3 PAC 学习框架

3.1 引言

在介绍完基础知识后，本章及后续几章将介绍计算学习理论. 计算学习理论 (computational learning theory) 研究的是关于通过" 计算" 来进行学习的理论, 即关于机器学习的理论基础, 其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. 例如, 哪些问题可以被高效地学习? 哪些问题本质上是学习困难的? 需要多少样例才能够有效学习? 上述这些问题都属于计算学习理论的研究范畴. 这一章, 主要介绍的内容包括: 概率近似正确 (Probably Approximately Correct), 概念类, 假设空间, 可学习性, 不可知 PAC 可学习性.

给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, $\boldsymbol{x}_i \in \mathcal{X}$, 本章主要讨论二分类问题, 若无特别说明, $y_i \in \mathcal{Y} = \{-1, +1\}$. 假设 $\mathcal{X}$ 中的所有样本来源于一个隐含未知的分布 $\mathcal{D}$, D 中的所有样本都是独立地从这个分布上采样得到, 即独立同分布 (independent and identically distributed, 简称 *i.i.d.*) 样本.

令 h 为从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的一个映射, 其泛化误差为

$$E(h; \mathcal{D}) = P_{\boldsymbol{x} \sim \mathcal{D}}(h(\boldsymbol{x}) \neq y), \tag{18}$$

h 在 D 上的经验误差为

$$\widehat{E}(h; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(h(\boldsymbol{x}_i) \neq y_i). \tag{19}$$

由于 D 是 $\mathcal{D}$ 的独立同分布采样, 因此 h 的经验误差的期望等于其泛化误差. 在上下文明确时, 我们将 $E(h; \mathcal{D})$ 和 $\widehat{E}(h; D)$ 分别简记为 $E(h)$ 和 $\widehat{E}(h)$. 令 ϵ 为 $E(h)$ 的上限, 即 $E(h) \leqslant \epsilon$; 我们通常用 ϵ 表示预先设定的学得模型所应满足的误差要求, 亦称" 误差参数". 若 h 在数据集 D 上的经验误差为 0, 则称 h 与 D 一致, 否则称其与 D 不一致. 对任意两个映射 h_1, $h_2 \in \mathcal{X} \to \mathcal{Y}$, 可通过其" 不和"(disagreement) 来度量它们之间的差别:

$$d(h_1, h_2) = P_{\boldsymbol{x} \sim \mathcal{D}}(h_1(\boldsymbol{x}) \neq h_2(\boldsymbol{x})). \tag{20}$$

本章的组织结构如下,3.2 节介绍 PAC 学习理论, 3.3 节介绍概念类与假设空间.

3.2 PAC 学习

计算学习理论中最基本的是概率近似正确 (Probably Approximately Correct, 简称 PAC)[Valiant, 1984], 这个名字看起来有些古怪, 我们稍后再解释.

令 c 表示" 概念"(concept), 这是从样本空间 $\mathcal{X}$ 到标记空间 $\mathcal{Y}$ 的映射, 它决定示例 $\boldsymbol{x}$ 的真实标记 y, 若对任何样例 $(\boldsymbol{x}, y)$ 有 $c(\boldsymbol{x}) = y$ 成立, 则称 c 为目标概念; 所有我们希望学得的目标概念所构成的集合称为" 概念类"(concept class), 用符号 $\mathcal{C}$ 表示.

本章主要讨论二分类问题, 即 $\mathcal{Y} = \{-1, 1\}$ 或$\{0, 1\}$, 因此可以将 c 等价于样本空间 $\mathcal{X}$ 的一个子集 $\{\boldsymbol{x} \in \mathcal{X} : c(\boldsymbol{x}) = 1\}$.

例 1：三角形概念

一个概念可以是一个三角形内的所有点, 或者说成是把三角形内的点映为 1, 把其余的点映为 -1 的一个映射, 此时可以简称要学习的概念是一个三角形. 一个" 概念类" 则是想要学习的概念的组成的集合. 例如, 平面上的所有的三角形即构成了一个概念类.

给定学习算法 $\mathfrak{L}$, 它所考虑的所有可能概念的集合称为" 假设空间"(hypothesis space), 用符号 $\mathcal{H}$ 表示. 由于学习算法 $\mathfrak{L}$ 事先并不知道概念类的真实存在, 因此 $\mathcal{H}$ 和 $\mathcal{C}$ 通常是不同的, 学习算法会把自认为可能的目标概念集中起来构成 $\mathcal{H}$, 对 $h \in \mathcal{H}$, 由于并不能确定它是否真是目标概念, 因此称为" 假设"(hypothesis). 显然, 假设 h 也是从样本空间 $\mathcal{X}$ 到标记空间 $\mathcal{Y}$ 的映射.

若目标概念 $c \in \mathcal{H}$, 则 $\mathcal{H}$ 中存在假设能将所有示例完全正确分开, 称该问题是" 可分的"(separable), 亦称" 一致的"(consistent)；若 $c \notin \mathcal{H}$, 则 $\mathcal{H}$ 中不存在任何假设能将所有示例完全正确分开, 称该问题是" 不可分的"(non-separable), 亦称" 不一致的"(non-consistent).

学习任务通常可如下刻画：学习算法接收数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_m, y_m)\}$ 或 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\}$ 以及标记 $\{c(\boldsymbol{x}_1), c(\boldsymbol{x}_2), \dots, c(\boldsymbol{x}_m)\}$ (c 是待学习的目标概念), 然后返回一个假设 $h \in \mathcal{H}$, 使得假设 h(对应于目标概念 c) 有最小的泛化误差. 给定假设 $h \in \mathcal{H}$、目标概念 $c \in \mathcal{C}$ 以及潜在的分布 $\mathcal{D}$, h 的泛化误差定义为

$$E(h; \mathcal{D}) = P_{\boldsymbol{x} \sim \mathcal{D}}\big(h(\boldsymbol{x}) \neq c(\boldsymbol{x})\big) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{I}(h(\boldsymbol{x}) \neq c(\boldsymbol{x}))],$$

一般不能直接得知一个假设的泛化误差, 因为分布 $\mathcal{D}$ 和目标概念 c 都是未知的. 通常可以利用数据集 D 来衡量假设的经验误差：

$$\widehat{E}(h; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(h(\boldsymbol{x}_i) \neq y_i).$$

对于 $h \in H$, 其经验误差的期望值等于泛化误差, 即：

$$\mathbb{E}_{D \sim \mathcal{D}^m}[\widehat{E}(h; D)] = E(h; \mathcal{D}).$$

事实上, 由期望的线性性和样本独立同分布这两个条件, 有:

$$\mathbb{E}_{D \sim \mathcal{D}^m}[\widehat{E}(h; D)] = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{D \sim \mathcal{D}^m}[\mathbb{I}(h(\boldsymbol{x}_i) \neq c(\boldsymbol{x}_i))] = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{D \sim \mathcal{D}^m}[\mathbb{I}(h(\boldsymbol{x}) \neq c(\boldsymbol{x}))],$$

对 $\forall \boldsymbol{x} \in D$. 因此

$$\mathbb{E}_{D \sim \mathbb{D}^m}[\widehat{E}(h; D)] = \mathbb{E}_{D \sim \mathcal{D}^m}[\mathbb{I}(h(\boldsymbol{x}) \neq c(\boldsymbol{x}))] = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\mathbb{I}(h(\boldsymbol{x}) \neq c(\boldsymbol{x}))] = E(h; \mathcal{D}).$$

给定训练集 D, 我们希望学得模型所对应的假设 h 尽可能接近目标概念 c. 读者可能会问, 为什么不是希望精确地学得目标概念 c 呢? 这是由于机器学习过程受到很多因素的制约, 例如我们获取的训练集 D 往往仅包含有限数量的样例, 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；再如, 从分布 $\mathcal{D}$ 采样得到 D 的过程有一定偶然性, 可以想象, 即使对同样大小的不同训练集, 学得结果也可能有所不同. 因此, 我们是希望以比较大的把握学得比较好的模型, 也就是说, 以较大的概率学得误差满足预设上限的模型；这就是" 概率"" 近似正确" 的含义, 形式化地说, 令 δ 表示置信度, 可定义：

定义 3.1. *PAC 辨识 (PAC Identify)*: *对 $0<\epsilon,\delta<1$, 所有 $c\in\mathcal{C}$ 和分布 $\mathcal{D}$, 若存在学习算法 $\mathfrak{L}$, 其输出假设 $h\in\mathcal{H}$ 满足*

$$P(E(h)\leqslant\epsilon)\geqslant 1-\delta, \tag{21}$$

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

这样的学习算法 $\mathfrak{L}$ 能以较大的概率 (至少 $1-\delta$) 学得目标概念 c 的近似 (误差最多为 ϵ), 在此基础上可定义：

定义 3.2. *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 $\mathfrak{L}$ 和多项式函数 $poly(.,.,.,.)$, 使得对于任何 $m\geqslant poly(1/\epsilon,1/\delta\ ,size(\boldsymbol{x}),size(c))$, $\mathfrak{L}$ 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.*

直观地来说, 如果概念类 $\mathcal{C}$ 是 PAC 可学习的, 则学习算法能够在观察了一定数量的样本点后以较高概率 (至少 $1-\delta$) 返回近似正确 (错误率小于 ϵ) 的假设, 这就是" 概率"" 近似正确" 学习, 其中 $\delta>0$ 是置信度参数, $\epsilon>0$ 是精度参数.

对算法来说, 必然还要考虑时间复杂度, 于是有

定义 3.3. *PAC 学习算法 (PAC Learning Algorithm)*: *若学习算法 $\mathfrak{L}$ 使概念类 $\mathcal{C}$ 为 PAC 可学习的, 且 $\mathfrak{L}$ 的运行时间也是多项式函数 $poly(1/\epsilon,1/\delta,size(\boldsymbol{x}),size(c))$, 则称概念类 $\mathcal{C}$ 是高效 PAC 可学习 (efficiently PAC learnable) 的, 称 $\mathfrak{L}$ 为概念类 $\mathcal{C}$ 的 PAC 学习算法.*

显然, PAC 学习给出了一个抽象地刻画机器学习能力的框架, 基于这个框架能对很多重要问题进行理论探讨, 例如研究某任务在什么样的条件下可学得较好的模型？需多少训练样例才能获得较好的模型?

对较为困难的学习问题, 目标概念 c 往往不存在于假设空间 $\mathcal{H}$ 中, 假定对于任何 $h\in\mathcal{H}$,$\widehat{E}(h)\neq 0$, 也就是说, $\mathcal{H}$ 中的任意一个假设都会在训练集上出现或多或少的错误. 由 Hoeffding 不等式易知：

引理 3.1. *若训练集 D 包含 m 个从分布 $\mathcal{D}$ 上独立同分布采样而得的样例, $0<\epsilon<1$, 则对任意 $h\in\mathcal{H}$, 有*

$$P(\widehat{E}(h)-E(h)\geq\epsilon)\leq\exp(-2m\epsilon^2)$$
$$P(E(h)-\widehat{E}(h)\geq\epsilon)\leq\exp(-2m\epsilon^2)$$
$$P(|E(h)-\widehat{E}(h)|\geq\epsilon)\leq 2\exp(-2m\epsilon^2)$$

推论 3.1. *若训练集 D 包含 m 个从分布 $\mathcal{D}$ 独立同分布采样而得的样例, $0<\epsilon<1$, 则对任意 $h\in\mathcal{H}$, 式 (22) 至少 $1-\delta$ 的概率成立:*

$$\widehat{E}(h)-\sqrt{\frac{\ln(2/\delta)}{2m}}\leq E(h)\leq\widehat{E}(h)+\sqrt{\frac{\ln(2/\delta)}{2m}}. \tag{22}$$

推论 3.1表明, 样例数目 m 较大时, h 的经验误差是其泛化误差很好的近似.

显然, 当 $c \notin \mathcal{H}$ 时, 学习算法 $\mathfrak{L}$ 无法学得目标概念 c 的 ϵ 近似. 但是, 当假设空间 $\mathcal{H}$ 给定时, 其中必存在一个泛化误差最小的假设, 找出此假设的 ϵ 近似也不失为一个较好的目标. $\mathcal{H}$ 中泛化误差最小的假设是 $\text{argmin}_{h\in\mathcal{H}} E(h)$, 于是, 以此为目标可将 PAC 学习推广到 $c \notin \mathcal{H}$ 的情形, 这称为" *不可知学习 "(agnostic learning)* . 相应的, 我们有：

定义 3.4. 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 $\mathfrak{L}$ 和多项式函数 $poly(.,.,.,.)$, 使得对于任何 $m \geqslant poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$, $\mathfrak{L}$ 能从假设空间 $\mathcal{H}$ 中输出满足下式的假设 h:*

$$P\left(E(h) - \min_{h'\in\mathcal{H}} E(h') \le \epsilon\right) \geqslant 1 - \delta, \tag{23}$$

则称假设空间 $\mathcal{H}$ 是不可知 PAC 可学习的.

与 PAC 可学习类似, 若学习算法 $\mathfrak{L}$ 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*, 学习算法 $\mathfrak{L}$ 则称为假设空间 $\mathcal{H}$ 的不可知 PAC 学习算法.

这里需要强调一下 PAC 定义中的几个关键点: PAC 框架是一种" 分布无关" 的模型：并未对产生样本的分布 $\mathcal{D}$ 作任何假设；训练样本和用来计算错误率的测试样本都来自于同一个分布 $\mathcal{D}$, 要使 PAC 模型能够得到推广, 这是一个必不可少的假定; PAC 模型处理的是某个概念类 $\mathcal{C}$ 的可学习性, 而不是某个特定的概念, 通常目标概念 $c \in \mathcal{C}$ 对学习算法来说是未知的.

3.3 概念类和假设空间

在 PAC 中有两个比较重要的概念, 分别是" 概念类" 和" 假设空间". 前面提及" 目标概念"c 是从样本空间 $\mathcal{X}$ 到标记空间 $\mathcal{Y}$ 的映射, 它决定示例 $\boldsymbol{x}$ 的真实标记 y, 对任何样例 $(\boldsymbol{x}, y)$ 有 $c(\boldsymbol{x}) = y$ 成立. 所有希望学得的目标概念所构成的集合称为" 概念类"(concept class), 用符号 $\mathcal{C}$ 表示.

给定学习算法 $\mathfrak{L}$, 它所考虑的所有可能概念的集合称为" 假设空间"(hypothesis space), 用符号 $\mathcal{H}$ 表示. 由于学习算法 $\mathfrak{L}$ 事先并不知道概念类的真实存在, 因此 $\mathcal{H}$ 和 $\mathcal{C}$ 通常是不同的, 学习算法会把自认为可能的目标概念集中起来构成 $\mathcal{H}$, 对 $h \in \mathcal{H}$, 由于并不能确定它是否真是目标概念, 因此称为" 假设"(hypothesis). 显然, 假设 h 也是从样本空间 $\mathcal{X}$ 到标记空间 $\mathcal{Y}$ 的映射.

若目标概念 $c \in \mathcal{H}$, 则 $\mathcal{H}$ 中存在假设能将所有示例按与真实标记一致的方式完全分开, 我们称该问题对学习算法 $\mathfrak{L}$ 是" 可分的"(separable), 亦称" 一致的"(consistent)；若 $c \notin \mathcal{H}$, 则 $\mathcal{H}$ 中不存在任何假设能将所有示例完全正确分开, 称该问题对学习算法是" 不可分的"(non-separable), 亦称" 不一致的"(non-consistent).

PAC 学习中一个关键因素是假设空间 $\mathcal{H}$ 的复杂度. $\mathcal{H}$ 包含了学习算法 $\mathfrak{L}$ 所有可能输出的假设, 若在 PAC 学习中假设空间与概念类完全相同, 即 $\mathcal{H} = \mathcal{C}$, 这称为" 恰 PAC 可学习"(properly PAC learnable); 直观上看, 这意味着学习算法的能力与学习任务" 恰好匹配". 然而, 这种让所有候选假设都来自概念类的要求看似合理, 但却不实际, 因为在现实应用中我们对概念类 $\mathcal{C}$ 通常一无所知, 更别说获得一个假设空间与概念类恰好相同的学习算法. 显然, 更重要的是研究假设空间与概念类不同的情形, 即 $\mathcal{H} \neq \mathcal{C}$. 一般而言, $\mathcal{H}$ 越大, 其包含的任意目标概念的可能性越大, 但从中找到某个具体目标概念的

难度就越大. $|\mathcal{H}|$ 有限时, 我们称 $\mathcal{H}$ 为" 有限假设空间", 否则称为" 无限假设空间". 对于有限假设空间, 由于其个数是有限的, 可以用个数来衡量复杂度

例 2：轴平行矩形概念类的学习

考虑如下场景, 数据集是平面上的点集, $\mathcal{X}=\mathbb{R}^2$, 概念类 C 是 $\mathbb{R}^2$ 上的所有四条边与坐标轴平行的矩形构成的集合. 因此, 每一个概念 c 是在某个特定的轴平行矩形中的所有的点. 学习问题是, 使用带标记的训练样本确定目标轴平行矩形. 我们将证明这个轴平行矩形概念类是 PAC 可学习的.

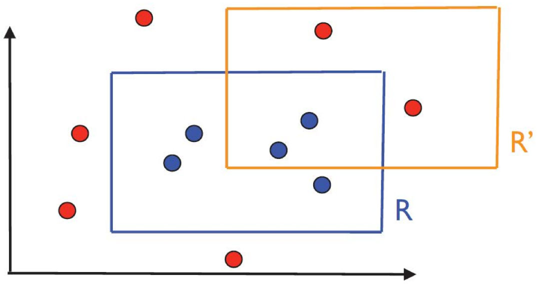

图 1: 目标概念 R 和可能的假设 R', 图中小圆代表训练样例, 蓝点的标记为 1, 其落在蓝色矩形 R 内, 红点的标记为 0.

图 1中 R 代表目标轴平行矩形, R' 代表一个假设. 从此图中可以看出, R' 的错误区域恰为 $(R-R')\cup(R'-R)$(即在 R 里但在 R' 外面和在 R' 里但在 R 外的区域), 第一块区域对应假反例 (false negative), 第二块区域则对应假正例 (false positive). 为了展现这个概念类是 PAC 可学习的, 我们先给出一个简单的 PAC 学习算法 $\mathcal{A}$, 该算法对于给定的数据集 D 返回包含所有被标记为 1 的点的最小的轴平行矩形 R_D, 即 $R'=R_D$.

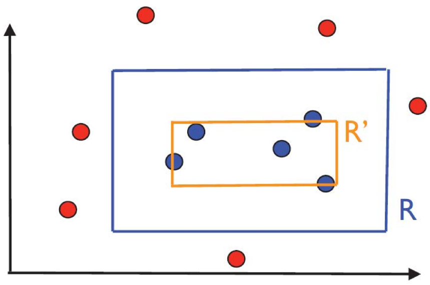

图 2: 由算法返回的假设 $R'=R_D$ 的示意图

图 2 展示了由此算法返回的假设. 由定义可知, R_D 不会产生任何假正例, 这是因为 R_D 中的点一定被包含在目标概念 R 中. 这样, R_D 的错误区域包含于 R 中. 令 $P[R]$ 表示由 R 代表区域的概率质量, 即按照分布 $\mathcal{D}$ 随机生成的点落在区域 R 中的概率. 由于前面给出算法的错误仅可能是由落在 R 内的点产生, 我们不妨设 $P[R]>\epsilon$, 否则, 不管输入何种的训练样本集 D, R_D 的错误率都不会超过 ϵ.

既然 $P[R]>\epsilon$, 我们可以沿着 R 的四条边, 定义 4 个矩形区域, $r_1.r_2, r_3, r_4$, 每个区域都有概率质量 $\epsilon/4$, 如图 3所示. 注意到, 如果 R_D 与全部的四个区域都相交, 则由于它为矩形, 对每个区域中它都有一条边落在区域内部. 由此, R_D 的错误区域被这 4 个区域包含, 这样概率质量不会超过 ϵ. 于是, 如

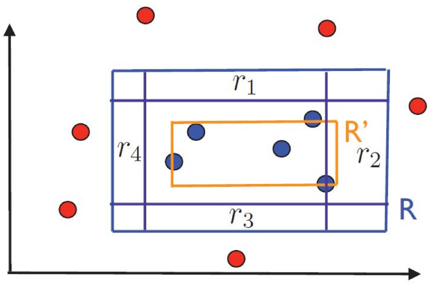

图 3: 区域 $r_1,\ldots,r_4$ 的示意图

若泛化误差 $E(R_D)>\epsilon$, 则 R_D 必然至少与 4 个区域中的某一个不相交. 由此:

$$\begin{aligned} P_{D\sim\mathcal{D}^m}[E(R_D)>\epsilon] &\le P_{D\sim\mathcal{D}^m}[\cup_{i=1}^4\{R_D\cap r_i=\emptyset\}] \\ &\le \sum_{i=1}^4 Pr_{D\sim\mathcal{D}^m}[\{R_s\cap r_i=\emptyset\}] \\ &\le 4(1-\epsilon/4)^m \\ &\le 4\exp(-m\epsilon/4). \end{aligned} \tag{24}$$

在上式最后一步中, 用到了不等式 $1-x\le\exp(-x)$ 对任意 $x\in\mathbb{R}$ 成立. 对任意 $\delta>0$, 为了确保 $P_{D\sim\mathcal{D}^m}[E(R_D)>\epsilon]\le\delta$, 可令 $4\exp(-\epsilon m/4)\le\delta$, 从而推出

$$m\ge\frac{4}{\epsilon}\log(\frac{4}{\delta}).$$

因此, 对于任意的 $\epsilon>0$ 和 $\delta>0$, 如果样本集的大小 m 不小于 $\frac{4}{\epsilon}\log\frac{4}{\delta}$, 则 $P_{D\sim\mathcal{D}^m}[E(R_D)>\epsilon]\le\delta$. 进一步, $\mathbb{R}^2$ 中点的表示的计算代价和轴平行矩形的计算代价 (可由四个角点确定) 均为常数. 到此我们证明了轴平行矩形这个概念类是 PAC 可学习的, 并且其样本复杂度是 $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$

另一种可以用来表示样本复杂度的等价方式是给出一个泛化界 (generalization bound), 其表述为, 以不小于 $1-\delta$ 的概率, 由大小为 m 的样本集可以获得某个泛化错误率上界为 $E(R_D)$ 的分类器. 为了获得此上界, 只要将 δ 设置为等于式 (24) 中给出的上界, 即 $\delta=4\exp(-m\epsilon/4)$, 并由此解出 ϵ. 由此得到, 以不小于 $1-\delta$ 的概率, 算法获取分类器的错误率上界为:

$$E(R_D)\le\frac{4}{m}\log\frac{4}{\delta}. \tag{25}$$

对于此例子, 也可以考虑其他的 PAC 可学习算法, 例如可以考虑返回不包含负例的最大的轴平行矩形. 该算法也是 PAC 可学习的, 证明过程可以直接参照上面进行.

注意, 此例考虑的假设空间 H 恰好与概念类 C 一致, 并且集合 H 的基数是无穷大, 然而关于其 PAC 可学习性却有一种非常简单的证明. 我们可能会问, 这样简单的证明是否可以推广到其他相似的概念类? 不幸的是, 直接推广难以实现, 因为该证明过程中用到了最为关键的几何讨论, 要把这种证明推广到其他场合并没有那么显然, 需要发展更加一般的证明技巧以及更一般的结论.

前面提到, PAC 学习中一个关键因素是假设空间 $\mathcal{H}$ 的复杂度. 对于假设空间, 我们可能会想, 既然要得到目标概念, 为何不像上例一样直接让假设空间与概念类完全相同, 即 $\mathcal{H}=\mathcal{C}$? 这种想法虽好, 但

却不总是可行, 因为在现实应用中我们对概念类 $\mathcal{C}$ 通常一无所知, 更别说获得一个假设空间与概念类恰好相同的学习算法. 同时注意到, 假设空间是可以人为控制大小的, 一般而言, $\mathcal{H}$ 越大, 其包含的目标概念的可能性也越大, 但从中找到某个具体目标概念的难度也越大. 在下一节将会介绍" 有限假设空间" 的 PAC 学习理论, 将会看到, PAC 学习框架对有限假设空间中的可学习性给出了较好的理论保证.

注: 关于机器学习理论方面的书近些年来有 [Mohri et al., 2012] 和 [Shalev-Shwartz and Ben-David, 2014]. 本章中例 2 来自于 [Mohri et al., 2012] 的第 1.2 节.

参考文献

Abbasi-yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320.

Abernethy, J., Bartlett, P. L., Rakhlin, A., and Tewari, A. (2008a). Optimal stragies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 415–423.

Abernethy, J., Hazan, E., and Rakhlin, A. (2008b). Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning*, pages 263–274.

Agarwal, A., Dekel, O., and Xiao, L. (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Proceedings of the 23rd Annual Conference on Learning Theory*, pages 28–40.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

Bartlett, P. L., Bousquet, O., and Mendelson, S. (2005). Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537.

Bartlett, P. L., Dani, V., Hayes, T. P., Kakade, S. M., Rakhlin, A., and Tewari, A. (2008). High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning*, pages 335–341.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2:499–526.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

Cesa-bianchi, N., Conconi, A., and Gentile, C. (2002). On the generalization ability of on-line learning algorithms. In *Advances in Neural Information Processing Systems 14*, pages 359–366.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008a). The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20*, pages 345–352.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008b). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning*, pages 355–366.

Daniely, A., Gonen, A., and Shalev-Shwartz, S. (2015). Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1405–1411.

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23*, pages 586–594.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix computations, 3rd Edition*. Johns Hopkins University Press.

Hazan, E., Agarwal, A., and Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192.

Hazan, E. and Kale, S. (2011). Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436.

Hazan, E. and Kale, S. (2014). Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15:2489–2512.

Hazan, E. and Seshadhri, C. (2007). Adaptive algorithms for online decision problems. *Electronic Colloquium on Computational Complexity*, 88.

Hazan, E. and Seshadhri, C. (2009). Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 393–400.

Hou, B.-J., Zhang, L., and Zhou, Z.-H. (2017). Learning with feature evolvable streams. In *Advances in Neural Information Processing Systems 30*.

Jun, K.-S., Bhargava, A., Nowak, R., and Willett, R. (2017). Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems 30*, pages 99–109.

Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. (2008). Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine Learning*, pages 440–447.

Kakade, S. M. and Tewari, A. (2009). On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems 21*, pages 801–808.

Koltchinskii, V. (2011). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*. Springer.

Kushner, H. J. and Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, second edition.

Mahdavi, M., Zhang, L., and Jin, R. (2015). Lower and upper bounds on the generalization of stochastic exponentially concave optimization. In *Proceedings of the 28th Annual Conference on Learning Theory*.

Mohri, M., Rostamizadeh, A., and Talwalkar, A., editors (2012). *Foundations of Machine Learning*. MIT Press, Cambridge, MA.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied optimization*. Kluwer Academic Publishers.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. Core discussion papers.

Nesterov, Y. (2011). Random gradient-free minimization of convex functions. Core discussion papers.

Rakhlin, A., Shamir, O., and Sridharan, K. (2012). Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning*, pages 449–456.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

Saha, A. and Tewari, A. (2011). Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 636–642.

Sauer, N. (1972). On the density of families of sets. *Journal of Combinatorial Theory - Series A*, 13(1):145–147.

Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2009a). Stochastic convex optimization. In *Proceedings of the 22nd Annual Conference on Learning Theory*.

Shalev-Shwartz, S., Shamir, O., Sridharan, K., and Srebro, N. (2009b). Learnability and stability in the general learning setting.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: primal estimated sub-gradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814.

Shelah, S. (1972). A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261.

Srebro, N., Sridharan, K., and Tewari, A. (2010). Optimistic rates for learning with a smooth loss. *ArXiv e-prints*, arXiv:1009.3896.

Sridharan, K., Shalev-shwartz, S., and Srebro, N. (2009). Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems 21*, pages 1545–1552.

Tseng, P. (2008). On acclerated proximal gradient methods for convex-concave optimization. Technical report, University of Washington.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vapnik, V. N. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280.

Yang, T., Zhang, L., Jin, R., and Yi, J. (2016). Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 449–457.

Zhang, L., Yang, T., and Jin, R. (2017). Empirical risk minimization for stochastic convex optimization: $O(1/n)$- and $O(1/n^2)$-type of risk bounds. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 1954–1979.

Zhang, L., Yang, T., Jin, R., Xiao, Y., and Zhou, Z.-H. (2016). Online stochastic linear optimization under one-bit feedback. In *Proceedings of the 33rd International Conference on Machine Learning*.

Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2018). Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936.