基于决策树规则的回归测试技术研究

廖 敏,李龙澍,李 森
(安徽大学 计算机系,安徽 合肥 230031)

摘 要:回归测试中测试用例的优化选择是个关键环节,借助黑盒测试中的等价类划分选择测试用例可以提高测试的效率。文中介绍一种基于决策树规则的分类方法实现等价类的划分。该方法通过决策树提取规则,在按照一定的优先级对提取的决策树规则进行排序后,对测试用例库中的每个测试用例,选择优先级最高的规则进行匹配分类,最后从每一分类中选择具有代表性的测试用例,同时介绍了怎样构造该模型。该方法在保证了分类精度的同时能够提高测试的效率,该方法是有效的。

关键词:决策树;决策规则分类;回归测试;规则排序;测试用例
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2011)05-0025-04

Research of Regression Testing Technology Based on Rule of Decision-Tree

LIAO Min, LI Long-shu, LI Sen
(College of Computer Science and Technology, Anhui University, Hefei 230031, China)

Abstract: The optimization choice of test case in regression testing is one of the key steps. Advance the efficiency of testing via equivalence partitioning method belonging to black box testing. There introduce a kind of classification method based on the decision tree algorithm to implement the sorting of equivalence class for advancing the efficient of testing. First get the decision rules via decision tree, and then choose the highest priority of rules to match the class for each test case in the test case library once ordering the decision rules which were ordered according to the priority; At last choose the most representative test case. By the way, demonstrate how to build the model. The method in this article can improve the efficiency of testing at the moment of ensuring the accuracy of the classification and it is effective.

Key words: decision-tree; decision rule classification; regression testing; rule ranking; test case

0 引 言

回归测试是在软件开发过程中为了确保软件质量而进行的一种常用的测试方法[1]。因为软件的输入条件不止一个,而每一个输入又有多种选择,当多个输入组合起来时输入条件是无限的,从而生成的测试用例也是无以计数的。但是,在软件周期中为了缩短时间和减少成本,在测试时就必须选择尽量少的测试用例,尤其是在回归测试中更加需要选择最有效的测试用例,因为回归测试所给予的时间往往比软件刚被开发出来时测试的时间要少。因此针对如何减少回归测试成本,提高回归测试效率的研究具有重要意义。回归测试的情况往往比刚开发时的测试要复杂:首先,在软件的整个生命周期中,为了保证软件能够满足客户的需求,经常要进行更改,每次更改之后都要进行回归测试来验证是否在满足现有功能的前提下导致了新的缺陷。经过多次更改之后,软件的功能和结构与原有的需求功能文档相比可能已经作了很大的调整,但是如果此时的文档没有根据需求功能相应做出调整修正,那么就不能再作为选取测试用例的依据。其次,有些新开发的软件可能是在原有的低版本系统基础上开发的,如果将这些系统完全重写并更新,代价是不可承受的。这些旧系统由于时间过长可能文档早已缺失或不一致。在这种文档不全的条件下,对原有系统部分的测试就只能通过分析可执行程序得出的结论[2]来构建选择测试用例。

基于回归测试的这些困难,可以通过黑盒测试的方法来生成测试用例建立回归测试用例库。其中比较有效的方法是等价类划分法[3]。根据系统的可执行程序的输入得出相应的输出,然后分析输入-输出的关

收稿日期:2010-09-29;**修回日期**:2011-01-09
基金项目:安徽省自然科学基金(090412054)
作者简介:廖 敏(1984-),女,安徽安庆人,硕士研究生,研究方向为软件分析与测试;李龙澍,教授,博士生导师,研究方向为知识工程、软件分析与测试。

系,将输入的范围划分为若干个等价类。鉴于输入条件的多元性及其取值的复杂性,想用人工手动去实现有效的等价类划分有一定的难度。可以采用数据挖掘的方法来解决这个问题[4],其中一种有效的方法是文献[5]提出的基于决策树规则的分类方法,该决策树规则的排序采用基于规则的排序策略。

1 基于决策树规则的分类方法简介

1.1 分类方法

软件的分类属于数据挖掘技术的一种实际应用,拥有一般分类分析过程的特征。分类分析是根据已知类别的样本数据来建立类别描述规则,并且对新样本观察的属性值进行判别归类,具体描述为:对于一个给定的数据集,该数据集具有 $n+1$ 个属性:$A_1,A_2,\cdots,A_n;C$,其中 C 作为类别属性,将此数据集按 C 的属性值分类,形成一个分类模型,再对该模型进行评估验证,生成类别的描述规则后,用该模型对新的数据集进行分类预测,即在已知新数据集 $A_1,A_2,\cdots,A_n$ 的值的情况下预测类别属性 C 的值[6]。文献[4]中给出了典型的解决分类分析问题的模型。文中确立采用文献[5]提出的基于决策树规则的分类方法来构建软件的分类模型。

决策树是一种类似于流程图的树结构[7],见图1;其中每个内部节点(非树叶节点)表示在一个属性上的测试,每个分枝代表一个测试输出,每个叶节点(或终节点)代表存放一个类标号,代表一个分类,树的最顶层是根,由根到各个叶节点的路径描述可得到各种分类规则。决策树采用自顶向下、分而治之的贪心方法来建构树,易于提取显示规则、计算量相对较小、分类速度快。

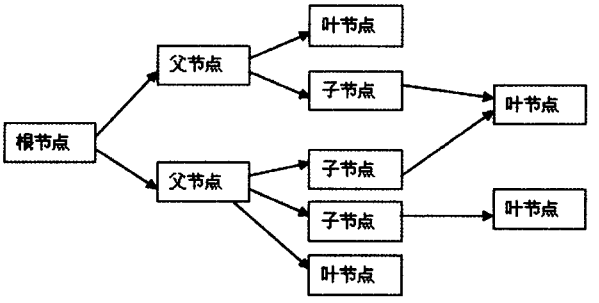

图1 一个决策树结构

基于决策树规则的分类法使用一组 IF-THEN 规则进行分类,该规则是从已构建好的决策树中提取。生成决策树规则后,需要对规则集进行按优先级排序。主要有两种排序方法:基于规则的排序方法和基于类的排序方法。文献[5]的决策树规则采用的是基于规则的排序方法。该方法与另一种排序方法基于类的排序方法相比较,在保证分类准确率的同时可以得到更好的分类精度[5]。

1.2 决策树算法

决策树通常通过两个阶段来生成:第一阶段,树的构建阶段,构建一棵巨大的树。这棵树描述了输入数据集的记录。第二阶段,树的剪枝阶段,在这个阶段确定最后的树的大小。通过减小树的大小,可以生成一些更一般的规则,数目要少得多。决策树又称为分类树算法,在根节点,通过检查数据集计算出局部最优的划分标准,然后按照根节点的划分标准对数据集进行划分:一分为多,每一部分对应一个子节点,最后算法在每个子节点上递归执行。一个通用的决策树策略如下[4,8]:

输入:节点 Node,训练集 Test,分支指标 SI

输出:以节点 Node 为根节点的基于训练集为 Test 分支指标为 SI 的决策树

1:buildTree(Node,Test,SI)

2:初始化根节点

3:在 Test 中计算 SI,求解节点 Node 的分支方案

4:if(节点 Node 满足分支条件)

5: 选择最好的分支方案划分 Test 为 Test1、Test2

6: 构建子节点 Node1、Node2

7: buildTree(Node1,Test1,SI)

8: buildTree(Node1,Test1,SI)

决策树的构造方法很多,文中采用 C4.5 算法构建决策树,该算法具体内容详见文献[8]中介绍。

1.3 基于决策树规则的分类方法

构建决策树以后,从根到树叶的每一个分支路径相应提取一条规则,以 IF-Then 形式形成分类规则,IF-Then 规则可以表示为如下形式:rulei:(Conditioni)→Resulti。规则左边为该规则的前提条件即规则前件,右边为该规则的结论即规则后件。

一个分类规则好不好,它的质量可以通过准确率和覆盖率两个属性来度量[9],所谓的准确率指的是该规则能正确预测(即满足规则前件和后件)的实例个数和分配给该规则的实例总个数之比,准确率属性度量的是该规则正确预测目标值的可能性有多大,可以记为 $\text{accuracy(rule)}=N_{\text{correct}}/N_{\text{cover}}$。所谓的覆盖率[10]指的是该规则能覆盖到(即满足规则前件)的实例个数和构造的测试用例集中实例总个数之比,覆盖率度量的是在构造的测试用例集中分配了多少实例给该规则,可以记为 $\text{coverage(rule)}=N_{\text{cover}}/|S|$。

其中:N_{cover}——满足规则的前提条件的用例数,N_{correct}——同时满足规则前提条件和结论的用例数,$|S|$ 是用例总数。

从决策树提取出决策规则集后,决策规则集的数

量往往很大或有冗余,因此要对决策规则进行排序,从而找出优先级最高的规则来进行匹配以确定分类,从而提高规则分类的速度和精准度。规则的匹配:是对于每一个测试用例,在已排好序的规则集中按优先级顺序查找能够与之相匹配的规则,如果只有唯一一条规则符合,那么把该测试用例纳入到所匹配的规则决策值对应的这一类中;如果有多个规则都符合,则将新测试用例归结到所有匹配的规则中优先级最高的规则所属的类别。基于规则的排序基本思想描述如下:第一步,计算出规则的长度、准确率与覆盖率,并算出长度与准确率的乘积;第二步,依照长度、覆盖率以及第一步中得出的乘积三者从高到低对规则集中的每一个规则进行排序,其中乘积较大的优先级越高;如果两个规则中的乘积一样,则长度较大的优先级越高;如果两个规则中的乘积和长度一样,则覆盖率较高的优先级越高。这里采用规则的长度与准确率的乘积是因为越长的规则就越接近于完全的匹配,同时准确率越大,规则的分类值的信赖度就越高[11]。这里是从规则的三个属性来考虑的:长度、准确率和覆盖率。用伪代码描述的规则排序构建算法过程如下:

输入:从决策树提取的决策规则集

输出:排好序的规则集

1:for i ← 0 to N do

2:length(rules[i])//计算规则的长度、准确率、覆盖率

3:accuracy(rules[i])

4:coverage(rules[i])

5:for i ← 0 to N do

6:for j← i to N do//按乘积大小排序

7:if length(rules[i]) * accuracy(rules[i])<length(rules[j]) * accuracy(rules[j])

8:swap(rules[i],rules[j])//交换规则的优先级顺序

9:else if length(rules[i]) * accuracy(rules[i])==length(rules[j]) * accuracy(rules[j])

10://如果规则长度与准确率的乘积相等,则按规则长度排序

11: if length(rules[i])<length(rules[j])

12:swap(rules[i],rules[j])

13:else if length(rules[i]) * accuracy(rules[i])==length(rules[j]) * accuracy(rules[j]) and length(rules[i])==length(rules[j])

14://如果规则长度与准确率的乘积相等,且规则长度也相等的话,则按覆盖率排序

15:if coverage(rules[i])<coverage(rules[j])

16:swap(rules[i],rules[j])

2　使用基于决策树规则的分类方法构造分类模型划分等价类

使用基于规则排序的决策规则划分测试用例等价类的系统结构见图2。

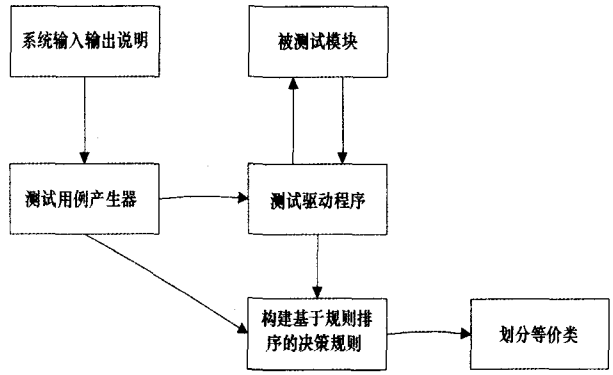

图2　系统结构

系统组成模块及功能如下:

系统可执行程序输入-输出说明:对系统待测试模块的输入输出条件说明,其中包括变量的个数、类型、取值范围、格式、举例等。

待测试模块:可以是一个程序、模块或组件。能够不断的更新。该模块是数据驱动的,输入、输出有固定的格式和类型。

测试用例生成器:根据系统可执行程序输入-输出说明,自动、随机的生成测试用例构建测试用例库。测试用例的数量可以自行决定,但是不要太少,生成的测试用例的输入取值交给待测试模块和构建基于规则排序的决策规则模块。

测试驱动程序:接收生成的测试用例,转交给待测试模块执行,然后将测试输入取值及测试执行输出的结果交给构建基于规则排序的决策规则模块,产生排好序的规则。

构建基于规则排序的决策规则模块:该模块用以构建决策树的数据集是接收来自测试用例生成器的测试用例输入取值和来自测试驱动程序的测试用例执行输出结果数据。首先构造出决策树,然后从决策树提取规则,利用规则排序策略对决策规则进行排序。

系统开始运行后,首先由测试用例生成器根据系统可执行程序输入-输出说明描述生成一组输入取值数据,然后将这组输入取值数据转交给测试驱动程序;测试驱动程序接收到输入取值数据后输入到待测试模块并得到相应的执行输出结果;随后,将输入取值及其相应的输出结果作为生成决策树的依据。对决策树提取的规则排序后,构建算法迭代的运行,依照按优先级高低排序的决策树规则,最后以输出作为等价类分类的目标,将测试用例库中的用例划分为若干等价类,划分等价类后可以选取少量的数据作为代表进行测试。

3 结束语

软件回归测试的关键是测试用例的优化选择[12]。文中借用数据挖掘中的规则分类技术给出一种等价类划分方法，主要介绍了在回归测试中利用基于决策树规则的分类技术来划分等价类的方法模型。可借由此方法编程实现测试用例的自动生成工具。该方法模型采用基于规则的排序策略对决策树规则进行排序，按分类规则将测试用例划分为若干个等价类，然后在每个类中选择少数有代表性的测试用例进行测试，测试成本，实现最小回归测试集的生成。尽管数据挖掘技术在其他方面得到了广泛的应用，但是在工程实践中，某些方法的效率和有效性还需要提高及检验；随着计算机学科技术的发展和研究的深入，将数据挖掘的相关技术越来越多的应用到软件测试中会成为软件测试有效方法之一。

参考文献：

[1] Rothermel G, Harrold M J. Analyzing regression test selection techniques[J]. IEEE Transactions on Software Engineering, 1996,22(8):529-551.

[2] Last K,Friedman M. The Data Mining Approach to Automated Software Testing[C]//Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining. [s. l.]:[s. n.],2003:388-396.

[3] 许 静,陈宏刚,王庆人. 软件测试方法简述与展望[J]. 计算机工程与应用,2003,39(13):75-78.

[4] 李克文,杨志霞. 基于回归测试模型的用例集的优化方法研究[J]. 微计算机应用,2008,29(10):7-11.

[5] 谭俊璐,武建华. 基于决策树规则的分类算法研究[J]. 计算机工程与设计,2010,31(5):1017-1019.

[6] Chen Leida. Data mining methods, applications, tools[J]. Information System Management,2000,17(1):65-70.

[7] 马 菁,顾景文. 决策树在软件测试用例生成中的应用[J]. 计算机技术与发展,2008,18(2):66-69.

[8] Han Jiawei,Kamber M. 数据挖掘：概念与技术[M]. 北京：机械工业出版社,2008.

[9] Tan Pangning,Steinbach M,Kumar V. 数据挖掘导论[M]. 北京：人民邮电出版社,2006:127-136.

[10] 王小丽,段永颢. 软件回归测试用例选取方法研究[J]. 空间控制技术与应用,2010,36(3):47-49.

[11] Wu Jianhua,Song Qinbao,Shen Junyi. A novel association rule mining based missing nominal data imputation method[C]// Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. [s. l.]:[s. n.],2007:44-249.

[12] 曾 强,洪 玫,杨昊苏,等. 软件回归测试中的自动化测试生成方法[J]. 计算机应用研究,2009,26(6):2349-2351.

(上接第 24 页)

被改进算法挖掘出来，虽然这些关联规则占总关联规则的比例不大。

4 结束语

文中对 Apriori 算法的复杂性及存在的缺陷进行分析，提出了 L_Apriori 算法，并在图书推荐服务进行了应用。测试发现以改进后的 L_Apriori 算法为核心的数据挖掘引擎比传统 Apriori 数据挖掘引擎在图书推荐服务模型中能发挥出更大的作用。通过对核心算法的改进，L_Apriori 算法能更好地应用于图书推荐服务模型中，为读者提供更为方便快捷的个性化服务。

参考文献：

[1] 康敏旸,张 安. 改进的 Apriori 数据挖掘算法的应用[J]. 火力与指挥控制,2009,34(10):111-114.

[2] 黄 鹤. 关联规则算法综述[J]. 软件导刊,2009,8(3):56-58.

[3] 陈则芝,李冬梅. 数据挖掘关联规则 Apriori 算法的优化[J]. 山西大同大学学报(自然科学版),2008,24(4),35-37.

[4] Han Jiawei,Kamber M. 数据挖掘：概念与技术[M]. 范明,孟小峰,译. 北京：机械工业出版社,2006.

[5] 鲍 静. 关联规则挖掘及其在图书流通数据中的应用研究[D]. 合肥：合肥工业大学,2007.

[6] 张 昀. 数据挖掘中一种改进的 Apriori 算法[J]. 软件导刊,2009,8(11):87-88.

[7] 陈文庆,许 棠. 关联规则挖掘 Apriori 算法的改进与实现[J]. 微机发展(现更名：计算机技术与发展),2005,15(8):155-157.

[8] Savasere A,Omieeinski E,Navathe S. An Efficient Algorithm for Mining Association Rules In Large Databases[C]//Proceedings of the 21st International Conference on Very Large Databases. New York:ACM,1995:432-443.

[9] Agrawal R,Srikant R. Fast Algorithms for Mining Association Rules[C]//Proceedings of the 20th International Conference on Very Large Databases. Santiago, Chile:Morgan Kaufmann Publisher,1994:487-499.

[10] Toiconen H. Sampling Large Databases for Association Rules [C]//Proceedings of the 22nd International Conference on Very Large Databases. Bombay, India: Morgan Kaufmann Publisher, 1996:134-145.

[11] 卢 露,丁才昌. 关联规则中 Apriori 算法改进的研究[J]. 长江大学学报(自然科学版),2009,6(2):241-243.

[12] 苟元琴,王钧玉. 关联规则在图书馆读者借阅记录中的挖掘应用[J]. 科技信息,2009(17):356-357.