Lecture3 Machine Learning

1. 学习算法

机器学习算法是一种能够**从数据中学习的算法**，什么是学习，Mitchell (1997) 提供了一个简洁的定义

> 对于某类任务 T 和性能度量 P，一个计算机程序被认为可以从经验 E 中学习是指，通过经验 E 改进后，它在任务 T 上由性能度量 P 衡量的性能有所提升

- 这里经验 E，任务 T 和性能度量 P 的定义范围非常宽广

任务 T

机器学习**任务**定义为**机器学习系统应该如何处理样本**

> **样本（example）**：从某些希望机器学习系统处理的对象或事件中收集到的已经量化的**特征（feature）**的集合
>
> - 通常样本会被表示成一个向量 $\boldsymbol{x} \in \mathbb{R}^n$，其中每一个元素 x_i 是一个特征
>   - 例如：一张图片的特征通常指这张图片的像素值

任务类型	描述	应用
分类	计算机程序需要指定某些**输入属于 k 类中的哪一类** 学习算法通常会返回一个函数：$f:\mathbb{R}^n \to \{1,\dots,k\}$ 当 $y=f(\boldsymbol{x})$ 时，模型将向量 $\boldsymbol{x}$ 所代表的输入分类到数字码 y 所代表的类别	**对象识别**：（输入-图片，通常由一组像素亮度值表示，输出-表示图片物体的数字码） **人脸识别**
输入缺失分类	当输入向量的每个度量不被保证的时候，即当一些输入可能丢失时，学习算法必须学习**一组函数**，不是单个分类函数 每个函数对应着分类具有不同缺失输入子集的 $\boldsymbol{x}$	**医疗诊断**：学习所有相关变量的概率分布，然后通过边缘化缺失变量来解决分类任务
回归	计算机程序需要对给定输入**预测数值** 学习算法需要输出函数 $f:\mathbb{R}^n \to \mathbb{R}$	预测投保人的索赔金额 预测证券未来的价格
转录	观测一些相对**非结构化**表示的数据，并转录信息为**离散**的文本形式	文本图片转文字序列 街道图片提取接到信息 语音转文字
机器翻译	输入是一种语言的符号序列，计算机程序必须将其**转化**成另一种语言的符号序列	中文翻译英文
异常检测	计算机程序在一组事件或对象中筛选，并标记**不正常或非典型的个体**	信用卡公司检测信用卡异常使用
合成和采样	机器学习程序生成一些**和训练数据相似的新样本**	为艺术家生成图片 采样和合成的过程根据特定的输入生成特定类型的输出（**结构化输出任务**） 每个输入并非只有一个正确输出的条件，我们明确希望输出有很多变化
缺失值填补	机器学习算法给定一个新样本 $\boldsymbol{x}\in\mathbb{R}$，$\boldsymbol{x}$ 中的某些元素 x_i 缺失，**算法必须填补这些数值**	

任务类型	描述	应用
去噪	干净样本 $\boldsymbol{x} \in \mathbb{R}^n$ 经过未知损坏过程后得到的损坏样本 $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ 算法根据损坏后的样本 $\tilde{\boldsymbol{x}}$ **预测干净的样本** $\boldsymbol{x}$，或者更一般的预测条件概率分布 $p(\boldsymbol{x}\|\tilde{\boldsymbol{x}})$	

性能度量 P

性能度量 P 是特定于系统执行的任务 T 而言的

性能度量	任务
准确率 （accuracy）/ 错误率（error rate）	分类、缺失输入分类、转录
模型在样本上概率对数的平均值	密度估计、回归任务

- 一般使用**测试集（test set）**来评估系统性能，将其与训练机器学习系统的训练集数据分开
- 还有一些情况，我们知道应该度量哪些数值，但是度量它们不太现实
    - 必须设计一个仍然对应于设计对象的**替代标准**，或者设计一个**理想标准的良好近似**

性能度量常见指标	公式	任务
均方误差（mean squared error）	$\mathrm{MSE}_{\text{test}} = \frac{1}{m}\sum_i \left(\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right)_i^2$	回归

经验 E

大部分学习算法可以被理解为在整个**数据集**上获取经验

> **数据集（dataset）**：指很多样本组成的集合
>
> - 也将样本称为**数据点（data point）**

无监督 Unsupervised

训练含有很多特征的数据集，然后学习出这个数据集上有用的结构性质

无监督学习可以用于密度估计、合成、去噪、聚类等

观察随机向量 $\mathbf{x}$ 的好几个样本，试图显式或者隐式地学习出概率分布 $p(\mathbf{x})$，或者该分布的一些有意思的性质

监督 Supervised

训练含有很多特征的数据集，不过数据集中的样本都有一个**标签（label）**或 **目标（target）**

观察随机向量 $\mathbf{x}$ 及其相关联的值或向量 $\mathbf{y}$，然后从 $\mathbf{x}$ 预测 $\mathbf{y}$，通常是估计 $p(\mathbf{y}|\mathbf{x})$

半监督 Semi-Supervised

一些样本有监督目标，一些没有

强化学习 Reinforcement Learning

不是训练于一个固定的数据集，算法会于环境交互

数据集的表示

表示数据集的常用方法是**设计矩阵（design matrix）**，每一个样本要能表示成向量，并且这些向量的维度相同（样本维度不相同的情况在这里暂不做讨论）

- 每一行：包含一个不同的样本
- 每一列：对应不同的特征

在监督学习中，样本包含标签或者目标和一组特征

示例：线性回归

任务 T

建立一个系统，将向量 $\boldsymbol{x} \in \mathbb{R}^n$ 作为输入，预测标量 $y \in \mathbb{R}$ 作为输出

线性回归的输出是其输入的**线性函数**，令 $\hat{y}$ 表示模型预测 y 应该取的值，定义输出为

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x}$$

- $\boldsymbol{w} \in \mathbb{R}^n$，是 **参数（parameter）**向量
  - 可以将 $\boldsymbol{w}$ 看作是一组决定每个特征如何影响预测的**权重（weight）**
    - 如果特征权重的大小很大，那么它对预测有很大的影响
    - 如果特征权重的大小是零，那么它对预测没有影响

> **参数（parameter）**是控制系统行为的值

我们拥有有每个样本对应的正确值 y 组成的回归目标向量，将输入的设计矩阵记作 $\boldsymbol{X}^{(test)}$，回归目标向量记作 $\boldsymbol{y}^{(test)}$

性能度量 P

使用均方误差来度量模型性能

$$\mathrm{MSE}_{\text{test}} = \frac{1}{m}\sum_i \left(\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})}\right)_i^2$$

经验 E

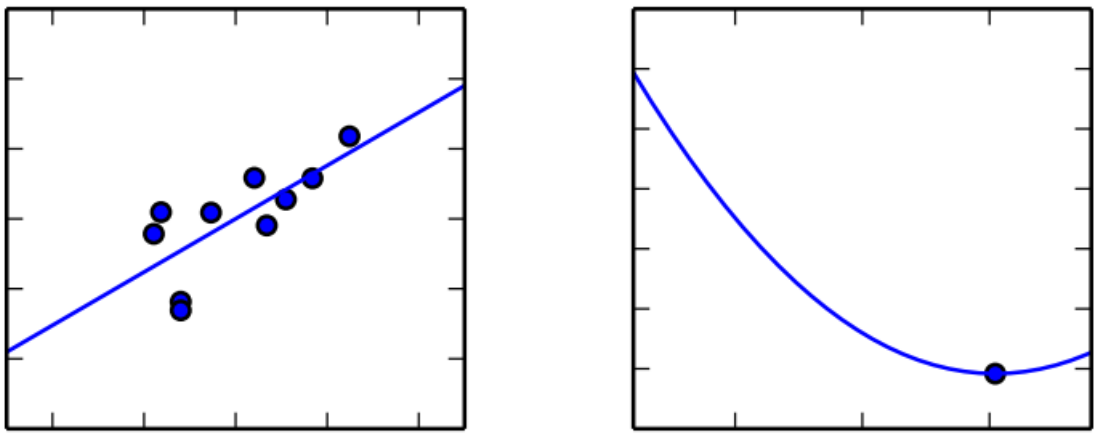
需要设计一个算法，通过观察训练集 $(\boldsymbol{X}^{(train)}, \boldsymbol{y}^{(train)})$ 获得经验，减少 $\mathrm{MSE}_{\text{test}}$ 来改进权重 $\boldsymbol{w}$

最小化均方误差的方式是，求 $\boldsymbol{w}$ 导数为 $\mathbf{0}$ 的情况

$$\min \mathrm{MSE}_{\text{test}} \rightarrow \nabla_w \mathrm{MSE}_{\text{train}} = 0$$

$$\begin{aligned}
&\Rightarrow \nabla_w \frac{1}{m}\left\|\hat{\boldsymbol{y}}^{(\text{train})} - \boldsymbol{y}^{(\text{train})}\right\|_2^2 = 0 \\
&\Rightarrow \frac{1}{m}\nabla_w \left\|\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right\|_2^2 = 0 \\
&\Rightarrow \nabla_w \left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right)^\top \left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right) = 0 \\
&\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - 2\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} + \boldsymbol{y}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}\right) = 0 \\
&\Rightarrow 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} = 0 \\
&\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\right)^{-1}\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}
\end{aligned}$$

分析



- 一个线性回归问题，其中训练集包括十个数据点，每个数据点包含一个特征，因为只有一个特征，权重向量 $\boldsymbol{w}$ 也只有一个要学习的参数 w_1
  - 左图：可以观察到线性回归学习 w_1，从而使得直线 $y = w_1 x$ 能够尽量接近穿过所有的训练点
  - 右图：通过经验 E 学习到的 w_1 的值

> **线性回归（linear regression）**通常用来指稍微复杂一些，附加额外参数（截距项 b）的模型
> 在这个模型中
>
> $$\hat{y} = \boldsymbol{w}^T \boldsymbol{x} + b$$
>
> **偏置（bias）**：截距项 b 通常被称为仿射变换的偏置（bias）参数

2. 机器学习概念

误差 error/loss

机器学习的主要挑战是我们的算法必须能够在**先前未观测的新输入**上表现良好

> **泛化（generalization）**：在先前未观测到的输入上表现良好的能力
>
> - 泛化能力：训练误差和测试误差之间的差距
> - 注意，只有当训练误差很小时，它才有意义
> - 在训练误差很大的情况下，没有必要再谈论泛化能力了

- **训练误差（training error）**：在训练集上计算的误差
- **测试误差（test error）/泛化误差（generalization error）**：在测试集（新输入）上的误差

在线性回归实例中，我们通过最小化误差训练模型，训练的性能度量为

$$\mathrm{MSE}_{\text{train}} = \frac{1}{m} \sum_i \left(\hat{\boldsymbol{y}}^{(\text{train})} - \boldsymbol{y}^{(\text{train})} \right)_i^2$$

但实际我们关注的是测试误差

$$\mathrm{MSE}_{\text{test}} = \frac{1}{m} \sum_i \left(\hat{\boldsymbol{y}}^{(\text{test})} - \boldsymbol{y}^{(\text{test})} \right)_i^2$$

数据生成 data generating process

训练集和测试集数据通过数据集上被称为**数据生成过程（data generating process）**的概率分布生成

通常，它们是满足**独立同分布假设**的，即每个数据集中的样本都是彼此**相互独立的（independent）**，并且训练集和测试集是**同分布的（identically distributed）**，训练集和测试集的样本共享这个潜在的分布——**数据生成分布（data generating distribution）**，记为 p_{data}

训练误差和测试误差之间的直接联系，**随机模型训练误差的期望和该模型测试误差的期望是一样的**

容量、过拟合和欠拟合

容量（capacity）：是指其拟合各种函数的能力

- 容量低的模型可能很难拟合训练集
- 容量高的模型可能会过拟合

表示容量（representational capacity）：学习算法可以从哪些函数族中选择函数

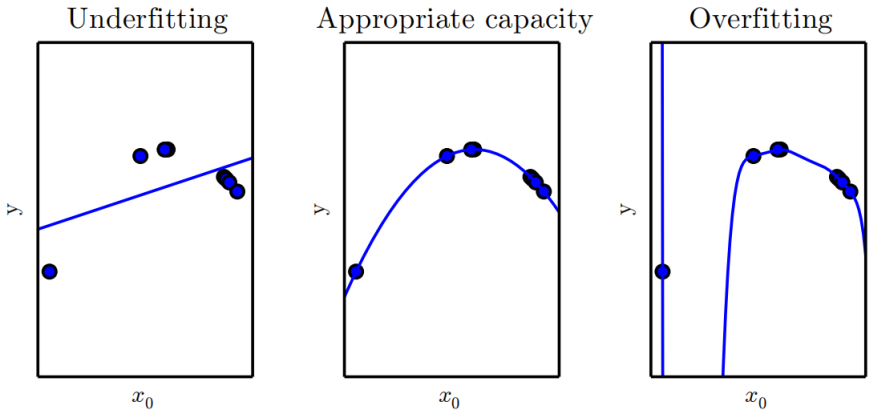
有效容量（effective capacity）：模型在训练的时候实际可选的容量，可能小于模型族的表示容量

控制训练算法容量

选择**假设空间（hypothesis space）**，即学习算法可以选择为解决方案的函数集

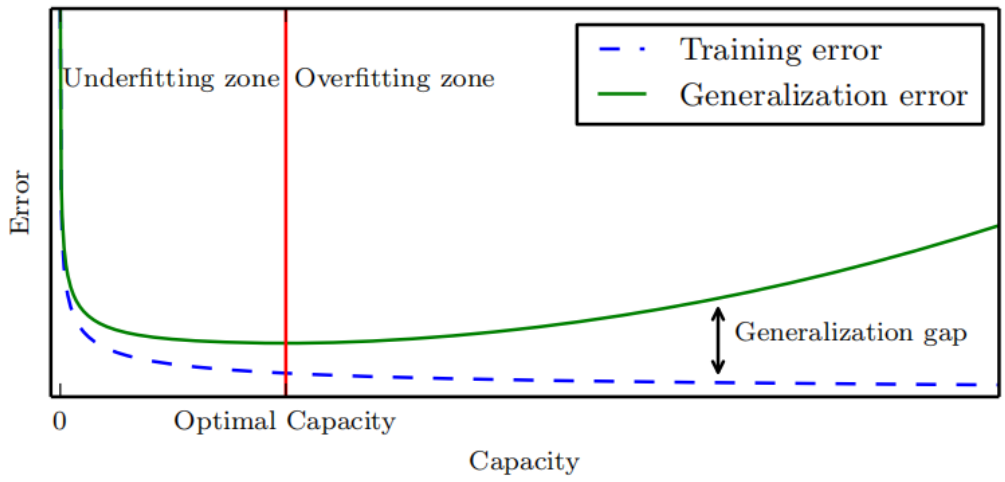
- 线性回归函数将关于其输入的所有线性函数作为假设空间，广义线性回归的假设空间包括多项式函数，会增加模型容量

当机器学习算法的容量适合于所执行任务的复杂度和所提供训练数据的数量时，算法效果通常会最佳



- 线性，二次和九次函数预测器拟合真实二次函数的效果
  - 用一个线性函数拟合数据会导致欠拟合
  - 用二次函数拟合数据在未观察到的点上泛化得很好
  - 一个 9 阶的多项式拟合数据会导致过拟合

容量和误差之间的典型关系



- 左端：训练误差和泛化误差都非常高，这是**欠拟合机制（underfitting regime）**
- 中间：当我们增加容量时，训练误差减小，但是训练误差和泛化误差之间的间距却不断扩大
- 右端：间距的大小超过了训练误差的下降，进入到了**过拟合机制（overfitting regime）**
    - 容量过大，超过了 **最佳容量（optimal capacity）**

没有免费午餐定理 no free lunch theorem

一定程度上，机器学习仅通过**概率法则**就可以避免在**逻辑上需要集合中所有元素的信息才能推断一个规则去描述集合中的元素**这样的逻辑定理

机器学习保证找到一个在**所关注的大多数**样本上可能正确的规则

但是，没有免费午餐定理表明：**在所有可能的数据生成分布上平均之后，每一个分类算法在未事先观测的点上都有相同的错误率**，换言之，**没有一个机器学习算法总是比其他的要好**

这意味着机器学习研究的目标不是**找一个通用学习算法或是绝对最好的学习算法**，反之，我们的目标是**理解什么样的分布与人工智能获取经验的 “真实世界” 相关，什么样的学习算法在我们关注的数据生成分布上效果最好**

正则化 regularization

正则化学习一个函数 $f(\boldsymbol{x};\boldsymbol{\theta})$ 的模型，可以给代价函数添加被称为**正则化项（regularizer）**的惩罚，可以用来表示**对函数的偏好**

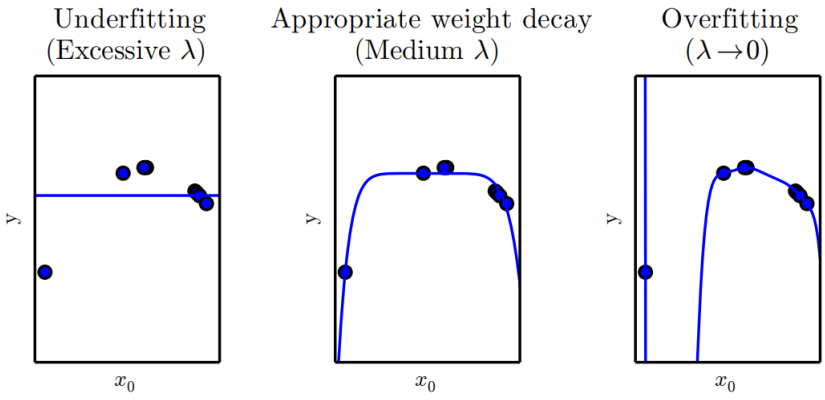
权重衰减 weight decay

在线性回归的示例中，我们可以加入**权重衰减（weight decay）**来修改线性回归的训练标准

带权重衰减的线性回归最小化训练集上的均方误差和正则项的和 $J(\boldsymbol{w})$

$$J(\boldsymbol{w}) = \text{MSE}_{\text{train}} + \lambda \boldsymbol{w}^T \boldsymbol{w}$$

- $\lambda \boldsymbol{w}^T \boldsymbol{w}$：正则化项的加入使得其参数优化偏好于平方 L^2 范数较小的权重
- λ：提前挑选好的值，控制我们偏好小范数权重的程度



- 如图为线性，二次和九次函数预测器拟合真实二次函数的效果（**加上了正则化项的效果**）

通过正则化项调整模型容量

表示对函数的偏好是比增减假设空间的成员函数更一般的控制模型容量的方法

超参数、验证集

超参数（hyper-parameter）：大多数机器学习算法都有超参数，可以设置来**控制算法行为**，它不是通过算法本身学习出来的

- 例如：线性回归的实例中，超参数有多项式的次数，控制权重衰减程度 λ
- 一个选项被设为学习算法不用学习的超参数，是因为它**太难优化**了，或者该选项必须是超参数，它**不适合在训练集上学习**，为此，我们需要一个训练算法无法观测的**验证集（validation set）**样本
    - 这些超参数总是趋向于**最大可能的模型容量**，导致**过拟合**

数据集的分类

- **训练数据**：模型训练的时候用到的数据
    - **训练集（train set）80%**：用于**学习参数**的数据子集通常仍被称为训练集
    - **验证集（validation set）20%**：用于**挑选超参数的数据子集**
- **测试集（test set）**：估计学习过程完成之后的学习器的**泛化误差**

k-折交叉验证

如果**数据集太小**的时候，固定训练数据和测试集是很有问题的

存在这样的替代方法，允许我们使用所有的样本估计平均测试误差，代价是增加了计算量

当给定数据集 $\mathbb{D}$ 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时

- 数据集 $\mathbb{D}$ 包含的元素是抽象的样本 $\boldsymbol{z}^{(i)}$
    - 监督学习的情况下，$\boldsymbol{z}^{(i)} = (\boldsymbol{x}^{(i)}, y^{(i)})$
    - 非监督学习情况下，$\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)}$
- 该算法返回 $\mathbb{D}$ 中中每个示例的误差向量 $\boldsymbol{e}$，其均值是估计的泛化误差

Define $\mathrm{KFoldXV}(\mathbb{D}, A, L, k)$

Input

- $\mathbb{D}$：给定数据集，其中元素为 $\boldsymbol{z}^{(i)}$
- A：学习算法，可视为一个函数
- L：损失函数
- k：折数

Return

- $\boldsymbol{e}$：$\mathbb{D}$ 中每个示例的误差向量

Algorithm

1. 将 $\mathbb{D}$ 分为 k 个互斥子集 $\mathbb{D}_i$，它们的并集为 $\mathbb{D}$
2. **for** i from 1 to k **do**

3. $f_i = A(\mathbb{D} \backslash \mathbb{D}_i)$
4. **for** $\boldsymbol{z}^{(j)}$ in $\mathbb{D}_i$ **do**
5. $e_j = L(f_i, \boldsymbol{z}^{(j)})$
6. **end for**
7. **end for**
8. **Return** e