

# 机器学习基础（一）

主讲：王皓 副研究员| 硕士导师

邮箱：wanghao3@ustc.edu.cn

主页：http://staff.ustc.edu.cn/~wanghao3

# 本章内容：机器学习概述

## ➤ 机器学习

- 基本概念
- 基本要素

## ➤ 示例：线性回归

- 定义
- 经验风险最小化
- 结构风险最小化
- 最大似然估计
- 最大后验估计

## ➤ 模型选择与评估

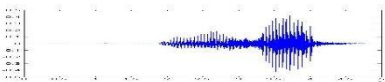
# 机器学习

---


# 机器学习 $\approx$ 构建一个映射函数

机器学习是对能通过经验自动改进的计算机算法的研究。  
——汤姆·米切尔


} 语音识别

$f(\text{  }) = \text{"你好"}$

} 图像识别

$f(\text{  }) = \text{"猫"}$

} 围棋

$f(\text{  }) = \text{"5-5" (落子位置)}$

} 对话系统

$f(\text{"你好"}) = \text{"今天天气真不错"}$

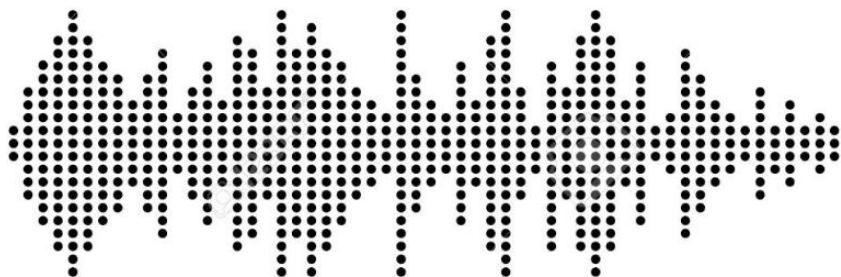
用户输入

机器

# 为什么要“机器学习”？

➤ 现实世界的问题都比较复杂

- 很难通过规则来手工实现

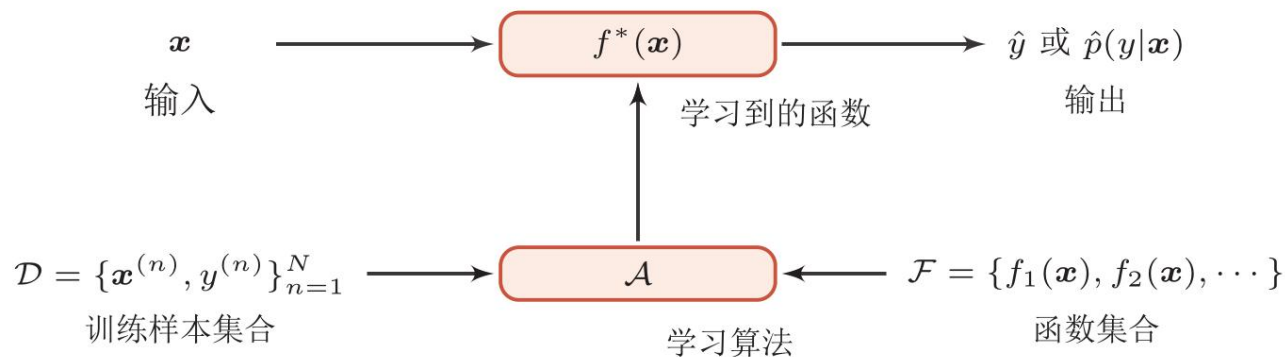


2	6	8	9	3	4	7	5	6
3	4	7	9	5	5	6	7	2
5	8	7	0	9	4	3	5	4
5	2	3	4	9	5	6	7	8

➤ 机器学习：从有限观测数据（**样本**）中寻找规律，并利用学习到的规律（**模型**）对未知或无法观测的数据进行**预测**。

# 机器学习基本概念

- 样本：特征（属性）+ 标签
  - 特征向量： $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$
  - 标签（连续值/离散值）
- 数据集：训练集（含验证集）+ 测试集
- 训练过程：寻找决策（预测）函数



假设 $\mathcal{D}$ 独立同分布  $p(\mathbf{x}, y)$

# 机器学习的三要素

## ➤ 模型

- 线性方法：
$$f(\mathbf{x}, \theta) = \mathbf{w}^T \mathbf{x} + b$$
- 广义线性方法：如果  $\phi(\mathbf{x})$  为可学习的非线性基函数， $f(\mathbf{x}, \theta)$  就等价于神经网络。
$$f(\mathbf{x}, \theta) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

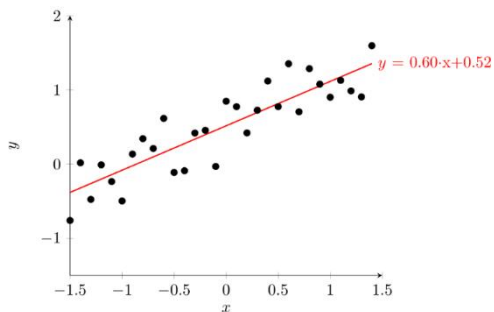
## ➤ 学习准则

- 风险最小化准则

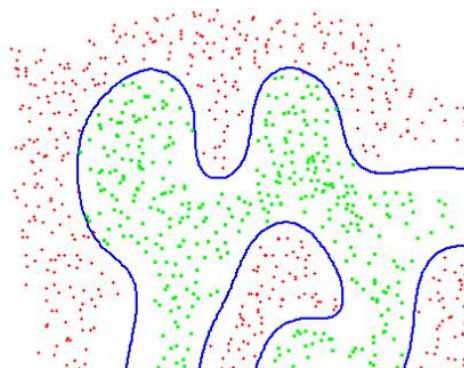
## ➤ 优化算法

- 最优化问题
- 梯度下降法

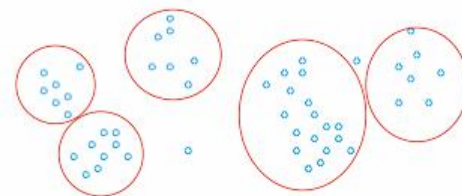
# 常见的机器学习问题



回归



分类



聚类



# 学习准则

➤ 训练集  $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$

- N个独立同分布的样本  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$
- 这些样本是从  $\mathcal{X}$  和  $\mathcal{Y}$  的联合空间中按照某个固定但未知分布  $p(\mathbf{x}, y)$  独立、随机产生的

损失函数

➤ 期望风险  $\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} [\mathcal{L}(f(\mathbf{x}), y)]$

- 期望风险未知，通过经验风险近似

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$

# 学习准则

## ➤ 损失函数

- 0-1 损失函数

$$\mathcal{L}(y, f(x, \theta)) = \begin{cases} 0 & \text{if } y = f(x, \theta) \\ 1 & \text{if } y \neq f(x, \theta) \end{cases}$$

- 平方损失函数

$$\mathcal{L}(y, \hat{y}) = (y - f(x, \theta))^2$$

- 交叉熵损失函数

对于C类别分类问题，可以用C维one-hot向量表示样本标签

$$\begin{aligned} \mathcal{L}(\mathbf{y}, f(\mathbf{x}; \theta)) &= -\mathbf{y}^\top \log f(\mathbf{x}; \theta) \\ &= -\sum_{c=1}^C y_c \log f_c(\mathbf{x}; \theta) \end{aligned}$$

- .....

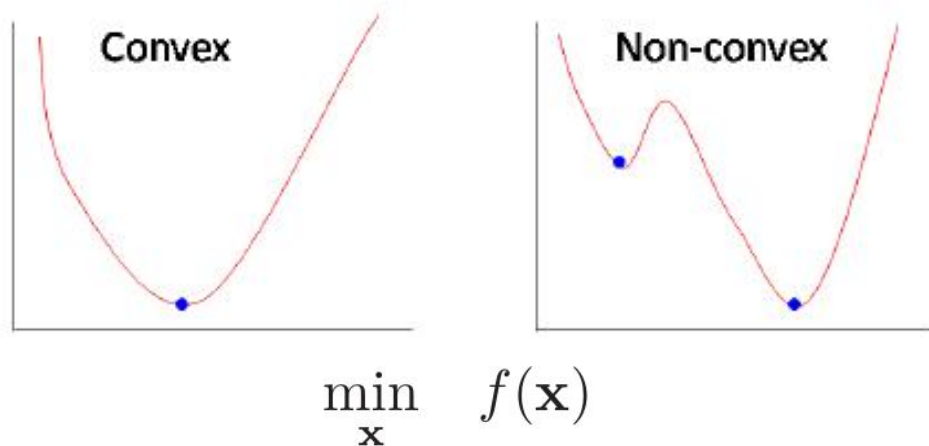
# 学习准则

## ➤ 经验风险最小化 (Empirical Risk Minimization, ERM) 准则

- 在选择合适的风险（损失）函数后，寻找一个参数 $\theta^*$ ，使得经验风险函数最小化。

$$\theta^* = \arg \min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta)$$

## ➤ 在确定了训练集 $\mathcal{D}$ 、假设空间 $\mathcal{F}$ 以及学习准则后，机器学习问题转化成为一个**最优化** (Optimization) 问题



# 优化算法

## ➤ 参数优化

- 模型  $f(x; \theta)$  中的  $\theta$  为 **参数**，可以通过 **优化算法** 进行 **学习**
- 例如：聚类任务中各簇的质心位置、神经网络的节点权重等

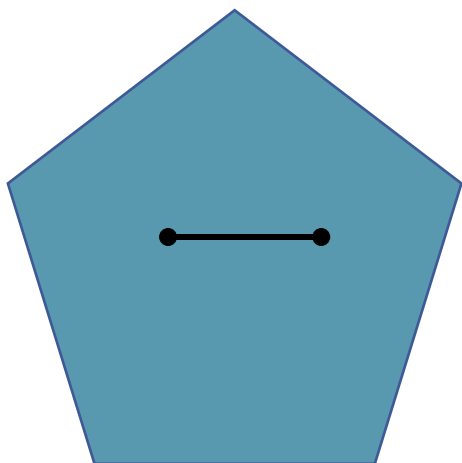
## ➤ 超参数优化

- **超参数** 用来定义模型结构或优化策略
- 例如：梯度下降法中的步长、神经网络的层数、支持向量机中的核函数等
- 超参数优化是机器学习的一个经验性很强的技术，通常是按照人的经验设定，或者通过搜索的方法对一组超参数组合进行不断试错调整

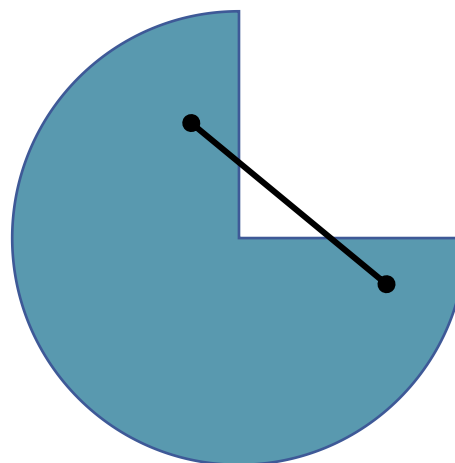
# 最优化的部分数学基础——凸函数

给定集合  $C \subseteq \mathbb{R}^n$ 。若  $\forall x, y \in C$  满足：  
 $\forall t \in [0, 1], tx + (1 - t)y \in C$   
那么集合  $C$  为凸集

凸集



非凸集



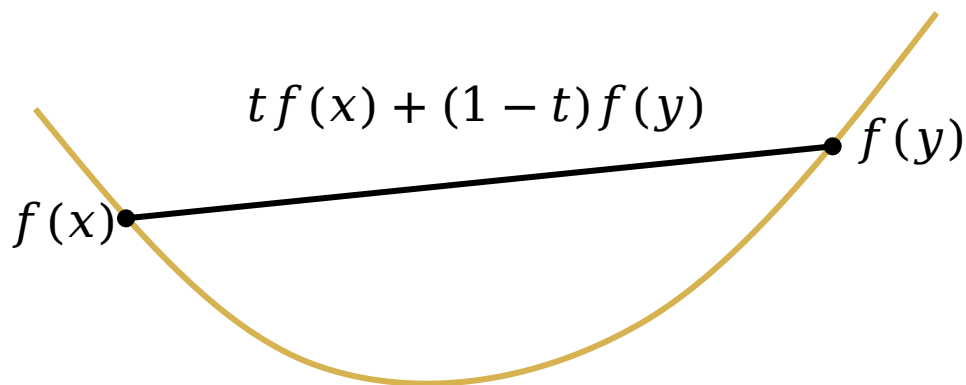
# 最优化的部分数学基础——凸函数

给定一个函数  $f: \mathbb{R}^n \mapsto \mathbb{R}$ 。

如果满足  $\text{dom}(f)$  是凸集而且  $\forall x, y \in \text{dom}(f)$ ,

$$\forall t \in [0, 1], f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y)$$

那么函数  $f$  是凸函数



# 最优化的部分数学基础——凸函数

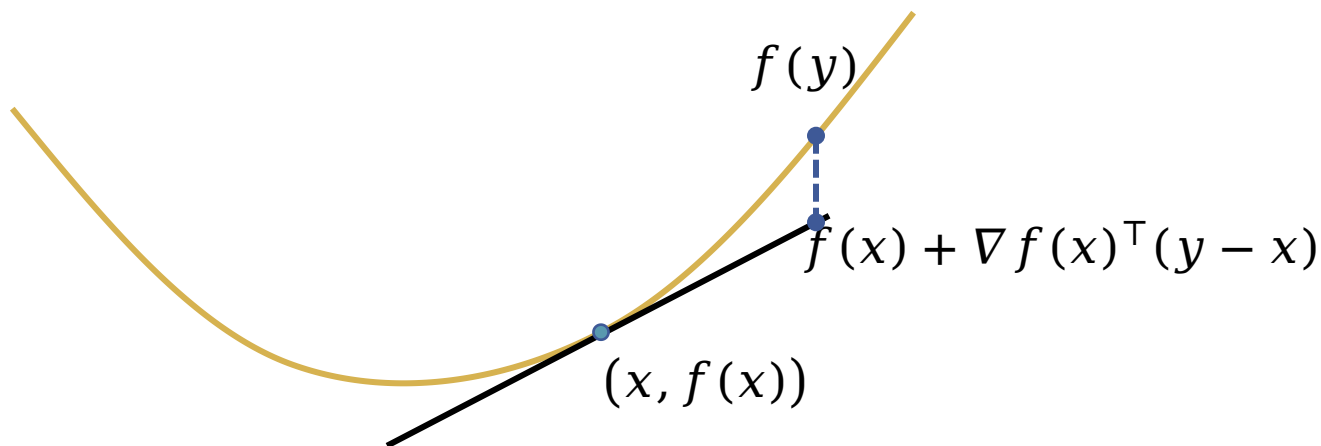
- 指数函数  $\exp(ax)$
- 负对数函数  $-\log(x)$
- 反射函数  $\mathbf{a}^\top \mathbf{x} + b$
- 二次函数  $\mathbf{x}^\top \mathbf{A} \mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$  ( $\mathbf{A}$  半正定)
- 范数  $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$
- 最大函数  $f(\mathbf{x}) = \max\{x_1, \dots, x_n\}$
- Softplus  $\log(1 + \exp(x))$
- LogSumExp  $\log(\sum_i \exp(x_i))$
- LogDeterminant  $-\log \det(X)$  在半正定矩阵定义域上

# 最优化的部分数学基础——凸函数

## 一阶条件

假设函数 $f$ 可微, 那么 $f$ 是凸函数当且仅当  $\forall x, y \in \text{dom}(f)$ ,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x)$$



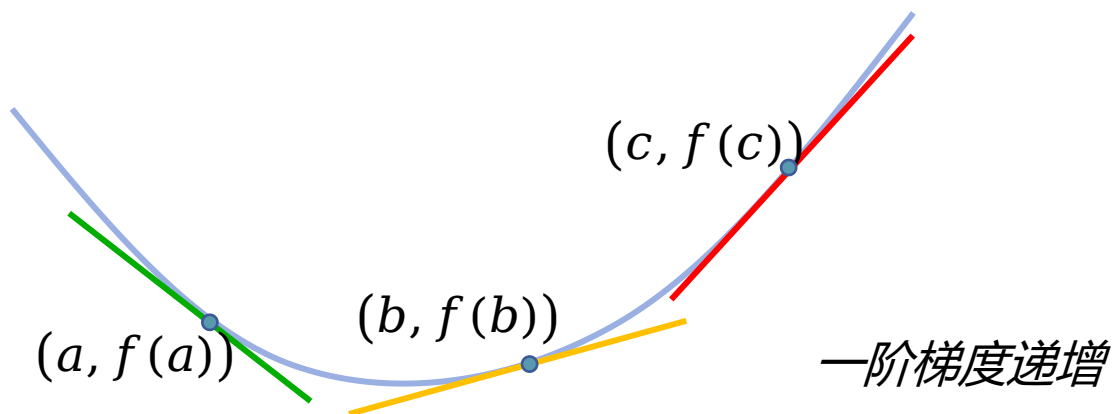


# 最优化的部分数学基础——凸函数

## 二阶条件

假设函数 $f$ 二阶可微，那么 $f$ 是凸函数当且仅当  $\forall x \in \text{dom}(f)$ ,

$$\nabla^2 f(x) \succeq 0, \text{ 即海森矩阵半正定}$$



# 最优化的部分数学基础——凸优化

## 凸优化问题

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{s.t. } & g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & \mathbf{a}_j^\top \mathbf{x} = b, \quad j = 1, 2, \dots, n \end{aligned}$$

其中  $f(\mathbf{x})$ ,  $g_i(\mathbf{x})$  是凸函数

➤ 凸优化中，局部最优等价于全局最优

假设函数  $f$  可微凸函数，那么  $\mathbf{x}$  是  $f$  的全局最优当且仅当，

$$\nabla f(\mathbf{x}) = 0$$

证明：因为  $\nabla f(\mathbf{x}) = 0$ ，所以

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) = f(\mathbf{x})$$

# 优化算法

➤ 无约束优化：梯度下降法（Gradient Descent）

➤ 目标：  $\min_{\mathbf{x}} f(\mathbf{x})$

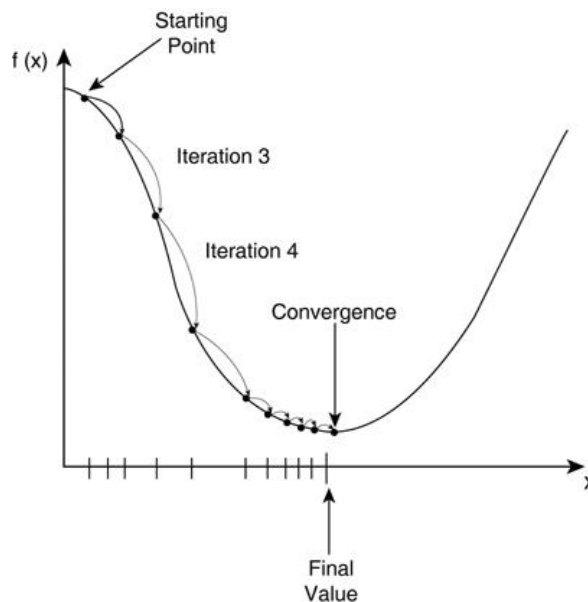
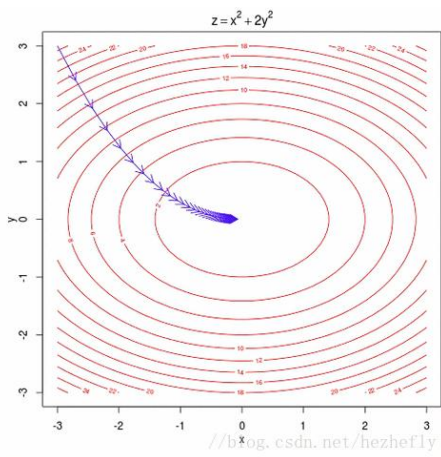
```
while  $\|\nabla f(\mathbf{x}_t)\| > \delta$   
do
```

```
     $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t -$ 
```

```
     $\alpha \nabla f(\mathbf{x}_t)$ 
```

```
end while
```

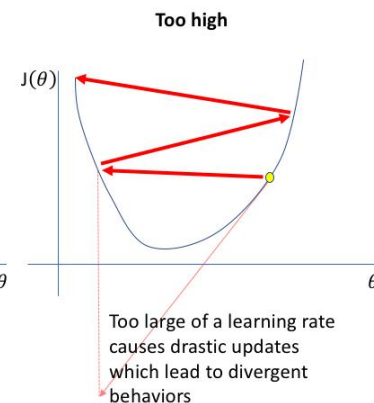
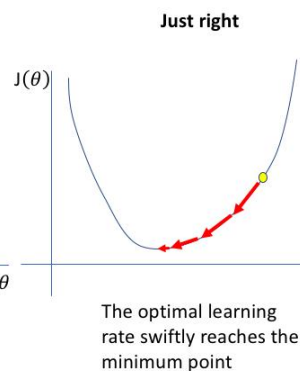
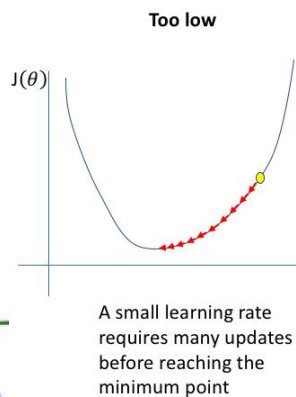
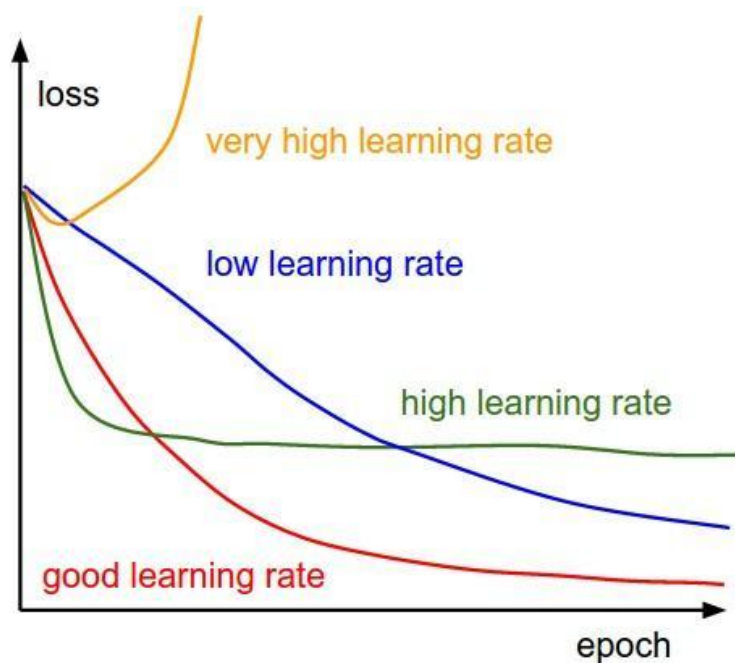
搜索步长  $\alpha$  中也叫作学习率  
(Learning Rate)



$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha \frac{\partial \mathcal{R}(\theta)}{\partial \theta_t} \\ &= \theta_t - \alpha \frac{1}{N} \sum_{i=1}^N \frac{\partial \mathcal{L}(\theta_t; x^{(i)}, y^{(i)})}{\partial \theta}.\end{aligned}$$

# 优化算法

学习率是十分重要的超参数！



# 优化算法

- 随机梯度下降法 (Stochastic Gradient Descent, SGD)  
也叫增量梯度下降, 每个样本都进行更新

---

## 算法 2.1: 随机梯度下降法

---

输入: 训练集  $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ , 验证集  $\mathcal{V}$ , 学习率  $\alpha$

1 随机初始化  $\theta$ ;

2 repeat

3     对训练集  $\mathcal{D}$  中的样本随机重排序;

4     **for**  $n = 1 \cdots N$  **do**

5         从训练集  $\mathcal{D}$  中选取样本  $(\mathbf{x}^{(n)}, y^{(n)})$ ;

       // 更新参数

6          $\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}(\theta; \mathbf{x}^{(n)}, y^{(n)})}{\partial \theta}$

7     **end**

8 **until** 模型  $f(\mathbf{x}; \theta)$  在验证集  $\mathcal{V}$  上的错误率不再下降;

输出:  $\theta$

---

# 优化算法

## ➤ 梯度下降法 vs 随机梯度下降法:

- 每次迭代的优化目标:

对所有样本的平均损失函数 vs 对单个样本的损失函数

- 收敛速度: 随机梯度下降快
- 针对非凸优化问题: 随机梯度下降更容易逃离局部最优点

## ➤ 小批量 (Mini-Batch) 梯度下降法

- 随机选取一小部分训练样本来计算梯度并更新参数
- 既可以兼顾随机梯度下降法的优点, 也可以提高训练效率

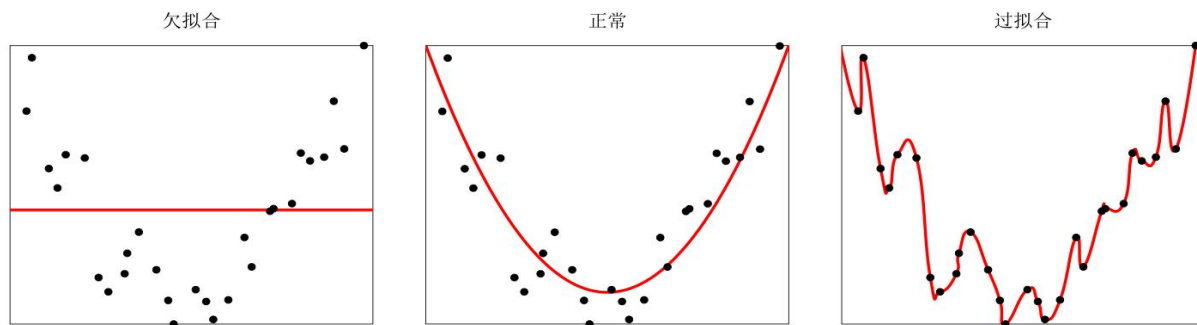
$$\theta_{t+1} \leftarrow \theta_t - \alpha \frac{1}{K} \sum_{(x,y) \in \mathcal{S}_t} \frac{\partial \mathcal{L}(y, f(x; \theta))}{\partial \theta}$$

$\mathcal{S}_t$  是随机选取的  $\mathcal{D}$  的子集;  
 $K$  通常设置为  $2^n$  且不会很大

# 机器学习 = 优化?

机器学习 = 优化?

NO!



**过拟合**：经验风险最小化原则很容易导致模型在训练集上错误率很低，但是在未知数据上错误率很高。  
过拟合问题往往是由于**训练数据少**和**噪声**等原因造成的。

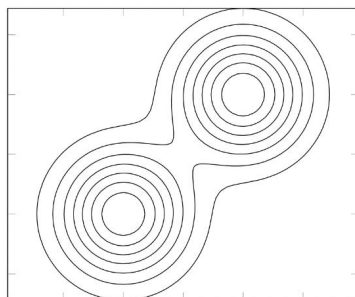
反之有**欠拟合**：模型不能很好地拟合训练数据，在训练集上错误率比较高。  
欠拟合一般是由于**模型能力不足**、**训练不充分**造成的。

# 泛化错误

期望风险

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} [\mathcal{L}(f(\mathbf{x}), y)],$$

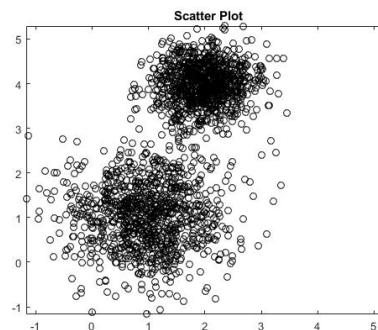
真实分布  $p_r$



$\neq$

经验风险

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$



$$\mathcal{G}_{\mathcal{D}}(f) = \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f)$$

泛化错误



# 如何减少泛化错误？

优化

经验风险最小

正则化

降低模型复杂度



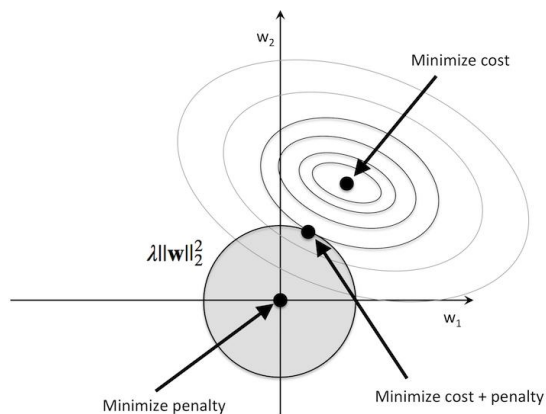
结构风险最小化！

# 正则化 (regularization)

所有损害优化的方法都是正则化

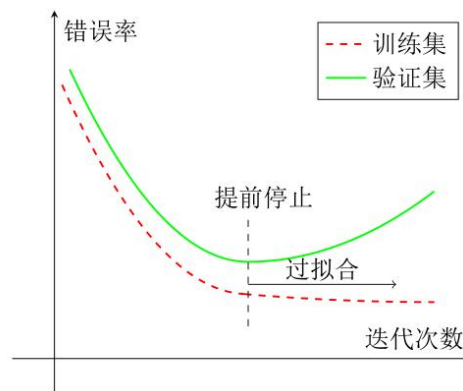
增加优化约束

L1/L2约束、数据增强



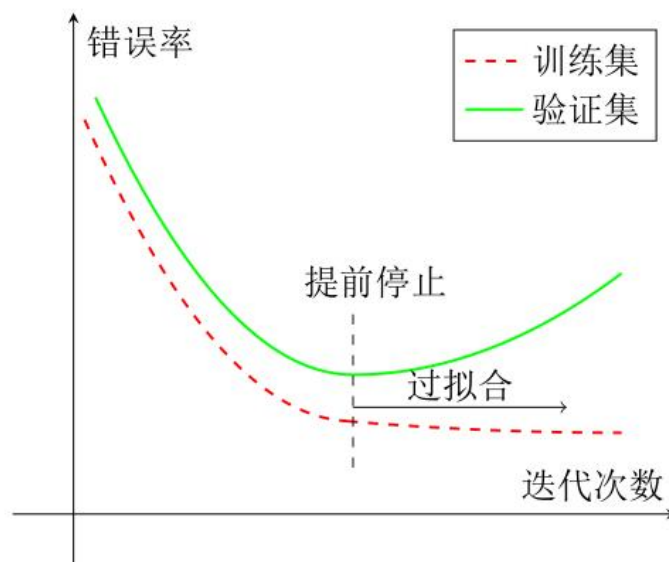
干扰优化过程

权重衰减、随机梯度下降、提前停止



# 提前停止

- 我们使用一个**验证集** (Validation Dataset) 来测试每一次迭代的参数在验证集上是否最优。如果在验证集上的错误率不再下降，就停止迭代。



# 线性回归

---

# 线性回归 (Linear Regression)

➤模型:

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b$$

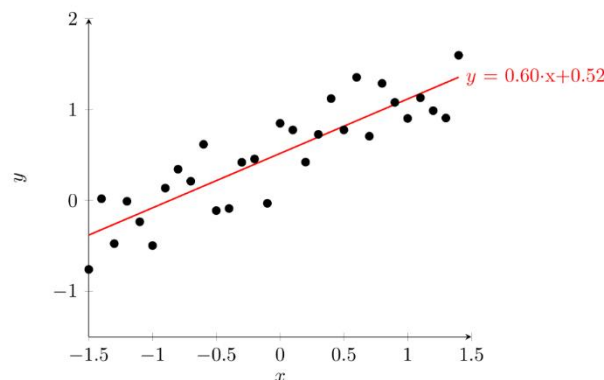
• 增广权重向量和增广特征向量

$$\hat{\mathbf{x}} = \mathbf{x} \oplus 1 \triangleq \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 1 \end{bmatrix},$$

$$\hat{\mathbf{w}} = \mathbf{w} \oplus b \triangleq \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_k \\ b \end{bmatrix},$$



$$f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{\mathbf{w}}^T \hat{\mathbf{x}},$$



# 参数估计方法

➤ 任务：  $f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{\mathbf{w}}^T \hat{\mathbf{x}},$

给定一组包含  $N$  个训练样本的训练集  $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$

学习一个最优的线性回归的模型参数  $\hat{\mathbf{w}}$

➤ 方法：

- 经验风险最小化（最小二乘法）
- 结构风险最小化（岭回归）
- 最大似然估计
- 最大后验估计

# 矩阵微积分

- 标量关于向量的偏导数

$$\frac{\partial y}{\partial \mathbf{x}} = \left[ \frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_M} \right]^\top$$

- 向量关于向量的偏导数

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_N}{\partial x_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial y_1}{\partial x_M} & \dots & \frac{\partial y_N}{\partial x_M} \end{bmatrix}$$

- 向量函数及其导数

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \mathbf{x}} &= \mathbf{I}, \\ \frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} &= \mathbf{A}^\top, \\ \frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} &= \mathbf{A} \end{aligned}$$

# 经验风险最小化

- 模型:  $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$
- (不失一般性,  $\mathbf{w}$  和  $\mathbf{x}$  都表示增广后的向量)

- 学习准则: 平方损失函数最小化

$$\begin{aligned}\mathcal{R}(\mathbf{w}) &= \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(\mathbf{x}^{(n)}; \mathbf{w})) \\ &= \frac{1}{2} \sum_{n=1}^N \left( y^{(n)} - \mathbf{w}^\top \mathbf{x}^{(n)} \right)^2 \\ &= \frac{1}{2} \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2,\end{aligned}$$

$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ x_D^{(1)} & x_D^{(2)} & \cdots & x_D^{(N)} \\ 1 & 1 & \cdots & 1 \end{bmatrix}$

$\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]^\top \in \mathbb{R}^N$



# 经验风险最小化

➤ 优化:

风险函数  $\mathcal{R}(\mathbf{w})$  是关于  $\mathbf{w}$  的凸函数, 其对  $\mathbf{w}$  的偏导数为

$$\begin{aligned}\frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2} \frac{\partial \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2}{\partial \mathbf{w}} \\ &= -\mathbf{X}(\mathbf{y} - \mathbf{X}^\top \mathbf{w}),\end{aligned}$$

令  $\frac{\partial}{\partial \mathbf{w}} \mathcal{R}(\mathbf{w}) = 0$ , 得到最优的参数  $\mathbf{w}^*$  为

$$\begin{aligned}\mathbf{w}^* &= (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{y} \\ &= \left( \sum_{n=1}^N \mathbf{x}^{(n)} (\mathbf{x}^{(n)})^\top \right)^{-1} \left( \sum_{n=1}^N \mathbf{x}^{(n)} y^{(n)} \right)\end{aligned}$$

# 结构风险最小化

## ➤ 结构风险最小化准则

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}^T \mathbf{w}\|^2 + \frac{1}{2} \lambda \|\mathbf{w}\|^2,$$

## ➤ 得到

$$\mathbf{w}^* = (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{y},$$

- **岭回归** (Ridge Regression) : 给 $\mathbf{X}\mathbf{X}^T$ 的对角线元素都加上一个常数 $\lambda$ 使得 $(\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})$ 满秩, 即其行列式不为0

# 概率的一些基本概念

## ➤ 概率 (Probability)

- 一个随机事件发生的可能性大小，为0到1之间的实数。

## ➤ 随机变量 (Random Variable)

- 比如随机掷一个骰子，得到的点数就可以看成一个随机变量 $X$ ，其取值为 $\{1, 2, 3, 4, 5, 6\}$ 。

## ➤ 概率分布 (Probability Distribution)

- 一个随机变量 $X$ 取每种可能值的概率

$$P(X = x_i) = p(x_i), \quad \forall i \in \{1, \dots, n\}.$$

- 并满足 
$$\sum_{i=1}^n p(x_i) = 1,$$

$$p(x_i) \geq 0, \quad \forall i \in \{1, \dots, n\}.$$

# 概率的一些基本概念

## ➤ 伯努利分布 (Bernoulli Distribution)

- 在一次试验中，事件A出现的概率为 $\mu$ ，不出现的概率为 $1 - \mu$ 。若用变量 $x$ 表示事件A出现的次数，则 $x$ 的取值为0和1，其相应的分布为

$$p(x) = \mu^x (1 - \mu)^{(1-x)}$$

## ➤ 二项分布 (Binomial Distribution)

- 在 $n$ 次伯努利分布中，若以变量 $X$ 表示事件A出现的次数，则 $X$ 的取值为 $\{0, \dots, n\}$ ，其相应的分布

$$P(X = k) = \binom{n}{k} \mu^k (1 - \mu)^{n-k}, \quad k = 1 \cdots n$$

二项式系数，表示从 $n$ 个元素中取出 $k$ 个元素而不考虑其顺序的组合的总数。

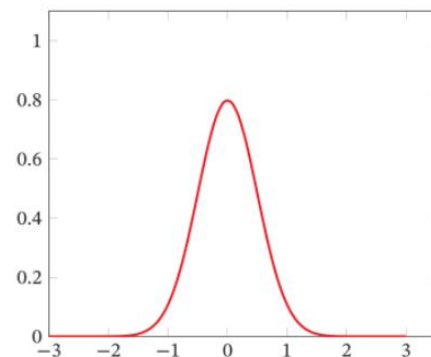
# 概率的一些基本概念

- 连续随机变量 $Y$ 的概率分布一般用概率密度函数 (Probability Density Function, PDF)  $p(x)$  来描述。

$$\int_{-\infty}^{+\infty} p(x)dx = 1$$

- 高斯分布 (Gaussian Distribution)

$$X \sim \mathcal{N}(\mu, \sigma^2) \quad p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



# 概率的一些基本概念

## ➤ 条件概率 (Conditional Probability)

- 对于离散随机向量 $(X, Y)$ , 已知 $X = x$ 的条件下, 随机变量 $Y = y$ 的条件概率为:

$$p(y|x) = P(Y = y|X = x) = \frac{p(x, y)}{p(x)}$$

## ➤ 贝叶斯公式

- 两个条件概率 $p(y|x)$ 和 $p(x|y)$ 之间的关系

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

- 似然 (Likelihood) :  $p(w|X) \propto p(X|w)p(w)$

后验

似然

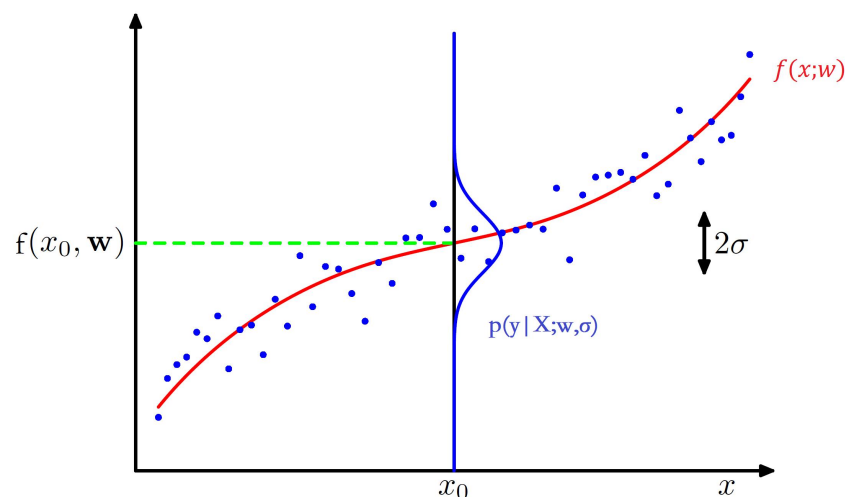
先验

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

# 从概率角度来看线性回归

- 假设标签 $y$ 为一个随机变量，其服从均值为 $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$ ，方差为 $\sigma^2$ 的高斯分布。

$$\begin{aligned} p(y|\mathbf{x}; \mathbf{w}, \sigma) &= \mathcal{N}(y; \mathbf{w}^T \mathbf{x}, \sigma^2) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \mathbf{w}^T \mathbf{x})^2}{2\sigma^2}\right). \end{aligned}$$



# 最大似然估计

➤ 参数  $\mathbf{w}$  在训练集  $\mathcal{D}$  上的似然函数 (Likelihood) 为

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}; \mathbf{w}, \sigma) &= \prod_{n=1}^N p(y^{(n)}|\mathbf{x}^{(n)}; \mathbf{w}, \sigma) \\ &= \prod_{n=1}^N \mathcal{N}(y^{(n)}; \mathbf{w}^\top \mathbf{x}^{(n)}, \sigma^2) \end{aligned}$$

➤ 最大似然估计 (Maximum Likelihood Estimate, MLE)

- 是指找到一组参数  $\mathbf{w}$  使得似然函数  $p(\mathbf{y}|\mathbf{X}; \mathbf{w}, \sigma)$  最大

$$\text{令 } \frac{\partial \log p(\mathbf{y}|\mathbf{X}; \mathbf{w}, \sigma)}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w}^{ML} = (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{y}.$$

与最小二乘法的解相同



# 最大后验估计

➤ 为了避免过拟合，给参数  $\mathbf{w}$  加一些先验知识：

- 假设  $\mathbf{w}$  为一个随机向量，并服从一个先验分布  $p(\mathbf{w}; \nu) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \nu^2 I)$
- 由贝叶斯公式，得到后验分布：

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}; \nu, \sigma) = \frac{p(\mathbf{w}, \mathbf{y}|\mathbf{X}; \nu, \sigma)}{\sum_{\mathbf{w}} p(\mathbf{w}, \mathbf{y}|\mathbf{X}; \nu, \sigma)}$$
$$\propto p(\mathbf{y}|\mathbf{X}, \mathbf{w}; \sigma) p(\mathbf{w}; \nu),$$

后验	似然	先验
posterior	likelihood	prior

➤ 最大后验估计 (Maximum A Posteriori Estimation, MAP)

- 最优参数为后验分布中概率密度最高的参数：

$$\mathbf{w}^{MAP} = \arg \max_{\mathbf{w}} p(\mathbf{y}|\mathbf{X}, \mathbf{w}; \sigma) p(\mathbf{w}; \nu)$$

# 最大后验估计

➤ 令似然函数为高斯密度函数：

$$\begin{aligned} p(\mathbf{y}|X; \mathbf{w}, \sigma) &= \prod_{n=1}^N p(y^{(n)}|\mathbf{x}^{(n)}; \mathbf{w}, \sigma) \\ &= \prod_{n=1}^N \mathcal{N}(y^{(n)}; \mathbf{w}^\top \mathbf{x}^{(n)}, \sigma^2) \end{aligned}$$

➤ 则后验分布的对数：

$$\begin{aligned} \log p(\mathbf{w}|X, \mathbf{y}; \nu, \sigma) &\propto \log p(\mathbf{y}|X, \mathbf{w}; \sigma) + \log p(\mathbf{w}; \nu) \\ &\propto -\frac{1}{2\sigma^2} \sum_{n=1}^N \left( y^{(n)} - \mathbf{w}^\top \mathbf{x}^{(n)} \right)^2 - \frac{1}{2\nu^2} \mathbf{w}^\top \mathbf{w}, \\ &= -\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2 - \frac{1}{2\nu^2} \mathbf{w}^\top \mathbf{w}. \end{aligned}$$

最大后验概率等价于平方损失的结构风险最小化，  
其中正则化系数  $\lambda = \sigma^2/\nu^2$

# 总结

	无先验	引入先验
平方误差	经验风险最小化	结构风险最小化
概率	最大似然估计	最大后验估计

$$\mathbf{w}^{ML} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{y}$$

$$\mathbf{w}^* = (\mathbf{X}\mathbf{X}^\top + \lambda I)^{-1}\mathbf{X}\mathbf{y}$$

# 常见的机器学习类型

	监督学习	无监督学习	强化学习
训练样本	训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$	训练集 $\{\mathbf{x}^n\}_{n=1}^N$	智能体和环境交互的 轨迹 $\tau$ 和累积奖励 $G_\tau$
优化目标	$y = f(\mathbf{x})$ 或 $p(y \mathbf{x})$	$p(\mathbf{x})$ 或带隐变量 $\mathbf{z}$ 的 $p(\mathbf{x} \mathbf{z})$	期望总回报 $\mathbb{E}_\tau[G_\tau]$
学习准则	期望风险最小化 最大似然估计	最大似然估计 最小重构错误	策略评估 策略改进

# 模型选择与评估

---

# 模型选择与评估

## ➤ 模型选择

- 给定一个数据集，如何根据“泛化”能力，选出最好的模型或选出最好的参数配置

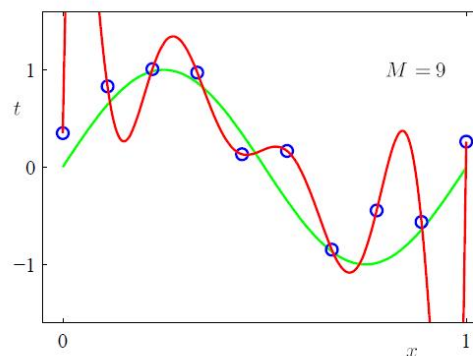
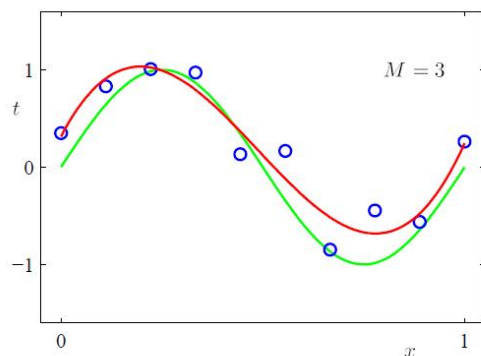
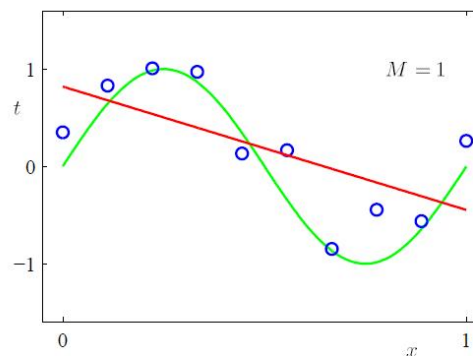
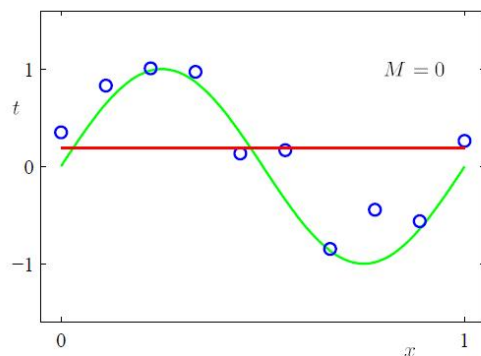
## ➤ 模型评估

- 给定一个数据集，如何估计一个模型的“泛化”能力？

# 一个例子：多项式曲线拟合

模型  $y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M$

损失函数  $E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$

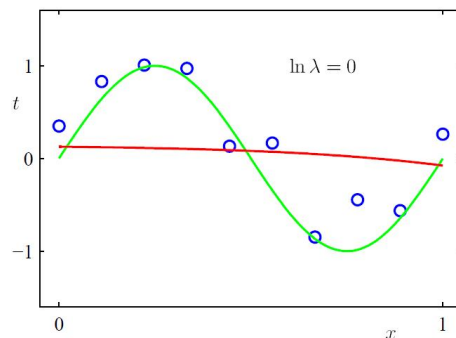
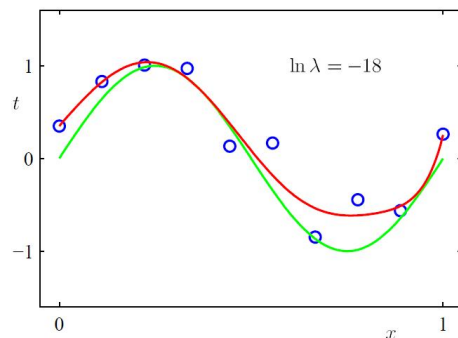


# 一个例子：多项式曲线拟合

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

引入正则化项

当 $M = 9$ :



	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^*$	0.35	0.35	0.13
$w_1^*$	232.37	4.74	-0.05
$w_2^*$	-5321.83	-0.77	-0.06
$w_3^*$	48568.31	-31.97	-0.05
$w_4^*$	-231639.30	-3.89	-0.03
$w_5^*$	640042.26	55.28	-0.02
$w_6^*$	-1061800.52	41.32	-0.01
$w_7^*$	1042400.18	-45.95	-0.00
$w_8^*$	-557682.99	-91.53	0.00
$w_9^*$	125201.43	72.68	0.01



# 如何选择一个合适的模型？

## ➤ 模型选择

- 拟合能力强的模型一般复杂度会比较高，容易**过拟合**
- 如果限制模型复杂度，降低拟合能力，可能会**欠拟合**

## ➤ 偏差与方差分解

- 期望误差可以分解为**偏差 + 方差 + 噪声**

$$\mathcal{R}(f) = (\text{bias})^2 + \text{variance} + \varepsilon.$$

$$\mathbb{E}_{\mathbf{x}} \left[ \left( \mathbb{E}_{\mathcal{D}} [f_{\mathcal{D}}(\mathbf{x})] - f^*(\mathbf{x}) \right)^2 \right]$$

学习算法本身的拟合能力

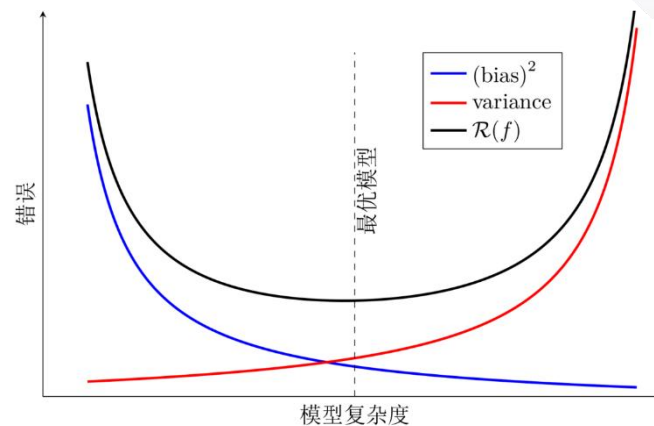
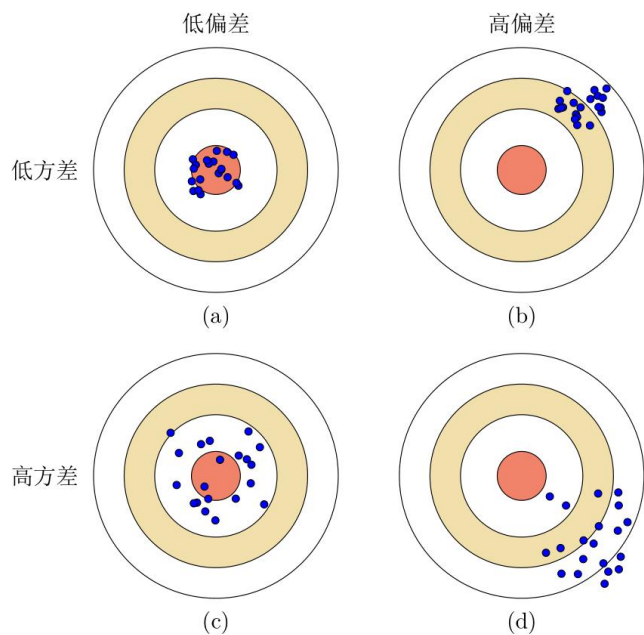
$$\mathbb{E}_{\mathbf{x}} \left[ \mathbb{E}_{\mathcal{D}} \left[ \left( f_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}} [f_{\mathcal{D}}(\mathbf{x})] \right)^2 \right] \right]$$

数据扰动所造成的影响

$$\mathbb{E}_{(\mathbf{x}, y) \sim p_r(\mathbf{x}, y)} \left[ \left( y - f^*(\mathbf{x}) \right)^2 \right]$$

学习问题本身的难度

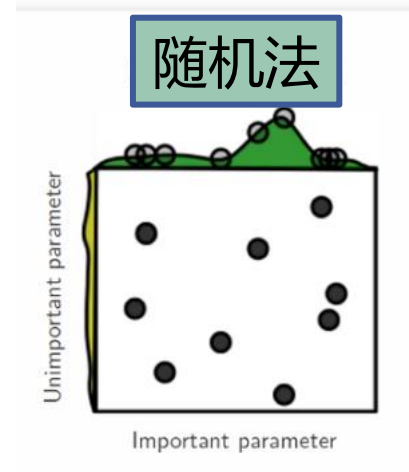
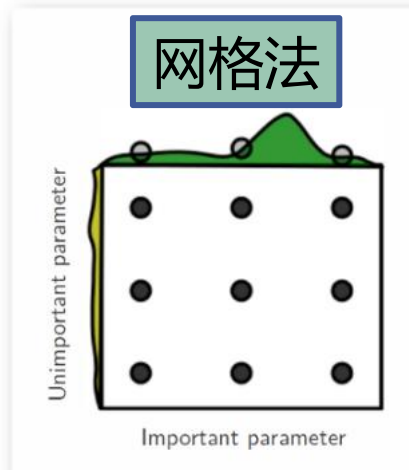
# 模型选择：偏差与方差



集成模型：有效的降低方差的方法

# 模型选择：调参

- 大多数学习算法都有些超参需要设定，不同参数设置，导致学得模型的性能有显著差别
- 模型选择中超参数配置的设定过程称为**调参**
- 调参的一般过程：
  - 将训练集划分为训练集和验证集
  - 通过**网格法**或**随机法**进行参数搜索，计算出验证集上的误差
  - 选出最佳的参数配置，在训练集上重新训练
  - 如果在验证集上的误差不再下降，则选定最优超参数组合



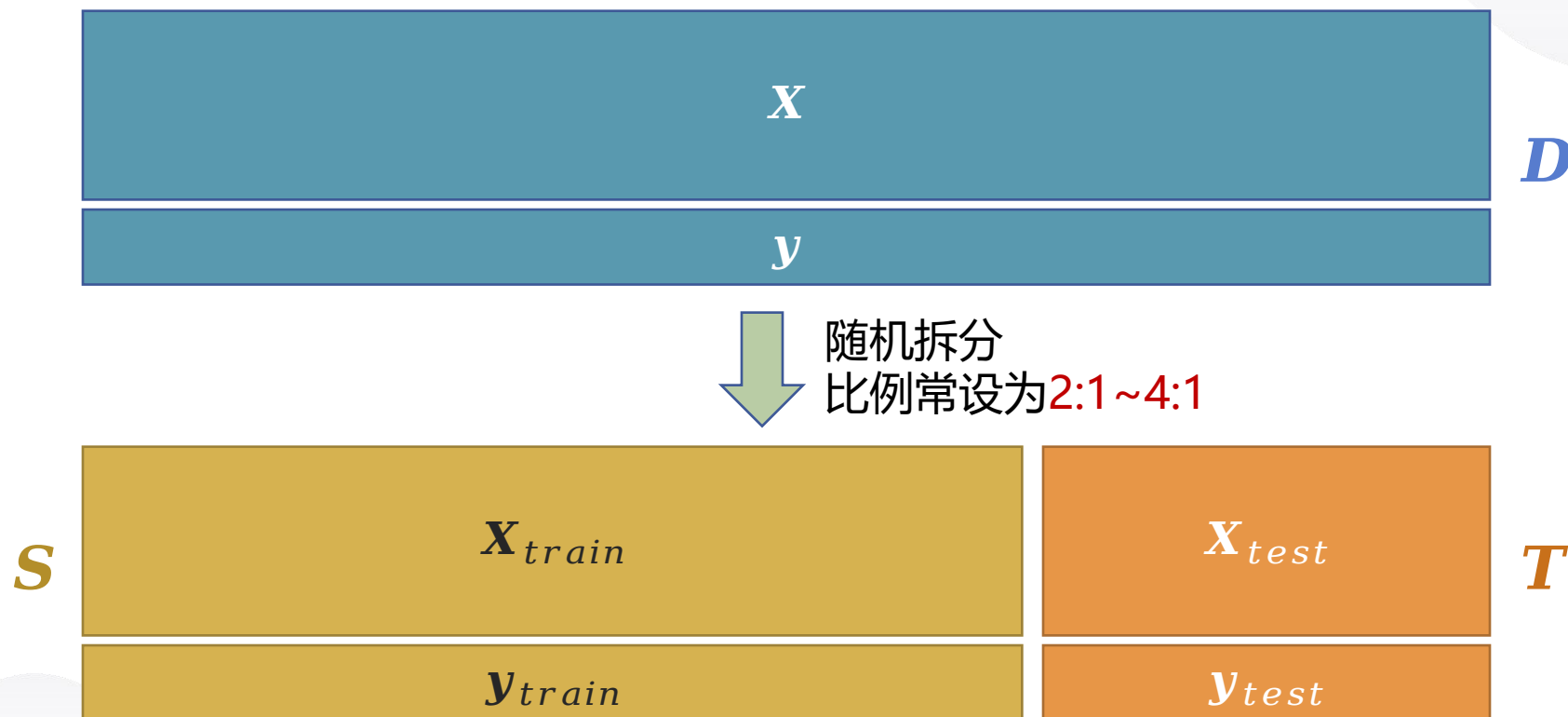
# 模型评估：经验误差

- 误差：样本真实输出与预测输出之间的差异，可以是错误率
  - 训练误差：训练集上的误差，也称经验误差
  - 测试误差：测试集上的误差
  - 泛化误差：除训练集外的所有样本上的误差
- 需要一个测试集来测试学习器对新样本的判别能力
- 假设测试集是从样本真实分布中独立采样获得，以测试集上的测试误差作为泛化误差的近似

测试集要和训练集中的样本尽量互斥，即测试样本尽量不在训练集中出现、未在训练集中使用过

# 评估方法——留出法 (hold-out)

- 假设数据集为  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$

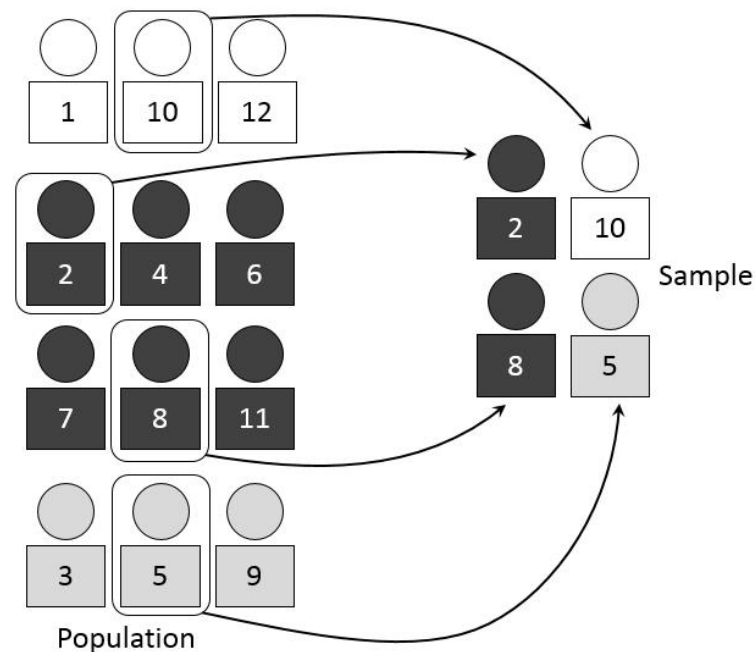
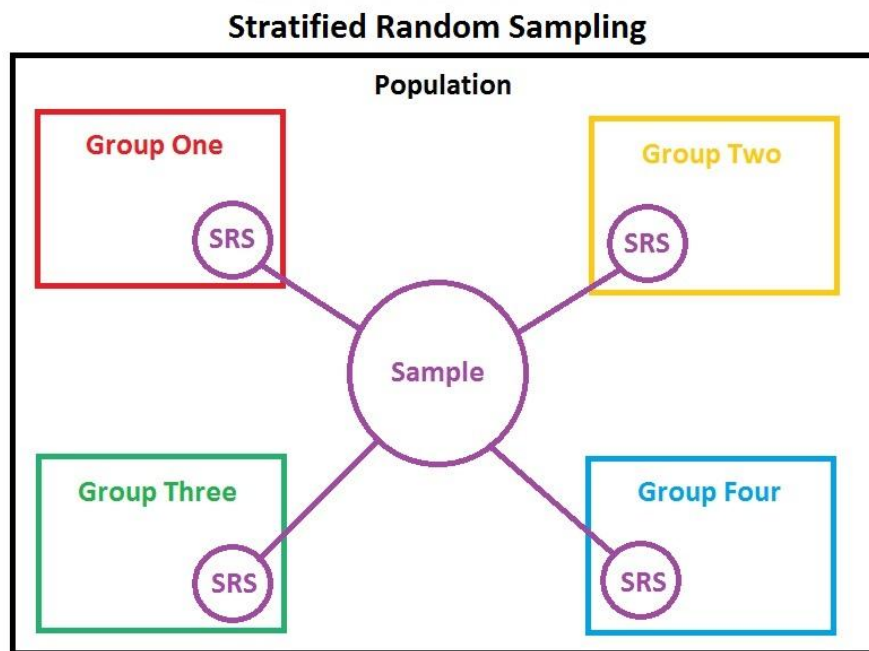


满足  $D = S \cup T$  且  $S \cap T = \phi$

# 评估方法——留出法 (hold-out)

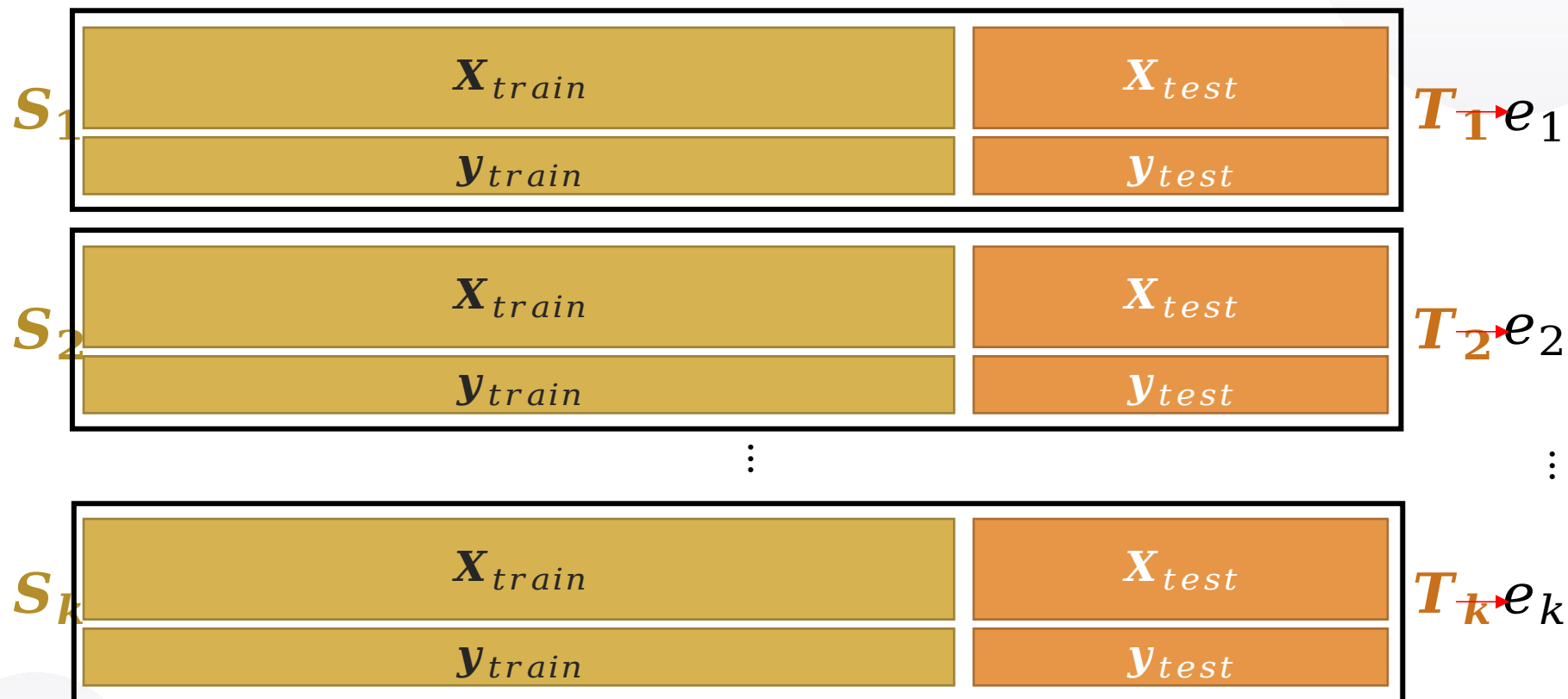
➤ **分层采样**：保持数据分布的一致性

- 例如：分类任务至少要保持样本类别比例相似



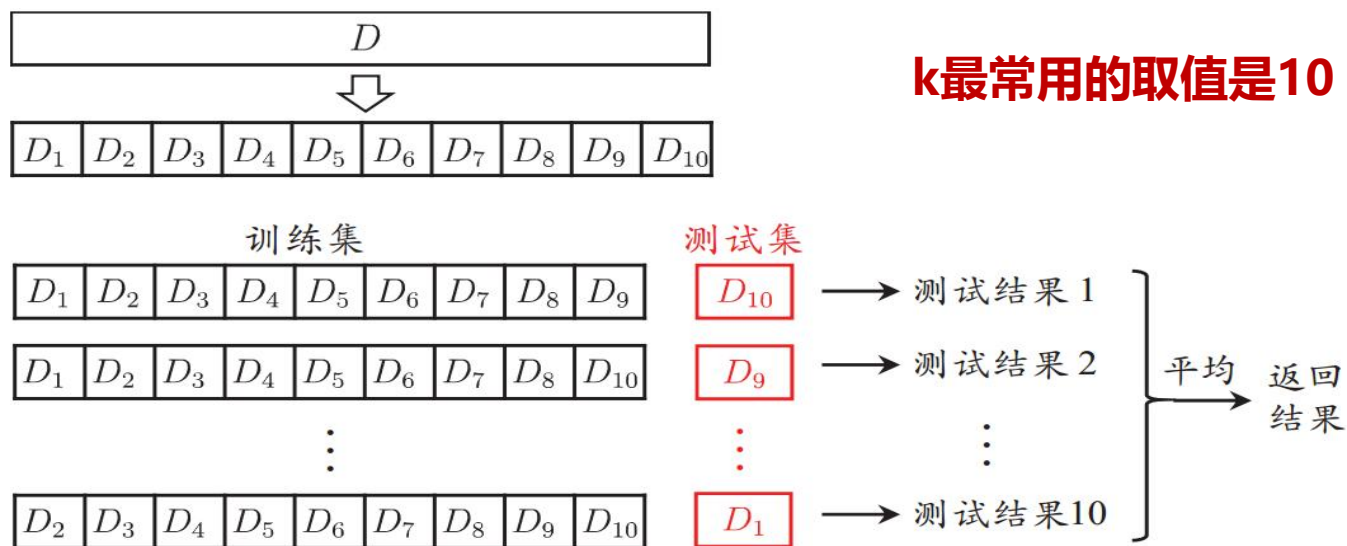
# 评估方法——留出法 (hold-out)

➤ 若干次随机划分、重复实验评估后取平均值



# 评估方法——交叉验证法

1. 将数据集分层采样划分为k个大小相似的互斥子集
2. 每次用k-1个子集的并集作为训练集，余下的子集作为测试集
3. 最终返回k个测试结果的均值



10 折交叉验证示意图



# 评估方法——交叉验证法

- 类似留出法，将数据集D划分为k个子集存在多种划分方式
  - 为了减小因样本划分不同而引入的差别，k折交叉验证通常随机使用不同的划分重复p次，最终的评估结果是这p次k折交叉验证结果的均值
  - 例如常见的“10次10折交叉验证”
- 当 $k = m$ （每个样本一个集合），得到留一法
  - 不受随机样本划分方式的影响
  - 结果往往比较准确
  - 当数据集比较大时，计算开销难以忍受

# 评估方法——自助法

- 减少训练样本规模不同造成的影响，同时较高效进行实验估计
- 以自助采样法为基础，对数据集  $D$  有放回采样  $m$  次得到训练集  $D'$ ,  $D \setminus D'$  用做测试集。

样本在  $m$  次采样中始终不被采样到的概率  $= \lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$

- 实际模型与预期模型都使用  $m$  个训练样本
- 约有1/3的样本没在训练集中出现
- 从初始数据集中产生多个不同的训练集，对集成学习有很大的好处
- 自助法在数据集较小、难以有效划分训练/测试集时很有用
- 由于改变了数据集分布可能引入估计偏差，在数据量足够时，留出法和交叉验证法更常用。

# 模型评估：性能度量

- 性能度量是衡量模型泛化能力的评价标准
- 反映任务需求，使用不同性能度量往往会导致不同的评判结果
- 在预测任务中，给定样例集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ，评估学习器的性能  $f$  也即把预测结果  $f(\mathbf{x})$  和真实标记比较

回归任务最常用的性能度量是 “均方误差”

$$E(f, D) = \frac{1}{m} \sum_m (f(\mathbf{x}_i) - y_i)^2$$

假设知道数据的分布，那么均方误差表达为

$$E(f, D) = \int_{\mathbf{x} \sim D} (f(\mathbf{x}) - y)^2 p(\mathbf{x}) d\mathbf{x}$$

# 性能度量——错误率和精度

➤ 对于分类任务,错误率和精度是最常用的两种性能度量:

错误率: 分错样本占样本总数的比例

$$E(f, D) = \frac{1}{m} \sum_i \mathbb{I}(f(\mathbf{x}_i) \neq y_i)$$

精度: 分对样本占样本总数的比率

$$acc(f, D) = \frac{1}{m} \sum_i \mathbb{I}(f(\mathbf{x}_i) = y_i)$$

$$acc(f, D) = 1 - E(f, D)$$

# 性能度量——查准率和查全率

- 错误率和精度虽然常用，但不能满足所有任务需求
  - 挑出的西瓜中有多少比例是好瓜，有多少比例的好瓜被挑选出来
  - 信息检索等场景经常需要衡量正例被预测出来的比率或者预测出来的正例中正确的比率
- 查准率和查全率比错误率和精度更适合

混淆矩阵 (confusion matrix)

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

查全率 (Recall)  $R = \frac{TP}{TP + FN}$

查准率 (Precision)  $P = \frac{TP}{TP + FP}$

# 性能度量——P-R曲线

## ➤ 查准率和查全率是一对矛盾的度量

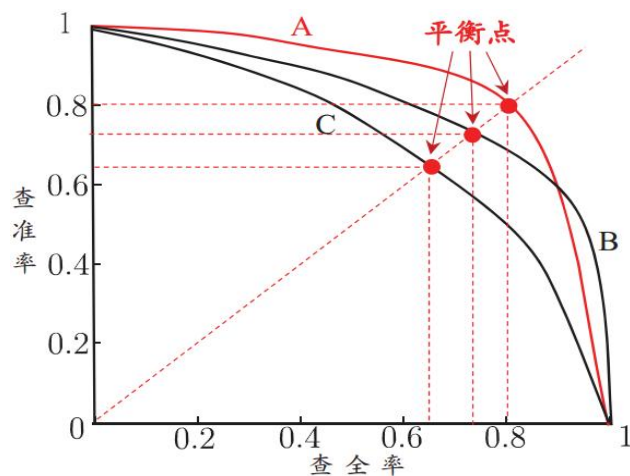
- 查准率高时，查全率低；查全率高时，查准率低

如何权衡这两个指标呢？

## ➤ 根据学习器的预测结果按正例可能性大小对样例进行排序，并逐个把样本作为正例进行预测，则可以得到查准率-查全率曲线，简称“P-R曲线”

如果一个学习器的P-R曲线被另一个学习器的曲线完全包住，那么后者性能更优

平衡点是曲线上“查准率=查全率”时的取值，可用来用于度量P-R曲线有交叉的分类器性能高低



P-R曲线与平衡点示意图

# 性能度量——F1度量

- 比P-R曲线平衡点更常用的是F1度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{1}{\frac{1}{2} \left( \frac{1}{P} + \frac{1}{R} \right)}$$

- 比F1更一般的形式  $F_{\beta}$  ,

$$F_{\beta} = \frac{(1 + \beta^2) \times P \times R}{\beta^2 P + R} = \frac{\frac{1}{\beta} + \beta}{\left( \frac{1}{\beta} \frac{1}{P} + \beta \frac{1}{R} \right)}$$

$\beta = 1$ ：标准的F1

$\beta > 1$ ：偏重查全率

$\beta < 1$ ：偏重查准率

# 性能度量——宏/微F1

## ➤如果有多个二分类混淆矩阵

多次训练/测试、多个数据集训练/测试、多分类中每两两类别组合

先在各个混淆矩阵上分别计算出查准率和查全率，记为  $(P_1, R_1), (P_2, R_2), \dots, (P_n, R_n)$ 。再计算均值，得到宏查准率 (macro-P)、宏查全率 (macro-R) 和相应的宏F1 (macro-F1)。

$$\begin{aligned}\text{macro-P} &= \frac{1}{n} \sum_i P_i \\ \text{macro-R} &= \frac{1}{n} \sum_i R_i \\ \text{macro-F1} &= \frac{2 \times \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}}\end{aligned}$$

先在各个混淆矩阵对应元素平均，得到TP、FP、TN、FN的平均值，再基于平均值计算微查准率 (micro-P)、微查全率 (micro-R) 和相应的微F1 (micro-F1)

$$\begin{aligned}\text{micro-P} &= \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \\ \text{micro-R} &= \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \\ \text{micro-F1} &= \frac{2 \times \text{micro-P} \times \text{micro-R}}{\text{micro-P} + \text{micro-R}}\end{aligned}$$



# 性能度量——ROC曲线与AUC

- 类似P-R曲线，根据学习器的预测结果对样例排序，并逐个作为正例进行预测，以“假正例率（FPR）”为横轴，“真正例率（TPR）”为纵轴可得到ROC曲线，全称“受试者工作特征（Receiver Operating Characteristics）”。

混淆矩阵 (confusion matrix)

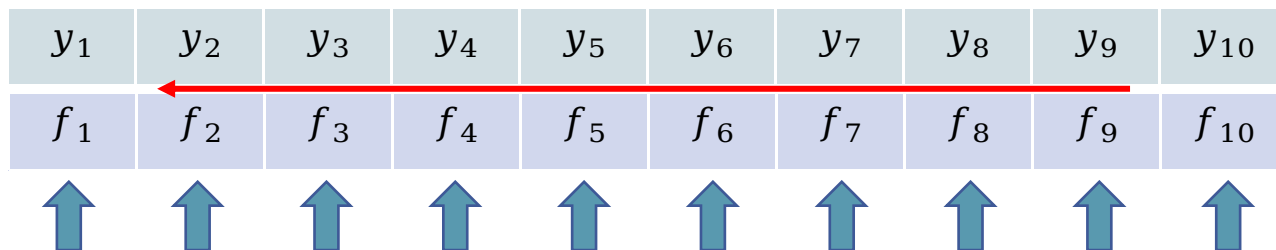
真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)



$$\begin{aligned} TPR &= \frac{TP}{TP + FN} \\ FPR &= \frac{FP}{FP + FN} \end{aligned}$$

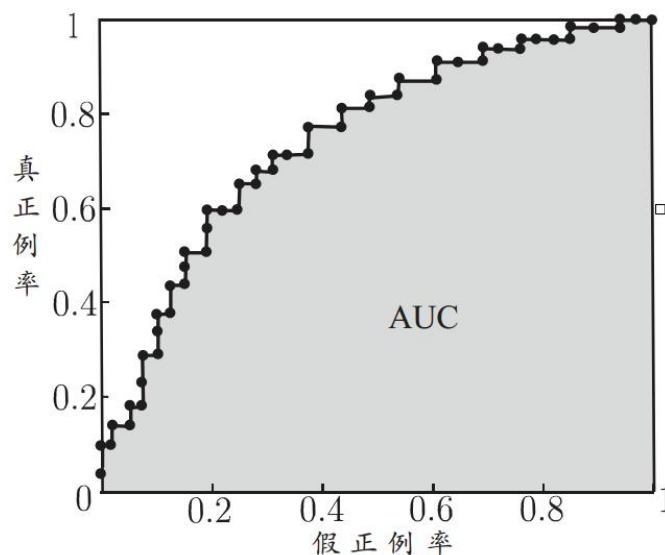
# 性能度量——ROC曲线与AUC

ROC图的绘制:



给定 $m^+$ 个正例和 $m^-$ 个负例，根据学习器预测结果对样例进行排序。

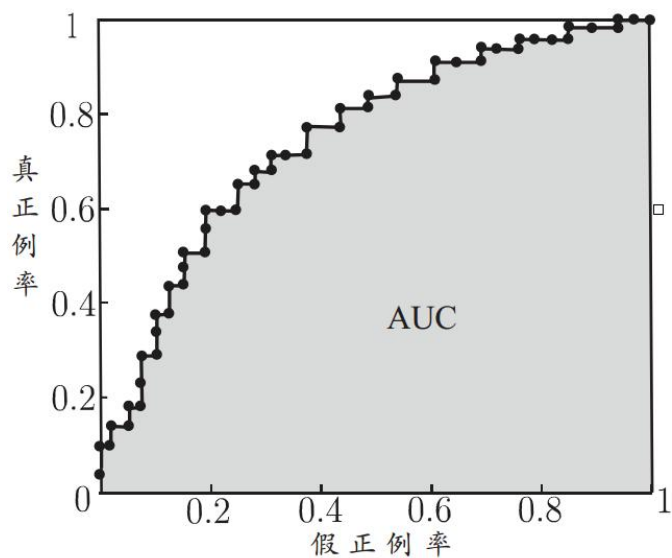
将分类阈值设为每个样例的预测值，当前标记点坐标为 $(x, y)$ ，当前若为真正例，则对应标记点的坐标为 $(x, y + \frac{1}{m^+})$ ；当前若为假正例，则对应标记点的坐标为 $(x + \frac{1}{m^-}, y)$ 。然后用线段连接相邻点



基于有限样例绘制的 ROC 曲线  
与 AUC

# 性能度量——ROC曲线与AUC

- 若某个学习器的ROC曲线被另一个学习器的曲线“包住”，则后者性能优于前者；否则如果曲线交叉，可以根据ROC曲线下面积大小进行比较，也即AUC值



基于有限样例绘制的 ROC 曲线  
与 AUC

假设ROC曲线由

$\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 的点按序连接而形成 ( $x_1 = 0, x_m = 1$ ), 则AUC可估算为

$$\text{AUC} = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \times (y_i + y_{i+1})$$

AUC衡量了样本预测的排序质量

# 比较检验——交叉验证t检验

现实任务中，更多时候需要对不同学习器性能进行比较

- 对两个学习器A和B，若k折交叉验证得到的测试错误率分别为 $\epsilon_1^A, \dots, \epsilon_k^A$ 和 $\epsilon_1^B, \dots, \epsilon_k^B$ ，可用k折交叉验证“成对t检验”进行检验
- 先对每个结果求差 $\Delta_i = \epsilon_i^A - \epsilon_i^B$
- 计算差值的均值 $\mu$ 和方差 $\sigma^2$

在显著度 $\alpha$ 下，若变量 $\tau_t = \left| \frac{\sqrt{k}\mu}{\sigma} \right|$ 小于临界值 $t_{\alpha/2, k-1}$ ，则假设不能被拒绝，即两个学习器没有显著差别。

否则认为两个学习器有显著差别，平均错误率小的那个学习器性能更优。

# 比较检验——McNemar检验

- 对于二分类问题，留出法不仅可以估计出学习器A和B的测试错误率，还能获得两学习器分类结果的差别，如下表所示

两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	$e_{00}$	$e_{01}$
错误	$e_{10}$	$e_{11}$

假设两学习器性能相同 $e_{01} = e_{10}$ ，则 $|e_{01} - e_{10}|$ 应服从正态分布，且均值为1，方差为 $e_{01} + e_{10}$ ，则

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}}$$

服从自由度为1的 $\chi^2$ 分布。

# 比较检验——Friedman检验

- 交叉验证t检验和McNemar检验都是在一个数据集上比较两个算法的性能
- Friedman检验在一组数据集上对多个算法进行比较

- 假设用 $D_1, D_2, D_3, D_4$ 四个数据集对算法 A, B, C进行比较
- 使用留出法或交叉验证法得到每个算法在每个数据集的测试结果
- 然后在每个数据集上根据性能好坏排序, 并赋序值1,2,...
- 若算法性能相同则平分序值,继而得到每个算法的平均序值

算法比较序值表

数据集	算法 A	算法 B	算法 C
$D_1$	1	2	3
$D_2$	1	2.5	2.5
$D_3$	1	2	3
$D_4$	1	2	3
平均序值	1	2.125	2.875

# 比较检验——Friedman检验

- 由平均序值进行Friedman检验来判断这些算法是否性能都相同

算法比较序值表

数据集	算法 A	算法 B	算法 C
$D_1$	1	2	3
$D_2$	1	2.5	2.5
$D_3$	1	2	3
$D_4$	1	2	3
平均序值	1	2.125	2.875

假设在N数据集上比较k个算法，令 $r_i$ 表示第i个算法的平均序值。若这些算法性能都相同，则它们的平均序值应当相同

若不考虑平分序值情况，那么

$$E[r_i] = \frac{k+1}{2}, \text{var}(r_i) = \frac{(k^2-1)}{12N}$$

当k和N都较大时，变量

$$\tau_{\chi^2} = \frac{12N}{k(k+1)} \left( \sum_{i=1}^k r_i^2 - \frac{k(k+1)^2}{4} \right)$$

服从自由度为k-1的卡方分布

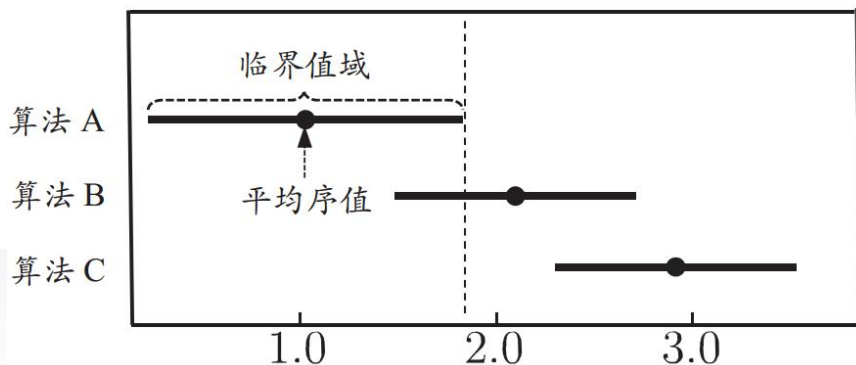
# 比较检验——Nemenyi后续检验

➤若“所有算法的性能相同”这个假设被拒绝，说明算法的性能显著不同，此时可用Nemenyi后续检验进一步区分算法。

➤Nemenyi检验计算平均序值差别的临界阈值  $CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$

➤如果两个算法的平均序值之差超出了临界阈值CD，则以相应的置信度拒绝“两个算法性能相同”这一假设。

➤根据上例的序值结果可绘制如下Friedman检验图，横轴为平均序值，每个算法圆点为其平均序值，线段为临界阈值的大小。



- 若两个算法有交叠(A和B)，则说明没有显著差别
- 否则有显著差别(A和C)，算法A明显优于算法C