第二章　机器学习概述

> 机器学习是对能通过经验自动改进的计算机算法的研究。
>
> — Mitchell [1997]

在介绍神经网络之前，我们先来了解下机器学习的基本概念。

通俗地讲，**机器学习**（Machine Learning，ML）就是让计算机从数据中进行自动学习，得到某种知识（或规律）。作为一门学科，机器学习通常指一类问题以及解决这类问题的方法，即如何从观测数据（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。

机器学习问题在早期的工程领域也经常称为**模式识别**（Pattern Recognition，PR），但模式识别更偏向于具体的应用任务，比如光学字符识别、语音识别，人脸识别等。这些任务的特点是对于我们人类而言，这些任务很容易完成，但我们不知道自己是如何做到的，因此也很难人工设计一个计算机程序来解决这些任务。一个可行的方法是设计一个算法可以让计算机自己从有标注的样本上学习其中的规律，并用来完成各种识别任务。随着机器学习技术的应用越来越广，现在机器学习的概念逐渐替代模式识别，成为这一类问题及其解决方法的统称。

以手写体数字识别为例，我们需要让计算机能自动识别手写的数字。比如像图2.1中的例子，将  识别为数字5，将  识别为数字6。手写数字识别是一个经典的机器学习任务，对人来说很简单，但对计算机来说却十分困难。我们很难总结每个数字的手写体特征，或者区分不同数字的规则，因此设计一套识别算法几乎是一项几乎不可能的任务。在现实生活中，很多问题都类似于手写体数字识别这类问题，比如物体识别、语音识别等。对于这类问题，我们不知道

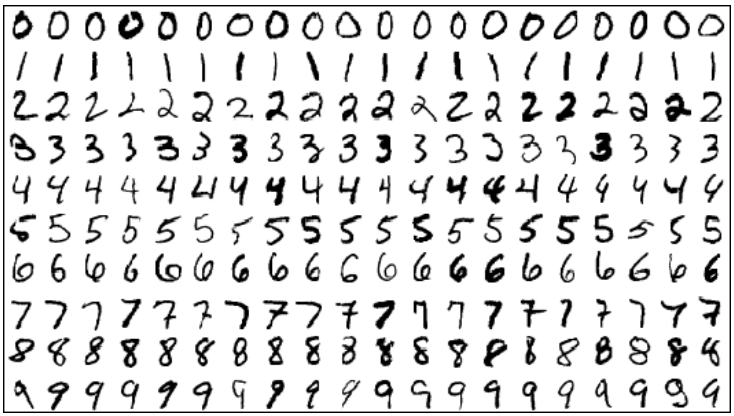

图 2.1: 手写体数字识别示例。图片来源：MNIST 数据集 [LeCun et al., 1998]

如何设计一个计算机程序来解决，即使可以通过一些启发式规则来实现，其过程也是极其复杂的。因此，人们开始尝试采用另一种思路，即让计算机“看”大量的样本，并从中学习到一些经验，然后用这些经验来识别新的样本。要识别手写体数字，首先通过人工标注大量的手写体数字图像（即每个图像都人工标记了它是什么数字），这些图像作为训练数据，然后通过学习算法自动生成一套模型，并依靠它来识别新的手写体数字。这和人类学习过程也比较类似，我们教小孩子识别数字也是这样的过程。这种通过数据来学习的方法就称为机器学习的方法。

2.1 基本概念

首先介绍下机器学习中的一些基本概念：包括样本、特征、标签、模型、学习算法等。

以一个生活中的经验学习为例，假设我们要到市场上购买芒果，但是之前毫无挑选芒果的经验，那么我们如何通过学习来获取这些知识？

特征也可以称为属性（attribute）。

首先，我们从市场上随机选取一些芒果，列出每个芒果的**特征**（feature），包括颜色，大小，形状，产地，品牌，以及我们需要预测的**标签**（Label）。标签可以连续值（比如关于芒果的甜度、水分以及成熟度的综合打分），也可以是离散值（比如“好”“坏”两类标签）。

一个标记好特征以及标签的芒果可以看作是一个**样本**（sample）。一组样本构成的集合称为**数据集**（Data Set）。一般将数据集分为两部分：训练集和测试集。**训练集**（training set）中的样本是用来训练模型的，也叫**训练样本**（training sample），而**测试集**（test set）中的样本是用来检验模型好坏的，也叫**测试样本**（test sample）。

样本（Sample），也叫示例（Instance）。

我们用一个 d 维向量 $\mathbf{x} = [x_1, x_2, \cdots, x_d]^{\mathrm{T}}$ 表示一个芒果的所有特征构成的向量，称为**特征向量**（feature vector），其中每一维表示一个特征。

在很多领域，数据集也经常称为语料库（corpus）。

并不是所有的样本特征都是数值型，需要通过转换表示为特征向量。

假设训练集由 N 个样本组成，其中每个样本都是独立地从相同的数据分布中抽取的，记为

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}. \tag{2.1}$$

给定训练集 $\mathcal{D}$，我们希望让计算机自动寻找一个**函数** $f(\mathbf{x}, \theta)$ 来建立每个样本特性向量 $\mathbf{x}^{(i)}$ 和标签 $y^{(i)}$ 之间的映射。对于一个样本 $\mathbf{x}$，我们可以通过决策函数来预测其标签 $\hat{y}$。

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

其中 θ 为可学习的参数。

通过一个**学习算法**（learning algorithm）$\mathcal{A}$，在训练集上找到一组参数 θ^*，使得函数 $f(\mathbf{x}, \theta^*)$ 可以近似真实的映射关系。这个过程称为**学习**（learning）或**训练**（training）过程，学习得到函数 $f(\mathbf{x}, \theta^*)$ 称为**模型**（model）。

学习算法在有些文献中也叫作学习器（learner）。

下次从市场上买芒果（测试样本）时，可以根据芒果的特征，使用学习到的模型来预测芒果的好坏。

为了评价模型 $f(\mathbf{x}, \theta^*)$ 的好坏，我们用测试集 $\mathcal{D}'$ 中的样本上进行测试，计算预测准确率。

$$Acc = \frac{1}{|\mathcal{D}'|} \sum_{i=1}^{|\mathcal{D}'|} I(f(\mathbf{x}^{(i)} = y^{(i)}), \tag{2.3}$$

其中 $I(\cdot)$ 为指示函数，$|\mathcal{D}'|$ 为测试集大小。第2.6节中会介绍更多的评价方法。

图2.2给出了机器学习的基本概念。对一个预测任务，输入特征向量为 $\mathbf{x}$，输出标签为 y，我们选择一个函数 $f(\mathbf{x}, \theta)$，通过**学习算法** $\mathcal{A}$ 和一组训练样本 $\mathcal{D}$，找到一组最优的参数 θ^*，得到最终的模型 $f(\mathbf{x}, \theta^*)$。有了模型，就可以对新的（训练集中没有见过）的输入 $\mathbf{x}$ 进行预测。

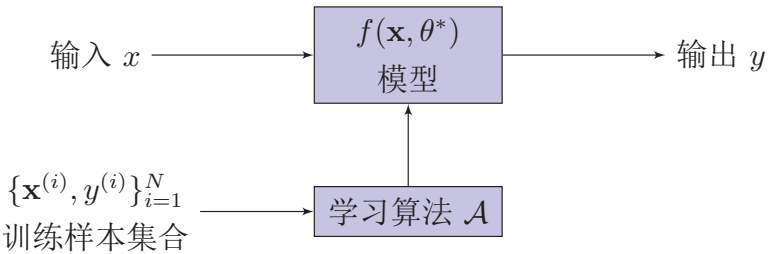

图 2.2: 机器学习系统示例

2.1.1 其它基本概念

数据 在计算机科学中，**数据**是指所有能计算机程序处理的对象的总称，可以是数字、字母和符号等。在不同的任务中，表现形式不一样，比如图像、声音、文字、传感器数据等。

在实际应用中，我们要处理的输入样本的类型也是多种多样，比较有代表性的类型为文本、音频、图像、视频等。不同类型的数据，其原始的特征空间也不相同。比如一张灰度图像（像素数为n）的特征空间为$[0, 255]^n$，一个自然语言句子（长度为L）的特征空间为$\mathcal{V}^L$，其中$\mathcal{V}$为词表集合。因此，在机器学习之前我们需要将这些不同类型的数据转换为特征向量。

在数字手写体识别中，样本x为待识别的图像，类别$y \in \{0, 1, \cdots, 9\}$分别对应10个不同数字。为了识别$x$是什么数字，我们可以从图像中抽取一些特征。如果图像是一张大小为$m \times n$的图片，其特征向量可以简单地表示为mn维的向量，每一维的值为图片中对应像素的灰度值。为了提高模型准确率，也会经常加入一个额外的特征，比如直方图、宽高比、笔画数，纹理特征、边缘特征等。假设我们总共抽取了d个特征，这些特征可以表示为一个向量$\mathbf{x} \in \mathbb{R}^d$。

在情感分类中，样本x为自然语言文本，类别$y \in \{+1, -1\}$分别表示正面或负面的评价。为了将样本x从文本形式转为为向量形式，我们可以使用**词袋模型**（Bag-of-Words，BoW）模型。假设训练样本对应的词典$\mathcal{V}$中包含v个词，则每个文本可以表示为一个维度为v的向量$\mathbf{x} \in \mathbb{R}^v$，向量中每一维对应词典中的一个词。如果向量中某一维对应的词在文本中出现，其值为1，否则为0。

正例和负例 对于两类分类问题，类别标签$y \in \{+1, -1\}$可以直接用正负号表示。因此，常用**正例**和**负例**来分别表示属于类别$+1$和-1的样本。

参数与超参数 机器学习可以归结为学习一个映射函数 $f : \mathbf{x} \to y$，将输入变量 $\mathbf{x}$ 映射为输出变量 y。一般我们可以假设映射函数为 $y = f(\mathbf{x}, \theta)$。其中 θ 即为函数的**参数**。参数可以通过学习算法进行学习。

除了可学习的参数之外，还有一类参数是用来定义模型结构或训练策略的，这类参数叫做**超参数**（Hyper-Parameter）。超参数和可学习的参数不同，通常是按照人的经验设定，或者通过网格搜索对一组超参数组合进行不断试错调整。

在贝叶斯方法中，超参数可以理解为参数的参数，即控制模型参数分布的参数。

常见的超参数包括：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。因此，优化超参数是机器学习的一个经验性很强的技术。

特征学习 原始数据的特征有很多，但是并不是所有的特征都是有用的。并且，很多特征通常是冗余并且易变的。我们需要抽取有效的、稳定的特征。传统的特征提取是通过人工方式进行的，这需要大量的人工和专家知识。即使这样，人工设计的特征在很多任务上也不能满足需要。因此，如何自动地学习有效的特征也成为机器学习中一个重要的研究内容，也就是**特征学习**，也叫**表示学习**。特征学习可以简化模型、缩短训练时间、提高模型泛华能力、避免过拟合等。

特征学习分成两种：特征选择和特征抽取。**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。简单地说，特征选择就是保留有用特征，移除冗余或无关的特征。假设原始的特征个数为 d，则共用有 2^d 个候选子集。特征选择的目标是选择一个最优的候选子集。最暴力的做法是测试每个特征子集，看机器学习模型哪个子集上的准确率最高。但这种方式效率太低。常用的方法是采样贪心的策略，由空集合开始，每一轮添加该轮最优的特征；或者从原始特征集合开始，每次删除最无用的特征。

例子

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。以线性投影为例，原始特征向量 $\mathbf{x} \in \mathbb{R}^d$，经过线性投影后得到在新空间中的特征向量 $\mathbf{x}'$。

$$\mathbf{x}' = P\mathbf{x}, \tag{2.4}$$

其中 $P \in \mathbb{R}^{k \times d}$ 为映射矩阵。

经典的特征抽取方法有主成分分析(Principle Components Analysis，PCA)和线性判别分析（Linear Discriminant Analysis，LDA）等。

特征选择和特征抽取又都可以分为有监督和和无监督学习。有监督学习目标是抽取对模型最有用的特征，而无监督学习的目标是减少冗余信息。

特征选择和特征抽取的优点是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）。对于很多没有正则化的模型，特征选择和特征抽取非常必要。

正则化参见第??节，第??页。

经过特征选择或特征抽取后，特征的数量一般会减少，因此特征选择和特征抽取也经常称为**维数约减**或**降维**（Dimension Reduction）。

2.2 机器学习的三个基本要素

机器学习是从有限的观测数据中学习（或“猜测”）出具有一般性的规律，并可以将总结出来的规律推广应用到未观测样本上。机器学习方法可以粗略地分为三个基本要素：模型、学习准则、优化算法。

2.2.1 模型

这里，“输入空间”默认为样本的特征空间。

一个机器学习任务要先需要确定其输入空间 $\mathcal{X}$ 和输出空间 $\mathcal{Y}$。不同机器学习任务的主要区别在于输出空间不同。如果是两类分类问题，$\mathcal{Y} = \{+1, -1\}$。如果是 K 类分类问题，$\mathcal{Y} = \{1, 2, \cdots, K\}$。如果是回归问题，$\mathcal{Y} = \mathbb{R}$。

输入空间 $\mathcal{X}$ 和输出空间 $\mathcal{Y}$ 构成了一个样本空间。对于样本空间中的样本 $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$，假定存在一个未知的**目标函数**（Target Function）$g : \mathcal{X} \to \mathcal{Y}$ 使得

这里的“目标函数”为需要去估计的未知函数。和优化中目标函数(Objective Function)意义不同。

$$y = g(\mathbf{x}), \tag{2.5}$$

或者一个未知的目标条件概率分布

$$p(y|\mathbf{x}), \tag{2.6}$$

机器学习的目标是找到一个模型 f^* 来近似目标函数或目标条件概率分布。

由于我们不知道真实目标函数 $g(\mathbf{x})$ 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。

常见的假设空间可以分为线性和非线性两种，对应的模型 f 也分别称为线性模型和非线性模型。

线性模型 线性的假设空间为一个参数化的线性函数族，

对于分类问题，一般为广义线性函数。

$$f(\mathbf{x}, \theta) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b, \tag{2.8}$$

其中参数 θ 包含了权重向量 $\mathbf{w}$ 和偏置 b。

非线性模型 非线性的假设空间可以为多个非线性基函数 $\phi(\mathbf{x})$ 的线性组合

$$f(\mathbf{x}, \theta) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b, \tag{2.9}$$

其中 $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_1(\mathbf{x}), \cdots, \phi_k(\mathbf{x})]^{\mathrm{T}}$ 为 k 个非线性基函数组成的向量，参数 θ 包含了权重向量 $\mathbf{w}$ 和偏置 b。如果 $\boldsymbol{\phi}(\mathbf{x})$ 本身为可学习的基函数，则 $f(\mathbf{x}, \theta)$ 可以看作是一个*神经网络*。

2.2.2 学习准则

假设训练集 $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ 是由 N 个**独立同分布**（Identically and Independently Distributed，IID）的样本组成，即每个样本 $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ 是从 $\mathcal{X}$ 和 $\mathcal{Y}$ 的联合空间中按照某个未知分布 $p(\mathbf{x}, y)$ 独立地随机产生的。这里要求样本分布 $p(\mathbf{x}, y)$ 未知但必须是固定的，不会随时间而变化。如果 $p(\mathbf{x}, y)$ 本身可变的话，我们就无法通过这些数据进行学习。

一个好的模型 $f(\mathbf{x}, \theta^*)$ 应该在所有 $(\mathbf{x}, y)$ 的可能取值上都和目标函数 $y = g(\mathbf{x})$ 一致，即

$$f(\mathbf{x}, \theta^*) \sim y, \qquad \forall (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}. \tag{2.10}$$

期望风险也称为期望错误（Expected Error）。

模型 $f(\mathbf{x}, \theta)$ 和目标函数 $g(\mathbf{x})$ 的差异可以通过**期望风险**（Expected Risk）$\mathcal{R}(\theta)$ 来衡量。

$$\mathcal{R}(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}, \theta), y)], \tag{2.11}$$

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

损失函数

损失函数是一个非负实数函数，用来量化模型预测和真实标签之间的差异。最直接的损失函数为错误率，即 0-1 损失。

0-1 损失函数（0-1 loss function）

$$\mathcal{L}(y, f(\mathbf{x}, \theta)) = \begin{cases} 0 & \textbf{if } y = f(\mathbf{x}, \theta) \\ 1 & \textbf{if } y \neq f(\mathbf{x}, \theta) \end{cases} \tag{2.12}$$

$$= I(y \neq f(\mathbf{x}, \theta)), \tag{2.13}$$

其中 $I(\cdot)$ 是指示函数。

虽然 0-1 损失能够客观的评价模型的好坏，但是有个缺点就是数学性质不是很好，比较难以优化。因此经常用以下几类损失函数替代。

平方损失函数（quadratic loss function）

$$\mathcal{L}(y, f(\mathbf{x}, \theta)) = (y - f(\mathbf{x}, \theta))^2. \tag{2.14}$$

平方损失函数经常用在预测标签 y 为实数值的任务中。

交叉熵损失函数（cross-entropy loss function） 对于分类问题，预测目标 y 为离散的类别，模型输出 $f(\mathbf{x}, \theta)$ 为每个类的条件概率。假设 $y \in \{1, \cdots C\}$，模型预测的第 i 个类的条件概率 $P(y = i|x) = f_i(\mathbf{x}, \theta)$，则 $f(\mathbf{x}, \theta)$ 满足

$$f_i(\mathbf{x}, \theta) \in [0, 1], \qquad \sum_{i=1}^{C} f_i(\mathbf{x}, \theta) = 1 \tag{2.15}$$

如果我们用 C 维的 one-hot 向量 $\mathbf{y} \in \{0,1\}^C$ 来表示目标类别 c，其中只有 $y_c = 1$，其余的向量元素都为0。$y_i, 1 \leq i \leq C$ 也可以看成是标签为 i 的概率。

我们可以用交叉熵来衡量真实的标签分布 y_i 和预测分布 $f_i(\mathbf{x}, \theta)$ 直接的差异。

$$\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \theta)) = -\sum_{i=1}^{C} y_i \log f_i(\mathbf{x}, \theta). \tag{2.16}$$

对于一个三类分类问题，类别为 $[0,0,1]$，预测的类别概率为 $[0.3,0.3,0.4]$，则

$$\begin{aligned}\mathcal{L}(\theta) &= -\big(0 \times \log(0.3) + 0 \times \log(0.3) + 1 \times \log(0.4)\big) \\ &= -\log(0.4).\end{aligned}$$

因为 $\mathbf{y}$ 为 one-hot 向量，公式 (2.16) 也可以写为

$$\mathcal{L}(y, f(\mathbf{x}, \theta)) = -\log f_y(x, \theta), \tag{2.17}$$

其中 $f_y(\mathbf{x}, \theta)$ 可以看作真实类别 y 的似然函数。因此，交叉熵损失函数也就是**负对数似然损失函数**（negative log-likelihood function）。

Hinge损失函数（hinge loss function） 对于两类分类问题，假设 y 和 $f(x, \theta)$ 的取值为 $\{-1, +1\}$。Hinge 损失函数的定义如下：

$$\mathcal{L}(y, f(x, \theta)) = \max(0, 1 - yf(x, \theta)) \tag{2.18}$$

$$= |1 - yf(x, \theta)|_+. \tag{2.19}$$

风险最小化准则

一个好的模型 $f(\mathbf{x}, \theta)$ 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目标函数，实际上无法计算期望风险 $\mathcal{R}(\theta; \mathbf{x}, y)$。给定一个训练集 $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$，我们可以计算的是**经验风险**（empirical risk），即在训练集上的平均损失。

经验风险也称为经验错误（empirical error）。

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y^{(i)}, f(x^{(i)}, \theta)). \tag{2.20}$$

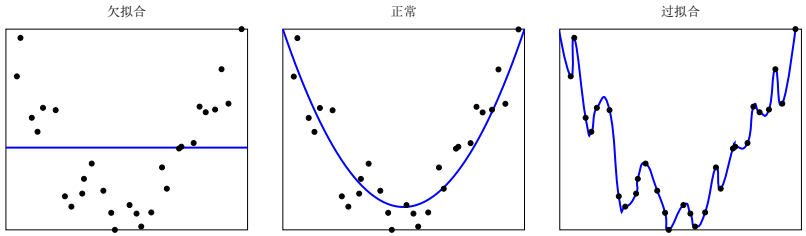

图 2.3: 欠拟合和过拟合示例

因此，一个切实可行的学习准则是找到一组参数θ^*使得经验风险最小，

$$\theta^* = \arg\min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta), \tag{2.21}$$

这就是**经验风险最小化**（empirical risk minimization，ERM）准则。

根据大数定理，当训练集大小$|\mathcal{D}|$趋向于无穷大时，经验风险就趋向于期望风险。然而通常情况下，我们无法获取无限的训练样本，并且训练样本往往是真实数据的一个很小的子集或者包含一定的噪声数据，不能很好地反映全部数据的真实分布。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。这就是所谓的**过拟合**（overfitting）。

如何选择训练样本个数可以参考PAC理论，第2.7.1节，第38页。

过拟合的标准定义为：给定一个假设空间$\mathcal{F}$，一个假设f属于$\mathcal{F}$，如果存在其他的假设f'也属于$\mathcal{F}$,使得在训练集上f的损失比f'小，但在整个样本空间上f'比f的损失小，那么就说假设f过度拟合训练数据 [Mitchell, 1997]。

过拟合问题往往是由于训练数据少和噪声以及模型能力强等原因造成的。为了解决过拟合问题，一般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。这种准则就是**结构风险最小化**（structure risk minimization，SRM）准则。

$$\theta^* = \arg\min_{\theta} \mathcal{R}_{\mathcal{D}}^{struct}(\theta) \tag{2.22}$$

$$= \arg\min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta) + \lambda\|\theta\|_2^2 \tag{2.23}$$

$$= \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(y^{(i)}, f(x^{(i)}, \theta)) + \lambda\|\theta\|^2, \tag{2.24}$$

其中$\|\theta\|_2$是L_2范数的**正则化项**，用来减少参数空间，避免**过拟合**；λ用来控制正则化的强度。

正则化项也可以使用其它函数，比如 L_1 范数。L_1 范数的引入通常会使得参数有一定稀疏性，因此在很多算法中也经常使用。在 Bayes 估计的角度来讲，正则化是假设了参数的先验分布，不完全依赖训练数据。

总之，机器学习中的学习准则并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。给定一个训练集，机器学习的目标是从假设空间中找到一个泛化错误较低的“理想”模型，以便更好地对未知的样本进行预测，特别是不在训练集中出现的样本。因此，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。

2.2.3 优化算法

在确定了训练集 $\mathcal{D}$、假设空间 $\mathcal{F}$ 以及学习准则后，如何找到最优的模型 $f(\mathbf{x}, \theta^*)$ 就成了一个**优化**（optimization）问题。机器学习的训练过程其实就是最优化问题的求解过程。

为了可以早到最优的模型以及充分利用凸优化中一些高效、成熟的优化方法，比如共轭梯度、拟牛顿法等，很多机器学习方法都都倾向于选择合适的假设空间和风险函数以构造一个凸的优化目标函数。但也有很多模型，比如神经网络，的优化目标是非凸的，只能退而求其次找到局部最优解。

不同机器学习算法的区别在于假设空间和学习算法（学习准则 + 优化算法）的差异。相同的假设空间也可以有不同的学习算法。比如，线性分类模型有感知器，logistic 回归和支持向量机，它们之间的差异在于使用了不同的学习准则和优化算法。

在最优化问题中，最简单、常用的优化算法就是随机梯度下降法。

梯度下降法参见第A.2.2节，第240页。

梯度下降是求得所有样本上的风险函数最小值，也叫做**批量梯度下降法**。

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathcal{R}(\theta)}{\partial \theta_t} \tag{2.25}$$

$$= \theta_t - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}\big(\theta_t; x^{(i)}, y^{(i)}\big)}{\partial \theta}. \tag{2.26}$$

在机器学习，搜索步长 α 中也叫作**学习率**（learning rate）。

提前停止 针对梯度下降的优化算法，除了加正则化项之外，还可以通过提前停止来防止过拟合。

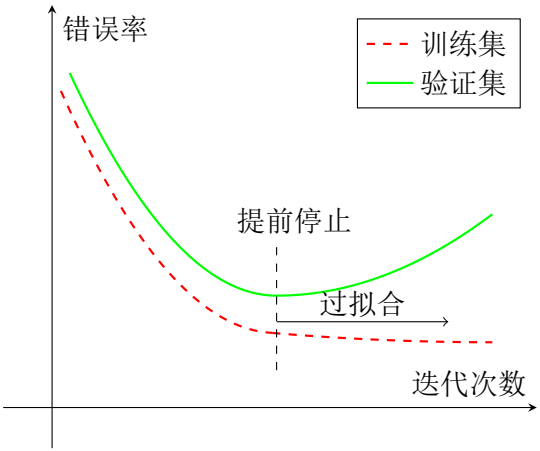

图 2.4: 前提停止

验证集也叫开发集（development set）。

在梯度下降训练的过程中，由于过拟合的原因，在训练样本上收敛的参数，并不一定在测试集上最优。因此，除了训练集和测试集之外，有时也会使用一个**验证集**（validation set）来进行模型选择，测试模型在验证集上是否最优。在每次迭代时，把新得到的模型 $f(\mathbf{x}, \theta)$ 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，就停止迭代。这种策略叫**提前停止**（early-stop）。如果没有验证集，可以在训练集上进行**交叉验证**（cross-validation）。

随机梯度下降法也叫增量梯度下降。

随机梯度下降法 机器学习中的风险函数通过为每个样本损失函数的平均，并且每个样本都是独立的。如公式 (2.26) 所示，批量梯度下降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。为了减少每次迭代的计算复杂度，可以在每次迭代时随机选取一个样本，计算这个样本损失函数的梯度并更新参数，即**随机梯度下降法**（stochastic gradient descent，SGD）。

随机梯度下降法的训练过程如算法2.1所示。

批量梯度下降和随机梯度下降之间的区别在于每次迭代的优化目标是对所有样本的平均损失函数还是单个样本的损失函数。随机梯度下降因为实现简单，收敛速度也非常快，因此使用非常广泛。还有一种折中的方法就是**小批量**（Mini-Batach）随机梯度下降法。每次迭代时，只采用一小部分的训练样本，兼顾了批量梯度下降和随机梯度下降的优点。每次迭代时，选取一个包含 m 个样本的

算法 2.1: 随机梯度下降法

输入: 训练集: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}, i = 1, \cdots, N$

验证集：$\mathcal{V}$

学习率：α

随机初始化 θ;

repeat

  对训练集 $\mathcal{D}$ 中的样本随机重排序;

  for $i = 1 \cdots N$ **do**

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

    更新参数 $\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}\left(\theta; x^{(i)}, y^{(i)}\right)}{\partial \theta}$

  end

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

输出: θ

子集 S_t，计算这个子集上每个样本损失函数的梯度并进行参数更新。

$$\theta \leftarrow \theta - \lambda \frac{1}{m} \sum_{i \in I_t} \frac{\partial \mathcal{R}\left(\theta; x^{(i)}, y^{(i)}\right)}{\partial \theta}. \tag{2.27}$$

m 通常不会设置很大，一般在 $1 \sim 100$ 之间。在实际应用中为了提高计算效率，通常设置为 2 的 n 次方。

在实际应用中，（小批量）随机梯度下降方法有收敛快，计算开销小的优点，因此逐渐成为大规模的机器学习中的主要优化算法 [Bottou, 2010]。

2.3 学习方式

到目前为止，已经有各种各样的机器学习方法。按照训练样本提供的信息以及反馈方式的不同，机器学习方法一般可以分为以下几类：

1. **监督学习**（Supervised Learning）是利用一组已知输入 x 和目标标签 y 的数据来学习模型的参数，使得模型预测的目标标签和真实标签尽可能的一致。

   根据目标标签的类型不同，有监督学习又可以分为*回归*和*分类*两类。

(a) **回归**（Regression）问题：目标标签 y 是连续值（实数或连续整数），$f(\mathbf{x}, \theta)$ 的输出也是连续值。对于所有已知或未知的 $(\mathbf{x}, y)$，使得 $f(\mathbf{x}, \theta)$ 和 y 尽可能地一致。

(b) **分类**（Classification）问题：目标标签 y 是离散的类别（符号）。在分类问题中，通过学习得到的决策函数 $f(\mathbf{x}, \theta)$ 也叫**分类器**。分类问题根据其类别数量又可分为**两类分类**（Binary Classification）和**多类分类**（Multi-class Classification）问题。

2. **无监督学习**（Unsupervised Learning）是用来学习的数据不包含目标标签，需要学习算法自动学习到一些有价值的信息。典型的无监督学习问题有**聚类**（Clustering）、**密度估计**（density estimation）等。

强化学习参见第12.1节，第174页。

3. **强化学习**（Reinforcement Learning）也叫增强学习，强调一种交互的学习方式。智能体根据环境的状态每做出一个动作，可以得到即时或延时的奖励。做出一系列的动作。智能体需要在和环境的交互中不断学习调整策略，以取得最大化的累积奖励。

	监督学习	无监督学习	强化学习
输入	带标签的样本	无标签数据	动作
反馈信息	损失	最大似然等	（延迟）奖励
目标	预测标签	发现隐藏结构	最大化累积奖励

表 2.1: 三种机器学习类型的比较

表2.1给出了三种机器学习类型的比较。有监督的学习方法需要每个样本都有标签，而无监督的学习方法则不需要标签。一般而言，一个监督学习模型需要大量的有标签数据集，而这些数据集是需要人工标注的，通常成本很高。因此，也出现了很多**弱监督学习**（weak supervised learning）和**半监督学习**（semi-supervised learning）的方法，希望从大规模的未标注数据中充分挖掘有用的信息，降低对标注样本数量的要求。强化学习和有监督学习的不同在于强化学习不需要显式地以“输入/输出对”的方式给出训练样本，是一种在线的学习机制。

2.4 示例：线性回归

我们先来介绍简单的机器学习示例：线性回归。

线性回归（linear regression）是机器学习和统计学中最基础和广泛应用的模型，是一种对自变量和因变量之间关系进行建模的回归分析。自变量数量为1时称为*简单回归*，自变量数量大于1时称为*多元回归*。

从机器学习的角度来看，自变量就是样本的特性向量$\mathbf{x} \in \mathbb{R}^d$（每一维对应一个自变量），因变量是标签$y$，这里$y \in \mathbb{R}$是连续值（实数或连续整数）。假设空间是带参数的线性函数

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b, \tag{2.28}$$

其中权重向量$\mathbf{w}$和偏置b都是可学习的参数。

线性函数$f(\mathbf{x}; \mathbf{w}, b) \in \mathbb{R}$的输出也是连续值，也称为*线性模型*。

为了简单起见，我们将公式(2.28)写为

$$f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{\mathbf{w}}^{\mathrm{T}}\hat{\mathbf{x}}, \tag{2.29}$$

其中$\hat{\mathbf{w}}$和$\hat{\mathbf{x}}$分别称为**增广权重向量**和**增广特征向量**。

$$\hat{\mathbf{x}} = \mathbf{x} \oplus 1 \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \\ 1 \end{bmatrix}, \tag{2.30}$$

$$\hat{\mathbf{w}} = \mathbf{w} \oplus b \triangleq \begin{bmatrix} w_1 \\ \vdots \\ w_k \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{w} \\ \\ b \end{bmatrix}, \tag{2.31}$$

这里，$\oplus$定义为两个向量的拼接操作。

不失一般性，在本章后面的描述中我们采用简化的表示方法，直接用$\mathbf{w}$和$\mathbf{x}$来表示增广权重向量和增广特征向量。线性回归的模型简写为$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$。

2.4.1 参数学习

最小二乘法估计

平方损失参见第2.2.2节，第26页。

由于线性回归的标签 y 和模型输出都为实数值，因此平方损失函数是一种非常合适的衡量真实标签和预测标签之间的差异。

我们可以通过经验风险最小化准则来选择合适的参数 $\mathbf{w}$。给定 N 个训练样本 $(\mathbf{x}^{(i)}, y^{(i)}), 1 \leq i \leq N$，其训练集的经验风险为

$$\mathcal{R}(\mathbf{w}) = \sum_{i=1}^{N} \mathcal{L}(y^{(i)}, f(\mathbf{x}^{(i)}, \mathbf{w})) \tag{2.32}$$

$$= \sum_{i=1}^{N} (\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} - y^{(i)})^2 \tag{2.33}$$

$$= \|X^{\mathrm{T}}\mathbf{w} - \mathbf{y}\|^2, \tag{2.34}$$

其中，$\mathbf{y} \in \mathbb{R}^N$ 是一个由目标值 $y^{(1)}, \cdots, y^{(N)}$ 组成的列向量，$X \in \mathbb{R}^{(d+1)\times N}$ 是所有输入 $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}$ 组成的矩阵

$$X = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ x_d^{(1)} & x_d^{(2)} & \cdots & x_d^{(N)} \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{2.35}$$

风险函数 $\mathcal{R}(\mathbf{y}, f(X, \mathbf{w}))$ 对 $\mathbf{w}$ 的偏导数

$$\frac{\partial \mathcal{R}\big(Y, f(X, \mathbf{w})\big)}{\partial \mathbf{w}} = \frac{\partial \|X^{\mathrm{T}}\mathbf{w} - \mathbf{y}\|^2}{\partial \mathbf{w}} \tag{2.36}$$

$$= X(X^{\mathrm{T}}\mathbf{w} - \mathbf{y}). \tag{2.37}$$

由于 $\mathcal{R}(\mathbf{w})$ 为凸函数，令 $\frac{\partial}{\partial \mathbf{w}}\mathcal{R}(\mathbf{w}) = 0$ 得到

$$\mathbf{w}^* = (XX^{\mathrm{T}})^{-1}X\mathbf{y} \tag{2.38}$$

$$= \big(\sum_{i=1}^{N} \mathbf{x}^{(i)}(\mathbf{x}^{(i)})^{\mathrm{T}}\big)^{-1}\big(\sum_{i=1}^{N} \mathbf{x}^{(i)}y^{(i)}\big). \tag{2.39}$$

这样求解线性回归参数的方法也叫**最小二乘法估计**（least square estimation）。

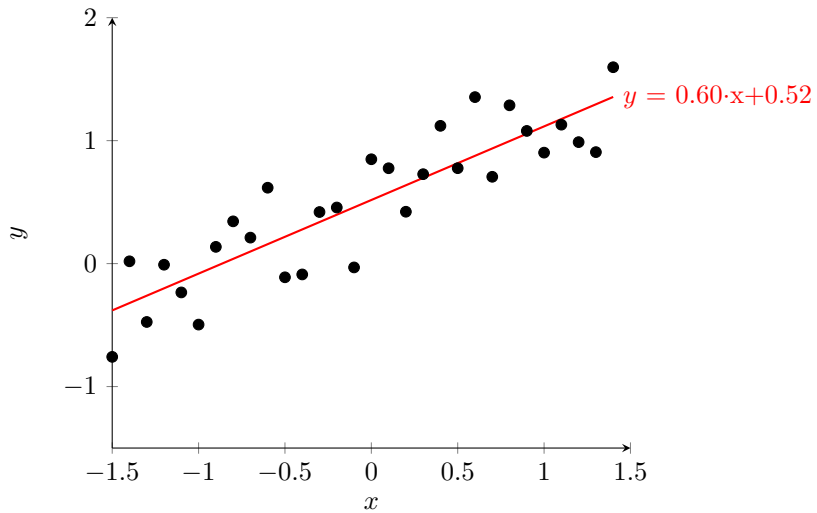

图 2.5: 线性回归示例

图2.5给出了用最小二乘法估计方法来进行参数学习的示例。最小二乘法估计要求 XX^{T} 是满秩的，存在逆矩阵，也就是要求 $\mathbf{x}$ 的每一维之间是线性不相关的。如果 XX^{T} 不可求逆矩阵，说明在训练数据上，输入的不同特征之间是线性相关的。

一种常见的 XX^{T} 不可逆情况为 $N < (d+1)$，即样本的数量小于特征维数。

因此，当 XX^{T} 不可逆时，可以使用主成分分析等方法来预处理数据，消除不同特征之间的相关性，然后再使用最小二乘估计方法来求解。或者是通过用梯度下降法来求解。初始化 $\mathbf{w}_0 = 0$，通过下面公式进行迭代，

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha X(X^{\mathrm{T}}\mathbf{w} - \mathbf{y}), \tag{2.40}$$

其中 α 是学习率。这种方法也称为**最小均方误差**（least mean squares，LMS）算法。

2.4.2 岭回归

最小二乘法估计的基本要求是各个特征之间要相互独立，但实际应用中，特征之间可能会有较大的**共线性**（multicollinearity），即 XX^{T} 的秩会解决于0，进而使得最小二乘法估计的计算变得很不稳定。数据集 X 中一些特征值小的扰动就会导致 $(XX^{\mathrm{T}})^{-1}$ 发生大的改变。为了解决这个问题，Hoerl and Kennard

[1970] 提出了**岭回归**（ridge regression），

$$\mathbf{w}^* = (XX^{\mathrm{T}} + \lambda I)^{-1} X\mathbf{y}, \tag{2.41}$$

其中 $\lambda > 0$ 为预先设置的超参数，I 为单位矩阵。

岭回归的解 $\mathbf{w}^*$ 可以看做是结构风险最小化准则下的最小二乘法估计。

$$\mathcal{R}(\mathbf{w}) = \|X^{\mathrm{T}}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2, \tag{2.42}$$

其中 $\lambda > 0$ 为正则化系数。

2.4.3 最大似然估计

机器学习任务可以分为两类，一类是样本的特征向量 $\mathbf{x}$ 和标签 y 之间如果存在未知的函数关系 $y = h(\mathbf{x})$，另一类是条件概率 $p(y|\mathbf{x})$ 服从某个未知分布。第2.4.1中介绍的最小二乘估计是属于第一类，直接建模 $\mathbf{x}$ 和标签 y 之间的函数关系。此外，线性回归还可以通过建模条件概率 $p(y|\mathbf{x})$ 的角度来进行参数估计。

这里 $\mathbf{x}$ 看作是确定值的参数。

假设标签 y 为一个随机变量，其服从以均值为 $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ 为中心，方差为 σ^2 的高斯分布。

$$p(y|\mathbf{x}, \mathbf{w}, \sigma) = \mathcal{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}, \sigma^2). \tag{2.43}$$

给定训练集 $\mathcal{D} = (\mathbf{x}^{(i)}, y^{(i)}), 1 \leq i \leq N$，其训练集上的似然函数为

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}, \sigma^2). \tag{2.44}$$

为了方便计算，对似然函数取对数得到对数似然函数

$$\mathcal{LL}(\mathbf{w}|\mathcal{D}) = \sum_{i=1}^{N} \log \mathcal{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}, \sigma^2). \tag{2.45}$$

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 $\mathcal{L}(\mathbf{w}|\mathcal{D})$ 最大，等价于对数似然函数最大。令 $\mathcal{LL}(\mathbf{w}|\mathcal{D})$ 关于 $\mathbf{w}$ 的偏导数为0，得到

参见习题（??），第??页。

$$\mathbf{w}^{ML} = (XX^{\mathrm{T}})^{-1} X\mathbf{y}, \tag{2.46}$$

其中 X 为所有样本特征向量组成的矩阵，$\mathbf{y}$ 为所有样本标签组成的向量。可以看出，最大似然估计的解和最小二乘估计的解相同。

最小二乘估计解参见公式（2.38），第34页。

2.5 偏差和方差

2.6 评价方法

为了衡量一个机器学习模型的好坏，需要给定一个测试集，用模型对测试集中的每一个样本进行预测，并根据预测结果计算评价分数。

对于分类问题，常见的评价标准有正确率、准确率、召回率和F值等。

给定测试集$\mathcal{T}=(\mathbf{x}^{(1)},y^{(1)}),\cdots,(\mathbf{x}^{(N)},y^{(N)})$，假设标签$y^{(i)}\in\{\omega_1,\cdots,\omega_C\}$，用学习好得模型$f$对测试集中的每一个样本进行预测，结果为$Y=\hat{y}^{(1)},\cdots,\hat{y}^{(N)}$。

最常用的的平均指标为**准确率**（Accuracy）

$$\mathcal{ACC}(f|\mathcal{T})=\frac{1}{N}\sum_{i=1}^{N}I(y^{(i)}=\hat{y}^{(i)}), \tag{2.47}$$

其中$I(\cdot)$为指示函数。

和准确率相对应的就是**错误率**（Error Rate）。

$$\mathcal{E}(f|\mathcal{T})=1-\mathcal{ACC}(f|\mathcal{T}) \tag{2.48}$$

$$=\frac{1}{N}\sum_{i=1}^{N}I(y^{(i)}\leq\hat{y}^{(i)}). \tag{2.49}$$

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召回率。查准率和召回率是广泛用于信息检索和统计学分类领域的两个度量值，在机器学习的评价中也被大量使用。

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

$$\mathcal{P}_c(f|\mathcal{T})=\frac{\sum_{i=1}^{N}I(y^{(i)}=\hat{y}^{(i)}=c)}{\sum_{i=1}^{N}I(y^{(i)}=c)}, \tag{2.50}$$

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

$$\mathcal{R}_c(f|\mathcal{T})=\frac{\sum_{i=1}^{N}I(y^{(i)}=\hat{y}^{(i)}=c)}{\sum_{i=1}^{N}I(\hat{y}^{(i)}=c)}, \tag{2.51}$$

F1 值（F1 value）是一个综合指标，为查准率和查全率的调和平均。

$$\mathcal{F}1_c(f|\mathcal{T}) = \frac{2\mathcal{P}_c(f|\mathcal{T})\mathcal{R}_c(f|\mathcal{T})}{\mathcal{P}_c(f|\mathcal{T}) + \mathcal{R}_c(f|\mathcal{T})} \tag{2.52}$$

在实际应用中，很多任务还有自己专门的评价方式，比如 TopN 准确率等。

2.7 理论和定理

在机器学习中，有一些非常有名的理论或定理，对理解机器学习的内在特性非常有帮助。

2.7.1 PAC 学习理论

当使用机器学习方法来解决某个特定问题时，通常靠经验或者多次试验来选择合适的模型、训练样本数量以及学习算法收敛的速度等。但是经验判断或多次试验往往成本比较高，也不太可靠，因此希望有一套理论能够分析问题难度、计算模型能力，为学习算法提供理论保证，并指导机器学习模型和学习算法的设计。这就是计算学习理论。**计算学习理论**（computational learning theory）是关于机器学习的理论基础，其中最基础的理论就是**可能近似正确**（Probably Approximately Correct，PAC）学习理论。

机器学习中一个很关键的问题是期望错误和经验错误之间的差异，称为**泛化错误**（generalization error）。泛化错误可以衡量一个机器学习模型 f 是否可以很好地泛化到未知数据。

“泛化错误”在有些文献中也指“期望错误”，指在未知样本上的错误。

$$\mathcal{G}_{\mathcal{D}}(f) = \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f). \tag{2.53}$$

根据大数定律，当训练集大小 $|\mathcal{D}|$ 趋向于无穷大时，泛化错误趋向于 0，即经验风险趋近于期望风险。

$$\lim_{|\mathcal{D}|\to\infty} \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f) = 0. \tag{2.54}$$

由于我们不知道真实的数据分布 $p(\mathbf{x}, y)$，也不知道真实的目标函数 $g(\mathbf{x})$，因此期望从有限的训练样本上学习到一个期望错误为 0 的函数 $f(\mathbf{x})$ 是不切实际

的。因此，需要降低对学习算法能力的期望，只要求学习算法可以以一定的概率学习到一个近似正确的假设，即 PAC 学习。

PAC 学习可以分为两部分：

一是“近似正确”（Approximately Correct）。一个假设 $f \in \mathcal{F}$ 是“近似正确”的，是指其在泛化错误 $\mathcal{G}_{\mathcal{D}}(f)$ 小于一个界限 ϵ。ϵ 一般为 0 到 $\frac{1}{2}$ 之间的数，$0 < \epsilon < \frac{1}{2}$。如果 $\mathcal{G}_{\mathcal{D}}(f)$ 比较大，说明模型不能用来做正确的“预测”。

二是“可能”。一个学习算法 $\mathcal{A}$ 有“可能”以 $1-\delta$ 的概率学习到这样一个“近似正确”的假设。δ 一般为 0 到 $\frac{1}{2}$ 之间的数，$0 < \delta < \frac{1}{2}$。

一个 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 $f(\mathbf{x})$。

$$P\left((\mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f)) \leq \epsilon\right) \geq 1 - \delta, \tag{2.55}$$

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

$$n(\epsilon, \delta) \geq \frac{1}{2\epsilon^2}(\ln|\mathcal{F}| + \ln\frac{2}{\delta}), \tag{2.56}$$

其中 $|\mathcal{F}|$ 为假设空间的大小。

[Blum et al., 2016] 定理 5.3。

PAC 学习理论也可以帮助分析一个机器学习方法在什么条件下可以学习到一个近似正确的分类器。从公式 (2.56) 可以看出，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。

2.7.2 没有免费午餐定理

没有免费午餐定理（no free lunch theorem，NFL）是由 Wolpert 和 Macerday 在最优化理论中提出的。没有免费午餐定理证明：对于基于迭代的最优化算法，不存在某种算法对所有问题（有限的搜索空间内）都有效。如果一个算法对某些问题有效，那么它一定在另外一些问题上比纯随机搜索算法更差。也就是说，不能脱离具体问题来谈论算法的优劣，任何算法都有局限性。必须要“具体问题具体分析”。

没有免费午餐定理对于机器学习算法也同样适用。不存在一种机器学习算法适合于任何领域或任务。如果有人宣称自己的模型在所有问题上都好于其他模型，那么他肯定是在吹牛。

2.7.3 丑小鸭定理

渡边慧（Satosi Watanabe），1910-1993，美籍日本学者，理论物理学家，也是模式识别的最早研究者之一。

这里的“丑小鸭”是指白天鹅的幼雏，而不是“丑陋的小鸭子”。

丑小鸭定理（ugly duckling theorem）是1969年由渡边慧提出的[Watanable, 1969]。“丑小鸭与白天鹅之间的区别和两只白天鹅之间的区别一样大”。这个定理初看好像不符合常识，但是仔细思考后是非常有道理的。因为世界上不存在相似性的客观标准，一切相似性的标准都是主观的。如果以体型大小的角度来看，丑小鸭和白天鹅的区别大于两只白天鹅的区别；但是如果以基因的角度来看，丑小鸭与它父母的差别要小于他父母和其他白天鹅之间的差别。

2.8 总结和深入阅读

本章简单地介绍了机器学习的理论知识，主要为后面讲解人工神经网络铺垫一些基础知识。如果需要快速全面地了解机器学习的基本概念可以阅读《Pattern Classification》[Duda et al., 2001]和《Pattern Recognition and Machine Learning》[Bishop, 2006]，进一步深入了解可以阅读《The Elements of Statistical Learning》[Hastie et al., 2001] 以及《Learning in Graphical Models》[Jordan, 1998]。

习题 2-1 证明最大似然估计中，公式(2.46)中的结论。

参考文献

C.M. Bishop. *Pattern recognition and machine learning*. Springer New York., 2006.

Avrim Blum, John Hopcroft, and Ravindran Kannan. Foundations of data science. *Vorabversion eines Lehrbuchs*, 2016.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2001. ISBN 0471056693.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

M.I. Jordan. *Learning in Graphical Models*. Kluwer Academic Publishers,

1998.

Yann LeCun, Corinna Cortes, and Christopher JC Burges. MNIST handwritten digit database. Online, 1998. URL http://yann.lecun.com/exdb/mnist.

T.M. Mitchell. *Machine learning*. Burr Ridge, IL: McGraw Hill, 1997.

S Watanable. Knowing and guessing: A quantitative study of inference and information, 1969.