第3章　PAC模型

引　　言

PAC(probably approximate correct)模型是由 Valiant 于 1984 年首先提出来的，是由统计模式识别、决策理论提出了一些简单的概念并结合了计算复杂理论的方法而提出的学习模型。它是研究学习及泛化问题的一个概率框架，不仅可用于神经网络分类问题，而且可广泛用于人工智能中的学习问题。PAC 模型的作用相当于提供了一套严格的形式化语言来陈述以及刻画所提及的可学习以及样本复杂度问题。在 PAC 框架下，学习器必须从某一特定类可能的函数中选择一个泛化函数(称为假设)。我们的目标是，以很高的概率，使所选择的函数具有低泛化误差。PAC 框架的一项重要创新是机器学习计算复杂性理论概念的引入，学习器预期找到更有效的函数。本章将主要介绍基本 PAC 模型，并进一步讨论在有限空间和无限空间下样本复杂度问题。本章中的讨论将限制在学习布尔值概念，且训练数据是无噪声的，许多结论可扩展到更一般的情形。

3.1　基本的 PAC 模型

3.1.1　PAC 简介

PAC 主要研究的内容包括：一个问题什么时候是可被学习的、样本复杂度、计算复杂度，以及针对具体可学习问题的学习算法。虽然也可以扩展用于描述回归以及多分类等问题，不过最初 PAC 模型是针对二分类问题提出的，和以前的设定类似，有一个输入空间 X，也称作实例空间。X 上的一个概念 c 是 X 的一个子集，或者简单来说，c 是从 X 到 $\{0,1\}$ 的函数。这里也采用这种模型，先介绍一下这种情况下的一些特有的概念。

3.1.2　基本概念

实例空间是指学习器能见到的所有实例，每个 $x \in X$ 为一个实例，$X=U_n \geqslant 1, X_n$ 为实例空间。概念空间是指目标概念可以从中选取的所有概念的集合，学习器的目标就是要产生目标概念的一个假设 h，使其能准确地分类每个实例，对每个 $n \geqslant 1$，定义每个 $C_n \subseteq 2^{x_n}$ 为 X_n 上的一系列概念，$C=U_n \geqslant 1, C_n$ 为 X 上的概念空间，也称为概念类。假设空间是指算法所能输出的所有假设 h 的集合，用 H 表示。对每个目标概念 $c \in C_n$ 和实例 $x \in X_n$，$c(x)$ 为

实例 x 上的分类值，即 $c(x)=1$ 当且仅当 $x\in C$。C_n 的任一假设 h 指的是一个规则，即对给出的 $x\in X_n$，算法在多项式时间内为 $c(x)$ 输出一个预测值。变型空间是指能正确分类训练样例 D 的所有假设的集合，$\mathrm{VS}=\{h\in H\mid \forall <x,c(X)>\in D(h(X)=c(X))\}$。变型空间的重要意义是每个一致学习器都输出一个属于变型空间的假设。样本复杂度(sample complexity)是指学习器收敛到成功假设时至少所需的训练样本数。计算复杂度(computational complexity)是指学习器收敛到成功假设时所需的计算量。出错界限是指在成功收敛到一个假设前，学习器对训练样本的错误分类的次数。在某一特定的假设空间中，对于给定的样本，若能找到一个假设 h，使得对该概念类的任何概念都一致，且该算法的样本复杂度仍为多项式，则该算法为一致算法。

3.1.3 问题框架

实例空间为 $X=\{0,1\}^n$，概念空间和假设空间均为 $\{0,1\}^n$ 的子集，对任意给定的准确度 $\varepsilon(0<\varepsilon<1/2)$及任意给定的置信度 $\delta(0<\delta<1)$，实例空间上的所有分布 D 及目标空间中的所有目标函数 t，若学习器 L 只需多项式 $P(n,1/\varepsilon,1/\delta)$个样本及在多项式 $P(n,1/\varepsilon,1/\delta)$ 时间内，最终将以至少 $1-\delta$ 的概率输出一个假设 $h\in H$，使得随机样本被错分类的概率 $\mathrm{error}_D(h,t)=P_r[\{x\in X;h(x)\neq t(x)\}]\leqslant\varepsilon$，则称学习器 L 是 PAC 可学习的，它是考虑样本复杂度及计算复杂度的一个基本框架，成功的学习被定义为形式化的概率理论。

假设 h 是另一个 X 上的二值函数，我们试图用 h 逼近 c，选择 X 上的一个概率分布 μ，则根据关于误差(风险)的定义，有 $\varepsilon(h)=\mu(h(X)\neq c(X))$，而这个量可以很容易并且很直观地用集合的对称差表示，如图 3-1 所示，误差很直观地用两个集合的对称差(阴影部分)的面积表示。

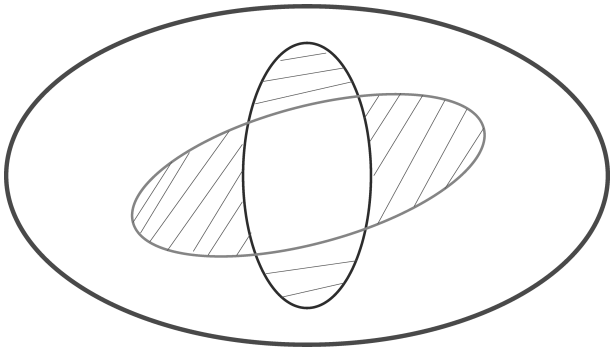

图 3-1 误差风险示意图(见彩图)

X 上的一个概念类 C 就是一堆这样的概念的集合。这里的 C 对应之前设定中的函数空间 F。类似地，学习问题实际上就是给定一个目标概念 $c\in C$，寻找一个逼近 $h\in C$ 的问题。*PAC* 模型与分布是无关的，因为对学习器来说，实例上的分布是未知的。该定义不要求学习器输出零错误率的假设，只要求其错误率限定在某常数 ε 的范围内(ε 可以任意小)；同时也不要求学习器对所有的随机抽取样本序列都能成功，只要其失败的概率限定在某个常数 δ 的范围内(δ 也可取任意小) 即可，这样将学习到一个可能近似正确的假设。

3.2　PAC 模型样本复杂度分析

3.2.1　有限空间样本复杂度

3.1 节的定义要求学习算法的运行时间在多项式时间内，且能用合理的样本数产生对目标概念的较好逼近。该模型是最坏情况模型，因为它要求在实例空间上对所有的目标概念及所有的分布 D、它所需的样本数都以某一多项式为界。

PAC 可学习性很大程度上由所需的训练样例数确定。当假设空间增大时，找到一个一致的假设将更容易，但需更多的样本来保证该假设有较高的概率是准确的。因此，在计算复杂度和样本复杂度之间存在一个折中。下面将以布尔文字的合取是 PAC 学习的为例，说明如何分析一个概念类是 PAC 学习的，并得到一致算法的样本复杂度的下界。

设学习器 L，其假设空间与概念空间相同，即 $H=C$，因假设空间为 n 个布尔文字的合取，而每个文字有 3 种可能：该变量作为文字包含在假设中；该变量的否定作为文字包含在假设中；假设中不包含该变量。所以，假设空间的大小为 $|H|=3^n$，可设计如下算法。

(1) 初始化假设 h 为 $2n$ 个文字的合取，即 $h=x_1\bar{x}_1x_2\bar{x}_2\cdots x_n\bar{x}_n$。

(2) 由样本发生器产生 $m=1/2(n\ln3+\ln1/\delta)$个样本，对每个正例，若 $x_i=0$，则从 h 中删去 x_i；若 $x_i=1$，则从 h 中删去$\bar{x}_i$。

(3) 输出保留下来的假设 h。

为分析该算法，需考虑 3 点：需要的样本数是否为多项式的；算法运行的时间是否为多项式的，即这两者是否均为 $p(n,1/\varepsilon,1/\delta)$；输出的假设是否满足 PAC 模型的标准，即 $P_r[\text{error}_D(h)\leqslant\varepsilon]\geqslant(1-\delta)$，$P_r[]$表示概率。针对本算法，由于样本数已知，显然它是多项式的；因运行每个样本的时间为一常量，而样本数又是多项式的，则算法的运行时间也是多项式的；因此只看它是否满足 PAC 模型的标准即可。若 h'满足 $\text{error}_D(h')>\varepsilon$，则称为 ε-bad 假设，否则称为 ε-exhausted 假设。若最终输出的假设不是 ε-bad 假设，则该假设必满足 PAC 模型的标准。

根据排除法计算学习一个假设所需要的样本个数，这里 ε-bad 假设混在 ε-exhausted 假设之中，我们试图排除这些假设来计算样本个数。根据 ε-bad 假设的定义，有：P_r[ε-bad 假设与一个样本一致]$\leqslant(1-\varepsilon)$，因每个样本独立抽取，所以 P_r[ε-bad 假设与 m 个样本一致]$\leqslant(1-\varepsilon)^m$。又因最大的假设数为 $|H|$，所以 P_r[存在 ε-bad 假设与 m 个样本一致]$\leqslant|H|(1-\varepsilon)^m$。又因要求 $P_r[h$ 是ε- bad 假设$]\leqslant\delta$，所以有

$$|H|(1-\varepsilon)^m\leqslant\delta \tag{3-1}$$

解得

$$m\geqslant\frac{\ln|H|+\ln1/\delta}{-\ln(1-\varepsilon)} \tag{3-2}$$

根据泰勒展开式：$e^x=1+x+\frac{x^2}{2}+\cdots>1+x$，将 $x=-\varepsilon$ 代入泰勒展开式中，得 $\varepsilon<-\ln(1-\varepsilon)$。将其代入式(3-1)中，得

$$m > \frac{1}{\varepsilon}\left(\ln | H | + \ln \frac{1}{\delta}\right) \tag{3-3}$$

式(3-3)提供了训练样例数目的一般理论边界，该数目的样例足以在所期望的值 δ 和 ε 程度下，使任何一致学习器成功地学习到 H 中的任意目标概念。其物理含义表示：训练样例数目 m 足以保证任意一致假设是可能(可能性为 $1-\delta$)近似(错误率为 ε)正确的，m 随着 $1/\varepsilon$ 的增大呈线性增长，随着 $1/\delta$ 和假设空间规模的增大呈对数增长。

针对本例有 $|H|=3^n$，将它代入式(3-2)中得到，当样本数 $m>\frac{1}{\varepsilon}\left(n\ln 3+ln\ \frac{1}{\delta}\right)$ 时，有 $P_r[\text{error}_D(h)>\varepsilon]\leqslant\delta$ 成立。同时也证明了布尔文字的合取是 PAC 学习的(算法见本节开始部分)，但也存在不是 PAC 学习的概念类，如 k-term-CNF 或 k-term-DNF。由于式(3-2)以 $|H|$ 刻画样本复杂度，它存在以下不足：可能导致非常弱的边界；对于无限假设空间的情形，式(3-2)根本无法使用，因此有必要引入另一度量标准——VC 维。

3.2.2 无限空间样本复杂度

使用 VC 维(vapnik-chervonenkis dimension)代替 $|H|$ 也可以得到样本复杂度的边界，基于 VC 维的样本复杂度比 $|H|$ 更紧凑，另外还可以刻画无限假设空间的样本复杂度。VC 维的概念是为了研究学习过程一致收敛的速度和推广性，是由统计学习理论定义的有关函数集学习性能的一个重要指标。传统的定义是：对一个指标函数集，如果存在 H 个样本能够被函数集中的函数按所有可能的 2^K 种形式分开，则称函数集能够把 H 个样本打散；函数集的 VC 维就是它能打散的最大样本数目 H。若对任意数目的样本，都有函数能将它们打散，则函数集的 VC 维是无穷大，有界实函数的 VC 维可以通过用一定的阈值将它转化成指示函数来定义。

VC 维反映了函数集的学习能力，VC 维越大，学习机器越复杂(分类能力越大)，所以 VC 维又是学习机器复杂程度的一种衡量。换个角度理解，如果用函数类$\{f(z,a)\}$代表一个学习机器，a 确定后就确定了一个判别函数，而 VC 维为该学习机器能学习的可以由其分类函数正确给出的所有可能二值标识的最大训练样本数。遗憾的是，目前尚没有通用的关于任意函数集 VC 维计算的理论，只知道一些特殊函数集的 VC 维。例如，在 n 维空间中线性分类器和线性实函数的 VC 维是 $n+1$。下面举一个简单的实例进一步理解 VC 维。

实例集合 X 为二维实平面上的点(x,y)，假设空间 H 为所有线性决策线。由图 3-2 可以看出：除 3 个点在同一直线上的特殊情况，x 中 3 个点构成的子集的任意划分均可被线性决策线打散，而 x 中 4 个点构成的子集，无法被 H 中的任一 h 打散，所以 $\text{VC}(H)=3$。

VC 维衡量假设空间复杂度的方法不是用不同假设的数量 $|H|$，而是用 X 中能被 H 彻底区分的不同实例的数量，这称为打散，可以简单理解为分类。H 的这种打散实例集合的能力是其表示这些实例上定义的目标概念的能力的度量，如果 X 的任意有限大的子集可被 H 打散，则 $\text{VC}(H)=\infty$，对于任意有限的 H，$\text{VC}(H)\leqslant\log_2|H|$。使用 VC 维作为 H 复杂度的度量，就有可能推导出该问题的另一种解答，类似于式(3-2)的边界，即

$$m \geqslant \frac{1}{\varepsilon}\left(4\log_2\frac{2}{\delta}+8\text{VC}(H)\log_2\frac{13}{\varepsilon}\right) \tag{3-4}$$

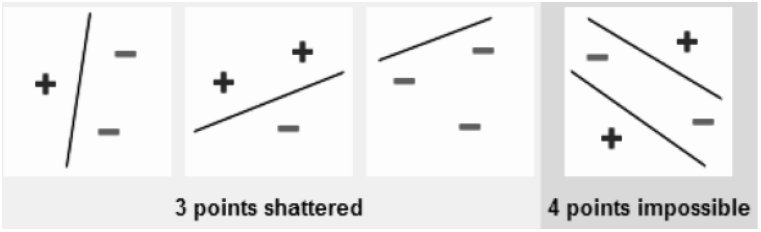

图 3-2　线性分类器三维示意图

由式(3-4)可以看到：要成功进行 PAC 学习，所需要的训练样本数应正比于$\frac{1}{\delta}$的对数，正比于 VC(H)，正比于 1/ε 的对数。

3.3　VC 维计算

设 $X=\{x_1,x_2,\cdots,x_m\}$是一个大小为 m 的采样集。每个假设 h 在 H 中标记一个样本在 X 中，结果表示为

$$h \mid X=\{h(x_1),h(x_2),\cdots,h(x_m)\} \tag{3-5}$$

随着 m 的增大，所有的假设 h 对 X 集合中的样本所能赋予的标记可能数也增大。当 $m\in N$，增长函数定义为

$$\Pi_H(m)=\max_{x_1,x_2,\cdots,x_m\subseteq X}\left|\{h(x_1),h(x_2),\cdots,h(x_m)\mid h\in H\}\right| \tag{3-6}$$

增长函数 $\Pi_H(m)$表示可以用假设空间 H 为 m 例子标记的可能结果的最大数目。H 可以为这些示例标记的可能结果越多，H 的表达能力就越强。

将样本集的数量翻倍 $X=\{x_1,x_2,\cdots,x_m,x_{m+1},\cdots,x_{2m}\}$，并生成子集 $X_1=\{x_1,x_2,\cdots,x_m\}$和 $X_2=\{x_{m+1},x_{m+2},\cdots,x_{2m}\}$，通常需要对原数据集进行复制操作，$X_1$、$X_2$ 样本集合类别定义见式(3-10)下方描述。X 的风险函数定义为

$$v(X)=\frac{1}{2m}\sum_{i=0}^{2m}\left|y_i-f(x_i)\right| \tag{3-7}$$

根据之前的研究，风险 $v(X_1)$和 $v(X_2)$之间的差异与样本集大小 m 正相关。两者的风险差异上界可以表示为

$$p=\sup\{v(X_1)-v(X_2)\}\propto m \tag{3-8}$$

进一步变化可得

$$\begin{aligned}&\frac{1}{m}\sum_{i=1}^{m}\left|y_i-f(x_i)\right|-\frac{1}{m}\sum_{i=m+1}^{2m}\left|y_i-f(x_i)\right|\\&=\left(1-\frac{1}{m}\sum_{i=1}^{m}\left|\widetilde{y}_i-f(x_i)\right|\right)-\frac{1}{m}\sum_{i=m+1}^{2m}\left|y_i-f(x_i)\right|\end{aligned} \tag{3-9}$$

即

$$p=\inf\left\{\left(\frac{1}{m}\sum_{i=1}^{m}\left|\widetilde{y}_i-f(x_i)\right|+\frac{1}{m}\sum_{i=m+1}^{2m}\left|y_i-f(x_i)\right|\right)\right\} \tag{3-10}$$

其中，$\widetilde{y}_i$ 是数据集的错误标签，实现层面需要将子集 X_1 的标签替换为错误标签，inf 表示下

限。当样本集大小 m 相同时，VC 维与 p 成正比。p 的结果将被标准化，存在 $\zeta,\varepsilon\in(0,\infty)$，这样 VC 维的形式如式(3-11)所示

$$\mathrm{VC}\propto\frac{\zeta}{\mathrm{e}^{-\frac{m\varepsilon^2}{8}}}p \tag{3-11}$$

p 是 VC 维的度量，可用来对不同模型的 VC 维进行排序。由于深度学习模型的巨大复杂性，式(3-9)中的最小值不容易估计。只能通过局部最优估计 p，这依赖深度学习优化器实现。此外，找到被打散样品的最大数量仍然是一个开放的问题，只能通过 VC 维的方法得到一个近似的 DNN 泛化性指标。该部分内容较难，更多细节参见文献[3]。

3.4 总　　结

PAC 学习是计算学习理论的基础，通过对 PAC 学习模型的分析，可帮助读者理解 VC 维的概念及训练数据对学习的有效性。当学习算法允许查询时是很有用的，并能提高其学习能力。此外，在实际的机器学习中，PAC 模型也存在不足之处：模型中强调最坏情况，它用最坏情况模型测量学习算法的计算复杂度及对概念空间中的每个目标概念和实例空间上的每个分布，用最坏情况下所需要的随机样本数作为其样本复杂度的定义，使得它在实际中不可用；定义中的目标概念和无噪声的训练数据在实际中是不现实的。

课 后 习 题

1. 简述可 PAC 学习的学习器需要满足什么条件。
2. VC 维理论是什么？为什么要提出 VC 维理论？