> 目的是分析学习任务的困难本质.为学习算法提供理论保证，并根据分析结果指导算法设计

一、PAC基本概念

误差参数ε： 预先设定的哦行目标误差要求

概念c：样本空间到标记空间的映射

h：假设出来的c

h与数据集D一致：h在D上经验误差为0

- 不合(disagreement)
  对于任意两个映射 $h_1, h_2 \in \mathcal{X} \rightarrow \mathcal{Y}$通过"不合"度量它们的差别

$$d(h_1, h_2) = P_{x \sim \mathcal{D}}(h_1(\boldsymbol{x}) \neq h_2(\boldsymbol{x}))$$

目标概念：把x都映射到了真实标记y上

概念类C：希望学得的目标概念集合

H：某个学习算法所考虑的所有可能概念的集合

> （与C通常不同）
> H包含了学习算法所有可能的输出假设

学习算法可分的&一致的：H中存在目标概念c

> 概率近似正确PAC：以较大概率学得满足ε的模型

PAC可辨识(PAC Identify)

学习算法若能以较大概率（1-δ）学得目标概念c的近似，称其能从假设空间H中PAC辨识概念类C

对$0 < \epsilon, \delta < 1$,所有 $c \in \mathcal{C}$ 和分布$\mathcal{D}$,若存在学习算法$\mathcal{L}$，其输出假设 $h \in \mathcal{H}$满足

$$P(E(h) \leq \epsilon) \geq 1 - \delta,$$

则称学习算法 $\mathcal{L}$ 能从假设空间 $\mathcal{H}$中PAC辨识概念类$\mathcal{C}$.

PAC可学习(PAC Learnable)

令 m表示从分布 $\mathcal{D}$中独立同分布采样得到的样例数目，$0 < \epsilon, \delta < 1$，对所有分布 $\mathcal{D}$,若存在学习算法$\mathcal{L}$ 和多项式时间 $poly(\cdot,\cdot,\cdot,\cdot)$，使得对于任何 $m \geq poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$, $\mathcal{L}$ 能从假设空间 $\mathcal{H}$ 中PAC辨识概念类$\mathcal{C}$，则称概念类$\mathcal{C}$对假设空间$\mathcal{H}$而言是PAC可学习的，有时也简称概念类 $\mathcal{C}$是PAC可学习的。

- 若算法的运行时间同时满足是如下的多项式函数：

$$poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$$

则概念类C是高效PAC可学习的，算法是概念类C的PAC学习算法

- 恰PAC可学习：假设空间H与概念类完全相同的情况

> 实际不可能

样本复杂度

假定学习算法$\mathcal{L}$处理每个样本的时间为常数，则$\mathcal{L}$的时间复杂度等价样本复杂度. 于是，我们对算法时间复杂度的关心就转化到对样本复杂度的关心.

定义 **样本复杂度(Sample Complexity)**

满足PAC学习算法$\mathcal{L}$ 所需的$m \geq poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$中最小的$m$，称为学习算法 $\mathcal{L}$ 的样本复杂度。

PAC意义

- 给出了一个抽象地刻画机器学习能力的框架，基于这个框架可以对很多重要问题进行理论探讨。
  - 研究某任务在什么样的条件下可学得较好的模型？
  - 某算法在什么样的条件下可进行有效的学习？
  - 需要多少训练样例才能获得较好的模型？
- 把对复杂算法的**时间复杂度**的分析转为对**样本复杂度**的分析

为什么不是希望精确地学到目标概念c呢？

机器学习过程受到很多因素的制约

- 训练集D只有有限的样例，因此通常会存在一些在D上“等效”的假设， 学习算法对它们无法区别。
- 从分布采样得到的过程有一定的偶然性，即便对同样大小的不同训练集， 学得结果也可能有所不同。

二、有限假设空间H

> H越大，越可能包含目标概念，但找到越难

1、可分情形

目标概念c属于H时

- 通常情形下，由于训练集规模有限，假设空间$\mathcal{H}$中可能存在不止一个与D一致的"等效"假设，对这些假等效假设，无法根据D来对它们的有优劣做进一步区分.

到底需要多少样例才能学得目标概念c的有效近似呢?

- 训练集 D的规模使得学习算法$\mathcal{L}$以概率$1-\delta$找到目标假设的ϵ近似，则：

$$m \geq \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta}).$$

- 有限假设空间$\mathcal{H}$都是PAC可学习的，所需的样例数目如上式所示，输出假设h的泛化误差随样例数目的增多而收敛到0，收敛速率为$O(\frac{1}{m})$.

2、不可分情形

定义 **不可知PAC可学习(agnostic PAC Learnable)**

令 m表示从分布$\mathcal{D}$中独立同分布采样得到的样例数目，$0<\epsilon,\delta<1$，对所有分布$\mathcal{D}$,若存在学习算法$\mathcal{L}$和多项式时间 $poly(\cdot,\cdot,\cdot,\cdot)$，使得对于任何 $m \geq poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$，$\mathcal{L}$ 能从假设空间$\mathcal{H}$中输出满足下式的假设 h：

$$P(E(h) - \min_{h'\in\mathcal{H}} E(h') \leq \epsilon) \geq 1-\delta,$$

则称假设空间 $\mathcal{H}$ 是不可知PAC可学习的.

- 若学习算法 $\mathcal{L}$ 的运行时间也是多项式函数 $poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$，则
  - 称假设空间$\mathcal{H}$是高效不可知PAC可学习的;
  - 称学习算法$\mathcal{L}$为假设空间$\mathcal{H}$的不可知PAC学习算法;
  - 称满足上述要求最小的m为学习算法 $\mathcal{L}$ 的样本复杂度.

三、VC维

1、增长函数

假设空间H对m个示例能赋予标记的最大可能数

定义 12.6 对所有 $m \in \mathbb{N}$，假设空间 $\mathcal{H}$ 的增长函数 $\Pi_{\mathcal{H}}(m)$ 为

$$\Pi_{\mathcal{H}}(m) = \max_{\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\subseteq\mathcal{X}} \left|\{(h(\boldsymbol{x}_1),\ldots,h(\boldsymbol{x}_m)) \mid h\in\mathcal{H}\}\right|.$$

> 2分类最大就是2^m

- 描述了H的表示能力，反映出H的复杂度，表示能力越强，对学习任务适应能力就越强
- 增长函数可以估计经验误差和泛化误差的关系

定理 12.2 对假设空间 $\mathcal{H}$, $m \in \mathbb{N}$, $0 < \epsilon < 1$ 和任意 $h \in \mathcal{H}$ 有

$$P\left(\left|E(h) - \widehat{E}(h)\right| > \epsilon\right) \leqslant 4\Pi_{\mathcal{H}}(2m)\exp\left(-\frac{m\epsilon^2}{8}\right).$$

对分dichotomy：对二分类而言，H中假设对D中示例赋予标记的每种结果都是对D的一种对分

打散shattering：H能实现D上的所有对分，称D被H打散

2、VC维

为什么引入VC维

> 实际学习任务中的无限假设空间，研究可学习性需要度量假设空间复杂度，最常用的是VC维

假设空间$\mathcal{H}$的VC维是能被$\mathcal{H}$打散的最大示例集的大小, 即

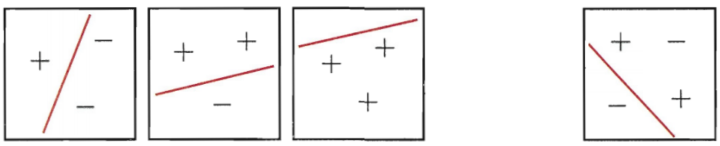
$$\mathrm{VC}(\mathcal{H}) = \max\{m : \textstyle\prod_{\mathcal{H}}(m) = 2^m\}.$$

存在大小为d的示例集能被打散就行，只需要构造出来

例子理解：

例 12.1 实数域中的区间 $[a,b]$: 令 $\mathcal{H}$ 表示实数域中所有闭区间构成的集合 $\{h_{[a,b]} : a,b \in \mathbb{R}, a \leqslant b\}$, $\mathcal{X} = \mathbb{R}$. 对 $x \in \mathcal{X}$, 若 $x \in [a,b]$, 则 $h_{[a,b]}(x) = +1$, 否则 $h_{[a,b]}(x) = -1$. 令 $x_1 = 0.5$, $x_2 = 1.5$, 则假设空间 $\mathcal{H}$ 中存在假设 $\{h_{[0,1]}, h_{[0,2]}, h_{[1,2]}, h_{[2,3]}\}$ 将 $\{x_1, x_2\}$ 打散, 所以假设空间 $\mathcal{H}$ 的 VC 维至少为 2; 对任意大小为 3 的示例集 $\{x_3, x_4, x_5\}$, 不妨设 $x_3 < x_4 < x_5$, 则 $\mathcal{H}$ 中不存在任何假设 $h_{[a,b]}$ 能实现对分结果 $\{(x_3,+), (x_4,-), (x_5,+)\}$. 于是, $\mathcal{H}$ 的 VC 维为 2.

例 12.2 二维实平面上的线性划分: 令 $\mathcal{H}$ 表示二维实平面上所有线性划分构成的集合, $\mathcal{X} = \mathbb{R}^2$. 由图 12.1 可知, 存在大小为 3 的示例集可被 $\mathcal{H}$ 打散, 但不存在大小为 4 的示例集可被 $\mathcal{H}$ 打散. 于是, 二维实平面上所有线性划分构成的假设空间 $\mathcal{H}$ 的 VC 维为 3.



存在这样的集合，其 $2^3 = 8$ 种对分均可被线性划分实现

(a) 示例集大小为 3

对任何集合，其 $2^4 = 16$ 种对分中至少有一种不能被线性划分实现

(b) 示例集大小为 4

图 12.1 二维实平面上所有线性划分构成的假设空间的 VC 维为 3

四、Rademacher复杂度

> 与VC维不同的是，它在一定程度上考虑了数据分布

基于Rademacher复杂度的泛化误差界依赖于具体学习问题的数据分布, 类似于为该问题“量身定制”的, 因此它通常比基于VC维的泛化误差界要更紧一些

五、稳定性

VC维和Rademacher复杂度来分析泛化性能, 得到的结果均与具体的学习算法无关

稳定性(stability)分析是这方面值得关注的一个方向。考察算法在输入(训练集)发生变化时，输出是否发生较大的变化

稳定性评价什么?

p 损失函数

$\ell(\mathcal{L}_D(\boldsymbol{x}), y) : \mathcal{Y} \times \mathcal{Y} \longmapsto \mathbb{R}^+$ 刻画假设$\mathcal{L}_D$的预测标记$\mathcal{L}_D(\boldsymbol{x})$与真实标记$y$之间的差别, 简记为$\ell(\mathcal{L}_D, \boldsymbol{z})$.

- 泛化损失

$$\ell(\mathcal{L}, D) = \mathbb{E}_{x \in \mathcal{X}, z=(\boldsymbol{x}, y)}[\ell(\mathcal{L}_D, \boldsymbol{z})].$$

- 经验损失

$$\hat{\ell}(\mathcal{L}, D) = \frac{1}{m} \sum_{i=1}^{m} \ell(\mathcal{L}_D, \boldsymbol{z}_i).$$

- 留一(leave-one-out)损失：

$$\ell_{loo}(\mathcal{L}, D) = \frac{1}{m} \sum_{i=1}^{m} \ell(\mathcal{L}_{D^{\setminus i}}, \boldsymbol{z}_i).$$