

扩展阅读

这部分的大部分内容，均出自一人之手，他就是[罗斯·昆兰](#)。下面来自西瓜书的扩展阅读：

决策树学习算法最著名的代表是 ID3 [Quinlan, 1979, 1986]、C4.5 [Quinlan, 1993] 和 CART [Breiman et al., 1984]。[Murthy, 1998] 提供了一个关于决策树文献的阅读指南。C4.5Rule 是一个将 C4.5 决策树转化为符号规则的算法 [Quinlan, 1993]，决策树的每个分支可以容易地重写为一条规则，但 C4.5Rule 算法在转化过程中会进行规则前件合并、删减等操作，因此最终规则集的泛化性能甚至可能优于原决策树。

本质上，各种特征选择方法均可用于决策树的划分属性选择。特征选择参见第 11 章。

在信息增益、增益率、基尼指数之外，人们还设计了许多其他的准则用于决策树划分选择，然而有实验研究表明 [Mingers, 1989b]，这些准则虽然对决策树的尺寸有较大影响，但对泛化性能的影响很有限。[Raileanu and Stoffel, 2004] 对信息增益和基尼指数进行的理论分析也显示出，它们仅在 2% 的情况下会有所不同。4.3 节介绍了决策树剪枝的基本策略；剪枝方法和程度对决策树泛化性能的影响相当显著，有实验研究表明 [Mingers, 1989a]，在数据带有噪声时通过剪枝甚至可将决策树的泛化性能提高 25%。

关于感知机和神经网络，参见第 5 章。

多变量决策树算法主要有 OC1 [Murthy et al., 1994] 和 [Brodley and Utgoff, 1995] 提出的一系列算法。OC1 先贪心地寻找每个属性的最优权值，在局部优化的基础上再对分类边界进行随机扰动以试图找到更好的边界；[Brodley and Utgoff, 1995] 则直接引入了线性分类器学习的最小二乘法。还有一些算法试图在决策树的叶结点上嵌入神经网络，以结合这两种学习机制的优势，例如“感知机树” (Perceptron tree) [Utgoff, 1989b] 在决策树的每个叶结点上训练一个感知机，而 [Guo and Gelfand, 1992] 则直接在叶结点上嵌入多层神经网络。

有一些决策树学习算法可进行“增量学习” (incremental learning)，即在接收到新样本后可对已学得模型进行调整，而不用完全重新学习。主要机制是通过调整分支路径上的划分属性次序来对树进行部分重构，代表性算法有 ID4 [Schlimmer and Fisher, 1986]、ID5R [Utgoff, 1989a]、ITI [Utgoff et al., 1997] 等。增量学习可有效地降低每次接收到新样本后的训练时间开销，但多步增量学习后的模型会与基于全部数据训练而得的模型有较大差别。

文中提到的 OC1 感兴趣可以了解。现代机器学习里，更多人会选择 **随机森林 / 梯度提升树 (GBDT)**，这些集成方法，它们的性能普遍比单棵树要好，OC1 就显得比较“小众”了。

来自群友的小 tip：



Java大王喵～ 昨天 20:55

不

这部分内容，看到多变量决策树一整个被前人的智慧折服了

@Java大王喵～ 决策树 trie 自动机，一个东东的不同名字

wiki 🤖 揉了揉魔理沙的呆毛