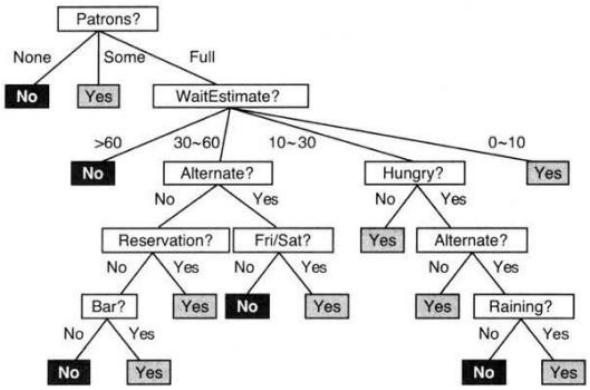
文献信息： 罗素和诺维格的《人工智能：一种现代方法》第三版中的第 18 章“样例学习”。

研究背景及意义：Agent 是善于学习的，但设计者不能预测 Agent 可能驻留的所有情境，如走迷宫的机器人可能遇到的新迷宫的布局；设计者也不能预测随时间推移可能出现的所有变化，如预测未来天气状况要根据条件发生的变化自身学习判断；设计者本身可能对程序求解没有思路，比如人脸识别不能一开始编出完成此任务的计算机程序，只能使用学习算法。利用 Agent 可以对自我经验学习的特性使用决策树学习，达到从一组“输入-输出”对中学习能够预测新输入相对应的输出的函数，之后可以应用到各种学习系统中：线性模型、神经网络、非参数模型和支持向量机。

研究内容：

聚焦于输入值是离散的和输出值为二值的情况即布尔分类，把样例分类为真（正例）或假（反例），决策树通过执行一系列测试达到决策，如书本中的示例：



可以发现在此例中目标决策能够表示成一系列输入属性的集合：

$$Goal \Leftrightarrow (Path_1 \vee Path_2 \vee \cdots)$$

接下来就是研究如何从样例中归纳出决策树，计算发现无论用何种方法进行度量规模的话，要找到极小一致树都是很困难的，不存在高效方法，但是如果研究良好的近似解——小规模的一致树，可用“分化-征服”的策略，总是优先测试最重要属性，并将问题分解为更小的子问题，就可以达到递归求解的目的迭，直到代到剩余样例全是正例或者反例时即事情完毕。

我们发现样例集对于树的构造是至关重要的，然而样例本身不会在树中出现，因此当存在几个重要性相似的变量，在随意抉择它们时，对于稍许不同的输入样例集，用作分裂的首选变量不同可能使得整个决策树看起来完全不同，虽然树所计算的函数是相似的，但是树的结构变化很大。为了验证精度，我们可以将样例划分为训练集和测试集，绘制学习曲线，测量其精度。

为了更好的选择测试属性的好坏，我们将一些属性判定为“相当好”、“真正无用”，并选

择使用熵和信息收益进行衡量:

$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right)$$

$$Gain(A) = B\left(\frac{p}{p+n}\right) - Remainder(A)$$

在对属性的影响有了衡量之后可以通过决策树剪枝的技术来减轻过度拟合。

在了解决策树之后仍存有一个主要问题：怎样确定学习算法已经产生了一个能够正确预测未知输入值的假说，或是说在不知道目标函数 f 的情况下,如何验证假说 h 接近 f?决策树里的学习曲线回答了部分这个问题，但是学习曲线是局限于特定问题的特定学习算法。

提出使用 PAC 学习算法即任何返回概率近似正确的假说的学习算法，定义一个期望泛化误差:

$$\text{error}(h) = GenLoss_{L_{0/1}}(h) = \sum_{x,y} L_{0/1}(y, h(x))P(x,y)$$

要使一个假说被称为是近似正确的，即$^{\text{error}(h)\leqslant\varepsilon}$，ε 是一个小常量，我们所假设的近似正确空间便在环绕真实函数 f 的 ε 球中，球体之外的假说空间称为$\mathcal{H}_{bad}$，那么任何一个严重错误的假说都是属于$\mathcal{H}_{bad}$中的，$\mathcal{H}_{bad}$至少包含一个一致假说的概率受限于个体概率之和:

$$P(\mathcal{H}_{bad}\text{包含一个一致假说}) \leqslant |\mathcal{H}_{bad}|\,(1-\varepsilon)^N \leqslant |\mathcal{H}|\,(1-\varepsilon)^N$$

进一步可将这个概率压缩小到低于某个小数值 δ

$$|\mathcal{H}|\,(1-\varepsilon)^N \leqslant \delta$$

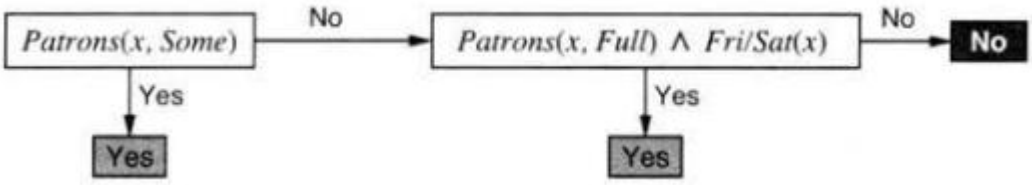
那么我们可以得出:

$$N \geqslant \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln|\mathcal{H}|\right)$$

N 作为 ε 和 δ 的函数便是所要求的样例数目，也可称为假设空间的样本复杂度。

然而，为了获得未知样例的真实泛化，似乎需要对假设空间做出某些限制；但是，一旦做出限制，就会将某些真实函数从假说空间中排除。有三种途径可摆脱这样的困境。第一，引入相关于问题的先验知识。第二，坚持让算法不是返回任意一致假说，而是优先返回简单的假说，在发现简答假说一致假设可行的条件下，样本复杂度的结果一般好于仅基于一致性所分析的结果。第三，聚焦于整个布尔函数假说空间中的可学习子集并依赖于一个假设：受限语言包含一个与真实函数 f 足够近的假说 h；其好处是受限假说空间允许有效泛化且一般容易搜索。

将 PAC 学习应用于一个新的假说空间：决策表。其规定，当施加于一个样例描述时，测试成功应返回的值。如果测试失败，将继续表中的下一项测试。决策表类似决策树，但其整体结构更简单：它们仅在一个方向分支。相反，个体测试更复杂。如果将每个测试限制为最多包含 k 个文字，则学习算法有可能从小数目的样例中成功泛化。这个语言被称为 k-DL，K-DL 以子集的形式包含 k-DT，深度最多为 k 的决策树集合。被 k-DL 指称的特定语言依赖于描述样例所使用的属性。便可将之前的决策树问题描述成决策表形式：



总结与启发：

通过对决策树和决策表的学习了解我们知道：决策树能够表示所有布尔函数，信息收益启发式提供了发现简单一致决策树的有效方法；学习算法的性能由学习曲线度量，它表明了在测试集上的预测精度，而该预测精度是训练集大小的函数。因其衡量使用的局限性，我们又提出了 PAC 学习算法，使用 PAC 对学习算法进行样本判定，并通过 PAC 提出决策表模型。这种由很自然的思维模式提出一种思想，再不断修改完善的建模思路，是普世值得学习的，并且决策学习作为当前机器学习的基础，为我在当前学业完善知识储备有了不可言喻的帮助。

习题：18.3 假设我们从决策树生成了一个训练集，然后将决策树学习应用于该训练集。当训练 集的大小趋于无穷时，学习算法将最终返回正确的决策树吗？为什么是或不是？

该算法可能不会返回“正确”的树，但它会返回一个逻辑上等价的树，假设生成示例的方法最终会生成所有可能的输入属性组合。因为根据定义，在所有可能的示例上一致的同一组属性上定义的任何两个决策树在逻辑上都是等效的。 树的实际形式可能会有所不同，因为有许多不同的方式来表示相同的功能。（例如，具有两个属性 A 和 B，我们可以拥有一棵以 A 为根的树，另一棵以 B 为根的树。）