单词索引

Probably Approximately Correct - 概率估计准确
Agnostic PAC - 不可知PAC

一个标准学习模型

本章我们将学习一个标准的学习模型——PAC学习模型。我们将考虑其他的学习？？在第七章。

3.1 PAC学习

前一章，我们看到，随着训练样本增大大，输出假设将越来越接近准确值。更一般的，我们在这里定义*概率估计准确(Probably Approximately Correct)*学习。

定义3.1(PAC Learnability) 若假设集合H是**PAC可学习(PAC Learnable)**的，当存在函数 $m_H:(0,1)^2 \rightarrow \mathbb{N}$，并且学习算法拥有下面性质：对于每个$\epsilon, \delta \in (0,1)$,以及每个X上的分布D,以及每个标签函数$f: X \rightarrow \{0,1\}$,如果满足 *可实现假定*. 当算法作用在训练集$m \geq m_H(\epsilon,\delta)$ 分布D时，能够以至少$1-\delta$的概率输出算法$h \in H$，使得$L_{D,f}(h) < \epsilon$

PAC的定义包含了两个参数ϵ,δ. 准确度参数ϵ描述了输出算法距离最优解有多远。置信参数δ描述了算法给出该准确度的可能性（对应PAC中的P）。在一些数据相关的模型里，不可避免地要去估计。因为训练集是基于概率分布的随机产生，因此总会存在一定（很小）几率出现非信息性样本（比如，样本集只有一个样本，被反复抽到）。进一步说，即使我们足够幸运，能够抽到足够多的样本，如实反映了分布D，但是总会有一些D的细节被忽视。准确度参数则允许学习器犯点小错误。

样本复杂度

函数$m_H:(0,1)^2 \rightarrow \mathbb{N}$决定了H的样本复杂度，即需要多少样本才能确定*概率估计准确*的解。样本复杂度是ϵ,δ的函数，她也依赖于假设集合H的属性。比如，对于有限集合的情况，样本复杂度依赖于H大小的对数。

注意，如果H是PAC可学习的，那么会存在很多个m_H满足上面PAC可学习的定义。因此，这里我们定义样本复杂度为最小函数，也就是m_H是使得上述定义成立的最小的整数。

让我们总结一下上一章：

推论3.2 每个有限假设集合都是PAC可学习的，样本复杂度$m_H \leq \left[\frac{log(H/\delta)}{\epsilon}\right]$

当然，也有无限集合是PAC可学习的（比如习题3）。之后我们会学到，决定一个假设集合是不是PAC可学习的，并不取决于它是否有限，而是取决于一个叫*VC维度*的东西。

3.2 更广义的学习模型

我们刚提到的模型其实很容易被广义化，所以它可以被用来做更大范围的学习任务。我们从两个方面广义化它。

去除实现假定

我们之前要求学习算法的D和f要满足实现假定。然而这个假定有些时候太强了。（比如。。）在下个小节中，我们会学到*不可知PAC*模型，其*实现假定*就被抛弃了。

超越二元分类

到目前为止我们讨论的学习任务都是二元的。但是生活中，我们需要预测各种数字啦，有限多的标签啦。这时候，分析方法可以很容易的拓展到这些情况。只需要用点不一样的误差函数。我们会稍后在3.2.2节讨论。

3.2.1 去除实现假定 - 不可知PAC

一个更现实的模型

之前实现假定要求必须存在一个$h^* \in H$使得$P(h^*(x) = y) = 1$。然而，在大多数实际问题中，这个假定并不成立。更进一步，标签可能并不完全由已知的观测量给出。（比如甜瓜例子中，两个各种属性都一样的甜瓜口味仍可能有不同。）因此，我们接下来去掉实现假定，用一个更灵活的判据——数据标签生成分布 (data-labels generating distribution) 来取代 目标标签函数。

我给你讲，从今以后，D就是$X \times Y$的概率分布了。也就是说，D是定义域和标签的联合分布。你可以把它看作由两部分组成的：一个是无标签的定义域分布D_x，一个是对于定义域上标签的条件概率$D((x, y)|x)$。在甜瓜例中，D_x描述了甜瓜的各种属性如色泽、硬度等的概率分布。而条件分布则描述在如此色泽、硬度等情况下，瓜是甜的的概率。你会发现，这种定义囊括了色泽硬度相同的瓜却有不同口味的情况。

一个经验的真实误差修正

对于$X \times Y$的概率分布D，我们可以测量它的误差。重新定义预测规则h的真实误差为：

$$L_D(h) = P(h(x) \neq y) = D((x, y) : h(x) \neq y) \tag{1}$$

我们想要找到一个h使得上述误差最小。然鹅，我们并不知道数据生成D。我们唯一能够知晓的就是训练集S。经验误差仍然保持之前的定义：

$$L_S(h) = \frac{|\{i \in [m], h(x_i) \neq y_i\}|}{m} \tag{2}$$

给定S，学习者可以机算$L_S(h)$对于任意的$h : X \to Y$. 注：
$L_S(h) = L_{D(uniform)}(h)$

目标

找到一个预测h使得真实误差$L_D(h)$最小

贝叶斯最优预测器

给定从X到{0,1}的任意概率分布D，最佳标签预测函数是：

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{若 } \mathbb{P}[y=1|x] \geq 1/2 \\ 0 & \text{其它情况} \end{cases} \tag{3}$$

很容易验证（不信你做习题7）对于所有概率分布D，贝叶斯最优预测器都是最优的，也就是说，不存在一个预测器$g : X \to \{0,1\}$能给出更小的误差。换句话说，对于任意预测器g, $L_D(f_D) \leq L_D(g)$

不幸的是，由于我们不知道分布D是啥，也就不能使用这个最优预测器。学习者能够知道的就是只有训练集。不过，我们现在可以给不可知PAC学习型下定义了。就是把之前的PAC定义拓展到这个更现实的，非实现假定的情况来。

定义3.3(不可知PAC学习性) 一个假设集H是不可知PAC学习性的。当存在函数$m_H : (0,1)^2 \to \mathbb{N}$，并且学习算法拥有下面性质：对于每个$\epsilon, \delta \in (0,1)$,以及每个$X \times Y$上的分布D,以及每个标签函数$f : X \to \{0,1\}$,如果满足 *可实现假定*. 当算法作用在训练集$m \geq m_H(\epsilon,\delta)$ 分布D时，能够以至少$1-\delta$的概率输出算法$h \in H$，使得$L_D(h) \leq \min_{h'} L_D(h') + \epsilon$

可以看到，如果存在实现假定，上述定义则回到PAC学习的情况。从这个意义上来说，不可知PAC是PAC定义的广义化。当实现假定不满足时，没有学习者可以保证一个任意小的误差。反之，在不可知PAC的定义下，学习者仍然有效，只要它的误差比最优预测器的误差大的不多。与PAC相比，不可知PAC下的学习者的误差并不是一个绝对值，而是与最优预测器相比的一个相对值。

学习问题模型范围

我们进一步拓展我们的模型，以便让她可以适用到更多种学习任务。我们先来考虑几个不同的学习样例。

- 多类分类 我们的分类不需要是二元的。比如，对于文档的分类。我们希望设计一个程序可以把这些文档依照特定主题分类（比如，新闻、体育、生物、医药）一个学习算法，应该能够把这些文档输入，然后输出他们的主题。定义集是所有可能的文档的集合。记得，我们通常把这些文档表示为一系列的特性，包含文档中关键词的个数，以及相关的特性，比如文档的大小或来源。任务中的标签集将会是一系列可能的文档主题（所以Y是一个大而有限的集）。一旦我们决定了定义集与标签集，其他的部分看起来就和甜瓜例子一样了。

- 回归 在这个任务中，人们希望在数据中找到一个简单的规律，一个数据中X和Y的函数关系。比方说，一个人希望找到一个线性函数，能够最好的预测婴儿出生体重，基于一些超声波测量结果，如头部环境、腹部环境、腿骨长度。这里，我们的定义域X是R3的子集（三个超声波测量值），标签集Y是实数集（体重）。在这个情境中，Y更合适的叫法是目标集。我们的训练数据与学习机的输出和之前一样。但是，成功的判定与之前大不相同。我们评估假定函数的水平是根据真实标签与预测值的 期待方差。

  即，$L_D(h) = E(h(x)-y)^2$

为了适用更广泛的学习任务，我们需要把我们测量公式推广如下：

广义化误差函数

给定任意集H与域Z，存在l是一个映射满足从$H \times Z$到一个非负实数，$l: H \times Z \rightarrow R_+$, 则我们称l是一个误差函数。

对于预测问题，我们有$Z = X \times Y$，然而我们的定义不仅限于预测问题，而是允许Z是任意的域（如第22章中，Z可以不是实例域域标签域的乘积）。

现在我们定义 危险函数 为一个分类器的期待误差。$L_D(h) = E[l(h, z)]$

上面我们考虑的是误差h在随机从D选取的z上的误差值的期待值。类似的，我们定义经验危险为给定样本 $S = (z1, z2, \ldots, z_m) \in Z^m$, 有$L_S(h) = \frac{1}{m}\sum_i l(h, z_i)$

常用的误差函数有，

- 0-1 误差

  $$l_{0-1}(h,(x,y)) = \begin{matrix} 0, & \text{if } h(x) = y \\ 1, & \text{if } h(x) \neq y \end{matrix}$$

  这个误差函数经常用在二元问题与多分类问题中。

- 平方误差

  $$l_{sq}(h,(x,y)) = (h(x)-y)^2 \tag{4}$$

我们之后会看到更多有用的误差函数的例子。

总结一下，我们正式定义广义误差下的不可知PAC学习性

定义3.4(广义误差下不可知PAC学习性) 当存在函数 $m_H : (0,1)^2 \rightarrow \mathbb{N}$，并且学习算法拥有下面性质：对于每个$\epsilon, \delta \in (0,1)$,以及每个$X \times Y$上的分布D,以及每个标签函数$f : X \rightarrow \{0,1\}$,如果满足 *可实现假定*. 当算法作用在训练集 $m \geq m_H(\epsilon, \delta)$ 分布D时，能够以至少$1-\delta$的概率输出算法$h \in H$，使得 $L_D(h) \leq \min_{h'} L_D(h') + \epsilon$. 其中$L_D(h) = \mathbb{E}_{z \sim D}[l(h, z)]$

评论3.1 前文中，我们定义$l(h, \cdot) : Z \rightarrow R_+$ 为一个随机变量，定义$L_D(h)$为这个随机变量的期望值。这就要求我们的l是可测量的。因此给出正式定义，我们假设存在σ是Z的子集。并且R_+的每个逆像都在σ中。对于01损失，我们对于l的假设就等同于假设对于每个h，集$(x, h(x)) : x \in X$ 都在σ中。

评论3.2

总结

本章，我们定义了正式的学习模型-PAC学习。这个模型基于了可实现假定，不过不可知型没有加入这个限制。我们也广义化了PAC模型到任意损失函数。我们有时候也会把最广义的模型简化为PAC学习，而省略 不可知的前缀，并让读者知晓损失函数的来历。当我们想要强调我们处理的是原始的PAC设定时，我们会提到可实现假定。第七章我们将会讨论学习性的其他问题。

练习

1. 样本复杂度
2. 让X
3. 让X
4. 在这个问题
5. 让X
6. 让H
7. 贝耶斯优化估计器
8. 我们
9. 考虑到