2009年人工智能和计算智能国际会议

使用遗传算法修剪决策树

陈杰，王锡锌，翟俊海
河北大学数学与计算机科学学院机器学习中心
保定，中国
电子邮件：jaychen008@163.com

***摘要*--遗传算法是机器学习中常用的方法之一。在本文中，我们提出了一种用于修剪决策树的遗传算法方法。本文采用了二进制编码，即群体中的个体由固定数量的权重组成，代表了一个候选解决方案。评价函数考虑了决策树在测试集上的错误率。遗传算法的三个常见运算符，如随机变异和单点交叉，被应用于群体。最后，该算法将一个具有最高适配度的个体作为局部最优权重返回。基于UCI的四个数据库，我们将我们的方法与其他几个传统的决策树修剪技术进行了比较，包括成本复杂度修剪、悲观误差修剪和减少误差修剪。结果表明，我们的方法与其他修剪方法有更好或相同的效果。**

关键词过拟合；修剪决策树；遗传算法

I. 简介

归纳任务的目的是为了从具体的例子中提取知识。有几种方法已经被开发出来，用于通过归纳推理建立基于知识的系统。作为一种归纳学习，决策树学习从例子的集合中构建知识，并将获得的知识描述为决策树。

由J.R. Quinlan开发的ID3算法是最广泛使用的决策树学习算法之一。这种方法使用由一组属性描述的例子创建一个规则树。

对于一个给定的例子集S，ID3评估每个属性的信息增益，并选择具有最大信息增益的属性作为分割属性。然后，S的例子被分割成N个子集，N是分割属性的数量。这个自上而下的过程一直持续到子集中的所有或大多数例子都属于同一类别为止。

ID3学习算法使用统计学属性，称为熵，它衡量一个属性的价值。它对错误具有鲁棒性，包括对训练实例的分类错误和描述这些实例的属性值的错误。

II. 过度拟合

由ID3生成的决策树是准确和高效的，它们往往有过度复杂的缺点，因此是不可理解的。

除了决策树的不透明性，还应该考虑以下现象。由于决策树的每个分支的生长都足够深入，直到它对训练实例进行完美分类，因此它在训练集中的误差很小。然而，这种决策树在独立的测试集中的表现并不像它在训练集中那样好。准确率先是上升，然后下降。如图1中的图示。

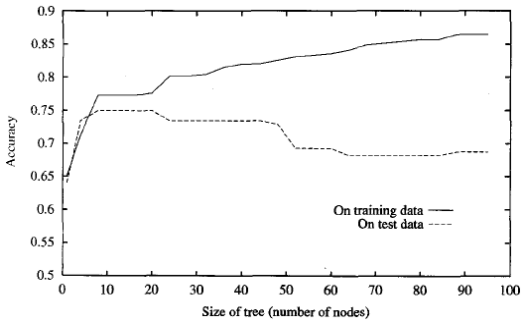


图1.决策树学习中的过拟合

随着决策树的成长，在训练集上测量的准确率单调地增加。然而，在测试集上测得的准确率首先增加，然后减少。

当一棵决策树（或一个假设）在训练集中的表现优于它在测试集中的表现时，我们将说该决策树过度适应训练集。定义如下：

给定一个假设空间H，如果存在一些替代假设h' H，使得h在训练实例上的误差比h'小，但h'在整个实例分布上的误差比h小，则称一个假设h□H过拟合训练数据。

978-0-7695-3816-7/09 $26.00 © 2009 IEE doi
10.1109/aici.2009.351

一个可能的原因是训练集中存在随机误差或噪音。 为了说明这一点，表1中给出了一个训练集。

表一：　　训练集

属性	级别
1.1	+
0.5	+
2.8	+

图2显示了由该训练集归纳出的决策树。

图2.从训练集得出的决策树

现在考虑有一个正面的训练例子被错误地标记为负面。

表二：有噪声的　　训练集

属性	级别
1.1	+
0.5	+
2.7	-

为了适应训练集，ID3将找到一个新的决策属性，将这个新的例子从两个正面例子中分离出来。其结果是，ID3将输出一个决策树（或假设h），如图3所示，比原始树（或假设h）更复杂。

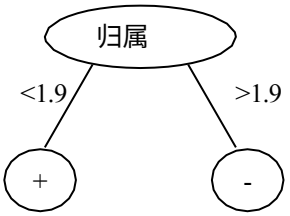


图3.从训练集得出的决策树

假设h将完全适合训练集，但h'不会。然而，决策树的右侧分支仅仅是拟合噪声实例的结果，我们期望h在从相同实例分布中抽取的后续数据中表现优于h'。

事实上，过拟合不仅发生在训练集有随机噪声的情况下，而且在训练集没有噪声的情况下也有可能发生，特别是当叶子节点中的例子数量很少时。在这种情况下，很可能会出现巧合的规律性，一些属性可以很好地划分这些例子，尽管它与分类无关。只要存在这种巧合的规律性，就有可能出现过度拟合的情况。

III.　修剪决策树的概述

为避免过度拟合，提出了两大类方法：

预修剪：在对训练集进行完美分类之前，根据一些停止标准提前停止树的生长。

一个直接的方法是，当树的大小达到一定的阈值时就停止生长。

预修剪方法通过一些标准停止分割节点，以避免增长不必要的子树，而不是先扩展一个完整的树，然后修剪一些分支。与使用ID3生长完整的决策树相比，它所花费的时间要少。然而，它在早期就停止生长，一些只能在以后发现的树的结构可能会被遗漏。

后期修剪：后修剪方法不限制树的生长。首先根据某种方法生成一棵完全适合训练集的树，然后根据某些标准对该树进行修剪。

在实践中，后修剪法比前修剪法有更好的表现。目前有很多基于不同启发式方法的方法。

Breiman等人介绍了成本-复杂性修剪（CCP）方法。这种方法提出了一个决策树的成本-复杂性模型。通过使用成本-复杂性修剪策略，从初始树中诱导出一系列的修剪树。最后，将错误率不大于标准误差的最小树作为修剪后的树来观察。

悲观的错误修剪（PEP）和减少错误修剪是由J. R. Quinlan提出的。

悲观的错误修剪使用二项分布的连续性修正来提供一个更现实的错误率，而不是训练集中乐观的错误率。这个过程从上到下修剪分支，每个子树最多被检查一次，因此速度非常快。此外，它不需要与生成树的训练集分开的测试集。

与悲观的错误修剪不同，减少的错误修剪直接使用测试集修剪决策树。

给定一个完整的树T，对于T的每一个非叶子的子树S，我们用S中最好的叶子来代替子树，得到一个新的树T'。如果新树T'在测试集上的错误数等于或少于原树T的错误数，并且S不包含具有相同性质的子树，那么S就被替换成叶子。

这个过程反复进行，直到任何进一步的替换都会增加测试集的错误数量。

减少误差修剪找到了关于修剪集的最准确子树的最小版本。

另一方面，减少误差的修剪有过度修剪的趋势，特别是有一些罕见的案例在测试集中没有体现。

其他一些修剪决策树的方法也被用于许多领域，有其优势和劣势。这些修剪方法没有一个比其他方法有更好的性能。

IV. 使用遗传算法对决策树进行修剪

从上面关于修剪方法的讨论来看，修剪是一个将原始决策树的一些子树切下来的过程。

现在我们把这个决策树看作是一个由顶点和边组成的有向无环图。一个顶点是一个属性节点或一个类别节点，两个顶点由一条边连接。

当修剪一棵决策树时，在某些标准下，一些边缘被移除。

如果给一条边的权重为0，意味着两个节点之间有一条边；而1意味着节点是断开连接的。决策树的结构将在修剪过程中保持不变，即把连接修剪后的子树S的边的权重定为0。例如，给定一个决策树T。

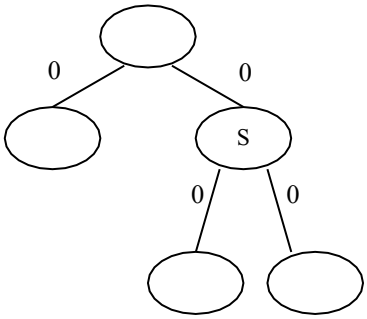


图4.原始决策树

现在我们通过一定的修剪方法对子树S进行修剪，修剪后的决策树可以表示为图5。

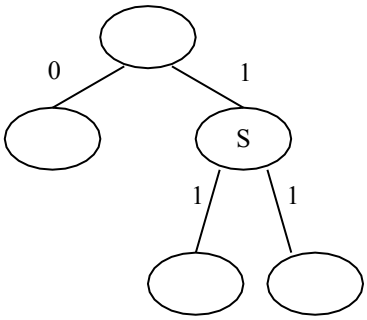


图5.修剪子树S后的决策树

当我们使用这个修剪过的树对未见过的例子进行分类时，如果权重为0，决策树就会给例子提供子树S中最常见的标签。

为了修剪决策树，我们需要寻找一组属性测试节点的权重，使测试集的错误率最小。我们将权重表示为一个向量，然后使用遗传算法来寻找一组最佳权重。

如上所述，修剪过程相当于寻找一组边缘权重，其值为0或1。当具有这些权重的决策树在测试集上的错误率最低时，它就是我们想要的最佳修剪树。

我们可以把修剪决策树的过程看作是优化决策树中边的权重的过程。

A. 遗传算法

遗传算法已被成功地应用于广泛的优化问题。它是一种以自然法则为动力的学习方法，即 "适者生存，不适者淘汰"。这个过程形成了一个生成

和测试的波束搜索策略，而不是从一般到特定的假说，或从简单到复杂的搜索。在每一步，一组被称为当前种群的假说通过使用遗传算子进行更新，它通过变异和重组种群的一些部分来产生后续假说。通过评估每个假说的适用性，当前最好的假说最有可能维持到下一代。

众所周知，进化是生物系统中一种自适应的、稳健的方法。遗传算法搜索候选假说的空间，它可以有效地减少时间成本。

要将遗传算法应用于一个优化问题，首先应该定义其中的一些要素。

1) 表示假设

遗传算法中的假设通常用位串表示，这样我们就可以很容易地利用遗传算子，如变异和交叉。就像与生物进化相类似，群体中的一个假设被称为基因。位串的数量是固定的或不固定的。

一个常见的编码模式是这样的

0	1	1	0	0

图6.基因的编码形式

2) 遗传操作符

该算法通过一组运算符重组和突变当前种群中的选定基因来生成下一代的继承者。两个最常见的运算符是交叉和变异。

交叉运算符通过复制每个父本的片段，从其父本产生两个新的子代。

突变算子通过在随机位置以小概率颠倒所选基因的一个位来产生一个新的后代。

图7说明了操纵位串以产生新假设的典型遗传操作符。

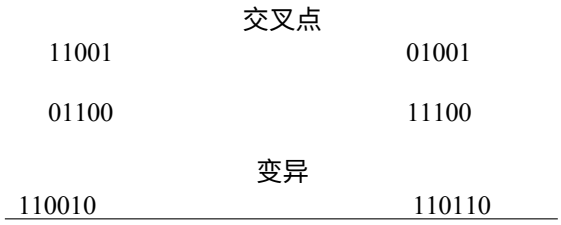
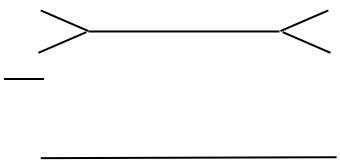


图7.遗传算法的常用操作符

3) 健身函数和选择

最后但最重要的概念是假设的衡量标准。在生成后继种群时，遗传算法根据一定的健身函数对当前种群中的个体进行排名。由于各种问题的存在，不同的健身函数被定义。这些健身函数的共同目的是正确和合理地估计每个个体的价值。

在按衡量标准对这些假说进行排序后，在当前群体中以概率方式选择一定数量的假说来产生下一代。一种有时被称为轮盘式选择的方法通常是



使用，它通过其适配度与其他成员的适配度之比来选择一个假设，如（1）所示。

$$\Pr(_{hi}) = \frac{健身(_{hi})}{\sum_{j=1}^{p} Fitness(h)_j} \quad (1)$$

B. 决策树的最佳权重

基于第四节的讨论，我们将通过优化树的边的权重来修剪决策树。

对于一个给定的由ID3归纳的决策树，边的数量是固定的。我们将假设hi的编码模式定义为

0	1	.....	0	0

基因的长度等于决策树中边的数量，基因的每个分量被赋予1或0，意味着该边是否被切割。

我们想找到一棵最小的树，它在测试集上的误差要小于或等于原始树。所以健身函数可以定义为（2）。

$$Fitness(_{hi}) = \alpha N(T) + \beta E(T) \quad (2)$$

N(T)是决策树T的节点数；E(T)是决策树T的错误数；α，β是关于树的大小和树的错误数的两个权重。

主要步骤如下：

步骤1：

随机生成p个假设，初始化种群P。 步骤2：

对P中的每个h计算Fitness(h) Step3：

创造新的一代，Ps：

- 选择：概率性地选择P中的某些成员加入到Ps中。 从P中选择假设hi的概率Pr(hi)由公式1给出。
- 交叉：根据上面给出的Pr(hi)，从P中概率性地选择某些假设对。对于每一对（hl, h2），通过应用交叉运算产生两个后代。将所有子代加入Ps。
- 突变：以均匀的概率选择Ps中m%的成员。对于每个成员，在其表示中随机选择一个位进行反转。

第四步：用P更新Ps。

该方法重复步骤2至步骤4，直到在迭代过程中保持最大的适配性。最终具有最大适配度的假设包括决策树中边的最佳权重。

V. 实验性

在本节中，我们讨论了通过遗传算法修剪决策树的实验结果。我们以UCI的虹膜数据集为例来说明这一过程。

连续值的属性。根据未修剪的决策树的结构，我们可以得到构建一个假设的元素数量。然后应用遗传算法来优化一组基因。最后得出最佳权重集

是该算法的输出。图9显示了未经处理的决策树的结构。

虹膜数据集中剩下的50个例子被用来评估T的错误分类数量。

算法，找到了最佳基因。

0	0	0	0	1	1	1	1	1	1	1	1

由该基因形成树时，图8中显示的普鲁德决策树。

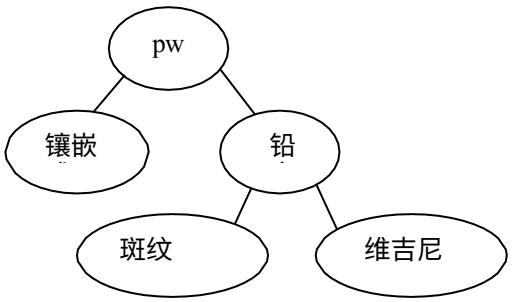


图8.用GAs修剪决策树

首先，我们用ID3归纳出一个决策树，作为未修剪的树。训练集包括随机选择的100个例子，并使用单个阈值来区分

本实验中使用了UCI的四个数据集。我们将其中三分之二的数据作为训练集，将剩余的数据作为测试集。

结果表明，基于遗传算法的修剪方法在这些数据集中具有良好的性能。在不损失精度的前提下，该方法可以有效地减少树的大小。

表3报告了节点数量和分类精度的汇总。

VI. 总结

在本文中，我们介绍了一种基于遗传算法的决策树修剪方法。通过将决策树视为一个有向无环图，我们将修剪操作转换为优化边的权重的过程。实验结果证明了遗传算法的应用是正确和有效的。该方法的性能接近于传统的修剪方法。与基于统计学的剪枝策略不同的是，该方法在一些例子较少或缺乏统计规律性的数据集上可能会有更好的结果。

鸣谢

该研究得到了河北省自然科学基金（F2008000635）、河北省应用基础研究重点项目基金（08963522D）、河北省教育厅第一批100名优秀创新科学家计划和河北省科学研究基金（06213548）的资助。

参考文献

[1] J. H. Holland，"自然和人工系统的适应"，密歇根大学出版社，1967。

[2] L.Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees.Belmont, Calcification：Wadsworth, 1984.

[3] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp.81-106, 1986.

[4] J.R. Quinlan, "Simplifying Decision Trees," International Journal of Man-Machine Studies, vol. 27, pp.221-234, 1987.

[5] J.Mingers, "An Empirical Comparison of Pruning Methods for Decision Tree Induction," Machine Learning, vol.4, no.2, pp.227-243, 1989.

[6] J.R. Quinlan, R. Rivest, "Inferring Decision Trees Using the Minimum Description Length Principle," Information and Computation, vol. 80, pp.227-248, 1989.

[7] J.R. Quinlan, C4.5: Program for Machine Learning, Morgan Kaufmann Publishers, Inc., 1993.

[8] Michael Pazzani, Christopher Merz, Patrick Murphy, Kamal Ali, Timothy Hume and Clifford Brunk, "Reducing misclassification costs," Proc. 11 International Conference on Machine Learning, pp.217-225, 1994。

[9] D.Mehta, J. Rissanen, R. Agrawal, "MDL-Based decision tree pruning," Proceedings ofth First International Conferce on Knowledge Discovery and Data Mining, pp.216-221, 1995。

[10] Floriana Esposito, Donato Malerba, Giovanni Semeraro, "A Comparative Analysis of Methods for Pruning Decision Trees, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, NO.5, 476-491, May 1997.

[11] Boonserm Kijsirikul和Kongsak Chongkasemwongse, "Decision Tree Pruning Using Backpropagation Neural Networks," Neural Networks, Proceedings.IJCNN apos;01.国际联合会议第3 卷，第1876- 1880 页，2001年。

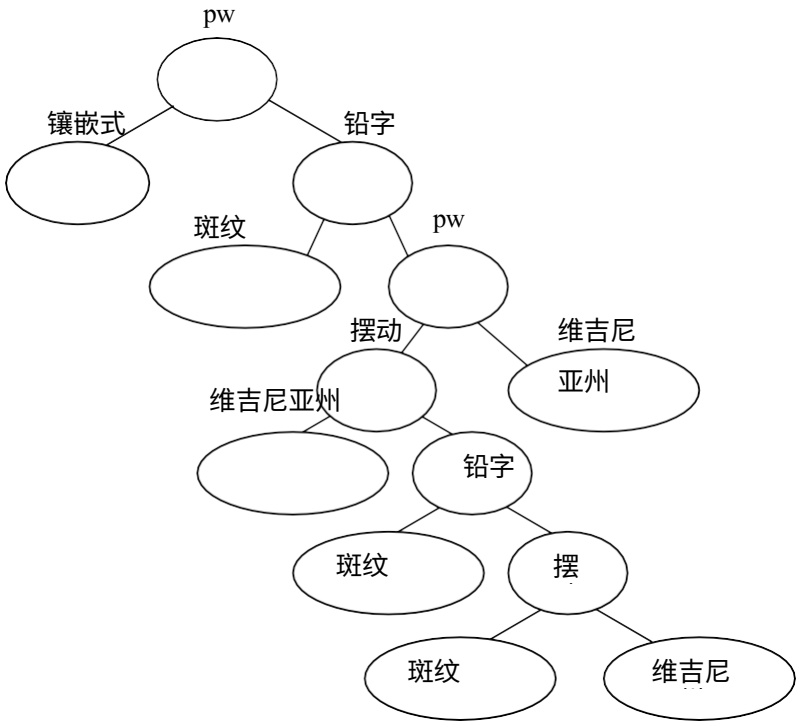


图9.一个未经修剪的决策树T通过ID3进行感应

表三：对UCI的四个数据集的实验结果

数据集	节点(未修剪)	节点(已修剪)	准确度（未修剪的）	准确度（已修剪）
IRIS	13	5	94.2%	94.2%
PIMA	145	55	69.1%	68.6%
玻璃	53	29	59.4%	60.4%
葡萄酒	9	7	91.3%	91.3%