软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(11):2814−2824 [doi: 10.13328/j.cnki.jos.005344]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

不一致数据上精确决策树生成算法*

王鹤澎, 王宏志, 李建中, 高 宏

(哈尔滨工业大学 计算机科学与技术系,黑龙江 哈尔滨 150006)

通讯作者: 王宏志, E-mail: wangzh@hit.edu.cn

摘 要: 近年来,随着现实生活中数据量的不断增大,不一致数据的出现也越发频繁,这使得人工修正不一致数据变得更加耗时.而且,人工修正数据方法本身也存在着不可避免的人为操作错误,因此,这种修正方法不再可行.如何不提前修复不一致数据,直接在不一致数据上进行分类,是该文的核心研究内容.对决策树生成算法的目标函数进行改进,使其能够直接对不一致数据进行分类,并得到较好的分类结果.对约束条件中的特征对分类结果的影响进行了多方面衡量,从而调整该特征的影响因子,使得决策树的节点分割更加精确,分类效果更优.

关键词: 不一致数据;决策树;分类;海量数据

中图法分类号: TP181

中文引用格式: 王鹤澎,王宏志,李建中,高宏.不一致数据上精确决策树生成算法.软件学报,2017,28(11):2814−2824. http://www.jos.org.cn/1000-9825/5344.htm

英文引用格式: Wang HP, Wang HZ, Li JZ, Gao H. Algorithms for accurate decision tree generation on inconsistent data. Ruan Jian Xue Bao/Journal of Software, 2017,28(11):2814−2824 (in Chinese). http://www.jos.org.cn/1000-9825/5344.htm

Algorithms for Accurate Decision Tree Generation on Inconsistent Data

WANG He-Peng, WANG Hong-Zhi, LI Jian-Zhong, GAO Hong

(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150006, China)

Abstract: In recent years, with the increasing amount of data in real life, inconsistent data becomes more frequent. This makes manual correction of inconsistent data more time-consuming. Moreover, manual correction prone to human errors. Thus, such correction method is no longer feasible. How to perform classification directly on inconsistent data without correcting data beforehand is the core research content of this paper. In this paper, the objective function of the decision tree generation algorithm is improved so that it can directly classify inconsistent data and achieve better results. Multidimensional measures of the influence of the feature are used on classification results to adjust the influence factor of the feature so that nodes of the decision tree can be split more accurate to achieve more effective classification results.

Key words: inconsistent data; decision tree; classification; massive data

近年来,随着数据量的不断增大,在存储、合并数据时难免会导致数据质量问题,例如缺失数据、不一致数据以及冗余数据等.随着数据的日益更迭,数据的大量更新也会导致大量的过时数据迅速产生,并且伴随着数据来源与形式上的多样化,不一致数据也变得更加容易出现.

在数据质量中,不一致数据是一个很重要的维度.由于数据来源的多样性以及实体形式的多样化等[1],不一致数据在实际数据集中普遍存在,并且清洗不一致数据需要大量的人力.

* 基金项目: 国家自然科学基金(U1509216, 61472099); 国家科技支撑计划(2015BAH10F01)

Foundation item: National Natural Science Foundation of China (U1509216, 61472099); National Key Technology R&D Program of China (2015BAH10F01)

本文由复杂环境下的机器学习研究专刊特约编辑胡清华教授推荐.

收稿时间: 2017-04-15; 修改时间: 2017-06-16; 采用时间: 2017-08-23

不一致数据会直接导致分类结果产生偏差,也更容易导致过拟合现象的出现.分类是指通过数据中的某个或某些特征,将相同、相似的数据归为一类.在机器学习中,是指利用训练集中每个类的不同特征值,将不同的类区分出来,利用模型不断学习某个类的特性,使模型最后符合该类特性[2].

当我们在不一致数据上利用传统的机器学习方法进行分类时,由于不一致数据本身包涵着一定的分歧,因此所得到的分类结果也存在着偏差.根据美国医疗委员会的统计,在美国,每年由于数据错误引发的医疗事故会导致高达 98 000 名患者的丧生[3].

为了得到高质量的分类结果,需要有效解决数据中的不一致问题.很自然的方法是对不一致数据进行修复.目前,不一致数据的修复方法有人工修复和自动修复[4,5]两种.对于海量数据,人工修复成本过高;而自动修复方法[4,5]通常复杂度较高,难以适用于海量数据,而且修复结果缺少质量保证.因此,本文拟另辟途径,在不清洗数据的情况下对不一致数据直接进行分类,从而得到高质量的分类结果.

不一致数据的分类带来了一系列的技术挑战.

(1) 由于不一致数据本身就包含着错误的数据,因此在其之上利用原本的机器学习算法进行分类会产生错误的结果.我们面临的第 1 个挑战是:如何改进机器学习算法,使其能够适应当下的场景,得到高准确度的分类结果.

(2) 在分类的约束条件中,不同的特征所具有的意义是不一样的.对于每个约束条件中的特征,其含有的不一致数据所对应的值不同时,所代表的意义也不相同.因此,如何针对不同的特征进行不同的改进,是我们所面临的第 2 个技术挑战.

(3) 在实际场景中,有多种分类约束条件,每种约束条件都是不同的.因此,我们面临的第 3 个挑战是,如何对不同的约束条件改进不同的目标函数.

当前的分类方法并不能有效地解决上述问题,因为当前分类模型是通过数据类别之间的差异性和数据类别之中的相似性来标识类别的.而当数据不一致时,如果不对数据做任何处理或是不对机器学习算法做任何改进来进行模型的学习、分类,不一致数据会使得模型无法正确学习出某类的特性,导致分类的效果下降.本文希望能够找到一种解决方法,改进机器学习模型,使其能够在事先不对不一致数据进行人工修复的情况下得到较好的分类结果.但由于基于海量不一致数据上的分类问题并没有得到充分的重视,该方向的研究论文较少.

基于上述讨论,本文对海量不一致数据上的分类问题进行研究,重点研究适用于海量不一致数据分类的决策树生成算法.由于构造决策树的核心是分割节点的目标函数,因此我们将不一致影响因素融入到目标函数中,最小化不一致数据对决策树精度的影响.本文的贡献可以归纳为如下 4 点.

(1) 本文研究了基于不一致数据的决策树生成问题,这是一种劣质容忍的决策树生成算法.

(2) 为了最小化不一致数据对决策树分类精度的影响,我们提出了融合不一致影响和分割纯度的节点分割目标函数.

(3) 我们针对函数依赖来改进决策树目标函数算法,使其能够最大限度地适用于其他的约束条件,例如函数集合、多属性函数依赖.

(4) 为了验证本文所提出方法的有效性,我们进行了大量的实验,实验结果表明,本文提出的方法能够有效地减小不一致数据对决策树分类结果的影响.

本文第 1 节对研究背景进行介绍.第 2 节详细阐述本文所提出的不一致数据上改进的精确决策树生成算法.第 3 节针对其他约束条件对该算法进行扩展.第 4 节通过大量实验验证所提出算法的有效性.最后给出结论和展望.

1 研究背景

近几年来,随着网络的普及,数据库可以从多个自关联的数据源进行数据集成[6],因而不一致数据的传播变得越发严重[7],在这种数据上的分类问题也变得更加困难.传统的分类问题,针对不一致数据需要预先对数据进行预处理,然后再对处理后的数据进行分类,因而整个分类过程大部分时间花费在数据预处理上.我们希望找到

一种解决方案,能够直接对不一致数据进行分类.本文主要研究海量不一致数据上无数据预处理的分类.

数据一致性是指数据集合中不包含语义错误或相互矛盾的数据[8],不一致数据是指不满足某种约束条件的数据.随着数据质量越来越被人们重视,为了便于描述数据的一致性,越来越多的约束规则被提出来.主要的约束条件有以下几种:否定约束[9]、包含依赖[10]、外键约束、函数依赖[11]、聚集约束[12]、元组生成和等值生成依赖[7].我们给出一个简单的例子来说明:有如下两条数据,我们给定条件函数依赖(*f*[《红楼梦》]→[曹雪芹]).显然,第 1 条数据满足我们所给定的条件函数依赖,而第 2 条数据则不满足我们所给定的条件函数依赖.

- 数据 1:《红楼梦》曹雪芹;
- 数据 2:《红楼梦》吴承恩.

数据依赖定义了关系数据库中不同实体间的内在联系,当数据所表达的语义不符合现实世界所提炼的数据依赖时,数据不一致问题就会产生.根据目前掌握的现实世界的知识,通过定义语义规则的方式来完成不一致数据检测和修复工作的研究已成为数据库领域的热门研究内容,同时也是难点之一.传统的解决数据不一致问题的方法主要包括修复[13−18]、一致性查询应答[13,19−26]以及压缩表示法[26−28]等方法.修复即是寻找另一个一致的数据库实例,且它最低限度地不同于原数据库;一致性查询应答是在不编辑数据的情况下,在原数据库的每一个修复中查找到一个给定查询的答案.压缩表示是找到不一致数据库的所有修复的一种压缩表示.

分类问题已在学术界被研究多年.被应用于预测数据实体间的关系,主要方法有决策树、贝叶斯分类器、KNN(*k*-nearest neighbor))分类器等.决策树是一种通过基于特征值对其进行排序来对实体分类的树形结构[29].决策树中的每一个节点代表实例中的一个特征,每一个分支代表一个可以被假定的值.实例从根节点开始,被排序后的特征值分类.决策树中最主要的技术点是如何划分子树,多年来学者们提出了多种方法来寻找最优特征以便划分出最优子树,例如信息增益[30]、GINI 指数[31]等.由于决策树中的每一个节点对应着相应的实体,最符合该问题的场景,因此,本文选择对决策树生成算法进行改进,使其适应不一致数据分类问题.

2 改进决策树算法

2.1 问题定义

本文默认在数据不一致的情况下,某特征值出现的次数越多,该值正确的可能性越大.为了便于描述算法,我们首先只对函数集合中的 1 条规则进行阐述,对于不一致函数依赖:$A\rightarrow B$,我们称 A 为前置,B 为后置.

由于在函数依赖中,前置与后置是不同的特征,前置决定了后置的特征值,而同一后置值又可以对应多种前置值,因此在决策树进行特征值分割时,需要对前置与后置分别进行考虑.对于前置来说,其对应的后置值越多,分布越平均,越不利于决策树对分类结果的划分.对于后置来说,其所占的比率在其对应的前置值中越少,越不利于决策树对分类结果的划分.因此,对于节点分割,我们考虑两种情况.

- 情况 1. 分割的特征非函数依赖中的特征,或分割的特征是函数依赖中的前置 A,但其对应的后置 B 值是唯一的.对于此种情况,利用正常的 GINI 指数即可完成对特征的分割.
- 情况 2. 分割的特征在函数依赖中有多个不一致值,此时又需要分 3 种情况进行讨论.
  - 情况 2.1. 分割的特征是前置 A,对应的后置 B 未被分割过且具有多个值,需计算分歧率与 GINI 指数.
  - 情况 2.2. 分割的特征是后置 B 且对应的前置 A 当前未被分割过,此时需计算 GINI 指数、占比率与混淆率.
  - 情况 2.3. 不属于以上两种情况的,都正常计算 GINI 指数即可.GINI 指数计算公式如下:

$$GINI_{Data父}=\frac{Data子1}{Data父}GINI_{Data子1}+...+\frac{Data子n}{Data父}GINI_{Data子n} \tag{1}$$

$$GINI_{Data子1}=1-\left[\left(\frac{Data子1中target为正数据量}{Data子1数据量}\right)^2+\left(\frac{Data子1中target为负数据量}{Data子1数据量}\right)^2\right] \tag{2}$$

接下来,我们详细介绍如何解决情况 2.1 和情况 2.2 的问题.

2.2 分割前置特征*A*

当分割前置特征 *A* 时,其对应的后置值越多,分布越平均,越不利于决策树对分类结果的划分.因此,我们通过利用当前前置特征 *A* 的值所对应的每种后置 *B* 值所占的比率来衡量该特征 *A* 对分类结果的影响程度.因而,我们将每种后置 *B* 所占的比率的乘积作为影响因子,某个前置 *A* 所对应的后置 *B* 值分布越平均,所占比率的乘积越大,因而对分类结果的划分影响偏差越大.

分割的特征是前置 *A*,对应的后置 *B* 未被分割过且具有多个值,需计算分歧率与 GINI 指数.该情况特征划分示意图如图 1 所示.

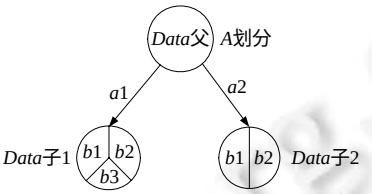

Fig.1 Example of tree model division in Condition 2.1

图 1 情况 2.1 树模型划分情况举例

决策树的目标函数为 GINI 指数+分歧率.其中,分歧率计算公式为

$$分歧率_{Data父}=\frac{Data子1}{Data父}分歧率_{Data子1}+...+\frac{Data子n}{Data父}分歧率_{Data子n} \tag{3}$$

$$分歧率_{Data子2}=\frac{Data子2中不一致数据量}{Data子2的数据量}*\frac{Data子2中b1的数据量}{Data子2的数据量}*\frac{Data子2中b2的数据量}{Data子2的数据量} \tag{4}$$

根据约束 $A\rightarrow B$ 得知:通过 $a1$ 值分割的子节点 $b1$,$b2$ 以及 $b3$ 中存在不一致数据,这 3 个值的可能性与数据量成正比,因而该子节点的数据划分纯度与值的分歧率正相关.从公式(4)可知,分歧率取决于同一前置值所对应的不同后置的占比乘积.我们可以根据公式推导出:该子节点中后置特征 *B* 的每个值分布越平均,所得到的分歧率越大.利用 *Data* 子 1 中不一致数据量除以 *Data* 子 1 的数据量,能够得到该分割子节点中不一致数据量所占的比重,比重越大,该分割子节点因为某种因素发生改变的数据越多.当分割一个节点时,我们希望子节点中的值更趋向于某一个常数,而不是更加混乱的一个分布.

当分歧率较大时,有两种可能性:第 1 种是后置特征 *B* 由于一些因素发生了改变,第 2 种是该前置特征 *A* 由于一些因素发生了改变.无论是哪一种情况,都降低了数据质量,从而导致该数据在此特征分割下有一定的概率被错误分割.因此分歧率越大,该特征越不可取.在分歧率计算时,我们不排除所占比重最多的数据量,因为我们能够通过所占比重最多的数据量与其他数据量的乘积,得到整体的分布.

2.3 分割后置特征*B*

当分割后置特征 *B* 时,由于每个后置特征 *B* 可以对应多个前置特征 *A*,因此前置特征 *A* 的改进方法不再适用于后置特征 *B*.当某个后置特征 *B* 对应多个前置特征 *A* 值时,我们应当考虑该后置特征 *B* 是否是前置特征 *A* 所对应的后置中所占比率最多的.否则,还应考虑当前的后置特征 *B* 值在每个前置特征 *A* 中占多大比率.

情况 2.2 中分割的特征是后置特征 *B*,且其对应的前置 *A* 当前未被分割过,此时需计算 GINI 指数、占比率与混淆率.该情况特征划分示意图如图 2 所示.

决策树的目标函数为:GINI 指数+混淆率+(1−占比率).对于后置特征 *B* 来说,其子节点的划分与前置特征 *A* 有明显的差异.对于前置特征 *A*,当划分子节点下的后置特征 *B* 有多个值时,一定存在某种因素导致的不一致数据的出现.对于后置特征 *B* 来说,划分子节点下的前置值出现多种值时,不能说明其中是否存在某种因素导致的不一致数据,需要通过其前置值所对应的后置数据来进行分析.因而该种情况下,混淆率计算公式为

$$混淆率_{Data父}=\frac{Data子1}{Data父}混淆率_{Data子1}+...+\frac{Data子n}{Data父}混淆率_{Data子n} \tag{5}$$

$$混淆率_{Data子1}=\frac{Data子1中a1数据量}{Data子1数据量}*\frac{Data子1中a2的数据量}{Data子1数据量} \tag{6}$$

混淆率主要用于计算该划分子节点的混淆程度.以图 2 中 $b1$ 子节点为例,$b1$ 的混淆率为($Data$ 子 1 中 $a1$ 的数据量/$Data$ 子 1 的数据量)*($Data$ 子 1 中 $a2$ 的数据量/$Data$ 子 1 的数据量),其中的前提为 $b1$ 不是 $a1$,$a2$ 所对应的后置占比最多值.若 $b1$ 是前置特征值 $a1$ 所对应的后置值中所占比率最多的值,那么 $b1$ 的混淆率应为 $Data$ 子 1 中 $a2$ 的数据量/$Data$ 子 1 的数据量,因为我们前提假设为后置值所占比率越多,为该值的可能性越大.因此我们认为 $a1$ 是被正确分割的数据,不被计算在混淆率中.

占比率计算公式为

$$占比率_{Data父}=\frac{Data子1}{Data父}占比率_{Data子1}+...+\frac{Data子n}{Data父}占比率_{Data子n} \tag{7}$$

$$占比率_{Data子1}=\frac{Data子1中a1数据量}{Data父中a1数据量}*\frac{Data子1中a2的数据量}{Data父中a2数据量} \tag{8}$$

其中,占比率主要用于描述某前置特征下特定值的各个后置值所占的比率.如图 2 所示,$b1$ 的占比率为($b1$ 中含 $a1$ 的数据量/$Data$ 父中 $a1$ 的数据量)*($b1$ 中含 $a2$ 的数据量/$Data$ 父中 $a2$ 的数据量),$b2$ 的占比率为($b2$ 中含 $a1$ 的数据量/$Data$ 父中 $a1$ 的数据量)*($b2$ 中含 $a2$ 的数据量/$Data$ 父中 $a2$ 的数据量),$b3$ 的占比率为 $b3$ 中含 $a1$ 的数据量/$Data$ 父中 $a1$ 的数据量.推导可得 $b2$ 占比率为($a1\rightarrow b1$ 的数据量/$a1$ 的数据量)*($a2\rightarrow b1$ 的数据量/$a2$ 的数据量).由此可以看出,某一后置值在其对应的前置值中所占的比率越大,占比率就越大,代表该划分节点越纯净可分.

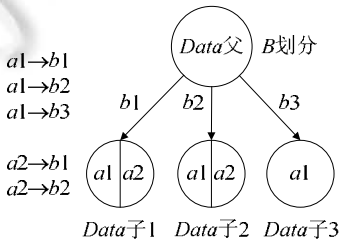

Fig.2 Example of tree model division in Condition 2.2

图 2 情况 2.2 树模型划分情况举例

2.4 讨 论

当我们对未修改过的不一致数据直接分类时,首先需要考虑的是约束条件中的特征对分类结果所带来的负面影响,以及如何通过改进目标函数来降低这种负面影响.对于约束条件中的前置来说,前置决定了后置所对应的值,如果其所对应的后置是多样的,那么后置中存在不一致数据.因此,当对前置进行节点分割时,需要考虑对应后置中每种值所占的比率,某一个后置值所占的比率越大,其他后置值所占的比率越小,证明该前置所对应的后置值不一致数据越少,对该前置的分割越有利于分类结果的提高.因而,我们利用前置所对应的不同后置值的乘积来代表前置的评判标准,可利用积分证明,乘积越大,分布越均衡.

对于约束条件中的后置来说,同一个后置值可能由不同的前置值来决定,因而,用前置的衡量标准已经无法进行评判,需要利用新的评判标准.对于某一个后置值来说,当其对应了多种前置值时,其中的某些可能不属于该前置值,而是由于不一致数据的产生使其变成了该值,因而我们需要考虑该后置值对应的前置值有多少,越少则证明该后置值越有可能是不一致数据.我们利用节点中每个前置值的数据量所占的比率乘积与父节点中所有对应的前置值中该后置值所占的比率乘积作为评判标准,来衡量后置对分类结果的影响程度.

2.5 举例说明

我们通过举例来说明该算法的有效性.表 1 是从真实数据集中抽出的 9 条不一致的数据,为了举例清晰,我们只抽取 9 条数据进行举例,其中,Col3 为前置,Col4 为对应的后置,Col7 为数据分类类别.这 9 条数据是通过父节点对 Col5 进行分割从而得到的一个子节点的数据.

Table 1 Inconsistent data

表 **1** 不一致数据

Num	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	1	1	11 111	123	8	7	0
2	2	1	11 111	8	8	7	0
3	22	2	8 000 000	8	8	7	0
4	3	1	11 111	8	8	7	0
5	22	2	11	2	8	7	10
6	22	2	111	3	8	7	10
7	22	2	1 111	123	8	7	10
8	33	2	1	123	8	7	10
9	333	3	11	2	8	7	10

表 2 为这 9 条数据进行人工修正后的一致数据,其中,对数据 1、数据 7 以及数据 8 的 Col4 进行了相应的修正.

Table 2 Consistent data

表 **2** 一致数据

Num	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	1	1	11 111	8	8	7	0
2	2	1	11 111	8	8	7	0
3	22	2	8 000 000	8	8	7	0
4	3	1	11 111	8	8	7	0
5	22	2	11	2	8	7	10
6	22	2	111	3	8	7	10
7	22	2	1 111	4	8	7	10
8	33	2	1	8	8	7	10
9	333	3	11	2	8	7	10

将表 1 与表 2 中的数据作为训练数据,从而进行树形模型训练.对于不一致数据集来说,通过计算 GINI 指数以及改进策略,每列所得到的结果分别为 0.167,0.178,0.899,0.901,0.494,0.494.通过结果,我们选择数值最小的 col1 作为第 1 层的分割特征.对于一致数据集来说,通过计算 GINI 指数,每列所得到的结果分别为 0.167,0.178,0,0,0.494,0.494.通过结果,我们选择数值最小的 col3 作为第 1 层的分割特征.

表 3 为两条测试数据.将其分别通过不一致树模型与一致树模型进行分类,能够得到不一致树模型将这两条数据分为类别 0,而一致数据将这两条数据分为类别 10.

Table 3 Test data

表 **3** 测试数据

Num	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	1	1	80 000	6	8	7	0
2	2	3	80 000	6	8	7	0

3 算法扩展

不一致数据的约束条件是多样的,不同场景下所对应的约束条件也不同.在真实数据集中,约束条件除了上述的函数依赖以外,还有其他多种约束条件规则.前几节所讨论的约束是最为常见的函数依赖.接下来我们将论述其他几种约束条件.

3.1 条件函数依赖

在该节中,我们讨论另一种常见的约束条件:条件函数依赖.条件函数依赖是函数依赖的扩展,在函数依赖这种约束条件上加上一个前提条件,因此,我们可以通过对函数依赖约束条件下的改进算法进行二次改进,使其适应条件函数依赖场景.

条件函数依赖是指前置特征 A 与后置特征 B 所满足的函数依赖,当且仅当特征 C 为某值时成立.利用决策树对条件函数依赖的不一致数据进行建模时,只需要考虑 1 种情况,即特征 C 是否发生变化:若未发生变化,则该模型建模方式与函数依赖的决策树改进方法一致;若发生变化,则更改其分割策略.图 3 为示例数据,其中,*代表该特征 B 的数据量最多.在该种约束条件下,我们不能像函数依赖那样只考虑前置特征 A 与后置特征 B 两个特征对类别结果的影响,还要考虑条件特征 C 的影响因素.

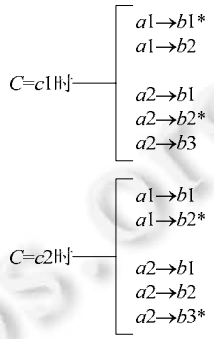

Fig.3 Example data

图 3 示例数据

因此,分割策略更改为如下形式:

$$更改率_{Data父}=\frac{Data子1}{Data父}更改率_{Data子1}+...+\frac{Data子n}{Data父}更改率_{Data子n} \tag{9}$$

$$更改率_{Data子1}=1-\left[\left(\frac{b_k}{Data子1中a1}\right)^2+\left(\frac{b_k}{Data子1中a2}\right)^2\right] \tag{10}$$

其中,b_k 为非最多的其他值数据量,针对 $c1$ 下的 $\left(\frac{b_k}{Data子1中a1}\right)^2$ 来说,就是数据 $a1\rightarrow b2$ 相关的数据量.

3.2 函数集合

在第 2 节中我们只讨论了含有 1 个条件规则的情况,当含有多种条件规则,例如 $A\rightarrow B,C\rightarrow D,E\rightarrow F$ 等这种函数集合时,我们所考虑的情况应与第 2 节相同.因为任意两种条件规则之间不存在交集,因此可以使用第 2 节中的改进算法进行决策树建模.

当多种条件规则为 $A\rightarrow B,B\rightarrow C$ 时,尽管两条条件规则中出现了交集,但其实质也和第 2 节一样.因为单独观察每个条件规则时,其作为独立的条件规则,所考虑的因素与第 2 节中的情况一致.当将两个条件规则联系起来时,A 尽管能够间接影响到 C,但同样可以从 B 的角度来考虑 C 的不一致,因此在这种情况下,所考虑的因素与第 2 节一致.

3.3 多属性函数依赖

多属性函数依赖是指 $AB\rightarrow C$ 这种约束条件,当 A 为 x、B 为 y 时,C 为 $f(x,y)$.当 A 与 B 同时对特征 C 进行约束时,可将其看成条件函数依赖的变形,即当 A 为 x 时,$B\rightarrow C$;当 B 为 y 时,$A\rightarrow C$ 这两种条件约束规则.在决策树改进算法中对节点分割时,当分割 A 时,所考虑的是 B 与 C 所存在的不一致数据的比率,因此可利用目标函数公式(9)与公式(10)来进行分割;当分割 B 时,所考虑的是 A 与 C 所存在的不一致数据的比率,因此同样可利用目标函

数公式(9)与公式(10)来进行分割.当对 C 进行分割时,所考虑的是 C 对应的 A 与 B 的相应后置值中,C 是否占的比率最多,因此利用公式(5)~公式(8)进行对 C 的分割评判.

4 实验结果

本文实验采用的是一台 Windows 7 64 位、4GB 内存的系统环境,用 C++语言实现.数据集包含两类:第 1 类采用的是 UCI 的 Chess(King-Rook vs. King) Data Set,并在其上进行了不一致数据生成;第 2 类采用的是合成数据集,通过约束条件下的函数依赖进行数据生成,共 5 万条数据,分布在 3 个类别中,每条数据有 10 个特征.

本文的决策树采用多叉决策树实现,并利用代价复杂度剪枝的方法对其进行剪枝.对比算法利用不加占比率、混淆率以及分歧率,且目标函数为 GINI 指数的多叉决策树.在 UCI 数据集上进行对比实验,效果如图 4、图 5 所示,横坐标为不一致数据在数据集中所占的比率,纵坐标为分类结果的准确度.

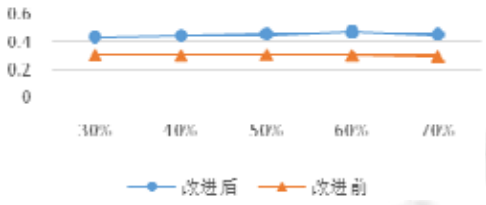

Fig.4 Comparison results of 10 000 data

图 4 10 000 条数据对比结果

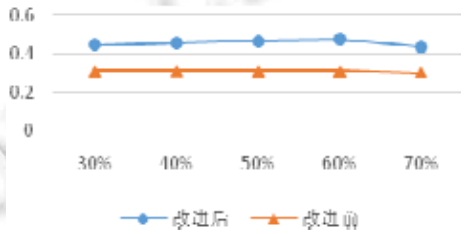

Fig.5 Comparison results of 28 000 data

图 5 28 000 条数据对比结果

改进前方法首先对不一致数据进行人工修正,然后从数据中随机筛选 2/3 作为训练数据,1/3 作为测试数据.利用目标函数为 GINI 指数的多叉决策树对纠正后的不一致数据进行分类,最后利用代价复杂度剪枝方法进行剪枝.图 4 中的实验对 10 000 条数据、18 种 target 进行分类.图 5 中的实验对 28 000 条数据、18 种 target 进行分类.由于每次随机筛选的训练数据不同,因此在对比实验中,改进前方法的实验结果会出现小幅度震荡.

我们利用 5 000 条、10 000 条、20 000 条以及 28 000 条数据进行对比实验,从图 6 可以看到,随着数据量的增加,本文算法的稳定性较好.

为了测试不一致数据比例对运行时间的影响,我们使用 28 000 条数据将不一致数据比例由 30%变化到 70%,时间单位为 ms.实验结果如图 7 所示:当不一致数据量不同时,所用时间均在 15 000ms~35 000ms 之间.运行时间比较稳定,并没有出现大幅度的起伏,体现了该算法可扩展性较好.

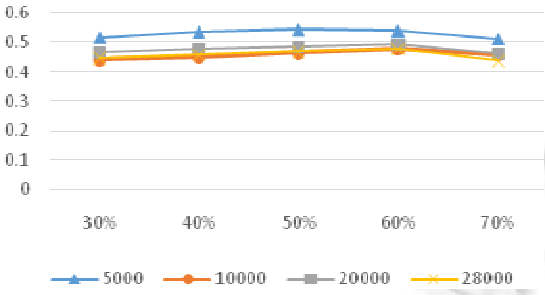

Fig.6 Experimental results of accuracy in different amounts of data

图 6 不同数据量精确性实验结果

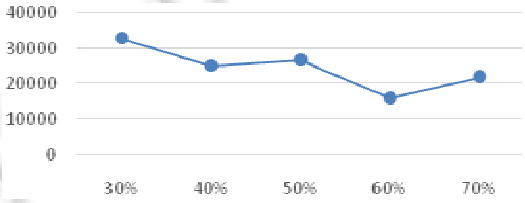

Fig.7 Experimental results of the influence of inconsistent data ratio on running time

图 7 不一致数据比例对运行时间影响实验结果

之后,我们在合成数据集上进行了对比实验,所用的对比实验模型是二叉决策树模型,并利用了代价复杂度剪枝方法.对比算法为不加占比率、混淆率以及分歧率,且目标函数为 GINI 指数的多叉决策树.我们在 50 000

条数据上进行了对比实验,实验结果如图 8 所示,其中,横坐标为不一致数据在数据集中所占的比率,纵坐标为分类结果的准确度.

和 UCI 数据集一样,对比实验首先对不一致数据集进行近似算法修正数据[17],然后对修改后的一致数据进行决策树分类.在数据中随机筛选 2/3 作为训练数据,1/3 作为测试数据.利用目标函数为 GINI 指数的多叉决策树对修正后的不一致数据进行分类,最后利用代价复杂度剪枝方法进行剪枝.从图 8 可以看出,在修改前的不一致数据上,进行改进算法模型构建所得到的分类结果优于改进前的分类器效果.由于在不一致数据量较大的情况下,修正的结果不一定正确,因而当不一致数据量增大时,分类效果会有所下降.

我们利用 5 000 条、10 000 条、20 000 条以及 28 000 条数据分别进行对比实验,从图 9 可以看出,随着数据量的增加,该算法的准确度仍然很稳定.可以看出,本文提出的算法稳定性较好.

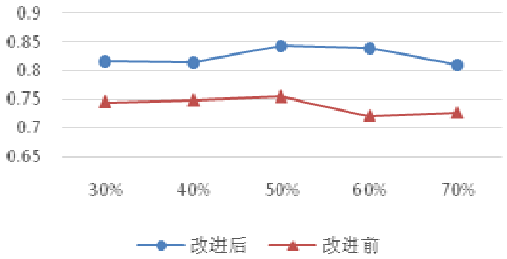

Fig.8 Comparison results of 50 000 data

图 8 50 000 条数据对比效果图

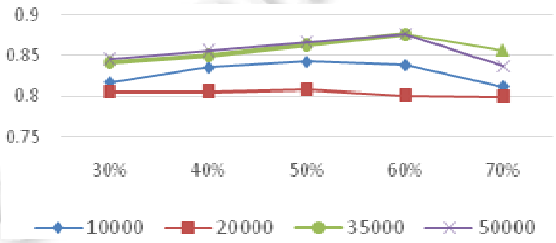

Fig.9 Different data volume contrast experiment effect diagram

图 9 不同数据量对比实验效果图

5 结论与展望

本文对决策树生成算法的目标函数进行了改进,调整了不一致数据相应特征的影响因子,根据约束条件的前置与后置在约束中所具有的不同特性,制定出了不同的评判标准,并通过实验证明了改进后的目标函数对未进行人工修改的不一致数据有较好的分类效果.通过与原始决策树算法进行分类对比,分类准确度要明显高于数据修改后的多叉决策树.针对约束条件中的前置与后置,对目标函数进行了改进,从而使算法适用于不进行人工修正的不一致数据上的分类问题,节省了大量的人工修正时间.

在后续的研究中,我们将尝试对不进行人工修正的不一致数据特征进行特征选择,通过针对不一致特征提出一个新的衡量因子,并考虑对随机森林进行改进,从而得到更好的分类效果.

References:

[1] Zhang AZ, Men XY, Wang HZ, Li JZ, Gao H. Inconsistent data detection and reparation based-on hadoop. Journal of Frontiers of Computer Science and Technology, 2015,9(9):1044−1055 (in Chinese with English abstract).

[2] Alpaydin E. Introduction to Machine Learning. 3rd ed., The MIT Press, 2014.

[3] Kohn LT, Corrigan JM, Donaldson MS. To Err is Human: Building a Safer Health System. 2000.

[4] Fan W. Data quality: Theory and practice. In: Proc. of the 13th Int'l Conf. in Web-Age Information Management (WAIM 2012). Berlin, Heidelberg: Springer-Verlag, 2012. 1−16. [doi: 10.1007/978-3-642-32281-5_1]

[5] Fan W, Geerts F. Foundations of data quality management. Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2012.

[6] Dung PM. Integrating data from possibly inconsistent databases. In: Proc. of the 1st IFCIS Int'l Conf. on Cooperative Information Systems. IEEE, 1996. 58−65. [doi: 10.1109/COOPIS.1996.554998]

[7] Liu XL, Li JZ. Consistent estimation of query result in inconsistent data. Chinese Journal of Computers, 2015,38(9):1727−1738 (in Chinese with English abstract).

[8] Li JZ, Liu XM. An important aspect of big data: Data usability. Journal of Computer Research and Development, 2013,50(6):1147−1162 (in Chinese with English abstract).

[9] Bertossi L, Bravo L, Franconi E, Lopatenko A. Complexity and approximation of fixing numerical attributes in databases under integrity constraints. Information Systems, 2008,33(4):407−434. [doi: 10.1016/j.is.2008.01.005]

[10] Bravo L, Bertossi L. Consistent query answering under inclusion dependencies. In: Proc. of the 2004 Conf. of the Centre for Advanced Studies on Collaborative research. IBM Press, 2004. 202−216.

[11] Molinaro C, Greco S. Polynomial time queries over inconsistent databases with functional dependencies and foreign keys. Data & Knowledge Engineering, 2010,69(7):709−722. [doi: 10.1016/j.datak.2010.02.007]

[12] Flesca S, Furfaro F, Parisi F. Querying and repairing inconsistent numerical databases. ACM Trans. on Database Systems (TODS), 2010,35(2):11−14. [doi: 10.1145/1735886.1735893]

[13] Arenas M, Bertossi L, Chomicki J. Consistent query answers in inconsistent databases. In: Proc. of the 18th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. ACM Press, 1999. 68−79. [doi: 10.1145/303976.303983]

[14] Cong G, Fan W, Geerts F, Jia XB, Ma S. Improving data quality: Consistency and accuracy. In: Proc. of the 33rd Int'l Conf. on Very Large Data Bases. VLDB Endowment, 2007. 315−326.

[15] Bohannon P, Fan W, Flaster M, Rastogi R. A cost-based model and effective heuristic for repairing constraints by value modification. In: Proc. of the 2005 ACM SIGMOD Int'l Conf. on Management of Data. ACM Press, 2005. 143−154. [doi: 10.1145/1066157.10661 75]

[16] Fellegi IP, Holt D. A systematic approach to automatic edit and imputation. Journal of the American Statistical Association, 1976, 71(353):17−35. [doi: 10.1080/01621459.1976.10481472]

[17] Lopatenko A, Bravo L. Efficient approximation algorithms for repairing inconsistent databases. In: Proc. of 2007 IEEE the 23rd Int'l Conf. on Data Engineering. IEEE, 2007. 216−225. [doi: 10.1109/ICDE.2007.367867]

[18] Winkler WE. Methods for evaluating and creating data quality. Information Systems, 2004,29(7):531−550. [doi: 10.1016/j.is.2003.12.003]

[19] Arenas M, Bertossi L, Chomicki J, He X, Raghavan V, Spinrad J. Scalar aggregation in inconsistent databases. Theoretical Computer Science, 2003,296(3):405−434. [doi: 10.1016/S0304-3975(02)00737-5]

[20] Bertossi L, Bravo L, Franconi E, Lopatenko A. Complexity and approximation of fixing numerical attributes in databases under integrity constraints. In: Proc. of the Int'l Workshop on Database Programming Languages. Berlin, Heidelberg: Springer-Verlag, 2005. 262−278. [doi: 10.1007/11601524_17]

[21] Calì A, Lembo D, Rosati R. On the decidability and complexity of query answering over inconsistent and incomplete databases. In: Proc. of the 22nd ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. ACM Press, 2003. 260−271. [doi: 10.1145/773153.773179]

[22] Chomicki J, Marcinkowski J. Minimal-Change integrity maintenance using tuple deletions. Information and Computation, 2005, 197(1):90−121. [doi: 10.1016/j.ic.2004.04.007]

[23] Fuxman A, Fazli E, Miller RJ. Conquer: Efficient management of inconsistent databases. In: Proc. of the 2005 ACM SIGMOD Int'l Conf. on Management of Data. ACM Press, 2005. 155−166. [doi: 10.1145/1066157.1066176]

[24] Fuxman AD, Miller RJ. First-Order query rewriting for inconsistent databases. In: Proc. of the Int'l Conf. on Database Theory. Berlin, Heidelberg: Springer-Verlag, 2005. 337−351. [doi: 10.1007/978-3-540-30570-5_23]

[25] Lopatenko A, Bertossi L. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In: Proc. of the Int'l Conf. on Database Theory. Berlin, Heidelberg: Springer-Verlag, 2007. 179−193. [doi: 10.1007/11965893_13]

[26] Wijsen J. Database repairing using updates. ACM Trans. on Database Systems (TODS), 2005,30(3):722−768. [doi: 10.1145/1093382.1093385]

[27] Arenas M, Bertossi LE, Chomicki J. Answer sets for consistent query answering in inconsistent databases. Theory and Practice of Logic Programming, 2003,3(4-5):393−424. [doi: 10.1017/S1471068403001832]

[28] Greco G, Greco S, Zumpano E. A logical framework for querying and repairing inconsistent databases. IEEE Trans. on Knowledge and Data Engineering, 2003,15(6):1389−1408. [doi: 10.1109/TKDE.2003.1245280]

[29] De'ath G, Fabricius KE. Classification and regression trees: A powerful yet simple technique for ecological data analysis. Ecology, 2000,81(11):3178−3192. [doi: 10.1890/0012-9658(2000)081[3178:CARTAP]2.0.CO;2]

[30] Hunt EB, Marin J, Stone PJ. Experiments in Induction. New York: Academic Press, 1996.

[31] Breiman L, Friedman J, Olshen R, Stone C. Classification and Regression Trees. New York: Chapman and Hall, 1984.

附中文参考文献:

[1] 张安珍,门雪莹,王宏志,李建中,高宏.大数据上基于 hadoop 的不一致数据检测与修复算法.计算机科学与探索,2015,9(9):1044−1055.

[7] 刘雪莉,李建中.不一致数据上查询结果的一致性估计.计算机学报,2015,38(9):1727−1738.

[8] 李建中,刘显敏.大数据的一个重要方面:数据可用性.计算机研究与发展,2013,50(6):1147−1162.

王鹤澎(1992－),女,山东郓城人,硕士,主要研究领域为海量数据计算.

李建中(1950－),男,教授,博士生导师,CCF 会士,主要研究领域为无线传感器网络,物联网,数据库,海量数据管理.

王宏志(1978－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为大数据管理,数据质量管理,XML 数据管理.

高宏(1966－)女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为无线传感器网络,物联网,海量数据管理,数据挖掘.