机器学习建模、优化与调参专题

邓月，电子科技大学，基础与前沿研究院

2022 年 10 月 29 日

机器学习常用于执行远远超出人类能力的任务。例如，机器学习程序通过浏览和处理大型数据，能够检测到超出人类感知范围的模式。机器学习（的经验）训练涉及的数据往往是随机生成的，机器学习的任务就是处理这些背景下的随机生成样本，得出与背景相符的结论。机器学习与统计学有密切关系，两个学科之间确实有很多共同点，尤其表现在目标和技术方面。但是，两者之间仍存在显著的差别：如果一个医生提出吸烟与心脏病之间存在关联这一假设，这时应该由统计学家去查看病人样本并检验假设的正确性（这是常见的统计任务—假设检验）。相比之下，机器学习的任务是利用患者样本数据找出心脏病的原因。我们希望自动化技术能够发现被人类忽略的、有意义的模式（或假设）。两者的另一个区别是，统计关心算法的渐进性（如随机样本量增长至无穷大，统计估计的收敛问题），机器学习理论侧重于有限样本。也就是说，给定有限可用样本，机器学习理论旨在分析学习器可达到的准确度。再如，在统计学中，常首先提出数据模型预测（生成数据呈正态分布或依赖函数为线性），在机器学习中常考虑“非参数”背景，对数据分布的性质假设尽可能地少，学习算法自己找出最接近数据生成过程的模型。一个成功的学习器应该能够从个别例子进行泛化，这也称为归纳推理。

几乎所有的机器学习算法都可以归结为一个寻找损失函数极值的优化问题。建立模型和构造合理的损失函数是机器学习方法的第一步。在损失函数确定的情况下，通常采用适当的数值或解析优化方法来求解优化问题 [1]。从理论的角度出发，计算学习理论 [2] 能够分析学习任务的困难，为学习算法提供理论保证，并根据分析结果指导算法设计。本文中关于此的内容主要包括：

（1）**假设空间的复杂度**（假设数、VC 维、Rademacher 复杂度、学习算法的稳定性），**可学习概念**（PAC 可学习、不一致可学习、一致收敛性），**算法的可学习条件**（可学习条件（有限 $\mathcal{H}$）、VC 维下的可学习条件（无限 $\mathcal{H}$）。因为无法直接计算得到“可学习性”所针对的 $E_{out}(h)$，所以需要使得通过 $E_{in}(h)$ 去推测 $E_{out}(h)$ 是可行的，因此引出了“可学习条件”）。注意，VC 维理论貌似只能用于二分类问题，对于多分类问题，可将 VC 维拓展为 Natarajan 维。

（2）基于假设空间复杂度和算法可学习性的**泛化误差界分析**（即 $E_{out}(h)$ 与 $E_{in}(h)$ 之间的误差）。其中，基于 VC 维的泛化误差界是分布无关、数据独立的，基于 Rademacher 复杂度的泛化误差界考虑了数据分布。它俩都和具体学习算法无关。学习算法的稳定性分析不必考虑假设空间中所有可能的假设，只需根据算法自身的特性（稳定性）来讨论输出假设的泛化误差界。

（3）**样本复杂度与可学习性的关系**。在假设类 $\mathcal{H}$ 为有限和无限的情况下，分别研究样本复杂度满足怎样的条件时能使得算法是可学习的。以及“不一致可学习性”，允许样本数量依赖于学习器所在假设空间而变化，不同于之前的结论。还有“一致收敛性”，即“不一致可学习性”的进一步松弛。

（4）**结构风险最小化**和**最小描述长度**（貌似一般更倾向用**正则化函数**来“描述”）。上述都是对于 $\mathcal{A}$ 在一个特定的学习任务上（我觉得可以理解成“数据的分布”吧，即“目标函数（集）”不同）而言的，现在开始关注 $\mathcal{A}$ 在不同学习任务上的表现情况，因为 No free lunch theorem。

（5）**学习规则总结**。包括：经验风险最小化（ERM）、结构风险最小化（SRM）和正则损失最小化（RLM）（用到凸性、利普利茨性和光滑性等概念）。

（6）**调参**。包括：模型的选择方法（结构风险最小化和验证法）、学习失败时的做法。

（7）**权衡**。包括：“偏差-复杂度”权衡和“适合-稳定性”权衡。

目录

1 基本概念与结论

1.1 Hoeffding 不等式、Jesen 不等式、McDiarmid 不等式

（1）若用 μ 表示样本期望，ν 表示总体期望，则 **Hoeffding 不等式**为 $P(\nu-\mu\geq\epsilon)\leq e^{-2\epsilon^2N}$ 和 $P(|\nu-\mu|\geq\epsilon)\leq 2e^{-2\epsilon^2N}$【推导过程参见百度】，其中 N 是样本数。可以看出，当 N 越来越大时，μ 和 ν 之差大于 ϵ 的概率的上界越来越接近 0，所以样本期望 μ 越来越接近总体期望 ν。

（2）**Jesen 不等式**为：对任意凸函数 $f(x)$，有 $f(E(x))\leq E(f(x))$。

（3）**McDiarmid 不等式**为：若 $x_1,x_2,...,x_m$ 为 N 个独立随机变量，且对任意 $1\leq i\leq N$，函数 f 满足 $\sup\limits_{x_1,...,x_m,x_i'}|f(x_1,...,x_m)-f(x_1,...,x_{i-1},x_i',x_{i+1},...,x_m)|\leq c_i$，则对任意 $\epsilon>0$ 有

$$P(f(x_1,...,x_m)-E(f(x_1,...,x_m))\geq\epsilon)\leq e^{\frac{-2\epsilon^2}{\sum_i c_i^2}}, \tag{1.1}$$

$$P(|f(x_1,...,x_m)-E(f(x_1,...,x_m))|\geq\epsilon)\leq 2e^{\frac{-2\epsilon^2}{\sum_i c_i^2}}. \tag{1.2}$$

理解：

（1）证明该不等式，需要用到马尔科夫不等式：$P(|X|\geq\epsilon)\leq\dfrac{E(X)}{\epsilon}$。马尔科夫不等式将概率关联到了期望。

1.2 学习算法、假设空间、分布、目标函数（集）、训练样本集、假设、损失函数、预期风险函数、经验风险函数、泛化误差（或期望误差）、经验误差、学习目标、经验风险最小化（ERM）

本节内容一部分参考自 CSDN 博客：https://tangshusen.me/2018/12/09/vc-dimension/。

（1）机器学习的基本成分为**算法** $\mathcal{A}$ 和**假设空间** $\mathcal{H}$。对于每个**分布** $\mathcal{X}\times\mathcal{Y}$（考虑为一个实例空间和一个目标空间的乘积），存在函数或分布 $f:\mathcal{X}\to\mathcal{Y}$，我们称之为**目标函数（集）**（注意：也可能有多个）。因为 $\mathcal{X}\times\mathcal{Y}$ 往往十分大而不可知的，所以需要从分布中独立重复地抽样得到的**训练样本集** $\mathcal{D}=\{(x_1,y_1),...,(x_N,y_N)|x_i\in\mathcal{X},y_i\in\mathcal{Y}\}$，机器学习的过程就是通过算法 $\mathcal{A}$，在假设空间 $\mathcal{H}$ 中，根据训练样本集 $\mathcal{D}$，学习出最好的**假设** $h\in\mathcal{H}$，使得 $h\approx f$（可见，假设空间中的假设也是从样本空间 $\mathcal{X}$ 到标记空间 $\mathcal{Y}$ 的函数或分布）。然后就能够用这个假设 h 来进行预测。

注意，除了用 $\mathcal{D}$ 表示训练样本集，$\mathcal{X}\times\mathcal{Y}$ 表示分布外，本文的一些地方也用 S 表示训练样本集，$\mathcal{D}$ 表示分布。

（2）任意函数 $L:\mathcal{X}\times\mathcal{Y}\times\mathcal{H}^k(\mathcal{X})\to\mathbb{R}^+\cup\{0\}:(x,y,h)\to L(x,y,h)$，如果满足 $h(x)=y$，则 $L(x,y,h)=0$（一致性）被称为**损失函数** [3]。其中 $\mathcal{H}^k(\mathcal{X})$ 表示 L^2 空间中的函数的 Sobolev 空间并且允许弱导数直到 k，这是更正式意义下的假设空间的定义。机器学习的作用是确定 $h\in\mathcal{H}^k(\mathcal{X})$。损失函数 L 不一定满足对称性 $L(x,y,h)=L(x,h,y)$。

（3）假设学习环境可由与对应密度 $p(x,y)$ 相关的概率分布 $P(x,y)$ 通过 $dP(x,y)=p(x,y)dxdy$ 来建模，一旦定义了 h，随机变量 X 和 Y 就会生成随机变量 L，即为损失函数 $L(x,y,h)$ 的值。**预期风险函数** [3] 是 $E(h)=E_{X,Y}V=\int_{\mathcal{X}\times\mathcal{Y}}L(x,y,h)dP(x,y)$。

不幸的是，$E(\cdot)$ 的最小化通常很难，因为 $P(x,y)$ 没有被明确给出。在现实问题中，分布 $P(x,y)$ 仅来自训练样本集 $\mathcal{D}$ 的样本的集合。我们可以将该集合视为分配概率密度 $p(x,y)=\frac{1}{N}\sum_{i=1}^{N}\delta(x-x_i)\cdot\delta(y-y_i)$ 的方式，这样做的话期望风险函数就变成了 $E(h)=\frac{1}{N}\sum_{i=1}^{N}L(x_i,y_i,h)$【推导参见 [3] 的 P42】，称为**经验风险函数**。这是一种可以直接优化的函数。

（4）学到的假设 h 在除了训练样本外的其他所有样本（out-of-sample）上的损失 $E_{out}(h)$ 称为**泛化误差（或期望误差）**。学到的假设 h 在训练样本（in-of-sample）上的损失 $E_{in}(h)$ 称为**经验误差**。机器学习的**学习目标**就是使得 $E_{out}(h)\approx E_{in}(h)\approx 0$。其中，$E_{out}(h)\approx 0$ 是因为我们有 $h\approx f$，而 $E_{out}(f)=0$，则我们的目标也是使 $E_{out}(h)\approx 0$。机器学习的泛化能力往往是通过研究泛化误差的概率上界所进行的。

（5）令 h 表示学习算法 $\mathcal{A}$ 输出的假设，若 h 满足 $E_{in}(h)=\min_{h'\in\mathcal{H}}E_{in}(h')$，则称 $\mathcal{A}$ 为满足**经验风险最小化**原则的算法（注意：这个概念是和 h 相关的哦，即在特定 h 下的 $\mathcal{A}$。而且只是在训练样本上进行的哦）

【例题参见 [3] 的 P43-44。(1) 对数似然的最大化可以认为是经验风险最小化问题，推导用到 $e^{\ln x}=x$，并最终将其与条件熵的形式联系起来解释。(2) 推导对数似然最大化在正态分布、二分类和多分类情形下崩溃到有限维。与“二次损失”(其实 BPR 模型的贝叶斯推理也说明了这一切)、“交叉熵”(与相对熵或 KL 散度都是针对于两个分布的) 和“softmox 损失”联系。这些对理解损失函数和经验风险最小化的实际含义很关键！关于熵的相关定义参见1.23节】

【最大似然估计还可参见 [3] 的 54-56，反映了统计学与机器学习的联系：统计学家已经在基于数据样本的推理过程中积累了巨量的专业知识，这一主题依赖于已经成为当前机器学习方法核心的方法论】

理解：

(1) 在现实应用中我们对分布 $\mathcal{X}\times\mathcal{Y}$，即目标函数（集）通常一无所知，更别说获得一个假设空间与目标函数集恰好相同的学习算法。因此要借助于训练样本集，它是真实世界（可以理解为目标函数 $f:\mathcal{X}\to\mathcal{Y}$）的一个缩影。假设 $\mathcal{X}$ 中的所有样本服从一个隐含未知的分布，$\mathcal{D}$ 中所有样本都是独立地从这个分布上采样而得，即独立同分布（i.i.d.）样本。必须注意的是，我们并不需要学习算法知道此概率分布的任何信息。

1.3 PAC 辨识、PAC 可学习、PAC 学习算法、样本复杂度

本节内容一部分参考自 [2]。这里先按照 [2] 中的概念，称“目标函数”为**目标概念**，表示为 c，即 $c(x)=y$。所有目标概念所构成的集合称为**概念类**，表示为 $\mathcal{C}$。计算学习理论中最基本的是概率近似正确（简称，PAC）。它的含义是：我们希望基于学习算法 $\mathcal{A}$ 学得的模型所对应的假设 h 尽可能接近目标概念 c，因此，我们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。由于无法保证标签预测绝对准确，因此以 ϵ 表示误差，称为**精确参数**（对应于 PAC 的“近似正确”部分）。同时，将采样到非代表性样本的概率表示为 δ，则 $1-\delta$ 称为**置信参数**（对应于 PAC 的“概率”部分）。

（1）对 $0<\epsilon,\delta<1$，所有 $c\in\mathcal{C}$ 和训练样本集 $\mathcal{D}$，若存在学习算法 $\mathcal{A}$，其输出假设 $h\in\mathcal{H}$ 满足 $P(E_{out}(h)\le\epsilon)\ge 1-\delta$，则称学习算法 $\mathcal{A}$ 能从假设空间 $\mathcal{H}$ 中 **PAC 辨识**概念类 $\mathcal{C}$，其中 $h\approx c$；

（2）令 N 表示从 $\mathcal{X}$ 所服从的分布中独立同分布采样得到的样例数目，$0<\epsilon,\delta<1$，对所有分布，若存在学习算法 $\mathcal{A}$ 和多项式函数 $\text{poly}(\cdot,\cdot,\cdot,\cdot)$，使得对于任何 $N\ge\text{poly}(1/\epsilon,1/\delta,\text{size}(x),\text{size}(c))$，其中 $\text{size}(x)$ 为数据本身的复杂度，$\text{size}(c)$) 为目标概念的复杂度（为什么要关注 N? 因为 $E_{out}(h)$ 是和训练样本集的数目 N 相关的奥），$\mathcal{A}$ 能从假设空间 $\mathcal{H}$ 中 PCA 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。

（3）若学习算法 $\mathcal{A}$ 使概念类 $\mathcal{C}$ 为 PAC 可学习的，且 $\mathcal{A}$ 的运行时间也是多项式函数 $\text{poly}(\cdot,\cdot,\cdot,\cdot)$，则称概念类 $\mathcal{C}$ 是高效 PAC 可学习的，称 $\mathcal{A}$ 为概念类 $\mathcal{C}$ 的 **PAC 学习算法**。

（4）满足 PAC 学习算法 $\mathcal{A}$ 所需要的 $N \geq \text{poly}(1/\epsilon, 1/\delta, \text{size}(x), \text{size}(c))$ 中最小的 N，称为学习算法 $\mathcal{A}$ 的**样本复杂度**。

理解：

（1）对于（4），假定学习算法 $\mathcal{A}$ 处理每个样本的时间为常数，则 $\mathcal{A}$ 的时间复杂度等价于样本复杂度，于是我们对算法时间复杂度的关心就转化为对样本复杂度的关心。

（2）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？某算法在什么样的条件下可进行有效学习？需要多少训练样例才能获得较好的模型？

（3）可见，可学习是针对于 E_{out} 而言的。但由于不可知分布和所有样本，E_{out} 往往是无法直接计算的，因此需要用 E_{in} 来推测 E_{out}，也就是后面会引出的可学习条件。

1.4 可分、不可分、有限假设空间、不可知 PAC 可学习

本节内容一部分参考自 [2]。

（1）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 $\mathcal{A}$ 是**可分的**，也称为一致的。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 $\mathcal{A}$ 是**不可分的**，亦称不一致的。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 $h \in \mathcal{H}$，$E_{in}(h) \neq 0$，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。

（2）一般而言，$\mathcal{H}$ 越大，其中含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也越大。$|\mathcal{H}|$ 为有限时，我们称 $\mathcal{H}$ 为**有限假设空间**，否则为无限假设空间。虽然现实学习任务所面临的通常是无限假设空间，度量其复杂度时最常见的办法是后面将要讲到的 “VC 维”，但我们还是先从有限假设空间说起。

（3）令 N 表示从 $\mathcal{X}$ 所服从的分布中独立同分布采样得到的样例数目，$0 < \epsilon, \delta < 1$，对所有分布，若存在学习算法 $\mathcal{A}$ 和多项式函数 $\text{poly}(\cdot, \cdot, \cdot, \cdot)$，使得对于任何 $N \geq \text{poly}(1/\epsilon, 1/\delta, \text{size}(x), \text{size}(c))$，$\mathcal{A}$ 能从假设空间 $\mathcal{H}$ 中输出满足下式的假设 h：

$$P(E_{out}(h) - \min_{h' \in \mathcal{H}} E_{out}(h') \leq \epsilon) \geq 1 - \delta, \tag{1.3}$$

则称假设空间 $\mathcal{H}$ 是**不可知 PAC 可学习**的。此外，“高效不可知 PAC 可学习的”，“不可知 PAC 学习算法” 和 “样本复杂度” 的概念与1.3类似。

结论：

注意，以下结论都是针对于有限假设空间而言的。

（1）**在可分情形下**。有限假设空间 $\mathcal{H}$ 都是 PAC 可学习的（因为可以使用剔除法不断筛选出和 $\mathcal{D}$ 表现一致的 h），为了学得目标概念 c 的有效近似（因为实际中训练集中的样例不可能包含全部，所以只要 $\mathcal{D}$ 能够使 $\mathcal{A}$ 以一定概率找到 c 的近似即可），所需的样例数目 $N \geq \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta})$，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 $O(\frac{1}{N})$【证明参见 [2] 的 P271】。

（2）**在不可分情形下**。首先引入下面两个结论：

（a）若训练集 $\mathcal{D}$ 包含 N 个从 $\mathcal{X}$ 所服从的分布上独立同分布采样而得的样例，$0 < \epsilon < 1$，则对任意 $h \in \mathcal{H}$，下式以至少 $1 - \delta$ 的概率成立：

$$E_{in}(h) - \sqrt{\frac{\ln(2/\delta)}{2N}} \leq E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{\ln(2/\delta)}{2N}}. \tag{1.4}$$

可见，样例数目 N 较大时，h 的经验误差是其泛化误差很好的近似，对于有限假设空间 $\mathcal{H}$，我们有：

（b）若 $\mathcal{H}$ 为有限假设空间，$0 < \delta < 1$，则对任意 $h \in \mathcal{H}$，有

$$P(|E_{out}(h) - E_{in}(h)| \leq \sqrt{\frac{\ln|\mathcal{H}| + \ln(2/\sigma)}{2N}}) \geq 1 - \delta. \tag{1.5}$$

【证明参见 [2] 的 P272】。

在不可分情形下。学习算法 $\mathcal{A}$ 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【证明参见 [2] 的 P272】。$\mathcal{H}$ 中泛化误差最小的假设是 $\arg\min_{h\in\mathcal{H}} E_{out}(h)$。

1.5 可学习条件、假设数、有效假设数

本节内容一部分参考自 CSDN 博客：https://tangshusen.me/2018/12/09/vc-dimension/。

因为我们无法获得训练样本外的其他所有样本，也就没法计算 $E_{out}(h)$ 了，那该怎么办呢？类比1.1，对于特定的假设 h，借助 Hoeffding 不等式，我们可以得到 $P(|E_{out}(h)-E_{in}(h)|\geq\epsilon)\leq 2e^{-2\epsilon^2 N}$。对于任意的假设 h，假设 $\mathcal{H}$ 中有 M 个假设 $h_1,h_2,...,h_M$，称 M 为**假设数**。则有 $P(E(h_1)\geq\epsilon\cup E(h_2)\geq\epsilon\cup...\cup E(h_M)\geq\epsilon)\leq P(E(h_1)\geq\epsilon)+P(E(h_2)\geq\epsilon)+...+P(E(h_M)\geq\epsilon)\leq 2Me^{-2\epsilon^2 N}$，其中 $E(h_i)=|E_{in}(h_i)-E_{out}(h_i)|$。因此，

$$\forall h\in\mathcal{H}, P(|E_{out}(h)-E_{in}(h)|\geq\epsilon)\leq 2Me^{-2\epsilon^2 N}. \tag{1.6}$$

可以看出，我们如果想要通过样本集上的经验误差 $E_{in}(h)$ 去推测总体的期望误差 $E_{out}(h)$，则需要满足以下两个机器学习的**可学习条件**：（a）算法 $\mathcal{A}$ 能够从假设空间 $\mathcal{H}$ 中选出的假设 h 满足 $E_{in}(h)\approx 0$；（b）假设空间 $\mathcal{H}$ 中假设数 M 是有限的且训练样本数 N 足够大。（这里说的是有限假设空间情形下哦，但现实中机器学习基本都是无限假设空间情形，所以后面会讲 VC 维下的可学习条件，这里只是先用最基本的给个感觉 ）

然而，假设数 M 往往很大甚至是趋于无穷（因为在现实中所面临的学习任务通常都是对于无限的假设空间，比如实数域中的所有区域、$\mathbb{R}^b$ 空间中的所有线性超平面等等），这样就使得不等式1.6右端（即上界）成了无穷大而没有约束意义。事实上，不等式 $P(A_1\cup A_2\cup...\cup A_M)\leq\sum_{i=1}^{M}P(A_i)$ 中，当 A_i 相互独立时不等式取等。也就是说，假设空间 $\mathcal{H}$ 里的 h_i 之间并不是完全独立的，它们是有很大的重叠的，也就是在 M 个假设中，有很多假设都可以归为同一类。虽然假设空间假设数 M 一般非常大 (甚至无穷)，但在特定的训练样本集上，**有效假设数**是有限的。因此可以将上述不等式重写为：

$$\forall h\in\mathcal{H}, P(|E_{out}(h)-E_{in}(h)|\geq\epsilon)\leq 2\,\text{effective}(M)e^{-2\epsilon^2 N}. \tag{1.7}$$

理解：

（1）假设数 M 在机器学习的科学系条件中有着重要作用，应该合理选取。因为当 M 较小时，条件（b）很容易满足，但条件（a）不容易满足。当 M 较大时，条件（a）很容易满足，但条件（b）不容易满足。

（2）两个机器学习的可学习条件中，条件（b）可以保证 $E_{in}(h)\approx E_{out}(h)$，再通过条件（a），即达到了 $E_{out}(h)\approx 0$ 的目的。也正好对应着训练和测试两个过程：（a）训练过程希望经验误差 $E_{in}(h)$ 尽可能小；（b）测试过程希望期望误差 $E_{out}(h)$ 尽可能小，即 $E_{out}(h)$ 接近于 $E_{in}(h)$。所以我们不应该只关心如何利用算法找到使 $E_{in}(h)$ 很小的 h。想让学习可行，也要保证 $E_{out}(h)$ 接近于 $E_{in}(h)$。

1.6 对分、增长函数、打散、断点

本节内容一部分参考自 CSDN 博客：https://tangshusen.me/2018/12/09/vc-dimension/。下面这些概念是为 VC 维理论进行的引入。注意，**VC 维理论，以及下面的这些概念，貌似只能解决二分类问题。对于多分类问题，可将 VC 维拓展为 Natarajan 维。**

（1）设假设空间 $\mathcal{H}=\{h:\mathcal{X}\to\{+1,-1\}\}$，我们称 $h(X_1,X_2,...,X_N)=(h(X_1),h(X_2),...,h(X_N))\in\{+1,-1\}^N$ 为一个**对分**，即一个对分表示样本的一种标记结果。$\mathcal{H}(X_1,X_2,...,X_N)$ 表示假设空间 $\mathcal{H}$ 在训练样本集 $\mathcal{D}$ 上的所有对分（强调一下，是针对于选定的 $\mathcal{D}$）【例题参见 CSDN 博客的图 2.1 和 2.2，以二维平面上的直线为例】。对分实际上就是限制无限大小的假设类 $\mathcal{H}$ 限制在 X 上，也就是利用先验知识在无限大小的 $\mathcal{H}$ 中根据特定的学习任务将其限定为 $\mathcal{H}_X$[4]。

（2）由上可见，$\mathcal{H}(X_1,X_2,...,X_N)$ 的元素个数 $|\mathcal{H}(X_1,X_2,...,X_N)|$ 是取决于具体的训练样本集 $\mathcal{D}$ 的。为了去掉它对具体 $\mathcal{D}$ 的依赖性，我们引入假设空间 $\mathcal{H}$ 的**增长函数**：$m_{\mathcal{H}}(N)=\max_{X_1,X_2,...,X_N\in\mathcal{X}}|\mathcal{H}(X_1,X_2,...,X_N)|$。其表示假设空间 $\mathcal{H}$ 对个任意 N 个样本所能赋予标记的的最大可能结果数，其上界为 2^N。显然，$m_{\mathcal{H}}(N)$

越大，$\mathcal{H}$ 的表示能力越强。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度。

（3）当假设空间 $\mathcal{H}$ 作用于大小为 N 的样本集 $\mathcal{D}$ 时，产生的对分数量等于 2^N，即 $m_{\mathcal{H}}(N)=2^N$ 时，就称 D 被 $\mathcal{H}$ **打散**【例题参见 CSDN 博客的图 2.3 和 2.4，以二维平面上的直线为例。例题参见 [4] 的 P34 例 6.2】。

（4）尽管增长函数把假设数从无穷缩小到有效（$\leq 2^N$），但是这个量级还是太大了，很难保证式1.6右边这个上界趋于 0（误差大于 ϵ 的概率趋于 0，则误差小于 ϵ 的概率趋于 1），所以能不能把量级再缩小一点？为了达到这个目的，引入**断点**的概念：对于假设空间 $\mathcal{H}$ 的增长函数 $m_{\mathcal{H}}(N)$，从 $N=1$ 出发逐渐增大，当增大到 k 时，出现 $m_{\mathcal{H}}(N)<2^N$ 的情形，则我们说 k 是该假设空间 $\mathcal{H}$ 的断点。换句话说，对于任何大小为 $N(N\geq k)$ 的数据集，H 都没有办法打碎它。进一步地，有如下定理：设断点存在且为 k 的假设空间 $\mathcal{H}$ 的增长函数上界为 $B(N,k)$，则 $B(N,k)$ 满足

$$m_{\mathcal{H}}(N)\leq B(N,K)\leq \sum_{i=1}^{k-1}\binom{N}{i}\leq N^{k-1}. \tag{1.8}$$

其中，最后一个不等号仅在 $N\geq 2$ 且 $k\geq 2$ 时成立【证明参见 CSDN 博客给出的知乎链接】。

所以我们得到结论：如果断点存在，则增长函数 $m_{\mathcal{H}}(N)$ 是多项式的，其最高幂次项为 N^{k-1}，多项式的量级就比 2^N 小多了。

结论：

（1）令 $\mathcal{H}$ 是从 $\mathcal{X}$ 到 $\{0,1\}$ 的函数构成的假设类。令 N 是训练集的大小。假定存在大小为 $2N$ 的集合 $C\subset\mathcal{X}$ 能被 $\mathcal{H}$ 打散。那么，对于任何学习算法 $\mathcal{A}$，在 $\mathcal{X}\times\{0,1\}$ 上必然存在一个分布 $\mathcal{D}$ 和预测器 $h\in\mathcal{H}$ 使得 $E_{out}(h)=0$，但是对于所选样本集 $S\sim\mathcal{D}^m$ 至少以 1/7 的概率有 $E_{out}(\mathcal{A}(S))\geq 1/8$。[4]

该结论说明了如果假设类 $\mathcal{H}$ 打散了大小为 $2N$ 的集合 C，那么我们将无法通过 N 个样本来学习 $\mathcal{H}$，因为剩余样本的标签的每一种可能的组合都可以在假设类 $\mathcal{H}$ 中找到某些假设与之对应。**从哲学上讲，如果有人可以解释每个现象，他的解释本身就是毫无意义的。**

（2）Sauer-Shelah-Perles（Sauer 引理）：令 $\mathcal{H}$ 是一个假设类，且 $VC(\mathcal{H})\leq d<\infty$。那么对于所有 $N, m_{\mathcal{H}}(N)\leq\sum_{i=0}^{d}\binom{N}{i}$。特别地，如果 $N>d+1$，那么 $m_{\mathcal{H}}(N)\leq(eN/d)^d$【证明参见 [4] 的 P38】。

理解：

（1）假设空间 $\mathcal{H}$ 中不同的假设对于 $\mathcal{D}$ 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，但其对 $\mathcal{D}$ 中示例赋予标记的可能结果数是有限的。

（2）对于（2），这个式子表述为下面这个形式更清楚：$m_{\mathcal{H}}(N)=\max\limits_{\{x_1,\ldots,x_N\}\in\mathcal{X}}|\{(h(x_1),\ldots,h(x_N))|h\in\mathcal{H}\}|$，显然，自变量就是从 $\mathcal{X}$ 中选样本集，每个样本集对应一个 $|\cdot|$。而且很显然它是个关于 N 的函数。

1.7 VC 界（VC 不等式）、VC 维、VC 维下的可学习条件、$E_{in}(h)$ 与 $E_{out}(h)$ 的关系

本节内容一部分参考自 CSDN 博客：https://tangshusen.me/2018/12/09/vc-dimension/。注意：**VC 维理论貌似只能解决二分类问题。对于多分类问题，可将 VC 维拓展为 Natarajan 维。**

（1）探究式1.8中的 effective(M) 是十分重要的，那我们可以直接用 $m_{\mathcal{H}}(N)$ 去替换掉 effective(M) 吗？答案是不可以。正确的形式应该为 **VC 界（VC 不等式）**

$$\forall h\in\mathcal{H}, P(|E_{out}(h)-E_{in}(h)|\geq\epsilon)\leq 4m_{\mathcal{H}}(2N)e^{-\frac{1}{8}\epsilon^2 N}. \tag{1.9}$$

其中 VC 代表 Vapnik–Chervonenkis【证明参见 CSDN 博客给出的论文链接】。

（2）假设空间 $\mathcal{H}$ 的 **VC 维**是能被 $\mathcal{H}$ 打散的最大数据集的大小，即

$$VC(\mathcal{H})=\max\{N: m_{\mathcal{H}}(N)=2^N\}. \tag{1.10}$$

如果 $\mathcal{H}$ 可以打散任意大的集合，我们就说 $\mathcal{H}$ 的 VC 维是无穷的。据此定义，有 $VC(\mathcal{H})=k-1$，其中 k 是 $\mathcal{H}$ 的断点。

再由式1.8可得，当 $N \geq 2$ 且 $k \geq 2$ 时，$m_{\mathcal{H}}(N) \leq N^{VC(\mathcal{H})}$，则式1.9可写作

$$\forall g \in \mathcal{H}, P(|E_{out}(g) - E_{in}(g)| \geq \epsilon) \leq 4(2N)^{VC(\mathcal{H})} e^{-\frac{1}{8}\epsilon^2 N}. \tag{1.11}$$

（3）引入 VC 维的概念后，可以得到机器学习的 **VC 维下的可学习条件**：（a）学习算法 $\mathcal{A}$ 能够从 $\mathcal{H}$ 选出的假设 g 满足 $E_{in}(g) \approx 0$；（b）假设空间 $\mathcal{H}$ 的 VC 维 $VC(\mathcal{H})$ 是有限的且训练样本数 N 足够大。

（4）将式1.11右边项令为 δ，即 $P(|E_{out}(g) - E_{in}(g) \geq \epsilon) \leq \delta$。通过反解可得

$$\epsilon = \sqrt{\frac{8}{N} \ln \frac{4(2N)^{VC(\mathcal{H})}}{\delta}}. \tag{1.12}$$

则有

$$E_{in}(g) - \epsilon \leq E_{out}(g) \leq E_{in}(g) + \epsilon. \tag{1.13}$$

其中，ϵ 也可以看做模型的复杂度，模型越复杂，$E_{out}(g)$ 与 $E_{in}(g)$ 离得越远。

当固定样本数 N 时，随着 VC 维的上升，$E_{in}(g)$ 会不断降低，而复杂度会不断上升，其上升与下降的速度在每个阶段都有所不同，因此我们要寻找一个二者兼顾的比较合适的 VC 维使 $E_{out}(g)$ 最小【图示见 CSDN 博客，泛化误差（期望误差）在最上面（即误差相对最大），趋势随着 VC 维先降低后增加】。

样本数 N 也会影响 $E_{out}(g)$。例如，当前有一个 $VC(\mathcal{H}) = 3$ 的假设空间，要使 $\epsilon = 0.1$ 且 $\delta = 0.1$，则要想满足式1.12，可计算出理论上样本数 N 需要达到 $10000VC(\mathcal{H})$ 这个量级，但实际应用中我们发现 N 达到 $10VC(\mathcal{H})$ 就够了。这是因为，VC 界是一个及其宽松的上界，因为它需要对任何学习算法，对任何数据分布，对任何目标函数都要成立，所以实际应用中的上界要比 VC 界小很多。

计算 VC 维：

通常这样来计算 $\mathcal{H}$ 的 VC 维：若存在大小为 d 的示例集能被 $\mathcal{H}$ 打散，但不存在任何大小为 $d+1$ 的示例集能被 $\mathcal{H}$ 打散，则 $\mathcal{H}$ 的 VC 维是 d。【例题参见 [2] 的 P275 的例 12.1 和例 12.2，实数域中的区间 $[a, b]$ 和二维实平面上的线性划分的 VC 维】【例题参见 [2] 的 P289 的 12.6、12.7，考虑了决策树分类器和最近邻分类器】【例题参见 [4] 的 P35-36】

结论：

（1）对于（2），可知 VC 维与增长函数有密切联系，下面的 Sauer 引理给出了二者之间的定量关系：若假设空间 $\mathcal{H}$ 的 VC 维为 d，则对任意 $N \in \mathbb{N}$，有 $m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d} \binom{N}{i}$【证明参见 [2] 的 P275-276】。

（2）若假设空间 $\mathcal{H}$ 的 VC 维为 d，则对任意整数 $m \geq d$ 有 $m_{\mathcal{H}}(N) \leq (\frac{e \cdot m}{d})^d$【证明参见 [2] 的 P276-277】。基于式1.9和本结论可以得到基于 VC 维的泛化误差界：$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{2d \ln \frac{eN}{d}}{N}} + \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$。

（3）若假设空间 $\mathcal{H}$ 的 VC 维为 d，则对任意 $m > d, 0 < \delta < 1$ 和 $h \in \mathcal{H}$ 有 $P(|E_{out}(h) - E_{in}(h)| \leq \sqrt{\frac{8d \ln \frac{2eN}{d} + 8 \ln \frac{4}{\delta}}{N}}) \geq 1 - \delta$【证明参见 [2] 的 P277】。可见，泛化误差只与样例数目 N 有关，收敛速率为 $O(\frac{1}{\sqrt{m}})$，与数据分布 $\mathcal{D}$ 和样例集 $\mathcal{D}$ 无关。因此，基于 VC 维的泛化误差界是分布无关、数据独立的。

（4）任何 VC 维有限的假设空间 $\mathcal{H}$ 都是（不可知）PAC 可学习的【证明参见 [2] 的 P278-279】。

（5）令 $\mathcal{H}$ 为无穷 VC 维的假设类，那么 $\mathcal{H}$ 不是 PAC 可学习的【证明参见 [4] 的 P35】。

理解：

（1）VC 界的意义在于：如果假设空间 $\mathcal{H}$ 存在有限的断点 k，即 $m_{\mathcal{H}}(2N)$ 会被最高幂次为 k–1 的多项式上界给约束住，那么，随着 N 的逐渐增大，指数式的下降会比多项式 $m_{\mathcal{H}}(2N)$ 的增长速度更快，所以此时可以推断出 VC 界是有界的。更进一步，当 N 足够大时，对于 H 中的任意一个假设 g，$E_{in}(g)$ 都将接近于 $E_{out}(g)$，即学习是可行的。

（2）VC 维反映了函数集的学习能力，VC 维越大，能学到的模型越复杂。根据前面的推导，我们知道 VC 维的大小与学习算法无关，与数据集的具体分布无关，与我们求解的目标函数也无关，只与模型和假设空间有关。另外，实践中有这样一个规律：一般情况下，假设空间的 VC 维约等于假设自由变量的数目。

（3）对于（2），$VC(\mathcal{H}) = d$ 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）令 $\mathcal{H}$ 是一个有限类。那么，很显然对于任何集合 C，有 $|\mathcal{H}_C| \leq |\mathcal{H}|$，因此如果 $|\mathcal{H}| \leq 2^{|C|}$，$C$ 将不会被打散。这意味着 $VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$。也就是说，**有限 VC 维的 PAC 可学习性比有限类的 PAC 可学习性更为一般** [4]。

1.8 经验 Rademacher 复杂度、Rademacher 复杂度

本节内容一部分参考自 [2]。

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 复杂度是另一种刻画假设空间复杂度的途径，与 VC 维不同的是，它在一定程度上考虑了数据分布。

（1）考虑实值函数空间 $\mathcal{F}: \mathcal{Z} \to \mathbb{R}$，令 $Z = (z_1, z_2, ..., z_N)$，其中 $z_i \in \mathcal{Z}$（这里写得更数学，但符号也很容易具体到机器学习背景下的含义）。则函数空间 $\mathcal{F}$ 关于 Z 的**经验 Rademacher 复杂度**

$$\hat{R}_Z(\mathcal{F}) = \mathbb{E}_\sigma[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i f(z_i)], \tag{1.14}$$

其中 $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_N\}$【推导参见 [2] 的 P279-280。推导过程是以书中的二分类问题 $y_i = \pm 1$ 为例，之前出自 [2] 的相关内容也是这样的哦】。经验 Rademacher 复杂度衡量了函数空间 $\mathcal{F}$ 与随机噪声在集合 Z 中的相关性。

（2）通常我们希望了解函数空间 $\mathcal{F}$ 在 $\mathcal{Z}$ 上关于分布 $\mathcal{D}$ 的相关性。因此，对所有从 $\mathcal{D}$ 独立同分布采样而得的大小为 N 的集合 Z 求期望可得：函数空间 $\mathcal{F}$ 关于 $\mathcal{Z}$ 上分布 $\mathcal{D}$ 的 **Rademacher 复杂度**

$$R_N(\mathcal{F}) = \mathbb{E}_{Z \subseteq \mathcal{Z}: |Z| = N}[\hat{R}_Z(\mathcal{F})]. \tag{1.15}$$

结论：

（1）对于**回归问题**。基于 Rademacher 复杂度可得关于函数空间 $\mathcal{F}$ 的泛化误差界：对实值函数空间 $\mathcal{F}: \mathcal{Z} \to [0, 1]$，根据分布 $\mathcal{D}$ 从 $\mathcal{Z}$ 中独立同分布采样得到示例集 $Z = \{z_1, z_2, ..., z_N\}, z_i \in \mathcal{Z}, 0 < \delta < 1$，对任意 $f \in \mathcal{F}$，以至少 $1 - \delta$ 的概率有

$$\mathbb{E}[f(z)] \leq \frac{1}{N} \sum_{i=1}^{N} f(z_i) + 2R_N(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{2N}}, \tag{1.16}$$

$$\mathbb{E}[f(z)] \leq \frac{1}{N} \sum_{i=1}^{N} f(z_i) + 2\hat{R}_Z(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2N}}. \tag{1.17}$$

【证明参见 [2] 的 P281-282】

（2）对于**二分类问题**。对假设空间 $\mathcal{H}: \mathcal{X} \to \{-1, +1\}$，根据分布 $\mathcal{D}$ 从 $\mathcal{X}$ 中独立同分布采样得到示例集 $D = \{x_1, x_2, ..., x_N\}, x_i \in \mathcal{X}, 0 < \delta < 1$，对任意 $h \in \mathcal{H}$，以至少 $1 - \delta$ 的概率有

$$E_{out}(h) \leq \hat{E}_{in}(h) + R_N(\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{2N}}, \tag{1.18}$$

$$E_{out}(h) \leq \hat{E}_{in}(h) + \hat{R}_D(\mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{2N}}. \tag{1.19}$$

【证明参见 [2] 的 P283-284】

（3）假设空间 $\mathcal{H}$ 的 Rademacher 复杂度 $R_N(\mathcal{H})$ 与增长函数 $m_{\mathcal{H}}(N)$ 满足

$$R_N(\mathcal{H}) \leq \sqrt{\frac{2 \ln m_{\mathcal{H}}(N)}{N}}. \tag{1.20}$$

利用该式还能推导出基于 VC 维的泛化误差界【推导参见 [2] 的 P284】。

理解：

（1）为什么有了 Rademacher 复杂度还需要经验 Rademacher 复杂度呢？因为在现实任务中样例的标记有时会受到噪声影响，即某些样例 (x_i, y_i)，其 y_i 或许已经受到随机因素的影响，不再是 x_i 的真实标

记。再辞情形下，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，即考虑概率随机变量 σ_i 为随机噪声。

（2）基于 Rademacher 复杂度的泛化误差界依赖于具体学习问题上的数据分布，有点类似于为学习问题“量身定制”的，因此它通常比基于 VC 维的泛化误差界更紧一些。

1.9 学习算法的稳定性

本节内容一部分参考自 [2]。

基于 VC 维和 Rademacher 复杂度来推导泛化误差界，所得到的结果均与具体学习算法无关，对所有学习算法都适用。这使得人们能够脱离具体学习算法的设计来考虑学习问题本身的性质。另一方面，若希望获得与算法有关的分析结果，稳定性是一个值得关注的方向。它考察的是算法在输入发生变化时，输出是否会随之发生较大的变化。

对任何 $x \in \mathcal{X}, z = (x, y)$，若学习算法 $\mathcal{A}$ 满足

$$|L(\mathcal{A}_D, z) - L(\mathcal{A}_{D\setminus i}, z)| \leq \beta, i = 1, 2, ..., N, \tag{1.21}$$

则称 $\mathcal{A}$ 关于损失函数 L 满足 $\beta-$ **均匀稳定性**【符号说明参见 [2] 的 P285】。显然也有

$$|L(\mathcal{A}_D, z) - L(\mathcal{A}_{D^i}, z)| \leq 2\beta, i = 1, 2, ..., N, \tag{1.22}$$

也就是说，移除示例的稳定性包含替换示例的稳定性。

结论：

（1）若损失函数 l 有界，则有：给定从分布 $\mathcal{D}$ 上独立同分布采样得到的大小为 N 的示例集 D，若学习算法 $\mathcal{A}$ 满足关于损失函数 l 的 $\beta-$ 均匀稳定性，且损失函数 l 的上界为 $M, 0 < \delta < 1$，则对任意 $N \geq 1$，以至少 $1 - \delta$ 的概率有：

$$l_{out}(\mathcal{A}, \mathcal{D}) \leq l_{in}(\mathcal{A}, \mathcal{D}) + 2\beta + (4m\beta + M)\sqrt{\frac{\ln(1/\delta)}{2N}}, \tag{1.23}$$

$$l_{out}(\mathcal{A}, \mathcal{D}) \leq l_{loo}(\mathcal{A}, \mathcal{D}) + \beta + (4m\beta + M)\sqrt{\frac{\ln(1/\delta)}{2N}}. \tag{1.24}$$

可得，经验损失与泛化损失之间差别的收敛率为 $\beta\sqrt{N}$；若 $\beta = O(\frac{1}{m})$，则可保证收敛率为 $O(\frac{1}{\sqrt{m}})$，这与基于 VC 维和 Rademacher 复杂度得到的收敛率一致。【例题参见 [2] 的 P289 的 12.10，通过交叉验证法来估计学习算法泛化能力的合理性】

（2）若学习算法 $\mathcal{A}$ 是 ERM 且稳定的，则假设空间 $\mathcal{H}$ 是可学习的【证明参见 [2] 的 P286-287】。

理解：

（1）稳定性分析不必考虑假设空间中所有可能的假设，只需根据算法自身的特点（稳定性）来讨论输出假设 $\mathcal{A}_D$ 的泛化误差界。

（2）对于结论（2），学习算法和假设空间事实上并非无关，由稳定性的定义可知，两者可以通过损失函数 l 联系起来。

1.10 ϵ-代表性样本、一致收敛

现在开始将视线转移到样本数上，因为前面有提到 $N \geq \text{poly}(1/\epsilon, 1/\delta, \text{size}(x), \text{size}(c))$ 这个式子，而且在结论中也常提到样本足够大，因此这里开始对其进行量化研究。本节内容部分来自 [4]，**以下内容应用于有限假设类**。

我们希望关于样本 $\mathcal{D}$ 的可以最小化经验风险的 h 也是一个关于真实数据概率分布的风险最小化（或者是风险接近最小化）。那么，它足以保证 $\mathcal{H}$ 中的所有元素的经验风险是它们真实风险的一个很好的近似。换句话说，我们需要假设类中所有的假设都是一致的，经验风险将会接近真实风险，表达式如下所示。

（1）如果满足下列不等式：在一个训练集 $\mathcal{D}$ 上，$\forall h \in \mathcal{H}, |E_{out}(h) - E_{in}(h)| \leq \epsilon$，就称该训练集为 **$\epsilon$-代表性样本**。

（2）如果一个假设类 $\mathcal{H}$ 满足如下条件，那么它就有**一致收敛**性质：存在一个函数 $N_{\mathcal{H}}^{UC}:(0,1)^2\to\mathbb{N}$ 使得对于所有 $\epsilon,\delta\in(0,1)$ 和在 $\mathcal{X}\times\mathcal{Y}$ 上的所有概率分布，如果 $\mathcal{D}$ 是从中得到的一个独立同分布的满足 $N\geq N_{\mathcal{H}}^{UC}(\epsilon,\delta)$ 的样本，那么至少在概率 $1-\delta$ 下，$\mathcal{D}$ 是 ϵ-代表性的。

“一致性”在这里指的是在 $\mathcal{X}\times\mathcal{Y}$ 中所有可能的概率分布下，用于所有 $\mathcal{H}$ 中的元素，有一个固定的样本大小。$N_{\mathcal{H}}^{UC}(\epsilon,\delta)$ 度量了获得一致收敛性质的（最小）样本复杂度，即我们需要多少样本来确保至少在概率 $1-\delta$ 下，样本是 ϵ-代表性的。

结论：

（1）假设一个训练集 $\mathcal{D}$ 是 $\epsilon/2$-代表性的。那么，任何一个 $h_{\mathcal{D}}\in\arg\min_{h\in\mathcal{H}}E_{in}(h)$ 都满足 $E_{out}(h_{\mathcal{D}})\leq\min_{h\in\mathcal{H}}E_{out}(h)+\epsilon$。该结论说明了只要一个样本是 $\epsilon/2$-代表性的，就可以保证 ERM 学习规则返回一个好的假设【证明参见 [4] 的 P24，用放缩法】。

这个结论表明为了确保 ERM 规则是一个不可知的 PAC 学习器，应该满足至少在概率 $1-\delta$ 下随机选择一个训练集，它将是 ϵ-代表性训练集。上述（2）中的一致收敛条件形式化了这个条件。

（2）如果类 $\mathcal{H}$ 关于函数 $N_{\mathcal{H}}^{UC}$ 有一致收敛的性质，那么这个类是样本复杂度为 $N_{\mathcal{H}}(\epsilon,\delta)\leq N_{\mathcal{H}}^{UC}(\epsilon/2,\delta)$ 的不可知 PAC 可学习的。而且，在那种情况下，$ERM_{\mathcal{H}}$ 范式是关于 $\mathcal{H}$ 的成功的不可知 PAC 可学习的。

也就是说：**一致收敛是可学习的充分条件**。只要我们确定对于一个**有限**假设类，一致收敛成立，那么每个有限假设类都是不可知 PAC 可学习的。为了说明一致收敛成立，固定 ϵ,δ，我们需要找到一个样本大小 N 可以保证下面的条件成立：对于任何分布，至少在概率 $1-\delta$ 下，从中采样得到的独立同分布的样本选择，对于所有 $h\in\mathcal{H}$，$|E_{out}(h)-E_{in}(h)|\leq\epsilon$ 成立。我们可以选择 $N\geq\dfrac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}$【证明参见 [4] 的 P25-26，联合界：对于任意的集合 A 和 B 以及分布 $\mathcal{D}$，有 $\mathcal{D}(A\cup B)\leq\mathcal{D}(A)+\mathcal{D}(B)$、大数定理、Hoeffding 测度集中度不等式】。

（3）令 $\mathcal{H}$ 是一个有限假设类，并且令 $L:\mathcal{X}\times\mathcal{Y}\times\mathcal{H}\to[0,1]$ 是一个损失函数。那么，$\mathcal{H}$ 具有一致收敛性质，而且样本复杂度是 $N_{\mathcal{H}}^{UC}(\epsilon,\delta)\leq\lceil\dfrac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}\rceil$。而且，用 ERM 算法，这个类是不可知 PAC 可学习的，样本复杂度是 $N_{\mathcal{H}}(\epsilon,\delta)\leq N_{\mathcal{H}}^{UC}(\epsilon/2,\delta)\leq\lceil\dfrac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2}\rceil$。

如果损失函数的范围是 $[a,b]$，那么样本复杂度满足：$N_{\mathcal{H}}(\epsilon,\delta)\leq N_{\mathcal{H}}^{UC}(\epsilon/2,\delta)\leq\lceil\dfrac{2\log(2|\mathcal{H}|/\delta)(b-a)^2}{\epsilon^2}\rceil$。

1.11 统计学习的基本定理、统计学习的基本定理——定量形式

本节内容部分来自 [4]，**以下内容应用于无限假设类**。

从上一节中我们可以知道：有限类是可学习的，且在这种情况下，假设类的样本复杂度上界由假设类大小的对数决定。对于无限大小的假设类而言，假设类的大小不是一个可用于描述样本复杂度的特征，即某些无限大小的假设类也是可学习的。简而言之，**虽然假设类 $\mathcal{H}$ 有限时可学习性的充分条件，但它并不是一个必要条件**。如下例子所示：令 $\mathcal{H}$ 是实线上阈值函数构成的集合，即 $\mathcal{H}=\{h_a:a\in\mathbb{R}\}$，其中 $h_a:\mathbb{R}\to\{0,1\}$ 是一个函数，使得 $h_a(x)=1_{[x<a]}$。显然该 $\mathcal{H}$ 是无限大小的。如下引理表明 $\mathcal{H}$ 在 PAC 模型下采用 ERM 算法是可学习的：令 $\mathcal{H}$ 为如之前定义的阈值函数。那么 $\mathcal{H}$ 在采用 ERM 规则时是 PAC 可学习的，其样本复杂度 $m_{\mathcal{H}}(\epsilon,\delta)\leq\lceil\log(2/\delta)\epsilon\rceil$【证明参见 [4] 的 P33-34】。

（1）**统计学习的基本定理**：令 $\mathcal{H}$ 是一个由从 $\mathcal{X}$ 到 $\{0,1\}$ 的映射函数构成的假设类，且令损失函数为 0-1 损失。那么下列陈述等价：(a) $\mathcal{H}$ 有一致收敛性；(b) 任何 ERM 规则都是对于 $\mathcal{H}$ 成功的不可知 PAC 学习器；(c) $\mathcal{H}$ 是不可知 PAC 可学习的；(d) $\mathcal{H}$ 是 PAC 可学习的；(e) 任何 ERM 规则都是对于 $\mathcal{H}$ 成功的 PAC 学习器；(f) $\mathcal{H}$ 的 VC 维有限【证明参见 [4] 的 P37-38 和 P39】。（正如1.7中已经说到的那样，有限 VC 维的 PAC 可学习性比有限类的 PAC 可学习性更为一般，所以该定理中我们用 VC 维来描述 PAC 可学习性）

（2）VC 维还可以决定样本复杂度。**统计学习的基本定理——定量形式**：令 $\mathcal{H}$ 是一个由 $\mathcal{X}$ 到 $\{0,1\}$ 的映射函数构成的假设类，且令损失函数为 0-1 损失。假定 $VC(\mathcal{H})=d<\infty$，那么，存在绝对常数 C_1,C_2 使得：(a) $\mathcal{H}$ 有一致收敛性，若其样本复杂度满足 $C_1\dfrac{d+\log(1/\delta)}{\epsilon^2}\leq N_{\mathcal{H}}^{UC}(\epsilon,\delta)\leq C_2\dfrac{d+\log(1/\delta)}{\epsilon^2}$；(b) $\mathcal{H}$ 是不可知 PAC 可学习的，若其样本复杂度满足 $C_1\dfrac{d+\log(1/\delta)}{\epsilon^2}\leq N_{\mathcal{H}}(\epsilon,\delta)\leq C_2\dfrac{d+\log(1/\delta)}{\epsilon^2}$；(c) $\mathcal{H}$ 是

PAC 可学习的，若其样本复杂度满足 $C_1\frac{d+\log(1/\delta)}{\epsilon} \leq N_{\mathcal{H}}(\epsilon,\delta) \leq C_2\frac{d\log(1/\epsilon)+\log(1/\delta)}{\epsilon}$。

结论：

（1）如果 $\mathcal{H}$ 有小的有效规模，那么 $\mathcal{H}$ 有一致收敛性，正式的表达如下：令 $\mathcal{H}$ 是一个类，令 $m_{\mathcal{H}}$ 为其生长函数。那么，对于每个 $\mathcal{D}$ 以及每个 $\delta \in (0,1)$，对于任意 $S \sim \mathcal{D}^m$，都以至少 $1-\delta$ 的概率有下式成立：$|E_{out}(h) - E_{in}(h)| \leq \frac{4+\sqrt{\log(m_{\mathcal{H}}(2N))}}{\delta\sqrt{2N}}$ 【证明参见 [4] 的 P39-40】。

【证明参见 [4] 的第 28 章 P263 起】

上述基本定理是针对二分类问题的，**对于其他学习问题，如采用绝对值损失或平方损失的回归问题也能得到类似结果**。然而，该定理并不是对于所有的学习问题都成立【例子参见 [4] 的 P37 索引】。

1.12 “没有免费的午餐”定理、先验知识、逼近误差、估计误差、偏差-复杂度权衡

本节内容部分来自 [4]。

（1）**“没有免费的午餐”定理**：对实例空间 $\mathcal{X}$ 上 0-1 损失的二分任务，令 $\mathcal{A}$ 表示任意的学习算法。样本大小 N 表示小于 $|\mathcal{X}|/2$ 的任意数。则在 $\mathcal{X} \times \{0,1\}$ 上存在一个分布，使得：（1）存在一个函数 $f:\mathcal{X} \to \{0,1\}$ 满足 $E(f)=0$；（2）在从该分布中独立重复抽取的大小为 N 的样本集 $\mathcal{D}$ 上，以至少 $\frac{1}{7}$ 的概率满足 $E(A(\mathcal{D})) \geq \frac{1}{8}$ 【证明参见 [4] 的 P29-30】。

该定理证明不存在通用的学习器，即通过证明没有学习器能在所有的任务上学习成功。这个定理陈述的是，对于每个学习器，都存在一个任务使其失败，即便这个任务能够被另一个学习器成功学习（一个特定的学习任务由 $\mathcal{X} \times \mathcal{Y}$ 上的一个未知分布 $\mathcal{D}$ 所定义）。

（2）从“没有免费的午餐”定理中，我们可以知道，从假设类 $\mathcal{H}$ 中选择输出假设的任意算法，尤其是 ERM 预测器，都存在着某个任务使其学习失败（下面的结论（1）对此给出了形式化描述）。因为先验知识的缺失，使得从域到标签集上的每个函数都看成是一个好的候选。为了避免这样的失败，我们可以利用特定学习任务的**先验知识**，结合“没有免费的午餐”定理，来预见并脱离这样的困境，从而避免学习任务时会导致失败的分布。这样的先验知识可以通过限制假设类来表示。

（3）我们将一个 $ERM_{\mathcal{H}}$ 预测器的误差分解为两部分，令 h 为一个 $ERM_{\mathcal{H}}$ 假设，则写作 $E(h) = \epsilon_{app} + \epsilon_{est}$，其中 $\epsilon_{app} = \min_{h' \in \mathcal{H}} E(h'), \epsilon_{est} = E(h) - \epsilon_{app}$。其中，**逼近误差** ϵ_{app} 表示假设类里预测器所取得的最小风险。这一项刻画由于限制到一个具体假设类所引起的风险，即所产生的的**归纳偏置**。逼近误差不依赖于样本大小而依赖于我们的先验知识（反映在对假设类 $\mathcal{H}$ 的选择）与潜在的未知分布是否吻合，取决于所选择的假设类。扩大假设类可以减小逼近误差。**估计误差**表示逼近误差与 ERM 预测器误差之间的差异。估计误差的产生是因为：经验风险（即训练误差）只是真实风险的一个估计，所以最小化经验风险预测器只是最小化真实风险预测器的一个估计。

预测器的估计好坏取决于样本集大小和假设类的大小或复杂度。如前所示，对一个有效假设类，ϵ_{est} 随 $\mathcal{H}$（以对数方式）递增，随 m 递减。我们可以将 $\mathcal{H}$ 的大小作为其复杂度的一种衡量。

（4）由于目标是将总风险最小化，因此我们面临着一个权衡，称为**偏差-复杂度权衡**。一方面，选择一个丰富的假设类作为 $\mathcal{H}$ 会导致**过拟合**，使得逼近误差减小的同时估计误差增大。另一方面，选择一个较小的假设类作为 $\mathcal{H}$，会导致估计误差减小的同时逼近误差增大，换言之会**欠拟合**。**这就是我们在利用特定的学习任务的先验知识来限定和选择假设类时所面临的一个权衡**。

学习理论研究的是我们如何使得 $\mathcal{H}$ 丰富的同时依然保持合理的估计误差。

结论：

（1）令 $\mathcal{X}$ 为一个无限定义域集，$\mathcal{H}$ 为从 $\mathcal{X}$ 到 $\{0,1\}$ 上的所有映射集，则 $\mathcal{H}$ 不是 PAC 可学习的【证明参见 [4] 的 P30】。

理解：

到目前为止的上述内容都是基于经验风险最小化（ERM，定义见1.2），即认为：带来经验风险最小的假设 h 所对应的模型就是最优模型，例如极大似然估计就是一个典型例子，其损失函数是对数损失函数。当样本容量足够大时，经验风险最小化能保证比较好的学习效果。然而，当训练样本集 $\mathcal{D}$ 的容量很小时，经验风险最小化学习的效果未必很好，容易产生过拟合现象。

相较于就此抛弃 ERM 范例，我们更倾向于寻找方法来修正它，即寻找保证 ERM 不会导致过拟合的条件。通常的解决方案是在一个受限的搜索空间中使用 ERM 学习准则。通过限制 $\mathcal{A}$ 从 $\mathcal{H}$ 中选择 h，我们的选择偏向于一个特别的假设集合。这种限制通常称为“归纳偏置”。因为这种选择决定于 $\mathcal{A}$ 接触训练数据之前，所以它应该基于一些需要学习问题的“先验知识”。在学习理论中，一个基本的问题是：选择哪种假设类不会导致过拟合。直观上，选择一个更加严格受限的假设类能够更好地防止过拟合，但与此同时，也会带来更强的归纳偏置。

1.13 竞争、不一致可学习、权重函数、结构风险最小化、最小描述长度、奥卡姆剃须刀、一致收敛性

本节内容部分来自 [4]。下述内容关于如何在假设空间中根据先验知识来限定针对特定学习任务的假设类。

前面所讨论的 PAC 可学习的概念是考虑依据**精度**和**置信参数**来决定样本数量（参见1.10和1.11），**前提条件是样本标签分布与内在的样本数据分布是一致的。因此，类别可学习是有条件的，样本必须具有有限的 VC 维**。这一节将考虑更松的、更弱化约束条件下可学习的概念。**这个概念下允许样本数量依赖于学习器所在假设空间而变化**。

（1）一个假设 h 以 (ϵ,δ) 可与另一个假设 h' **竞争**，如果下式成立的概率不少于 $1-\delta$，$E_{out}(h)\leq E_{out}(h')+\epsilon$。

（2）若存在一个学习算法 $\mathcal{A}$ 和一个函数 $N_{\mathcal{H}}^{NUL}:(0,1)^2\times\mathcal{H}\to\mathbb{N}$，使得对任意的 $\epsilon,\delta\in(0,1)$ 和 $h\in\mathcal{H}$，如果样本数量 $N\geq N_{\mathcal{H}}^{NUL}(\epsilon,\delta,h)$，那么对每个分布 $\mathcal{D}$ 和所有的样本 $S\sim\mathcal{D}^N$，下列成立的概率不少于 $1-\delta$，$E_{out}(A(S))\leq E_{out}(h)+\epsilon$，则假设类 $\mathcal{H}$ 是**不一致可学习**的。

（3）前面讲过利用先验知识来限制假设类的搜索范围。另一种利用先验知识的方式是将假设类 $\mathcal{H}$ 上的偏好具体化。具体做法为：首先假定 $\mathcal{H}$ 能写成 $\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$（例如：$\mathcal{H}$ 可能是所有多项式分类器构成的类，$\mathcal{H}_n$ 表示 n 次多项式分类器构成的类），然后具体化一个**权重函数** $\omega:\mathbb{N}\to[0,1]$，这个权重函数给每个假设类赋予一个权重，高的权值表示对该假设类的强烈偏好。权重函数可以反映每个假设类学习属性的**重要性**，或者不同假设类**复杂性的度量**【例子参见 [4] 的 P46】。如果你认为某个假设类更有可能包含正确的目标函数，就可以给该假设类赋予较大的权重来反映这种先验知识。

（4）**结构风险最小化**是一种“最小化界”的方法，也就是要寻找一个假设类来最小化真实风险的上确界。可以由下列定理给出：

令 $\omega:\mathbb{N}\to[0,1]$ 是一个权值函数，满足 $\sum_{n=1}^{\infty}\omega(n)\leq 1$。$\mathcal{H}$ 是一个假设类可以写成 $\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，对于任一个 n，$\mathcal{H}_n$ 满足一致收敛性，并且复杂度表示函数为 $N_{\mathcal{H}_n}^{UC}$，令 ϵ_n 由 $\epsilon_n(N,\delta)=\min\{\epsilon\in(0,1):N_{\mathcal{H}_n}^{UC}(\epsilon,\delta)\leq N\}$ 定义。然后，对于任一个 $\delta\in(0,1)$，样本 $S\sim\mathcal{D}^N$，对于任一个 $n\in\mathbb{N}$ 和 $h\in\mathcal{H}_n$，下式成立的概率不低于 $1-\delta$，$|E_{out}(h)-E_{in}(h)|\leq\epsilon_n(N,\omega(n)\cdot\delta)$。则对于任一个 $\delta\in(0,1)$ 和分布 $\mathcal{D}$，下列成立的概率不低于 $1-\delta$，$\forall h\in\mathcal{H},E_{out}(h)\leq E_{in}(h)+\min_{n:h\in\mathcal{H}_n}\epsilon_n(N,\omega(n)\cdot\delta)$，结构化风险最小化寻找假设 h 来最小化这个界【证明和范式参见 [4] 的 P47】。

与 ERM 不同，我们不仅关心经验风险 $E_{in}(h)$，而且为了最小化估计误差，更加关心在最小经验风险的偏置和 $\epsilon_{n(h)}(N,\omega(n(h))\cdot\delta)$ 最小化之间取得一个平衡。

（5）有一个假设类，我们想要知道如何描述和表示每一个类中的假设。

首先来形式化这些概念：令 $\mathcal{H}$ 是我们要描述的假设类，定义有限符号集合 $\Sigma=\{0,1\}$ 称之为字母表，一个字符串 σ（如 $\sigma=(0,1,1,1,0)$）是 Σ 中的有限符号序列。所有有限字符串的集合用 Σ^* 表示。对 $\mathcal{H}$ 的**描述语言**用一个函数 $d:\mathcal{H}\to\Sigma^*$，将 $\mathcal{H}$ 中每一个假设 h 映射为一个字符串 $d(h)$。$d(h)$ 称为 h 的**描述长度**，并且 h 的描述长度用 $|h|$ 表示。我们要求描述语言**无前缀**，也就是说不同的 h,h'，$d(h)$ 不是 $d(h')$ 的前缀。

无前缀的字符串集合满足 **Kraft 不等式**：如果 $S\subseteq\{0,1\}^*$ 是一个无前缀的字符串的集合，则 $\sum_{\sigma\in S}\frac{1}{2^{|\sigma|}}\leq 1$【证明参见 [4] 的 P49】。

根据上述不等式，任何假设 $\mathcal{H}$ 的无前缀描述语言都能给出假设类 $\mathcal{H}$ 的权重函数 ω，我们可以简单地设置为 $\omega(h)=\frac{1}{2^{|h|}}$。可直接得到以下结论：

令 $\mathcal{H}$ 是一个假设类，$d:\mathcal{H}\to\{0,1\}^*$ 是 $\mathcal{H}$ 的一个无前缀描述语言。对于样本数量 N，置信度参数 $\delta>0$ 和概率分布 $\mathcal{D}$，样本 $S\sim\mathcal{D}^N$，下式成立的概率大于 $1-\delta$：$\forall h\in\mathcal{H}, E_{out}(h)\le E_{in}(h)+\sqrt{\frac{|h|+\ln(2/\delta)}{2N}}$，这里 $|h|$ 是指 $d(h)$ 的长度【证明参见 [4] 的 P49】。

上述结果给出了对于训练集 S，搜索假设 $h\in\mathcal{H}$ 最小化界 $E_{in}(h)+\sqrt{\frac{|h|+\ln(2/\delta)}{2N}}$ 的 $\mathcal{H}$ 的一个学习范式。具体说，这种方法折中考虑了经验风险和减少描述长度，即**最小描述长度**（MDL）范式【范式参见 [4] 的 P49】。

（6）上述结果还指出，对于经验风险相同的两个假设，最小描述长度较小的假设，其真实风险的风险误差界更小。因此，这个结果表达了一个哲学理念：短的解析（也就是长度短的解析）比长的解析更有效，称之为**奥卡姆剃刀**。也就是说，假设 h 越复杂（在这里就是描述长度越长），就需要更多的样本来保证真实风险 $E_{out}(h)$。该定理仍存在一些 bug，参见 [4] 的 P50。

（7）**一致收敛性**：令 Z 表示一种域的集合，$\mathcal{P}$ 表示 Z 上的概率分布，$\mathcal{H}$ 表示假设类。若存在一个函数 $m_{\mathcal{H}}^{CON}:(0,1)^2\times\mathcal{H}\times\mathcal{P}\to\mathbb{N}$ 使得对于任意一个 $h\in\mathcal{H},\mathcal{D}\in\mathcal{P},\epsilon,\delta\in(0,1)$。如果 $m\ge m_{\mathcal{H}}^{CON}(\epsilon,\delta,h,\mathcal{D})$，样本 $S\sim\mathcal{D}^N$，下式成立的概率不低于 $1-\delta$，$E_{out}(A(S))\le E_{out}(h)+\epsilon$，我们就认为一个学习规则 $\mathcal{A}$ 关于 $\mathcal{H}$ 和 $\mathcal{P}$ **一致收敛**。如果 $\mathcal{P}$ 是所有分布的集合，则我们说 $\mathcal{A}$ **全局收敛**到 $\mathcal{H}$。

一致收敛性的概念是不一致可学习概念的进一步松弛，即所需样本数量还依赖于概率分布 $\mathcal{D}$（$\mathcal{D}$ 用于产生训练样本和决定风险）。显然，如果一个算法能不一致可学习一个类 $\mathcal{H}$，那么它一定全局一致收敛到类 $\mathcal{H}$【例子参见 [4] 的 P51】。

结论：

（1）结合之前讲过的“可学习条件”等，有如下结论：如果 $\mathcal{H}$ 是有限的，$\mathcal{A}$ 将不会过拟合，前提是拥有足够多的训练样本（其大小依赖于 $\mathcal{H}$ 的势 [4]，这里的“势”应该就是“大小”的意思，结论为 $N\ge\frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$）【证明参见 [4] 的 P14-15，注意：$\epsilon-\delta$ 不等式、“差”的假设集合、误导集、联合界、推论 2.3 的数学描述，其中有个“可实现性假设”（也就是可分性）的前提】。前面各小节在分析算法的泛化误差界时，得出的结论也是泛化误差界主要与训练样本的数目相关。

（2）二分类器的假设类 $\mathcal{H}$ 是不一致可学习的当且仅当它是不可知 PAC 可学习假设类的可数并。

（3）令一个假设类 $\mathcal{H}$ 能够写成假设类的可数并，$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，如果 $\mathcal{H}_n$ 是一致收敛的，那么 $\mathcal{H}$ 是不一致可学习的【证明参见 [4] 的 P45】。

（4）令 $\mathcal{H}$ 是假设类，满足 $\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，$\mathcal{H}_n$ 满足一致收敛性，并且复杂度表示函数为 $N_{\mathcal{H}_n}^{UC}$，如果 $\omega:\mathbb{N}\to[0,1]$ 满足 $\omega(n)=\frac{6}{\pi^2n^2}$，那么，$\mathcal{H}$ 是不一致可学习的，结构风险最小化率为 $m_{\mathcal{H}}^{NUL}(\epsilon,\delta,h)\le m_{\mathcal{H}_{n(h)}}^{UC}(\epsilon/2,\frac{6\delta}{(\pi n(h))^2})$【证明参见 [4] 的 P47-48】。

理解：

（1）不一致可学习与 PAC 可学习的区别：（a）在 PAC 可学习中，我们寻找具有绝对的最小风险的假设（在可能的情况下）或者寻找一个与最小风险差不多风险（在绝对最小风险不可知情况下）的假设，样本数仅仅依赖于精度和置信度。然而，在不一致学习中，我们允许样本数量以 $m_{\mathcal{H}}(\epsilon,\delta,h)$ 的形式表示，也就是说，不一致可学习在表现形式上也依赖竞争力变量 h。（b）与不可知条件下 PAC 可学习 $E_{out}(A(S))\le\min_{h'\in\mathcal{H}}E_{out}(h')+\epsilon$ 相比，两类可学习中要求输出假设与在假设类中的其他假设相比具有 (ϵ,δ) 竞争力，但两类假设也有区别：在不一致可学习中，样本数依赖于 $A(S)$ 错误对应的假设 h，而不可知条件下 PAC 可学习不依赖于 h。（c）不一致可学习比 PAC 可学习对假设条件要求更少，也就是说，如果一个假设类是不可知条件下 PAC 可学习，那么它也是不一致可学习的。

（2）不一致可学习是不可知条件下 PAC 可学习的严格松弛。也就是说，存在假设类是不一致可学习的，但不是不可知条件下 PAC 可学习的【例子参见 [4] 的 P45 例 7.1】。

1.14 三种可学习概念的比较

内容提炼自 [4] 的 P51-53。

	PAC 可学习	不一致可学习	一致收敛性
学习假设的风险 （输出预测风险界）	基于经验风险给出了学习假设的真实风险上界		没有提供界，但通常可以使用验证集来估计输出预测器的风险（后面会讲）
得到最好假设需要的样本数	直接给出了答案 但不能保证一个干脆的答案 依赖于先验知识的质量	没有给出答案 不一致学习中样本数依赖于最好的假设 一致收敛性中样本数还依赖于数据潜在的分布	
如何表达先验	通过假设类的选择给出了应用先验知识的直接方式	在假设类或它的子集上定义权重	没有编码利用先验的方式
如何学习 （学习理论所提供的最有用的方面，假设类选定后的通用学习规则）	经验风险最小化	结构风险最小化	没有自然的学习范式

【例子参见 [4] 的 P52-53】

理解：

（1）PAC 保证也能帮助我们理解，当一个学习算法返回一个大风险假设时我们下一步应该怎么做，这是因为我们对部分错误建立一个界，这个界来源于对误差的估计，因此知道有多少错误造成了近似误差。如果一个假设的误差很大，我们知道应该使用一个不同的假设。同样地，如果一个不一致学习算法失败，我们可以考虑在假设类上使用不同的权重函数。然而，当一个一致收敛算法失败，我们不知道这是由于估计误差还是近似误差造成的，甚至，即使我们确定问题是由估计误差造成的，我们也不能确定需要多少样本可以得到估计误差变小。

（2）我们偏好什么样的学习算法？参见 [4] 的 P52-53。

1.15 凸集、凸组合、凸函数、上镜图、局部最小值、全局最小值、利普希茨性、光滑性、自有界函数

我们知道一些常见的凸问题和非凸问题，例如，具有平方损失和逻辑斯蒂回归的线性回归问题都是凸问题，对于交叉熵损失，在 logistic 回归中是凸函数，在 MLP 中是非凸函数。半空间的 0-1 损失问题是非凸问题。通常，一个凸学习问题的**假设类是一个凸集**，并且对于每一个样本而言，它的**损失函数是一个凸函数**。下面依次给出基本概念的定义描述。

（1）设 C 是向量空间的一个集合，若对 C 中任意两点 u 和 v，连接它们的线段仍在 C 中，那么集合 C 是一个**凸集**。换言之，对任一实数 $\alpha \in [0,1]$，都有 $\alpha u + (1-\alpha)v \in C$【例子参见 [4] 的 P93】。给定 $\alpha \in [0,1]$，$\alpha u + (1-\alpha)v$ 称为 u 和 v 的**凸组合**。

（2）设 C 是一个凸集，如果对任意的 $u, v \in C$ 及 $\alpha \in [0,1]$，函数 $f : C \to \mathbb{R}$ 满足 $f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v)$，则称 f 为 C 上的**凸函数**。换句话说，对于任意 u 和 v，如果函数 f 在 u 和 v 之间的图形位于连接 $f(u)$ 和 $f(v)$ 的线段的下方，那么 f 是凸函数。（由此应该注意：长得“凸”的不一定就是凸函数，比如像山一样的那种凸。由定义，因为要保证最小值，所以需要将其限定为像峡谷一样的那种凸（这里就不管叫凹哦））

（3）函数 f 的**上镜图**是集合 $\text{epigraph}(f) = \{(x, \beta) : f(x) \leq \beta\}$。

（4）设 $B(u,r) = \{v : \|v - u\| \leq r\}$ 是一个以 u 为球心 r 为半径的球。如果存在某个 $r > 0$ 使得对任意的 $v \in B(u,r)$ 都有 $f(v) \geq f(u)$，那么我们就说 $f(u)$ 是 f 在 u 处的一个**局部最小值**。如果该式对每一个 v 都成立，则 $f(u)$ 是 f 的**全局最小值**【证明：凸函数的每一个局部极小值也是全局最小值，参见 [4] 的 P94】。

（5）设 $C \subset \mathbb{R}^d$，$f : \mathbb{R}^d \to \mathbb{R}^k$，如果对于任意函数的 $\omega_1, \omega_2 \in C$，有 $\|f(\omega_1) - f(\omega_2)\| \leq \rho\|\omega_1 - \omega_2\|$，那么 f 是 ρ-**利普希茨**。如果 f 的导数按绝对值处处以 ρ 为界，那么函数 f 是 ρ-利普希茨【证明和例子参见 [4] 的 P96-97】。直观地说，一个利普希茨函数不会变化太快。

（6）如果可微函数 $f:\mathbb{R}^d\to\mathbb{R}$ 的梯度是 β-利普希茨，即对于所有的 v,ω，满足 $\|\nabla f(v)-\nabla f(\omega)\|\le\beta\|v-\omega\|$，那么 f 是 β-**光滑**。进一步，假设对于所有的 v 有 $f(v)\ge 0$，那么可以推断光滑性也意味着 $\|\nabla f(\omega)\|^2\le 2\beta f(\omega)$，满足这个性质的函数也称为**自有界函数**【推导和例子参见 [4] 的 P97-98】。

结论：

（1）函数 f 是凸的当且仅当它的上镜图是一个凸集。

（2）对于凸函数 f 的每一个 ω，我们可以构造 f 在 ω 处的切线，该切线始终位于函数 f 的下方。如果 f 可微，则有 $\forall u, f(u)\ge f(\omega)+\langle\nabla f(\omega),u-\omega\rangle$。[4] 的 14 章将该不等式推广至不可微函数。

（3）设 $f:\mathbb{R}\to\mathbb{R}$ 是一个二阶可微的标量函数，f' 和 f'' 分别表示函数 f 的一阶导和二阶导函数，那么下面的命题是等价的：（a）f 是凸的；（b）f' 是单调不减的；（c）f'' 是非负的【例子参见 [4] 的 P95】。

（4）假设对于某个 $x\in\mathbb{R}^d, y\in\mathbb{R}$ 和 $g:\mathbb{R}\to\mathbb{R}$，函数 $f:\mathbb{R}^d\to\mathbb{R}$ 可以写成 $f(\omega)=g(\langle\omega,x\rangle+y)$，那么 g 的凸性蕴含着 f 的凸性【证明和例子参见 [4] 的 P95】。意思就是：一个凸标量函数和一个线性函数的组合得到的是一个凸向量值函数。

（5）设 $f_i:\mathbb{R}^d\to\mathbb{R}$ 是凸函数，$i=1,2,...,r$，那么下面定义在 $\mathbb{R}^d$ 上的实函数也都是凸函数。（a）$g(x)=\max_{i\in[r]}f_i(x)$；（b）$g(x)=\sum_{i=1}^{r}\omega_i f_i(x)$，其中对于任意的 i，$\omega_i\ge 0$【证明和例子参见 [4] 的 P96】。意思就是：凸函数的最大化是凸的；加权的凸函数的和也是凸的。

（6）设 $f(x)=g_1(g_2(x))$，其中 g_1 是 ρ_1-利普希茨，g_2 是 ρ_2-利普希茨，那么 f 是 $(\rho_1\rho_2)$-利普希茨。特别地，如果 g_2 是线性函数，对于 $v\in\mathbb{R}^d, b\in\mathbb{R}, g_2(x)=\langle v,x\rangle+b$，那么 f 是 $(\rho_1\|v\|)$-利普希茨【证明参见 [4] 的 P97】。意思就是：利普希茨函数的组合仍具有利普希茨性。

（7）设 $f(\omega)=g(\langle\omega,x\rangle+b)$，其中函数 $g:\mathbb{R}\to\mathbb{R}$ 是 β-光滑，$x\in\mathbb{R}^d, b\in\mathbb{R}$，那么 f 是 $(\beta\|x\|^2)$-光滑【证明和例子参见 [4] 的 P98】。意思就是：一个光滑的标量函数在线段函数上的组合仍具有光滑性。

1.16 凸学习问题、凸利普希茨有界学习问题、凸光滑有界学习问题、替代损失函数

本节内容部分来自 [4]。之前我们把 $\mathcal{H}$ 考虑为从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的函数的集合。在这里，我们考虑 $\mathcal{H}$ 为欧几里得空间 $\mathcal{R}^d$ 的子集。也就是说，每个假设是某个实值向量。所以，我们可以将 $\mathcal{H}$ 记为 ω。

（1）令 $Z=\mathcal{X}\times\mathcal{Y}$，如果假设类 $\mathcal{H}$ 是凸集，且对于任意的 $z\in Z$，损失函数 $L(\cdot,z)$ 是凸函数，那么学习问题 $(\mathcal{H},Z,L)$ 是凸的，即为**凸学习问题**。这里，对于任意的 z，$L(\cdot,z)$ 表示由 $f(\omega)=L(\omega,z)$ 定义的函数 $f:\mathcal{H}\to\mathbb{R}$【例子参见 [4] 的 P98，具有平方损失的线性回归】。

（2）从 [4] 的 P100 的例 12.9 看出，有界性和凸性仍不能保证问题是可学习的。说明对于学习问题我们需要其他的一些假设条件，这次的解决方法是假设损失函数具有利普希茨性或光滑性。这就促使我们给出了下面两类学习问题的定义。

如果假设类 $\mathcal{H}$ 是一个凸集，且对于所有的 $\omega\in\mathcal{H}$ 都成立 $\|\omega\|\le B$；对于所有的 $z\in Z$，损失函数 $L(\cdot,z)$ 是凸的且是 ρ-利普希茨，则称学习问题 $(\mathcal{H},Z,L)$ 是凸利普希茨有界的，即**凸利普希茨有界学习问题**。其中，ρ,B 是参数【例子参见 [4] 的 P100】。

（3）如果假设类 $\mathcal{H}$ 是一个凸集且对于所有的 $\omega\in\mathcal{H}$ 都成立 $\|\omega\|\le B$；对于所有的 $z\in Z$，损失函数 $L(\cdot,z)$ 是凸的、非负的且是 β-光滑，则称学习问题 $(\mathcal{H},Z,L)$ 是凸光滑有界的，即**凸光滑有界学习问题**。其中，ρ,B 是参数【例子参见 [4] 的 P100-101】。注意到我们要求损失函数是非负的，这是为了保证损失是自有界的。

（4）在许多情况下，自然的损失函数不是凸的，特别地，实施 ERM 准则是困难的【例子参见 [4] 的 P101】。一个流行的方法是通过一个凸的替代损失函数来定义非凸损失函数的上界。正如这个名字所指示的，一个**替代损失函数**需要满足：（1）它是凸的；（2）它是原来损失函数的一个上界【例子参见 [4] 的 P101-102，在学习半空间的情况下，定义所谓的合页损失作为 0-1 损失的凸替代】。

结论：

（1）如果损失函数 L 是凸函数，假设类 $\mathcal{H}$ 是凸集，那么 $ERM_{\mathcal{H}}$ 问题（在 $\mathcal{H}$ 上极小化经验损失）是一个凸优化问题；也相当于在一个凸集上极小化一个凸函数【证明参见 [4] 的 P99】。

（2）不是 $\mathbb{R}^d$ 上所有的凸学习问题都是可学习的（这与 VC 维理论并不矛盾，因为 VC 维理论只解决二分类问题，而这里我们考虑的是一类更广泛的问题。这和“离散技巧”也不矛盾，因为我们假设损失是

有界的，同时假设用有限数量的位来表示每个参数就足够了）【例子参见 [4] 的 P99-100。“线性回归的不可学习性”，说明了在许多实际情况中，如果添加一些额外的约束条件，那么凸问题是可学习的。】。

（3）我们断言凸利普希茨有界学习问题和凸光滑有界学习问题是可学习的。也就是说，**损失函数具有凸性、有界性和利普利茨性或光滑性是可学习的充分性**。

1.17 正则损失最小化（RLM）、Tikhonov 正则化、岭回归、on-average-replace-one-stable

本节内容部分来自 [4]。首先重温一下正则化函数：前面已经讲过，正则化函数描述了**假设的复杂度**。对于正则化函数的另一个认识是**学习算法的稳定剂**，如果一个算法的输入的一个小的变化不会太多地改变输出，则称**算法是稳定的**（具体参见1.9的定义和符号说明）。下面会说明：用平方 l_2 范数作为正则化函数，可以让所有的凸利普希茨有界和凸光滑有界的学习问题都是稳定的。因此，对于这些学习问题的族，RLM 可以被用来作为一个一般学习规则。

（1）**正则损失最小化（RLM）**是一个同时最小化经验风险和一个正则化函数的学习规则。形式上，一个正则化函数是一个映射 $R:\mathbb{R}^d\to\mathbb{R}$，正则损失最小化规则输出一个假设：$\arg\min_{\omega}(E_{in}(\omega)+R(\omega))$（其实这里的 ω 应该不包含超参数吧?）。正则损失最小化共有最小描述长度算法和结构风险最小化的相似性。

假设的“复杂性”用正则化函数的值来描述（注意这里不是指假设空间的复杂性哦，而是指其中的一个假设 ω 的复杂性，这个记号我们在前面已经交代过了），而算法平衡了低经验风险与“更简单”或者“不那么复杂”的假设。

（2）我们可以用很多可能的正则化函数，它们反映了一些问题的**先验知识**（类似于在最小描述长度中的描述语言）。其中一个最常见的正则化函数为 $R(\omega)=\lambda\|\omega\|^2$，其中 $\lambda>0$ 是一个标量而且范数 $\|\omega\|$ 是 l_2 范数。这种形式的正则化函数通常叫做 **Tikhonov 正则化**。我们可以定义假设类的一个序列 $\mathcal{H}_1\subset\mathcal{H}_2\subset\mathcal{H}_3\subset\ldots$，其中 $\mathcal{H}_i=\{\omega:\|\omega\|_2\le i\}$。如果每个 $\mathcal{H}_i$ 的样本复杂度依赖于 i，那么对于这个嵌套类的序列，RLM 规则类似于 SRM 规则。

（3）把有 Tikhonov 正则化的 RLM 规则用到有平方损失的线性回归中，我们得到下面的学习规则：$\arg\min_{\omega\in\mathbb{R}^d}(\lambda\|\omega\|_2^2+\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\langle\omega,x_i\rangle-y_i)^2)$，称为**岭回归**【求解参见 [4] 的 P105，$\omega=(2\lambda mI+A)^{-1}b$】。

（4）令 $\epsilon:\mathbb{N}\to\mathbb{R}$ 是一个单调递减函数。我们说如果对于所有的分布 $\mathcal{D}$ 下式成立，一个学习算法 $\mathcal{A}$ 就是在比率 $\epsilon(N)$ 下的 **on-average-replace-one-stable**：

$$\mathbb{E}_{(S,z')\sim\mathcal{D}^{N+1},i\sim U(N)}[L(\mathcal{A}(S^{(i)}),z_i)-L(\mathcal{A}(S),z_i)]\le\epsilon(m). \tag{1.25}$$

结论：

（1）令 $\mathcal{D}$ 是一个 $\mathcal{X}\times[-1,1]$ 上的分布，其中 $\mathcal{X}=\{x\in\mathbb{R}^d:\|x\|\le 1\}$。令 $\mathcal{H}=\{\omega\in\mathbb{R}^d:\|\omega\|\le B\}$。对于任何 $\epsilon\in(0,1)$，令 $N\ge 150B^2/\epsilon^2$。那么，用参数为 $\lambda=\epsilon/(3B^2)$ 的岭回归算法满足 $\mathbb{E}_{S\sim\mathcal{D}^N}[E_{out}(\mathcal{A}(S))]\le\min_{\omega\in\mathcal{H}}E_{out}(\omega)+\epsilon$。该定理告诉了我们需要的样本个数，也便于感受下稳定的学习规则不会过拟合。

（2）**（感觉这个挺重要的）**令 $\mathcal{D}$ 是一个分布。令 $S=(z_1,\ldots,z_N)$ 是一个独立同分布的样本的序列，z' 是另一个独立同分布的样本。令 $U(N)$ 是一个在 $[N]$ 上的均匀分布。那么，对于任何学习算法，

$$\mathbb{E}_{S\sim\mathcal{D}^N}[E_{out}(\mathcal{A}(S))-E_{in}(\mathcal{A}(S))]=\mathbb{E}_{(S,z')\sim\mathcal{D}^{N+1},i\sim U(N)}[L(\mathcal{A}(S^{(i)}),z_i)-L(\mathcal{A}(S),z_i)]. \tag{1.26}$$

其中 $S^{(i)}$ 表示用 z' 代替 S 中的第 i 个样本得到的训练集【证明参见 [4] 的 P106】。当上式右边是非常小的时候，我们说 $\mathcal{A}$ 是稳定的算法——训练集中改变一个样本不会引起很大的变化。

这个定理告诉我们当且仅当一个算法是 on-average-replace-one-stable，它就不会过拟合。但是，一个不会过拟合的学习算法也不一定是一个好的学习算法，比如一个总是输出相同假设的算法。**一个好的算法应该找到一个既适合训练集（也就是一个低的经验风险）又不会过拟合的假设，RLM 规则中的参数 λ 平衡了适合训练集与稳定性**。

1.18 强凸函数、利普利茨损失、光滑和非负损失

本节内容部分来自 [4]。用有 Tikhonov 正则化 $\lambda\|\omega\|^2$ 的 RLM 规则可以得到一个稳定的算法。我们假设损失函数是凸的，而且它是利普希茨的或光滑的。我们依赖的 Tikhonov 正则化的主要性质是它能够让 RLM 的目标函数是强凸的。

（1）**强凸函数**：如果对于所有的 ω, u 和 $\alpha \in (0,1)$ 都有下列不等式成立，我们就说这个函数 f 是 λ-强凸的：$f(\alpha\omega + (1-\alpha)u) \leq \alpha f(\omega) + (1-\alpha)f(u) - \frac{\lambda}{2}\alpha(1-\alpha)\|\omega - u\|^2$【图示参见 [4] 的 P106-107，即约定了在某点处“凸”的程度，用距离表示】。显然，每个凸函数都是 0-强凸的。

结论：

（1）强凸的一些重要性质：（a）函数 $f(\omega) = \lambda\|\omega\|^2$ 是 2λ-强凸的。（b）如果 f 是 λ-强凸的而且 g 是凸的，那么 $f+g$ 是 λ-强凸的。（c）如果 f 是 λ-强凸的而且 u 是 f 的一个极小值，那么，对于任何 ω，$f(\omega) - f(u) \geq \frac{\lambda}{2}\|\omega - u\|^2$【证明参见 [4] 的 P107-108】。

（2）假设损失函数是凸的和 ρ-利普希茨的（即**利普希茨损失**）。那么，正则化项为 $\lambda\|\omega\|^2$ 的 RLM 规则是比率为 $\frac{2\rho^2}{\lambda N}$ 的 on-average-replace-one-stable。因此有：$\mathbb{E}_{S\sim\mathcal{D}^N}[E_{out}(\mathcal{A}(S)) - E_{in}(\mathcal{A}(S))] \leq \frac{2\rho^2}{\lambda N}$【推导参见 [4] 的 P108】。

（3）假设损失函数是 β-光滑和非负的。那么，正则化项为 $\lambda\|\omega\|^2$ 的 RLM 规则满足下式成立，其中 $\lambda \geq \frac{2\beta}{N}$：$\mathbb{E}[L(\mathcal{A}(S^{(i)}), z_i) - L(\mathcal{A}(S), z_i)] \leq \frac{48\beta}{\lambda N}\mathbb{E}[E_{in}(\mathcal{A}(S))]$。也意味着：$\mathbb{E}[L(\mathcal{A}(S^{(i)}), z_i) - L(\mathcal{A}(S), z_i)] \leq \frac{48\beta C}{\lambda N}$【证明参见 [4] 的 P109】。

（4）关于一般范数的强凸参见 [4] 的 P112-113。

1.19 控制“适合-稳定性”的权衡

本节内容部分来自 [4]。

对一个学习算法的期望风险进行重写，如下：

$$\mathbb{E}_S[E_{out}(\mathcal{A}(S))] = \mathbb{E}_S[E_{in}(\mathcal{A}(S))] + (\mathbb{E}_S[E_{out}(\mathcal{A}(S)) - E_{in}(\mathcal{A}(S))]. \tag{1.27}$$

上式第一项反映了 $\mathcal{A}(S)$ 适合训练数据的程度，第二项反映了 $\mathcal{A}(S)$ 的真实风险与经验风险之间的差别，即等价于 $\mathcal{A}$ 的稳定性。由于目标是最小化算法的风险，我们需要两项的和是小的。前面，我们给稳定性项加了边界，并且已经说明了，随着正则化参数 λ 的增加，稳定性项减少。另一方面，经验风险随着 λ 的增加而增加。因此我们面临 **“适合-稳定性”权衡**。

结论：

前面我们已经给出了稳定性项（也就是这里的第二项）的边界。下面我们推到 RLM 规则的经验风险项（也就是这里的第一项）的边界【推导参见 [4] 的 P110】。最后可以给出：4 个推论参见 [4] 的 P110-111（假设损失函数分别是是“凸的、ρ-利普希茨的”（神谕不等式）和“凸的、β-光滑的、非负的”时 $\mathbb{E}_S[E_{out}(\mathcal{A}(S))]$（相当于融合了两项后的结论）的边界，以及分别对应的 PAC 类似的和可学习的保证）。

1.20 模型选择、调参、用结构风险最小化进行模型选择、验证法

本节内容部分参考自 [4]。

（1）当面对实际问题时，我们通常可以想出几种可能取得好结果的算法，每一种算法可能有集合参数。我们如何为解决发生在身边的问题选择一种最佳算法？如何设置算法参数？这就是通称的**模型选择**问题。

在**调参**方面，但是仅用经验风险来进行模型选择是不够的。例子如 [4] 的 P85 所示，在获得训练样本集后，我们用多项式来对其进行拟合，不难看出，经验风险随着多项式次数增加而减少，但直观上告诉我们，设置多项式次数为 3 要比次数为 10 更好。

如下所示，有两种常用的模型选择方法，一个是建立在结构风险最小化原则之上，另一个是建立在验证的概念之上。在模型选择中，我们尽力寻找逼近误差和估计误差的平衡点。

（2）**用结构风险最小化进行模型选择**旨在调整偏差和复杂度的权衡，它在学习算法依赖于某一个参数控制偏差-复杂度权衡考虑时非常有用。

取一个可计算的假设类序列 $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, ...$（例如：多项式回归问题中，用 $\mathcal{H}_d$ 表示次数至多为 d 的多项式构成的集合）。假定对于任意 d，类 $\mathcal{H}_d$ 满足一致收敛属性，样本复杂度函数具有以下形式 $N^{UC}_{\mathcal{H}_d}(\epsilon, \delta) \leq \frac{g(d)\log(1/\delta)}{\epsilon^2}$，其中 $g : \mathbb{N} \to \mathbb{R}$ 是单调递增函数。在这个例子中，对于 $d \in \mathbb{N}$ 和 $h \in \mathcal{H}_d$，最小误差界为下式成立的概率不低于 $1-\delta$：$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{g(d)(\log(1/\delta) + 2\log(d) + \log(\pi^2/6))}{N}}$。它揭示了：对于任意 d 和 $h \in \mathcal{H}_d$，真实风险界取决于以下两项：经验风险 $E_{in}(h)$ 和依赖于 d 的复杂度表达形式。结构风险最小化规则搜索 d 和 $h \in \mathcal{H}_d$ 来最小化上式。

结构风险最小化在多数情形下都非常有用，但是在很多实际情况下上述方程给出的上界过于悲观，下面将介绍更实用的方法。

（3）对于上述结构风险最小化，尽管这些界是松弛的、悲观的，但是它可以反映所有假设和所有可能的数据分布。通过使用一部分训练数据作为验证集，我们能够得到真实风险的更精确估计，在验证集上可以估计算法输出预测器的有效性，这个过程就称为**验证法**。

包括：（a）**留出的样本集**：将样本集拆分为两个部分，一部分用于训练，另一部分用于验证。（b）**模型选择的验证法**：在验证集上应用经验风险最小化（只要 $\mathcal{H}$ 不太大，验证集的错误就会近似真实误差，但如果尝试更多的方法，结果是 $|\mathcal{H}|$ 与验证集的样本数强相关，就会有过拟合的风险）。（c）**模型选择曲线**：显示训练误差和验证误差作为一个模型考虑的复杂度函数。在参数更多、取值更大的情况下，可以采用更好的网格搜索。但是没有采用一个基于正确结果曲线的网格搜索，最终会得到一个较差的模型。（d）目前为止的上述验证程序假定数据是足够大的，但是在一些应用中数据很少，我们不想将数据浪费在验证集上。**k 折交叉验证**：将原训练样本集拆分为样本数量为 N/k 的 k 折样本子集。对于每一折样本，这个算法是在其他折样本的联合样本上训练，然后由这一折的样本上估计输出的误差。最终，所有误差的平均即为真实误差的估计。特殊情形 $k = N$，称为留一验证法（LOO）。一旦选择了最好的参数，这个算法被限制使用这组最优的参数在整个训练集上。一些研究显示交叉验证对稳定性算法非常有效【伪代码参见 [4] 的 P89 】。

在大多数实际应用中，我们将可利用的样本拆分成三个集合，即进行**训练-验证-测试拆分**。

【数学角度理解，即对 $E_{out}(h)$ 和 $E_v(h)$ 的误差界来分析这些验证法的有效性，参见 [4] 的 P86-89，其中 v 表示验证集】

1.21 学习失败时的做法

本节内容部分来自 [4]。

当接到一个学习任务，需要选择一个假设类、一个学习算法和参数来想办法解决它。使用一个验证集来优化参数并在测试集上测试学习预测器。不幸的是，测试结果并不令人满意。那么，**问题出在哪儿？接下来应该怎么做呢？**

（1）常见的做法包括：（a）增大样本集；（b）改变假设类，包括扩大假设类、缩减假设类、彻底改变它、改变参数；（c）改变数据的特征表示；（d）应用学习规则改变优化算法。

更具体地说，真实误差可以分解为逼近误差和估计误差。（a）对于逼近误差。前面已经讲过，它不依赖于样本数量或所使用的算法，而只依赖于假设类（即先验知识）和潜在分布之间的吻合。所以，**如果逼近误差太大**，它将不会帮助我们扩大训练样本数量，而且对于降低假设类没有意义。在这种情况下，扩大假设类或将其彻底改变是有用的（如果我们通过不同的假设类形式有一些可选的先验知识）。我们能考虑应用同样的假设类，但是应用数据的不同的特征表示【具体做法参见 [4] 的 25 章】。（b）对于估计误差。它强烈依赖于样本数量。因此，**如果估计误差太大**，我们可以努力获取更多的训练样本。我们也可以考虑减少假设类。但是，在这种情况下，它对于扩大假设类没有什么意义。

（2）由上可见，确定问题产生的原因有助于我们选取合适的做法。为了确定问题出在哪儿，我们一般采用如下的 5 个步骤：（a）如果学习包括参数优化，画出模型选择曲线来确认你已经近似优化参数（上面已经讲过）；（b）如果扩大假设类，训练误差特别大，那就彻底改变它，或者改变数据的特征表示方法；（c）如果训练误差很小，画出学习曲线，尽力推断误差是来源于估计误差还是近似误差；（d）如果近似误差看起来足够小，尝试获得更多的数据。如果这不太可能，我们则考虑减少假设类的复杂度；（e）如果近似误差很大，尝试改变假设类或者彻底改变特征表示方法【分析参见 [4] 的 P90-91，使用验证分解误差：

$E_{out}(h) = (E_{out}(h) - E_v(h)) + (E_v(h) - E_{in}(h)) + E_{in}(h)$，这样一来将可以从训练集和验证集估计得到，后两项提供了有用的信息，其中第一项可以用本章的定理来建立一个很紧的界，第二项反映了过拟合或欠拟合，注意区分逼近误差 ϵ_{app} 和训练误差 $E_{in}(h)$。在上述分解中，当 $E_{in}(h)$ 较大或较小时，可以用学习曲线（注意画法，它和模型选择曲线是不一样的哦）来区分这两中场景】。

1.22 核方法（待补充）

1.23 自信息、熵、联合熵、条件熵、相对熵 (KL 散度)/JS 散度、交叉熵、互信息、群体稳定性指标 (PSI)

本节内容一部分参考自博客：https://www.cnblogs.com/kyrieng/p/8694705.html。

（1）一条信息的信息量大小和它的不确定性有直接的关系。我们需要搞清楚一件非常非常不确定的事，或者是我们一无所知的事，就需要了解大量的信息。相反，如果我们对某件事已经有了较多的了解，我们就不需要太多的信息就能把它搞清楚。所以，从这个角度，我们可以认为，信息量的度量就等于不确定性的多少。考虑一个随机变量 X，由上可知，信息的量度应该依赖于概率分布 $p(x)$（不确定性的多少）。用 $I(x) = -\log p(x)$ 描述随机变量 X 的某个事件 x 发生所带来的信息量，称之为**自信息**（至于为什么是这种形式，在高统笔记中已经交代很清楚了，即需要满足“单调性”、“累加性”和“非负性”，这三条性质是从现实意义下得到的）。其平均信息量可以通过求 $I(x)$ 关于概率分布 $p(x)$ 的期望得到，即 $H(X) = -\sum_x p(x)\log p(x)$，称之为**熵**，它是表示随机变量 X 不确定的度量，是对所有可能发生的事件产生的信息量的期望。将一维随机变量分布推广到多维随机变量分布，则其**联合熵**为 $H(X,Y,...) = -\sum_{x,y,...} p(x,y,...)\log p(x,y,...)$。

（2）**条件熵** $H(Y|X)$ 表示在已知随机变量 X 的条件下随机变量 Y 的不确定性，它定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望，即 $H(Y|X) = -\sum_x p(x)H(Y|X=x) = -\sum_{x,y} p(x,y)\log p(y|x)$。与对数似然最大化的联系：监督学习的目的的是当 x 可以通过 f 的最佳选择获得时，最小化随机变量 Y 的不确定性，从而得到 $f(X)$ 的预测。最优值与信息缺乏相对应，即 $H(Y|f(X)) = 0$，也就是说理想情况下智能体总是能预测 Y。

（注意：$P(A|B) = \dfrac{P(AB)}{P(B)}$）

（3）设 $p(x)$、$q(x)$ 是随机变量 X 中取值的两个概率分布，则 p 对 q 的**相对熵（或 KL 散度）** $D_{KL}(p\|q) = \sum_x p(x)\log\dfrac{p(x)}{q(x)} = E_{p(x)}\log\dfrac{p(x)}{q(x)}$。相对熵可以用来衡量两个概率分布之间的差异，上面公式的意义就是求 p 与 q 之间的对数差在 p 上的期望值（注意和条件熵的记号含义区分开）。

但是 KL 散度存在不对称的问题，为了解决该问题，于是定义 **JS 散度：** $D_{JS}(P\|Q) = \dfrac{1}{2}D_{KL}(P\|\dfrac{P+Q}{2}) + \dfrac{1}{2}D_{KL}(Q\|\dfrac{P+Q}{2})$。

（4）现在有关于样本集的两个概率分布 $p(x)$ 和 $q(x)$，其中 $p(x)$ 为真实分布，$q(x)$ 为非真实分布。如果使用非真实分布 $q(x)$ 来表示来自真实分布 $p(x)$ 的平均编码长度，则是：$H(p,q) = \sum_x p(x)\log\dfrac{1}{q(x)} = -\sum_x p(x)\log q(x)$，称之为**交叉熵**。交叉熵可以来衡量在给定的真实分布下，使用非真实分布所指定的策略消除系统的不确定性所需要付出的努力的大小。

交叉熵常所为机器学习中的分类的损失函数，用于衡量模型预测分布和实际数据分布之间的差异性。

（5）对于两个随机变量 X, Y，如果其联合分布为 $p(x,y)$，边缘分布为 $p(x), p(y)$，则**互信息**可以定义为：$I(X;Y) = \sum_{x,y} p(x,y)\log\dfrac{p(x,y)}{p(x)p(y)}$。

互信息用于衡量两个变量之间的关联程度，衡量了知道这两个变量其中一个，对另一个不确定度减少的程度。

（6）**群体稳定性指标 PSI**：$PSI = \dfrac{D_{KL}(p\|q) + D_{KL}(q\|p)}{2}$。PSI 可以看做是解决 KL 散度非对称性的一个对称性度量指标，用于度量分布之间的差异（常用于风控领域的评估模型预测的稳定性）。PSI 与 JS 散度的形式是非常类似的，其含义等同 P 与 Q，Q 与 P 之间的 KL 散度之和。

结论：

（1）随机变量的取值个数越多，状态数也就越多，信息熵就越大，混乱程度就越大。当随机分布为均匀分布时，熵最大。且 $0 \leq H(X) \leq \log n$【证明参见博客】。

（2）熵是传输一个随机变量状态值所需的比特位下界（最短平均编码长度），使用更短的编码来描述更可能的事件，使用更长的编码来描述不太可能的事件。因此，信息熵可以应用在数据压缩方面。【证明参见博客】

（3）$H(Y|X) = H(X,Y) - H(X)$。可以这样理解：描述 X 和 Y 所需的信息是描述 X 自己所需的信息，加上给定 X 的条件下具体化 Y 所需的额外信息【证明参见博客】。

（4）相对熵有以下重要结论：（a）如果 $p(x)$ 和 $q(x)$ 两个分布相同，则相对熵等于 0；（b）$D_{KL}(p\|q) \neq D_{KL}(q\|p)$，即相对熵具有不对称性；（c）$D_{KL}(p\|q) \geq 0$【证明参见博客】。

（5）相对熵与交叉熵之间的关系：$D_{KL}(p\|q) = H(p,q) - H(p)$。当 $H(p)$ 为常量时（注：在机器学习中，训练数据分布是固定的），最小化相对熵 $D_{KL}(p||q)$ 等价于最小化交叉熵 $H(p,q)$ 也等价于最大化似然估计。交叉熵可以用来计算学习模型分布与训练分布之间的差异（希望学到的模型的分布和真实分布一致，但是真实分布不可知，所以假设训练数据是从真实数据中独立同分布采样的，因此，我们希望学到的模型分布至少和训练数据的分布一致。在训练数据上 $H(p)$ 为常量，因此可以用交叉熵）。交叉熵广泛用于逻辑回归的 Sigmoid 和 Softmax 函数中作为损失函数使用【证明参见博客】。

（6）互信息有以下重要结论：（a）互信息具有“非负性”和“对称性”；（b）$I(X;Y) = H(X) - H(X|Y)$；（c）$H(X,Y) = H(X|Y) + H(Y|X) + I(X;Y)$；（d）$I(X;Y) = E_Y[D_{KL}(p(x|y)\|p(x))]$，即是 $p(x|y)$ 对 $p(x)$ 的 KL 散度里，对随机变量 Y 的期望【证明参见百度或知乎】。

理解：

（1）简而言之，信息熵是完美编码，交叉熵是不完美编码，相对熵是两者的差值，即交叉熵减去信息熵。从信息编码的理解它们可参见知乎：https://www.zhihu.com/question/41252833。

（2）对互信息的理解：原来我对 X 有些不确定（不确定性为 $H(X)$），告诉我 Y 后我对 X 不确定性变为 $H(X|Y)$，这个不确定性的减少量就是 X,Y 之间的互信息 $I(X;Y) = H(X) - H(X|Y)$。更直观的意义可以理解为，当你完整的学到 Y 的所有知识的时候，你对 X 的知识的增长量就是 $I(X;Y)$。摘自知乎：https://www.zhihu.com/question/24059517/answer/26750918。还可以这么理解：互信息指的是两个随机变量之间的关联程度，即给定一个随机变量后，另一个随机变量不确定性的削弱程度。

1.24 相似性度量技术

以下公式中，如无特殊说明，则默认为行向量和行矩阵。A 为 $m \times s$ 矩阵，B 为 $n \times s$ 矩阵。此外，对于以下没有给出的指标计算公式，可参考知乎链接：https://www.zhihu.com/question/272195313/answer/2260755496。

1.24.1 闵氏距离（Minkowski Distance）类

包括：闵氏距离（Minkowski Distance）、曼哈顿距离（Manhattan Distance）、欧几里得距离（Euclidean Distance）、切比雪夫距离（Chebyshev Distance）。

（1）闵氏距离： $(\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$。

（2）曼哈顿距离： 闵氏距离中 $p = 1$ 时。

（3）欧几里得距离： 闵式距离中 $p = 2$ 时。**矩阵形式**为：$S = \sqrt{|-2A \cdot B^T + \mathcal{A} + \mathcal{B}|}$，其中

$$\mathcal{A}^{m\times n} = (\alpha^T, \alpha^T, ..., \alpha^T), \alpha = (\sqrt{\sum_{j=1}^{s} A_{1j}^2}, \sqrt{\sum_{j=1}^{s} A_{2j}^2}, ..., \sqrt{\sum_{j=1}^{s} A_{mj}^2}),$$

$$\mathcal{B}^{m\times n} = (\beta, \beta, ..., \beta)^T, \beta = (\sqrt{\sum_{j=1}^{s} B_{1j}^2}, \sqrt{\sum_{j=1}^{s} B_{2j}^2}, ..., \sqrt{\sum_{j=1}^{s} B_{nj}^2})。$$

（4）切比雪夫距离： 闵式距离中 $p \to \infty$ 时，即为 $\max|x_i - y_i|$。它起源于国际象棋中国王的走法。

以上距离指标的一个有点就是它们可以用于构造度量空间等。包括后面的相似性度量技术的选用，一定要根据自己的模型需要构造什么样的空间来合理选择。

但是，(1) 也会面临“维度灾难”：距离度量随着空间的维度的不断增加，计算量复杂度也逐增。另外，在高维空间下，维度越高，任一样本之间的距离越趋于相等（即样本的最大与最小闵式距离之间的相对差距就趋于 0）。对于维度灾难，常用的有 PCA 方法进行降维计算。(2) 此外，闵式距离可能会引起量纲差异问题，基本的解决方法就是对数据进行“标准化（如 z-score normalization）”或“归一化 (Max-Min normalization)”。

1.24.2 相似度（Similarity)

包括：余弦相似度 (Cosine Similarity)、协方差、皮尔逊相关系数 (Pearson Correlation)、卡方检验。

(1) 余弦相似度： $\cos(\theta) = \dfrac{x \cdot y}{|x| \cdot |y|} = \dfrac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$。**矩阵形式**为：$S = \dfrac{A \cdot B^T}{\alpha^T \cdot \beta}$，其中 $\alpha = (\sqrt{\sum_{j=1}^{s} A_{1j}^2}, \sqrt{\sum_{j=1}^{s} A_{2j}^2}, ..., \sqrt{\sum_{j=1}^{s} A_{mj}^2}), \beta = (\sqrt{\sum_{j=1}^{s} B_{1j}^2}, \sqrt{\sum_{j=1}^{s} B_{2j}^2}, ..., \sqrt{\sum_{j=1}^{s} B_{nj}^2})$。

余弦相似度的取值为 $[-1, 1]$, 1 表示两者完全正相关，-1 表示两者完全负相关，0 表示两者之间独立。余弦相似度与向量的长度无关，只与向量的方向有关。

但余弦相似度会受向量平移的影响。

(2) 协方差： $Cov(x, y) = E[(x - \overline{x})(y - \overline{y})] = E[xy] - E[x]E[y]$。

如果两个变量之间的协方差为正值，则这两个变量之间存在正相关，反之亦然。**矩阵形式**为：$\dfrac{1}{s}(A - \overline{A}) \cdot (B - \overline{B})^T$，其中 $\overline{A} = (\alpha^T, \alpha^T, ..., \alpha^T), \alpha = (\dfrac{1}{s}\sum_{j=1}^{s} A_{1j}, \dfrac{1}{s}\sum_{j=1}^{s} A_{2j}, ..., \dfrac{1}{s}\sum_{j=1}^{s} A_{mj})$; $\overline{B} = (\beta^T, \beta^T, ..., \beta^T), \beta = (\dfrac{1}{s}\sum_{j=1}^{s} B_{1j}, \dfrac{1}{s}\sum_{j=1}^{s} B_{2j}, ..., \dfrac{1}{s}\sum_{j=1}^{s} B_{nj})$。

(3) 皮尔逊相关系数： $\dfrac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$。**矩阵形式**为：$\dfrac{(A - \overline{A})(B - \overline{B})}{\alpha^T \beta}$，其中 $\alpha = \Big(\sqrt{\sum_{j=1}^{s}(A_{1j} - \overline{A}_{1*})^2}, \sqrt{\sum_{j=1}^{s}(A_{2j} - \overline{A}_{2*})^2}, ..., \sqrt{\sum_{j=1}^{s}(A_{mj} - \overline{A}_{m*})^2}\Big)$,

$\beta = \Big(\sqrt{\sum_{j=1}^{s}(B_{1j} - \overline{B}_{1*})^2}, \sqrt{\sum_{j=1}^{s}(B_{2j} - \overline{B}_{2*})^2}, ..., \sqrt{\sum_{j=1}^{s}(B_{nj} - \overline{B}_{n*})^2}\Big)$。

皮尔逊相关系数的取值为 $[-1, 1]$。是在余弦相似度或协方差基础上做了优化（变量的协方差除以标准差）。它消除乐每个分量标准不同（分数膨胀）的影响，具有平移不变性和尺度不变性。

(4) 卡方检验： $\chi^2 = \sum \dfrac{(A - E)^2}{E} = \sum_{i=1}^{s} \dfrac{(A_i - E_i)^2}{E_i} = \sum_{i=1}^{s} \dfrac{(A_i - np_i)^2}{np_i}$。其中 A 代表实际频数，E 代表期望频数。

卡方检验主要是比较两个变量之间的关联性、独立性分析。

1.24.3 字符串距离 (Distance of Strings)

包括：Levenshtein 距离、汉明距离、带权重的字符串距离。

(1) Levenshtein 距离： 编辑距离的一种，指两个字串之间，由一个转成另一个所需要的最少编辑操作次数。允许的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。

(2) 汉明距离： 为两个等长字符串对应的不同字符的个数，也就是将一个字符串变换成另一个字符串所偶需要替换的字符个数。

(3) 带权重的字符串距离： 考虑字符（特征）权重的相似度方法，包括 TF-IDF、BM25、WMD 算法等。

因为就字符串距离来说，不同字符所占的分量是不一样的，比如“我乐了”与“我怒了”、“我乐了啊”的 Levenshtein 距离都是 1，但其实两者差异还是很大的，因为像“啊”这种语气词的重要性明显不如“乐”。

1.24.4 集合距离

包括：Jaccard 系数、Dice 系数、Tversky 系数。

（1）Jaccard 系数： $s = \dfrac{|X \cap Y|}{|X \cup Y|} = \dfrac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$。

取值范围为 $[0, 1]$，0 表示两个集合没有重合，1 表示两个集合完全重合。

（2）Dice 系数： $s = \dfrac{2|X \cap Y|}{|X| + |Y|}$。

取值范围为 $[0, 1]$，与 Jaccard 系数可以相互转换：$s_d = 2s_j/(1 + s_j)$。

但 Dice 系数不满足距离函数的三角不等式，不是一个合适的距离度量。

（3）Tversky 系数： $s = \dfrac{|X \cap Y|}{|X \cap Y| + \alpha|X \backslash Y| + \beta|Y \backslash X|}$。其中 $X \backslash Y$ 表示集合的相对补集。

Tversky 系数可以理解为 Jaccard 系数和 Dice 系数的一般化，当 α, β 为 1 时为 Jaccard 系数，当 α, β 为 0.5 时为 Diec 系数。

1.24.5 信息论距离

参见1.23。常用于损失函数的设置。要打开思路哦！

1.24.6 时间系列、图结构的距离

包括：DTW (Dynamic Time Warping) 距离、图结构的距离。

（1）DTW (Dynamic Time Warping) 距离： 用于衡量两个序列之间的相似性，适用于不同长度、不同节奏的时间序列。DTW 采用了动态规划的方法来进行时间规整的计算，通过自动 warping 扭曲时间序列（即在时间轴上进行局部的缩放），使得两个序列的形态尽可能的一致，得到最大可能的相似度。【具体细节需用到时再去了解】

（2）图结构的距离： 图结构间的相似度计算，有图同构、最大共同子图、图编辑距离、Graph Kernel、图嵌入计算距离等方法。【具体细节需要用到时再去了解】

1.24.7 度量学习 (Metric Learning)

度量学习的对象通常是样本特征向量的距离，度量学习的关键在于如何有效的度量样本间的距离，目的是通过训练和学习，减小或限制同类样本之间的距离，同时增大不同类别样本之间的距离。其分类可以参见知乎链接：https://www.zhihu.com/question/272195313/answer/2260755496。

理解：

（1）**维度灾难问题**：（a）距离度量随着空间的维度不断增大，计算复杂度也逐增。（b）在维度越高的情况下，任意样本之间的距离越趋于相等。对于维度灾难问题，可以用 PCA 方法进行降维。

（2）**量纲差异问题**：通常用到的就是1.25节的归一化和标准化技术。另外还可以使用**马氏距离（协方差距离）**，其考虑到各种特性之间的联系是（量纲）尺度无关的，可以排除变量之间的相关性的干扰，缺点是夸大了变化微小的变量的作用。

1.25 归一化和标准化技术

首先，需要对“归一化（Standarlization）”和“标准化 (Normalization)”的概念进行**区分**：前者表示将一列数据变化到某个固定区间（范围）中，通常，这个区间是 $[0, 1]$，也可以是 $[-1, 1]$，也可以是任何一个固定的范围。后者将一列数据变化为均值为 0，标准差为 1 的分布。切记，并非一定是正态的。对于它俩的**共同点**：本质上都是对数据的线性变换，都是对数据进行的一种缩放操作，即一种无量纲处理操作。

（1）常用的**归一化技术**，包括：**线性归一化**、**标准差归一化**、**非线性归一化**（用在数据分析比较大的场景，一般使用的函数包括 log、指数、正切等，需要根据数据分布的具体情况来决定非线性函数的曲线）。【计算公式可参考见知乎链接：https://zhuanlan.zhihu.com/p/91125751 和 https://zhuanlan.zhihu.com/p/424518359】。对于神经网络中归一化的方式策略，可参考知乎链接：https://zhuanlan.zhihu.com/p/378598488。

（a）**线性归一化：** $x' = \dfrac{x - \min(x)}{\max(x) - \min(x)}$。取值为 $[0, 1]$。

（b）**标准差归一化：**也叫 Z-score 标准化，$x^* = \frac{x-\mu}{\sigma}$。使得处理后的数据符合标准正态分布，即均值为 0，标准差为 1。

（c）**非线性归一化：**这种方法一般使用在数据分析比较大的场景，有些数值很大，有些很小，通过一些数学函数，将原始值进行映射。一般使用的函数包括 log、指数、正切等，需要根据数据分布的具体情况来决定非线性函数的曲线。

（2）常用的**标准化技术**，包括：Z-score 标准化、离差标准化（即 min-max 标准化）、MaxAbsScaler([-1,1])、RobustScaler、log 函数转换、atan 函数转换 ([-1,1])。

【计算公式可参考知乎链接：https://zhuanlan.zhihu.com/p/260349978、

理解：

(1) 推荐用“归一化”的情形：(a) 对处理后的数据范围有严格要求时。(b) 在不涉及距离度量、协方差计算的时候，可以使用归一化。

(2) 推荐用“标准化”的情形：(a) 在机器学习中常用标准化，尤其是在无从下手时，就先尝试标准化。(b) 数据不稳定，存在极端的最大最小值时，不要归一化。(c) 在分类、聚类算法中，需要使用距离来度量相似性的时候，或者使用 PCA 技术进行降维的时候，标准化表现更好。

2 常见激活函数

在理论层面，激活函数的“有界/无界，连续/非连续，恒定/非恒定”等性质会得到不同的定理，达到不同的逼近效果等，相关内容可以参照我的汇报 PPT 的 GraphSAGE case。因此有必要对常见的激活函数做归纳总结。以下内容部分参考自：https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/和 [5]。

2.1 Linear Functions

（1）Linear function:$f(x) = kx, k > 0$。

（2）Identity function：$f(x) = x$。

2.2 Step Functions

（1）Threshold step function：$H(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases}$。导数为 $\delta(x)$，注意 $\delta(x)$ 的积分公式。

（2）Bipolar Step Function：$S(x) = \begin{cases} -1, & x < 0, \\ 1, & x \geq 0. \end{cases}$。导数为 $2\delta(x)$。

【关于这些函数的图像、性质和求导，参见 [5] 的 P21-23】

2.3 Hockey-stick functions

（1）Rectified Linear Unit (ReLU)：$ReLU(x) = xH(x) = \max\{x, 0\} = \begin{cases} 0, & x < 0, \\ x, & x \geq 0. \end{cases}$。导数为 $H(x)$。

（2）Parametric Rectified Linear Unit (PReLU)：$PReLU(\alpha, x) = \begin{cases} \alpha x, & x < 0, \\ x, & x \geq 0. \end{cases}, \alpha > 0$。与 Leaky ReLU 的区别在于，Leaky ReLU 的这个参数是提前设定的，而 PReLU 的这个参数是根据数据而定的。

（3）Exponential Linear Units (ELU)：$ELU(\alpha, x) = \begin{cases} x, & x > 0, \\ \alpha(e^x - 1), & x \leq 0. \end{cases}$

（4）Scaled Exponential Linear Units (SELU)：$SELU(\alpha, \lambda, x) = \lambda \begin{cases} x, & x > 0, \\ \alpha(e^x - 1), & x \leq 0. \end{cases}$。

（5）Sigmoid Linear Units (SLU)：$\phi(x) = x\sigma(x) = \frac{x}{1+e^{-x}}$ 或 $\phi_c(x) = x\sigma(cx) = \frac{x}{1+e^{-cx}}$。

（6）Softplus：$sp(x) = \ln(1 + e^x)$。

【关于这些函数的图像、性质和求导，参见 [5] 的 P23-26】

2.4 Sigmoid functions

（1）Logistic function with parameter $c > 0$：$\sigma_c(x) = \sigma(c, x) = \dfrac{1}{1+e^{-cx}}$。当 $c = 1$ 时，则为 the standard logistic function。

（2）Hyperbolic tangent：$t(x) = \tanh x = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$。

（3）Arctangent function：$h(x) = \dfrac{2}{\pi}\tanh^{-1}(x), x \in \mathbb{R}$。

（4）Softsign function：$so(x) = \dfrac{x}{1+|x|}, x \in \mathbb{R}$。

（5）Piecewise Linear：$f_\alpha(x) = f(\alpha, x) = \begin{cases} -1, & x \le -\alpha, \\ x/\alpha, & -\alpha < x < \alpha, \\ 1, & x \ge \alpha. \end{cases}$。

【关于这些函数的图像、性质和求导，参见 [5] 的 P26-30】

2.5 Bumped-type functions

（1）Gaussian：$g(x) = e^{-x^2}, x \in \mathbb{R}$。

（2）Double exponential：$f(x) = e^{-\lambda|x|}, x \in \mathbb{R}, \lambda > 0$。

【关于这些函数的图像、性质和求导，参见 [5] 的 P30-31】

2.6 Classification functions

（1）softmax function：$softmax_c(x) = (\dfrac{e^{cx_1}}{\sum_{i=1}^n e^{cx_i}}, \dfrac{e^{cx_2}}{\sum_{i=1}^n e^{cx_i}}, ..., \dfrac{e^{cx_n}}{\sum_{i=1}^n e^{cx_i}}), c > 0$。当 $c = 1$ 时，即得到 the usual softmax function。当 $c \to \infty$ 时，值为 $e_k = \max\{x_1, x_2, ..., x_n\}$。

【关于这个函数的图像、性质和求导，参见 [5] 的 P31-32】

2.7 激活函数建模

[5] 中列举了两类可以指导建模的 generic classes of activation functions: sigmoidal 【[5] 的 P32-36】and squashing functions【[5] 的 P36-37】。包括：定义（告诉我们要造什么）、引理、命题、定理和推论及例子。

3 常见损失函数

以下内容的组织框架大部分总结自文献 [1]。根据模型目的和所要解决的问题，[1] 将机器学习的任务分为了“有监督”（有标签数据）、“半监督”（有标签和无标签数据）、“无监督”（无标签数据）和“强化学习”四类，并依次讨论了在这四类任务下损失函数及其特点。

[5] 中详细总结归纳了常用的损失函数，可供参考学习。

3.1 监督学习任务

监督学习任务包括“分类任务”和“回归任务”（注意，常见的“聚类任务”属于无监督任务）。在回归中，人们通常对保持 $y - h(x)$ 接近于零感兴趣。在分类中，将这种损失最小化对于完美地获得分类技巧而言并不是必要的，这与目标的离散性有关，真正重要的是 $yh(x)$。

监督学习任务的风险函数一般形式为：

$$\arg\min_\theta \frac{1}{N}\sum_{i=1}^N L(y_i, h(x_i, \theta)) + \lambda\|\theta\|_2^2, \tag{3.1}$$

其中，N 是训练样本的总数，i 代表其中的第 i 个样本，x^i 和 y^i 分别是第 i 个样本的特征向量和标签。h 是需要学习出的最优映射函数，L 是损失函数。λ 是折衷参数，能够通过交叉验证进行确定。

3.1.1 分类任务

对于分类任务（若无特别说明，假设目标满足 $\mathcal{Y}=\{-1,+1\}$），常见的损失函数有：

（1）自然损失（0-1 损失）： $L(x,y,h)=[y\neq h(x)]$。其中，$[\cdot]$ 表示计数，即满足括号中条件的个数。

（2）铰链损失（凸函数）： $L(x,y,h)=(\theta-yh(x))_+$。其中，$(\cdot)_+$ 是 $(a)_+=[a>0]\cdot a$ 的分量方式的扩展，显然等价于 $\max(a,0)$。相对（1）中的自然损失，铰链损失进一步反映出相同的分类质量，$\theta>0$ 引入了损失所需要的鲁棒性。

铰链损失在标量情况下的另一种平滑估计为 $L(x,y,h)=\log(1+\exp(\theta-yh(x)))$。

（3）相对熵（KL 散度）损失： $-\dfrac{1+y}{2}\log\dfrac{1+h(x)}{2}-\dfrac{1-y}{2}\log\dfrac{1-h(x)}{2}$，其中我们假设 $-1\leq h(x)\leq 1$。用于衡量两个概率分布之间的差异。

（4）交叉熵损失（logistic 回归下是凸函数，MLP 下是非凸函数）： 考虑 $\mathcal{Y}=\{0,1\}$ 情况有 $L(x,y,h)=-y\log h(x)-(1-y)\log(1-h(x))$。如果使用 softmax 输出，由于固有的概率约束，我们可以简单地使用 $L(x,y,h)=-y\log h(x)$，即为交叉熵损失，适用于多分类问题。最小化相对熵可以等价于最小化交叉熵。机器学习中常用交叉熵损失。

3.1.2 回归任务

对于回归任务，常见的损失函数有：

（1）二次损失（凸函数）： $L(x,y,h)=\dfrac{1}{2}(y-h(x))^2$。它是 $L(x,y,h)=\dfrac{1}{p}\|y-h(x)\|_p^p$ 在 $p=2$ 下的一个特例，其中 $\|u\|_p=(\sum_{i=1}^{d}u_i^p)^{\frac{1}{p}}$。$p=2$ 并不是唯一值得关注的值，$p=1$ 和 $p=\infty$ 也产生显著的性质。只要 $|y-h(x)|<1$，当 p 的高值就会返回几乎为零的损失。在极端情况下，$p=\infty$ 时损失返回 $\max_j|y_j-h(x_j)|$。

该函数的数学特性很好，这使得计算梯度变得更容易。

（2）浴盆损失： $L(x,y,h)=(|y-h(x)|-\epsilon)_+$。

3.2 半监督学习任务

3.3 无监督学习任务

3.4 强化学习任务

4 常见优化算法

参考文献

[1] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.

[2] 周志华. 机器学习. 清华大学出版社, 2016.

[3] 谢宇等（译）. 机器学习：基于约束的方法. 机械工业出版社, 2020.

[4] Shai Shalev-Shwartz 等（著） 张文生等（译）. 深入理解机器学习：从原理到算法. 机械工业出版社, 2016.

[5] Ovidiu Calin. *Deep Learning Architectures*. Springer, 2020.

附录 A 英文术语

假设：hypothesis

泛化能力：generalization performance
折衷参数：compromise parameter
不收敛：non-convergence
少样本学习：few-shot learning
偏差：deviation
极值：extremum
概率近似正确（PAC）学习：probably approximately correct learning
经验损失：empirical loss
经验风险最小化；empirical risk minimization
结构风险最小化：structured risk minimization (SRM)
最小描述长度：minimum description length (MDL)
正则损失最小化：regularized loss minimization (RLM)
松弛变量：slack variables
惩罚系数：penalty coefficient
折扣因子：discount factor
增长函数：growth function
对分：dichotomy
打散：shattering
VC 维：VC dimension
概率近似正确：probably approximately correct (PAC)
独立同分布：independent and identically distributed
假设空间：hypothesis space
可分的/不可分的：separable/non-separable
分布无关的：distribution-free
数据独立的：data-independent
均匀稳定性：uniform stability
一致连续性/一致连续的：uniform continuity/uniformly continuous (UC)
不一致可学习：nonuniform learning (NUL)
自信息：self-information
联合熵：joint entropy
相对熵/KL 散度：relative entropy/Kullback–Leibler divergence
分类任务：classification task
回归任务：regression task
铰链损失：hinge loss
相对熵损失：relative entropy loss
交叉熵损失：cross entropy loss
相对熵：conditional entropy
二次损失：quadratic loss
浴盆损失：bathtub loss
对数似然：log-likelihood
上镜图：epigraph