基本概念

假设训练集由 N 个样本组成，其中每个样本都是独立同分布（Identically and Independently Distributed， IID）的，即独立地从相同的数据分布中抽取的，记为

$$D = \{(\mathbf{x}(1), y(1)), (\mathbf{x}(2), y(2)), \cdots, (\mathbf{x}(N), y(N))\}.$$

给定训练集 D，我们希望让计算机自动寻找一个函数 f(x， θ）来建立每个样本特性向量 x 和标签 y 之间的映射。对于一个样本 x，我们可以通过决策函数来预测其标签的值

$$\hat{y} = f(x, \theta), (2.2)$$

或标签的条件概率

$$p(y|x) = f_y(x, \theta), (2.3)$$

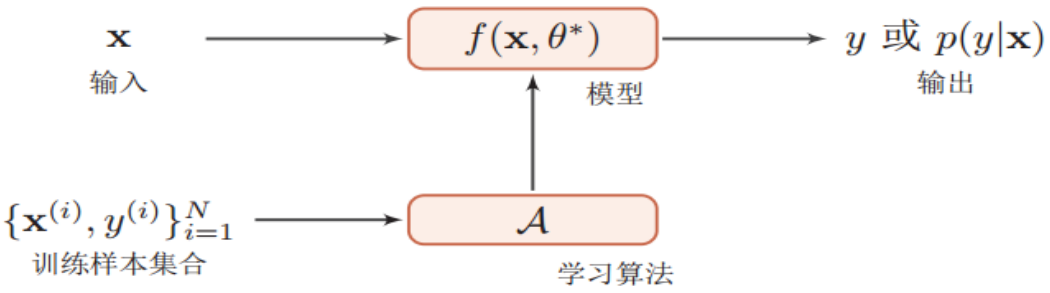
其中 θ 为可学习的参数。

为了评价的公正性，我们还是独立同分布地抽取一组样本作为测试集并在测试集中所有样本上进行测试，计算预测结果的准确率。

$$Acc(f(\mathbf{x}, \theta^*)) = \frac{1}{|\mathcal{D}'|} \sum_{(\mathbf{x},y)\in\mathcal{D}'} I(f(\mathbf{x}, \theta^*) = y),$$

其中 I(•)为指示函数， |D′|为测试集大小

对一个预测任务，输入特征向量为 x，输出标签为 y，我们选择一个函数 f(x， θ)，通过学习算法 A 和一组训练样本 D，找到一组最优的参数 θ∗，得到最终的模型 f(x， θ∗)。这样就可以对新的输入 x 进行预测。



三个基本要素（模型、学习准则、优化算法）

模型：机器学习任务要先需要确定其输入空间 X 和输出空间 Y。

由于我们不知道真实的映射函数 $g(\mathbf{x})$ 或条件概率分布 $p_r(y|\mathbf{x})$ 的具体形式，只能根据经验来确定一个假设函数集合 $\mathcal{F}$，称为假设空间（hypothesis space），然后通过观测其在训练集 $\mathcal{D}$ 上的特性，从中选择一个理想的假设（hypothesis）$f^* \in \mathcal{F}$。

假设空间 $\mathcal{F}$ 通常为一个参数化的函数族

$$\mathcal{F} = \{f(\mathbf{x}, \theta)|\theta \in \mathbb{R}^m\}, \tag{2.7}$$

其中 $f(\mathbf{x}, \theta)$ 为假设空间中的模型，θ 为一组可学习参数，m 为参数的数量。

学习准则：

不仅仅是拟合训练集上的数据，同时也要使得泛化错误最低。机器学习可以看作是一个从有限、高维、有噪声的数据上得到更一般性规律的泛化问题。

一个好的模型 $f(\mathbf{x}, \theta^*)$ 应该在所有 $(\mathbf{x}, y)$ 的可能取值上都与真实映射函数 $y = g(\mathbf{x})$ 一致，即

$$|f(\mathbf{x}, \theta^*) - y| < \epsilon, \qquad \forall(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}, \tag{2.11}$$

或与真实条件概率分布 $p_r(y|\mathbf{x})$ 一致，即

$$|f_y(\mathbf{x}, \theta^*) - p_r(y|\mathbf{x})| < \epsilon, \qquad \forall(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}, \tag{2.12}$$

其中 ϵ 是一个很小的正数，$f_y(\mathbf{x}, \theta^*)$ 为模型预测的条件概率分布中 y 对应的概率。

模型 $f(\mathbf{x}, \theta)$ 的好坏可以通过期望风险（Expected Risk）$\mathcal{R}(\theta)$ 来衡量。

$$\mathcal{R}(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim p_r(\mathbf{x},y)}[\mathcal{L}(y, f(\mathbf{x}, \theta))], \tag{2.13}$$

其中 $p_r(\mathbf{x}, y)$ 为真实的数据分布，$\mathcal{L}(y, f(\mathbf{x}, \theta))$ 为损失函数，用来量化两个变量之间的差异。

损失函数：

0-1 损失函数：能够客观的评价模型的好坏，但缺点是数学性质不是很好：不连续且导数为 0，难以优化。

平方损失函数：一般不适用于分类问题，回归

交叉熵损失函数：用于分类问题

Hinge 损失函数：$L=max(0,1-yf(x,w))$(二分类情况)

风险最小化准则：

一个好的模型 f(x, θ)应当有一个比较小的期望错误，但由于不知道真实的数据分布和映射函数，切实可行的学习准则是找到一组参数 $\theta*$ 使得经验风险最小。

根据大数定理可知，当训练集大小 |D|趋向于无穷大时，经验风险就趋向于期望风险。

过拟合：训练数据少和噪声以及模型能力强等原因

结构风险最小化（Structure Risk Minimization， SRM）：在经验风险最小化的基础上再引入参数的正则化（regularization），来限制模型能力，使其不要过度地最小化经验风险。

学习算法：

参数：f(x, θ)中的 θ 称为模型的参数，可以通过优化算法进行学习

超参数（hyper-parameter）：用来定义模型结构或优化策略的。聚类算法中的类别个数、梯度下降法的步长、正则项的系数、神经网络的层数、支持向量机中的核函数等。超参数的选取一般都是组合优化问题，很难通过优化算法来自动学习。

批量梯度下降（Batch Gradient Descent， BGD）：每次迭代时需要计算每个样本上损失函数的梯度并求和。当训练集中的样本数量 N 很大时，空间复杂度比较高，每次迭代的计算开销也很大。

提前停止（early stop，ES）:每次迭代时，把新得到的模型 f(x, θ)在验证集上进行测试，并计算错误率。如果在验证集上的错误率不再下降，就停止代。

随机梯度下降（Stochastic Gradient Descent， SGD）：单个样本进行调整。

小批量梯度下降法（Mini-Batch Gradient Descent）：既可以兼顾随机梯度下降法的优点，也可以提高训练效率。

参数学习：

经验风险最小化（各个特征之间要相互独立**）**：1）满秩，直接求解；2）不可逆，利用主成分预处理，消除不同特征之间的相关性；3）梯度下降法

结构风险最小化：特征之间可能会有较大的共线性，对角线元素都加上一个常数 λI，岭回归

最大似然估计（Maximum Likelihood Estimate， MLE）：条件概率 p(y|x)服从某个未知分布

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

偏差-方差分解

最小化期望错误等价于最小化偏差和方差之和。方差一般会随着训练样本的增加而减少。随着模型复杂度的增加，模型的拟合能力变强，偏差减少而方差增大，从而导致过拟合。

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

$$=\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\mathbf{x})-\mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})]+\mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})]-f^*(\mathbf{x})\right)^2\right] \tag{2.64}$$

$$=\underbrace{\left(\mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})]-f^*(\mathbf{x})\right)^2}_{(\text{bias})^2}+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\mathbf{x})-\mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})]\right)^2\right]}_{\text{variance}}. \tag{2.65}$$

其中第一项为偏差（bias），是指一个模型的在不同训练集上的平均性能和最优模型的差异。偏差可以用来衡量一个模型的拟合能力；第二项是方差（variance），是指一个模型在不同训练集上的差异，可以用来衡量一个模型是否容易过拟合。

1）当一个模型在训练集上的错误率比较高时，说明模型的拟合能力不够，偏差比较高。这种情况可以增加数据特征、提高模型复杂度，减少正则化系数等操作来改进模型。当模型在训练集上的错误率比较低，但验证集上的错误率比较高时，说明模型过拟合，方差比较高。这种情况可以通过降低模型复杂度，加大正则化系数，引入先验等方法来缓解；2）集成模型，即通过多个高方差模型的平均来降低方差。

机器学习算法的类型

	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 τ 和累积奖励 G_τ
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

数据的特征表示

（1）特征比较单一，需要进行（非线性的）组合才能发挥其作用；（2）特征之间冗余度比较高；（3）并不是所有的特征都对预测有用；（4）很多特征通常是易变的；（5）特征中往往存在一些噪声。

特征选择（Feature Selection）是选取原始特征集合的一个有效子集，使得基于这个特征子集训练出来的模型准确率最高。保留有用特征，移除冗余或无关的特征。贪心的策略

前向搜索（forward search）：由空集合开始，每一轮添加该轮最优的特征；

反向搜索（backward search）：从原始特征集合开始，每次删除最无用的特征；

过滤式（filter）：不依赖具体的机器学习模型。每次增加最有信息量的特征，或删除最没有信息量的特征，通过信息增益（information gain）来衡量；

包裹式（wrapper）：用后续机器学习模型的准确率来评价一个特征子集。每次增加对后续机器学习模型最有用的特征，或删除对后续机器学习任务最无用的特征。将机器学习模型包裹到特征选择过程的内部；

L1 正则化：会导致稀疏特征，间接实现了特征选择。

特征抽取（Feature Extraction）是构造一个新的特征空间，并将原始特征投影在新的空间中。监督的特征学习的目标是抽取对一个特定的预测任务最有用的特征，比如线性判别分析（Linear DiscriminantAnalysis， LDA）。而无监督的特征学习和具体任务无关，其目标通常是减少冗余信息和噪声，比如主成分分析（Principle Components Analysis， PCA）。

	监督学习	无监督学习
特征选择	标签相关的子集搜索、ℓ_1 正则化、决策树	标签无关的子集搜索
特征抽取	线性判别分析	主成分分析、独立成分分析、流形学习、自编码器

可以用较少的特征来表示原始特征中的大部分相关信息，去掉噪声信息，并进而提高计算效率和减小维度灾难（Curse Of Dimensionality）。

评价指标

准确率 最常用的的评价指标为准确率（Accuracy）

$$\mathcal{ACC} = \frac{1}{N}\sum_{n=1}^{N} I(y^{(n)} = \hat{y}^{(n)}), \tag{2.71}$$

其中 $I(\cdot)$ 为指示函数。

错误率 和准确率相对应的就是错误率（Error Rate）。

$$\mathcal{E} = 1 - \mathcal{ACC} \tag{2.72}$$

$$= \frac{1}{N}\sum_{n=1}^{N} I(y^{(n)} \neq \hat{y}^{(n)}). \tag{2.73}$$

查准率和查全率：准确率是所有类别整体性能的平均，如果希望对每个类都进行性能估计，就需要计算查准率和查全率。查准率和查全率是广泛用于信

息检索和统计学分类领域的两个度量值。

1. 真正例（True Positive，TP）：一个样本的真实类别为c并且模型正确地预测为类别c。这类样本数量记为

$$TP_c = \sum_{n=1}^{N} I(y^{(n)} = \hat{y}^{(n)} = c). \tag{2.74}$$

2. 假负例（False Negative，FN）：一个样本的真实类别为c，模型错误地预测为其它类。这类样本数量记为

$$FN_c = \sum_{n=1}^{N} I(y^{(n)} = c \wedge \hat{y}^{(n)} \neq c). \tag{2.75}$$

3. 假正例（False Positive，FP）一个样本的真实类别为其它类，模型错误地预测为类c。这类样本数量记为

$$FP_c = \sum_{n=1}^{N} I(y^{(n)} \neq c \wedge \hat{y}^{(n)} = c). \tag{2.76}$$

4. 真负例（True Negative，TN）：一个样本的真实类别为其它类，模型也预测为其它类。这类样本数量记为TN_c。对于类别c来说，这种情况一般不需要关注。

		预测类别	
		$\hat{y} = c$	$\hat{y} \neq c$
真实类别	$y = c$	TP_c	FN_c
	$y \neq c$	FP_c	TN_c

查准率（Precision），也叫精确率或精度，类别c的查准率为是所有预测为类别c的样本中，预测正确的比例。

$$\mathcal{P}_c = \frac{TP_c}{TP_c + FP_c}, \tag{2.77}$$

查全率（Recall），也叫召回率，类别c的查全率为是所有真实标签为类别c的样本中，预测正确的比例。

$$\mathcal{R}_c = \frac{TP_c}{TP_c + FN_c}, \tag{2.78}$$

F值（F Measure）是一个综合指标，为查准率和查全率的调和平均。

$$\mathcal{F}_c = \frac{(1 + \beta^2) \times \mathcal{P}_c \times \mathcal{R}_c}{\beta^2 \times \mathcal{P}_c + \mathcal{R}_c}, \tag{2.79}$$

其中β用于平衡查全率和查准率的重要性，一般取值为1。$\beta = 1$时的F值称为F1值，是查准率和查全率的调和平均。

宏平均和微平均：为了计算分类算法在所有类别上的总体准确率、召回率和 F1 值，经常使用两种平均方法，分别称为宏平均（macro average）和微平均（micro average）

$$\mathcal{P}_{macro} = \frac{1}{C}\sum_{c=1}^{C}\mathcal{P}_c, \tag{2.80}$$

$$\mathcal{R}_{macro} = \frac{1}{C}\sum_{c=1}^{C}\mathcal{R}_c, \tag{2.81}$$

$$\mathcal{F}1_{macro} = \frac{2\times\mathcal{P}_{macro}\times R_{macro}}{P_{macro}+R_{macro}}. \tag{2.82}$$

宏平均是每一类的性能指标的算术平均值，而微平均是每一个样本的性能指标的算术平均。对于单个样本而言，它的准确率和召回率是相同的（要么都是 1，要么都是 0）。因此准确率的微平均和召回率的微平均是相同的。同理，F1 值的微平均指标是相同的。当不同类别的样本数量不均衡时，使用宏平均会比微平均更合理些。宏平均会更关注于小类别上的评价指标。

交叉验证（Cross Validation）：一种比较好的可能衡量机器学习模型的统计分析方法，可以有效避免划分训练集和测试集时的随机性对评价结果造成的影响。我们可以把原始数据集平均分为 K 组不重复的子集，每次选 K-1 组子集作为训练集，剩下的一组子集作为验证集。这样可以进行 K 次试验并得到 K 个模型。这 K 个模型在各自验证集上的错误率的平均作为分类器的评价。

理论和定理

PAC 学习理论：PAC 可学习的算法是指该学习算法能够在多项式时间内从合理数量的训练数据中学习到一个近似正确的 f(x)。

$$P\Big((\mathcal{R}(f)-\mathcal{R}_{\mathcal{D}}^{emp}(f))\le\epsilon\Big)\ge 1-\delta, \tag{2.85}$$

其中 ϵ,δ 是和样本数量 n、假设空间 $\mathcal{F}$ 相关的变量。如果固定 ϵ,δ，可以反过来计算出样本复杂度为

$$n(\epsilon,\delta)\ge\frac{1}{2\epsilon^2}(\ln|\mathcal{F}|+\ln\frac{2}{\delta}), \tag{2.86}$$

其中 $|\mathcal{F}|$ 为假设空间的大小。

如果希望模型的假设空间越大，泛化错误越小，其需要的样本数量越多。

不存在一种机器学习算法适合于任何领域或任务；

不存在相似性的客观标准，一切相似性的标准都是主观的；

简单的模型泛化能力更好。如果有两个性能相近的模型，我们应该选择更简单的模型。