Computer Science and Application 计算机科学与应用, 2015, 5, 171-178
Published Online May 2015 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2015.55022

Hans汉斯

C4.5 of Decision Tree Algorithm Optimization of Property Values

Shifan Huang*, Yong Shen, Ruifang Wang, Huali Ma, Changgeng Chen, Yuhao Zhang

School of Software, Yunnan University, Kunming Yunnan
Email: *974794674@qq.com

Received: May 7th, 2015; accepted: May 23rd, 2015; published: May 28th, 2015

Abstract

About the decision tree algorithm, the quantity of the attribute value types determines the quantity of the decision tree branch. Based on this, we put forward a new method which can optimize attribute value. The examples show that the method can optimize the quantity of the decision tree branch, and reach the purpose that simplifies the decision tree structure. This method has no effect on the classification accuracy of the C4.5 algorithm.

Keywords

Decision Tree, C4.5 Algorithm, Property Values, Optimization

决策树C4.5算法属性取值优化研究

黄世反*，沈　勇，王瑞芳，马华丽，陈长赓，张宇昊

云南大学软件学院，云南　昆明
Email: *974794674@qq.com

收稿日期：2015年5月7日；录用日期：2015年5月23日；发布日期：2015年5月28日

摘　要

在决策树算法中，属性取值种类的多少决定着决策树分支数量的多少。基于此，提出了一种新的属性取

*通讯作者。

值优化的方法，实例证明该方法确实能优化生成决策树的分支数量，达到精简生成决策树结构的目的，且该方法对原C4.5算法的分类正确率没有影响。

关键词

决策树，C4.5算法，属性取值，优化

1. 引言

决策树算法是一种最简单、最直接、最有效的文本分类算法。最早的决策树算法是 ID3 算法[1]，于 1986 年由 Quinlan 提出，该算法是一种基于信息熵的决策树分类算法。由于该算法是以信息熵作为属性选择的标准，偏向于选择属性取值较多的属性，而属性取值较多的属性往往分类的贡献不大。因此，于 1993 年 Quinlan 在 ID3 算法的基础上又提出了一种改进算法，即 C4.5 算法[2]。该算法采用信息增益率作为属性选择的标准，继承了 ID3 算法的所有优点，克服了 ID3 算法中偏向于选择属性取值较多的属性作为测试属性的不足，同时还能对连续属性与未知属性进行处理，在剪枝方面也有很大的改进。

C4.5 算法作为经典的决策树分类算法，已被广泛的应用到各个领域。但其仍然存在以下不足之处：1) 在计算信息增益的过程中(包括：分类所需信息量、信息熵、分割信息量)涉及的复杂的对数运算，计算机每一次计算都需要调用库函数，增大了生成决策树所需的时间开销；2) 生成决策树中分支数量过多，部分分支还能进行合并，进一步精简生成决策树的结构。

针对以上不足，已有很多学者提出了改进优化算法。文献[3]，作者通过引入高等数学中等价无穷的原理，对 C4.5 算法中的计算公式进行了改进，用简单的四则混合运算取代了复杂的对数运算，减少了生成决策树所需的时间开销。文献[4]，作者提出了一种对属性取值进行优化合并的方法，该方法通过将属性的不同取值分成多个样本子集，并计算各个样本子集的熵以及样本子集的平均熵值，并将熵值大于平均熵值的样本子集进行合并，并重新定义一个属性取值替代原数据表中的属性取值，实例证明该方法确实能起到精简生成决策树结构的作用。

本文针对生成决策树分支数量过多的不足，提出了一种新的属性取值优化方法，并用实例分析验证了该方法的有效性。

2. C4.5 算法介绍

C4.5 算法是一种基于信息熵的决策分类算法，该算法的核心思想是根据信息熵原理，选择样本集中信息增益率最大的属性作为分类属性，并根据该属性的不同取值构造决策树的分支，再对子集进行递归构造，直到完成决策树的构造[5]。

假设样本空间 D 中有正例集 P 个、反例集 N 个，且 $D = P + N$，则一棵决策树能对待分类数据集做出正确类别判断所需的信息量为：

$$I(P,N) = -\frac{P}{D}\log_2\frac{P}{D} - \frac{N}{D}\log_2\frac{N}{D} \tag{1}$$

如果以属性 A 作为决策树的根节点，且 A 具有 V 个值 $(V_1,V_2,V_3,\cdots,V_v)$，它将 H 分为 V 个样本子集 $(H_1,H_2,H_3,\cdots,H_v)$，设 H_i 中共有 d 个数据集，其中有 p 个正例和 n 个反例，则子集 H_i 的信息熵 $E(H_i)$ 可由下式计算：

$$E(H_i) = -\frac{p}{d}\log_2\frac{p}{d} - \frac{n}{d}\log_2\frac{n}{d} \tag{2}$$

以属性 A 为根节点分类的信息熵为：

$$E(A)=\sum_{i=1}^{v}\frac{d}{D}E(H_i) \tag{3}$$

则，以属性 A 为根节点的信息增益为：

$$G(A)=I(p,n)-E(A) \tag{4}$$

在 C4.5 算法中，是以信息增益率来选择分类属性的，而信息增益率等于信息增益与分割信息量的比值，其中分割信息量可由下式计算：

$$\text{SplitI}(A)=-\sum_{i=1}^{m}p_i\log_i p_i \tag{5}$$

其中，p_i 表示属性 A 的所有取值中，任意样本子集占总样本集的比例。则，信息增益率可由下式计算：

$$\text{G-R}(A)=\frac{G(A)}{\text{SplitI}(A)} \tag{6}$$

3. C4.5 算法中属性取值的优化改进

3.1. 算法改进的基本思想

如何很好的选择测试属性决定着所构造决策树模型的质量，而所选测试属性的可能取值的数量与树的分支数量成正比。在 C4.5 算法中，一切计算的依据来源于信息论中的熵。因此，本文在信息熵的计算过程中，通过对测试属性的优化改进，以减少决策树的分支数量，提高决策树的质量。

根据信息熵的定义，熵值越大，对分类决策的贡献就越小。反之，熵值越小，对分类决策的贡献就越大，即能确定更多待分类对象所属类别。当信息熵为零时，对分类决策的贡献最大。当信息熵为零时，只存在两种情况，所有样本子集同属于正例集或者反例集(本文只研究只有两种决策属性的情况)。基于此，本节算法改进的基本思想为：对于给定的一组数据集，计算每个属性的信息增益率，选取信息增益最大的属性，将该属性按照属性取值的不同划分为多个样本子集，计算每一个样本子集的信息熵，取出信息熵为零的所有样本子集，并将其同属正例集或反例集的样本子集进行合并，定义新的属性取值，得到一个新的、复合的样本子集，并用该复合样本子集在测试集中替换构成该复合子集的样本子集。最后，再利用新的数据集来建立决策树。

3.2. 改进后算法的计算步骤

由上一节的分析，我们可以得到改进后算法的基本计算步骤，如下所示：

1) 根据给定的测试集，计算各个属性的信息增益率；

2) 选择信息增益率最大的属性，将该属性按照属性取值的不同划分为多个样本子集；

3) 计算各样本子集的信息熵；

4) 选取出所有信息熵为零的样本子集，并将其中同属于正例集或反例集的样本子集进行合并，自定义新的属性取值，得到一个新的、复合的样本子集；

5) 用新形成的复合样本子集取代原数据集中组成该复合样本子集的各个样本子集；

6) 利用新形成的数据集，计算信息增益率，其余部分与 C4.5 算法相同。

4. 实例分析

表 1 是对学生某次考试成绩的描述，已经过离散化处理，并按照分数段将各科目分为优、良、中、

差，总评分为两类：合格与不合格，如表 1 所示。

4.1. 采用第 3 节中的改进算法对表 1 进行优化及决策树的生成过程

本文中对分类所需信息量、信息熵、信息增益、分割信息的计算采用文献[6]中改进的 C4.5 算法计算公式进行计算。与文献[3]类似，文献[6]也是通过引入高等数学中等价无穷小的理论，对 C4.5 算法计算公式中繁多的对数运算进行了优化。不同的是，二者优化后的计算公式略有不同，文献[6]优化后得到的计算公式更加简洁、直观，方便计算。因此，本文选用文献[6]中的优化结果来进行相关计算。

1) 计算各属性的信息增益率

$$I(D)=\frac{17\times 3}{20^2}=0.13$$

$$\text{Gain-Ratio}(\text{语文})=-2.21 \qquad \text{Gain-Ratio}(\textbf{数学})=-2.64$$

$$\text{Gain-Ratio}(\text{英语})=-3 \qquad \text{Gain-Ratio}(\text{文综})=-1.06$$

2) 选取信息增益率最大的属性划分样本子集

由表 1 可得，我们可以将属性文综划分为：$\text{综}_{\text{优}}$、$\text{综}_{\text{良}}$、$\text{综}_{\text{中}}$、$\text{综}_{\text{差}}$四个样本子集。

3) 计算各样本子集的信息熵

属性文综各样本子集信息熵如下：

$$E(\text{综}_{\text{优}})=\frac{3\times 0}{3}=0;\ E(\text{综}_{\text{良}})=\frac{7\times 0}{7}=0;\ E(\text{综}_{\text{中}})=\frac{7\times 1}{8}=0.88;\ E(\text{综}_{\text{差}})=\frac{0\times 2}{2}=0;$$

4) 合并信息熵为零且同属正例集或反例集的样本子集

在属性文综的各样本子集中，信息熵为零的有三个子集：$\text{综}_{\text{优}}$、$\text{综}_{\text{良}}$及$\text{综}_{\text{差}}$，其中$\text{综}_{\text{优}}$和$\text{综}_{\text{良}}$同属合格，而$\text{综}_{\text{差}}$属于不合格，因此可以将$\text{综}_{\text{优}}$、$\text{综}_{\text{良}}$进行合并，并定义新的属性取值为：优良。

5) 合并后的新数据集如表 2 所示。

经属性优化合并后，属性文综的属性取值由原来的四个变为现在的三个，这样就能减少生成决策树的分支数量。

6) 决策树的生成

a) 以属性文综为根节点建立决策树如图 1 所示。

b) 将剩余属性计算其信息增益率如下：

$$I(D)=\frac{7\times 1}{8^2}=0.11$$

$$\text{Gain-Ratio}(\text{语文})=-0.62;\ \text{Gain-Ratio}(\textbf{数学})=-1.08;\ \text{Gain-Ratio}(\text{英语})=-0.59$$

由计算结果可得，属性英语的信息增益率最大，计算属性英语各个样本子集的信息熵，合并属性取值后得到表 3。

选取属性英语作为第二层分类属性，生成的决策树如图 2 所示。

c) 将剩余属性调用 1 到 5 得到表 4

到此，在剩余属性中，随便选择哪个都能确定分类。最终的决策树如图 3 所示。

4.2. 利用文献[6]中改进的 C4.5 算法及表 1 生成决策树

由表 1 可知，决策属性为总评，共有 20 个属性值，其中正例值有 17 个，反例值有 3 个，分类所需信息量为：

Table 1. Instantiation data table
表 1. 实例数据表

序号	语文	数学	英语	文综	总评
1	中	良	差	中	合格
2	中	良	良	良	合格
3	良	中	中	良	合格
4	优	中	良	良	合格
5	中	中	中	中	合格
6	良	差	中	差	不合格
7	优	差	差	中	不合格
8	良	优	优	优	合格
9	中	中	优	良	合格
10	优	中	中	中	合格
11	差	良	良	差	不合格
12	良	中	良	良	合格
13	中	差	良	中	合格
14	中	中	优	良	合格
15	良	差	良	中	合格
16	优	中	优	良	合格
17	优	优	优	优	合格
18	良	差	良	中	合格
19	良	中	差	优	合格
20	中	中	中	中	合格

Table 2. After the first method processing and merging data table
表 2. 经方法一处理第一次合并后的数据表

序号	语文	数学	英语	文综	总评
1	中	良	差	中	合格
2	中	良	良	优良	合格
3	良	中	中	优良	合格
4	优	中	良	优良	合格
5	中	中	中	中	合格
6	良	差	中	差	不合格
7	优	差	差	中	不合格
8	良	优	优	优良	合格
9	中	中	优	优良	合格
10	优	中	中	中	合格
11	差	良	良	差	不合格
12	良	中	良	优良	合格
13	中	差	良	中	合格
14	中	中	优	优良	合格
15	良	差	良	中	合格
16	优	中	优	优良	合格
17	优	优	优	优良	合格
18	良	差	良	中	合格
19	良	中	差	优良	合格
20	中	中	中	中	合格

Table 3. After the first method processing and second merging data table
表 3. 经方法一处理第二次合并后的数据表

序号	语文	数学	英语	总评
1	中	良	差	合格
5	中	中	中良	合格
7	优	差	差	不合格
10	优	中	中良	合格
13	中	差	中良	合格
15	良	差	中良	合格
18	良	差	中良	合格
20	中	中	中良	合格

Table 4. After the first method processing and third merging data table
表 4. 经方法一处理第三次合并后的数据表

序号	语文	数学	总评
1	中	良	合格
7	优	差	不合格

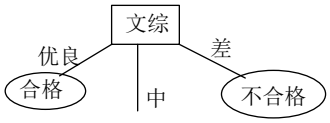

Figure 1. The first step in generating the decision tree
图 1. 生成决策树第一步

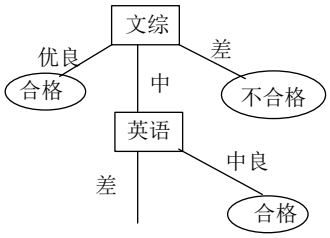

Figure 2. The second step in generating the decision tree
图 2. 生成决策树第二步

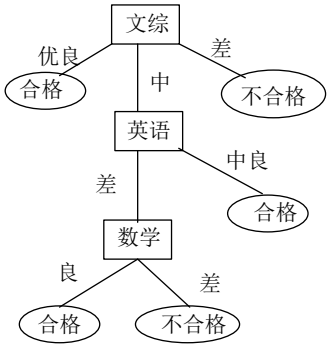

Figure 3. By the improved algorithm generates decision tree figure
图 3. 由本文改进算法最终生成的决策树图

$$I(D)=0.13$$

各个属性的信息增益率如下(对信息熵、分裂信息量的计算此处省略，直接给出信息增益率的值)：

$$\text{Gain-Ratio}(\text{语文})=-2.21 \qquad \text{Gain-Ratio}(\textbf{数学})=-2.64$$

$$\text{Gain-Ratio}(\text{英语})=-3 \qquad \text{Gain-Ratio}(\text{文综})=-1.06$$

从以上计算结果可看出，文综的信息增益率最大，因此，将属性文综作为决策树的根节点，如图 4 所示。

剩余属性的信息增益率如下：

$$I(D)=0.11$$

$$\text{Gain-Ratio}(\text{语文})=-0.62;\ \text{Gain-Ratio}(\textbf{数学})=-1.08;\ \text{Gain-Ratio}(\text{英语})=-0.59$$

由计算结果可知，属性英语的信息增益率最大，选取属性英语作为下一层分类属性，则可得到决策树生成第二步图如图 5 所示。

到此，在剩余属性中，随便选择哪个都能确定分类。最终的决策树如图 6 所示。

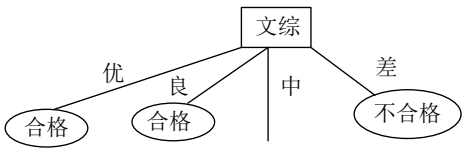

Figure 4. The first step by not optimized attribute ingesting data to generate decision tree
图 4. 由未优化属性取值数据生成决策树第一步

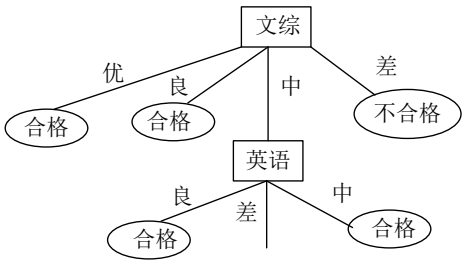

Figure 5. The second step by not optimized attribute ingesting data to generate decision tree
图 5. 由未优化属性取值数据生成决策树第二步

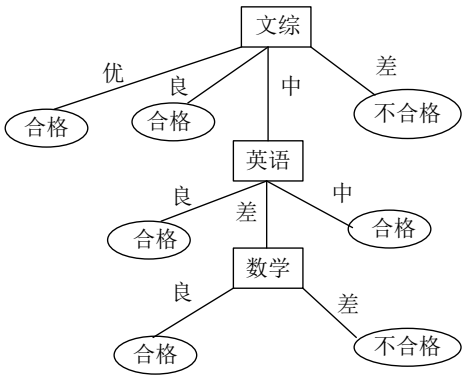

Figure 6. By not optimized attribute ingesting data to generate decision tree
图 6. 由未优化属性取值数据生成决策树

由图 3、图 4 可知，本文改进算法明显起到了减少生成决策树分支数量的作用，精简了生成决策树的结构。同时，因为文献[6]中的改进算法并没有触及到 C4.5 算法的核心步骤，只是对计算公式进行了化简，因此，不会影响的生成决策树的分类正确率。可见，本文所提出的改进算法与文献[6]中的改进算法巧妙合并是非常有效的。

5. 结束语

本文提出了一种新的属性取值优化算法。由实验结果可得：该优化算法明显减少了生成决策树的分支数量，精简了生成决策树的结构，值得进一步推广该算法的应用领域。因，本文的研究只针对只有两类的情况，当出现多类的情况时不再适用。因此，下一步将主要研究多类情况下的属性取值优化算法。

基金项目

南省软件工程重点实验室面上基金项目(2012SE306；2011SE12)。

参考文献 (References)

[1] Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning*, **1**, 81-106.

[2] Quinlan, J.R. (1993) C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo.

[3] 黄爱辉 (2009) 决策树 C4.5 算法的改进及应用. *科学技术与工程*, **1**, 34-36.

[4] 刘鹏, 姚正, 尹俊杰 (2006) 一种有效的 C4.5 改进模型. *清华大学学报：自然科学版*, **46**, 996-1001.

[5] 李强 (2006) 创建决策树算法的比较研究-ID3, C4.5, C5.0 算法的比较. *甘肃科学学报*, **12**, 84-87.

[6] 周琦 (2012) 改进的 C4.5 决策树算法研究及在高考成绩预测分析中的应用. 广西大学, 南宁.