

$$P(y = 1|\mathbf{x}; \theta) = h_{\theta}(\mathbf{x}) = \frac{1}{1 + e^{-\theta^T \mathbf{x}}}$$

$$P(y = 0|\mathbf{x}; \theta) = 1 - h_{\theta}(\mathbf{x}) = \frac{e^{-\theta^T \mathbf{x}}}{1 + e^{-\theta^T \mathbf{x}}}$$

# Logistic regression

- Odds(几率、几率比)

- 在统计和概率理论中，一个事件或者一个陈述的发生比（英语：Odds）是该事件发生和不发生的比率，又称几率、几率比，公式为：

$$\frac{p}{1-p}$$

其中， $p$ 是该事件或陈述的概率）

- 例如，如果一个人随机选择一星期7天中的一天，选择星期日的发生比是：

$$\frac{1/7}{1 - (1/7)} = \frac{1/7}{6/7} = \frac{1}{6}$$

不选择星期日的发生比是： $\frac{6}{1}$ 。

- 几率比其实是一种相对概率。一般来说，日常不太使用几率比来描述概率。

# Logistic regression

- Logistic regression

$$P(y = 1|\mathbf{x}; \theta) = h_{\theta}(\mathbf{x}) = \frac{1}{1 + e^{-\theta^T \mathbf{x}}}$$

- Odds of positive samples

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

Odds ratio

- Logit Transformation

$$\ln \frac{P(y = 1|\mathbf{x}; \theta)}{1 - P(y = 1|\mathbf{x}; \theta)} = \ln \frac{h_{\theta}(\mathbf{x})}{1 - h_{\theta}(\mathbf{x})} = \theta^T \mathbf{x}$$

Logit Transformation



# Logistic regression

- 似然函数

- 1. 似然与概率的区别

似然 (likelihood) 与概率 (probability) 在英语语境中是可以互换的。但是在统计学中，二者有截然不同的用法。

概率描述了已知参数时的随机变量的输出结果；**似然则用来描述已知随机变量输出结果时，未知参数的可能取值。**

例如，对于“一枚**正反对称**的硬币上抛十次”这种事件，我们可以问硬币落地时十次都是正面向上的“概率”是多少；而对于“一枚硬币上抛十次”，我们则可以问，这枚硬币正反面对称的“似然”程度是多少。

区别似然和概率的直接方法为，“XXX的概率”中XXX只能是事件，也就是，事件(发生)的概率是多少；而“XXX的似然”中的XXX只能是参数，比如说，参数等于某个值时的似然是多少。

# Logistic regression

- 似然函数

- 2. 似然与概率的联系

先看似然函数的定义。关于参数  $\theta$  的似然函数（在数值上）等于给定参数  $\theta$  后变量  $data$  的概率（两者的相等并不是说两个函数是同一个，只是数值上的相等）：

$$L(\theta|data) = P(data|\theta) = \prod_{i=1}^N P(x_i|\theta)$$
$$data = (x_1, x_2, \dots, x_n)$$

似然函数的主要用法在于比较它相对取值，虽然这个数值本身不具备任何含义。例如，考虑一组样本，当其输出固定时，这组样本的某个未知参数往往会倾向于等于某个特定值，而不是随便的其他数，此时，似然函数是最大化的。

似然函数乘以一个正的常数之后仍然是似然函数，其取值并不需要满足归一化条件

$$\sum_x \alpha \cdot L(\theta|x) \neq 1, \alpha > 0$$

# Logistic regression

- 似然函数

- 3. 最大似然估计

最大似然估计是似然函数最初也是最自然的应用。似然函数取得最大值表示相应的参数能够使得统计模型最为合理。从这样一个想法出发，最大似然估计的做法是：首先选取似然函数（一般是概率密度函数或概率质量函数），整理之后求最大值。实际应用中一般会取似然函数的对数作为求最大值的函数，这样求出的最大值和直接求最大值得到的结果是相同的。似然函数的最大值不一定唯一，也不一定存在。

# Logistic regression

- Assuming that we have  $m$  **training examples generated independently**, then


$$\begin{aligned}\max_{\theta} L(\theta) &= p(\vec{y} \mid X; \theta) \\ &= \prod_{i=1}^m p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}\end{aligned}$$

•  
•  
•  
最大化似然目标函数


# Logistic regression

- It will be easier to maximize the log likelihood:

$$\begin{aligned}\max_{\theta} l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^m y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))\end{aligned}$$



如何求解  
此优化问题？！

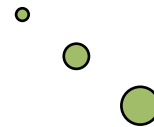


最大化对数似然  
目标函数

# Logistic regression

- Gradient ascent

$$\theta := \theta + \alpha \nabla \ell(\theta)$$

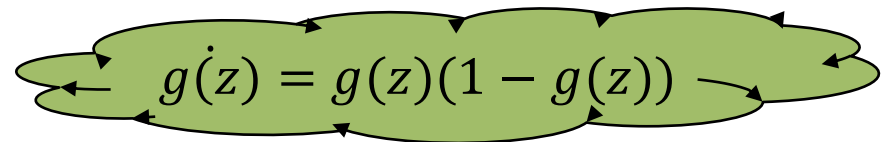


注意: 与梯度下降法对比  
的符号差异 (+? , -? )

# Logistic regression

- Again, if only one sample, then

$$\begin{aligned}
 \frac{\partial}{\partial \theta_j} \ell(\theta) &= \left( y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{1 - g(\theta^T x)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x) \\
 &= \left( y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{1 - g(\theta^T x)} \right) \frac{g(\theta^T x)(1 - g(\theta^T x))}{g(\theta^T x)(1 - g(\theta^T x))} \frac{\partial}{\partial \theta_j} \theta^T x \\
 &= (y(1 - g(\theta^T x)) - (1 - y)g(\theta^T x)) x_j \\
 &= (y - h_\theta(x)) x_j
 \end{aligned}$$



$$g'(z) = g(z)(1 - g(z))$$



# Logistic regression

- Stochastic gradient **ascent** for Logistic regression

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_{\theta}(\mathbf{x}^{(i)}))x_j^{(i)}$$

Some comments on  
LSM

似曾相识?  
殊途同归?

$$h_{\theta}(\mathbf{x}^{(i)}) = \begin{cases} \theta^T \mathbf{x}^{(i)} \\ g(\theta^T \mathbf{x}^{(i)}) = \frac{1}{1 + e^{-\theta^T \mathbf{x}^{(i)}}} \end{cases}$$

Linear regression

Logistic regression

# Logistic regression

- 实验数据

- 该数据集共包括150行，每行为 1个样本。每个样本有5个字段，包含4种属性和1个分类信息。

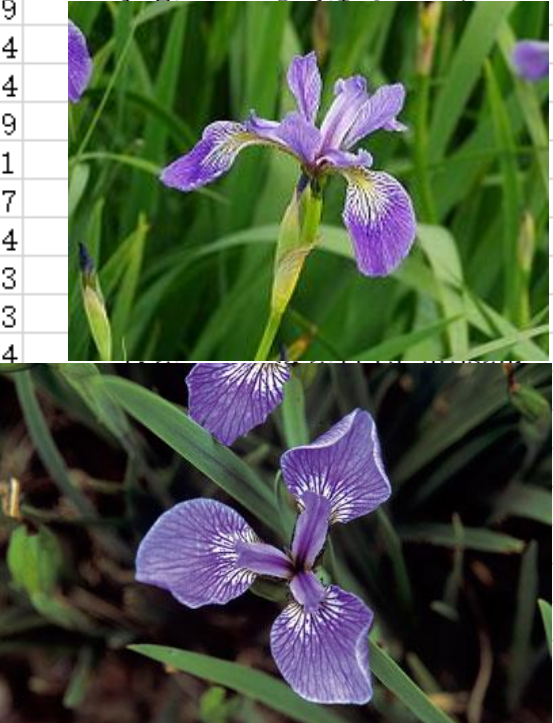
- 4种属性分别是：

- (1) 花萼长度 (单位 cm)
- (2) 花萼宽度 (单位: cm)
- (3) 花瓣长度 (单位: cm)
- (4) 花瓣宽度 (单位: cm)

- 类别信息：

- (1) Iris Setosa (山鸢尾)
- (2) Iris Versicolour (杂色鸢尾)
- (3) Iris Virginica (维吉尼亚鸢尾)

5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9			
4.6	3.4			
5	3.4			
4.4	2.9			
4.9	3.1			
5.4	3.7			
4.8	3.4			
4.8	3			
4.3	3			
5.8	4			
5.7	4.			
5.4	3.			
5.1	3.			
5.7	3.			
5.1	3.			
5.4	3.			
5.1	3.			
4.6	3.			



<http://archive.ics.uci.edu/ml/datasets/Iris>

# Logistic regression

- 实验过程
- **sklearn** 的线性回归

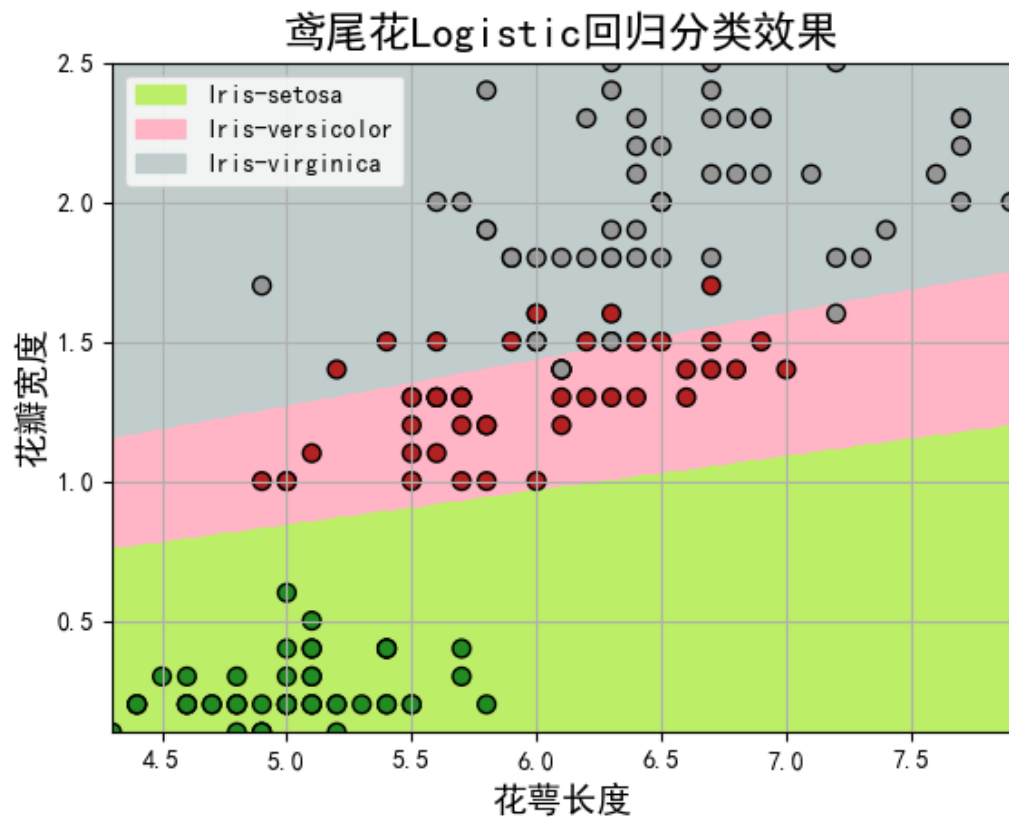
`from sklearn.linear_model`

`lr = LogisticRegression`

`lr.fit(x, y.ravel())`

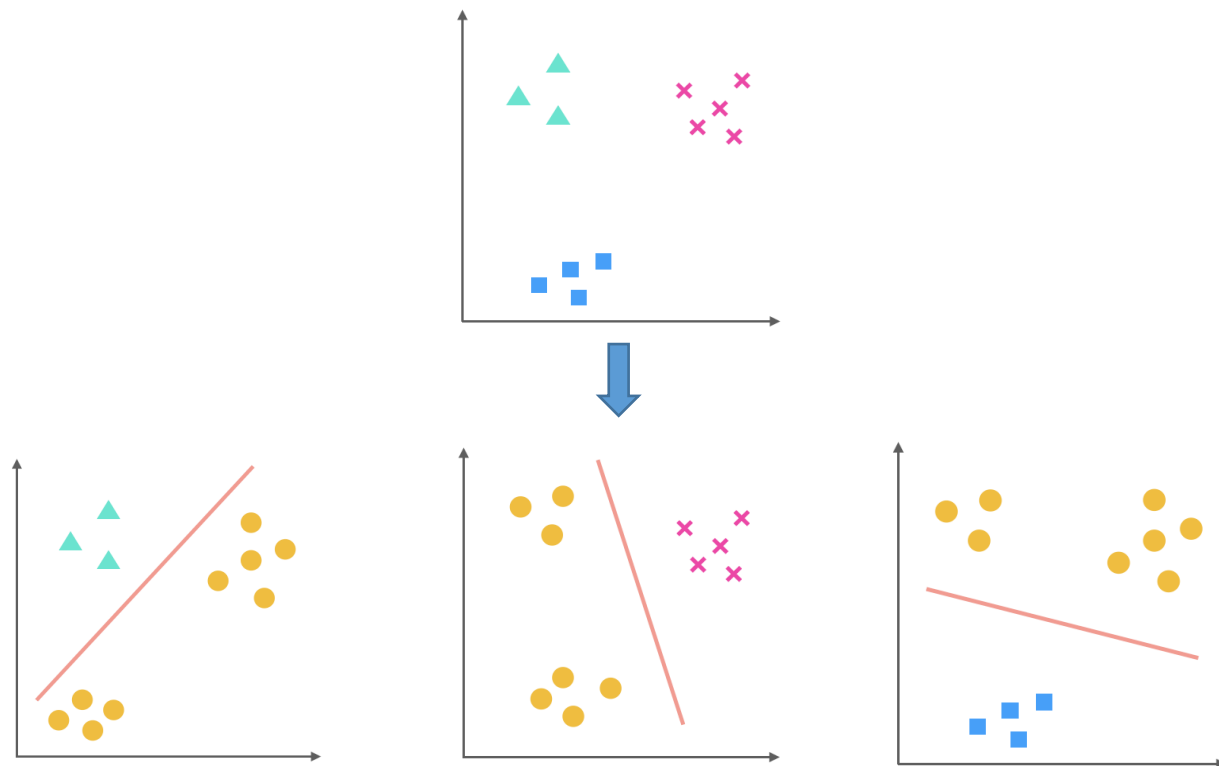
`y_hat = lr.predict(x)`

`y_hat_prob = lr.predict_proba(x)`



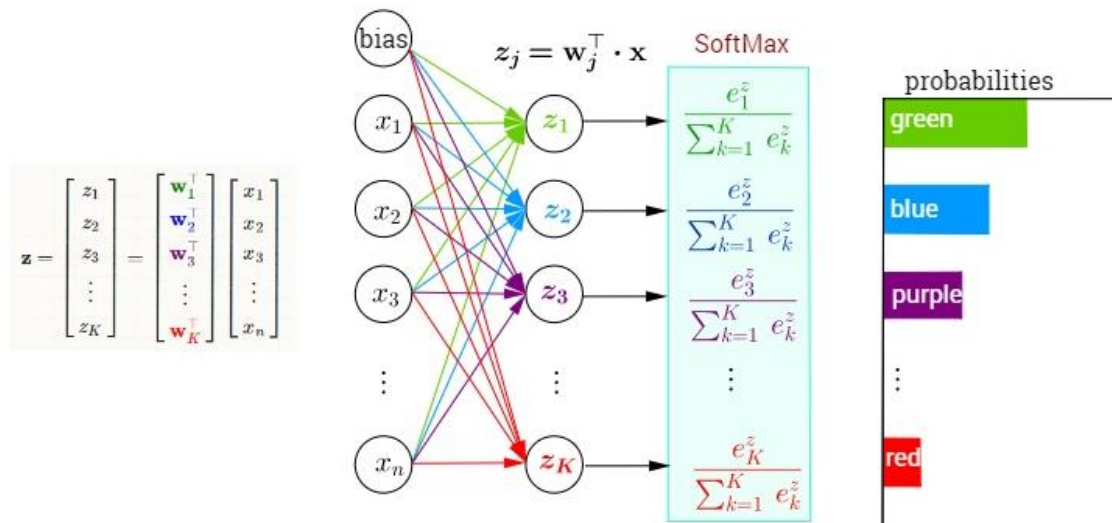
# Logistic regression

- Logistic回归多分类的实现
- One-vs-Rest:  
把一个多分类的问题变成多个二分类的问题



# Logistic regression

- Logistic回归多分类的实现
- One-vs-One:  
每次选择两类数据进行二分类
- 使用Softmax函数:



# Logistic regression

- Softmax回归

对于输入数据 $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 有  $k$  个类别，即  $y_i \in \{1, 2, \dots, k\}$ ，那么 softmax 回归主要估算输入数据  $x_i$  归属于每一类的概率，即

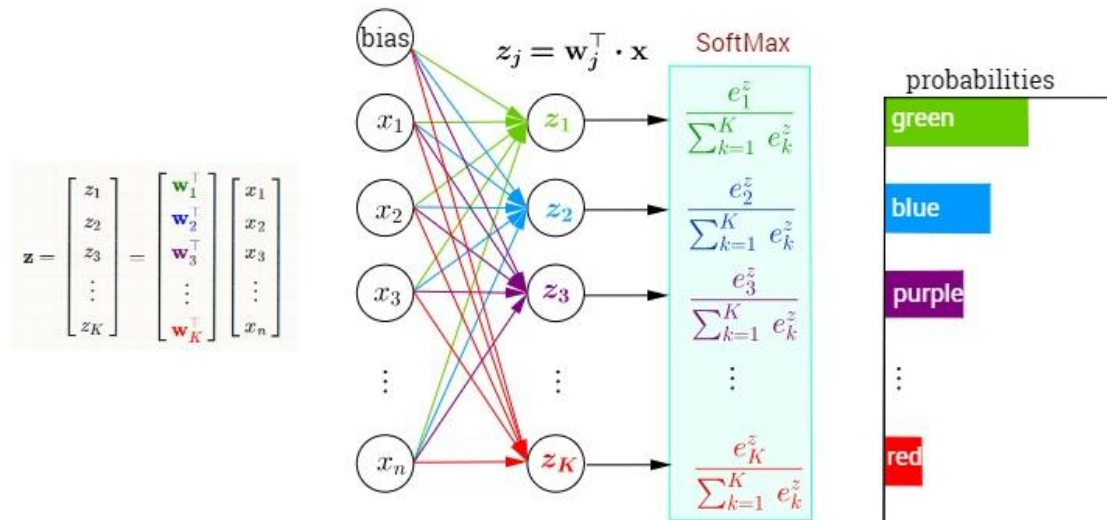
$$h_{\theta}(x_i) = \begin{bmatrix} p(y_i = 1 | x_i; \theta) \\ p(y_i = 2 | x_i; \theta) \\ \vdots \\ p(y_i = k | x_i; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ \vdots \\ e^{\theta_k^T x_i} \end{bmatrix}$$

# Logistic regression

- Softmax回归

输入数据  $x_i$  归属于类别  $j$  的概率为:

$$p(y_i = j | x_i; \theta) = \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^K e^{\theta_l^T x_i}}$$





# 牛顿法 求解对数似然函数的极大值点

# 牛顿法

- 牛顿迭代法求近似解

- 求方程 $f(x)=0$ 的解

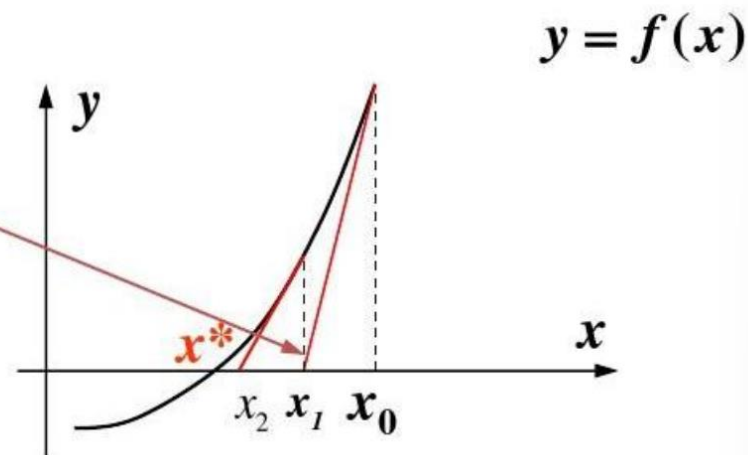
**牛顿迭代公式：**  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

.....

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

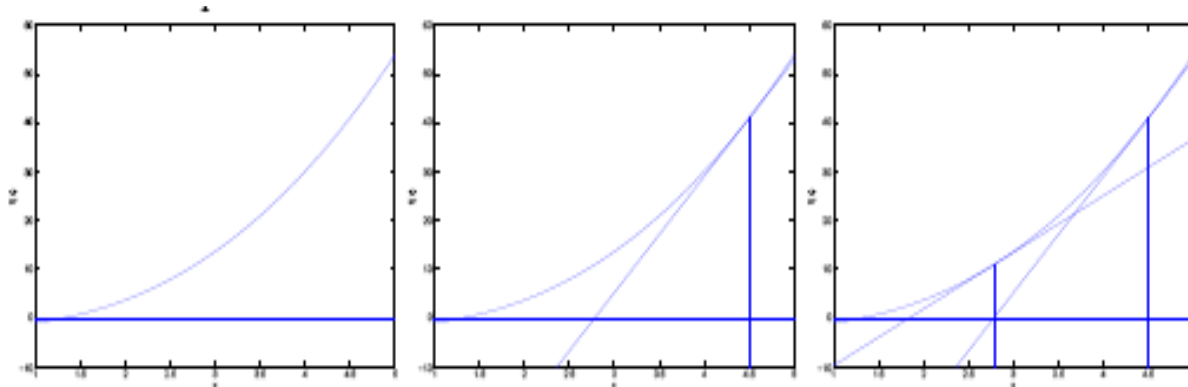


# Newton's method

- 牛顿法的函数
  - 给出计算 $f(\theta) = 0$ 的方法
- 如果  $\theta \in R$ , 为了计算 $f(\theta) = 0$ , 牛顿法执行以下更新过程:

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)}$$

- 牛顿法的实际执行过程



# 牛顿法

- 如何求解逻辑回归对数似然函数的极大值点

- 令  $f(\theta) = l'(\theta) = 0$ , 则牛顿迭代公式为

$$\theta := \theta - \frac{l'(\theta)}{l''(\theta)}$$

Newton-Raphson  
方法

- 当  $\theta$  为向量时

$$\theta := \theta - H^{-1} \nabla_{\theta} l(\theta)$$

其中  $\nabla_{\theta} l(\theta)$  是  $l(\theta)$  偏导数向量,  $H$  是 Hessian 矩阵, 且

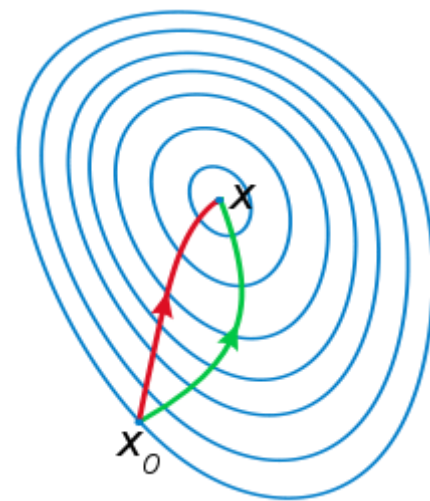
$$H_{ij} = \frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j}$$

牛顿法比批量梯度下降法快, 当参数量  $n$  不大时, 比较有效

# 求解对数似然函数的极大值点

- 牛顿法最大化logistic 回归的log 似然函数:  
Fisher scoring

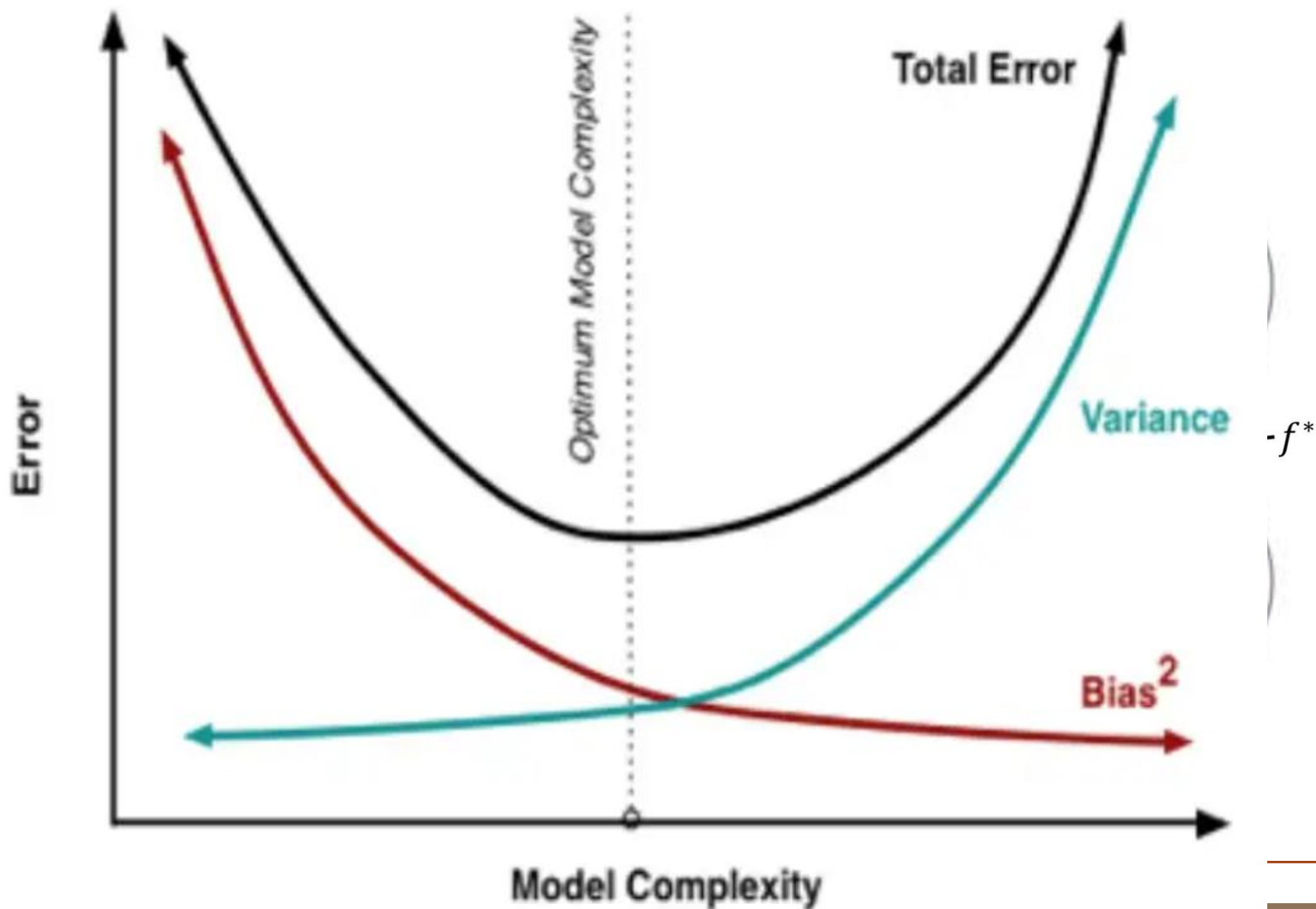
$$\theta := \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$



- When applied to maximize the logistic regression log likelihood function  $\ell(\theta)$ , the resulting method is also called **Fisher scoring**.

# 模型评估方法和性能评价指标

# 模型评估方法和性能评价指标



# 模型评估方法和性能评价指标

## 过拟合与欠拟合

- 过拟合指的是模型对于训练数据拟合程度过当的情况。当某个模型过度的学习训练数据中的细节和噪音,以至于模型在新的数据上表现很差。





# 模型评估方法—样本集的划分

## 训练集, 验证集, 测试集

### 训练集 ( $S_{\text{training}}$ ):

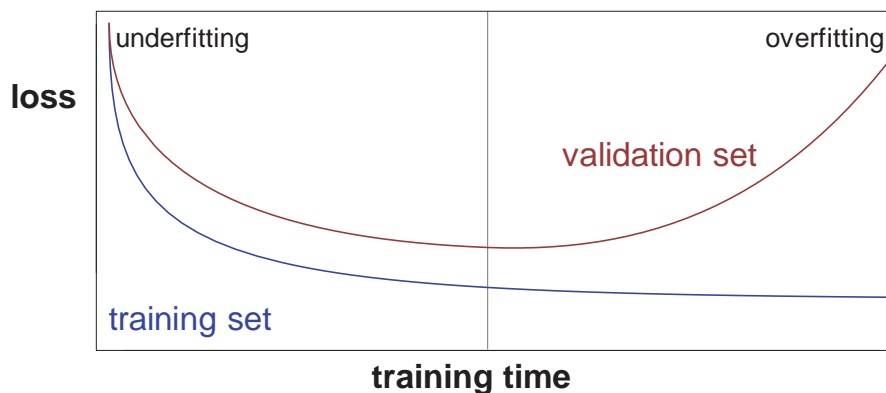
- 用于学习的数据集合
- 通常50 - 80 %的数据

### 验证集 ( $S_{\text{validation}}$ ):

- 用于设置和调整超参数的数据集合
- 通常10 - 20 %的数据

### 测试集 ( $S_{\text{test}}$ ):

- 一组用于评估完整训练模型性能的数据
- 在评估测试集性能之后, 不能进一步调优模型
- 通常10 - 30 %的数据



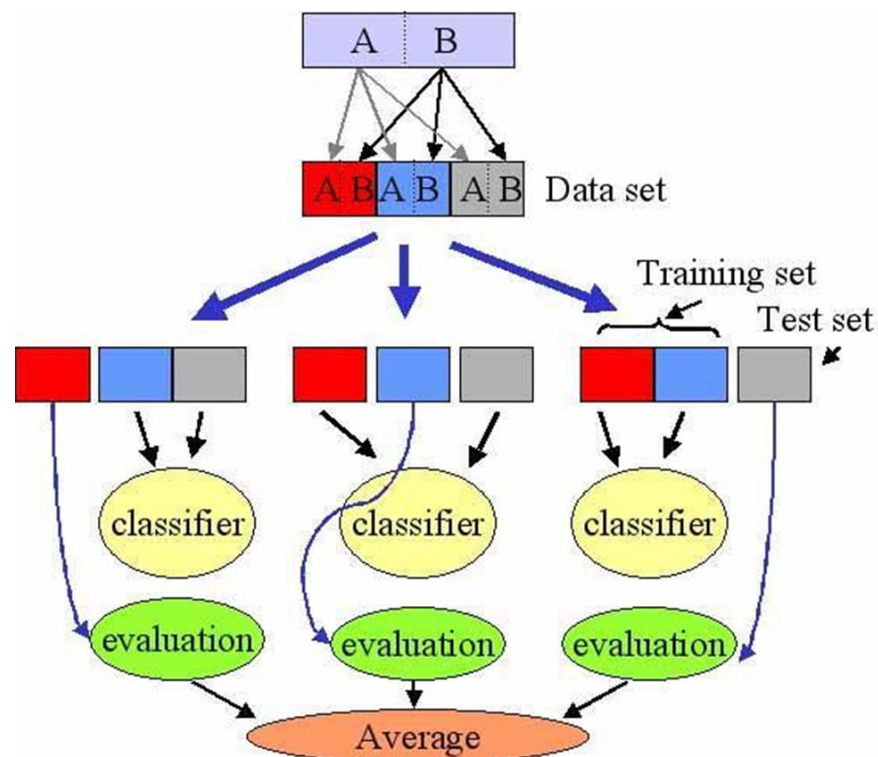
# 模型评估方法—样本集的划分

- 留出法

- 分层采样
- 2/3~4/5 training
- >30 validation

- 交叉验证

- K个互斥的子集
- 分层采样
- K-1 training
- 1 validation
- K=10
- K=样本数: 留一法 leave-one-out



# 模型评估方法

- 自助法 bootstrapping
  - 可重复采样 or 有放回的采样
  - 大约有1/3的测试集
  - 适用数据集规模较小
  - 对集成学习有利
  - 会引入估计偏差

# 性能评价指标

## 性能评价指标-分类

**准确率(Accuracy)**是指在分类中，分类正确的记录个数占总记录个数的比。

$$accuracy = \frac{n_{correct}}{n_{total}}$$

**召回率(Recall)**也叫查全率，是指在分类中样本中的正例有多少被预测正确了。

通常，准确率同召回率不一定一致。

### 1. 地震的预测

对于地震的预测，我们希望的是召回率非常高，也就是说每次地震我们都希望预测出来。

### 2. 疾病预测

对于疾病的预测，我们希望的是召回率非常高，也就是说每个疾病病人我们都希望预测出来。

# 性能评价指标

## 二分类效果评估方法

**准确率(Accuracy):** 分类正确的样本个数占所有样本个数的比例

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

**平均准确率(Average per-class accuracy):** 每个类别下的准确率的算术平均

$$average\_accuracy = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2}$$

**精确率(Precision):** 分类正确的正样本个数占预测结果中所有的正样本个数的比例

$$Precision = \frac{TP}{TP + FP}$$

**召回率(Recall):** 分类正确的正样本个数占正样本个数的比例

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

**F1-Score:** 精确率与召回率的调和平均值, 它的值更接近于Precision与Recall中较小的值

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

		预测情况	
		正例 (Positive)	反例 (Negative)
真实情况	正例 (True)	TP	FN
	反例 (False)	FP	TN

<https://blog.csdn.net/qq407790847>

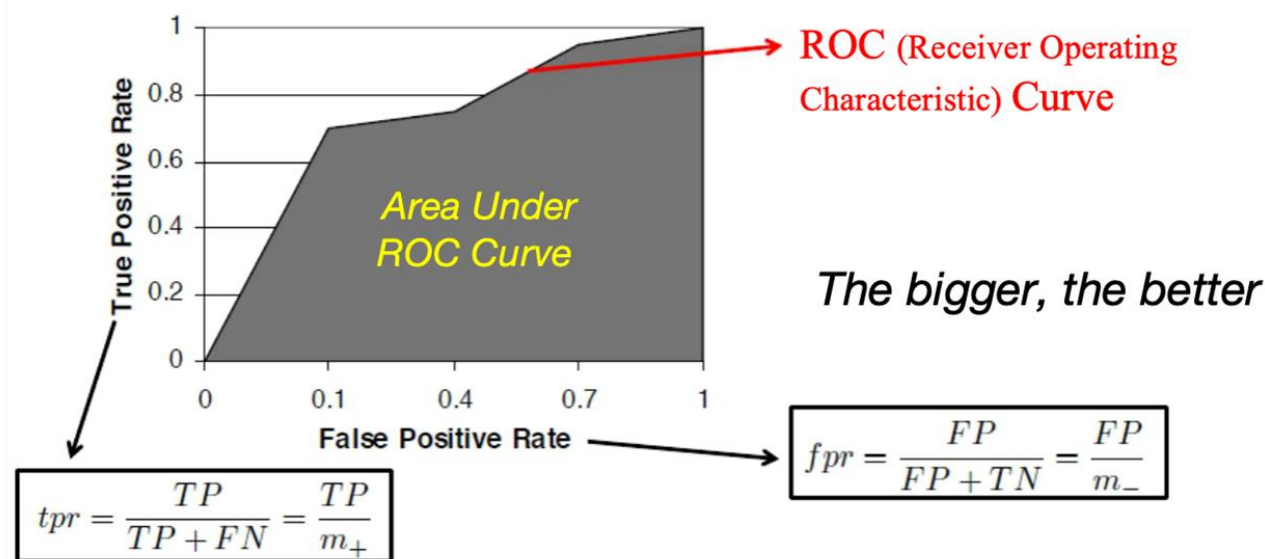
# 性能评价指标

## ● AUC (area under curve)

**AUC(Area under the Curve(Receiver Operating Characteristic, ROC))**

**ROC**: 纵轴: 真正例率TPR; 横轴: 假正例率FPR

**AUC**是ROC曲线下的面积。一般来说, 如果ROC是光滑的, 那么基本可以判断没有太大的overfitting, 这个时候调模型可以只看AUC, 面积越大一般认为模型越好。



# Acknowledgement

《机器学习方法》 李航

《机器学习》 李宏毅

《动手学深度学习》 李沐