

计算机技术

# 一种改进的 C4.5 算法

李 瑞 程亚楠\*

(大连交通大学软件学院,大连 116028)

**摘 要** 为了提高 C4.5 算法的有效性,提出了一种改进的 MB—C4.5 算法。该算法主要改进了 C4.5 算法的分枝策略和属性选取的标准。把分类效果较差的分枝合并到分类效果较好的分枝中。引进一个平衡度系数,系数大小由决策者依靠先验知识或领域知识确定。MB—C4.5 算法在提高重要属性的选择、减少无意义分枝、过度拟合等方面有一定提高。用该算法构造出的决策树进行分类更为准确、合理。对改进前后的算法用实例进行分析,说明 MB—C4.5 算法的有效性。

**关键词** C4.5 算法 MB—C4.5 算法 合并分枝 平衡度系数

**中图法分类号** TP301.6; **文献标志码** A

决策树方法的起源是概念学习系统 CLS (concept learning system), 然后发展出了多种算法。决策树算法的结果容易理解, 而且其分类模式也容易转化为规则<sup>[1]</sup>。构建决策树的关键环节是属性的选择和样本集的划分。设计产生较少的树结点和深度的决策树算法是构建决策树所面临的一种挑战<sup>[2]</sup>。

决策树算法中以 J. Ross Quinlan 提出的 ID3 系列发展尤为迅速, 应用也较广。Quinlan 于 1979 年提出 ID3 算法, 极大地推动了决策树算法的应用, 但由于 ID3 所固有的一些缺点, Quinlan 于 1993 年在 ID3 算法的基础上又提出了 C4.5 算法, 新算法保持原有算法的优点并改进原有的缺点, 从而成为决策树算法的主流。对 C4.5 算法进行深入学习与分析后, 发现 C4.5 算法在避免过度拟合、准确性等方面并不令人非常满意。因此, 提出一种决策树的改进算法: MB\_C4.5 (Merge & Balance C4.5) 算法。该算法基于 C4.5 算法, 但在分枝过程中, 把信息熵值较高的部分分枝分别合并到信息熵值较低的部分分

枝中, 即通过合并对分类贡献较小的分枝, 避免了碎片问题; 另外, C4.5 算法在构造树的内部节点时是局部最优的搜索方式, 它所得到的结果尽管有很高的准确性, 但仍然达不到全局最优的结果。因此引入平衡度系数对算法进行改进, 使建立的决策树具有更高的准确性<sup>[3]</sup>。通过对 MB\_C4.5 算法的分析和比较, 说明了改进算法的有效性。

## 1 改进的 MBC4.5 算法

### 1.1 C4.5 算法

在决策树算法中, ID3 算法是著名的算法。C4.5 算法是对 ID3 算法的改进, 主要克服了 ID3 算法选择偏向于取值较多的属性等的不足之处。

C4.5 算法主要思想为<sup>[3]</sup>:

设  $T$  为数据集, 类别集合为  $\{C_1, C_2, \dots, C_k\}$ , 选择一个属性  $V$  把  $T$  分为多个子集。  $V$  有互不重合的  $n$  个取值  $\{v_1, v_2, \dots, v_n\}$ , 则  $T$  被分为  $n$  个子集  $T_1, T_2, \dots, T_n$ , 其中  $T_i$  中所有实例的取值均为  $v_i$ 。令  $|T|$  为数据集  $T$  的例子数,  $|T_i|$  为  $V = v_i$  的例子数,  $|C_j| = \text{freq}(C_j, T)$  为  $C_j$  的例子数,  $|C_{jv}|$  是  $V = v_i$  例子中具有类别  $C_j$  的例子数。则有:

$$(1) \text{ 类别 } C_j \text{ 的发生概率为: } P(C_j) = \frac{|C_j|}{|T|} =$$

2010年6月30日收到 辽宁省自然科学基金(20072161)资助  
第一作者简介:李 瑞(1963—),吉林四平人,副教授,研究生导师,研究方向:数据挖掘、决策支持系统。

\* 通信作者简介:程亚楠, E-mail: cyn\_413@163.com。

$freq(C_j, T)$ 。

(2) 属性  $V=v_i$  的发生概率为:  $P(v_i) = \frac{|T_i|}{|T|}$ 。

(3) 属性  $V=v_i$  的例子中,具有类别  $C_j$  的条件概率为:  $P(C_j|v_i) = \frac{|C_{ji}|}{|T_i|}$ 。

(4) 类别信息熵计算:

$$H(C) = - \sum_j P(C_j) \lg P(C_j) = - \sum_{j=1}^k \frac{freq(C_j, T)}{|T|} \lg \frac{freq(C_j, T)}{|T|} = \text{Info}(T) \quad (1)$$

(5) 类别条件熵:

$$H\left(\frac{C}{V}\right) = - \sum_j P(v_j) \sum_i P\left(\frac{C_j}{v_i}\right) \lg P\left(\frac{C_j}{v_i}\right) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \text{Info}(T_i) = \text{Info}_v(T) \quad (2)$$

(6) 信息增益:

$$I(C, V) = H(C) - H\left(\frac{C}{V}\right) = \text{Info}(T) - \text{Info}_v(T) = \text{gain}(v) \quad (3)$$

(7) 属性  $V$  的信息熵:

$$H(V) = - \sum_i P(v_i) \lg P(v_i) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \lg \frac{|T_i|}{|T|} = \text{split\_Info}(v) \quad (4)$$

(8) 信息增益率:

$$\text{gain\_ratio}(v) = \frac{I(C, V)}{H(V)} = \frac{\text{gain}(v)}{\text{split\_Info}(v)} \quad (5)$$

C4.5 算法虽然得到广泛的应用,但也存在固有的一些不足之处。

第一, C4.5 算法使用信息增益比来选择属性,该选取标准解决了偏向取值较多的属性的问题,但使属性选择度量的信息理论含义模糊且可解释性变差。另外,这种选择标准偏向于选择对统一属性值取值比较集中的属性(即熵值较小的属性),但不一定是对分类有重大作用的属性<sup>[4]</sup>。

第二,在 C4.5 算法中,对于分类型属性,每个结点产生的分枝个数为属性已知值的数目。但其中存在较多的空枝(即分枝的结点中样本数为 0 或接近于 0 的分枝),或部分分枝对分类并没有任何

贡献。这些分枝的存在,增大了决策树的大小,降低了决策树的可用性,也导致了过度拟合问题<sup>[5]</sup>。

第三,在分枝过程中,可能会有大量的碎片(即一些结点的样本数太少,缺乏统计意义)产生而导致决策树过度拟合。结点的样本数太少的原因是:这些样本本身就是异常数据或噪音数据;或是由不合理的分枝标准导致。

第四, C4.5 算法采用分而治之的策略,在构造树的内部节点时是局部最优的搜索方式,所以它所得到的结果尽管有很高的准确性,但仍然达不到全局最优的结果<sup>[3]</sup>。

第五, C4.5 算法评价决策最主要的依据是决策树的错误率,而对树的深度、节点的个数等不进行考虑,而树平均深度直接对应着决策树的预测速度,树的节点个数则代表树的规模。

## 1.2 MB—C4.5 算法

对属性的选择是建立决策树模型的关键技术之一。为了解决 C4.5 算法存在的某些不足之处,在 C4.5 算法的基础上提出了改进的 MB—C4.5 算法。

MB—C4.5 算法主要思想为:

(1) 对于数据集  $T$  中的每个分类型属性(因为连续型属性的分支为 2,所以不再考虑合并连续型属性的分支),一个属性  $V$  中的每个样本子集  $V_1, V_2, \dots, V_n$  对应该属性的已知值  $v_1, v_2, \dots, v_n$ , 计算该属性所有样本子集的熵。

(2) 对该属性所有样本子集的熵计算出它们的平均值,然后将样本子集的熵的值不小于平均值的样本子集挑选出来,按这些样本子集的熵值的大小进行降序排列并且得到按降序排列的样本子集的个数为  $m$ ;接着将样本子集的熵的值小于平均值的样本子集挑选出来按照熵值大小进行升序排列,得到前  $m$  个熵值对应的样本子集;再使降序排列的样本子集分别与升序排列的样本子集一一对应的合并,形成临时的复合样本子集;最后计算复合样本子集的熵值。

(3) 根据该属性的复合样本子集的熵值和未合并样本子集的熵值计算该属性的修正信息增益。

(4) 对公式(2)和公式(4)中的加权和引入一个平衡度系数(平衡度系数  $\lambda$  ( $0 < \lambda < 1$ ) 是一个模糊的概念,其大小由决策者根据先验知识或领域知识来确定<sup>[6]</sup>。),降低了某些属性的信息熵,相应地提高了其他属性的信息熵。

如果指定某一属性的平衡度系数为  $\lambda$ ,引入平衡度系数后公式(2)、公式(4)、公式(5)分别变形为公式(6)、公式(7)和公式(8)所示。

$$H\left(\frac{C}{V_\lambda}\right) = - \sum_j (P(v_j) + \lambda) \sum_i P\left(\frac{C_j}{v_i}\right) \lg P\left(\frac{C_j}{v_i}\right) = \text{Info}_v(T_\lambda) \quad (6)$$

$$H(V_\lambda) = - \sum_i (P(v_i) + \lambda) \lg P(v_i) = \text{split\_Info}(v_\lambda) \quad (7)$$

$$\text{gain\_ratio}(v_\lambda) = \frac{I(C, V_\lambda)}{H(V_\lambda)} = \frac{\text{gain}(v_\lambda)}{\text{split\_Info}(v_\lambda)} \quad (8)$$

(5) 选择最高修正信息增益的属性为当前结点的测试属性,该属性的分枝对应于未合并样本子集和复合样本子集。

(6) 决策树构造的其余部分与 C4.5 算法相同。

MB—C4.5 算法对具有较高熵值的分枝分别进行合并,即将这些对划分无贡献的分枝分别进行合并,有效控制了碎片问题,减少了无意义的分枝和空枝,限制了过度拟合问题的影响。设定平衡度系数后,构造决策树并进行规则提取,具有更高的准确率。

## 2 C4.5 与 MB—C4.5 的比较

下面表 1 给出了根据天气情况决定是否适合做运动的几个相关指标的数据集合<sup>[3]</sup>,共有 4 个属性: outlook、temperature、humidity 和 windy。这 4 个属性被分为 yes 和 no 两类。

以表 1 样本数据集为训练集,采用 C4.5 算法构造决策树对训练数据进行分类,可得到决策树如图 1 所示。

用改进后的 MB—C4.5 算法对表 1 的样本数据重新生成决策树。指定 outlook 属性的平衡度系数  $\lambda = 0.3$ ,其它属性的平衡度系数设为 0。由改进后

的算法重新构造的决策树如图 2 所示。

表 1 某地天气样本数据集

Attribute	outlook	temperature	humidity	windy	yes/no
1	sunny	hot	normal	false	no
2	sunny	hot	normal	true	no
3	sunny	hot	high	true	no
4	overcast	mild	high	false	yes
5	overcast	hot	high	true	yes
6	rain	cool	high	false	no
7	rain	cool	normal	true	no
8	rain	hot	low	false	yes
9	rain	hot	low	true	no
10	sunny	cool	normal	false	no
11	sunny	cool	normal	true	no
12	rain	cool	low	false	no
13	rain	mild	low	true	no
14	sunny	cool	low	true	yes
15	sunny	cool	low	true	yes
16	overcast	cool	normal	true	yes
17	overcast	cool	normal	true	yes
18	overcast	mild	low	false	yes
19	rain	cool	high	true	no
20	overcast	mild	normal	true	yes

比较图 1 和图 2,可以发现 MB - C4.5 算法生成的决策树的第一层中的分枝 humidity = high 和 humidity = low 发生了合并,并且下面几层中的部分分枝也发生了合并。合并本身不会直接对分类有贡献,但合并可以避免过多无意义的分枝,减少产生碎片的可能。通过比较图 1 和图 2 可以得到表 2,采用 MB - C4.5 算法,树的叶子结点个数得到减少,部分叶子结点的样本数也得到增加。另外,根据表 1,分析图 2 可以看出 outlook 属性离根结点距离变远, humidity、temperature 属性离根节点距离缩短。即 MB - C4.5 算法降低了 outlook 属性在分类中的重要性,同时提高了 humidity、temperature 属性在分类中的重要性。改进算法令分类结果更准确合理一些,便于决策者做出正确的决策。但因样本数据集太小的原因,生成的规则中会有少数与实际

情况不符,如果增大训练集并对数据进行预处理消除噪音干扰等则效果更好。

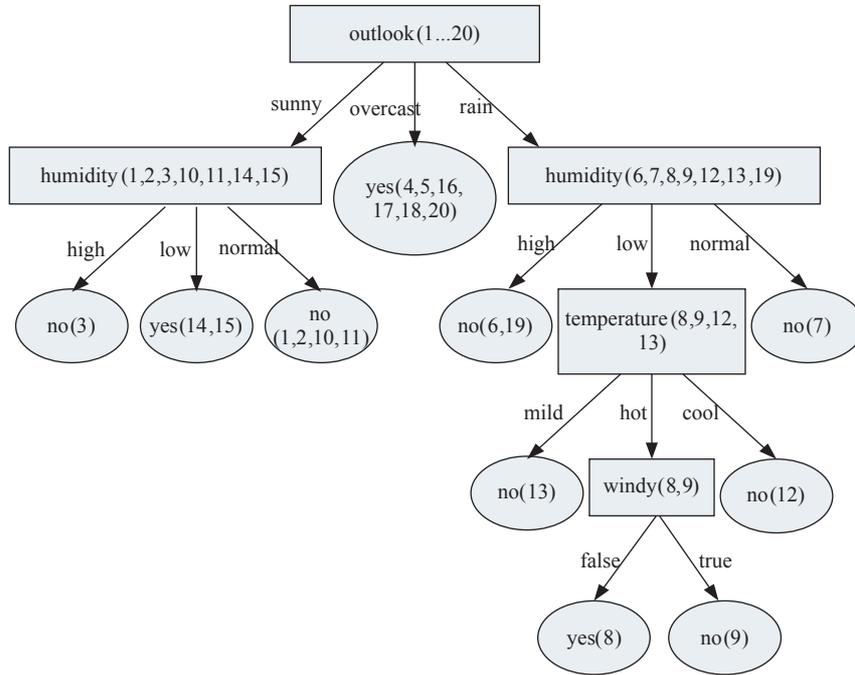


图1 C4.5 算法生成的决策树

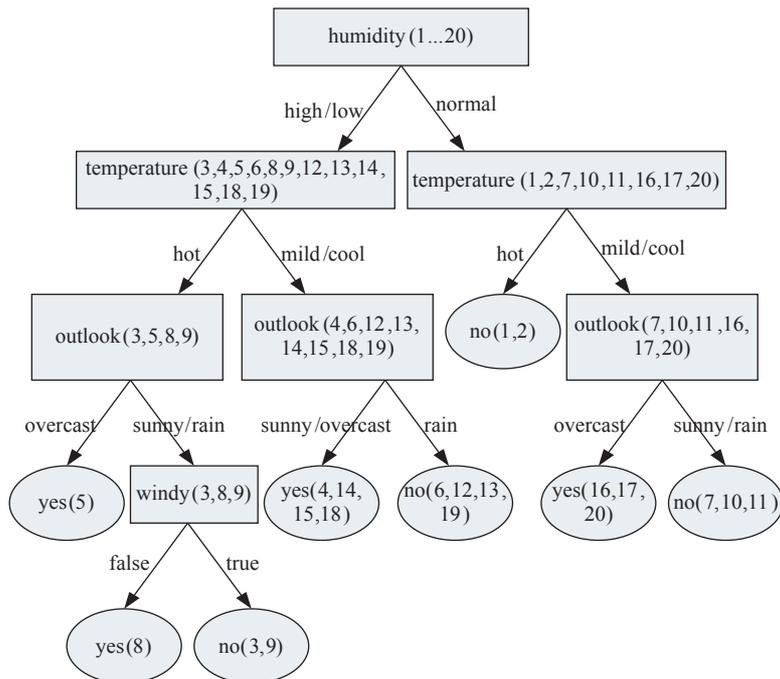


图2 改进算法生成的决策树

表 2 C4.5 和 MB-C4.5 算法的决策树模型的比较

模型	树的层次	叶子结 点数	叶子结点样本 比例平均值
C4.5 算法决策树模型	5	10	$(20/10)/20 \times 100\%$ = 10%
MB-C4.5 算法 决策树模型	5	8	$(20/8)/20 \times 100\%$ = 12.5%

### 3 结束语

C4.5 算法是一种经典的决策树算法,改进 C4.5 算法有很大的价值。在传统 C4.5 算法的基础上,本文提出了决策树模型的改进算法——MB-C4.5 算法,通过实验验证了改进算法的有效性、合理性等。MB-C4.5 算法没有对连续型属性进行改进,也没有涉及缺失数据的数据集。MB-C4.5 算法还有可以继续改进的地方,使改进的算法更加合理

有效。

在实际领域中,改进决策树分类算法有很广泛的应用;在数据挖掘技术的研究和发展中,改进决策树分类算法起到了一定的推动作用。

### 参 考 文 献

- 1 Quinlan J R. C 4. 5: program for machine learning. San Mateo : Morgan Kaufmann Publisher - s, 1993;21—31
- 2 Kothari R, Dong M . Decision Trees for Classification; A review and some new results. In: Pal R S, Pal N R, Eds. Lecture Notes in Pattern Recognition, Singapore, World Scientific Publishing Company, 2001
- 3 李 瑞,魏现梅,黄 明,等. 一种改进的决策树学习算法. 科学技术与工程,2009;9(20):6038—6041
- 4 史忠植. 知识发现. 北京:清华大学出版社,2002
- 5 刘 鹏. 一种健壮有效的决策树改进模型. 计算机工程与应用, 2005;(33):172—175
- 6 曲开社,成文丽,王俊红. ID3 算法的一种改进算法. 计算机工程与应用,2003;(25):104—107

## An Improved C4.5 Algorithm

LI Rui, CHENG Ya-nan\*

(School of Software of Dalian Jiaotong University, Dalian 116028, P. R. China)

[Abstract] To improve the effectiveness of C4.5 algorithm, an improved MB-C4.5 algorithm is introduced. The algorithm is mainly improved in the criterion of partitioning rules and attribution selection of the C4.5 algorithm; the branches which have poor appearances in classification are united into the ones which have good appearances in classification. A balanced coefficient is introduced and it can be fixed by decision maker according to priori intellectual and domain intellectual. MB-C4.5 enhances importance of test attribute selection, reduces the number of insignificant branches and avoid the appearance of over fitting. The classification is more veracious and rational by the decision tree made from the improved algorithm. And compared the improved algorithm to C4.5 algorithm by analyzing examples, to prove the efficiency of the improved algorithm.

[Key words] C4.5 algorithm MB-C4.5 algorithm combined branches balance coefficient