第1章 绪 论

1.1 引言

傍晚小街路面上沁出微雨后的湿润, 和煦的细风吹来, 抬头看看天边的晚霞, 嗯, 明天又是一个好天气. 走到水果摊旁, 挑了个根蒂蜷缩、敲起来声音浊响的青绿西瓜, 一边满心期待着皮薄肉厚瓤甜的爽落感, 一边愉快地想着, 这学期狠下了工夫, 基础概念弄得清清楚楚, 算法作业也是信手拈来, 这门课成绩一定差不了!

希望各位在学期结束时有这样的感觉. 作为开场, 我们先大致了解一下什么是“机器学习”(machine learning).

回头看第一段话, 我们会发现这里涉及很多基于经验做出的预判. 例如, 为什么看到微湿路面、感到和风、看到晚霞, 就认为明天是好天呢? 这是因为在我们的生活经验中已经遇见过很多类似情况, 头一天观察到上述特征后, 第二天天气通常会很好. 为什么色泽青绿、根蒂蜷缩、敲声浊响, 就能判断出是正熟的好瓜? 因为我们吃过、看过很多西瓜, 所以基于色泽、根蒂、敲声这几个特征我们就可以做出相当好的判断. 类似的, 我们从以往的学习经验知道, 下足了工夫、弄清了概念、做好了作业, 自然会取得好成绩. 可以看出, 我们能做出有效的预判, 是因为我们已经积累了许多经验, 而通过对经验的利用, 就能对新情况做出有效的决策.

上面对经验的利用是靠我们人类自身完成的. 计算机能帮忙吗?

机器学习正是这样一门学科, 它致力于研究如何通过计算的手段, 利用经验来改善系统自身的性能. 在计算机系统中, “经验”通常以“数据”形式存在, 因此, 机器学习所研究的主要内容, 是关于在计算机上从数据中产生“模型”(model) 的算法, 即“学习算法”(learning algorithm). 有了学习算法, 我们把经验数据提供给它, 它就能基于这些数据产生模型; 在面对新的情况时(例如看到一个没剖开的西瓜), 模型会给我们提供相应的判断(例如好瓜). 如果说计算机科学是研究关于“算法”的学问, 那么类似的, 可以说机器学习是研究关于“学习算法”的学问.

[Mitchell, 1997] 给出了一个更形式化的定义: 假设用 P 来评估计算机程序在某任务类 T 上的性能, 若一个程序通过利用经验 E 在 T 中任务上获得了性能改善, 则我们就说关于 T 和 P, 该程序对 E 进行了学习.

本书用“模型”泛指从数据中学得的结果. 有文献用“模型”指全局性结果(例如一棵决策树), 而用“模式”指局部性结果(例如一条规则).

例如[Hand et al., 2001].

1.2 基本术语

要进行机器学习, 先要有数据. 假定我们收集了一批关于西瓜的数据, 例如(色泽=青绿; 根蒂=蜷缩; 敲声=浊响), (色泽=乌黑; 根蒂= 稍蜷; 敲声=沉闷), (色泽=浅白; 根蒂=硬挺; 敲声=清脆), ……, 每对括号内是一条记录, "=" 意思是"取值为".

这组记录的集合称为一个"数据集" (data set), 其中每条记录是关于一个事件或对象(这里是一个西瓜)的描述, 称为一个"示例" (instance) 或"样本" (sample). 反映事件或对象在某方面的表现或性质的事项, 例如"色泽""根蒂""敲声", 称为"属性" (attribute) 或"特征" (feature); 属性上的取值, 例如"青绿""乌黑", 称为"属性值" (attribute value). 属性张成的空间称为"属性空间" (attribute space)、"样本空间" (sample space)或"输入空间". 例如我们把"色泽""根蒂""敲声"作为三个坐标轴, 则它们张成一个用于描述西瓜的三维空间, 每个西瓜都可在这个空间中找到自己的坐标位置. 由于空间中的每个点对应一个坐标向量, 因此我们也把一个示例称为一个"特征向量" (feature vector).

有时整个数据集亦称一个"样本", 因为它可看作对样本空间的一个采样; 通过上下文可判断出"样本"是指单个示例还是数据集.

一般地, 令 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ 表示包含 m 个示例的数据集, 每个示例由 d 个属性描述(例如上面的西瓜数据使用了 3 个属性), 则每个示例 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; \ldots; x_{id})$ 是 d 维样本空间 $\mathcal{X}$ 中的一个向量, $\boldsymbol{x}_i \in \mathcal{X}$, 其中 x_{ij} 是 $\boldsymbol{x}_i$ 在第 j 个属性上的取值(例如上述第 3 个西瓜在第 2 个属性上的值是"硬挺"), d 称为样本 $\boldsymbol{x}_i$ 的"维数" (dimensionality).

从数据中学得模型的过程称为"学习" (learning)或"训练" (training), 这个过程通过执行某个学习算法来完成. 训练过程中使用的数据称为"训练数据" (training data), 其中每个样本称为一个"训练样本" (training sample), 训练样本组成的集合称为"训练集" (training set). 学得模型对应了关于数据的某种潜在的规律, 因此亦称"假设" (hypothesis); 这种潜在规律自身, 则称为"真相"或"真实" (ground-truth), 学习过程就是为了找出或逼近真相. 本书有时将模型称为"学习器" (learner), 可看作学习算法在给定数据和参数空间上的实例化.

训练样本亦称"训练示例" (training instance) 或"训练例".

学习算法通常有参数需设置, 使用不同的参数值和(或)训练数据, 将产生不同的结果.

如果希望学得一个能帮助我们判断没剖开的是不是"好瓜"的模型, 仅有前面的示例数据显然是不够的. 要建立这样的关于"预测" (prediction) 的模型, 我们需获得训练样本的"结果"信息, 例如"((色泽=青绿; 根蒂=蜷缩; 敲声=浊响), 好瓜)". 这里关于示例结果的信息, 例如"好瓜", 称为"标记" (label); 拥有了标记信息的示例, 则称为"样例" (example). 一般地, 用

将"label"译为"标记"而非"标签", 是考虑到英文中"label"既可用作名词、也可用作动词.

若将标记看作对象本身的一部分, 则“样例”有时也称为“样本”.

$(\boldsymbol{x}_i, y_i)$ 表示第 i 个样例, 其中 $y_i \in \mathcal{Y}$ 是示例 $\boldsymbol{x}_i$ 的标记, $\mathcal{Y}$ 是所有标记的集合, 亦称“标记空间”(label space)或“输出空间”.

若我们欲预测的是离散值, 例如“好瓜”“坏瓜”, 此类学习任务称为“分类”(classification); 若欲预测的是连续值, 例如西瓜成熟度 0.95、0.37, 此类学习任务称为“回归”(regression). 对只涉及两个类别的“二分类”(binary classification)任务, 通常称其中一个类为“正类”(positive class), 另一个类为“反类”(negative class); 涉及多个类别时, 则称为“多分类”(multi-class classification) 任务. 一般地, 预测任务是希望通过对训练集 $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$ 进行学习, 建立一个从输入空间 $\mathcal{X}$ 到输出空间 $\mathcal{Y}$ 的映射 $f : \mathcal{X} \mapsto \mathcal{Y}$. 对二分类任务, 通常令 $\mathcal{Y} = \{-1, +1\}$ 或 $\{0, 1\}$; 对多分类任务, $|\mathcal{Y}| > 2$; 对回归任务, $\mathcal{Y} = \mathbb{R}$, $\mathbb{R}$ 为实数集.

亦称“负类”.

学得模型后, 使用其进行预测的过程称为“测试”(testing), 被预测的样本称为“测试样本”(testing sample). 例如在学得 f 后, 对测试例 $\boldsymbol{x}$, 可得到其预测标记 $y = f(\boldsymbol{x})$.

亦称“测试示例”(testing instance) 或“测试例”.

我们还可以对西瓜做“聚类”(clustering), 即将训练集中的西瓜分成若干组, 每组称为一个“簇”(cluster); 这些自动形成的簇可能对应一些潜在的概念划分, 例如“浅色瓜”“深色瓜”, 甚至“本地瓜”“外地瓜”. 这样的学习过程有助于我们了解数据内在的规律, 能为更深入地分析数据建立基础. 需说明的是, 在聚类学习中, “浅色瓜”“本地瓜”这样的概念我们事先是不知道的, 而且学习过程中使用的训练样本通常不拥有标记信息.

否则标记信息直接形成了簇划分; 但也有例外情况, 参见 13.6 节.

根据训练数据是否拥有标记信息, 学习任务可大致划分为两大类: “监督学习”(supervised learning) 和“无监督学习”(unsupervised learning), 分类和回归是前者的代表, 而聚类则是后者的代表.

亦称“有导师学习”和“无导师学习”.

需注意的是, 机器学习的目标是使学得的模型能很好地适用于“新样本”, 而不是仅仅在训练样本上工作得很好; 即便对聚类这样的无监督学习任务, 我们也希望学得的簇划分能适用于没在训练集中出现的样本. 学得模型适用于新样本的能力, 称为“泛化”(generalization)能力. 具有强泛化能力的模型能很好地适用于整个样本空间. 于是, 尽管训练集通常只是样本空间的一个很小的采样, 我们仍希望它能很好地反映出样本空间的特性, 否则就很难期望在训练集上学得的模型能在整个样本空间上都工作得很好. 通常假设样本空间中全体样本服从一个未知“分布”(distribution) $\mathcal{D}$, 我们获得的每个样本都是独立地从这个分布上采样获得的, 即“独立同分布”(independent and identically distributed, 简称 *i.i.d.*). 一般而言, 训练样本越多, 我们得到的关于 $\mathcal{D}$ 的信息

更确切地说, 是“未见示例”(unseen instance).

现实任务中样本空间的规模通常很大(例如 20 个属性, 每个属性有 10 个可能取值, 则样本空间的规模已达 10^{20}).

越多, 这样就越有可能通过学习获得具有强泛化能力的模型.

1.3 假设空间

归纳(induction)与演绎(deduction)是科学推理的两大基本手段. 前者是从特殊到一般的“泛化”(generalization)过程, 即从具体的事实归结出一般性规律; 后者则是从一般到特殊的“特化”(specialization)过程, 即从基础原理推演出具体状况. 例如, 在数学公理系统中, 基于一组公理和推理规则推导出与之相洽的定理, 这是演绎; 而“从样例中学习”显然是一个归纳的过程, 因此亦称“归纳学习”(inductive learning).

归纳学习有狭义与广义之分, 广义的归纳学习大体相当于从样例中学习, 而狭义的归纳学习则要求从训练数据中学得概念(concept), 因此亦称为“概念学习”或“概念形成”. 概念学习技术目前研究、应用都比较少, 因为要学得泛化性能好且语义明确的概念实在太困难了, 现实常用的技术大多是产生“黑箱”模型. 然而, 对概念学习有所了解, 有助于理解机器学习的一些基础思想.

概念学习中最基本的是布尔概念学习, 即对“是”“不是”这样的可表示为 0/1 布尔值的目标概念的学习. 举一个简单的例子, 假定我们获得了这样一个训练数据集:

表 1.1 西瓜数据集

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

这里要学习的目标是“好瓜”. 暂且假设“好瓜”可由“色泽”“根蒂”“敲声”这三个因素完全确定, 换言之, 只要某个瓜的这三个属性取值明确了, 我们就能判断出它是不是好瓜. 于是, 我们学得的将是“好瓜是某种色泽、某种根蒂、某种敲声的瓜”这样的概念, 用布尔表达式写出来则是“好瓜 $\leftrightarrow$ (色泽=?) $\wedge$ (根蒂=?) $\wedge$ (敲声=?)”, 这里“?”表示尚未确定的取值, 而我们的任务就是通过对表 1.1 的训练集进行学习, 把“?”确定下来.

更一般的情况是考虑形如 $(A \wedge B) \vee (C \wedge D)$ 的析合范式.

读者可能马上发现, 表 1.1 第一行: “(色泽=青绿) $\wedge$ (根蒂=蜷缩) $\wedge$ (敲声=浊响)”不就是好瓜吗? 是的, 但这是一个已见过的瓜, 别忘了我们学习的目的是“泛化”, 即通过对训练集中瓜的学习以获得对没见过的瓜进行判断的

“记住”训练样本, 就是所谓的“机械学习” [Cohen and Feigenbaum, 1983], 或称“死记硬背式学习”, 参见 1.5 节.

能力. 如果仅仅把训练集中的瓜“记住”, 今后再见到一模一样的瓜当然可判断, 但是, 对没见过的瓜, 例如“(色泽=浅白) ∧ (根蒂=蜷缩) ∧ (敲声= 浊响)”怎么办呢?

我们可以把学习过程看作一个在所有假设(hypothesis)组成的空间中进行搜索的过程, 搜索目标是找到与训练集“匹配”(fit)的假设, 即能够将训练集中的瓜判断正确的假设. 假设的表示一旦确定, 假设空间及其规模大小就确定了. 这里我们的假设空间由形如“(色泽=?) ∧ (根蒂=?) ∧ (敲声=?)”的可能取值所形成的假设组成. 例如色泽有“青绿”“乌黑”“浅白”这三种可能取值; 还需考虑到, 也许“色泽”无论取什么值都合适, 我们用通配符“$*$”来表示, 例如“好瓜 $\leftrightarrow$ (色泽= $*$) ∧ (根蒂=蜷缩) ∧ (敲声=浊响)”, 即“好瓜是根蒂蜷缩、敲声浊响的瓜, 什么色泽都行”. 此外, 还需考虑极端情况: 有可能“好瓜”这个概念根本就不成立, 世界上没有“好瓜”这种东西; 我们用 $\varnothing$ 表示这个假设. 这样, 若“色泽”“根蒂”“敲声”分别有3、2、2 种可能取值, 则我们面临的假设空间规模大小为 $4 \times 3 \times 3 + 1 = 37$. 图 1.1 直观地显示出了这个西瓜问题假设空间.

这里我们假定训练样本不含噪声, 并且不考虑“非青绿”这样的 $\neg A$ 操作. 由于训练集包含正例, 因此 $\varnothing$ 假设自然不出现.

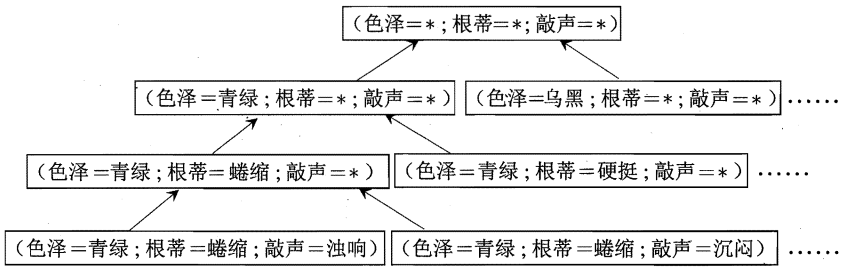


图 1.1 西瓜问题的假设空间

可以有许多策略对这个假设空间进行搜索, 例如自顶向下、从一般到特殊, 或是自底向上、从特殊到一般, 搜索过程中可以不断删除与正例不一致的假设、和(或)与反例一致的假设. 最终将会获得与训练集一致(即对所有训练样本能够进行正确判断)的假设, 这就是我们学得的结果.

有许多可能的选择, 如在路径上自顶向下与自底向上同时进行, 在操作上只删除与正例不一致的假设等.

需注意的是, 现实问题中我们常面临很大的假设空间, 但学习过程是基于有限样本训练集进行的, 因此, 可能有多个假设与训练集一致, 即存在着一个与训练集一致的“假设集合”, 我们称之为“版本空间”(version space). 例如, 在西瓜问题中, 与表 1.1 训练集所对应的版本空间如图 1.2 所示.

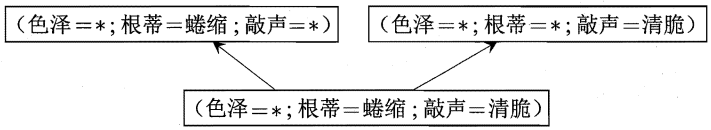


图 1.2 西瓜问题的版本空间

1.4 归纳偏好

通过学习得到的模型对应了假设空间中的一个假设. 于是, 图 1.2 的西瓜版本空间给我们带来一个麻烦: 现在有三个与训练集一致的假设, 但与它们对应的模型在面临新样本的时候, 却会产生不同的输出. 例如, 对(色泽=青绿; 根蒂= 蜷缩; 敲声= 沉闷)这个新收来的瓜, 如果我们采用的是“好瓜 $\leftrightarrow$ (色泽= $*$) $\wedge$ (根蒂=蜷缩) $\wedge$ (敲声= $*$)”, 那么将会把新瓜判断为好瓜, 而如果采用了另外两个假设, 则判断的结果将不是好瓜. 那么, 应该采用哪一个模型(或假设)呢?

若仅有表 1.1 中的训练样本, 则无法断定上述三个假设中哪一个“更好”. 然而, 对于一个具体的学习算法而言, 它必须要产生一个模型. 这时, 学习算法本身的“偏好”就会起到关键的作用. 例如, 若我们的算法喜欢“尽可能特殊”的模型, 则它会选择“好瓜 $\leftrightarrow$ (色泽= $*$) $\wedge$ (根蒂=蜷缩) $\wedge$(敲声=浊响)”; 但若我们的算法喜欢“尽可能一般”的模型, 并且由于某种原因它更“相信”根蒂, 则它会选择“好瓜 $\leftrightarrow$ (色泽= $*$) $\wedge$ (根蒂=蜷缩) $\wedge$(敲声= $*$)”. 机器学习算法在学习过程中对某种类型假设的偏好, 称为“归纳偏好”(inductive bias), 或简称为“偏好”.

尽可能特殊即“适用情形尽可能少”; 尽可能一般即“适用情形尽可能多”.

对“根蒂”还是对“敲声”更重视, 看起来和属性选择, 亦称“特征选择”(feature selection) 有关, 但需注意的是, 机器学习中的特征选择仍是基于对训练样本的分析进行的, 而在此处我们并非基于特征选择做出对“根蒂”的重视; 这里对“根蒂”的信赖可视为基于某种领域知识而产生的归纳偏好. 关于特征选择方面的内容参见第 11 章.

任何一个有效的机器学习算法必有其归纳偏好, 否则它将被假设空间中看似在训练集上“等效”的假设所迷惑, 而无法产生确定的学习结果. 可以想象, 如果没有偏好, 我们的西瓜学习算法产生的模型每次在进行预测时随机抽选训练集上的等效假设, 那么对这个新瓜“(色泽= 青绿; 根蒂=蜷缩; 敲声=沉闷)”, 学得模型时而告诉我们它是好的、时而告诉我们它是不好的, 这样的学习结果显然没有意义.

归纳偏好的作用在图 1.3 这个回归学习图示中可能更直观. 这里的每个训练样本是图中的一个点 (x, y), 要学得一个与训练集一致的模型, 相当于找到一条穿过所有训练样本点的曲线. 显然, 对有限个样本点组成的训练集, 存在着很多条曲线与其一致. 我们的学习算法必须有某种偏好, 才能产出它认为“正确”的模型. 例如, 若认为相似的样本应有相似的输出(例如, 在各种属性上都

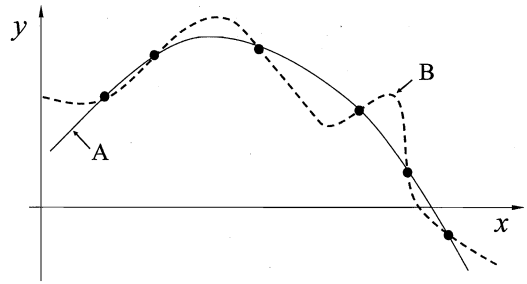


图 1.3 存在多条曲线与有限样本训练集一致

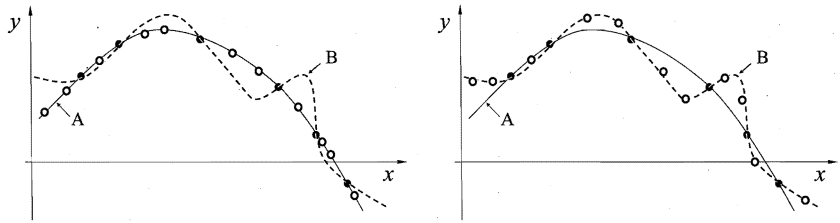
很相像的西瓜, 成熟程度应该比较接近), 则对应的学习算法可能偏好图 1.3 中比较“平滑”的曲线 A 而不是比较“崎岖”的曲线 B.

归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的启发式或“价值观”. 那么, 有没有一般性的原则来引导算法确立“正确的”偏好呢? “奥卡姆剃刀”(Occam's razor)是一种常用的、自然科学研究中最基本的原则, 即“若有多个假设与观察一致, 则选最简单的那个”. 如果采用这个原则, 并且假设我们认为“更平滑”意味着“更简单”(例如曲线 A 更易于描述, 其方程式是 $y = -x^2 + 6x + 1$, 而曲线 B 则要复杂得多), 则在图 1.3 中我们会自然地偏好“平滑”的曲线 A.

然而, 奥卡姆剃刀并非唯一可行的原则. 退一步说, 即便假定我们是奥卡姆剃刀的铁杆拥趸, 也需注意到, 奥卡姆剃刀本身存在不同的诠释, 使用奥卡姆剃刀原则并不平凡. 例如对我们已经很熟悉的西瓜问题来说, “假设 1: 好瓜 $\leftrightarrow$ (色泽$=*$) $\wedge$ (根蒂$=$ 蜷缩) $\wedge$ (敲声$=$浊响)”和假设 2: “好瓜 $\leftrightarrow$ (色泽$=*$) $\wedge$ (根蒂$=$蜷缩) $\wedge$ (敲声$=*$)”这两个假设, 哪一个更“简单”呢? 这个问题并不简单, 需借助其他机制才能解决.

事实上, 归纳偏好对应了学习算法本身所做出的关于“什么样的模型更好”的假设. 在具体的现实问题中, 这个假设是否成立, 即算法的归纳偏好是否与问题本身匹配, 大多数时候直接决定了算法能否取得好的性能.

让我们再回头看看图 1.3. 假设学习算法 $\mathfrak{L}_a$ 基于某种归纳偏好产生了对应于曲线 A 的模型, 学习算法 $\mathfrak{L}_b$ 基于另一种归纳偏好产生了对应于曲线 B 的模型. 基于前面讨论的平滑曲线的某种“描述简单性”, 我们满怀信心地期待算法 $\mathfrak{L}_a$ 比 $\mathfrak{L}_b$ 更好. 确实, 图 1.4(a)显示出, 与 B 相比, A 与训练集外的样本更一致; 换言之, A 的泛化能力比 B 强.



(a) A 优于 B (b) B 优于 A

图 1.4 没有免费的午餐. (黑点: 训练样本; 白点: 测试样本)

但是, 且慢! 虽然我们希望并相信 $\mathfrak{L}_a$ 比 $\mathfrak{L}_b$ 更好, 但会不会出现图 1.4(b)的情况: 与 A 相比, B 与训练集外的样本更一致?

很遗憾, 这种情况完全可能出现. 换言之, 对于一个学习算法 $\mathfrak{L}_a$, 若它在某些问题上比学习算法 $\mathfrak{L}_b$ 好, 则必然存在另一些问题, 在那里 $\mathfrak{L}_b$ 比 $\mathfrak{L}_a$ 好. 有趣的是, 这个结论对任何算法均成立, 哪怕是把本书后面将要介绍的一些聪明算法作为 $\mathfrak{L}_a$ 而将“随机胡猜”这样的笨拙算法作为 $\mathfrak{L}_b$. 惊讶吗? 让我们看看下面这个简短的讨论:

这里只用到一些非常基础的数学知识, 只准备读第 1 章且有“数学恐惧”的读者可以跳过这个部分而不会影响理解, 只需相信, 上面这个看起来“匪夷所思”的结论确实是成立的.

为简单起见, 假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 都是离散的. 令 $P(h|X,\mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 X 产生假设 h 的概率, 再令 f 代表我们希望学习的真实目标函数. $\mathfrak{L}_a$ 的“训练集外误差”, 即 $\mathfrak{L}_a$ 在训练集之外的所有样本上的误差为

$$E_{ote}(\mathfrak{L}_a|X,f)=\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\,\mathbb{I}\left(h\left(\boldsymbol{x}\right)\neq f\left(\boldsymbol{x}\right)\right)P\left(h\mid X,\mathfrak{L}_a\right)\ , \tag{1.1}$$

其中 $\mathbb{I}(\cdot)$ 是指示函数, 若 $\cdot$ 为真则取值 1, 否则取值 0.

考虑二分类问题, 且真实目标函数可以是任何函数 $\mathcal{X}\mapsto\{0,1\}$, 函数空间为 $\{0,1\}^{|\mathcal{X}|}$. 对所有可能的 f 按均匀分布对误差求和, 有

$$\begin{aligned}\sum_{f}E_{ote}(\mathfrak{L}_a|X,f)&=\sum_{f}\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\,\mathbb{I}(h(\boldsymbol{x})\neq f(\boldsymbol{x}))\,P(h\mid X,\mathfrak{L}_a)\\&=\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\sum_{h}P(h\mid X,\mathfrak{L}_a)\sum_{f}\mathbb{I}(h(\boldsymbol{x})\neq f(\boldsymbol{x}))\\&=\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\sum_{h}P(h\mid X,\mathfrak{L}_a)\frac{1}{2}2^{|\mathcal{X}|}\\&=\frac{1}{2}2^{|\mathcal{X}|}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\sum_{h}P(h\mid X,\mathfrak{L}_a)\end{aligned}$$

若 f 均匀分布, 则有一半的 f 对 $\boldsymbol{x}$ 的预测与 $h(\boldsymbol{x})$ 不一致.

$$= 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x} \in \mathcal{X}-X} P(\boldsymbol{x}) \cdot 1 \ . \tag{1.2}$$

式(1.2)显示出, 总误差竟然与学习算法无关! 对于任意两个学习算法 $\mathfrak{L}_a$ 和 $\mathfrak{L}_b$, 我们都有

$$\sum_f E_{ote}(\mathfrak{L}_a|X,f) = \sum_f E_{ote}(\mathfrak{L}_b|X,f) \ , \tag{1.3}$$

也就是说, 无论学习算法 $\mathfrak{L}_a$ 多聪明、学习算法 $\mathfrak{L}_b$ 多笨拙, 它们的期望性能竟然相同! 这就是“没有免费的午餐”定理 (No Free Lunch Theorem, 简称 NFL 定理) [Wolpert, 1996; Wolpert and Macready, 1995].

严格的 NFL 定理证明比这里的简化论述繁难得多.

这下子, 读者对机器学习的热情可能被一盆冷水浇透了: 既然所有学习算法的期望性能都跟随机胡猜差不多, 那还有什么好学的?

我们需注意到, NFL 定理有一个重要前提: 所有“问题”出现的机会相同、或所有问题同等重要. 但实际情形并不是这样. 很多时候, 我们只关注自己正在试图解决的问题(例如某个具体应用任务), 希望为它找到一个解决方案, 至于这个解决方案在别的问题、甚至在相似的问题上是否为好方案, 我们并不关心. 例如, 为了快速从 A 地到达 B 地, 如果我们正在考虑的 A 地是南京鼓楼、B 地是南京新街口, 那么“骑自行车”是很好的解决方案; 这个方案对 A 地是南京鼓楼、B 地是北京新街口的情形显然很糟糕, 但我们对此并不关心.

事实上, 上面 NFL 定理的简短论述过程中假设了 f 的均匀分布, 而实际情形并非如此. 例如, 回到我们熟悉的西瓜问题, 考虑 {假设 1: 好瓜 $\leftrightarrow$ (色泽= $*$) $\wedge$ (根蒂=蜷缩) $\wedge$ (敲声=浊响)} 和 {假设 2: 好瓜 $\leftrightarrow$ (色泽= $*$) $\wedge$ (根蒂=硬挺) $\wedge$ (敲声=清脆)}. 从 NFL 定理可知, 这两个假设同样好. 我们立即会想到符合条件的例子, 对好瓜(色泽=青绿; 根蒂=蜷缩; 敲声=浊响)是假设 1 更好, 而对好瓜(色泽=乌黑; 根蒂= 硬挺; 敲声=清脆)则是假设 2 更好. 看上去的确是这样. 然而需注意到, “(根蒂= 蜷缩; 敲声= 浊响)”的好瓜很常见, 而“(根蒂=硬挺; 敲声=清脆)”的好瓜罕见, 甚至不存在.

所以, NFL 定理最重要的寓意, 是让我们清楚地认识到, 脱离具体问题, 空泛地谈论“什么学习算法更好”毫无意义, 因为若考虑所有潜在的问题, 则所有学习算法都一样好. 要谈论算法的相对优劣, 必须要针对具体的学习问题; 在某些问题上表现好的学习算法, 在另一些问题上却可能不尽如人意, 学习算法自身的归纳偏好与问题是否相配, 往往会起到决定性的作用.

1.5 发展历程

机器学习是人工智能(artificial intelligence)研究发展到一定阶段的必然产物. 二十世纪五十年代到七十年代初, 人工智能研究处于“推理期”, 那时人们以为只要能赋予机器逻辑推理能力, 机器就能具有智能. 这一阶段的代表性工作主要有 A. Newell 和 H. Simon 的“逻辑理论家”(Logic Theorist)程序以及此后的“通用问题求解”(General Problem Solving)程序等, 这些工作在当时取得了令人振奋的结果. 例如, “逻辑理论家”程序在 1952 年证明了著名数学家罗素和怀特海的名著《数学原理》中的 38 条定理, 在 1963 年证明了全部 52 条定理, 特别值得一提的是, 定理 2.85 甚至比罗素和怀特海证明得更巧妙. A. Newell 和 H. Simon 因为这方面的工作获得了 1975 年图灵奖. 然而, 随着研究向前发展, 人们逐渐认识到, 仅具有逻辑推理能力是远远实现不了人工智能的. E. A. Feigenbaum 等人认为, 要使机器具有智能, 就必须设法使机器拥有知识. 在他们的倡导下, 从二十世纪七十年代中期开始, 人工智能研究进入了“知识期”. 在这一时期, 大量专家系统问世, 在很多应用领域取得了大量成果. E. A. Feigenbaum 作为“知识工程”之父在 1994 年获得图灵奖. 但是, 人们逐渐认识到, 专家系统面临“知识工程瓶颈”, 简单地说, 就是由人来把知识总结出来再教给计算机是相当困难的. 于是, 一些学者想到, 如果机器自己能够学习知识该多好!

所谓“知识就是力量”.

1965 年, Feigenbaum 主持研制了世界上第一个专家系统 DENDRAL.

事实上, 图灵在 1950 年关于图灵测试的文章中, 就曾提到了机器学习的可能; 二十世纪五十年代初已有机器学习的相关研究, 例如 A. Samuel 著名的跳棋程序. 五十年代中后期, 基于神经网络的“连接主义”(connectionism)学习开始出现, 代表性工作有 F. Rosenblatt 的感知机(Perceptron)、B. Widrow 的 Adaline 等. 在六七十年代, 基于逻辑表示的“符号主义”(symbolism)学习技术蓬勃发展, 代表性工作有 P. Winston 的“结构学习系统”、R. S. Michalski 等人的“基于逻辑的归纳学习系统”、E. B. Hunt 等人的“概念学习系统”等; 以决策理论为基础的学习技术以及强化学习技术等也得到发展, 代表性工作有 N. J. Nilson 的“学习机器”等; 二十多年后红极一时的统计学习理论的一些奠基性结果也是在这个时期取得的.

参见 p.22.

1980 年夏, 在美国卡耐基梅隆大学举行了第一届机器学习研讨会(IWML); 同年, 《策略分析与信息系统》连出三期机器学习专辑; 1983 年, Tioga 出版社出版了 R. S. Michalski、J. G. Carbonell 和 T. Mitchell 主编的《机器学习: 一种人工智能途径》[Michalski et al., 1983], 对当时的机器学习研究工作进行了总结; 1986 年, 第一本机器学习专业期刊 *Machine Learning* 创刊; 1989 年, 人

IWML 后来发展为国际机器学习会议 ICML.