第二章 机器学习理论

什么是“学习”？学习就是人类通过观察、积累经验，掌握某项技能或能力。就好像我们从小学习识别字母、认识汉字，就是学习的过程。而机器学习（Machine Learning），顾名思义，就是让机器（计算机）也能向人类一样，通过观察大量的数据和训练，发现事物规律，获得某种分析问题、解决问题的能力。机器学习可以被定义为：Improving some performance measure with experience computed from data. 也就是机器从数据中总结经验，从数据中找出某种规律或者模型，并用它来解决实际问题。

顾名思义，机器学习理论研究的是机器学习的理论基础，其目的是分析学习任务的困难本质，为学习算法提供理论保证，并根据分析结果指导算法设计。接下来我们将学习以下四个基础理论：1.机器可学习性分析（PAC 学习模型）；2.模型的偏差与方差及其泛化性能；3.Hoeffding 不等式与模式二分性；4.VC 维理论与模型复杂性。

2.1 机器可学习性分析

2.1.1 PAC 学习框架

了解机器学习首先要理解机器为什么可以学习，什么情况下可以进行学习。下面我们举例说明。

如图所示，输入为x，是一个三维数据，且元素都为布尔值，如果以D来做训练数据，那么要预测未知的情况，那请问当x为101,110,111的时候，预测输出y是什么呢？我们看到图表中，会有8中不同的假设（hypothesis），所以我们无论预测是哪种输出，都有可能让我们的预测是完全错误的。这是不是就说明这种条件下，学习器是不可学习的呢？现在我们就从这个角度出发，看看如何训练我们的学习器，才能让学习器真正学到有用的知识，进而来产生有效的预测。

x	y	g	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0 0 0	o	o	o	o	o	o	o	o	o	o
0 0 1	×	×	×	×	×	×	×	×	×	×
0 1 0	×	×	×	×	×	×	×	×	×	×
0 1 1	o	o	o	o	o	o	o	o	o	o
1 0 0	×	×	×	×	×	×	×	×	×	×
1 0 1		?	o	o	o	o	×	×	×	×
1 1 0		?	o	o	×	×	o	o	×	×
1 1 1		?	o	×	o	×	o	×	o	×

我们先简要描述一下我们要处理的具体问题。总结一个可能近似正确（probably approximately correct，PAC）学习模型问题框架：

假定数据按照某概率分布P从X中随机产生，一般，D可为任意分布，并且它对学习型算法是未知的。对于P，所要求的是它的稳定性，即该分布不会随时间变化（不然我们就没有学习的意义了）。训练数据的由P分布随机抽取而产生x，然后x及其目标值（可以理解为y，标签）被提供给学习器，学习器在学习目标函数时考虑可能假设的集合H。在观察了一系列训练数据后，学习器需要从假设集合H中得到最终的假设g，这是对未知的符合D分布的理想模型f的估计。最后，我们通过精心挑选出来的假设g对X中新的数据的性能来评估训练器。

为了描述学习器输出的假设h对真实目标函数f的逼近程度，我们要定义两种错误率：1.真实错误率（$E_{in}(h)$),也可以说是 out-of-sample error,即样本之外，对于从任意分布中抽取的所有数据而言。$E_{in}(h)$表示在抽样样本中，h(x)与y不相等的概率。2.样本错误率（$E_{out}(h)$），也可以说是 in-sample error，即针对所训练的样本数据的。$E_{out}(h)$表示实际所有样本中，h(x)与f(x)不相等的概率是多少。因为h关于f的错误率不能直接由学习器观察到。学习器只能观察到在训练数据上h的性能如何，所以训练器也只能在此性能基础上选择其假设输出。我们用训练错误

率（training error）来指代训练样本中被h误分类的数据所占的比例，以区分真实错误率。那么，数据集合S的样本错误率为数据集合S中被h误分类的数据所占的比例。训练错误率就是当S为训练数据集合时的样本错误率。

我们训练学习器的目标是，能够从合理数量的训练数据中通过合理的计算量可靠的学习到知识。机器学习的现实情况：1.除非对每个可能的数据进行训练，否则总会存在多个假设使得真实错误率不为 0，即学习器无法保证和目标函数完全一致。2.训练样本是随机选取的，训练样本总有一定的误导性。为此，我们要弱化对学习器的要求：1.我们不要求学习器输出零错误率的假设，只要求错误率被限制在某常数ε范围内，ε可为任意小。2.不要求学习器对所有任意抽取的数据都能成功预测，只要求其失败的概率被限定在某个常数μ的范围内，μ可取任意小。

简而言之，我们只要求学习器可能学习到一个近似正确的假设，故得到了“可能近似正确学习”或 PAC 学习。一个可 PAC 学习的学习器要满足两个条件：1、学习器必须以任意高的概率输出一个错误率任意低的假设。2、学习过程的时间最多以多项式方式增长。

对于 PAC 学习来说，训练样本的数量和学习所需的计算资源是密切相关的。如果学习器对每个训练样本需要某最小处理时间，那么为了使目标函数f是可 PAC 学习的，学习器必须在多项式数量的训练样本中进行学习。实际上，为了显示某输出空间的类别C是可 PAC 学习的，一个典型的途径是证明中每个C可以从多项式数量的训练样本中学习到，而后证明每个样本处理时间也限制于多项式级。

2.1.1 PAC 学习概念

初步清楚了 PAC 模型的作用，我们再介绍相关概念整体地理解 PAC 学习。

1.概念c（Concept）：从样本空间到标记空间的映射，c(x) = y，若对任何样例c都成立，则c成为目标概念；所有我们希望学习得到的目标概念构成的集合叫概念类（Concept Class），用C表示。

2.假设空间（Hypothesis Space）：通常用H表示，是给定学习算法$\mathcal{L}$，它所考虑的所有可能的概念集合（即学习算法所有可能的映射，比如线性分类器对应的假设空间是在空间中所有符合该线性参数的可能的超平面），H和C是不同的，学习算法会把自认为可能的目标概念集中成H，而h属于H，称为假设（Hypothesis）是算法输出的一个映射，$h(x) = y$。

3.可分性：若一个算法的假设空间包含目标概念，则称该数据及对该算法是可分的（Separable）或一致的（Consistent）；若一个算法的假设空间不包含目标概念，则称为不可分或不一致的。

4.PAC 学习（Probably Approximately Correct）：概率近似正确学习理论。给定一个数据集，我们希望学得的假设h尽可能与目标概念c一致，这便是概率近似正确的来源，即以较大的概率学的模型满足误差的预备上限。

5.PAC 辨识（PAC Identify）：对ϵ大于 0，δ小于 1，若存在学习算法$\mathcal{L}$，使输出假设h满足：

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

则称学习算法$\mathcal{L}$能从假设空间H中 PAC 辨识概念类C。即若该学习算法能以一个置信度学得假设满足泛化误差的预备上限，或者说学习算法能以较大的概率（$1 - \delta$）学目标概念 c 的近似（误差最多ϵ）则称能 PAC 辨识。

6.PAC 可学习（PAC Learning Algorithm）：若m表示从分布D中独立分布采样得到的样例数目，对所有分布D若存在学习算法$\mathcal{L}$和多项式函数poly使得对于任何$m >= poly(1/\epsilon, 1/\delta, size(x), size(c))$，$\mathcal{L}$能从假设空间H中 PAC 辨识概念类C，则称概念类C对于假设空间H是 PAC 可学习的。该概念在 PAC 辨识的基础上将样本数量考虑进来，当样本超过一定数量学习算法总能 PAC 辨识概念类，则成为 PAC 可学习的。

7.PAC 学习算法：若学习算法$\mathcal{L}$使概念类C为 PAC 可学习的，且$\mathcal{L}$的运行时间也是多项式函数$poly(1/\epsilon, 1/\delta, size(x), size(c))$能表示的，则称概念类C是高效 PAC 可学习（efficiently PAC

Learnable），称$\mathcal{L}$为概念的 PAC 学习算法。该概念将学习器的运行时间（时间复杂度）也考虑了进来。

8.样本复杂度（Sample Complexity）：满足 PAC 学习算法$\mathcal{L}$所需的$m >= poly(1/\epsilon, 1/\delta, size(x), size(c))$的最小的m，称为学习算法$\mathcal{L}$的样本复杂度。

显然，PAC 学习给出一个抽象刻画机器学习能力的框架。PAC 学习的一个关键因素就是假设空间H的复杂度，H包含了算法$\mathcal{L}$所有可能输出的假设。若 PAC 学习中H输出的假设h与概念类完全相同，即$H = C$，称为 PAC 可学习（Properly PAC Learnable）。对于h个数很多的情况，只要有h个数M是有限的，且N足够大，就能保证 $E_{in}(h) \approx E_{out}(h)$，从而证明机器学习是可行的。

2.2 模型的偏差与方差及其泛化性能

2.2.1 泛化性能简介

机器学习的目标是对从真实概率分布（已隐藏）中抽取的新数据做出良好预测。遗憾的是，模型无法查看整体情况；模型只能从训练数据集中取样。如果某个模型在拟合当前样本方面表现良好，那么你如何相信该模型也会对从未见过的样本做出良好预测呢？这就要提及模型的泛化性能。

泛化性能是指机器学习算法对新鲜样本的适应能力。学习的目的是学到隐含在数据背后的规律，对具有同一规律的学习集以外的数据，经过训练的网络也能给出合适的输出，该能力称为泛化能力。训练往往是为了得到泛化性能好的模型，前提假设是训练数据集是实际数据的无偏采样估计。但实际上这个假设一般不成立，针对这种情况我们会使用训练集训练，测试集测试其性能,对于模型估计出泛化性能，同时我们还希望了解它为什么具有这样的性能。这里所说的偏差-方差分解就是一种解释模型泛化性能的一种工具。它是对模型的期望泛化错误率进

行拆解。在有监督学习中，模型的期望泛化误差可以分解成三个基本量的和---偏差、方差和噪声。

样本可能出现噪声，使得收集到的数据样本中的有的类别与实际真实类别不相符。对测试样本x，另y_d为x在数据集中的标记，y 为真实标记，f(x; D) 为训练集D上学得模型f 在x 上的预测输出。以回归任务为例：

模型的期望预测：

$$\bar{f}(x) = E_D[(f(x; D) - \bar{f}(x)^2]$$

噪声：

$$2\epsilon^2 = E_D[(y_D - y)^2]$$

期望输出与真实标记的差别称为偏差：

$$bias^2(x) = [\bar{f}(x) - y)]^2$$

为便于讨论，假设噪声期望为 0，即：$E_D[y - y_D]=0$，通过简单的多项式展开与合并，模型期望泛化误差分解后可得：

$$E(f; D) = bias^2(x) + var(x) + \epsilon^2$$

我们知道，模型在不同训练集上学得的结果很可能不同，即便这些训练集是来自同一个分布。接下来我们来理解它们的概念。

2.2.2 概念

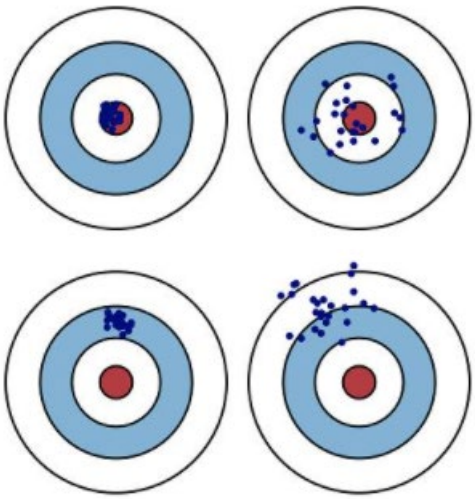
偏差：指的是由所有采样得到的大小为m的训练数据集训练出的所有模型的输出的平均值和真实结果之间的差异，度量了模型的期望预测与真实结果的偏离程度，即刻画了模型本身的拟合能力。偏差通常是由于我们对模型做了错误的假设所导致的，比如真实模型是某个二次函数，但我们假设模型是一次函数。由偏差带来的误差通常在训练误差上就能体现出来。

方差：指的是由所有采样得到的大小为m的训练数据集训练出的所有模型的输出的方差，度量了同样大小的训练集的变动所导致的学习性能的变化，即刻画了数据扰动对模型所造成的影响。方差通常是由于模型的复杂度相对于训练样本数m过高导致的，比如一共有100个训练样本，而我们假设模型是阶数不大于 200的多项式函数。由方差带来的误差通常体现在测试误差相对于训练误差的增量上，换句话说就是体现为训练误差可能很小，但是测试误差却很大。

噪声：表达了在当前任务上任何模型所能达到的期望泛化误差的下界，即刻画了学习问题本身的难度。

上述概念说明，模型的泛化性能是由模型本身的拟合能力、数据的充分性以及学习任务本身的难度所共同决定的。给定学习任务，为了取得好的泛化性能，则需使偏差较小，即能够充分拟合数据，并且使方差较小，即使得数据扰动产生的影响小。

定义可能不够直观，为了更清晰的理解偏差和方差，我们用一个射击的例子，对照上边的描述可以更好的理解这二者的区别和联系。假设一次射击就是一个机器学习模型对一个样本进行预测。射中靶心位置代表预测准确，偏离靶心越远代表预测误差越大。我们通过n次采样得到n个大小为m的训练样本集合，训练出n个模型，对同一个样本做预测，相当于我们做了n次射击，射击结果如图所示。我们最期望的结果就是左上图的结果，射击结果又准确又集中，说明模型的偏差和方差都很小；右上图虽然射击结果的中心在靶心周围，但分布比较分散，说明模型的偏差较小但方差较大；同理，左下图说明模型方差较小，偏差较大；右下图说明模型方差较大，偏差也较大。

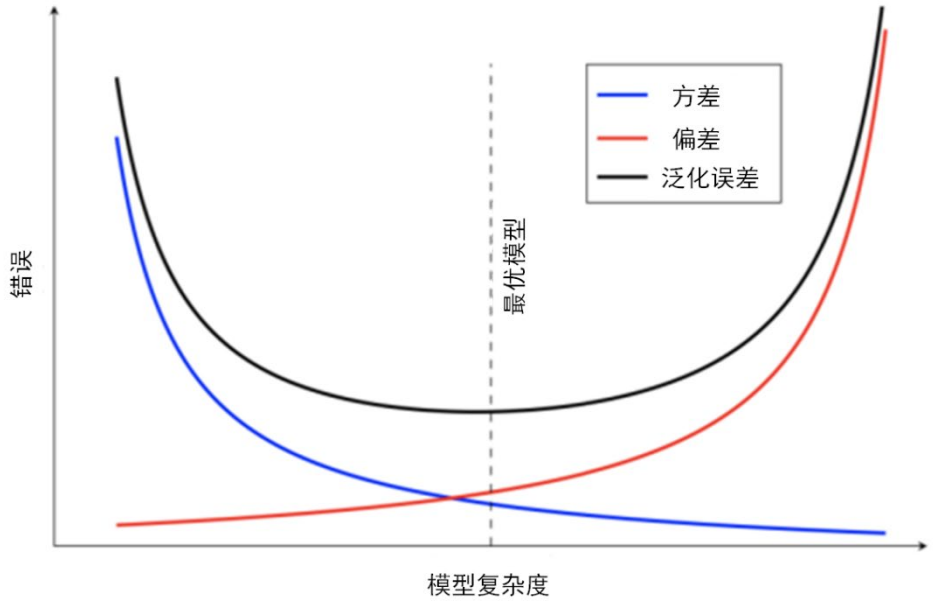


2.2.3 提高泛化性能

为了得到泛化性能好的模型，我们需要使偏差较小，即能充分拟合数据，并且使方差小，使数据扰动产生的影响小。一般来讲，偏差和方差在一定程度上是有冲突的，这称作为偏差-方差窘境。

下图展现了方差、偏差和泛化误差之间的关系。在模型训练不足时，拟合能力不够强，训练数据的扰动不足以使学习器产生显著变化，此时偏差主导泛化误差，此时称为欠拟合现象。当随着训练程度加深，模型的拟合能力增强，训练数据的扰动慢慢使得方差主导泛化误差。当训练充足时，模型的拟合能力非常强，数据轻微变化都能导致模型发生变化，如果过分学习训练数据的特点，则会发生过拟合。

针对欠拟合，我们提出集成学习的概念并且对于模型可以控制训练程度，比如神经网络加多隐层，或者决策树增加树深。针对过拟合，我们需要降低模型的复杂度，提出正则化惩罚项。



我们可以提取几个关键词：新鲜样本、适应能力、规律、合适输出。由此可见，经训练样本训练的模型需要对新样本做出合适的预测，这是提高泛化能力的体现。

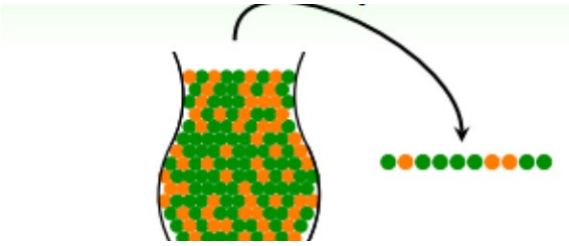
奥卡姆的威廉是 14 世纪一位崇尚简单的修士和哲学家。他认为科学家应该优先采用更简单（而非更复杂）的公式或理论。奥卡姆剃刀定律在机器学习方面的运用如下：机器学习模型越简单，良好的实证结果就越有可能不仅仅基于样本的特性。现今，我们已将奥卡姆剃刀定律正式应用于统计学习理论和计算学习理论领域。这些领域已经形成了泛化边界，即统计化描述模型根据以下因素泛化到新数据的能力：1.模型的复杂程度。2.模型在处理训练数据方面的表现。这些理论都有利于提高模型的泛化性能。

2.3 Hoeffding 不等式与模式二分性

2.3.1 Hoeffding 不等式的应用

在训练集 D 以外的样本上，机器学习的模型是很难，似乎做不到正确预测或分类的。那是否有一些工具或者方法能够对未知的目标函数 f 做一些推论，让我们的机器学习模型能够变得有用呢?

如果有一个装有很多（数量很大数不过来）橙色球和绿色球的罐子，我们能不能推断橙色球的比例 u？统计学上的做法是，从罐子中随机取出 N 个球，作为样本，计算这 N 个球中橙色球的比例 v，那么就估计出罐子中橙色球的比例约为 v。



这种随机抽取的做法能否说明罐子里橙色球的比例一定是 v 呢？答案是否定的。但是从概率的角度来说，样本中的 v 很有可能接近我们未知的 u。下面从数学推导的角度来看 v 与 u 是否相近。

已知 u 是罐子里橙色球的比例, v 是 N 个抽取的样本中橙色球的比例。当 N 足够大的时候, v 接近于 u。根据 Hoeffding 不等式：

$$P[|v - u| > \epsilon] \leq 2exp(-2\epsilon^2 N)$$

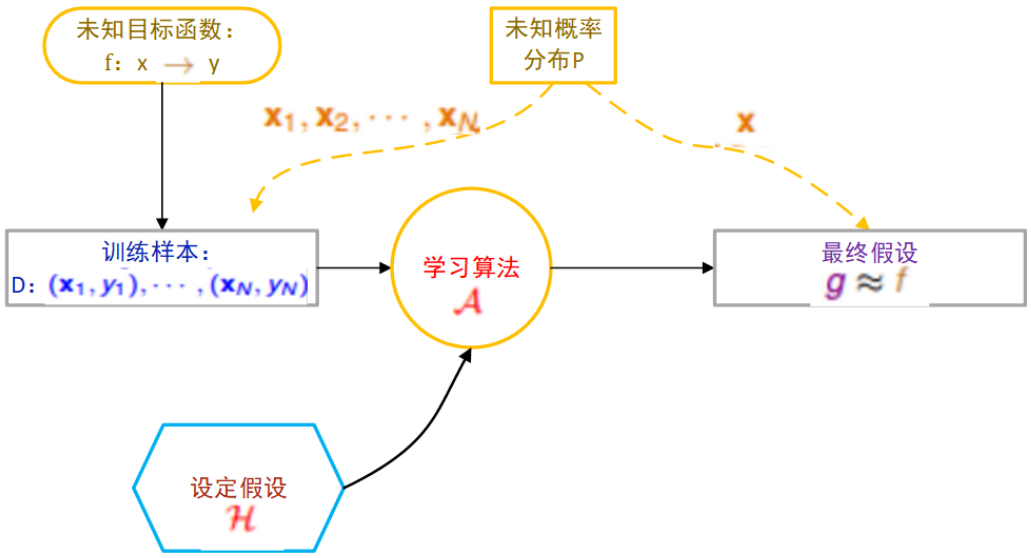
Hoeffding 不等式说明当 N 很大的时候，v 与 u 相差不会很大，它们之间的差值被限定在之内。

所以 PAC 可学习性很大程度上由所需的训练样本数量决定。随着问题规模的增长所带来的所需训练样本的增长称为学习问题的样本复杂度（sample complexity）。在多数实际问题中，最限制学习器成功的因素是有限的可用的训练数据。

我们通常都喜欢能与训练数据拟合程度更高的假设，当一个学习器在可能时都输出能完美拟合训练数据的假设时，我们称该学习器是一致的（consistent）。这就要求训练错误率是 0。遗憾的是，如果假设空间里不总是能找到一个零错误率的假设，这时，最多能要求学习器输出的假设在训练数据上有最小的错误率。在更一般的情形下，我们要考虑学习器有非零训练错误

率的假设时，仍能找到一个边界来限定学习器所需的样本数量。而 Hoeffding 不等式能帮助我们划定这个边界。

我们再来看一下基于统计学的机器学习流程图：



令D代表学习器可观察的特定的训练数据集合，而 P 代表整个数据集合背后满足的概率分布。令$E_{in}(h)$代表假设h的训练错误率，确切的说，$E_{in}(h)$是数据集D中被h误分类的训练数据所占比例，$E_{in}(h)$是定义在训练数据集D上的，而真实错误率$E_{out}(h)$是定义在整个概率分布P上的。现在令g代表H中有最小训练错误率的假设。问题是多少训练数据才足以保证真实错误率$E_{out}(h)$和训练错误率$E_{in}(h)$很接近，并且接近 0。

$E_{in}(h)$和$E_{out}(h)$也可以用 Hoeffding 不等式表示：

$$P[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 2exp(-2\epsilon^2 N)$$

该不等式表明，$E_{in}(h) = E_{out}(h)$ 也是 PAC 的。如果$E_{in}(h) \approx E_{out}(h)$，$E_{in}(h)$很小，那么就能推断出$E_{out}(h)$很小，也就是说在该数据分布P下，h与f非常接近，机器学习的模型比较准确。

但是一般地，h 如果是固定的，N 很大的时候，$E_{in}(h) \approx E_{out}(h)$，但是并不意味着$g \approx f$。因为 h 是固定的，不能保证$E_{in}(h)$足够小，即使$E_{in}(h) \approx E_{out}(h)$，也可 能使$E_{out}(h)$偏大。所

以，一般会通过演算法 A，选择最好的 h，使$E_{in}(h)$足够小，从而保证$E_{out}(h)$很小。固定 h，使用新数据进行测试，验证其错误率是多少。

假设现在有很多罐子 M 个（即有 M 个 hypothesis），如果其中某个罐子抽样的球全是绿色，那是不是应该选择这个罐子呢？我们先来看这样一个例子：150 个人抛硬币，那么其中至少有一个人连续 5 次硬币都是正面朝上的概率是

$$1-\left(\frac{31}{32}\right)^{150}>99\%$$

可见这个概率是很大的，但是能否说明 5 次正面朝上的这个硬币具有代表性呢？答案是否定的！并不能说明该硬币单次正面朝上的概率很大，其实都是 0.5。一样的道理，抽到全是绿色求的时候也不能一定说明那个罐子就全是绿色球。当罐子数目很多或者抛硬币的人数很多的时候，可能引发 Bad Sample，Bad Sample 就是$E_{in}(h)$和$E_{out}(h)$差别很大，即选择过多带来的负面影响，选择过多会恶化不好的情形。

根据许多次抽样的到的不同的数据集 D，Hoeffding's 不等式保证了大多数的 D 都是比较好的情形（即对于某个 h，保证$E_{in}(h) \approx E_{out}(h)$），但是也有可能出现 Bad Data，即$E_{in}(h)$和$E_{out}(h)$差别很大的数据集 D，这是小概率事件。

也就是说，不同的数据集D_n，对于不同的 hypothesis，有可能成为 Bad Data。只要D_n在某个 hypothesis 上是 Bad Data，那么D_n就是 Bad Data。只有当D_n在所有的 hypothesis 上都是好的数据，才说明D_n不是 Bad Data，可以自由选择算法 A 进行建模。那么，根据 Hoeffding's inequality，Bad Data 的上界可以表示为连级（union bound）的形式：

$$
\begin{aligned}
&P_D[BAD\ D] \\
&= P_D[BAD\ D\ for\ h_1\ or\ BAD\ D\ for\ h_2\ or\ \cdots\ or\ BAD\ D\ for\ h_M] \\
&\le P_D[BAD\ D\ for\ h_1]\ + P_D[BAD\ D\ for\ h_2]\ \ + \cdots + P_D[BAD\ D\ for\ h_M] \\
&\le 2\exp(-2\epsilon^2 N) + 2\exp(-2\epsilon^2 N) + \cdots + 2\exp(-2\epsilon^2 N) \\
&= 2M\exp(-2\epsilon^2 N)
\end{aligned}
$$

其中，M 是 hypothesis 的个数，N 是样本 D 的数量， ϵ是参数。该 union bound 表明，当 M 有限，且 N 足够大的时候，Bad Data 出现的概率就更低了，即能保证 D 对于所有的 h 都有 $E_{in}(h) \approx E_{out}(h)$，满足 PAC，演算法A的选择不受限制。那么满足这种 union bound 的情况，我们就可以和之前一样，选取一个合理的演算法（PLA/pocket），选择使$E_{in}(h)$最小的h_m作为矩g，一般能够保证$g \approx f$，即有不错的泛化能力。

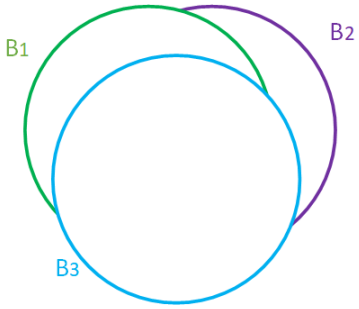
所以，如果 hypothesis 的个数M是有限的，N足够大，那么通过演算法A任意选择一个矩g，都有$E_{in}(h) \approx E_{out}(h)$成立；同时，如果找到一个矩g，使$E_{in}(h) \approx 0$，PAC 就能保证$E_{out}(h) \approx 0$。至此，就证明了机器学习是可行的。

而接下来大的问题是 M 的选择，M 的选择直接影响机器学习两个核心问题是否满足，M 不能太大也不能太小。那么如果 M 无限大的时候，是否机器就不可以学习了呢？例如 PLA 算法中直线是无数条的，但是 PLA 能够很好地进行机器学习，这又是为什么呢？如果我们能将无限大的 M 限定在一个有限的m_H内，问题似乎就解决了。

根据 Hoeffding 不等式，每个 hypothesis 下的 BAD events 级联的形式B_m满足下列不等式：

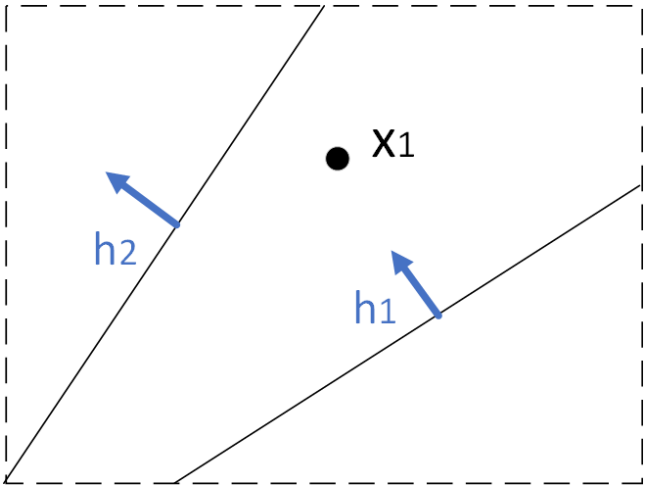
$$P[B_1 \text{ or } B_2 \text{ or } \cdots B_M] \leq P[B_1] + P[B_2] + \cdots + P[B_M]$$

当$M = \infty$时，上面不等式右边值将会很大，似乎说明 BAD events 很大，$E_{in}(g)$ 与$E_{out}(g)$ 也并不接近。但是 BAD events B_M级联的形式实际上是扩大了上界，union bound 过大。这种做法假设各个 hypothesis 之间没有交集，这是最坏的情况，可是实际上往往不是如此，很多情况下，都是有交集的，也就是说 M 实际上没那么大，如下图所示：

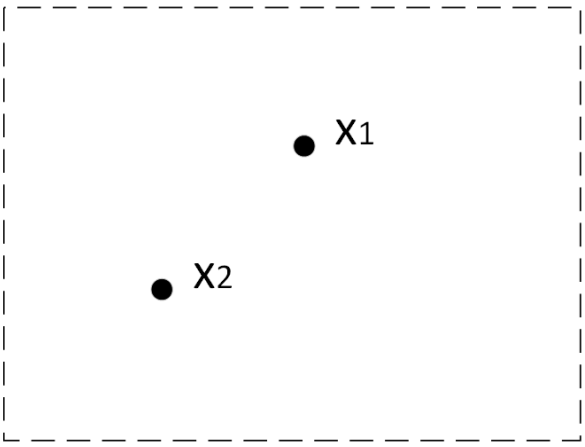


也就是说 union bound 被估计过高了（overestimating）。所以，我们的目的是找出不同 BAD events 之间的重叠部分，也就是将无数个 hypothesis 分成有限个类别。如何将无数个 hypothesis 分成有限类呢？我们先来看这样一个例子，假如平面上用直线将点分开，也就跟 PLA 一样。

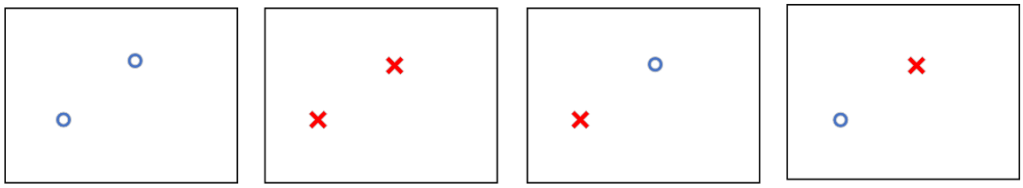
如果平面上只有一个点x_1，那么直线的种类有两种：一种将x_1划为+1，一种将x_1划为-1：



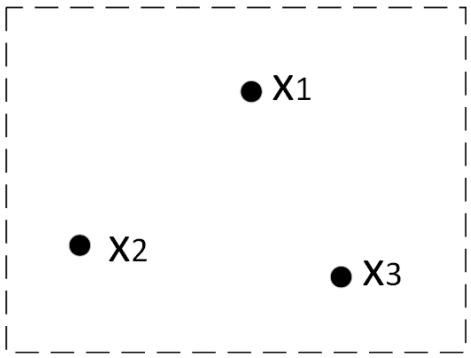
如果平面上有两个点x_1、x_2，那么直线的种类共 4 种：x_1、x_2都为+1，x_1、x_2都为-1，x_1为+1 且x_2为-1，x_1为-1 且x_2为+1：



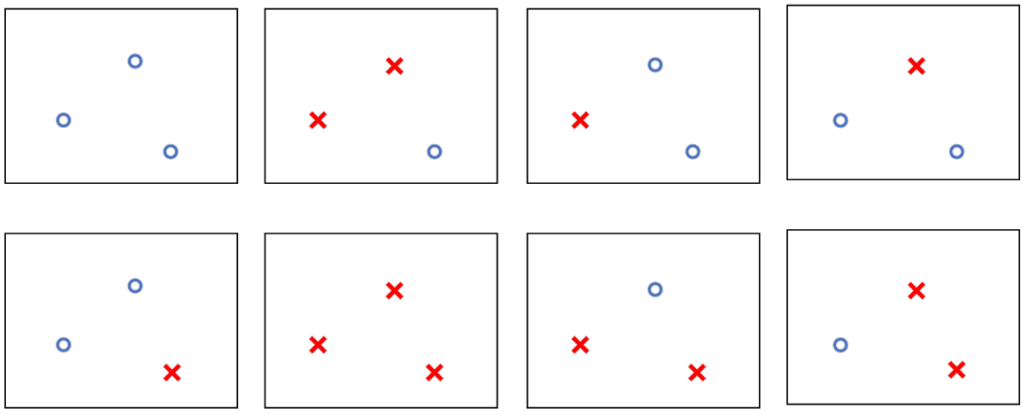
4：



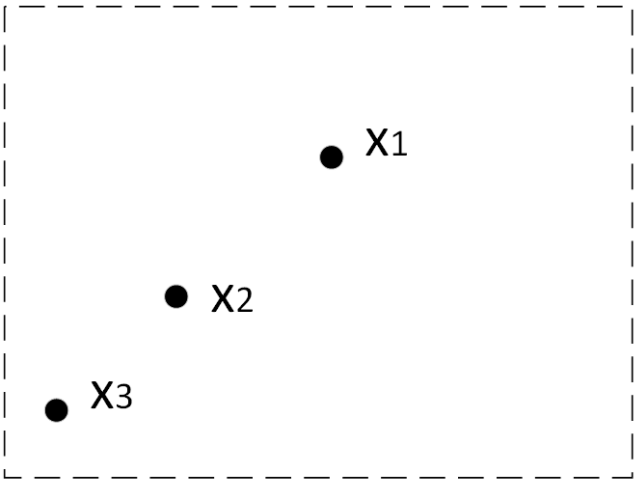
如果平面上有三个点x_1、x_2、x_3，那么直线的种类共 8 种：

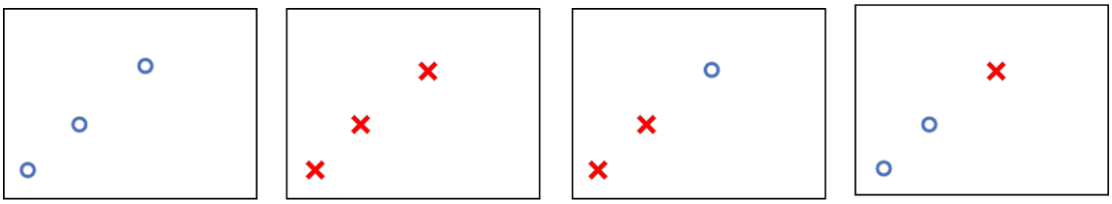


8：



但是，在三个点的情况下，也会出现不能用一条直线划分的情况：





也就是说，对于平面上三个点，不能保证所有的 8 个类别都能被一条直线划分。那如果是四个点x_1、x_2、x_3、x_4，同样的，我们发现，平面上找不到一条直线能将四个点组成的 16 个类别完全分开，最多只能分开其中的 14 类，即直线最多只有 14 种。

经过分析，我们得到平面上线的种类是有限的，1 个点最多有 2 种线，2 个点最多有 4 种线，3 个点最多有 8 种线，4 个点最多有 14（$< 2^4$）种线等等。我们发现，有效直线的数量总是满足$\leq 2^{\mathrm{N}}$，其中，N 是点的个数。所以，如果我们可以用 effective(N)代替 M，Hoeffding 不等式可以写成：

$$\mathrm{P}[|\mathrm{E_{in}}(\mathrm{h}) - \mathrm{E_{out}}(\mathrm{h})| > \epsilon] \leq 2 \cdot \mathrm{effective}（\mathrm{N}） \cdot \exp(-2\epsilon^2 \mathrm{N})$$

已知$\mathrm{effective(N)} < 2^{\mathrm{N}}$，如果能够保证$\mathrm{effective(N)} \ll 2^{\mathrm{N}}$，即不等式右边接近于零，那么即使 M 无限大，直线的种类也很有限，机器学习也是可能的。至此，Hoeffding 不等式展现了其在机器学习理论的重要性，可以说是理论的基石之一。

2.3.2 模式二分性

接下来先介绍一个新名词：二分类（dichotomy）。dichotomy 就是将空间中的点（例如二维平面）用一条直线分成正类（蓝色 o）和负类（红色 x）。H 是将平面上的点用直线分开的所有 hypothesis h 的集合，dichotomy H 与 hypotheses H 的关系是：hypotheses H 是平面上所有直线的集合, 个数可能是无限个, 而 dichotomy H 是平面上能将点完全用直线分开的直线种类, 它的上界是2^{N}。接下来，我们要做的就是尝试用 dichotomy 代替 M。

再介绍一个新的名词：成长函数（growth function），记为$\mathrm{m_H(N)}$。成长函数的定义是：对于由 N 个点组成的不同集合中，某集合对应的 dichotomy 最大，那么这个 dichotomy 值就是$\mathrm{m_H(N)}$，它的上界是2^{N}：

$$m_H(N) = \max_{x_1,\ x_1,\ \cdots x_N \in x} \left| H(x_1,\ x_1,\ \cdots x_N) \right|$$

成长函数其实就是有效直线的数量最大值。根据成长函数的定义，二维平面上，$m_H(N)$随N的变化关系是：

N	$m_H(N)$
1	2
2	4
3	8
4	14

对于2D 实平面，我们之前分析了3个点，可以做出8种所有的dichotomy，而4个点，就无法做出所有16个点的dichotomy了。所以，我们就把4称为2D perceptrons的 break point（5、6、7等都是break point）。另外设有k个点，如果k大于等于break point 时，它的成长函数一定小于2的k次方。 根据break point的定义，我们知道满足的k的最小值就是break point。对于2D 实平面，它的成长函数$m_H(H) = O(N^{k-1})$。那么就可以用$m_H(H)$代替M，就满足了机器能够学习的条件，借此可以完成模式二分性。

总结下，前几节着重介绍了机器能够学习的条件并做了详细的推导和解释。机器能够学习必须满足两个条件：1.假设空间H的M是有限的，即当N足够大的时候，那么对于假设空间中任意一个假设g，$E_{in}(h) \approx E_{out}(h)$。2.利用算法A从假设空间H中，挑选一个g，使$E_{in}(h) \approx 0$，则$E_{out}(h) \approx 0$。这两个条件，正好对应着测试和训练两个过程。训练的目的是使损失期望$E_{in}(g) \approx 0$；test的目的是使将算法用到新的样本时的损失期望也尽可能小，即$E_{out} \approx 0$。 正因为如此，我们引入了break point并得到只要break point存在，则M有上界，一定存在$E_{in}(h) \approx E_{out}(h)$。

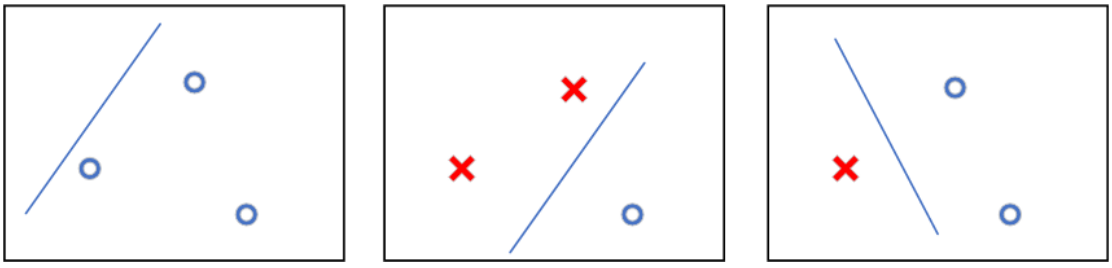
2.4 VC 维理论与模型复杂性

本节介绍一个新的名词：VC 维。VC 维就是某假设集 H 能够 shatter 的最多 inputs 的个数，即最大完全正确的分类能力。（注意，只要存在一种分布的 inputs 能够正确分类也满足）。

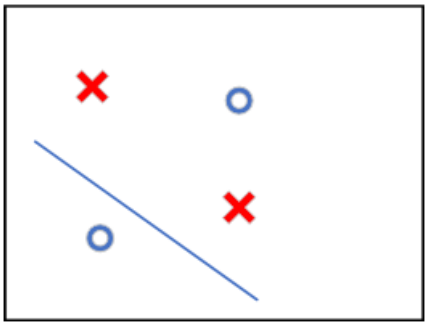
shatter 的英文意思是“粉碎”，对于一个给定集合 S=$\{x_1, x_1, \cdots x_d\}$，如果一个假设类 H 能够实现集合 S 中所有元素的任意一种标记方式，则称 H 能够分散 S。例如对二分类问题来说，N 个输入，如果能够将2^N种情况都列出来，则称该 N 个输入能够被假设集 H shatter。H 的 VC 维表示为 VC(H)，指能够被 H 分散的最大集合的大小。若 H 能分散任意大小的集合，那么 VC(H)为无穷大。通常这样来计算 VC 维度：若存在大小为 d 的示例集能被 H shatter，但不存在任何大小为 d+1 的示例集能被 H 打散，则 H 的 VC 维是 d。下面给出两个例子来计算 VC 维：

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

2.二维实平面上的线性划分：令 H 表示二维实平面上所有线性划分构成的集合，x = R^2。由图可知，存在大小为 3 的示例集可被 shatter，但不存在大小为 4 的示例集可被 H 打散。于是，二维实平面上所有线性划分构成的假设空间 H 的 VC 维为 3。



存在这样的集合，其$2^3 = 8$种对分均可被线性划分实现



对任意集合，其$2^4 = 16$种对分中至少有一种不能被线性划分实现

接下来我们讨论用d_{vc}代替上节介绍的 k，那么 VC 维的问题也就转换为与d_{vc}和 N 相关了。同时，如果一个假设集 H 的d_{vc}确定了，则就能满足机器能够学习的第一个条件$E_{in}(h) \approx E_{out}(h)$，与算法、样本数据分布和目标函数都没有关系。

2D 下的 PLA 算法，已知 Perceptrons 的 k=4，即$d_{vc} = 3$。根据 VC Bound 理论，当 N 足够大的时候，$E_{in}(h) \approx E_{out}(h)$。如果找到一个 g，使$E_{in}(h) \approx 0$，那么就能证明 PLA 是可以学习的。这是在 2D 情况下，那如果是多维的 Perceptron，它对应的d_{vc}又等于多少呢?

已知在一维空间里，$d_{vc} = 2$，在二维空间里，$d_{vc} = 3$，那么我们有如下假设：$d_{vc} = d + 1$，其中 d 为维数。在 d 维里，我们只要找到某一类的 d+1 个 inputs 可以被 shatter 的话，那必然得到$d_{vc} \geq d + 1$。所以，我们有意构造一个 d 维的矩阵X能够被 shatter 就行。X是 d 维的，有 d+1 个 inputs，每个 inputs 加上第零个维度的常数项 1，得到X的矩阵：

$$X = \begin{bmatrix} — x_1^T — \\ — x_2^T — \\ — x_3^T — \\ \vdots \\ — x_{d+1}^T — \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & 0 \\ 1 & 0 & \dots & 0 & 1 \end{bmatrix}$$

矩阵中，每一行代表一个 inputs，每个 inputs 是 d+1 维的，共有 d+1 个 inputs。这里构造的X很明显是可逆的。shatter 的本质是假设空间 H 对X的所有情况的判断都是对的，即总能找

到权重 W，满足$X * W = y$ ，$W = X^{-1} * y$ 。W又名 features，即自由度。自由度是可以任意调节的。VC Dimension 代表了假设空间的分类能力，即反映了 H 的自由度，产生 dichotomy 的数量, 也就等于 features 的个数, 但也不是绝对的。例如, 对 2D Perceptrons, 线性分类, $d_{vc} = 3$, 则$W = \{W_0, W_1, W_2\}$ ，也就是说 只要 3 个 features 就可以进行学习，自由度为 3。

而由于这里我们构造的矩阵X 的逆矩阵存在，那么 d 维的所有 inputs 都能被 shatter，也就证明了第一个不等式。

在 d 维里，如果对于任何的 d+2 个 inputs，一定不能被 shatter，则不等式成立。我们构造一个任意的矩阵X，其包含 d+2 个 inputs，该矩阵有 d+1 列，d+2 行。这 d+2 个向量的某一列一定可以被另外 d+1 个向量线性表示，例如对于向量X_{d+2}，可表示为：

$$X_{d+2} = a_1 * X_1 + a_2 * X_2 + \cdots + a_d * X_d$$

其中，假设$a_1 > 0$，a_2，$\cdots a_d < 0$， 那么如果X_1是正类，X_2，…，X_d均为负类，则存在 W，得到如下表达式：

$$X_{d+2} * W = a_1 * X_1 * W + a_2 * X_2 * W + \cdots + a_d * X_d * W > 0$$

对于这种情况，X_{d+2}一定是正类，无法得到负类的情况。也就是说，d+2 个 inputs 无法被 shatter。 证明完毕！ 综上证明可得 $d_{vc} = d + 1$。

下面，我们将更深入地探讨 VC Dimension 的意义。根据之前的泛化不等式，如果 $|E_{in}(h) - E_{out}(h)| > \epsilon$，即出现 bad 坏的情况的概率最大不超过δ。那么反过来，对于 good 好的情况发生的概率最小为$1 - \delta$，则对上述不等式进行重新推导：

$$P[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 4 \cdot 2N^{d_{vc}} \cdot \exp(-\frac{1}{8}\epsilon^2 N)$$

其中等式后半部分记为δ。根据之前的泛化不等式，如果$|E_{in}(h) - E_{out}(h)| > \epsilon$，即出现 bad 坏的情况的概率最大不超过δ。那么反过来，对于 good 好的情况发生的概率最小为$1 - \delta$，则对上述不等式进行重新推导：

$$\epsilon = \sqrt{\frac{8}{N}\ln\left(\frac{4\ (2N)^{d_{vc}}}{\delta}\right)}$$

ϵ表现了假设空间 H 的泛化能力，ϵ越小，泛化能力越大。

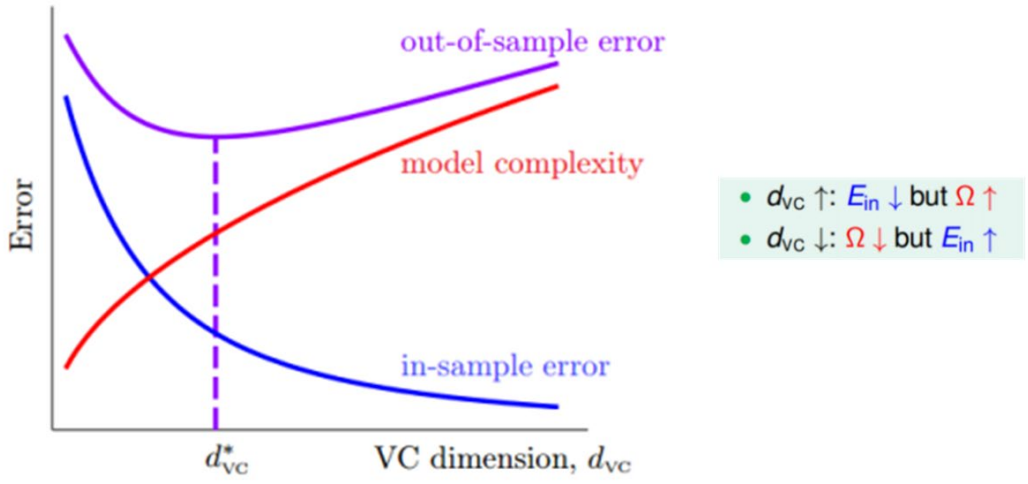
进而得到：

$$E_{in}(g) - \sqrt{\frac{8}{N}\ln\left(\frac{4\ (2N)^{d_{vc}}}{\delta}\right)} \le E_{out}(g) \le E_{in}(g) + \sqrt{\frac{8}{N}\ln\left(\frac{4\ (2N)^{d_{vc}}}{\delta}\right)}$$

至此，已经推导出泛化误差E_{out}的边界，因为我们更关心其上界（E_{out}可能的最大值），即：

$$E_{out}(g) \le E_{in}(g) + \sqrt{\frac{8}{N}\ln\left(\frac{4\ (2N)^{d_{vc}}}{\delta}\right)}$$

上述不等式的右边第二项称为模型复杂度，其模型复杂度与样本数量 N、假设空间 H(d_{vc})、ϵ有关。E_{out}由E_{in}共同决定。下面绘出E_{out}、模型复杂度Ω、 E_{in}随d_{vc}变化的关系：



通过该图可以得出如下结论：

- d_{vc}越大，E_{in}越小，Ω越大（复杂）。

- d_{vc}越小，E_{in}越大，Ω越小（简单）。
- 随着d_{vc}增大，E_{out}会先减小再增大。

所以，为了得到最小的E_{out}，不能一味地增大d_{vc}以减小E_{in}，因为E_{in}太小的时候，模型复杂度会增加，造成E_{out}变大。也就是说，选择合适的d_{vc}，选择的自由度个数要合适。

习题.

2.1

分别简述下 PAC 学习、PAC 辨识以及 PAC 可学习。

2.2

谈谈自己对机器为什么可以学习的理解。

2.3

试推论公式 $E(f;D) = bias^2(x) + var(x) + \epsilon^2$

2.4

如何提升机器学习的泛化能力?

2.5

解释下 Hoeffding 不等式与模式二分性之间的关系。

2.6

试证明R^d空间中线性超平面构成的假设空间的 VC 维是 d+1。