4 复杂度

4.1 引言

上一章介绍了 PAC 学习框架, 该框架刻画了假设空间的可学习性. 从 PAC 可学习的定义可以发现, 假设空间的可学习性与样本复杂度有关. 通常样本复杂度的定义如下:

定义 4.1. 样本复杂度 *满足 (不可知)PAC 学习算法 $\mathfrak{L}$ 所需要的 $m \geq poly(1/\epsilon, 1/\delta, size(\boldsymbol{x}), size(c))$ 中最小的 m, 称为学习算法 $\mathfrak{L}$ 的样本复杂度.*

假定学习算法 $\mathfrak{L}$ 处理每个样本的时间为常数, 则 $\mathfrak{L}$ 的时间复杂度等价于样本复杂度. 于是, 对时间复杂度的关心就转化为对样本复杂度的关心.

影响样本复杂度的重要因素之一就是假设空间的复杂程度. 直观上来说, 假设空间越大, 找到目标概念的近似的难度也越大, 学习算法需要的样本数目可能也就越多, 即样本复杂度越大. 假设空间可分为有限假设空间和无限假设空间. 对于有限假设空间, 一个很简单的方法就是通过其中包含假设的数量 $|\mathcal{H}|$ 来衡量它其复杂度; 对于无限假设空间则要困难的多, 需要引入 VC 维以及 Rademacher 复杂度的概念.

本章组织结构如下: 4.2节介绍假设空间的 VC 维, 4.3节介绍函数空间的 Rademacher 复杂度, 4.4节介绍一些假设空间或函数空间实例的复杂度.

4.2 VC 维

现实学习任务中的假设空间通常是无限假设空间, 例如实数域中的区间, $\mathbb{R}^d$ 空间中的所有的线性超平面. 为了对这些无限假设空间进行研究, 通常考虑其 VC 维. 在介绍 VC 维之前, 先引入几个概念: 假设空间 $\mathcal{H}$ 在数据集 D 上的限制、增长函数、对分、打散

4.2.1 VC 维的定义

定义 4.2. 假设空间 $\mathcal{H}$ 在数据集 D 上的限制: *令 $\mathcal{H}$ 表示假设空间, 其中的假设是 $\mathcal{X}$ 到 $\mathcal{Y} = \{0, 1\}$ 的映射, 数据集 $D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subset \mathcal{X}$, 则 $\mathcal{H}$ 在数据集 D 上的限制是从 D 到 $\{0, 1\}^m$ 的一族映射:*

$$\mathcal{H}_{|D} = \{(h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_m)) : h \in \mathcal{H}\}$$

其中 h 在 D 上的限制是一个 m 维向量.

定义 4.3. 增长函数: *对于 $m \in \mathbb{N}$, 假设空间 $\mathcal{H}$ 的增长函数 $\Pi_{\mathcal{H}}(m)$ 为*

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

二分类问题中有:

$$\Pi_{\mathcal{H}}(m) = \max_{|D|=m} |\mathcal{H}_{|D}|$$

增长函数 $\Pi_{\mathcal{H}}(m)$ 表示假设空间 $\mathcal{H}$ 对 m 个示例所能赋予标记的最大可能的结果数. 显然, $\mathcal{H}$ 对示例所能赋予标记的可能结果数越大, $\mathcal{H}$ 的表示能力越强, 对学习任务的适应能力也很强. 因此增长函数描述了假设空间 $\mathcal{H}$ 的表示能力, 由此反映出假设空间的复杂程度.

假设空间 $\mathcal{H}$ 中不同的假设对于 D 中示例赋予标记的结果可能相同, 也可能不同; 尽管 $\mathcal{H}$ 可能包含无穷多个假设, 但是 $\mathcal{H}_{|D}$ 却是有限的, 即 $\mathcal{H}$ 对 D 中示例赋予标记的可能结果数是有限的. 对于 m 个示例, 最多有 2^m 个可能的结果 (二分类问题), 这是一种将无限转化为有限的思想.

定义 4.4. 打散: 对于二分类问题来说, 假设空间 $\mathcal{H}$ 中的假设对 D 中的示例赋予标记的每种可能结果称为对 D 的一种"对分", 或者可以把 $\mathcal{H}_{|D}$ 中每一个向量视为 $\mathcal{H}$ 中的假设对 D 的一个"对分". 如果假设空间 $\mathcal{H}$ 能实现示例集 D 上的所有对分, 即 $|\mathcal{H}_{|D}| = 2^m$, 则称示例集 D 能被假设空间 $\mathcal{H}$ 打散, 此时 $\Pi_{\mathcal{H}}(m) = 2^m$.

例如, $\mathcal{H}$ 表示 $\mathbb{R}$ 上的的阈值函数构成的集合, 其中的阈值函数表示为: $h_a = \mathbb{I}_{\{x<a\}}$, 则 $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$. 令 $D = \{x_1\}$, 如果取 $a = x_1 + 1$, 则 $h_a(x_1) = 1$, 如果取 $a = x_1 - 1$, 则 $h_a(x_1) = 0$, 因此 $\mathcal{H}$ 能打散 $D = \{x_1\}$. 令 $D' = \{x_1, x_2\}$, 不妨假设 $x_1 \le x_2$, 则易知同时将 x_1 分类为 0 但把 x_1 分类为 1 的结果不能被 $\mathcal{H}$ 中的任何阈值函数实现, 这是因为如果 $h_a(x_1) = 0$, 则必有 $h_a(x_2) = 0$. 所以 $\mathcal{H}$ 不能打散 D'.

定义 4.5. *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:

$$VC(\mathcal{H}) = \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}$$

$VC(\mathcal{H}) = d$ 表明存在大小为 d 的示例集能被假设空间 $\mathcal{H}$ 打散. 注意: 这并不意味着所有大小为 d 的示例集都能被假设空间 $\mathcal{H}$ 打散. VC 维的定义与数据分布 $\mathcal{D}$ 无关! 因此, 在数据分布未知时仍能计算出假设空间 $\mathcal{H}$ 的 VC 维.

4.2.2 有限假设空间的 VC 维

令假设空间 $\mathcal{H}$ 为有限集合. 对于任意数据集 D, $|\mathcal{H}_{|D}| \le \mathcal{H}$. 除此之外, 还可知当 $|\mathcal{H}| < 2^{|D|}$ 时, $\mathcal{H}$ 无法打散 D. 因此, 可得 $VC(\mathcal{H}) \le \log_2(|\mathcal{H}|)$. 这表明有限 VC 维假设空间相对于有限维假设空间的条件更为宽松, 即有限 VC 维空间有更广泛的适用性. 除此之外, 对于有限空间 $\mathcal{H}$, $VC(\mathcal{H})$ 通常可以远小于 $\log_2(|\mathcal{H}|)$.

4.2.3 VC 维与增长函数的关系

由增长函数的定义可知,VC 维与增长函数关系密切, 引理 4.1给出了二者之间的定量关系 [Sauer, 1972]

引理 4.1. 若假设空间 $\mathcal{H}$ 的 VC 维为 d, 则对任意 $m \in \mathbb{N}$ 有

$$\Pi_{\mathcal{H}}(m) \le \sum_{i=0}^{d} \binom{m}{i}.$$

证明. 利用数学归纳法证明. 当 $m=1$, $d=0$ 或 $d=1$ 时, 定理成立. 假设定理对 $(m-1,d-1)$ 和 $(m-1,d)$ 成立, 令 $D=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$, $D'=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{m-1}\}$,

$$\mathcal{H}_{|D}=\{(h(\boldsymbol{x}_1),\ldots,h(\boldsymbol{x}_m))|h\in\mathcal{H}\},$$

$$\mathcal{H}_{|D'}=\{(h(\boldsymbol{x}_1),\ldots,h(\boldsymbol{x}_{m-1}))|h\in\mathcal{H}\},$$

分别为假设空间在 D 和 D' 上的限制. 任何假设 $h\in\mathcal{H}$ 对 $\boldsymbol{x}_m$ 的分类结果为 $+1$ 或 -1, 因此任何出现在 $\mathcal{H}_{|D'}$ 的串都会在 $\mathcal{H}_{|D}$ 出现一次或者两次. 令 $\mathcal{H}_{D'|D}$ 表示 $\mathcal{H}_{|D}$ 中出现两次的 $\mathcal{H}_{|D'}$ 中串组成的集合, 即

$$\begin{aligned}\mathcal{H}_{D'|D}=\{&(y_1,\ldots,y_{m-1})\in\mathcal{H}_{|D'}|\exists h,h'\in\mathcal{H},\\&(h(\boldsymbol{x}_i)=h'(\boldsymbol{x}_i)=y_i)\wedge(h(\boldsymbol{x}_m)\neq h'(\boldsymbol{x}_m)),1\leq i\leq m-1\}\end{aligned}$$

考虑到 $\mathcal{H}_{D'|D}$ 中的串在 $\mathcal{H}_{|D}$ 中出现了两次, 但在 $\mathcal{H}_{|D'}$ 中仅出现了一次, 有

$$|\mathcal{H}_{|D}|=|\mathcal{H}_{|D'}|+|\mathcal{H}_{D'|D}|$$

D' 的大小为 $m-1$, 由假设可得

$$|\mathcal{H}_{|D'}|\leq\Pi_{\mathcal{H}}(m-1)\leq\sum_{i=0}^{d}\binom{m-1}{i}$$

令 Q 表示能被 $\mathcal{H}_{D'|D}$ 打散的集合, 由 Q 的定义可知 $Q\cup\{\boldsymbol{x}_m\}$ 必能被 $\mathcal{H}_{|D}$ 打散. 由于 $\mathcal{H}$ 的 VC 维为 d, 因此 $\mathcal{H}_{D'|D}$ 的 VC 维最大为 d-1, 于是有

$$|\mathcal{H}_{D'|D}|\leq\Pi_{\mathcal{H}}(m-1)\leq\sum_{i=0}^{d-1}\binom{m-1}{i}$$

综合上面几个不等式可得

$$\begin{aligned}|\mathcal{H}_{|D}|&\leq\sum_{i=0}^{d}\binom{m-1}{i}+\sum_{i=0}^{d-1}\binom{m-1}{i}\\&=\sum_{i=0}^{d}\left(\binom{m-1}{i}+\binom{m-1}{i-1}\right)\\&=\sum_{i=0}^{d}\binom{m}{i}\end{aligned}$$

由 D 的任意性, 定理得证. ■

由引理 4.1可以计算出增长函数的上界.

定理 4.1. ***Sauer* 定理**: *若假设空间 $\mathcal{H}$ 的 VC 维为 d, 则对任意整数 $m\geq d$ 有*

$$\Pi_{\mathcal{H}}(m)\leq\left(\frac{e\cdot m}{d}\right)^{d}.$$

证明.

$$\begin{aligned}
\Pi_{\mathcal{H}}(m) &\leq \sum_{i=0}^{d} \binom{m}{i} \\
&\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&= \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{d} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \\
&\leq \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \\
&= \left(\frac{m}{d}\right)^{d} \left(1+\frac{d}{m}\right)^{m} \\
&\leq \left(\frac{e \cdot m}{d}\right)^{d}
\end{aligned}$$

■

当假设空间 $\mathcal{H}$ 的 VC 维为无穷大时, 任意大小的示例集 D 都能被 $\mathcal{H}$ 打散, 此时有 $\Pi_{\mathcal{H}(m)} = 2^m$, 增长函数是随着数据集的大小指数级增长; 当 VC 维有限为 d 且 $m \geq d$ 时, 由上述定理可知增长函数随数据集的大小多项式级增长.

4.3 Rademacher 复杂度

之前提到 VC 维的定义与数据分布无关的, 所以基于 VC 维的泛化误差界是分布无关、数据独立的, 也就是说对任何数据分布都成立. 这使得基于 VC 维的可学习性分析结果具有一定的" 普适性"; 但从另一方面来说, 由于没有考虑数据自身, 基于 VC 维得到的泛化误差界通常比较" 松", 对那些与学习问题的典型情况相差甚远的较" 坏" 分布来说尤其如此. Rademacher 复杂度是另外一种刻画假设空间复杂度的途径, 与 VC 维不同的是, 它一定程度上考虑了数据分布. 先考虑给定数据集 $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, 假设 h 的经验误差为:

$$\begin{aligned}
\hat{E}(h) &= \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(h(\boldsymbol{x}_i) \neq y_i) \\
&= \frac{1}{m} \sum_{i=1}^{m} \frac{1 - y_i h(\boldsymbol{x}_i)}{2} \\
&= \frac{1}{2} - \frac{1}{2m} \sum_{i=1}^{m} y_i h(\boldsymbol{x}_i)
\end{aligned}$$

其中 $\frac{1}{m}\sum_{i=1}^{m} y_i h(\boldsymbol{x}_i)$ 体现了预测值 $h(\boldsymbol{x}_i)$ 与样例真实标记 y_i 之间的一致性, 若 $h(\boldsymbol{x}_i) = y_i \ \forall i \in [m]$, 则 $\frac{1}{m}\sum_{i=1}^{m} y_i h(\boldsymbol{x}_i)$ 取得最大值 1, 也就是说具有最小经验误差的假设是:

$$\arg\max_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} y_i h(\boldsymbol{x}_i)$$

然而, 现实任务中样例的标记有时会受到噪声的影响, 即对某些样例 $(\boldsymbol{x}_i, y_i)$, 其 y_i 或许已经受到随机因素的影响, 不再是 $\boldsymbol{x}_i$ 的真实标记. 在此情形下, 选择假设空间 $\mathcal{H}$ 中在训练集上表现最好的假设, 有时还不如选择 $\mathcal{H}$ 中事先已考虑了随机噪声影响的假设.

考虑随机变量 σ_i, 它以 0.5 的概率取值 -1, 以 0.5 的概率取值 +1, 称为 Rademacher 随机变量. 基于 σ_i, 可用下式衡量假设空间复杂度:

$$\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(\boldsymbol{x}_i).$$

对所有的 σ_i 求期望有:

$$\mathbb{E}_\sigma[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(\boldsymbol{x}_i)].$$

此式和增长函数有着相似的作用, 表现了假设空间在数据集 D 上的表示能力. 上式的取值范围为 $[0,1]$. 当上式取值为 1 时, 意味着对任意 $\sigma=\{\sigma_1,\ldots,\sigma_m\}$, 其中 $\sigma_i\in\{-1,+1\}$, 有

$$\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(\boldsymbol{x}_i) = 1.$$

在有限假设空间的情况下, 即 $\exists\ h\in\mathcal{H},\ s.t.\ h(\boldsymbol{x}_i)=\sigma_i\ \forall i\in[m]$, 类似于 $|\mathcal{H}_{|D}|=2^m$, 也就有 $\Pi_{\mathcal{H}}(m)=2^m$. 根据之前的定义, $\mathcal{H}$ 能打散 D. 总的来说上式值越接近 1, 假设空间的表达能力越强. 考虑实值函数空间 $\mathcal{F}:\mathcal{Z}\to\mathbb{R}$, 令 $Z=\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_m\}$, 其中 $\boldsymbol{z}_i\in\mathcal{Z}$, 将原式中 $\mathcal{X}$ 和 $\mathcal{H}$ 替换为 $\mathcal{Z}$ 和 $\mathcal{F}$ 可得

定义 4.6. *函数空间 $\mathcal{F}$ 关于 Z 的经验 Rademacher 复杂度为*

$$\hat{R}_Z(\mathcal{F}) = \mathbb{E}_\sigma\Big[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i f(\boldsymbol{z}_i)\Big].$$

注意此处的 Z 是一个给定的集合, 经验 Rademacher 复杂度衡量了函数空间 $\mathcal{F}$ 与随机噪声在数据集 Z 中的相关性. 除了固定的数据集 Z, 通常更加关心 Z 服从分布 $\mathcal{D}$ 时的性质复杂度. 对从分布 $\mathcal{D}$ 上独立同分布采样得到的大小为 m 的集合 Z 求期望得:

定义 4.7. *函数空间 $\mathcal{F}$ 关于 $\mathcal{Z}$ 在分布 $\mathcal{D}$ 上的 Rademacher 复杂度为*

$$R_m(\mathcal{F}) = \mathbb{E}_{Z\subset\mathcal{Z}:|Z|=m}\Big[\hat{R}_Z(\mathcal{F})\Big].$$

这里需要注意到, 在 Rademacher 复杂度的定义中 σ 是 $\{-1,+1\}$ 上服从均匀分布的随机变量, 如果将均匀分布改为其他分布, 会得到其他一些复杂度的定义, 例如 Gauss 复杂度.

定义 4.8. *函数空间 $\mathcal{F}$ 关于 Z 的经验 Gauss 复杂度为*

$$\hat{G}_{\mathcal{Z}}(\mathcal{F}) = \mathbb{E}_g\Big[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} g_i f(\boldsymbol{z}_i)\Big],$$

其中 g 服从高斯 $N(0,1)$ 分布, 即标准正态分布.

同样地, 可以对经验 Gauss 复杂度求期望得到 Gauss 复杂度.

定义 4.9. *函数空间 $\mathcal{F}$ 关于 $\mathcal{Z}$ 在分布 $\mathcal{D}$ 上的 Gauss 复杂度为*

$$G_m(\mathcal{F}) = \mathbb{E}_{Z\subset\mathcal{Z}:|Z|=m}\Big[\hat{G}_Z(\mathcal{F})\Big],$$

其中 g 服从高斯 $N(0,1)$ 分布, 即标准正态分布.

4.4 实例

本节将介绍一些假设空间的 VC 维或函数空间的 Rademacher 复杂度.

4.4.1 假设空间的 VC 维

要证明假设空间 $\mathcal{H}$ 的 VC 维为 d 时, 需要证明两点:

- 存在大小为 d 的示例集 D 能被 $\mathcal{H}$ 打散;
- 任意大小为 $d+1$ 的示例集 D' 都不能被 $\mathcal{H}$ 打散.

阈值函数的 VC 维

令 $\mathcal{H}$ 表示所有定义在 $\mathbb{R}$ 上的阈值函数组成的集合, 由上述的分析可知存在大小为 1 的示例集 D 能被 $\mathcal{H}$ 打散, 但任意大小为 2 的示例集 $|D'|$ 都不能被 $\mathcal{H}$ 打散, 根据定义可知 $VC(\mathcal{H}) = 1$.

区间函数的 VC 维

令 $\mathcal{H}$ 表示所有定义在 $\mathbb{R}$ 上的区间组成的集合 $\mathcal{H} = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}$, 其中 $h_{a,b}(x) = \mathbb{I}_{\{x\in(a,b)\}}$ 为定义在开区间上的指示函数. 令 $D = \{1, 2\}$, 易知 $\mathcal{H}$ 能打散 D, 因此 $VC(\mathcal{H}) \geq 2$. 对于任意大小为 3 的示例集 $D' = \{x_1, x_2, x_3\}$, 不妨设 $x_1 \leq x_2 \leq x_3$, 则分类结果 $(1, 0, 1)$ 不能被 $\mathcal{H}$ 中的任何区间函数实现, 因为当 $h_{a,b}(x_1) = 1$ 且 $h_{a,b}(x_3) = 1$ 时, 必有 $h_{a,b}(x_2) = 1$. 所以 $\mathcal{H}$ 无法打散任何大小为 3 的示例集, 即得出结论 $VC(\mathcal{H}) = 2$.

线性超平面的 VC 维

对于二分类问题, 线性超平面假设空间 $\mathcal{H}$ 可以表示为

$$\Big\{h_{\boldsymbol{w},b} : h_{\boldsymbol{w},b}(\boldsymbol{x}) = \operatorname{sign}\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b\big) = \operatorname{sign}\Big(\big(\sum_{i=1}^{d} \boldsymbol{w}_i \boldsymbol{x}_i\big) + b\Big)\Big\},$$

当 $b = 0$ 时, 为齐次线性超平面.

定理 4.2. *$\mathbb{R}^d$ 空间中, 由齐次线性超平面构成假设空间的 VC 维为 d.*

证明. 首先考虑 $\mathbb{R}^d$ 中的单位向量, $D = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$, 每个单位向量 $\boldsymbol{e}_i$ 的第 i 个元素为 1, 其余元素为 0. 对于任意标记 $(y_1, \ldots, y_d)$, 取 $\boldsymbol{w} = (y_1, \ldots, y_d)$, 则有 $\langle \boldsymbol{w}, \boldsymbol{e}_i\rangle = y_i$. 所以示例集 D 能被齐次线性超平面打散.

接下来, 令 $D' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1}\}$ 为 $\mathbb{R}^d$ 中任意 $d+1$ 个向量, 则必存在不全为 0 的实数 $a_1, \ldots, a_{d+1}$, 使得 $\sum_{i=1}^{d+1} a_i \boldsymbol{x}_i = 0$. 令 $I = \{i : a_i > 0\}$, $J = \{j : a_j < 0\}$, 则 I, J 至少一个不为空集. 首先假设两者都不为空集, 则有

$$\sum_{i\in I} a_i \boldsymbol{x}_i = \sum_{j\in J} |a_j| \boldsymbol{x}_j.$$

下面采用反证法, 如果假设 D' 能被 $\mathcal{H}$ 打散, 则存在向量 $\boldsymbol{w}$ 使得 $\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0, i \in I$, 且 $\langle \boldsymbol{w}, \boldsymbol{x}_j \rangle < 0, j \in J$(这里取标记 $y_i = 1, i \in I$, $y_j = -1, j \in J$). 由此推导出

$$0 < \sum_{i \in I} a_i \langle \boldsymbol{x}_i, \boldsymbol{w} \rangle = \Big\langle \sum_{i \in I} a_i \boldsymbol{x}_i, \boldsymbol{w} \Big\rangle = \Big\langle \sum_{j \in J} |a_j| \boldsymbol{x}_j, \boldsymbol{w} \Big\rangle = \sum_{j \in J} |a_j| \langle \boldsymbol{x}_j, \boldsymbol{w} \rangle < 0.$$

此式矛盾, 即 D' 能被 $\mathcal{H}$ 打散不成立. 当 I, J 只有一个不为空集时同理可证, 综上, 得出结论 $VC(\mathcal{H}) = d$. ■

定理 4.3. $\mathbb{R}^d$ 空间中, 由非齐次线性超平面构成假设空间的 VC 维为 $d+1$.

证明. 由上例的证明可知 $D = \{0, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$ 能被非齐次线性超平面 $\mathcal{H}$ 打散. 下面将非齐次线性超平面转化为其次线性超平面:

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = \langle \boldsymbol{w}', \boldsymbol{x}' \rangle, \boldsymbol{w} \in \mathbb{R}^d, \boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{w}' \in \mathbb{R}^{d+1}, \boldsymbol{x}' \in \mathbb{R}^{d+1},$$

其中 $\boldsymbol{w}' = (\boldsymbol{w}, b)$, $\boldsymbol{x}' = (\boldsymbol{x}, 1)$. 由此, 如果 $D' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+2}\}$ 能被 $\mathbb{R}^d$ 中非齐次线性超平面打散, 则 $D'' = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{d+2}\}$ 能被 $\mathbb{R}^{d+1}$ 中齐次线性超平面打散, 这与前一定理矛盾. 因此, 非齐次线性超平面的 VC 维为 $d+1$. ■

SVM 通常是将原特征空间映射到高维空间中, 并且在高维空间中训练线性超平面进行分类. 根据定理 4.3可知. 若高维空间的维度为 d, 则该 SVM 的 VC 维为 d+1.

神经网络的 VC 维

以上定义的增长函数是对二值函数的, 即函数值取值为 $\{-1, +1\}$, 很容易将其推广到值域为有限集的函数族上, 同样将这个有限集记为 $\mathcal{Y}$, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ 即为该函数族. 现在定义该函数族上的增长函数为:

$$\Pi_{\mathcal{F}}(m) = \max_{\mathbf{x}^m \in \mathcal{X}^m} |\mathcal{F}_{|\mathbf{x}^m}|$$

其中 $\mathbf{x}^m$ 为大小为 m 独立同分布从 $\mathcal{X}$ 中采样得到示例集, 易知 $\Pi_{\mathcal{F}}(m) \leq |\mathcal{Y}|^m$.

引理 4.2. 令 $\mathcal{F}^{(1)} \subset \mathcal{Y}_1^{\mathcal{X}}$, $\mathcal{F}^{(2)} \subset \mathcal{Y}_2^{\mathcal{X}}$ 为两个函数族, $\mathcal{F} = \mathcal{F}^{(1)} \times \mathcal{F}^{(2)}$ 为它们的笛卡尔积, 有

$$\Pi_{\mathcal{F}}(m) \leq \Pi_{\mathcal{F}^{(1)}}(m) \cdot \Pi_{\mathcal{F}^{(2)}}(m)$$

证明. 固定 $\mathbf{x}^m$, 根据笛卡尔积的定义可知,

$$\begin{aligned} |\mathcal{F}_{|\mathbf{x}^m}| &= |\mathcal{F}^{(1)}_{|\mathbf{x}^m}||\mathcal{F}^{(2)}_{|\mathbf{x}^m}| \\ &\leq \Pi_{\mathcal{F}^{(1)}}(m) \cdot \Pi_{\mathcal{F}^{(2)}}(m) \end{aligned}$$

由 $\boldsymbol{x}^m$ 的任意性, 引理成立. ■

引理 4.3. 令 $\mathcal{F}^{(1)} \subset \mathcal{Y}_1^{\mathcal{X}}$, $\mathcal{F}^{(2)} \subset \mathcal{Y}_2^{\mathcal{X}}$ 为两个函数族, $\mathcal{F} = \mathcal{F}^{(2)} \circ \mathcal{F}^{(1)}$ 为它们的复合, 有

$$\Pi_{\mathcal{F}}(m) \leq \Pi_{\mathcal{F}^{(2)}}(m) \cdot \Pi_{\mathcal{F}^{(1)}}(m).$$

证明. 固定 $\mathbf{x}^m \in \mathcal{X}^m$, 根据 $\mathcal{F}$ 的定义有

$$\begin{aligned}\mathcal{F}_{|\mathbf{x}^m} &= \left\{(f_2(f_1(\boldsymbol{x}_1)), \ldots, f_2(f_1(\boldsymbol{x}_m)))|f_1 \in \mathcal{F}^{(1)}, f_2 \in \mathcal{F}^{(2)}\right\} \\ &= \bigcup_{\mathbf{u}\in\mathcal{F}^{(1)}_{|\mathbf{x}^m}} \left\{(f_2(\boldsymbol{u}_1), \ldots, f_2(\boldsymbol{u}_m))|f_2 \in \mathcal{F}^{(2)}\right\}.\end{aligned}$$

因此有

$$\begin{aligned}|\mathcal{F}_{|\mathbf{x}^m}| &\leq \sum_{\mathbf{u}\in\mathcal{F}^{(1)}_{|\mathbf{x}^m}} \left|\left\{(f_2(\boldsymbol{u}_1), \ldots, f_2(\boldsymbol{u}_m))|f_2 \in \mathcal{F}^{(2)}\right\}\right| \\ &\leq \sum_{\mathbf{u}\in\mathcal{F}^{(1)}_{|\mathbf{x}^m}} \Pi_{\mathcal{F}^{(2)}}(m) \\ &= |\mathcal{F}^{(1)}_{|\mathbf{x}^m}| \cdot \Pi_{\mathcal{F}^{(2)}}(m) \\ &\leq \Pi_{\mathcal{F}^{(2)}}(m) \cdot \Pi_{\mathcal{F}^{(1)}}(m).\end{aligned}$$

根据 $\mathbf{x}^m$ 的任意性, 引理得证. ■

一般来说, 神经网络中的一个节点 v 计算一个函数

$$\sigma(\boldsymbol{w}^{(v)} \cdot \boldsymbol{x} - \theta^{(v)}),$$

这个函数以 $\boldsymbol{x}$ 为输入, 输出激活信号, 函数 σ 被称为激活函数. 下面是一些常用的激活函数的例子.

- 二元函数: $\sigma(t) = \text{sign}(t)$;
- Sigmoid 型函数: $\sigma(t) = \frac{1}{1+e^{-t}}$;
- Sigmoid 型函数: $\sigma(t) = arctan(t)$.

这里主要考虑使用二元激活函数的多层神经网咯, 也有一些方法可以用于分析使用 Sigmoid 型激活函数神经网络的 VC 维, 此处不再赘述.

假设输入空间 $\mathcal{X} = \mathbb{R}^{d_0}$, 一个 l 层的多层网络可以简化为一系列映射的复合:

$$f_l \circ \cdots \circ f_2 \circ f_1(\boldsymbol{x}),$$

其中

$$\begin{aligned}f_i &: \mathbb{R}^{d_{i-1}} \to \{\pm 1\}^{d_i}, 1 \leq i \leq l-1 \\ f_l &: \mathbb{R}^{d_{l-1}} \to \{\pm 1\}.\end{aligned}$$

f_i 是一个多维到多维的映射, 可以将其分解为若干个二值多元函数, 对于 f_i 的每一个分量 $f_{i,j} : \mathbb{R}^{d_{i-1}} \to \{\pm 1\}$ 表示为:

$$f_{i,j}(u) = \text{sign}(\boldsymbol{w}^{i,j} \cdot \boldsymbol{u} - \theta^{i,j}),$$

其中 $\boldsymbol{w}^{i,j} \in \mathbb{R}^{d_i-1}$, $\theta^{i,j} \in \mathbb{R}$ 分别为关于第 i 层第 j 个节点的权重与偏置. 将这个多元函数的函数族记为 $\mathcal{F}^{i,j}$, 则关于第 i 层的函数族则可以简单地表示为

$$\mathcal{F}^{(i)} = \mathcal{F}^{(i,1)} \times \cdots \times \mathcal{F}^{(i,d_i)},$$

则整个网络的函数族则可以表示为

$$\mathcal{F} = \mathcal{F}^{(l)} \circ \cdots \circ \mathcal{F}^{(2)} \circ \mathcal{F}^{(1)}.$$

根据上述两个引理, Sauer 定理, 以及 d 维空间中超平面的 VC 维为 $d+1$, 可以得到如下不等式

$$\begin{aligned}\Pi_{\mathcal{F}}(m) &\leq \prod_{i=1}^{l} \Pi_{\mathcal{F}^{(i)}}(m) \\ &\leq \prod_{i=1}^{l} \prod_{j=1}^{d_i} \Pi_{\mathcal{F}^{(i,j)}}(m) \\ &\leq \prod_{i=1}^{l} \prod_{j=1}^{d_i} \left(\frac{me}{d_{i-1}+1}\right)^{d_{i-1}+1}.\end{aligned}$$

定义:

$$N = \sum_{i=1}^{l} \sum_{j=1}^{d_i} (d_{i-1}+1)$$

为整个网络的参数数目, 可以将上述不等式简化为

$$\Pi_{\mathcal{F}}(m) \leq (me)^N.$$

接下来就可以算出 $\mathcal{F}$ 的 VC 维的界:

定理 4.4. *令 $\mathcal{F}$ 表示计算上述多层神经网络的函数族, 则 $VC(\mathcal{F}) = O(N \log_2(N))$*

证明. 令能被 $\mathcal{F}$ 打散的最大示例集大小为 m, 则 $\Pi_{\mathcal{F}}(m) = 2^m$, 结合上述不等式可得

$$2^m \leq (me)^N,$$

为了使等式成立, 必然有 $m = O(N \log_2(N))$. ∎

AdaBoost 的 VC 维

Boosting 起源于 1989 年计算机理论科学 STOC 会议中强学习与弱学习之间是否存在等价性问题. 给定输入空间 $\mathcal{X}$ 和输出空间 $\mathcal{Y} = \{+1, -1\}$, 假设 $\mathcal{D}$ 为空间 $\mathcal{X}$ 上潜在未知分布函数, 函数 $f\colon \mathcal{X} \to \mathcal{Y}$ 为所学习的目标函数. 一个训练集由 m 个样本组成

$$S = \{(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_m, f(x_m))\},$$

其中每个样本是根据分布 $\mathcal{D}$ 独立同分布采样所得. 函数空间 $\mathcal{H} = \{h\colon \mathcal{X} \to \mathcal{Y}\}$, 学习过程则是根据训练集 S 产生一个假设 $h \in \mathcal{H}$，使其最小化错误率

$$P_{x\sim\mathcal{D}}\left(I(f(x) \neq h(x))\right).$$

一个学习问题是弱可学习的是指存在一种算法, 对任意给定的分布 $\mathcal{D}$, 都能在关于 m 和 d 的多项式时间内学习得到一个分类器 h 满足

$$P_{(x,y)\sim\mathcal{D}}(I(h(x)\neq f(x)))<\frac{1}{2}-\frac{1}{\mathrm{poly}(m,d)},$$

这里 $\mathrm{poly}(m,d)$ 是关于训练集大小 m 和维度 d 的多项式.

一个学习问题是强可学习的是指对任意给定的分布 $\mathcal{D}$, 任意给定 $\epsilon>0$ 和 $\delta>0$, 存在一种关于 m 和 d 的多项式时间学习算法, 学习得到一个分类器 h 满足

$$P_{S\sim\mathcal{D}^m}(P_{x\sim\mathcal{D}}(I(h(x)\neq f(x)))>\epsilon)\leq\delta.$$

从直观上看, 弱学习器指在多项式时间内存在一个学习器, 其错误率略低于 50%, 即存在比随机猜略好的学习方法; 强可学习器指在多项式时间内存在高精度学习器. Kearns & Valiant 在 1989 年 STOC 会议论文中提出一个重要的理论问题: 弱学习算法与强学习算法之间是否存在等价性? 如果两者等价, 那么只要找到一些比随机猜略好的弱学习算法就可以直接将其提升为强学习算法, 而不必直接去找通常情况下很难获得的强学习算法.

Schapire 于 1990 年对这个问题通过构造性证明给出了肯定性回答, 而这个证明就是最初的 boosting 算法, Freund 于 1995 年提出一种更高效的算法. 这两种算法在解决实际问题时存在一个问题, 即需要事先知道弱学习算法在数据分布 $\mathcal{D}$ 中学习正确率的下限. 然而在实际应用中, 仅仅知道训练数据集, 而不知道数据的整体分布, 前面所提两种算法不能用于实际问题, 仅存在理论意义.

第一个可用于实际问题的 Boosting 算法是 Freund & Schapire 于 1995 年提出的 AdaBoost (Adaptive Boosting) 算法, 如 Algorithm 1所示, 该算法不需要任何关于弱学习器在真实分布中正确率等先验信息, 可直接应用于各种实际学习问题.

AdaBoost 算法可描述为: 在每一轮迭代中, 产生一个弱学习器 h_t, 该学习器比随机猜算法的性能略好. 然后根据产生的弱学习器 h_t, 每一个样本被赋予一个权重, 如果某样本点已经被准确地分类, 那么它的权重越小, 表明在下一论训练被选中的概率越低; 相反, 如果某样本点没有被准确地分类, 那么它的权重就会得到提高, 即根据分类器来提高被它错分的样本权重, 并降低被正确分类的样本权重. 然后更新过样本集并用于下一轮的学习训练, 整个训练过程如此迭代地进行下去, 最后将所有得到的弱分类器通过权重结合起来.

AdaBoost 算法有很多种推导, 比较容易理解的是基于 “加性模型”(additive model), 即基学习器的线性组合

$$H(x)=\sum_{t=1}^{T}\alpha_t h_t(x),$$

优化指数损失函数 (exponential loss)

$$\ell_{\exp}(H|\mathcal{D})=E_{x\sim\mathcal{D}}(e^{-f(x)H(x)}).$$

Algorithm 1 AdaBoost 算法

输入: 训练数据集 $S = \{(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_m, f(x_m))\}$ 和迭代轮数 T

初始化: $\mathcal{D}_1(x_i) = 1/m$

1: **for** $t = 1$ to T **do**
2: 根据分布 $\mathcal{D}_t$ 构造基学习器 $h_t\colon \mathcal{X} \to \mathcal{Y}$
3: 计算基学习器 h_t 的错误率 $\epsilon_t = P_{x\sim\mathcal{D}_t}(f(x) \neq h_t(x))$
4: **if** $\epsilon_t > 1/2$ **then**
5: 算法停止 %%% 基学习器比随机猜性能还差
6: **end if**
7: 计算学习器 h_t 权重 $\alpha_t = \frac{1}{2}\ln((1-\epsilon_t)/\epsilon_t)$
8: 根据基学习器预测能力更新分布

$$\mathcal{D}_{t+1}(x_i) = \mathcal{D}_t(x_i)\exp(-\alpha_t f(x_i)h_t(x_i))/Z_t$$

其中 Z_t 是规范化因子, 保证 D_{t+1} 是一个分布.

9: **end for**

输出: 最后的分类器 $\mathrm{sgn}(H(x))$, 其中 $\mathrm{sgn}(\cdot)$ 表示符号函数, 以及

$$H(x) = \sum_{t=1}^{T}\alpha_t h_t(x).$$

在 AdaBoost 算法中, 第一个基分类器 h_1 是将基学习器算法用于初始数据训练数据得到的; 此后迭代生成 h_t 和 α_t, 当基分类器 h_t 基于分布 $\mathcal{D}_t$ 产生后, 该基分类器的权重 α_t 应使得 $\alpha_t h_t$ 最小化损失函数

$$\begin{aligned}
\ell_{\exp}(\alpha_t h_f|\mathcal{D}_t) &= E_{x\sim\mathcal{D}_t}(e^{-f(x)\alpha_t h_t(x)}) \\
&= E_{x\sim\mathcal{D}_t}(e^{-\alpha_t}I(f(x) = h_t(x))) + E_{x\sim\mathcal{D}_t}(e^{\alpha_t}I(f(x) \neq h_t(x))) \\
&= e^{-\alpha_t}E_{x\sim\mathcal{D}_t}(I(f(x) = h_t(x))) + e^{\alpha_t}E_{x\sim\mathcal{D}_t}(I(f(x) \neq h_t(x))) \\
&= (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t}
\end{aligned}$$

其中 $\epsilon_t = E_{x\sim\mathcal{D}_t}(I(f(x) \neq h_t(x))) = P_{x\sim\mathcal{D}_t}(f(x) \neq h_t(x))$. 对上面指数函数求导, 并令其导数为零, 即

$$\frac{d\ell_{\exp}(\alpha_t h_f|\mathcal{D}_t)}{d\alpha_t} = -(1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = 0,$$

求解可得

$$\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

这正是 Algorithm 1中第 7 行的分类器权重更新公式.

下面的定理表明 AdaBoost 算法能在训练集上将一系列弱学习器提升为一强学习器:

定理 4.5. *假设 AdaBoost 算法每轮迭代过程中基学习器产生的错误率分别为 $\epsilon_1, \epsilon_2, \ldots, \epsilon_T$, 则最终产*

生的学习器 $H(x)=\sum_{t=1}^{T}\alpha_t h_t(x)$ 满足

$$\frac{1}{m}\sum_{i=1}^{m} I(sgn(H(x_i))\neq f(x_i)) \leq 2^T \prod_{t=1}^{T}\sqrt{\epsilon_t(1-\epsilon_t)}.$$

如果存在 $\gamma\in(0,1/2)$, 使得错误率 $\epsilon_i<1/2-\gamma$ $(i\in(T))$ 成立, 则有

$$\frac{1}{m}\sum_{i=1}^{m} I(sgn(H(x_i))\neq f(x_i)) \leq (1-4\gamma^2)^{T/2} \leq e^{-2\gamma^2 T}.$$

因此, 只要 AdaBoost 在每次迭代过程中比随机猜略好, 即 $\epsilon_i<1/2-\gamma$, 则训练错误率随迭代轮数的增加而呈指数性下降, 从而构造出一个强学习器.

证明. 在训练过程中, 初始分布 $\mathcal{D}_1=1/m$, 以及在第 t 轮迭代中

$$\mathcal{D}_{t+1}(x_i)=\mathcal{D}_t(x_i)\exp(-\alpha_t f(x_i)h_t(x_i))/Z_t. \tag{26}$$

这里 Z_t 是使 $\mathcal{D}_{t+1}(x_i)$ 成为分布的正规化项, 即

$$Z_t=\sum_{i=1}^{m}\mathcal{D}_t(x_i)\exp(-\alpha_t f(x_i)h_t(x_i)).$$

由 $\alpha_t=\frac{1}{2}\ln((1-\epsilon_t)/\epsilon_t)$ 可知

$$\begin{aligned} Z_t &= e^{-\alpha_t}\sum_{i=1}^{m}\mathcal{D}_t(x_i)I(f(x_i)=h_t(x_i)) + e^{\alpha_t}\sum_{i=1}^{m}\mathcal{D}_t(x_i)I(f(x_i)\neq h_t(x_i)) \\ &= (1-\epsilon_t)e^{-\alpha_t}+\epsilon_t e^{\alpha_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)}. \end{aligned}$$

由式(26)可知

$$\begin{aligned} Z_t\mathcal{D}_{t+1}(x_i) &= \mathcal{D}_t(x_i)\exp(-\alpha_t f(x_i)h_t(x_i)) \\ Z_{t-1}\mathcal{D}_t(x_i) &= \mathcal{D}_{t-1}(x_i)\exp(-\alpha_{t-1} f(x_i)h_{t-1}(x_i)) \\ &\vdots \\ Z_1\mathcal{D}_2(x_i) &= \mathcal{D}_1(x_i)\exp(-\alpha_1 f(x_i)h_1(x_i)) \end{aligned}$$

对上式两边同时相乘, 以及 $\mathcal{D}_1(x_i)=1/m$ 可得

$$\mathcal{D}_{t+1}(x_i)\prod_{t=1}^{T}Z_t=\frac{1}{m}\exp(-f(x_i)H_t(x_i)).$$

再对上式两边中 x_i 求和, 以及利用

$$\frac{1}{m}\sum_{i=1}^{m} I(\mathrm{sgn}(H(x_i))\neq f(x_i)) \leq \frac{1}{m}\sum_{i=1}^{m}\exp(-f(x_i)H(x_i)) = \sum_{i=1}^{m}\mathcal{D}_{t+1}(x_i)\prod_{t=1}^{T}Z_t = \prod_{t=1}^{T}Z_t.$$

定理得证. ■

假设 AdaBoost 基学习器函数空间为 $\mathcal{H}$, 以及 AdaBoost 算法运行了 T 轮, 根据 Baum & Haussler 的理论可知 $\mathcal{C}(\mathcal{H})$ 的 VC 维为 $O(Td)$, 根据上面的定理有:

$$P_{\mathcal{D}}(y \neq f(x)) \leq P_S(y \neq f(x)) + O(\sqrt{(Td\ln(m/Td) + \ln 1/\delta)/m}).$$

从这个结论可知: 随着迭代轮数 T 的增加, AdaBoost 所产生的学习器函数空间复杂度将增加, 从而导致 AdaBoost 学习方法过拟合的风险.

4.4.2 函数空间的 Rademacher 复杂度

由 4.3节中 Rademacher 复杂度定义可知,Rademacher 复杂度的计算依赖于数据分布 $\mathcal{D}$ 以及假设空间 $\mathcal{H}$, 这使得 Rademacher 复杂度的计算非常复杂. 因此, 本节只介绍一些简单假设空间或函数空间的 Rademaher 复杂度.

超平面的 Rademacher 复杂度

超平面的 Rademacher 复杂度与 VC 维不同,Rademacher 复杂度与数据相关, 因此在计算 Rademahcer 复杂度需要数据分布 $\mathcal{D}$ 的信息. 具体有如下定理.

定理 4.6. *令 $D \subset \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq r\}$ 是大小为 m 的数据集, 则典型超平面族 $\mathcal{H} = \{\boldsymbol{x} \mapsto \boldsymbol{w} \cdot \boldsymbol{x} : \|\boldsymbol{w}\| \leq \Lambda\}$ 的经验 Rademacher 复杂度满足下列不等式*

$$\hat{R}_D(\mathcal{H}) \leq \sqrt{\frac{r^2\Lambda^2}{m}}.$$

5.4.1节中将会给出该定理的具体证明. 不难发现定理 4.6只给出了 Rademacher 复杂度的上界. 这是由于我们先前提到的 Rademacher 复杂度依赖于数据的分布, 这使得确定 Rademacher 复杂度的具体数值变得十分困难.

注:VC 维由 [Vapnik and Chervonenkis, 1971] 提出, 它的出现使研究无限假设空间的复杂度成为可能. Sauer 引理由于 [Sauer, 1972] 而命名, 但 [Vapnik and Chervonenkis, 1971] 和 [Shelah, 1972] 也分别独立的推导出了该结果. 本章中神经网络的 VC 维部分参考了 Nati Srebro 在课程 CMSC 35900 中关于 Multilayer Neural Networks 的讨论.

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.