第一讲 机器学习概述

2025年2月18日

概要

期望

随机变量X的期望$E[X]$定义为

$$E[X] = \sum_x xPr[X = x]$$

如果X服从概率分布$\mathcal{D}$，也可以将X的期望$E[X]$写成$E_{x\sim\mathcal{D}}[x]$。

期望的性质

- 对任意随机变量X和Y以及$a, b \in \mathbb{R}$,

$$E[aX + bY] = aE[X] + bE[Y].$$

- 如果X和Y是独立随机变量，则

$$E[XY] = E[X]E[Y].$$

马尔可夫不等式

Markov's inequality

设X为非负随机变量且$E[X] < \infty$, 则对任意$t > 0$,

$$Pr\left[X \geq tE[X]\right] \leq \frac{1}{t}.$$

证明:

$$\begin{aligned} Pr\left[X \geq tE[X]\right] &= \sum_{x \geq tE[X]} Pr[X = x] \\ &\leq \sum_{x \geq tE[X]} Pr[X = x] \frac{x}{tE[X]} \\ &\leq \sum_{x} Pr[X = x] \frac{x}{tE[X]} = E\left[\frac{X}{tE[X]}\right] = \frac{1}{t}. \ \square \end{aligned}$$

方差

随机变量X的方差$\mathrm{Var}[X]$定义为

$$\mathrm{Var}[X] = E[(X - E[X])^2].$$

随机变量X的标准差σ_X定义为

$$\sigma_X = \sqrt{\mathrm{Var}[X]}.$$

方差的性质

- 对任意随机变量X以及$a \in \mathbb{R}$,
  - $\mathrm{Var}[aX] = a^2\mathrm{Var}[X]$.
  - $\mathrm{Var}[X] = E[X^2] - E[X]^2$.
- 如果X和Y是独立随机变量, 则

$$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y].$$

切比雪夫不等式

Chebyshev's inequality

设X为随机变量且$\mathrm{Var}[X] < \infty$, 则对任意$t > 0$,

$$Pr\left[|X - E[X]| \geq t\sigma_X\right] \leq \frac{1}{t^2}.$$

证明: 注意到

$$Pr\left[|X - E[X]| \geq t\sigma_X\right] = Pr\left[(X - E[X])^2 \geq t^2\sigma_X^2\right]$$

且 $E\left[(X - E[X])^2\right] = \mathrm{Var}[X] = \sigma_X^2$, 应用马尔可夫不等式可得:

$$Pr\left[(X - E[X])^2 \geq t^2\sigma_X^2\right] \leq \frac{1}{t^2}. \quad \square$$

本课程需要的工具: 集中不等式 (不熟悉的同学请提前熟悉)

协方差

随机变量X和Y的协方差$\mathrm{Cov}(X, Y)$定义为

$$\mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])].$$

- 如果$\mathrm{Cov}(X, Y) = 0$，则称随机变量X和Y是不相关的。
- 协方差的性质
  - 对任意随机变量X, X', Y以及$a \in \mathbb{R}$,
    - $\mathrm{Cov}(X, X) = \mathrm{Var}[X] \geq 0$.
    - $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$.
    - $\mathrm{Cov}(X + X', Y) = \mathrm{Cov}(X, Y) + \mathrm{Cov}(X', Y)$, $\mathrm{Cov}(aX, Y) = a\mathrm{Cov}(X, Y)$.
  - 对满足$\mathrm{Var}[X] \leq +\infty$, $\mathrm{Var}[Y] \leq +\infty$的随机变量$X$和$Y$, 有

$$|\mathrm{Cov}(X, Y)| \leq \sqrt{\mathrm{Var}[X]\mathrm{Var}[Y]}.$$

协方差矩阵

随机向量$\mathsf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_\mathrm{N})$的协方差矩阵$\mathsf{C}(\mathsf{X}) \in \mathbb{R}^{N \times N}$定义为

$$\mathsf{C}(\mathsf{X}) = E\left[(\mathsf{X} - E[\mathsf{X}])(\mathsf{X} - E[\mathsf{X}])^T\right].$$

- $\mathsf{C}(\mathsf{X}) = \left(\mathrm{Cov}(\mathrm{X_i}, \mathrm{X_j})\right)_{ij}$.
- $\mathsf{C}(\mathsf{X}) = E\left[\mathsf{X}\mathsf{X}^T\right] - E[\mathsf{X}]E[\mathsf{X}]^\mathsf{T}$.

高斯分布与Laplace分布

高斯分布$N(\mu, \sigma^2)$

概率密度函数：$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Laplace分布$Laplace(\mu, b),\ b > 0$

概率密度函数：$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$

假设$f(x)$, $c_i(x)$, $h_j(x)$是定义在R^n上的连续可微函数, 考虑约束最优化问题:

$$\begin{aligned} & \min_{x \in R^n} f(x) \\ s.t. \quad & c_i(x) \leq 0, i = 1, 2, \ldots, k \\ & h_j(x) = 0, j = 1, 2, \ldots, l \end{aligned}$$

引入拉格朗日乘子$\alpha_i \geq 0, \beta_j$ $(1 \leq i \leq k,\ 1 \leq j \leq l)$, 构造广义拉格朗日函数

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k} \alpha_i c_i(x) + \sum_{j=1}^{l} \beta_j h_i(x).$$

进一步引入对偶函数$\theta_D(\alpha,\beta)=\min_x L(x,\alpha,\beta)$, 构造对偶问题

$$\begin{aligned}&\min_{\alpha,\beta}\theta_D(\alpha,\beta)\\ s.t.\quad &\alpha_i(x)\geq 0, i=1,2,\ldots,k\end{aligned}$$

- Karush-Kuhn-Tucker (TTK)条件:

$$\begin{aligned}&\nabla_x L(x^*,\alpha^*,\beta^*)=0\\ &\alpha_i^* c_i(x^*)=0, i=1,2,\ldots,k\\ &c_i(x^*)\leq 0, i=1,2,\ldots,k\\ &\alpha_i^*\geq 0, i=1,2,\ldots,k\\ &h_j(x^*)=0, j=1,2,\ldots,l\end{aligned}$$

- 梯度下降法
- 牛顿法与拟牛顿法
- 矩阵的基本子空间
- 特征值与特征向量
- 向量相对于标量的导数
- 矩阵对于标量的导数
- 标量对于矩阵的导数
- ……

概要

机器学习:为什么和是什么?

为什么需要机器学习?

- We are entering the era of big data.
- This deluge of data calls for **automated methods of data analysis**, which is what machine learning provides.

什么是机器学习?

We define *machine learning* as a set of methods that can automatically **detect patterns** in **data**, and then use the uncovered patterns to **predict** future data, or to **perform** other kinds of decision making under uncertainty.

— 《Machine Learning: A probabilistic perspective》 by Kevin Patrick Murphy, MIT Press, 2012

数据:学习的起点

- 我们通常将用于学习的数据对象或者实例称为**样例**或者**样本**.
- 每个样例x采用一个向量 $x = (x^{(1)}, x^{(2)}, \cdots, x^{(n)})^T$ 来表示.
- 向量的每个分量对应样例的一个**特征**或者**属性**.
  - n为样例x的特征个数，也称为**维数**.
  - $x^{(i)}$为样例x的第i 维属性的**属性值**.
- 属性张成的空间$\mathcal{X}$为**属性（特征）空间**，也称为**样本空间**或**输入空间**，记作$\mathcal{X}$.
  - $x = (x^{(1)}, x^{(2)}, \cdots, x^{(n)})^T \in \mathcal{X}$
- 一般而言，数据对象的特征和学习任务相关.

学习范例：带标记的数据

学习范例

- 不仅知道数据，而且也知道这些数据对应的学习**结果**或者**目标**.
  - 一位GPA=4.0、TOEFL=120的同学 申请到了 哈佛大学的奖学金
- 我们称数据所对应的学习结果为**标记**（label）.
  - 我们一般用y来表示实例x的标记.
  - (x, y):带标记的数据
- 我们用$\mathcal{Y}$来表示所有标记的集合，并称之为**输出空间**.

基本的数据集

- 无标记的数据集 $T = \{x_1, x_2, \ldots, x_N\}$ (简记为 $T = \{x_i\}_{i=1}^N$)
- 标记数据集 $T = \{(x_i, y_i)\}_{i=1}^N$

监督学习与无监督学习(1)

监督学习 (Predictive or Supervised learning)

- 基于给定的标记数据集$T = \{(x_i, y_i)\}_{i=1}^{N}$ (**训练数据集**或训练样本).
- 学习从输入空间$\mathcal{X}$ 到输出空间$\mathcal{Y}$ 的映射（**模型**）.
- 并利用该映射对**未见**（unseen）实例x对应的输出y进行**预测**.
- 标记的角色:
  - 用于模型**学习**.
  - 通过对比模型对x_i的预测和y_i之间的差异能对学习性能进行一定程度的**评估**.

监督学习与无监督学习(2)

监督学习的两个核心问题

- 分类(classification)问题: 输出空间$\mathcal{Y}$是一个离散值的集合(通常也是有限的).
  - $\mathcal{Y} = \{c_1, c_2, \cdots, c_M\}$, 其中$M$为类别的个数.
  - 二分类(binary classification)问题: $M = 2$.
    - $\mathcal{Y} = \{+1, -1\}$.
    - $\mathcal{Y} = \{0, 1\}$.
  - 多分类(multi-class classification) 问题: $M > 2$.
- 回归（regression）问题: 输出空间$\mathcal{Y} = \mathrm{R}$.

监督学习与无监督学习(3)

无监督学习(Descriptive or unsupervised learning)

- 基于给定的无标记的数据集$T = \{x_i\}_{i=1}^N$.
- 发现数据中隐含的知识或者模式(interesting patterns).
- 并将学得的模式应用于未见实例.
- 无监督学习通常也被称为知识发现(knowledge discovery).
- 通常没有明确的知识模式类型、衡量学习结果等的度量
- 依赖于具体学习场景和应用领域.
- 更具有主观性和挑战性.

监督学习与无监督学习(4)

聚类: 典型的无监督学习任务

- 将$T = \{x_i\}_{i=1}^N$划分成若干子集
  - 这些子集通常互不相交
  - 属于同一子集的样本数据尽可能相互相似
  - 不同子集的样本尽可能不同
- 称每个子集为簇
- 每个簇对应于一个潜在的概念

监督学习与无监督学习(5)

半监督学习 (Semi-supervised learning)

- 既有标记过的数据.
- 也有未标记过的数据(通常所占比例比较大).
- 希望未标记数据的分布能帮助学习器获得比监督情形更好的性能.

强化学习(Reinforcement learning)

- 学习使得系列动作的长期累加回报最大化的策略
  - 搜集学习器对环境主动施加动作以后环境状态的变化以及所获得的即时回报或者惩罚.
  - 同时学习者需要在探索未知动作的回报和利用已经收集到信息之间进行权衡.

概要

目标概念与假设

监督学习：基于$T = \{(x_i, y_i)\}_{i=1}^{N}$)来学习从输入空间$\mathcal{X}$ 到输出空间$\mathcal{Y}$ 的映射，假定

- 输入空间$\mathcal{X}$中的所有样本相互独立且服从同一**固定但未知**的分布$\mathcal{D}$.
- 且T中的每个样例x_i都是依分布$\mathcal{D}$独立同分布产生的，其标记$y_i = c(x_i)$，其中
  - $c \in \mathcal{C}$是目标概念，它是从输入空间$\mathcal{X}$到输出空间$\mathcal{Y}$的映射，决定实例x的真实标记y.
  - $\mathcal{C}$是概念类，即希望被学习的一个概念集.

学习阶段

- 它所考虑的所有可能的概念的集合称为假设空间$\mathcal{H}$ (未必与$\mathcal{C}$相同).
- 给定学习算法$\mathcal{L}$基于T依一定策略选择出一个假设$h_T \in \mathcal{H}$.

泛化误差

- 如何评估学习算法$\mathcal{L}$学得的模型h_T? 考察h_T的泛化能力,即对未见数据的预测能力!
- 给定假设$h \in \mathcal{H}$,
  - 损失函数$L(h(x), y)$度量h一次预测的"好坏"
    - 0-1损失函数

$$L_{0-1}(h(x), y) = I(h(x) \neq y) = \begin{cases} 1, & h(x) \neq y \\ 0, & h(x) = y \end{cases}.$$

    - 平方损失函数

$$L_2(h(x), y) = (h(x) - y)^2.$$

  - 平均损失(期望)度量平均意义下h预测的"好坏":

$$R(h) = E_{x \sim \mathcal{D}}[L(h(x), c(x))]$$

  - 称平均损失(风险函数)$R(h)$为泛化误差.

泛化误差

- 给定假设$h \in \mathcal{H}$, 采用0-1损失函数$L_{0-1}(h(x), y)$, 则泛化误差为

$$\begin{aligned} R_{0-1}(h) &= E_{x\sim\mathcal{D}}[I(h(x) \neq c(x))] \\ &= Pr_{x\sim\mathcal{D}}(h(x) \neq c(x)) \end{aligned}$$

  即h在整个输入空间中预测错误的概率.
- 注意事先既不清楚c的具体存在，也不能知道分布$\mathcal{D}$!
- h关于训练数据集T的平均损失（也称为经验风险）如下：

$$\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} L(h(x_i), y_i).$$

训练误差

针对分类问题，对算法$\mathcal{L}$基于T学得的模型h_T来说

- 其经验误差为

$$\hat{R}(h_T) = \frac{1}{N}\sum_{i=1}^{N} L(h_T(x_i), y_i).$$

- 如果采用0-1损失函数，则经验误差$\hat{R}(h_T)$就是h_T在T上的预测误差率(error rate):

$$\hat{e}_r(h_T) = \frac{1}{N}\sum_{i=1}^{N} I(h_T(x_i) \neq y_i).$$

相应的，h_T关于T的预测精度(accuracy)定义为

$$\hat{a}(h_T) = 1 - \hat{e}_r(h_T).$$

经验误差与泛化误差

经验误差$\hat{R}(h_T)$小并不能一定保证模型h_T的泛化性能好!

- 泛化性能与学习算法捕获所有样本的**共有知识模式**的能力有关
- 经验误差反映的是学习算法捕获训练数据蕴含的知识模式的能力
- 过小的训练误差可能导致所谓的过拟合（Overfitting）现象

测试误差

- 测试集（testing set）$T' = \{(x_i', y_i')\}_{i=1}^{N'}$：没有参与训练模型的独立数据集，这里$N'$是测试样本容量.
- 模型h_T关于测试数据集T'的平均损失为

$$\hat{R}_{test}(h_T) = \frac{1}{N'} \sum_{i=1}^{N'} L(h_T(x_i'), y_i').$$

- h_T关于测试数据集T'的误差率为

$$\hat{e}_{test}(h_T) = \frac{1}{N'} \sum_{i=1}^{N'} I(h_T(x_i') \neq y_i').$$

- h_T关于测试数据集T'的的预测精度为

$$\hat{a}_{test}(h_T) = 1 - \hat{e}_{test}(h_T).$$

- $E_{T' \sim \mathcal{D}^{N'}}[\hat{R}_{test}(h_T)] = R(h_T).$

准确率和召回率

对二分类任务来说，

- 比较关注的类为正类(P).
- 另一个类为负类(N).

h_T 对 T' 的样本预测结果有四类：

- 真正例(true positive)，即预测为正类的样例实际是 T' 中的正类样例，真正例的总数为 TP；
- 假正例(false positive)，即预测为正类的样例实际是 T' 中的负类样例，假正例的总数为 FP；
- 真负例(true negative)，即预测为负类的样例实际是 T' 中的负类样例，真负例的总数为 TN；
- 假负例(false negative)，即预测为负类的样例实际是 T' 中的正类样例，假负例的总数为 FN.

准确率和召回率

真实	预测	
	P	**N**
P	TP	FN
N	FP	TN

Figure: 混淆矩阵

- $TP + FN$为测试集中正类样例的个数.
- $TN + FP$为测试集中负类样例的个数.
- $TP + FN + TN + FP = N'$.
- $TP + FP$为测试集中被h_T 预测为正类的样例个数.

准确率和召回率

- 准确率(查准率)P: 被h_T 预测为正类的样例中真正例所占的比例，即

$$P = \frac{TP}{TP + FP}.$$

- 召回率(查全率)R: 测试集中正类样例中被h_T 预测为正类的样例所占的比例，即

$$R = \frac{TP}{TP + FN}.$$

- 准确率和召回率是相互抵触.
  - 调和均值F_1 度量:

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R},$$

即 $F_1 = \frac{2PR}{P+R}$.

过拟合与正则化

学习算法$\mathcal{L}$按照什么样的策略来选择模型h_T?

- 经验风险最小化策略:

$$h_T = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \hat{R}(h)$$

过拟合风险比较高.

- 正则化策略:

$$h_T = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \left[\hat{R}(h) + \lambda J(h) \right],$$

这里

- $J(h)$是模型h的复杂度的单调递增函数,
- $\lambda \geq 0$是权衡经验误差$\hat{R}(h)$和复杂度函数$J(h)$的系数.

正则化策略

- 正则化项$J(h)$也称惩罚项，用以刻画模型的复杂度所带来的过拟合“风险”.
- 如果$\lambda = 0$，对应于经验风险最小化策略.
- 如果λ相当大
  - 过于强调模型的复杂度所带来的过拟合“风险”.
  - 导致所选择的模型过于简单.
  - 学习能力比较低，从而导致所谓**欠拟合**(underfitting)现象.

如何选择合适的λ?

- 先给出λ的若干个候选值，然后对每个λ值训练一个模型.
- 在测试集上进行测试，选择测试评估指标最佳的模型所对应的λ值作为模型的λ 值.
- 做模型选择的测试集通常被称为**验证集**（Validation set）.

基于数据集划分的模型选择

如何从数据集D中划分出训练集T和测试（验证）集T'进行模型选择?
随机采样!

- 留出法(hold-out)
- k折交叉验证法（k-fold cross validation）
  - 留一（Leave-one-Out）验证法
- 自助法（Bootstrapping）

留出法(hold-out)（也称简单交叉验证法）

将数据集D随机划分为两个互不相交的子集，其中一个作为训练集T，另一个作为测试集T':

- 采用无放回的随机采样方式从数据集D中抽出一部分数据（设定的比例或个数）作为T，剩下的数据作为T'.
- 要在采样中尽可能保持数据分布的一致性，可采用分层无放回随机采样方式.
- 通常重复若干次随机划分过程，以每次划分对应的测试评估的均值作为留出法的评估结果.

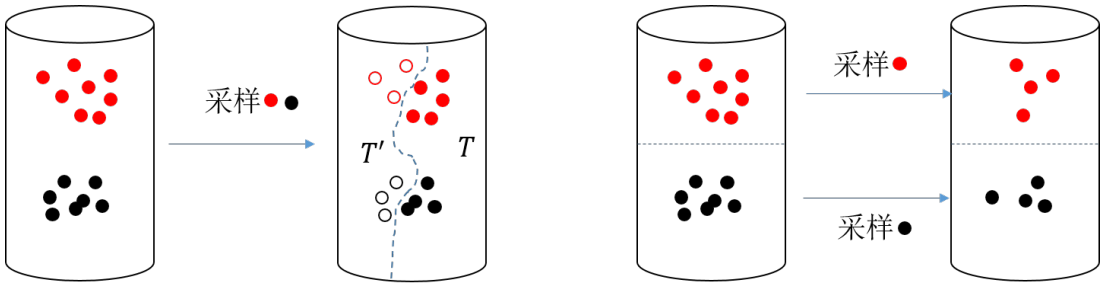

Figure: 留出法

k折交叉验证法

- 将数据集D随机划分为k个互不相交、大小相似的子集$D_1, D_2, \cdots, D_k$.
- 进行k次训练-测试过程，其中第i次训练-学习过程中
  - 以$D - D_i$为训练数据集学得模型h_{D-D_i}，
  - 以D_i为测试集对h_{D-D_i}进行测试评估，得到测试误差$\hat{R}_{test}(h_{D-D_i})$.
- 以

$$\frac{1}{k}\sum_{i=1}^{k}\hat{R}_{test}(h_{D-D_i})$$

作为h_D在本次数据集随机划分下的测试评估结果。

k折交叉验证法

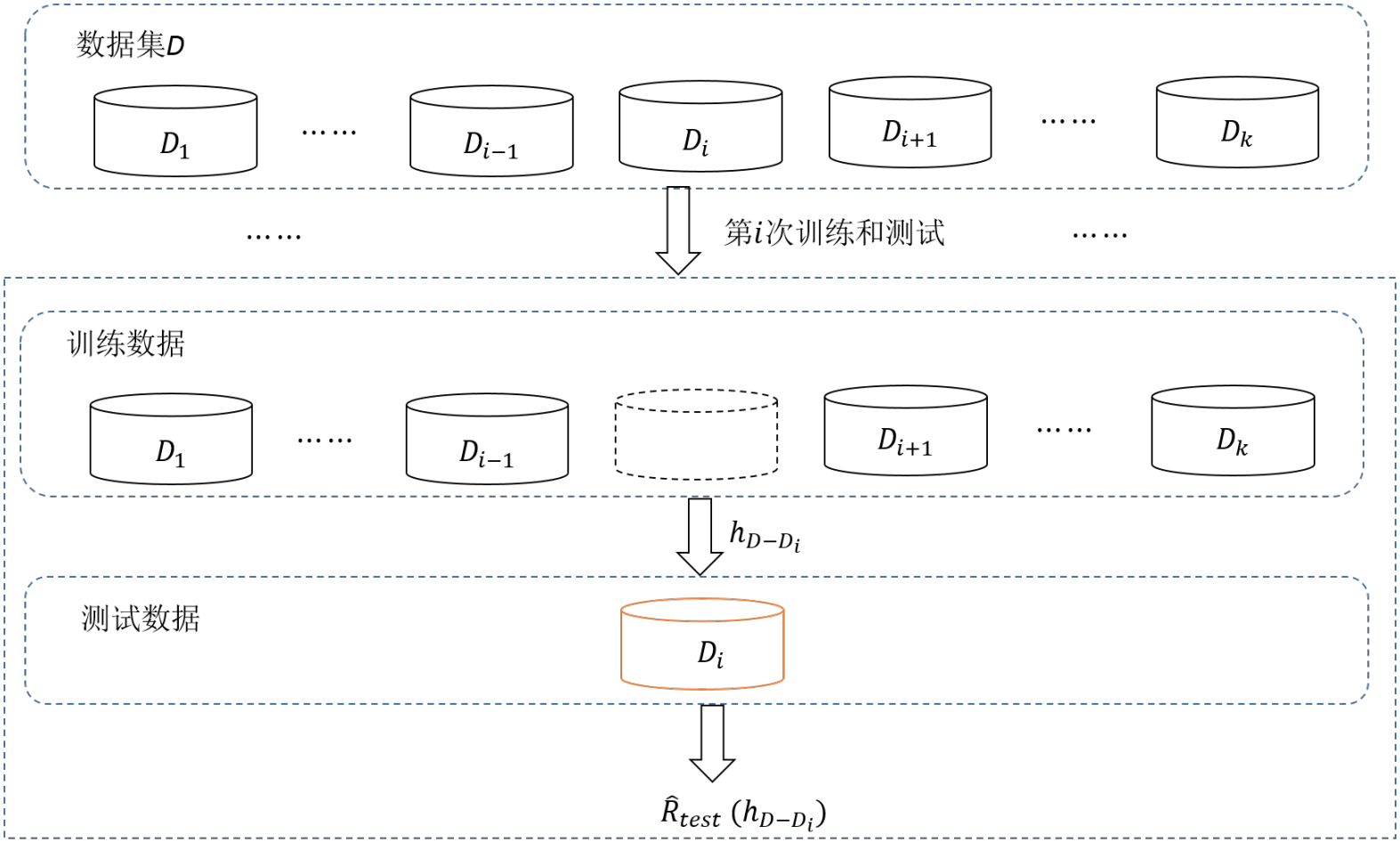

Figure: k-折交叉验证法

k折交叉验证法

- 增强测试评估的稳定性和可靠性
  - 通常重复进行若干次数据集的随机划分过程.
  - 以每次划分对应的测试评估的均值作为最终的评估结果.
- k值的选择对评估结果有一定影响
  - 常用10折交叉验证法.
  - $k = |D|$: 留一（Leave-one-Out）验证法

留一（Leave-one-Out）验证法

- 每次基于$|D|-1$个数据进行训练
- 只用本次未参与训练的数据作为测试数据.
- 对每个样本x来说，恰好参与了$|D|-1$次训练，只参与了1次测试。
- 留一误差：

$$\hat{R}_{loo}(h_D) = \frac{1}{|D|}\sum_{x\in D} L(h_{D-\{x\}}(x), y).$$

- 可以认为模型$h_{D-\{x\}}$和h_D很接近，因此留一法进行测试评估通常也比较准确可信.
- 训练次数等于样本容量，当样本容量比较大的时候计算开销比较大.

自助法（Bootstrapping）

- 经过模型选择以后，基于整个数据集D重新训练出最终模型h_D.
- 基于数据集随机划分的模型选择中，采用无放回抽样的方式：
  - 每次使用的训练数据集都是D的一个真子集，其样本容量最大为$|D|-1$.
- D和模型选择阶段的训练样本集的规模方面的差异会对最终模型的评估造成一些偏差.
- 从留出法到自助法： 采用**有放回的抽样**方法对留出法进行改造

自助法（Bootstrapping）

- 先从D中以有放回的抽样方式随机抽取$|D|$个数据来构建训练数据集T,
- 然后以D中没有被抽中的数据构建测试数据集T'.

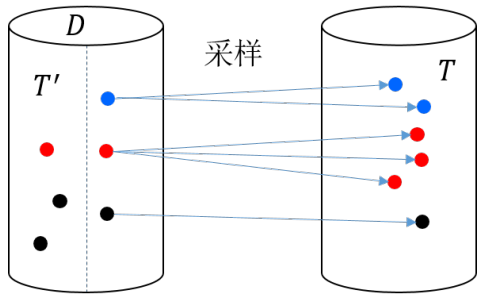

Figure: 自助法

- 自助法解决了交叉验证法中模型选择阶段和最终模型训练阶段的训练集规模差异问题.
- 但训练集T和原始数据集D中数据的分布未必相一致， 因此对一些对数据分布敏感的模型选择并不适用.

概要

模型复杂度与泛化误差

随着模型复杂度的增加

- 学习算法的学习能力越来越强
- 训练误差越来越小

泛化误差

- 先是随着训练误差的缩小而减小，
- 但随着模型复杂度的进一步增加泛化误差不降反升

泛化误差与模型复杂度的关系为什么这样?

针对回归任务，进一步讨论泛化误差由哪几个部分构成.
我们设h_T是基于训练数据集T学习到的回归模型，对给定的x,
则学习算法的泛化误差为

$$E_T[(h_T(x) - c(x))^2].$$

偏差

- 定义学习算法对数据x的期望输出为

$$\bar{h}(x) = E_T[h_T(x)].$$

- x的期望输出与真实标记$c(x)$之间的差别称为**偏差**，即

$$Bias(x) = E_T[(h_T(x) - c(x)] = \bar{h}(x) - c(x).$$

- 偏差描述了学习算法对x的预测期望相对于x的真实输出的偏离程度.
- 偏差反映了学习算法的学习能力.
- 偏差越小，说明学习算法的学习能力越强.

方差

- 基于相同样本容量的不同训练数据集产生的**预测方差**为

$$Var(x) = E_T[(h_T(x) - \bar{h}(x))^2].$$

- 方差刻画学习算法使用相同容量的不同训练数据集所导致的学习性能的变动情况.
- 方差越小，说明学习算法对数据扰动的容忍能力越强.

偏差-方差分解

进一步，我们对泛化误差进行如下分解：

$$\begin{aligned} & E_T[(h_T(x) - c(x))^2] \\ = \; & E_T[h_T^2(x) - 2h_T(x)c(x) + c^2(x)] \\ = \; & E_T[h_T^2(x)] - 2E_T[h_T(x)]c(x) + c^2(x) \end{aligned}$$

偏差-方差分解

进一步，我们对泛化误差进行如下分解：

$$
\begin{aligned}
& E_T[(h_T(x) - c(x))^2] \\
= \; & E_T[h_T^2(x) - 2h_T(x)c(x) + c^2(x)] \\
= \; & E_T[h_T^2(x)] - 2E_T[h_T(x)]c(x) + c^2(x) \\
= \; & E_T[h_T^2(x)] - 2\bar{h}(x)c(x) + c^2(x)
\end{aligned}
$$

偏差-方差分解

进一步，我们对泛化误差进行如下分解：

$$
\begin{aligned}
& E_T[(h_T(x) - c(x))^2] \\
= \; & E_T[h_T^2(x) - 2h_T(x)c(x) + c^2(x)] \\
= \; & E_T[h_T^2(x)] - 2E_T[h_T(x)]c(x) + c^2(x) \\
= \; & E_T[h_T^2(x)] - 2\bar{h}(x)c(x) + c^2(x) \\
= \; & E_T[h_T^2(x)] - \bar{h}^2(x) + \bar{h}^2(x) - 2\bar{h}(x)c(x) + c^2(x)
\end{aligned}
$$

偏差-方差分解

进一步，我们对泛化误差进行如下分解：

$$
\begin{aligned}
& E_T[(h_T(x) - c(x))^2] \\
= \; & E_T[h_T^2(x) - 2h_T(x)c(x) + c^2(x)] \\
= \; & E_T[h_T^2(x)] - 2E_T[h_T(x)]c(x) + c^2(x) \\
= \; & E_T[h_T^2(x)] - 2\bar{h}(x)c(x) + c^2(x) \\
= \; & E_T[h_T^2(x)] - \bar{h}^2(x) + \bar{h}^2(x) - 2\bar{h}(x)c(x) + c^2(x) \\
= \; & E_T[(h_T(x) - \bar{h}(x))^2] + (\bar{h}(x) - c(x))^2 \\
= \; & Var(x) + Bias^2(x).
\end{aligned}
$$

这说明泛化误差可分解为**方差**和**偏差**的平方之和.

偏差-方差分解

- 由于噪声等的存在，使得x对应的观测y未必一定有$y = c(x)$.
- 我们不妨设

$$y = c(x) + \varepsilon,$$

其中ε为噪声，假定ε服从分布$\mathcal{E}$且其期望为0，即$E[\varepsilon] = 0$.

则

$$E_{T\sim\mathcal{D}^{|T|},\varepsilon\sim\mathcal{E}}[(h_T(x) - y)^2] = Var(x) + Bias^2(x) + E[\varepsilon^2],$$

即泛化误差可以分解为**方差**、**偏差**和**噪声**三部分，其中

- 噪声部分也称为不可约误差，反映了学习问题本身的难度.

偏差-方差分解

偏差-方差分解:

$$E_{T\sim\mathcal{D}^{|T|},\varepsilon\sim\mathcal{E}}[(h_T(x)-y)^2] = Var(x) + Bias^2(x) + E[\varepsilon^2]$$

- 方差和偏差通常是相互抵触的.
- 当模型复杂度过于简单时
  - 拟合能力比较弱, 对数据扰动不敏感
  - 此时偏差在泛化误差中起主导作用.
- 随着模型复杂度的提高
  - 算法的拟合能力不断增强，偏差逐渐减少.
  - 但学习能力的提高也带来过拟合的风险，使得学习算法对数据扰动逐渐敏感.
  - 方差在泛化误差中的比重逐渐增大，最终导致泛化误差不断增大.

偏差-方差困境

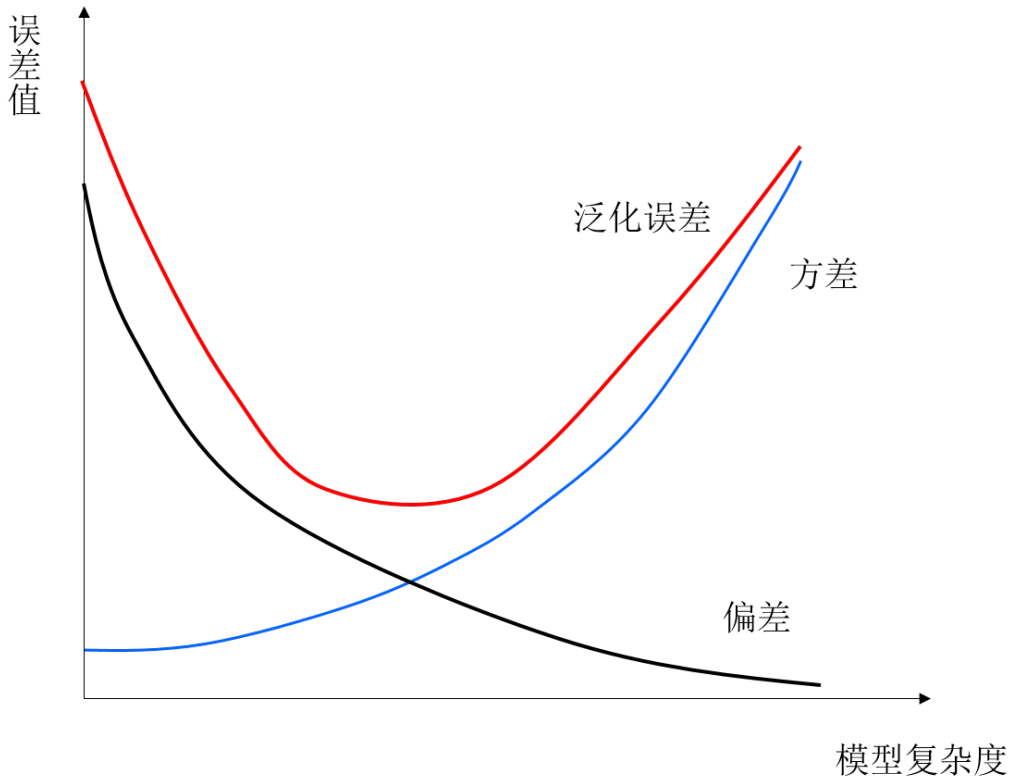

Figure: 偏差、方差与泛化误差

总结

- 基本概念
  - 数据、特征、样例、标记、特征空间、输入空间、输出空间、监督学习、非监督学习、分类、回归、聚类
- 模型评估与选择
  - 泛化误差、训练误差、测试误差、正则化、过拟合、留出法、k-折交叉验证法、留一法、自助法
- 偏差-方差分解
  - 偏差、方差、偏差-方差分解、偏差-方差困境