分类号　　密级

江南大学
硕士学位论文

题　　目：C4.5 决策树算法优化及其应用

英文并列题目：Optimization and Application of C4.5 Decision Tree Algorithm

研　究　生：黄秀霞

专　　业：计算机科学与技术

研究方向：计算机软件与理论

导　　师：孙　力

指导小组成员：

学位授予日期：2017 年 1 月

答辩委员会主席：张曦煌

江南大学

地址：无锡市蠡湖大道 1800 号

二 O 一七年 一月

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含本人为获得江南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名：黄秀霞　　日期：　　年　月　日

关于论文使用授权的说明

本学位论文作者完全了解江南大学有关保留、使用学位论文的规定：江南大学有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅，可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文，并且本人电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

签名：黄秀霞　　导师签名：

日期：　　年　月　日

摘 要

C4.5 算法是一种分类预测算法，是数据挖掘算法中的十大经典算法之一。C4.5 算法的优化与应用广泛存在于各个领域，如商业决策、医学领域的病症预测以及生物学领域的基因识别等。为了改进 C4.5 算法的特征选择正确性和数据处理能力，将 C4.5 算法与粒子群算法和模糊算法等相结合是目前比较流行的改进方法。本文主要针对 C4.5 算法的对数运算、属性间相关性影响以及冗余计算等问题进行优化，并将改进后算法应用于学生英语统考成绩预测中。

针对 C4.5 算法计算时间长、属性间相关性影响的问题，提出了一种基于属性间 GINI 指数均值的 C4.5 算法（GC4.5）。首先，运用泰勒级数和等价无穷小的原理对信息增益率的公式进行简化，用“加”，“减”，“乘”，“除”来代替对数运算，目的是减少调用对数运算函数的时间；其次，在简化后的信息增益率公式中引入条件属性间的 GINI 指数均值，用于调整因条件属性间相关性导致的误差。通过大量的 UCI 数据集进行分析预测实验，结果验证，GC4.5 算法较现有的一些 C4.5 改进算法拥有相对较高的分类正确率和较短计算时间。

针对 C4.5 算法的无关属性的影响和相关性的问题，提出了基于属性依赖度计算和 PCA 算法的 C4.5 算法（RPC4.5）。首先，根据属性依赖度的计算公式计算出数据集中条件属性与类属性依赖度，删除依赖度很小的条件属性，避免无关计算；其次，运用 PCA 算法的压缩原理处理数据集，经 PCA 处理后数据集属性组合的主成分相互独立，从而解决属性间相关性的影响问题。通过对大量的 UCI 数据集进行实验，结果证明，RPC4.5 算法与 C4.5 算法以及其它一些 C4.5 的改进算法相比，在分类正确率上有一定提高，建模时间有相对优势。

成绩预测是当前数据挖掘研究的热门应用。由于 C4.5 算法的简单易懂，建模时间短，分类正确率相对较高的特点，成为成绩预测所用算法的首选。本文将 GC4.5 和 RPC4.5 算法应用于学校的英语统考成绩预测中，借助 JAVA 开发平台 Eclipse 和数据挖掘分析工具 WEKA 进行应用实验，结果表明，与改进之前的 C4.5 算法相比较，GC4.5 算法和 RPC4.5 算法的分类预测结果具有较高的正确率，建模时间更短，因此，本文对于 C4.5 算法的改进应用于成绩预测系统中是可行的，并具有一定的实用性。

关键词：C4.5 算法；泰勒级数；GINI 指数；属性依赖度；PCA。

Abstract

C4.5 algorithm, which is one of the top ten classic algorithms in data mining, is classified algorithm for prediction. Optimization and application of C4.5 algorithm is widely exited in various fields. Such as business decision, Condition forecast in medical domain and gene identification in the field of biology, etc. In order to improve correctness of feature selection and data processing ability of C4.5 algorithm, combine C4.5 algorithm with particle swarm optimization (PSO) algorithm or fuzzy algorithm is the most popular methods of improvement. This paper mainly aims at the logarithmic calculation, attribute correlation and redundancy computing problems of C4.5 algorithm to optimize, and the improved algorithm is applied for the student English exam forecast.

The C4.5 algorithm needs large number of logarithm operations, interference from attributes correlation, etc. So the average of GINI index between conditional attributes based C4.5 algorithm (GC4.5) was put forward to solve the problems. Firstly, use Taylor series and equivalent infinitesimal principle to simplify information gain ratio formula, with the "addition", "subtraction", "multiplication", "division" instead of logarithm operations, saved the time of the call logarithmic function. Secondly, introduce the average of GINI index between conditional attributes simplified information gain ratio formula to deal with the error caused by condition attribute correlation. The proposed GC4.5 had been evaluated on a large number of UCI data set. The experimental results show that GC4.5 performs better than some existing C4.5 optimization algorithms.

The C4.5 algorithm always has irrelevant attributes and correlation problems, so improved C4.5 algorithm based on calculation of dependency for attributes and PCA (RPC4.5) was put forward to solve these problems. Firstly, calculating attribute dependency between condition attribute and class attribute according to the attribute dependency calculation formula, delete the condition attribute which dependence is very small, and avoid unrelated calculation. Secondly, the simplified data set will be processed by compression principle of PCA. After PCA handling, the data set attributes combination of principal components are independent of each other, to solve the problem of the influence of the correlation between attributes. Through test on a large number of UCI data set, the results show that, compared with some existing C4.5 optimization algorithms, the accuracy of RPC4.5 algorithm improved significantly, modeling speed of RPC4.5 algorithm has certain advantages.

Performance prediction is current research focus in the field of data mining. Owing to simple and understandability, the modeling time is short, relatively high classification accuracy. Those characteristics make C4.5 algorithm is the first choice of the performance prediction algorithm is used. GC4.5 and RPC4.5 algorithm will be used in the English exam prediction of schools this paper. With the aid of the JAVA development platform Eclipse and data mining analysis tools WEKA, to precede application experiments. Results show that, compared with C4.5 algorithm, classification prediction results of the GC4.5 and RPC4.5 algorithm has higher accuracy, the modeling time is shorter. Therefore, the improvement of the C4.5 algorithm is feasible, and has a certain practicality.

Keywords: C4.5 algorithm; Taylor series; GINI index; dependency for attributes; PCA.

目 录

第一章 绪论

1.1 课题背景与意义

由于大量电子表格数据的出现，数据挖掘（Data Mining）[1-2]得到了广泛的应用。数据挖掘是从资料库或数据库或者其他的信息库所储存的数据中发现有用知识的一个过程，例如关联分析，建立模型，数据变换，重要结构建立和异常现象分析等。经典的数据挖掘算法主要有十个[3]，包括朴素贝叶斯（Naïve Bayes）[4-6]，ID3 算法(Iterative Dichotomiser3) [7-10]，C4.5 算法[11-19]，CART(Classification and Regression Trees) [20-22]等等。这些算法在人工智能(Artificial Intelligence)[23]和机器学习(Machine Learning)[24-25]等领域等到了广泛的应用。

ID3 算法是由 J.Ross Quinlan 在 1975 年的时候提出的用于从数据集中生成决策树的一种算法，是一种基于信息熵计算的分类算法。算法思想简单，构建的模型易于理解。但是，ID3 算法只能处理离散型条件属性的数据集，并且容易造成优先选择属性值数量较多的属性的问题。因此，J.Ross Quinlan 又在此 ID3 算法的基础上进行优化改进，然后提出了 C4.5 算法。C4.5 算法不仅能处理离散型数据集还可以处理连续性数据集，同时，C4.5 算法根据信息增益率计算进行分类，解决了选择属性值比较多的条件属性的偏倚。C4.5 算法进行数据分类时，先将数据进行预先处理，步骤大概有数据清洗、整合和规范化数值等；其次，运用属性度量（信息增益率）的计算公式对条件属性进行计算，得到属性选择的顺序；最后，根据属性选择顺序进行建模，得到分类预测的决策树。较 ID3 算法，C4.5 算法有较大的改进优化，更适用于实际问题。然而，现实世界中信息量迅速增长，并且数据形式越来越多样化，优化 C4.5 算法，提高 C4.5 算法的数据处理能力和分类正确性具有重大的应用价值和现实意义。

C4.5 算法是学术研究的热点，从各个方面对 C4.5 算法进行优化研究，研究成果也非常丰富。算法优化的核心都是提高算法计算效率和分类预测的正确率。C4.5 该算法应用非常广泛[26-40]，主要包括商业决策，医学病症预防，生物特征辨别和教育机构等等，C4.5 算法在这些领域中都具有非常重要的作用和意义。本文将改进的 C4.5 算法应用于学生的英语统考成绩的预测当中，根据实际的数据进行建模分析，帮助学生更好地提高统考的英语成绩，并帮助学校提高教育质量。

第二章首先概述了决策树分类算法，简单介绍了各个分类算法的度量方法和剪枝方法。接着具体描述了 ID3 算法的算法思想、步骤和流程。其次，介绍了 C4.5 算法思想、步骤和流程，并分析了算法的缺点。最后，介绍了两种改进的 C4.5 算法，分别是基于粒子群优化的 C4.5 算法和基于模糊思想的 C4.5 算法。第三章先介绍了泰勒级数和等价无穷小的原理，然后给出了属性间相关性问题的概念，接着根据泰勒级数简化思想和 GINI 指数均值的作用改进 C4.5 算法，并详细地描述了该算法思想、步骤和伪代码。最后，通过 UCI 数据集进行实验，结果说明了该算法的性能有一定的优势。第四章详细地描述了第二个改进思想：基于属性依赖度计算和 PCA 的优化算法。该算法

是在简化对数运算的 C4.5 算法基础上，首先介绍了属性依赖度计算的原理，然后给出了 PCA 算法的理论，提出基于属性依赖度计算和 PCA 的算法优化的 C4.5 算法，同时，给出了详细的算法步骤和伪代码，经过实验验证，说明了该算法具有较高的分类预测的正确率和较快的建模速度。第五章将第三章和第四章的改进算法用于学生英语统考成绩预测中。先介绍了分类预测系统的重要意义，然后算法选择的重要性，通过仿真实验证明了将改进的 C4.5 算法应用于成绩预测方法中是可行的。

1.2 课题研究现状

1.2.1 国外对 C4.5 算法的研究动态

自 1993 年 C4.5 算法被提出以来，对于该算法的优化改进研究从未停止。特别是国外对于 C4.5 算法的研究，时间上比国内早很多。比较显著的优化改进有基于多类属性处理的 C4.5 算法，基于在线学习的 C4.5 算法，基于 Boosting 交互决策树的 C4.5 算法，基于多维数据处理的 C4.5 算法，以及基于模糊系统思想的 C4.5 算法和基于粒子群优化的 C4.5 算法等。

早在 1995 年，Thomas G.Dietterich 和 Ghulum Bakiri[41]提出了可以处理多个类属性的 C4.5 算法。这是首次对 C4.5 算法的较大改进。该改进算法结合 C4.5 算法和二元概念学习算法，以纠错编码作为一个分布式输出表示。这些输出表示提高了 C4.5 算法的泛化性能和大范围多类学习任务的反向传播；能灵活应对训练集样本大小的变化，并用分布式表示特定类的分配给，同时用决策树后剪枝方法防止过度拟合；最后，输出表示还提供可靠的类概率估计。经实验证明，纠错输出编码提供了一个提高 C4.5 算法处理多类属性问题性能的通用方法。

1997 年，Yoav Freund 和 Robert E.Schapire[42]提出在线学习的 C4.5 算法。该算法基于混合损失函数（Mixture Loss Function），并结合 Boosting 算法（Boosting Algorithm），得到一种在线预测模型产生的决策规则。该改进算法不需要任何先验知识而提高了 C4.5 算法的性能，并且形成的学习模型应用于各种领域，包括赌博，多重结果预测，重复游戏等等。

2005 年，Jennifer Lin[43]等人提出了一种基于 boosting 交互决策树的 C4.5 算法；该算法以交互决策树为基础，结合 boosting 思想进行改进，实验证明，改进算法提高了分类正确率。

2009 年，L.Rocach[44]提出了基于集成方法（Ensemble Methods）改进的 C4.5 算法。该算法运用继承方法进行模式分类，提高了分类预测的正确率，并将优化改进之后的 C4.5 算法应用于房地产领域和遗产分配等。

2013 年，随着数据量的增加，Mai.Q[45]等人提出了处理多维数据集的 C4.5 算法。该改进算法结合正则化选择变量技术（Via Regularization Techniques）和线性判别技术（Linear Discriminant Analysis）进行优化改进。经过实验证明，改进的算法可以处理多维的数据集，并且不降低分类正确率。

最近，基于模糊系统（Fuzzy System）的 C4.5 算法[46-53]成为了研究和应用的热点，

这种改进的 C4.5 算法模型叫模糊决策树（FuzzyDT）。该算法先将数据集属性值模糊化表示，根据自然语言的逻辑将模糊值进行分类，从而提高分类正确率。还有一种热门研究是将粒子群优化算法（Particle Swarm Optimization, PSO）[54-61]与 C4.5 算法相结合，通过提高 C4.5 算法的特征选择性能来提高 C4.5 算法的分类正确率。

1.2.2 国内对 C4.5 算法的研究动态

国内对于 C4.5 算法的研究晚于国外，1993 年的时候国家的自然科学基金才关注数据挖掘，但是这并影响 C4.5 算法在国内成为研究的热点。国内对于 C4.5 算法的研究偏重于应用，比较有代表性的研究包括基于信息熵以及加权和的思想改进的 C4.5 算法；基于多叉树结构的直接输出改进的 C4.5 算法；基于粗糙集的 C4.5 算法以及基于信息度量的流特征的 C4.5 算法等等。

2001 年，华南理工大学的唐华松和姚耀文[62]等人提出了一种基于信息熵以及加权和的思想改进的 C4.5 算法。该算法运用“加权和”的原理，按照 C4.5 算法决策树中划分的子集在总数中的比例分配一个权值，接着将这个权值应用于加权熵的计算，最后按照加权熵的大小顺序进行特征选择。该算法通过加权熵改进属性选择度量，优化特征选择，从而提高了算法的分类正确率。经过实验实例证明，该算法的分类预测结果更合理科学，正确率更高。

2003 年，北京邮电大学的姜欣[63]等人提出了一种基于多叉树结构的直接输出方式进行改进的 C4.5 算法，并将算法应用于电信的用户关系管理的系统中。该算法根据 C4.5 算法生成的决策树和多叉树结构的特点优化改进 C4.5 算法。受多叉树存储方式的启发，将 C4.5 算法分类得到决策树的问题转化成了多叉树的存储结构输出问题，再结合直接输出的思想，得到一种更加直观、正确率更高的分类预测的决策树算法。该算法还可以控制决策树的深度，算法更加简单、灵活。将改进的算法借助 TESGMiner 平台应用于客户关系管理系统中，实验证明，输出的分类决策树根据直观、准确。

2009 年，江西理工大学的杨舒晴[64]提出了一种基于粗糙集的 C4.5 算法。该算法根据粗糙集中决策表的理论，将 C4.5 算法的属性选择进行优化改进，给出了一种基于信息熵的离散化连续条件属性的算法；然后根据 C4.5 算法构造决策树的问题，提出一种基于分辨矩阵的简化算法，启发式构造决策树，得到规模更小规则更加简单的决策树。实验证明，基于粗糙集思想原理改进的 C4.5 算法，分类规则个数更少、决策树更简单。

2012 年，郭磊[65]等人提出了一种基于信息度量的流特征选取的 C4.5 算法。该算法包括粗粒度和细粒度两种方式进行特征选择，同时，简约已经选取的属性，提高分类正确率的同时减少算法的计算时间。同一年，周剑峰[66]等人提出了一种简化信息熵计算的 C4.5 改进算法。该算法根据泰勒级数和等价无穷小的原理简化 C4.5 算法信息熵中的对数运算，并将改进后的算法用于网络流量的分类。实验证明，优化后的算法拥有较高的分类效率。

2014 年，阳爱民[67]等人有提出了一种基于二元分配的 C4.5 算法。该算法基于二元搭配词库和 C4.5 算法建模原理，结合统计方法改进 C4.5 算法。并将改进的算法用于博文状态的情感特征分析。实验证明，优化改进的算法具有更加准确的情感分析结果。

2015 年，上海交通大学的宋媛媛[68]等人提出了一种基于多变量的 C4.5 算法,。该算法将数据集处理成一个倒置的决策树模型，不需要参数就可以处理复杂的大数据集。实验证明，改进的 C4.5 算法不仅能处理多变量的数据集，而且还能处理大数据集。同时，改进后的 C4.5 算法拥有较高的分类预测正确率和更简单的计算方式。

综上所述，国内对 C4.5 算法的优化研究主要包括信息熵和特征选择的优化，约简分类规则和决策树规模等，也有一些对于大数据集进行优化改进的 C4.5 算法。这些改进的 C4.5 算法大多应用于实际事例中。

1.3 论文研究内容

本文主要针对 C4.5 算法进行改进和应用。

因为 C4.5 算法主要问题是需要进行大量的对数运算和没有考虑条件属性之间相关性的影响。本文结合泰勒级数等价无穷小的化简原理和 GINI 指数均值的原理提出了一种改进的 C4.5 算法（GC4.5 算法），通过大量的 UCI 数据集进行实验，表明该算法具有较短的建模时间和较高的分类预测正确率。

同时，简化对数运算后的 C4.5 算法依然存在无关属性的影响，本文依据属性依赖度的计算和 PCA 的压缩原理，提出了一种基于属性依赖度计算和 PCA 的 C4.5 优化算法。通过大量的 UCI 数据集进行对比实验，结果表明，较其他比较算法，本文提出的改进算法减少了计算时间和避免了属性相关性的影响。

最后，将本文提出的两个改进的 C4.5 算法应用于学生英语统考成绩预测中，通过实际的数据集进行实验，结果表明，较 C4.5 算法，本文提出的两个改进算法具有较高的分类正确率和较快的建模速度。

1.4 论文结构

本论文共六章。具体安排如下：

第一章主要概述了本次课题的研究背景与意义、国内外对 C4.5 算法的研究动态、本文研究内容和结构。

第二章主要介绍了分类算法的理论和现有的两个改进算法。首先概述了决策树算法，给出了属性选择度量和剪枝方法的概念。其次，介绍了 ID3 算法。在决策树分类预测的模型下，给出 ID3 算法的算法思想、步骤和流程。接着，描述了 C4.5 算法思想、步骤和流程，并分析了算法的缺点。最后，介绍了两个现有的 C4.5 改进算法，分别是基于粒子群优化的 C4.5 算法和基于模糊系统思想的 C4.5 算法。

第三章详细介绍本文提出的基于 GINI 指数均值的优化算法。先介绍泰勒级数和等价无穷小的理论，然后针对属性间相关性问题，将泰勒级数简化思想和 GINI 指数均值引入 C4.5 算法，给出该算法思想、步骤和伪代码。最后，通过实验结果说明该算法的优越性。

第四章详细介绍本文提出的基于属性依赖度计算和 PCA 的优化算法。该算法是在简化对数运算的 C4.5 算法基础上进行优化的，首先详细描述属性依赖度计算的原理，然后 PCA 算法，提出基于属性依赖度计算和 PCA 的算法，并给出算法步骤和伪代码。

最后，通过实验结果说明了改进的算法拥有更高的分类预测正确率和较快的建模速度。

第五章将本文提出的 C4.5 两个改进算法用于学生英语统考成绩预测中。介绍了分类预测系统的重要意义，介绍分类算法的选择对成绩预测的最终结果具有至关重要的影响，通过实际的数据集进行实验，证明了将改进的 C4.5 算法应用于成绩预测方法中是可行的。

第六章对本文内容进行总结，给出论文的主要创新思想点，并指出这篇论文提出的论点存在的问题以及未来研究方向。

第二章　决策树分类算法

2.1 决策树分类算法概述

构造决策树对已经存在的数据集进行分析处理是现在数据挖掘技术的普遍方法。在各种各样的分类算法中，决策树分类算法最为直接。决策树是对数据进行分类，在这个过程中，得到净现值的期望值不小于零的概率，以此来估算项目的风险问题，判断该项目是否可行。是直观运用概率分析的一种图解法。决策树由决策点，状态节点和叶子节点三种节点组成。一般根据由顶向下的递归方法来生成决策树。在决策树构造完成之后，还需对其进行剪枝，剪枝算法有预先剪枝和后剪枝这两种不同的方式。一般情况下使用后剪枝算法。建立决策树的算法有很多，主要有 ID3 算法、C4.5 算法和 CART 算法等都是常用的决策树算法。

属性选择度量

属性选择度量的公式计算是计算属性分裂的规则，是一种将已知类属性的训练数据样本集的分区 D 划分成单个类的启发式分类策略。

决策树分类算法常用的属性选择度量主要有三种，如表 2-1 所示。

表 2-1　决策树算法常用属性度量

属性选择度量	缺点	使用的算法
信息增益	通常选择拥有较多值的属性	ID3
信息增益率	随着不断递归划分增益率会变的不稳定	C4.5
GINI 指数	考虑每个属性的二元划分	CART

综上所述，决策树分类算法还存在以下几个常见的主要问题。

- 选择分裂属性度量；
- 对分裂属性进行排序；
- 选取分裂的数目；
- 平衡树的结构并修剪；
- 停止判断的规则。

预先剪枝和后剪枝

选择属性度量和决策树的修剪策略[69-71]是区分决策树不同算法的主要因素。有

两种基本的分类决策树的剪枝方式：预先剪枝和后剪枝。

先剪枝：预先停止决策树的构建，然后对分类决策树进行剪枝处理。停止的时候，当前节点就是叶子节点。

先剪枝的特点：停止构建决策树的判断标准是：预定义的阀值。选定恰当的阀值是非常困难的。

后剪枝：当决策树“完全生长”结束之后，再将决策树进行剪枝处理。把被删除的节点用一个叶子节点代替，并剪掉相应的子树。

后剪枝的特点：用被剪子树中出现频率最高的类代替该子树，作为叶子节点。

决策树分类算法的剪枝方式一般都在以上两种策略范围内，比如 CART 算法使用代价复杂度剪枝方式就是后剪枝的一种；C4.5 算法沿用的一种被称为悲观剪枝的方法也是一种后剪枝策略。

另外，还有一种剪枝策略，就是预先剪枝跟后剪枝相组合的方法。这种剪枝方法可以结合两种剪枝策略的优点，使得生成的最终决策树可以更加紧凑、简单且更加准确。

2.2 ID3 算法

在决策树学习中，ID3 (Iterative Dichotomiser3) [8]是由 J. Ross Quinlan 在 1975 年提出的用于从数据集中生成决策树的一种算法。ID3 通常用于机器学习和自然语言的处理领域。决策树技术包括构造树模型的分类过程。一旦决策树被构建,它将用来分类预测数据库中的每一个样本，并把分类结果显示出来。

ID3 算法是一种基于信息熵的分类算法,其基本思想是所有的例子都根据属性设置中值的不同，映射到不同的分类，其核心是从条件属性集中确定最好的分类属性。ID3 算法的分裂属性选择标准是信息增益，一般是拥有最高的信息增益的条件属性被选中为当前节点的分裂属性，为了使分裂的子集所需要的信息熵最小。根据不同的属性值,可以建立决策树的分支结构,分支结构节点和下一分支结构的建立都是递归地执行以上步骤，直到所有的样本都在同一个分支上，即属于同一类别中。

2.2.1 信息熵和信息增益

ID3 算法分裂属性的选择准则中将使用到信息熵的概念。

信息熵

信息熵在物理学上是描述信源的不确定度的，而在数学上则表示了信息冗余度和概率之间的关系。信息增益度就是两个信息熵之间的差值。数据集中样本对于一个类的信息熵即为其中一个信息熵的值；而另外一个信息熵已知某一属性的值后这个数据集中的样本关于某一个类的信息熵。其具体定义如下：

假定 P_i 是样本取属性值 C_i 的概率，并且规定$\sum p_i = 1$，则公式(2.1) 定义了熵的概念。

$$\mathrm{H}\left(p_1, p_2, \cdots, p_s\right) = \sum_{i=1}^{s} -\left(p_i \log p_i\right) \tag{2.1}$$

熵决定了该属性在其数据集中排序的位置状态。H = 0 表示属性最好的分类设置。换句话说，也就是熵的值越高，分类过程便存在越高的改善空间。

信息增益

信息增益即为原先所需的信息量与分类后的新的信息需求之间的差值。这是通过计算每个细分出来的数据集中原始熵的值和加权后熵的和来确定的，该计算可用公式(2.2)来表示。

$$G\left(D,S\right) = \mathrm{H}\left(D\right) - \sum P\left(D_i\right)\mathrm{H}\left(D_i\right) \tag{2.2}$$

D：表示一个数据集；

D_i：表示属于第 i 类的样本；

S：表示数据的类属性个数。

2.2.2 ID3 算法思想

在实际应用时，运用 ID3 算法中的信息增益的运算公式对数据集中的每一个条件属性进行计算，得到一系列的值，选取其中最大的值并将其对应的属性作为当前数据集的测试属性。若果用决策树来描述数据集，节点则为经过计算选出的测试属性，再以这个属性为标记，它的每个值都是它的一个分支,以此类推来划分数据集中的样本。算法的核心思想如下：

对于一个数据集 T，设 T 为已知类属性样本的一个训练集。设

C_i（i=1,2，……，n）：为类标记属性具有 n 个不同的值；

TC_i：训练集 T 中属于 C_i 类的样本集合；。

$|T|$：是训练集 T 的样本个数；

$|TC_i|$：是训练集 TC_i 中的样本个数。

公式(2.3)表示了样本分类的所需要的期望信息（也就是信息熵）。

$$Info(T) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2.3}$$

p_i：是 T 中任意一个样本属于类 C_i 的不等于零的概率，并用 $\frac{|TC_i|}{|T|}$ 进行估算得到。

$Info(T)$：是识别数据集 T 中样本的类标号所需的平均信息熵。

该计算公式只用到每个类的元组在数据集元组中所占的比例。

现在，再把训练集 T 中的样本按属性 A 进行划分，具有 m 个不一样的值，是 $\{a_1, a_2, \cdots, a_m\}$，假设 A 是离散性属性，则值与属性 A 上测试的 m 个输出直接一一对应，属性 A 把训练集 T 分成了 m 个子集 $\{T_1, T_2, \cdots, T_m\}$，其中 T_j 包含了 T 中的元组，在属性 A 上的值为 a_j。

则公式(2.4)为按 A 划分 T 的元组分类的信息熵。

$$Info_A(T) = \sum_{j=1}^{m} \frac{|T_j|}{|T|} Info(T_j) \tag{2.4}$$

$\frac{|T_j|}{|T|}$：表示属性 A 是第 j 个值的时候，样本的个数在样本总数中所占的比重；

$Info_A(T)$：表示按属性 A 划分对数据集 T 的样本进行分类所需要的期望值信息。该值越小，表明每个属性值所属类别的纯度越高。

根据信息增益的定义，具体计算公式为公式(2.5)所示。

$$Gain(A) = Info(T) - Info_A(T) \tag{2.5}$$

公式(2.5)是在已知属性 A 的值的条件下导致的信息需求的期望值的减少量。选择信息增益最大的一个条件属性为当下的节点的分裂属性。

2.2.3 ID3 算法流程图

决策树由决策点，状态节点和叶子节点三种节点组成。一般根据由顶向下的递归策略来生成决策树。在决策树构造完成之后，还需对其进行剪枝，剪枝算法有预先剪枝和后剪枝这两种。

ID3 算法创建一棵决策树的算法流程如图 2-1 所示。

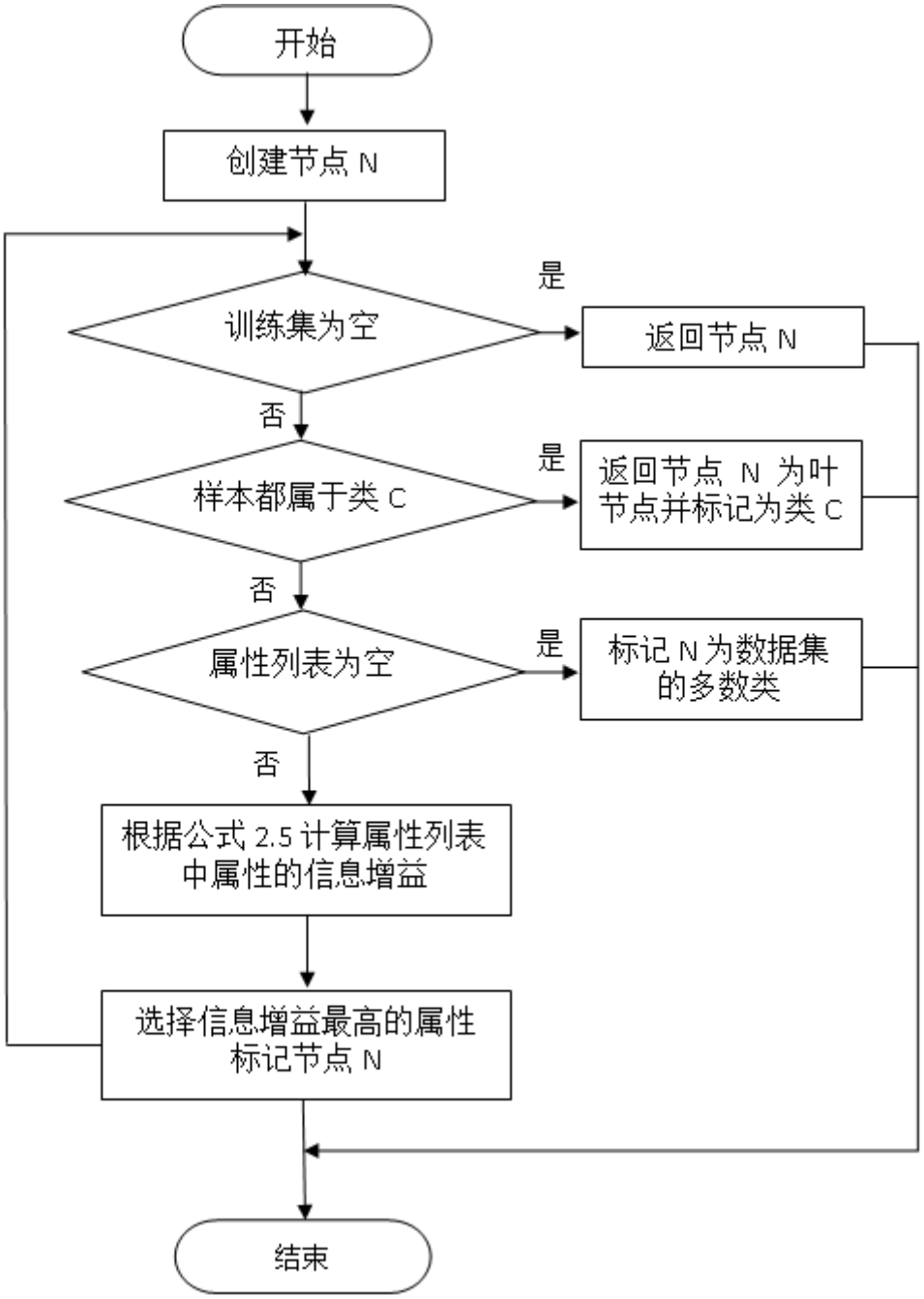


图 2-1 ID3 算法创建决策树流程图

2.3 C4.5 算法

C4.5 算法[11]是用于生成决策树的一种经典算法，是 ID3 算法的一种延伸和忧化。因为根据 C4.5 算法建立的决策树可以进行分类预测，所以，C4.5 算法也被叫做统计分类器。C4.5 算法对 ID3 算法作了好几点改进，其中主要的改进有以下几个方面。

- 能够处理具有缺失属性值的训练数据；
- 能处理不同代价的属性；
- 构造决策树之后进行剪枝；
- 能处理离散型和连续型的属性类型，即可以将连续型的属性进行离散化处理。

同时，决策树每生成一个节点时，C4.5 算法都选择数据中最有效的分裂属性来对某一个样本进行分类，将其分在一个类别中或其他另外的类别中。信息增益（熵的差）规范化是来自于选择分类数据的属性。选择拥有最高的规范化信息增益为当下的分裂属性的一个元素。然后，C4.5 算法再继续计算下一子集中最高规范化的信息增益。规范化的信息增益即为信息增益率。C4.5 算法根据每一个条件属性的信息增益率的大小选择当下的分裂属性，消除了信息增益用作属性选择度量时趋向于选择属性的值较多的属性从而影响分类所带来的困扰。

2.3.1 C4.5 算法思想

C4.5 算法与 ID3 算法生成决策树的步骤过程基本相同，主要区别在于连续型属性和属性度量的计算中，ID3 算法不能处理连续型属性，而 C4.5 算法可以先离散化连续型属性，然后进行属性选择计算；属性度量计算时，ID3 算法利用信息增益进行属性选择计算，C4.5 算法则运用信息增益率计算。

信息增益率

信息增益率就是对信息增益进行了规范化，即 C4.5 算法思想运用信息增益率公式替换了 ID3 算法中的信息增益的计算思想。

信息增益的规范化用到了“分裂信息(split information)”的概念。

在训练集 T 中，公式（2.6）表达了属性 A 的分裂信息。

$$SplitInfo_A(T) = -\sum_{j=1}^{m} \frac{|T_j|}{|T|} \log_2 \frac{|T_j|}{|T|} \tag{2.6}$$

$SplitInfo_A(T)$：表示训练集 T 划分成与属性 A 相对应的 m 个输出中的 m 个分区所生成的信息。即每一个输出的元组数都是相对于 T 中元组的总数来确定的。

与信息增益不同的是，信息增益率是用来计量相同的划分所获得的信息。

公式（2.7）即为属性 A 的增益率的计算公式。

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(T)} \tag{2.7}$$

在用 C4.5 算法构造决策树时，信息增益率最大的属性即为当前节点的分裂属性，随着递归计算，被计算的属性的信息增益率会变得越来越小，到后期则选择相对比较大的信息增益率的条件属性作为分裂属性。

公式(2.7)和公式(2.6)结合 ID3 的计算公式(2.5)则可得到 C4.5 算法具体的计算公式。即为：

$$GainRatio(A)=\frac{Info(T)-Info_A(T)}{SplitInfo_A(T)} \tag{2.8}$$

2.3.2 C4.5 算法步骤

C4.5 算法处理数据集时，依其属性做成数据表，去除无关属性和相应数据（即制定规则），称为训练数据。该数据集必须是完整的数据，包括预测条件属性和最后分类结果属性，被称为训练数据集。

C4.5 算法生成决策树的过程步骤为：

Step1: 创建一个节点 N

Step2: IF 训练数据集为空，THEN 返回单个节点 N 作为空的叶子节点

Step3: IF 训练集中的所有样本都属于同一个类 C，THEN 返回节点 N 为叶节点并将该节点标记为类 C

Step4: IF 训练集的属性列表为空，THEN 返回 N 作为叶节点，并标记为数据集中样本多的类别

Step5: FOR EACH 属性列表中的属性 AttributeList

Step6: IF 属性是连续型的，THEN 对该属性进行离散化

Step7: 根据公式（2.8）计算属性列表中属性的信息增益率

Step8: 选择属性 AttributeList 中拥有最高的信息增益率的属性 A，并把节点 N 标记为属性 A

Step9: 删除属性列表 AttributeList 中的属性 A

Step10: FOR EACH 属性 A 的属性值 a，由节点 N 分出一个条件为 A=a 的分支，得到子树

Step11: 递归方式循环 Step3-10，得到初步决策树

Step12: 利用更大的训练数据集对决策树进行修剪（优化）

在进行决策树构造时，会根据数据集中的信息判断是否满足停止建树的条件，否则继续迭代。一般情况下，结束的条件主要有：属性列表为空；数据集中样本都已经归类；所剩样本都属于同一个类。满足其中一个条件便结束建树，得到初始的决策树。接着运用后剪枝的策略进行剪枝，简化决策树。

2.3.3 C4.5 算法流程图

C4.5 算法流程图如下：

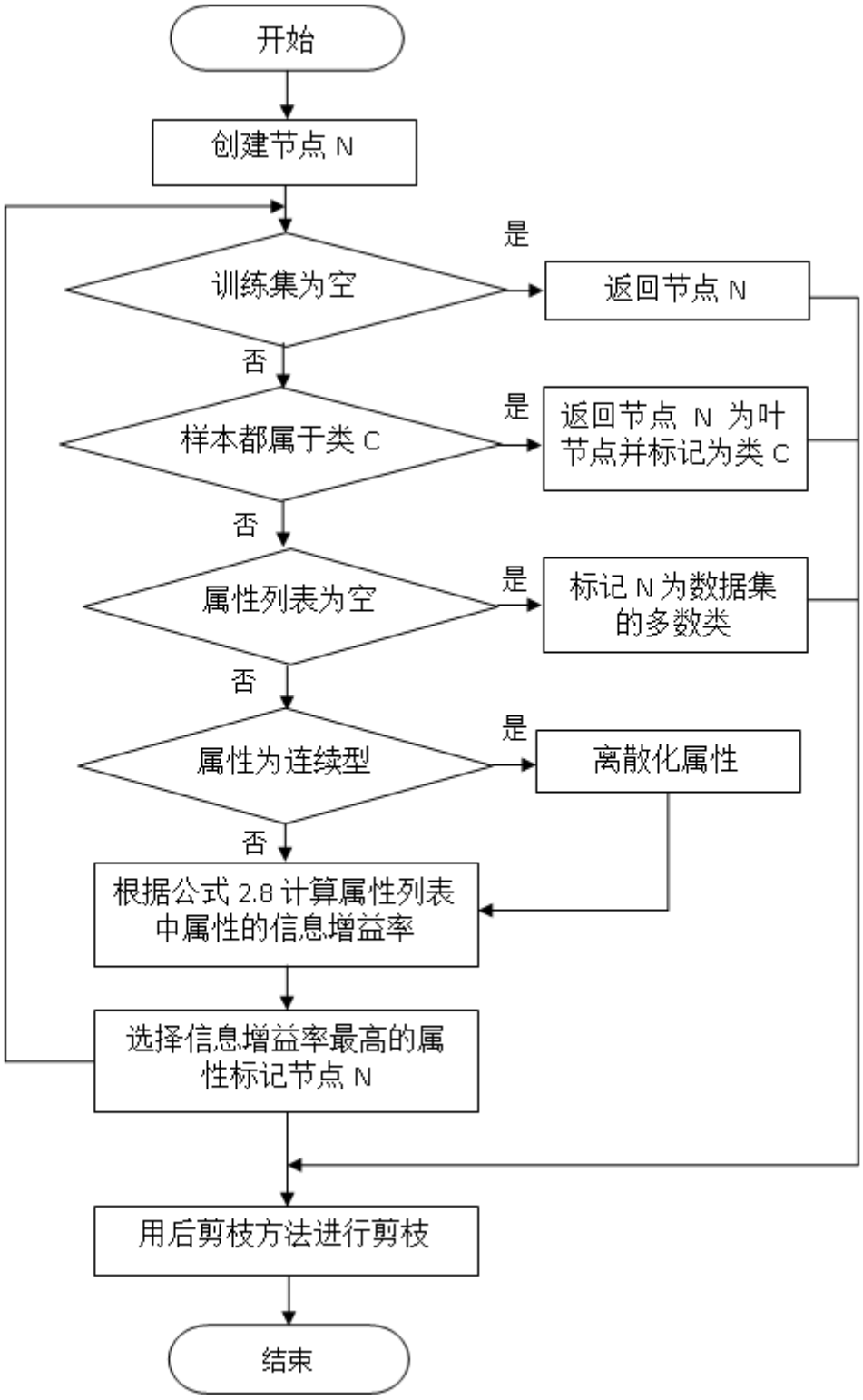


图 2-2 C4.5 算法流程图

2.3.4 C4.5 算法的优缺点

C4.5 算法优点

第一：可以处理数据不完整和连续型属性的数据集；

第二：C4.5 算法进行分类之后产生的分类预测准则比较容易理解；

第三：分类的正确率比较高；

第四：建模速度较快。

C4.5 算法缺点

第一：在建立决策树的步骤流程中，必须重复地对相应的数据集进行依次扫描和逐个排序，所以造成了算法的分类效率不高；

第二：C4.5 算法的计算公式涉及了大量的对数运算，计算机在进行计算时，会频繁地调用函数，增加了算法的时间开销；

第三：算法在选择分裂属性时没有考虑到条件属性间的相关性问题，只计算数据集中每一个属性与类属性之间的期望信息，有可能影响到属性选择的正确性；

第四：C4.5 算法尽管是 ID3 算法的改进，可以处理数据不完整和连续型属性的数据集，但是其处理数据集样式的宽度仍需提高，即还不能处理很多其他形式的数据集。

2.4 改进的 C4.5 算法

这里主要介绍最近相对较热门的两种 C4.5 优化改进的算法思想：基于粒子群优化算法改进的 C4.5 算法和基于模糊系统思想改进的 C4.5 算法。

2.4.1 基于粒子群算法的 C4.5 算法

基于粒子群算法改进的 C4.5 算法（PSOC4.5），以 C4.5 算法作为分类决策基础，用粒子群算法进行特征选择，这是与 C4.5 算法最大的区别。

基本粒子群算法

粒子群优化算法（Particle Swarm Optimization，PSO）[72]是一种基于群体协作原理的随机搜索的技术，受群体的社会行为的启发，如鸟群觅食或鱼群集训，获得有前景的位置，以实现一定的目标。在粒子群优化算法中，每个粒子都有一个位置，并且基于一个更新的速度移动。在一个种群中的每一个粒子都有一个由适应度函数计算的适应度值。基本 PSO 中粒子的主要特征是位置、速度和能力，与邻近点来交换信息时能够记住先前的位置，并有通过信息做出决定的能力。

粒子群优化算法的初始化是随机的粒子群，每个粒子有速度和位置这两个属性，第 i 个粒子的速度和位置可以表示为：$v_i=(v_{i1},v_{i2},\cdots,v_{id})$ 和 $x_i=(x_{i1},x_{i2},\cdots,x_{id})$。这些粒子根据迭代方式找到最优解，在每一次的求解过程中，粒子依据跟踪两个最优值（个体极值 p_i 和全局极值 p_g）来更新自己所在的位置。在获取这两个最优值后，粒子可按照以下公式计算速度和位置：

$$V_i(t+1)=w\times V_i(t)+c_1\times y\times(p_i-x_i)+c_2\times y\times(p_g-x_i) \tag{2.9}$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \tag{2.10}$$

w：是惯性权重参数；

y：是介于$[0,1]$的随机数；

c_1,c_2：表示学习因子，是两个自定义常数。

算法思想

PSOC4.5 算法思想以 C4.5 算法作为分类方法的基础，用粒子群优化算法（PSO）进行特征选择和适应度评估。特征选择是影响分类正确率的关键步骤，故用粒子群优

化算法优化特征选择，从而改进 C4.5 算法的分类正确率。

PSOC4.5 算法处理数据集时，进行特征选择之前的数据预处理都沿用了 C4.5 算法的思想，数据预处理完毕之后，用粒子群优化算法进行处理，优化特征选择；然后再用 C4.5 算法的计算公式对被选中的特征进行计算分类；迭代方式得到初始的决策树，最后再用剪枝方法进行剪枝优化，防止过度拟合。

算法步骤

PSOC4.5 算法步骤描述如下：

Step1：初始化。就是将数据集进行粒化

Step2：选择一个粒子作为一个类

Step3：从群里选择具有相同索引的粒子

Step4：FOR EACH 粒子，评估每一个粒子的适应度函数

Step5：根据公式（2.9）和公式（2.10）计算每个粒子最新的速度和位置，并获得粒子两个新的极值

Step6：找出最优粒子，用 C4.5 算法的思想评估每一个被选中的粒子，并对其进行分类，直到最后一个粒子选择完毕

2.4.2 基于模糊算法的 C4.5 算法

基于模糊系统思想改进的 C4.5 算法又叫模糊决策树（FuzzyDT）[48]。FuzzyDT 集合了模糊系统的优点和 C4.5 算法的思想，可以处理不确定和不精确的变量，同时提高了规则的可解释性。

模糊逻辑系统

模糊逻辑推理系统有一些模型可以用作浏览或者论证的原理，例如人类推理过程的方法，这些模型包括 Tsukamoto，Mamdani 和 TSK（Takagi Sugeno Kang）。因为接近人类思考的方式和根据语言学规则推理的规则，模糊 Mamdani 广泛地被当作构建系统的推理过程的工具。主要通过四个阶段得到 Mamdani 模糊输出：模糊化阶段、模糊知识库的构建完成、解模糊化函数的应用和影响。

模糊逻辑有一个隶属函数，该函数表明了元素属于这个集合的程度。隶属函数可用图形化的形式表达模糊集合。有很多隶属函数图形可以被用来确定一个模糊集的隶属函数，如三角形隶属函数、梯形隶属函数和高斯隶属函数等。其中三角形隶属函数是最常用也最简单的一种隶属函数，其模糊化公式如下：

$$trigle(x;a,b,c)=\begin{cases}0 & x\le a\\ \dfrac{x-a}{b-a} & a\le x\le b\\ \dfrac{c-x}{c-b} & b\le x\le a\\ 0 & c\le x\end{cases} \tag{2.11}$$

进行模糊化时，先将数据划分几个区域。

x：表示在区域内的变量，确定定义域

a,b,c：表示区域的边界，确定曲线形状。参数 a 和 c 对应的是三角形下方左右两个点的值，而参数 b 则对应三角形上方顶点的值。

算法思想

FuzzyDT 算法思想沿用了 C4.5 算法决定特征重要性的方法（信息熵和信息增益率），同时，也运用归纳策略递归地构造分类决策树。不同的是，FuzzyDT 算法在推导决策树之前将连续型属性值定义为一系列的模糊集，用这种方法进行“离散化”。模糊化属性值，就是将属性值转换成相对应得自然语言变量，再根据自然语言变量进行分类计算。下图简单地描述了数据模糊化的简单示例过程：

NO.	A_{t1}	A_{t2}	A_{t3}	class
1	15	0.6	3.1	C_a
2	17	0.9	2.2	C_b
3	19	0.4	2.3	C_c
		…		
n	16	0.8	2.9	C_a

原始数据

NO.	A_{t1}	A_{t2}	A_{t3}	class
1	Small	Low	High	C_a
2	High	High	Low	C_b
3	High	Low	Low	C_c
		…		
n	Low	High	High	C_a

模糊化数据

图 2-3 简单的模糊化示例

该数据集中含有 n 个样本，3 个条件属性（ A_{t1},A_{t2},A_{t3} ）和 1 个类属性（Class），3 个类属性值（ C_a,C_b,C_c ）。上图第二个区域则是属性值模糊化的描述，最后是根据模糊化数据变量进行分类的决策树模型。

算法步骤

FuzzyDT 算法的步骤描述如下：

Step1：定义模糊数据集。将连续型属性按照数据域进行模糊粗糙化

Step2：确定一种隶属函数

Step3：用模糊集的语言标签代替训练集中的连续型属性值，得到具有最高兼容性的输入值

Step4：计算每一个属性的信息增益率，对训练样本进行分类。将每一个样本进行归类直到候选属性为空，或者训练样本集为空

Step5：运用后剪枝方法对决策树进行剪枝优化，类似于 C4.5 算法，用 25%的默认置信度进行修剪

2.5 本章小结

本章首先概述了决策树分类算法，通过对 ID3 算法的介绍，引出了 C4.5 算法。在了解了决策树分类算法的属性度量和剪枝方法之后，详细地描述了 ID3 算法，介绍了信息熵和信息增益的概念，以及 ID3 算法的思想和流程；接着，介绍了 C4.5 算法与 ID3 算法的关系，详细描述了 C4.5 算法的思想、步骤和流程图，以及 C4.5 算法的优缺点等。最后，介绍了两种改进的 C4.5 算法：基于粒子群算法改进的 C4.5 和基于模糊算法改进的 C4.5 算法，简单的介绍了算法的思想和步骤。

第三章　基于 GINI 指数均值的 C4.5 优化算法

C4.5 算法主要问题是对数运算多，分类正确率受属性间相关性影响。属性选择顺序对于分类正确率的影响至关重要，为了提高属性选择正确率，近年来很多学者提出 PSOC4.5 算法，如 T.R.Sivapriya[55]等，该算法用粒子群优化算法（PSO）进行特征选择，然后再用 C4.5 算法的分类基础思想进行分类。通过优化特征选择来提高 C4.5 算法的分类预测正确率。还有人提出模糊 C4.5 算法，比如 Marcos E. Cintra 等人提出的 FuzzyDT 算法，将模糊算法与 C4.5 算法相结合，减少分类规则，同时提高分类正确率。这些改进算法的分类正确率有了相对的提高。本文提出了一种基于 GINI 指数均值的 C4.5 优化算法 GC4.5，该算法运用泰勒级数简化对数运算，减少算法计算时间；运用 GINI 指数均值消除属性间相关性影响，提高属性选择正确率。

3.1 泰勒级数

在数学中，泰勒级数（Taylor series）[73]是用无限项的连加式，即把一个函数表示为级数形式，相加的项可以根据这个函数在某一个点上的导数来求取。其定义如下：一个在 a（a 是实数或者复数）邻域上的无穷可微的实变函数或者复变函数 $f(x)$，它的泰勒级数具有如下的幂级数：

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n \tag{3.1}$$

$n!$ ：表示 n 的阶乘；

$f^{(n)}(a)$：表示函数 f 在点 a 处的 n 阶导数。

如果 $a=0$，则上式的泰勒级数也被称为麦克劳伦级数。即：

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(0)}{n!}x^n \tag{3.2}$$

综上所述，当 $f(x)$ 为自然对数时，即

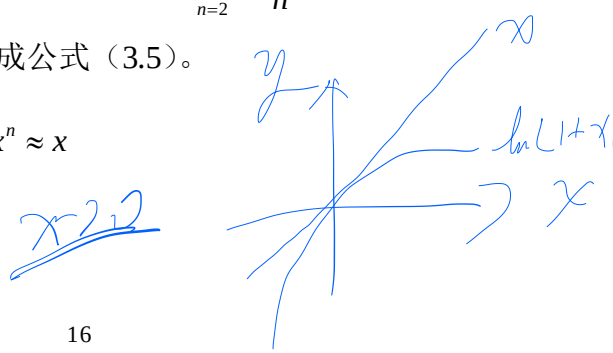
$$f(x)=\ln(1+x) \tag{3.3}$$

$f(x)$ 的泰勒级数为：

$$\ln(1+x)=\sum_{n=0}^{\infty}\frac{(-1)^{n+1}}{n!}x^n \tag{3.4}$$

根据等价无穷小原理，当 x 的值无穷小时，可得 $\sum_{n=2}^{\infty}\frac{(-1)^{n+1}}{n}x^n$ 的值也趋近于 0。则当 x 无穷小时，公式（3.4）又可以简化成公式（3.5）。

$$\ln(1+x)=\sum_{n=0}^{\infty}\frac{(-1)^{n+1}}{n!}x^n\approx x \tag{3.5}$$



3.2 属性相关性和 GINI 指数原理

属性相关性

在一个数据集的所有属性（包括类属性和条件属性）中，并不是所有的属性的相应属性值都拥有一样的描述某一事物或者划分某一事物的所需的信息量。有的条件属性和类属性拥有很强的相关性，这个条件属性的相应值所包含的信息对数据集的分类预测结果又有很大的影响；而有的条件属性与类属性间的相关性不大，则其相应值所包含的信息对分类预测结果基本没影响。这就说明：一个条件属性与类属性之间的相关性的强弱决定了该属性对数据集的最终的分类预测结果的影响的严重程度。

若将上面的条件属性与类属性之间的关系用相关性来表示，则根据一个条件属性与类属性之间相关性的强弱，可以将数据集中的所有属性分为强相关的；弱相关的和不相关的这三种属性。

• 强相关属性，是指该属性的值对数据集的分类有很大的影响，一般为最先的分裂属性；

• 弱相关属性，一般对分类结果影响不大，信息增益率的值也很小；

• 不相关属性，是指这些属性与数据集的分类毫无影响，与类属性没有关联的。

在数据挖掘过程中，进行属性选择时，一般只考虑条件属性与类属性间相关性，而往往忽略了条件属性之间的相关性。一个条件属性与其它的条件属性之间的关联性强弱，也表示了这个条件属性与其它的条件属性之间相互关联的深浅程度（一般用条件属性间的冗余度表示）。正是这些被忽略的条件属性间的相互依赖（属性间冗余度），可能造成不正确的属性选择，从而影响分类的正确率。

GINI 指数

GINI 指数[74]是一种不纯度函数（Impurity Function），已经用于一些分类算法，它是一种适合可分类型和数值型的分类的方法。不纯度函数可以计算数据集中的样本“纯度”。若在一个数据集中，样本相对集中地分布在某一个类中，那么这个数据集的“纯度”就会比较高。反之这个数据集的“纯度”就小。也就是说，将一个数据集依据相应的取值范围进行拆分，其“纯度”将减增大。其定义为：

如 2.2.2 中提到的数据集 T 包含 n 个类别的样本集，则其 GINI 指数可以用公式（3.6）表示。

$$gini(T)=1-\sum_{i=1}^{n}\left(\frac{|TC_i|}{|T|}\right)^2 \tag{3.6}$$

C_i（i=1,2，……，n）：为类标记属性具有 n 个不同的值；

TC_i：训练集 T 中属于 C_i 类的样本集合；。

$|T|$：是训练集 T 的样本个数；

$|TC_i|$：是训练集 TC_i 中的样本个数。

如果某个属性 A 有 m 个取值，根据该属性的取值把集合 T 分为 m 个部分，$|T_j|(j=1,2,\ldots,m)$为属性 A 取第 j 个值时所对应的数据集样本个数，属性 A 的 GINI 指数

则用公式（3.7）计算：

$$giniSplit_A(T)=\sum_{j=1}^{m}\left[\frac{|T_j|}{|T|}gini(T_j)\right] \tag{3.7}$$

$|T_j|$：是训练集 T 中按属性 A 的第 j 个属性值划分的样本个数；

m：表示属性 A 属性个数值。

在决策树分类选择根节点属性时，信息增益率越高，说明该属性越纯净，则首先选择该属性进行分割。与此相反的，GINI 指数主要是以衡量数据的划分或训练样本集 T 的不纯度为主，如果计算的 GINI 指数的值较大，就表示数据集的不纯度比较高。即样本属于同一个类的概率就越高。这是条件属性与类属性之间的关系。

3.3 算法描述

GC4.5 算法与 C4.5 算法最大区别就是简化了对数运算，并在信息增益率计算公式中加入了 GINI 指数。

简化计算

根据泰勒级数简化原理和 C4.5 算法的信息增益率计算公式（2.8），可以得到出 C4.5 算法简化后的信息增益率运算公式：

$$GainRatio(A)=\frac{Gain(A)}{SplitInfo_A(T)}=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}} \tag{3.8}$$

但是，由于在化简的过程中会带来一定的误差，所以不能直接使用化简得到的公式(22)来计算数据集的信息增益率，为了弥补化简产生的误差，在公式（3.8）中加入数据集中每个属性的属性值个数 M[75]：

$$GainRatioM(A)=\frac{Gain(A)}{SplitInfo_A(T)}\times M \tag{3.9}$$

结合公式（3.8）和（3.9）可以得到简化后的信息增益率的最终计算公式：

$$GainRatioM(A)=\frac{Gain(A)}{SplitInfo_A(T)}\times M=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}} \tag{3.10}$$

计算机在进行对数运算时需要进行函数调用，特别是在涉及大量对数运算的 C4.5 算法中会大大增加时间开销。而在简化后，运算公式由原来的对数运算转换为加减乘除的基本运算，在时间复杂度相同的情况下，简化后的算法省去了函数调用的时间开销，从一定程度上来讲，提高了算法的效率。

消除属性相关性

根据 GINI 指数原理可知，在条件属性之间，如果一个条件属性与其他放入条件属性之间的信息增益率越大，则该条件属性与其他条件属性之间的关联性就越大，即冗

余度就越大；相反的，如果一个条件属性与其他条件属性之间的 GINI 指数越小，则它们之间的冗余度就越大。公式（3.11）计算的是一个属性(以属性 A 为例)与其他属性间的 GINI 指数之和。

$$Sum\_giniSplit_{A_F}(T)=\sum_{i=1}^{s}\sum_{j=1}^{x}\left[\frac{|T_{ij}|}{|T|}gini(T_{ij})\right] \tag{3.11}$$

$A_F T$：表示不包含类属性和属性 A 的属性集；

S：表示除开类属性和属性 A 以外的属性个数；

X：是一个变量，表示每个条件属性（除属性 A 外）的属性值个数，这个变量会随着 i 的变化而变化；

$|T_{ij}|$：表示第 i 个条件属性（除属性 A 外）取第 j 个属性值时样本的个数；

$gini(T_{ij})$：表示每个条件属性（除属性 A 外）关于属性 A 的 GINI 指数，其计算如式（3.12）所示。

$$gini(T_{ij})=1-\sum_{k=1}^{m}\left(\frac{|TA_{ijk}|}{|T_{ij}|}\right)^2 \tag{3.12}$$

$|TA_{ijk}|$： 表示第 i 个条件属性（除属性 A 外）取第 j 个属性值的同时，属性 A 为第 k 个值时的样本个数。

那么属性 A 与其他属性（不包括类属性）之间的 GINI 指数之和的平均值的计算公式为（3.13）：

$$\overline{Sum\_giniSplit_{A_F}(T)}=\frac{\sum_{i=1}^{s}\sum_{j=1}^{x}\left[\frac{|T_{ij}|}{|T|}gini(T_{ij})\right]}{s} \tag{3.13}$$

为此，在计算属性信息增益率时，加入该属性与其他属性（不包括类属性）之间的 GINI 指数的平均值来提高属性选择的正确性。即属性 A 的信息增益率的计算在运用泰勒级数和等价无穷小的原理进行简化后，属性 A 的分裂信息减去该属性与其他条件属性间的 GINI 指数的均值，作为新的分裂信息，计算出信息增益率。如式（3.14）所示。

$$GainRatioMG(A)=\frac{Gain(A)\times M}{SplitInfo_A(T)-\overline{Sum\_giniSplit_{A_F}(T)}}=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}-\overline{Sum\_giniSplit_{A_F}(T)}}\times M \tag{3.14}$$

根据公式（3.14）可以看出，如果属性 A 与其他条件属性间的相关性越小，即冗余度越小，那么属性 A 与其他条件属性间的 GINI 指数平均值就越大，即 $\overline{Sum\_giniSplit_{A_F}(T)}$ 的值就越大，相反的，$SplitInfo_A(T)-\overline{Sum\_giniSplit_{A_F}(T)}$ 的值就越小，则属性 A 的信息增益率越大。因此消除了条件属性间的冗余度对属性选择准确性的影响，提高了算法的分类准确性。

3.4 算法流程图

GC4.5 算法的流程图如下：

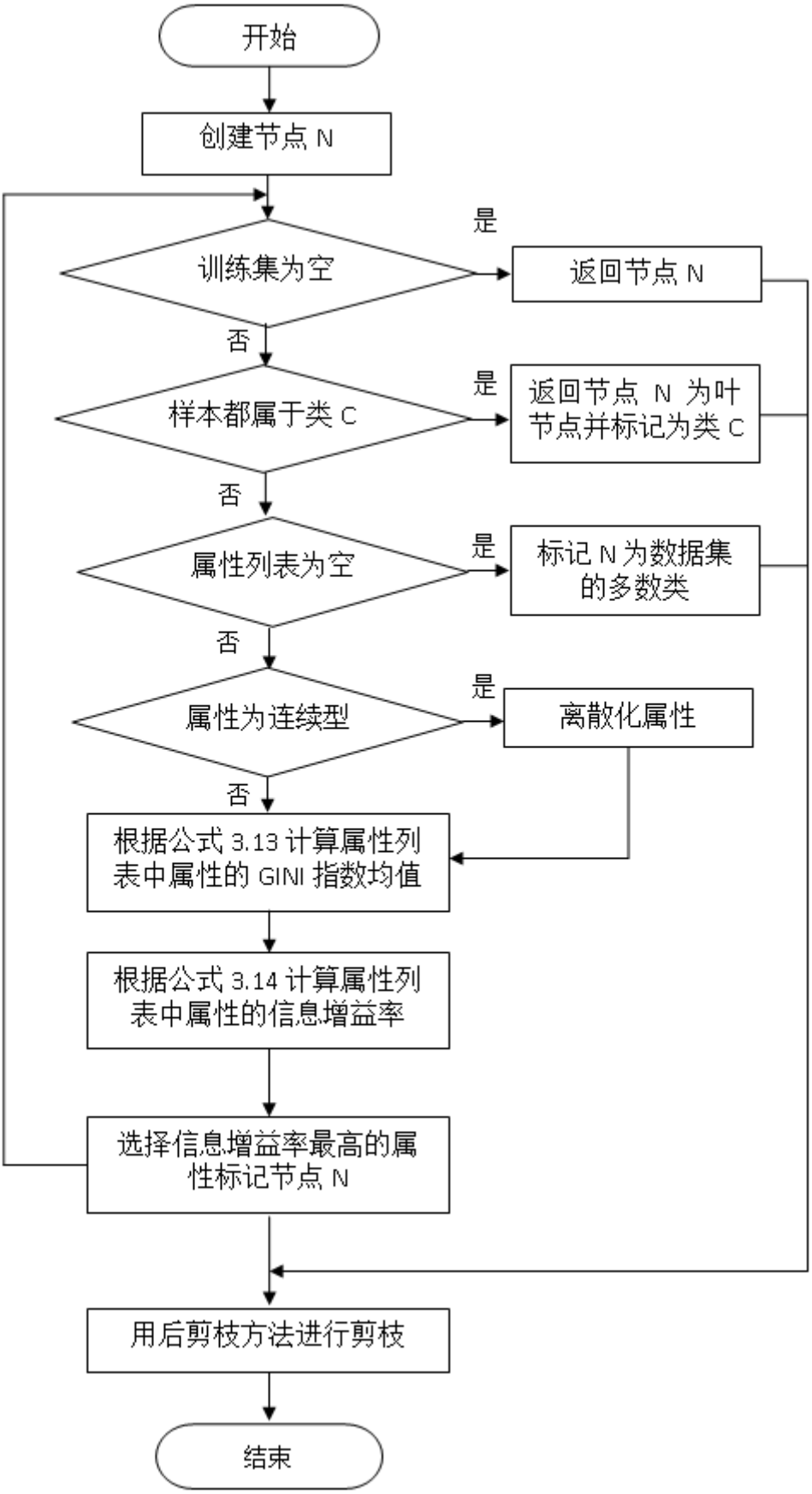


图 3-1 GC4.5 算法流程图

3.5 算法伪代码

GC4.5 算法的伪代码如下：

Begin
创建节点 N
For Each 属性列表 AttributeList 中的属性
  If 训练样本为空 Then
    返回节点 N 为叶子节点
  Else If 属性列表为空 Then
    返回节点 N
  Else If 样本均属于同一类 C Then
    返回节点 N 为类 C
  Else If 属性为连续型 Then
    离散化属性
  Else
    用公式 3.13 计算属性的 $\overline{Sum\_giniSplit(A_F T)}$
    用公式 3.14 计算属性的信息增益率
    选择列表中信息增益率高的条件属性作为当下的节点 N
End
用后剪枝方法进行剪枝
End

其中，计算 $\overline{Sum\_giniSplit(A_F T)}$ 的伪代码具体描述为：

Function：AGSumGiniSplit();
Begin
  For 所有的属性 R（Attribute(p)）Do{
    For 所有的属性 R（Attribute(i)）Do{
      If 属性 Attribute(*i*)不等于属性 Attribute(p)，并且不等于类属性，则
    Begin
      For 属性 Attribute(*i*)的每一个值（Attribute(*i*).numValues(*j*)）Do{
        For 属性 Attribute(*p*)的每一个值（Attribute(*p*).numValues(*k*)）Do
          计算 gini(T_{ij})的值赋给 G*i*；
          计算出属性 Attribute(*i*)与其他条件属性 Attribute(*p*)的 Gini 指数，
          赋给 GiS;}
      End
    计算出属性 Attribute(*p*)与其他条件属性 Attribute(*i*)的 Gini 指数之和，
    赋给 SumG*i*S*p*;}
    将 SumGiSp 的均值依次赋给(SG*j*/*j*=1,2...m);}
  返回 SG*j*；

End

3.6 实验与分析

3.6.1 实验设计

通过 UCI 数据集进行实验对本章提出 GC4.5 算法性能进行测定。将 GC4.5 算法与 C4.5 算法、PSOC4.5.算法[55]、FuzzyDT 算法[48]进行对比实验，对比这些算法的建模速度和分类正确率。

实验是在 Window10 操作系统下进行的，系统软硬件环境为：内存 2G，Celeron(R) Dual-Core CPU 1.80GHz 上进行的，配置的实验环境还有怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），Java 的 JDK、JRE 和 Eclipse。这些环境配置的版本信息如表所示：

表 3-1 实验配置信息

实验配置	版本
WEKA	3.7.11
JDK	jdk1.7.0_51
Eclipse	4.4.0

采用 16 个 UCI 数据集作为实验数据集，数据集的具体信息如表 3-2 所示。数据集信息包括样本数(Examples)、条件属性个数(Features)和类属性值个数(Classes)，其中条件属性中还包括连续型(c)和离散型(d)的个数。

表 3-2 UCI 数据集

No.	Dataset	Examples	Features(c d)	Classes
1	Breast	699	9 (9 0)	2
2	Letter	20000	16 (16 0)	26
3	Credit	1000	20 (7 13)	2
4	Diabetes	768	8 (8 0)	2
5	Gamma	19020	10 (10 0)	2
6	Lymph	148	18(3 15)	4
7	Haberman	306	3(3 0)	2
8	Heart	270	13 (13 0)	2
9	Anneal	898	38 (6 32)	6
10	Liver	345	6 (6 0)	2
11	Waveform	5000	40 (40 0)	3
12	Soybean	683	35 (0 35)	19
13	Spam	4601	57 (57 0)	2
14	Car	1728	6 (0 6)	4
15	Vehicle	846	18 (18 0)	4
16	Segment	2310	19 (19 0)	7

No.:表示数据集的序号。

实验主要利用 Eclipse 平台重新编译运行 WEKA 进行，具体的实验步骤为：

Step1：从 WEKA 官方网站下载 WEKA 程序包。把程序包进行解压得到 weka-src.jar 源文件，然后把源代码导入到一个 JAVA 开发的工具中(Eclipse)；

Step2：为了避免和 WEKA 平台中已有的后台算法发生冲突，重新再创建一个 JAVA 包，把重新编写的新的算法存放在这个包中；

Step3：在新创建的 JAVA 包内新建一个 java 类，建好后双击这个类，接着开始编写新的或改进的数据挖掘算法的代码；

Step4：修改 weka.gui 包中的 GenericObjectEditor.props 文件。具体操作为：需要在 GenericObjectEditor.props 中的# Lists the Classifiers I want to choose from 段后添加 weka.classifiers.JAVA 包名.类名；

Step5：在 Eclipse 中重新编译整个 WEKA 的算法项目了，在选择需要运行的主类的时候选 weka.gui.GUIChooser 这个类。接着在 WEKA 界面中用上面的 16 个数据集进行改进的算法实验。

3.6.2 实验结果与分析

实验设置为 10 次交叉验证和 25%置信度进行剪枝，每个数据集运行 10 次取平均值得到实验数据。

首先是算法的建模时间，这里的时间是取 10 次实验的平均建模时间，如下表所示：

表 3-3 建模时间

No.	GC4.5	C4.5	PSOC4.5	FuzzyDT
1	**0.016**	0.018	0.020	0.022
2	**5.747**	6.736	6.714	6.170
3	0.038	0.042	**0.020**	**0.020**
4	0.020	0.023	**0.019**	0.020
5	1.636	2.142	0.756	**0.322**
6	0.001	0.001	0.001	0.001
7	**0.004**	0.008	0.006	0.020
8	**0.001**	0.008	0.016	0.010
9	0.078	0.070	**0.020**	0.056
10	0.004	0.012	**0.001**	**0.001**
11	0.854	1.128	0.610	**0.354**
12	**0.032**	0.040	**0.032**	**0.032**
13	1.034	1.248	**0.660**	0.984
14	0.016	0.018	**0.004**	0.020
15	0.058	0.084	0.054	**0.034**
16	0.126	0.138	0.074	**0.066**

表中数据的单位为秒，表示建模时间。

根据表 3-3 的实验数据可知，PSOC4.5 算法和 FuzzyDT 算法的建模速度较快：在 16 个数据集中，有 7 个数据集的建模速度为 PSOC4.5 和 FuzzyDT 算法最快，只有 5 个数据集 GC4.5 算法的建模时间较短。

其中，第 6 个数据集（Lymph）所有算法的建模时间一样；第 3 个和第 10 个数据集（Credit、Liver）的建模时间是 PSOC4.5 和 FuzzyDT 算法一样，但都快于其他算法；而第 12 个数据集（Soybean）G4.5 算法、PSOC4.5 算法和 FuzzyDT 算法的建模时间一样，都优于 C4.5 算法。

与 C4.5 算法相比，GC4.5 算法的建模时间中，有 14 个数据集的建模时间低于 C4.5 算法的；有一个数据集（第 6 个数据集，Lymph）的建模时间一样，还有一个数据集（第 9 个数据集，Anneal）的建模时间比 C4.5 算法的长。

与 PSOC4.5 算法相比，GC4.5 算法的建模时间只有 4 个数据集的建模时间比 PSOC4.5 算法的短；有 10 个数据集的建模时间比 PSOC4.5 算法的长；还有两个数据集的建模时间一样，分别是第 6 个数据集（Lymph）、第 12 个数据集（Soybean）。

与 FuzzyDT 算法相比，GC4.5 算法的建模时间只有 5 个数据集的建模时间比 FuzzyDT 算法的短；有 8 个数据集的建模时间比 FuzzyDT 算法的长；还有 3 个数据集的建模时间一样，分别是第 4 个数据集（Diabetes）、第 6 个数据集（Lymph）、第 12 个数据集（Soybean）。

为了直观地查看对比结果，现将上表中的实验数据用调整系数进行处理比较，然后再根据调整后的数据进行比较分析。

转换后的数据如下：

表 3-4 处理后建模时间

No	GC4.5	C4.5	PSOC4.5	FuzzyDT	调整系数
1	16	18	20	22	0.001
2	5.747	6.736	6.714	6.17	1
3	3.8	4.2	2	2	0.01
4	2	2.3	1.9	2	0.01
5	16.36	21.42	7.56	3.22	0.1
6	10	10	10	10	0.0001
7	4	8	6	20	0.001
8	1	8	16	10	0.001
9	7.8	7	2	5.6	0.01
10	4	12	1	1	0.001
11	8.54	11.28	6.1	3.54	0.1
12	3.2	4	3.2	3.2	0.01
13	10.34	12.4	6.6	9.84	0.1
14	1.6	1.8	0.4	2	0.001
15	5.8	8.4	5.4	3.4	0.01
16	12.6	13.8	7.4	6.6	0.01

根据上表数据可以得到图 3-2 的对比折线图。

结合一行数据分析结果和下图，可以看出，GC4.5 算法、PSOC4.5 算法和 FuzzyDT 算法的建模时间都比 C4.5 算法的理想，PSOC4.5 算法和 FuzzyDT 算法的建模时间相当，略优于 GC4.5 算法。

虽然 GC4.5 算法略逊于 PSOC4.5 和 FuzzyDT 算法，但 GC4.5 算法的建模时间都优于 C4.5 算法， 所以从建模时间来说，GC4.5 算法的改进是可行的。

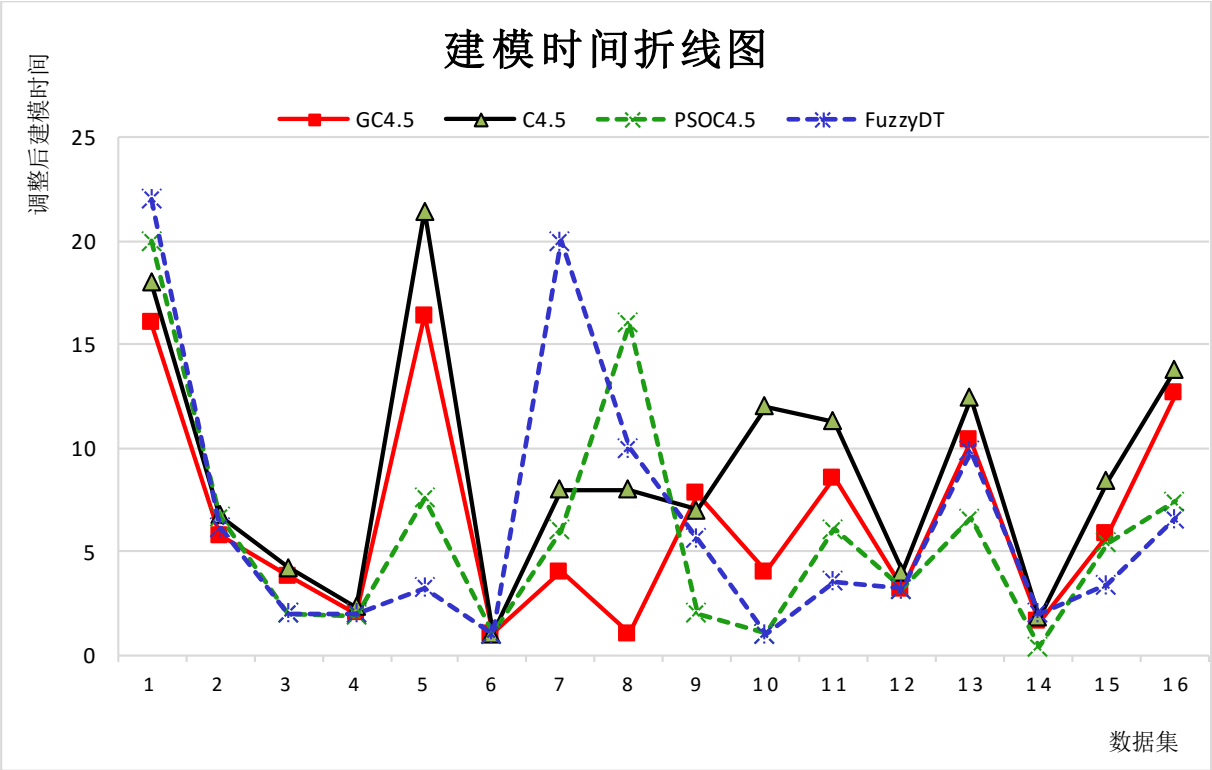


图 3-2 建模时间对比图

同样，取 10 次实验结果的平均分类正确率作为最终的结果数据，如下表所示：

表 3-5 分类正确率

No	GC4.5	C4.5	PSOC4.5	FuzzyDT
1	**0.9499**	0.9456	0.9456	0.9499
2	**0.8789**	**0.8798**	**0.8798**	0.4883
3	0.7220	0.7050	0.7310	**0.8519**
4	0.7396	0.7383	**0.7487**	0.7389
5	0.8504	**0.8506**	0.8280	0.8018
6	**0.8041**	0.7703	0.7568	0.7724
7	0.7288	0.7190	0.7255	**0.7293**
8	**0.8074**	0.7667	0.7852	0.7453
9	**0.9866**	0.9844	0.9643	0.9426
10	0.6725	**0.6870**	0.6086	0.6081
11	**0.7670**	0.7508	0.7632	0.7404
12	**0.9209**	0.9150	0.9165	0.8971
13	0.9289	0.9298	**0.9337**	0.7102
14	**0.9236**	**0.9236**	0.7002	0.9201
15	**0.7411**	0.7246	0.7163	0.6280
16	0.9662	**0.9693**	0.9680	0.8947

表中的数据为正确率的小数形式，结果保留到小数点后 4 位。

根据表 3-5 实验结果数据可知，在 16 个数据集中，有 9 个数据集的分类正确率属 GC4.5 算法的较优，5 个数据集分类正确率属 C4.5 算法的较高，PSOC4.5 算法占 3 个，而 FuzzyDT 算法仅有 2 个数据集的分类正确率较高。

GC4.5 算法相较于 C4.5 算法：有 10 个数据集的正确率高于 C4.5 算法；2 个数据集的正确率一样；4 个数据集的正确率低于 C4.5 算法。

GC4.5 算法相较于 PSOC4.5 算法：有 10 个数据集的正确率高于 PSOC4.5 算法；1 个数据集的正确率一样；5 个数据集的正确率低于 PSOC4.5 算法。

GC4.5 算法相较于 FuzzyDT 算法：有 14 个数据集的正确率高于 FuzzyDT 算法；2 个数据集的正确率低于 FuzzyDT 算法。

为了直观地查看对比结果，用折线图 3-2 来展现实验结果数据。

下图中，横坐标：表示 16 个数据集；

纵坐标：表示分类正确率（百分比的正确率，省去了百分号）

综合实验结果图表可知，从综合性能来说，本章提出的 GC4.5 算法是对于 C4.5 算法的成功改进。虽然建模速度比不上 PSOC4.5 算法和 FuzzyDT 算法，但是分类正确率上却有较大的优势。

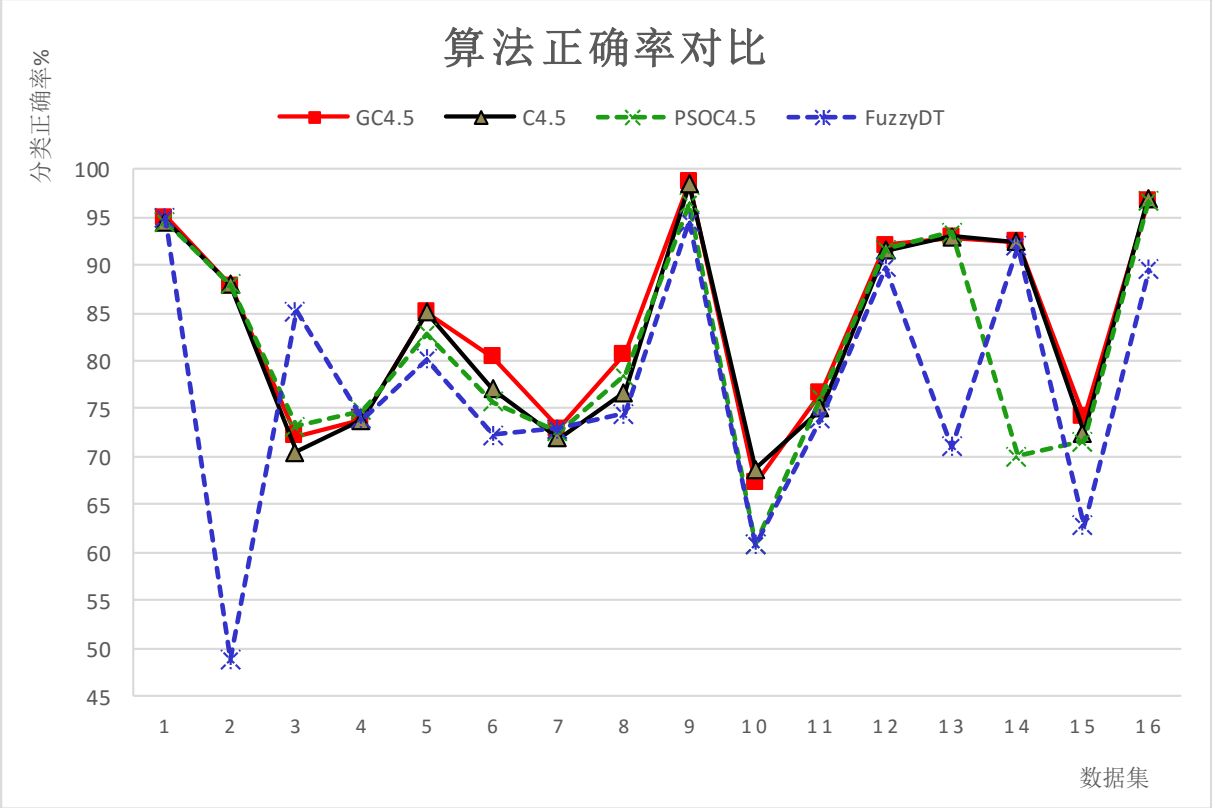


图 3-3 分类正确率对比图

3.7 本章小结

本章对 C4.5 算法进行了优化改进，提出了一种基于 GINI 指数均值的 C4.5 优化算法(GC4.5)，首先，根据泰勒级数和等价无穷小原理简化对数运算，减少函数调用的时间开销。然后，在信息增益率计算公式中加入属性的 GINI 指数均值，消除了属性间相关性对特征选择的影响。本章详细介绍了 GC4.5 算法思想，给出算法的流程图和 GINI 指数均值计算的伪代码，通过大量的 UCI 数据集对改进算法的建模速度和分类正确率进行测试，实验结果表明，从整体上，本章提出的改进算法建模速度快于 C4.5 算法而略逊于 PSOC4.5 算法和 FuzzyDT 算法，分类正确率高于现有的一些 C4.5 优化算法。

第四章 基于属性依赖度计算和 PCA 的 C4.5 优化算法

2013 年 Marcos E. Cintra 等人[48]提出了一种基于模糊算法的 C4.5 改进算法（FuzzyDT 算法）。该算法采用了模糊算法中变量模糊化的思想，将属性值进行模糊化，用自然语言表示具体的精确数值，意在优化属性选择和提高分类正确率，同时减少分类产生的规则。本章从这个角度出发，并在简化对数运算的基础上，提出了基于属性依赖度计算和 PCA 的 C4.5 优化算法（RPC4.5 算法），该算法根据属性依赖度计算原理删除无关属性，减少决策树分支；然后运用 PCA 算法的压缩原理简化数据集，消除属性相关性影响。

4.1 属性依赖度计算原理

属性依赖度是属性的一种重要性的衡量标准。依赖度一般是粗糙集理论中广泛应用的一种基本的概念，也有人提出以包含度的理论来描述依赖度的定义。在这里集合粗糙集和概率统计的原理，得到一种属性重要性的衡量标准，这种属性重要性的衡量标准表示了条件属性与类属性间的关联性。其定义是：

定义 1[76]设$U=\{T,A,B,C\}$是一个决策表，$|T|=M$，$T/ind(C)=C^*=\{Y_1,Y_2,\cdots,Y_m\}$，$T/ind(A)=A^*=\{X_1,X_2,\cdots,X_n\}$，定义知识 C 相对于知识 A 的依赖度为公式(4.1)：

$$\gamma(C^*\mid A^*)=\frac{1}{M(m-1)}\sum_{i=1}^{n}\sum_{j=1}^{m}\left\{\left[\frac{\left(P_{ji}-\frac{|X_i||Y_j|}{M}\right)^2}{\frac{|X_i||Y_j|}{M}}\right]\right\} \tag{4.1}$$

P_{ji}：是知识 A 的第$i\left(i\in\{1,2,\cdots,n\}\right)$个等价类在知识 C 的第$j\left(j\in\{1,2,\cdots,m\}\right)$个等价类中的样本个数；

$|X_i|$：表示知识 A 的第$i\left(i\in\{1,2,\cdots,n\}\right)$个等价类；

$|Y_j|$：分别知识 C 的第$j\left(j\in\{1,2,\cdots,m\}\right)$个等价类中所具有的元素个数。

如果知识 C 的等价类个数为 1，则规定$\gamma(C^*\mid A^*)=1$。

当$\gamma(C^*\mid A^*)=1$时，则称知识 C 完全依赖于知识 A；

当$0<\gamma(C^*\mid A^*)=k<1$时，称知识 C 是 K 度依赖于知识 A；

当$\gamma(C^*\mid A^*)=0$时，则称知识 C 完全独立于知识 A。

根据上面的定义可以得到一下定理。

定理 1[76] 知识 C 相对于知识 A 的依赖度$\gamma(C^*\mid A^*)$满足以下条件：

⑴ $0\le\gamma(C^*\mid A^*)\le 1$；

⑵当 $ind(A)\subseteq ind(C)$ 时，有 $\gamma(C^*\mid A^*)=1$；

⑶ $\forall A,D,C\subseteq B$，当 $ind(A)\subseteq ind(D)\subseteq ind(C)$ 时，有 $\gamma(A^*\mid C^*)\le\gamma(A^*\mid D^*)$。

根据属性依赖度的定义和定理，推导出在数据集中类属性相对于条件属性的依赖度，从而依据属性依赖度的计算删除无关属性。公式（4.2）为属性依赖度的具体计算公式：

$$\gamma(C\mid A)=\frac{1}{|T|(m-1)}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\left(\left|T_{ciAj}\right|-\frac{|T_{ci}|\left|T_{Aj}\right|}{|T|}\right)^2}{\frac{|T_{ci}|\left|T_{Aj}\right|}{|T|}} \tag{4.2}$$

$\gamma(C\mid A)$：表示的是类属性 C 对属性 A 的依赖度。

$\left|T_{ciAj}\right|$：是指属性 A 取第 j 个值时属于类 C_i 的样本个数。该式的取值范围为 0 到 1 之间。

同理，如果类属性值的个数为 1，则 $\gamma(C\mid A)$ 的值为 1，表示类属性完全依赖于属性 A；

当 $\gamma(C\mid A)$ 的值等于 0 时，表示属性 A 与类属性完全独立；

当 $\gamma(C\mid A)$ 为具体值 k(0<k<1)时，表类属性是 k 度依赖于属性 A。

根据公式（4.2）计算出类属性对各条件属性之间的依赖度，删除依赖度的值为 0 或者接近 0 的属性，从而优化了决策树和减少了算法的计算时间。

4.2 PCA 算法

算法描述

PCA[77]是 Principal Component Analysis 的缩写，被译为主成分分析法。顾名思义，主成分分析的主要目标就是确定一个最有意义的主成分来重新表达一个数据集。它运用线性代数的分析解，揭示了复杂数据集底层背后简单的结构。通过这个主成分可以过滤数据集中的一些噪音，从而挖掘出其中隐藏的结构。使原本复杂的数据简单化，而又不丢失数据中的重要信息。

PCA 最主要的好处就是量化地描述了可变数据集的每一个维度的重要性。每个主成分方差的测量都提供了一种比较每一维度相对重要性的方法。一个隐式期望的本质思想就是利用该方差沿着主成分少的方向（即：不足测量类型的数目），从而为完整数据集提供了一个合理的描述。PCA 算法通过线性变换和计算协方差来确定数据集中的主要数据元素和结构，其中协方差是两个变量之间线性关系程度的度量。

算法步骤

PCA 算法降维的过程概述为：

Step1：将原始的数据集结构“打乱”，重新整理成 $m\times n$ 的矩阵形式，其中 m 是测量类型（即属性）的数量，n 是样本（元组）数；

Step2：将每一维上的数值减去其相应维度上的平均值，得到一个调整后的新矩阵；

Step3：计算新矩阵的协方差矩阵，然后算出其特征值和特征向量（用 QR 分解法中的豪斯赫德方法来具体实现）；

Step4：对特征值进行排序，将特征值较大的向量正交化，然后找出其主要的数据元素，得到新的数据集。

PCA 算法作用

在 C4.5 算法中加入 PCA 算法具有如下几个意义：

首先，避免过度拟合。数据进行 supervised leaning（监督学习，例如 C4.5 分类算法）时，由于模型复杂，容易出现过度拟合，而经过 PCA 算法的处理，可以防止过度拟合。

其次，克服了由于属性间相关性的问题可能造成的属性选择不准确的缺陷。数据经过 PCA 算法处理之后输出的属性组合的主成分间是相互独立的，解决了相关性的问题，保证了属性选择的准确性。同时也就确保了分类的正确率。

最后，减少计算量，提高效率。数据集经过 PCA 算法对数据集进行特征选取，保留有效属性，减少属性数目，从而达到减少计算量，提高效率的目的。

4.3 算法描述

RPC4.5 算法的思想与 C4.5 算法最大的区别就是算法在进行属性选择之前，先进行了属性依赖度计算（公式（4.2））和 PCA 算法的处理。首先，根据公式（4.2）的计算结果进行属性筛选；然后，用 PCA 算法的压缩原理进行处理，优化特征选择，消除属性相关性影响。PCA 算法压缩原理的主要计算方式是特征值和特征向量的计算，具体的计算公式为：

$$y=\left(\mathrm{I}-2\mu\mu^{\mathrm{T}}\right)x=\mathrm{H}x \tag{4.3}$$

上式当且仅当$\|x\|_2=\|y\|_2$且$y^{\mathrm{T}}x$为实数时成立，称为 x 与 y 的 Householder 变换定理。

因为，$\|\alpha\|_2=\sqrt{(\alpha,\ \alpha)}$称为向量$\alpha$的长度或模或范数。而当$\|\alpha\|_2$=1时，则称$\alpha$为单位向量。所以：

上式中，$\|\mu\|_2$=1，表示单位向量；

$\mathrm{H}=\mathrm{I}-2\mu\mu^{\mathrm{T}}$：称为 Householder 矩阵。

根据公式（4.3）的计算，得到对称三对角矩阵，然后求相应的特征值和特征向量。

经由 PCA 算法压缩原理处理后输出的数据集用一下公式进行计算：

$$GainRatioMRP_{(A)}=\frac{Gain_{(A)}}{SplitInfo_A(T)}\times M=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times\left(|T|-|TC_i|\right)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times\left(|T_j|-|TC_{ij}|\right)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times\left(|T|-|T_j|\right)}{|T|}}\times M \tag{4.4}$$

优化特征选择之后，按照公式（4.4）进行计算，消除了对数运算，在提高分类正确率的同时缩短了计算时间，提高了建模速度。

4.4 算法流程图

RPC4.5 算法的流程图如下：

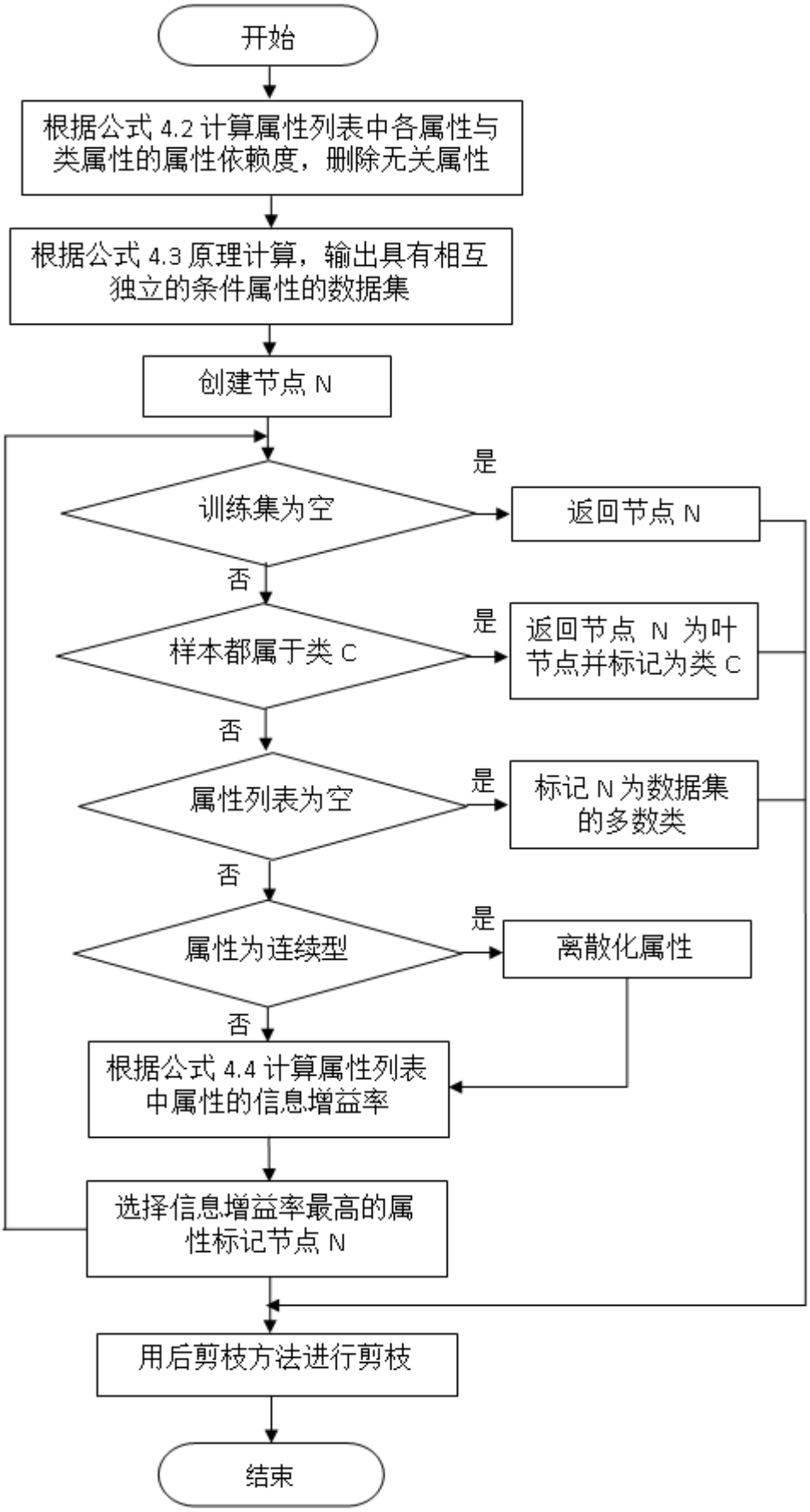


图 4-1 RPC4.5 算法流程图

4.5 算法伪代码

RPC4.5 算法的伪代码如下：

Begin

Do While 属性列表 AttributeList 不为空

用公式（4.2）计算属性依赖度 γ_i

If $\gamma_i = 0$ Then 删除第 i 个属性

End Do

For 属性 i = 1：m(属性个数)

For 样本 j = 1：n（样本个数）

根据公式（4.3）计算，优化训练集

End

End

创建节点 N

For Each 属性列表 AttributeList 中的属性

If 训练样本为空 Then

返回节点 N 为叶子节点

Else If 属性列表为空 Then

返回节点 N

Else If 样本均属于同一类 C Then

返回节点 N 为类 C

Else If 属性为连续型 Then

离散化属性

Else

用公式 (4.4)计算属性的信息增益率

选择属性列表中信息增益率高的条件属性作为当下的节点 N

End

End

用后剪枝方法进行剪枝

End

PCA 算法中 QR 分解函数的具体实现如下：

协方差矩阵（n*n）的 QR 分解（Householder 变换）的核心思想伪代码：

For 矩阵每一行 x[i] Do{

计算 $\mu = x[i] + sign(x[i])\|x[i]\|_2$；

算出 $\mu = \mu / \|\mu\|_2$；

再计算 $\mathrm{H}_i = \mathrm{I} - 2\mu\mu^{\mathrm{T}}$； }

4.6 实验与分析

4.6.1 实验设计

本章实验还是通过 UCI 数据集对这章提出的 RPC4.5 算法的性能进行测定。将 RPC4.5 算法与 C4.5 算法、PSOC4.5.算法[55]、FuzzyDT 算法[48]以及上一章提出的 GC4.5 算法进行对比实验，对比这些算法的建模速度和分类正确率。

实验是在 Window10 操作系统下进行的，系统软硬件环境为：内存 2G，Celeron(R) Dual-Core CPU 1.80GHz 上进行的，配置的实验环境还有怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），Java 的 JDK、JRE 和 Eclipse。这些数据集在 Eclipse 重新编译的 WEKA 平台中进行实验。

同第三章，采用 16 个 UCI 数据集作为实验数据集，数据集的具体信息如表 4-1 所示。数据集信息包括样本数(Examples)、条件属性个数(Features)和类属性值个数(Classes)，其中条件属性中还包括连续型(c)和离散型(d)的个数。

表 4-1 UCI 数据集

No.	Dataset	Examples	Features(c d)	Classes
1	Breast	699	9 (9 0)	2
2	Letter	20000	16 (16 0)	26
3	Credit	1000	20 (7 13)	2
4	Diabetes	768	8 (8 0)	2
5	Gamma	19020	10 (10 0)	2
6	Lymph	148	18(3 15)	4
7	Haberman	306	3(3 0)	2
8	Heart	270	13 (13 0)	2
9	Anneal	898	38 (6 32)	6
10	Liver	345	6 (6 0)	2
11	Waveform	5000	40 (40 0)	3
12	Soybean	683	35 (0 35)	19
13	Spam	4601	57 (57 0)	2
14	Car	1728	6 (0 6)	4
15	Vehicle	846	18 (18 0)	4
16	Segment	2310	19 (19 0)	7

No.:表示数据集的序号。

实验主要利用 Eclipse 平台重新编译运行 WEKA 进行，具体的实验步骤与第三章实验步骤相同，都是先利用 Eclipse 开发工具编写改进的算法，然后重新编译 WEKA，并运用改进的算法结合 16 个 UCI 数据集并借助 Explorer 实验平台进行实验。与第三章不同的是，本章的实验算法比较多，新增了一个本章提出 RPC4.5 算法。将 RPC4.5 算法与 C4.5 算法、PSOC4.5 算法、FuzzyDT 算法以及上一章提出 GC4.5 算法进行对比实验，对比算法的建模时间和分类正确率。

4.6.2 实验结果与分析

同第三章实验，将改进算法的实验参数设置为 10 次交叉验证，并将剪枝策略的置

信度设置为 25%进行剪枝，每个数据集都将运行 10 次取平均值得到实验数据。

首先是算法的建模时间，这里的时间是取 10 次实验的平均建模时间，如下表所示：

表 4-2 建模时间

No.	RPC4.5	GC4.5	C4.5	PSOC4.5	FuzzyDT
1	**0.014**	0.016	0.018	0.020	0.022
2	7.118	**5.747**	6.736	6.714	6.170
3	0.048	0.038	0.042	**0.020**	**0.020**
4	0.020	0.020	0.023	**0.019**	0.020
5	1.660	1.636	2.142	0.756	**0.322**
6	0.016	**0.001**	**0.001**	**0.001**	**0.001**
7	**0.004**	**0.004**	0.008	0.006	0.020
8	0.008	**0.001**	0.008	0.016	0.010
9	0.112	0.078	0.070	**0.020**	0.056
10	**0.001**	0.004	0.012	**0.001**	**0.001**
11	0.488	0.854	1.128	0.610	**0.354**
12	0.152	**0.032**	0.040	**0.032**	**0.032**
13	1.176	1.034	1.248	**0.660**	0.984
14	0.082	0.016	0.018	**0.004**	0.020
15	0.036	0.058	0.084	0.054	**0.034**
16	0.128	0.126	0.138	0.074	**0.066**

表中实验数据的单位为秒，表示建模时间。

从上面表中的实验数据结果可知，16 个实验数据集和 5 个对比算法中，建模时间较短的数据集：RPC4.5 占 3 个；GC4.5 算法一共有 6 个；C4.5 算法只有 1 个；而加入 PSO 改进的算法（PSOC4.5）占了 7 个；FuzzyDT 算法有 8 个数据集。

其中，第 3 个数据集（Credit）的建模时间是 PSOCC4.5 算法和 FuzzyDT 算法的一样，都快于其他数据集；第 6 个数据集（Lymph）的建模时间是 RPC4.5 算法的最慢，其他的算法建模时间相同；第 7 个数据集（Haberman）的建模时间是 RPC4.5 算法与 GC4.5 算法相同，比其他算法的时间短；第 10 个数据集（Liver）建模时间中，RPC4.5 算法、PSOCC4.5 算法和 FuzzyDT 算法的一样，高于 GC4.5 算法和 C4.5 算法；第 12 个数据集（Soybean）则是 GC4.5 算法、PSOCC4.5 算法和 FuzzyDT 算法一样，高于 RPC4.5 算法和 C4.5 算法。

与 GC4.5 算法相比，RPC4.5 算法的建模时间中，有 4 个数据集的建模时间低于 GC4.5 算法的；还有 10 个数据集的建模时间比 GC4.5 算法的长；有 2 个数据集的建模时间一样。

与 C4.5 算法相比，RPC4.5 算法的建模时间有 9 个数据集的建模时间低于 C4.5 算法的；有 1 个数据集的建模时间一样，还有 6 个数据集的建模时间比 C4.5 算法的长。

与 PSOC4.5 算法相比，RPC4.5 算法的建模时间只有 5 个数据集的建模时间比 PSOC4.5 算法的短；有 10 个数据集的建模时间比 PSOC4.5 算法的长；还有 1 个数据集的建模时间一样。

与 FuzzyDT 算法相比，RPC4.5 算法的建模时间只有 3 个数据集的建模时间比 FuzzyDT 算法的短；有 11 个数据集的建模时间比 FuzzyDT 算法的长；还有 2 个数据集

的建模时间一样。

为了直观地查看对比结果，现将上表中的实验数据用调整系数进行处理比较，然后再根据调整后的数据进行比较分析。

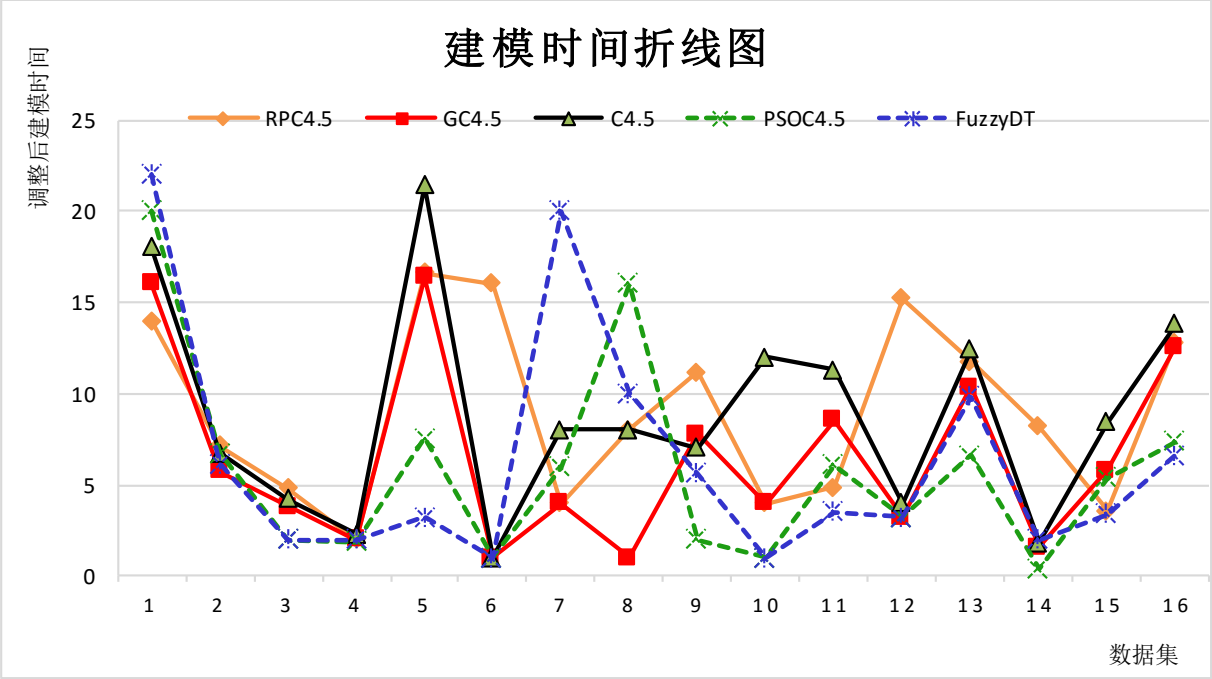


图 4-2 建模时间比对图

结合上图和实验数据分析结果可知，建模时间较短并且较稳定的算法是 PSOC4.5 算法，其次是 GC4.5 算法，虽然 FuzzyDT 的建模时间比较短，但是不够稳定；然后是 RPC4.5 算法，最后是 C4.5 算法。可见，虽然 RPC4.5 算法的建模速度比不上 PSOC4.5 算法和 FuzzyDT 算法，但对于 C4.5 算法，还是有提高的。

同样，取 10 次实验的平均分类正确率作为实验的结果数据，如下表 4-3 所示。

表 4-3 分类正确率

No.	RPC4.5	GC4.5	C4.5	PSOC4.5	FuzzyDT
1	**0.9728**	0.9499	0.9456	0.9456	0.9499
2	0.7961	**0.8789**	**0.8798**	**0.8798**	0.4883
3	0.8000	0.7220	0.7050	0.7310	**0.8519**
4	0.7070	0.7396	0.7383	**0.7487**	0.7389
5	0.8001	0.8504	**0.8506**	0.8280	0.8018
6	0.7838	**0.8041**	0.7703	0.7568	0.7724
7	0.7124	0.7288	0.7190	0.7255	**0.7293**
8	0.7704	**0.8074**	0.7667	0.7852	0.7453
9	0.9688	**0.9866**	0.9844	0.9643	0.9426
10	0.5797	0.6725	**0.6870**	0.6086	0.6081
11	**0.8440**	0.7670	0.7508	0.7632	0.7404
12	0.8507	**0.9209**	0.9150	0.9165	0.8971
13	0.8970	0.9289	0.9298	**0.9337**	0.7102
14	**0.9641**	0.9236	0.9236	0.7002	0.9201
15	0.6300	**0.7411**	0.7246	0.7163	0.6280
16	0.9104	0.9662	**0.9693**	0.9680	0.8947

从上面表中的实验数据结果可知，16 个实验数据集和 5 个对比算法中，分类正确率较高的数据集：RPC4.5 占 3 个；GC4.5 算法一共有 6 个；C4.5 算法有 4 个；而加入 PSO 改进的算法（PSOC4.5）占了 3 个；FuzzyDT 算法却只有 2 个数据集的分类正确率较高。

其中，第 2 个数据集（Letter）的分类正确率是 GC4.5 算法、C4.5 算法和 PSOC4.5 算法一样高，都高于 RPC4.5 算法和 FuzzyDT 算法。

RPC4.5 算法与 GC4.5 算法相比，只有 4 个数据集的分类正确率较高，还有 12 个数据集的正确率较低。

RPC4.5 算法与 C4.5 算法相比，RPC4.5 有 6 个数据集的分类正确率高于 C4.5 算法；而 10 数据集的正确率较低。

RPC4.5 算法与 PSOC4.5 算法相比，有 6 个数据集的分类正确率较高，还有 10 个数据集的正确率较低。

RPC4.5 算法与 FuzzyDT 算法相比，有 10 个数据集的分类正确率较高，有 6 个数据集的正确率较低。

为了直观地查看实验结果，根据表中数据得到相应的折线图：

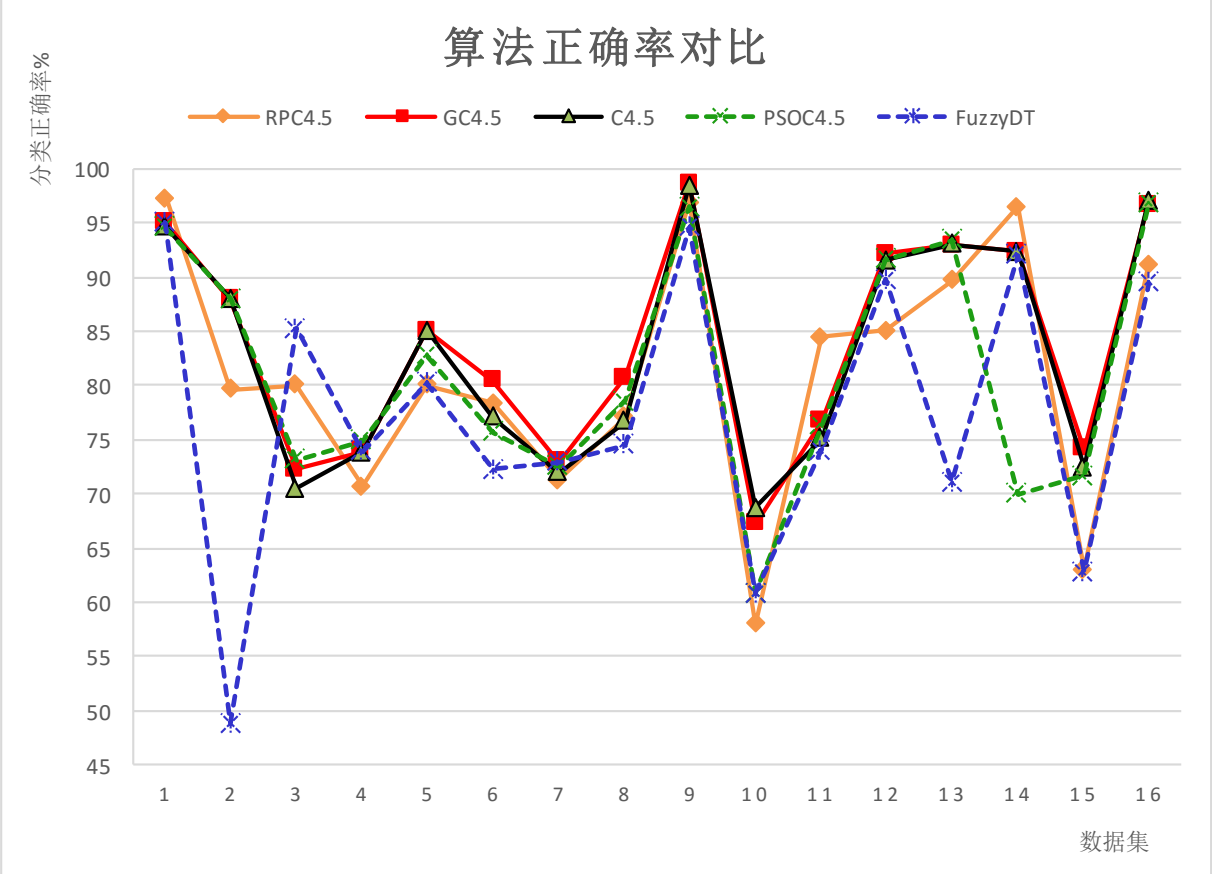


图 4-3 分类正确率比对图

从上面的折线图可以看出， RPC4.5 算法的分类正确率较 FuzzyDT 算法的高，但是较其他的对比算法低。RPC4.5 算法在分类正确率的改进上没有 GC4.5 算法的好；GC4.5 算法的分类正确率相对较高，并且相对必较稳定，还需再继续探究相关知识，提高 RPC4.5 算法的性能。

根据以上实验结果可以得到 5 个算法在 16 个数据集中的平均正确率和平均建模时间：

表 4-4 平均分类正确率和建模时间

	RPC4.5	GC4.5	C4.5	PSOC4.5	FuzzyDT
平均正确率	0.8117	**0.8367**	0.8287	0.8107	0.7733
平均建模时间	0.6914	0.6040	0.7318	0.5629	**0.5055**

综上所述，可知：对于 C4.5 算法，本章提出 RPC4.5 算法在建模时间上有一定的改进，但是分类正确率比较不理想；相比于上一章提出的 GC4.5 算法，建模时间和分类正确率都比不过 GC4.5 算法，因此本文提出的两个改进算法中，GC4.5 算法是比较成功的改进，RPC4.5 算法在时间上稍微的提高。但 RPC4.5 算法性能优于 C4.5 算法和 FuzzyDT 算法，并且在平均正确率上，RPC4.5 算法优于 PSOC4.5 算法，说明 RPC4.5 算法比较稳定，相对来说改进也是可行的。

两种改进算法建模时间较长，效果不太理想，是因为：GC4.5 虽然简化了对数运算，运用 GINI 指数均值消除属性间相关性影响，但是 GC4.5 算法中的 GINI 指数均值计算较为复杂，所以建模时间较长。而 RPC4.5 算法虽然根据属性依赖度计算原理删除无关属性，并运用 PCA 算法的压缩原理简化数据集，消除属性相关性影响，但是 RPC4.5 算法需要进行多重判断，并且运用了 PCA 进行过滤，所以在时间上不具有明显的优势，也降低了分类规则的可解释性。

从 4-2 图中可以看出，数据集样本个数较大或者属性值比较多时，算法建模时间都比较长；而从 4-3 图中可以看出，个别分类效果不佳的数据集在所有算法的分类正确率中都不太理想，所以结果也与数据集内部特征有关。

4.7 本章小结

本章对基于 C4.5 算法的冗余计算和属性相关性影响考虑，提出了基于属性依赖度计算和 PCA 的 C4.5 优化算法（RPC4.5），先介绍了属性依赖度计算的原理，延伸到条件属性与类属性间的依赖度计算。然后简单概述了 PCA 算法的思想和步骤，根据 PCA 算法压缩原理，得到启发：将数据集经由 PCA 算法处理后输出互相独立的属性集，消除属性相关性。接着，描述了 RPC4.5 算法的思想，并介绍了 RPC4.5 算法的流程图及伪代码。最后通过大量的 UCI 数据集对算法进行测试实验，实验结果表明，本章提出的 RPC4.5 算法分类正确率较 FuzzyDT 算法的高，却低于 GC4.5 算法、C4.5 算法和 PSOC4.5 算法；在建模速度上有一定的提高。根据实验得到结论：GC4.5 算法的性能较好，改进比较成功。但在综合性能来说 GC4.5 算法和 RPC4.5 算法都相对稳定，所以改进都是可行的。

第五章 学生英语统考成绩预测

5.1 英语统考成绩预测系统目的与意义

学生成绩预测是一门研究热点，现如今成绩预测和学生表现预测等在各教育机构应用广泛。选择合适的数据挖掘算法进行预测分析，根据预测结果进行针对性的教育和培训等，可以大幅地提高教育质量和学生能力。

对学生英语统考成绩进行预测分析，为了：

（1）更好地进行统计分析。在统考之前先整体估量统考结果；

（2）针对性采取相应措施。根据预测分析结果，针对性地对学生进行相关的培训和指导，从而提高学生的应考能力，提高统考成绩；

（3）对学生成绩进行预测分析是教育工作过程的关键步骤。通过对学生英语统考成绩的预测，教育机构可以更具明确性地进行相关工作。

（4）从学生出发，对自己的成绩进行预测分析，可以发现自身的不足，进行改进，提高自身的能力。

综上所述，对学生英语统考成绩进行预测分析，具有重大的意义。

5.2 基于 GC4.5 算法和 RPC4.5 算法的成绩预测

预测系统一般选择分类算法进行，基于 C4.5 算法的预测系统很多，但是由于 C4.5 算法的一些缺点，可能导致系统的不足，因此本章提出基于 GC4.5 算法成绩预测、基于 RPC4.5 算法成绩预测。将改进的 C4.5 算法应用于学生英语统考成绩预测中。

基于 GC4.5 算法成绩预测

首先是基于 GC4.5 算法的学生英语统考成绩预测，该流程是将第三章提出的改进算法（GC4.5）用作英语统考成绩预测系统中的分类算法，实际的英语统考成绩数据集来源于江南大学继续教育和网络教育学院的学生数据库，用 GC4.5 处理的流程图如图 5-1 所示。

在流程图第一步中，数据整理好之后，由于原始的数据集时 Excel 表的.xls 格式，需要将数据集转换成适合 WEKA 平台处理的数据格式.arff 格式，所以数据整理好之后需要进行数据格式的转换。

图中删除姓名与学号的操作，是因为学号和姓名具有唯一性，考虑到 C4.5 算法的剪枝策略问题，将这两个属性删除，否则决策树模型在进行剪枝时，会将决策树分支都剪掉，最后只剩下一个节点，因此在数据预处理的时候将姓名和学号这两个属性删除，然后再建立预测模型。

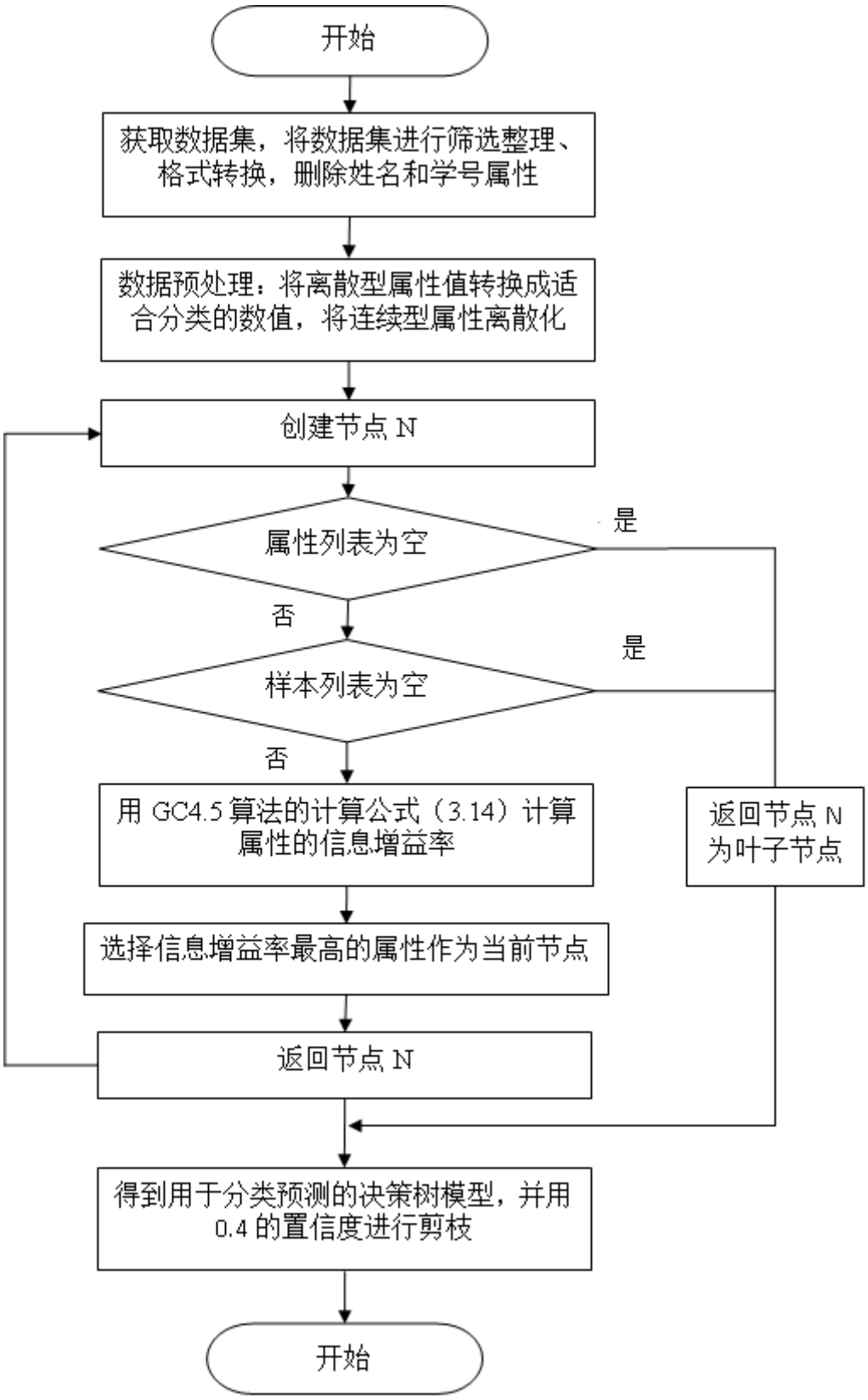


图 4-1 基于 GC4.5 算法成绩预测流程图

基于 RPC4.5 算法成绩预测

同理，基于 RPC4.5 算法的学生英语统考成绩预测，是将第四章提出的改进算法（RPC4.5）用作英语统考成绩预测系统中的分类算法，与 GC4.5 不同的是，该算法中删除姓名和学号属性是因为 PCA 算法的过滤问题，具体的流程如下：

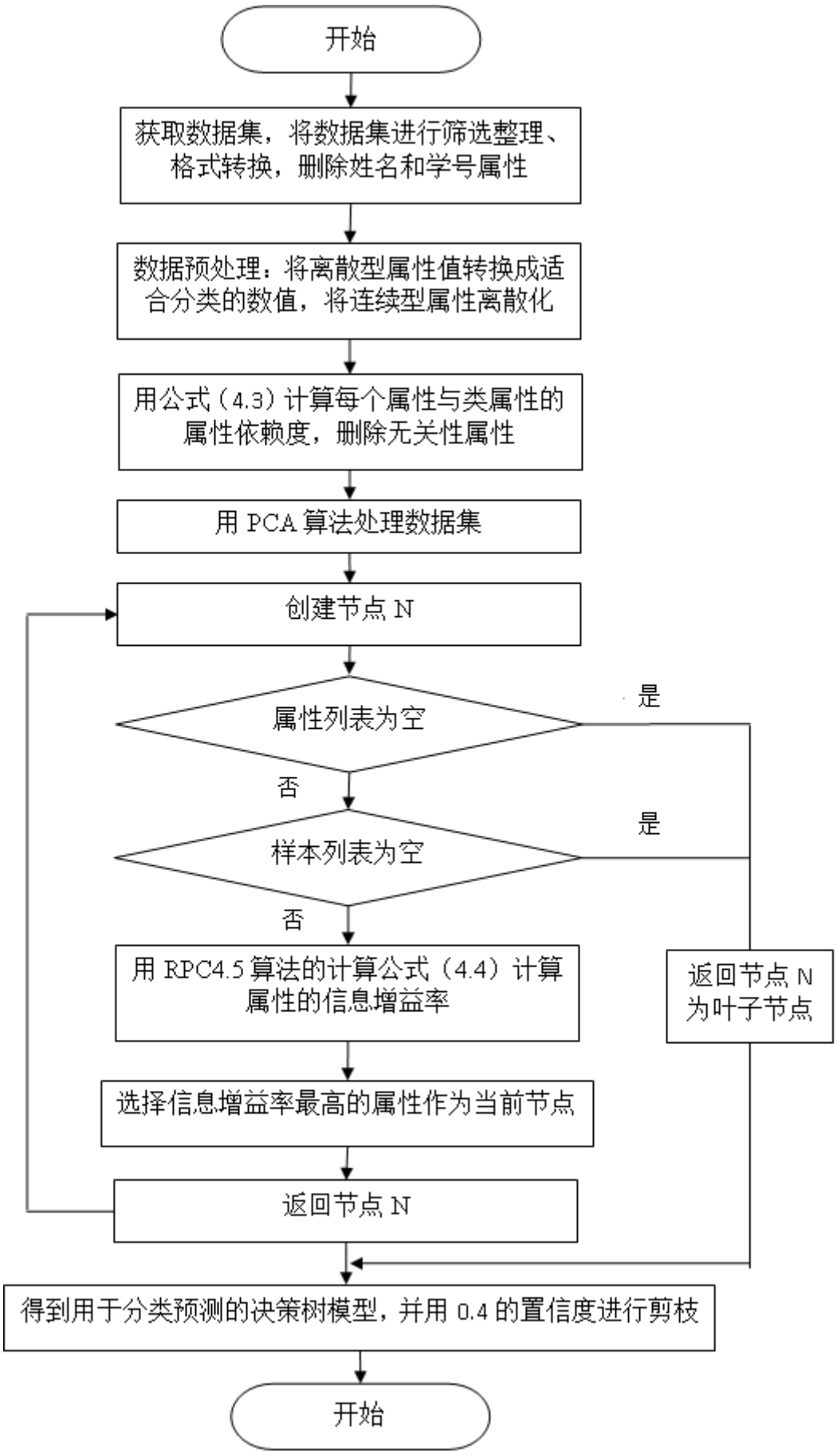


图 5-2 基于 RPC4.5 算法的成绩预测流程图

5.3 实验与分析

5.3.1 实验设计

本章实验是用学校学生数据库中的实际数据集进行评测。将 GC4.5 算法、RPC4.5 算法与 C4.5 算法分别对实际的数据集进行实验，对比这些算法在实际应用中的建模速度和分类正确率。同时，将两个对比算法 PSOC4.5 算法[55]和 FuzzyDT 算法[48]也加入实际应用中进行实验对比。

实验环境

实验是在 Window10 操作系统下进行的，系统软硬件环境为：内存 2G，Celeron(R) Dual-Core CPU 1.80GHz 上进行的，配置的实验环境还有怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），Java 的 JDK、JRE 和 Eclipse。利用实际的数据集、改进算法和对比算法借助 Eclipse 进行实验验证。

数据信息

实验数据是从江南大学继续教育和网络教育学院的英语统考成绩数据库中随机抽取的数据。数据的属性包括学号、姓名、性别、入学年龄、专业名称、入学测试大学英语、入学总成绩、大学英语二、大学英语三、已学课程平均成绩、学习情况、统考大学英语等 12 个属性。根据基于 GC4.5 算法的成绩预测和基于 RPC4.5 算法成绩预测的流程，将姓名和学号这两个属性删除，则还有 10 个属性，其中“统考大学英语”为类属性，其余的 9 个属性为条件属性。数据集的具体信息如下表所示：

表 5-1 学生英语统考成绩数据信息表

Dataset	Examples	Features(c d)	Classes
EnglishAchive	2895	6（6 0）	2

表中，6（6 0）表示有 6 个条件属性，其中 6 个是连续型属性，0 个是离散型属性。

离散型属性分别是：性别、专业名称和学习情况；其余的为连续型属性。

两个类属性值（Classes）分别是：合格、不合格。

具体数据集样本如表 5-2 所示，表中的数据为预处理之后的数据样本，已经将原本离散型属性的值转换成适合分类的数值；并且已经将连续型属性的值进行了离散化处理。

实验步骤

Step1. 获取数据集。从江南大学继续教育和网络教育学院的学生数据库中随机抽取 3050 个数据记录，将该数据集进行筛选整理，剩余 2895 个样本。

Step 2. 数据格式转换。因为 WEKA 数据挖掘工具有自己的数据格式，支持最好的数据格式是 arff 格式，所以需要将数据集进行格式转换。首先，将数据集表格格式用 Excel 另存为 CSV 格式；然后，用 WEKA 打开 CSV 格式的数据集，并另存为.arff 格式。

Step 3. 数据属性处理。数据集整理好后，将原本离散型属性的值转换成适合分类的数值；将连续型属性的值离散化处理，如：入学年龄、入学测试大学英语、入学总成绩、大学英语二、大学英语三、已学课程平均成绩等属性。

Step 4. 算法选择和实验参数设置。将预处理后的数据集选择相应算法（GC4.5、RPC4.5、 C4.5 算法、PSOC4.5 算法或者 RPC4.5 算法）进行实验，然后进行实验参数设置：每次设置 10 次交叉验证，并决策树的剪枝置信度设置为 0.4。

Step 5. 获取实验数据。每个算法都进行 10 次实验，取每次实验结果数据的平均值作为最终的实验数据，将结果数据汇总成表格；分析实验结果图，得到相关结论。

表 5-2 学生英语统考成绩数据样表

属性 1	属性 2	属性 3	属性 4	属性 5	属性 6	类属性
5	4	2	5	4	4	合格
5	3	4	2	1	3	合格
5	4	3	3	4	4	不合格
5	4	4	3	2	3	合格
5	4	3	3	2	4	合格
5	4	4	4	4	3	合格
5	4	4	4	3	3	合格
4	3	4	5	3	2	合格
5	4	2	4	4	3	不合格
5	5	2	5	3	4	合格
5	4	4	2	4	3	合格
5	4	3	3	4	2	不合格
5	4	3	4	2	2	合格
5	4	3	5	3	2	不合格
5	4	1	2	2	4	合格
5	4	2	4	2	2	合格
5	4	2	4	3	5	合格
5	4	4	4	4	3	合格
5	3	4	4	3	4	合格
…	…	…	…	…	…	…
4	3	4	3	3	2	合格

表中，属性 1 到属性 9 分别对应属性为：入学测试大学英语、入学总成绩、大学英语二、大学英语三、已学课程平均成绩、学习情况；类属性对应：统考大学英语。

离散化说明：上表中的属性值都是经过离散化处理之后的数值，离散化过程为：属性 1 到属性 5 均为分数制成绩，属连续型属性，用 1 到 5 个等级进行离散化，5 代表分数在 90-100 之间的，4 代表 80-90 之间的成绩，3 代表 70-80 之间的成绩，2 代表 60-70 之间的成绩，而 1 则是 60 以下的成绩。

而属性 6——学习情况，是一个根据学生在某一段时间内登录英语学习系统学习的次数来进行定义的一个属性，主要描述学生学习英语的频繁程度，在这一段时期的登录次数也是一个连续型数值，同样用 1 到 5 的等级进行离散化，5 表示登录次数在 200 以上的，4 代表登录次数在 150 到 200 之间的，而 3 代表登录次数在 100 到 150 之间，2 则表示登录次数在 50-100 的，1 是 50 以下的登录次数。

5.3.2 实验结果与分析

将实验平台 WEKA 中的实验参数设置为 10 次交叉验证，并将剪枝策略的置信度设置为 40%进行剪枝，每个数据集都将运行 10 次取平均值得到实验数据。

表 5-3 列出了每种算法的建模时间和分类正确率。

表 5-3 实验中 3 种算法的建模时间和分类正确率对比

	GC4.5	RPC4.5	C4.5	PSOC4.5	FuzzyDT
建模时间	**0.028**	0.039	0.110	0.064	0.052
分类正确率	**0.8248**	0.8245	0.8062	0.8241	0.8167

根据上表可知 GC4.5 算法具有一定的优势，下图是 5 个算法在建模时间和分类正确率上的性能对比图：

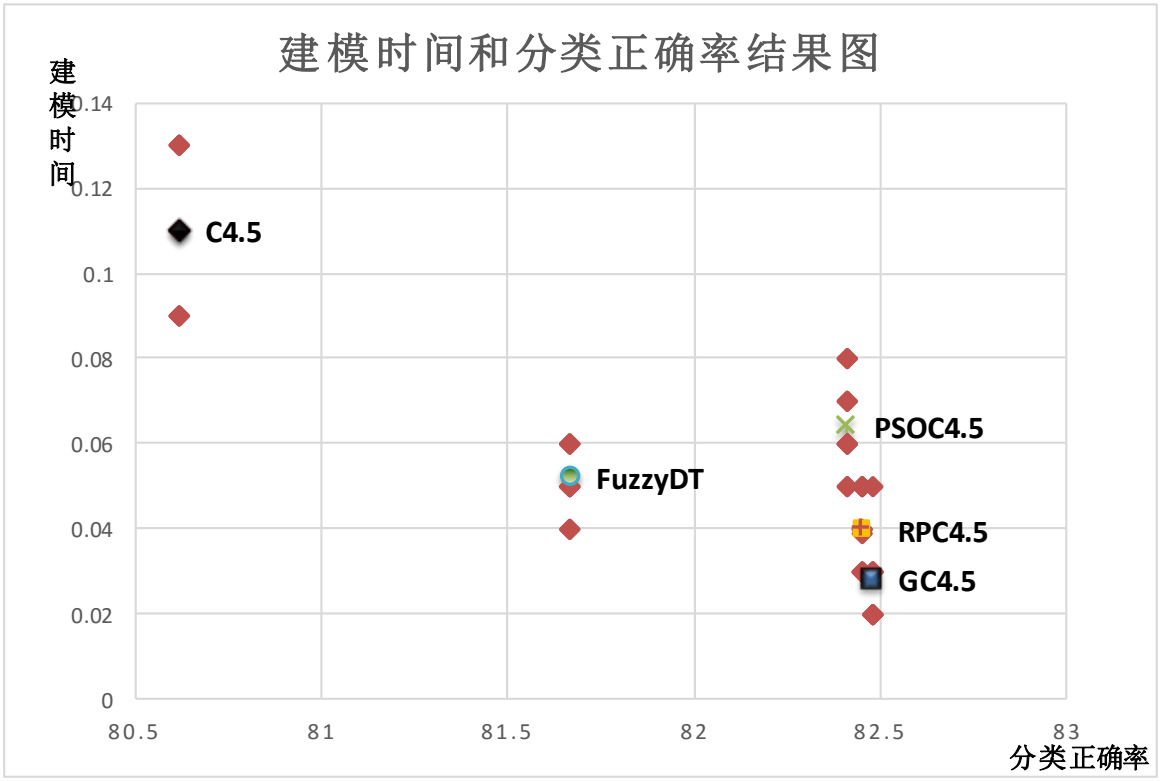


图 5-3 算法性能对比图

图中，横坐标表示分类正确率(*100%)，纵坐标表示建模时间。

标记的点是各个算法 10 次实验的平均结果值，而散列在周边的点是该算法在实验过程中取得的实验值，其中有很对点重叠在一起了，所以视觉上看不出是 10 个结果。

从图 5-3 中可以直观地看出，对于分类正确率，GC4.5 算法的分类正确率最高，C4.5 算法的分类正确率最低；对于建模时间，GC4.5 算法的建模时间最短，C4.5 算法的建模时间最长。从上图中可以看出，GC4.5 算法的综合性能最好。

下图是数据集分类的决策树模型：

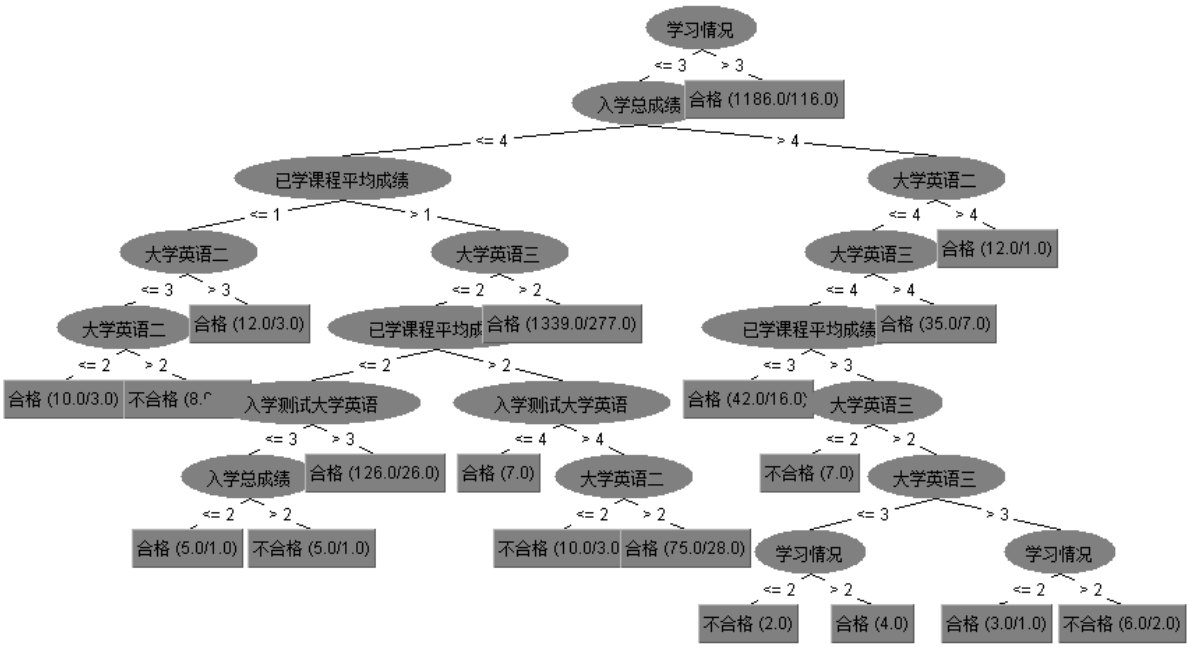


图 5-4 决策树模型

经过对比实验结果图表得出结论：在建模速度上，GC4.算法在这个实际数据集中具有一定的优势，因为算法将对数运算简化成了基本的加减乘除运算，所以会有较快的计算速度；在分类正确率上， GC4.5 算法的分类预测结果正确率相比于比 RPC4.5 算法和 C4.5 算法的分类预测结果具有一定的优势。理论上，GC4.5 算法将对数运算都转换成了加减乘除的基本运算，并且加入 GINI 指数消除属性相关性，所以在分类预测的正确率上有一定的优势。

从综合性能来说，两个改进的算法应用于英语统考成绩预测中是成功的，不管是在建模速度上还是在分类正确率上，都优于 C4.5 算法和其他两个对比算法。从应用角度来讲，GC4.5 算法又优于 RPC4.5 算法，因为经过实验发现 GC4.5 算法的综合性能较高。

5.4 本章小结

本章首先描述了学生成绩预测系统的目的和意义，预测系统好坏的关键在于分类算法的选择。本章提出了基于 GC4.5 算法的成绩预测和基于 RPC4.5 算法的成绩预测，将本文对于 C4.5 算法改进两个算法应用到成绩预测系统中，详细地给出了应用的流程和设计；接着，运用改进后的算法和对比算法进行实验，详细的描述了实验设计和实验的步骤，从数据获取、预处理到获取实验数据都详细描述了相关内容；最后，得出最终的实验结果，分析实验结果数据表和实验图可以得到实验结论：改进的 C4.5 算法的分类正确率和建模速度都优于 C4.5 算法、PSOC4.5 算法和 FuzzyDT 算法，说明改进的算法应用于成绩预测系统是可行的。

第六章　总结与展望

6.1 论文总结

本文第一章绪论，主要阐述了数据挖掘在生活中的重大意义。ID3 算法是十大数据挖掘算法之一，是著名的决策树分类算法。将 ID3 算法的改进延伸得到 C4.5 算法，介绍了 C4.5 算法的特点和广泛应用，说明对于 C4.5 算法的优化研究具有非常重大的意义。同时，介绍了国内外对于 C4.5 算法的研究动态。

第二章决策树分类算法，首先概述了决策树分类算法，介绍了属性选择度量，预先剪枝和后剪枝两种剪枝方法。然后详细描述了 ID3 算法的思想，介绍了信息熵和信息增益的概念和计算公式，由这两个概念推出 ID3 算法的计算公式，并给出了 ID3 算法建立决策树的流程图，描述了 ID3 算法创建决策树的过程。接着详细描述了基于 ID3 算法进行优化的 C4.5 算法，介绍了 C4.5 算法的思想和信息增益率计算公式，给出了 C4.5 算法的步骤和流程图，并分析、概述了 C4.5 算法的优缺点。最后介绍了两种改进的 C4.5 算法，先介绍基于粒子群优化算法的 C4.5 算法（PSOC4.5），简单概述了 PSOC4.5 算法的思想和步骤；然后介绍了基于模糊算法的 C4.5 算法（FuzzyDT），给出了 FuzzyDT 算法的思想和步骤。

第三章基于 GINI 指数均值的 C4.5 优化算法（GC4.5），首先，介绍了泰勒级数的数学原理，结合等价无穷小的原理，简化 C4.5 算法中的对数运算，接着介绍属性相关性的概念和 GINI 指数理论，根据 GINI 指数均值的原理改进 C4.5 算法，将条件属性的 GINI 指数均值引入信息增益率计算公式中，用于消除属性间相关性对于分裂属性选择的影响。然后，详细地描述了 GC4.5 算法的流程，给出了具体的计算公式和算法的流程图，并给出了具体的伪代码；最后，通过将 GC4.5 算法与 PSOC4.5 算法、FuzzyDT 算法以及 C4.5 算法进行对比实验，将 16 个 UCI 数据集用改进后的算法以及对比算法进行实验，实验设置为 10 次交叉验证，实验数据表明，从总体上来说，GC4.5 算法的性能较高，优于如今存在的一些 C4.5 的改进算法。

第四章基于属性依赖度计算和 PCA 的 C4.5 优化算法（RPC4.5），首先，详细介绍了属性依赖度计算的原理，并由此推导出了条件属性与类属性之间的依赖度计算公式，用于删除不相关的条件属性，避免无关计算，减少算法计算时间。接着，介绍了 PCA 算法原理，概述了 PCA 算法压缩原理与 C4.5 算法相结合的意义和作用，经由 PCA 算法处理后输出的数据集属性间相互独立，从而消除了属性间相关性的影响。然后，详细地描述了 RPC4.5 算法，给出了 RPC4.5 算法的具体计算公式和流程图，并给出了具体的伪代码以及一些函数的详细步骤过程。最后，进行实验验证，将 RPC4.5 算法与 GC4.5 算法、PSOC4.5 算法和 FuzzyDT 算法进行对比，实验结果表明，RPC4.5 算法分类正确率和建模速度有一定的提高。

第五章学生英语统考成绩预测，首先介绍了英语统考成绩预测的意义和必要性；接着，描述了基于 GC4.5 算法和 RPC4.5 算法的英语成绩预测系统，将改进后的 C4.5

算法应用于实际的数据集中，根据决策树分类模型对学校学生的英语统考进行成绩预测，根据预测结果实施相应的措施，提高统考成绩的通过率。最后，用学校学生成绩数据库中的数据进行实验对比，证明改进算法的优越。

6.2 论文的主要创新点

本文主要针对 C4.5 算法的改进和应用进行研究。本文在分析了 C4.5 决策树算法不足的基础上，提出了两个优化改进的 C4.5 算法。并将改进算法用于学生英语统考成绩预测中。主要进行了以下工作：

（1）针对 C4.5 算法对数运算多、属性间相关性影响属性选择正确性的问题，提出了一种基于 GINI 指数均指的 C4.5 优化算法（GC4.5）。首先，根据泰勒级数的数学原理，结合等价无穷小的原理，将 C4.5 算法中的对数运算简化为加、减、乘、除运算，目的是减少算法的计算时间；接着，根据 GINI 指数均值的原理改进 C4.5 算法，将条件属性的 GINI 指数均值引入信息增益率计算公式中，用于消除属性间相关性对于属性选择的影响，提高属性选择的准确性，从而提高算法的分类正确率。通过对大量的 UCI 数据集进行实验，结果表明，GC4.5 算法较现有的一些 C4.5 改进算法具有更快的建模速度和跟高分类正确率。

（2）针对 C4.5 算法冗余计算影响分类效率和属性间相关性影响的问题，提出了一种基于属性依赖度计算和 PCA 的 C4.5 优化算法（RPC4.5）。首先，根据属性依赖度计算的原理，得到条件属性与类属性之间的依赖度计算公式，并将该计算公式应用于训练和测试数据集中，用于删除无关条件属性，避免冗余计算，减少算法的时间开销。接着，根据 PCA 算法压缩原理，将数据集用 PCA 算法处理，得到条件属性间相互独立的数据集，从而消除了属性间相关性的影响。然后，结合泰勒级数和等价无穷小原理简化计算公式，得到新的属性度量公式。通过对大量的 UCI 数据集进行实验，结果表明，RPC4.5 算法高于其他一些 C4.5 改进算法，具有更快的建模速度和跟高分类正确率。

（3）学生英语统考成绩预测系统是分类算法的一个重要应用，预测结果正确率的高低取决于分类算法分类正确率的高低，所以学生英语统考成绩预测系统中，算法的选择是关键。为了优化英语统考成绩预测，就应该合理地选择分类算法。因此本文将基于 GINI 指数均指的 C4.5 优化算法和基于属性依赖度计算和 PCA 的 C4.5 优化算法应用于英语统考成绩预测中，提高大学生英语统考成绩预测准确率。

6.3 论文存在的问题以及未来工作的展望

科技发展的步伐从未间断，本文中对决策树分类算法的性能虽然有所改进，但是优化改进的算法依旧有不足的地方，如本文改进的数据挖掘中决策树分类算法只是应对一个类属性和两个类属性值数据集的情况，但是随着电子信息科学技术的快速发展，大数据的信息时代已经到来，而大数据不仅数据量大，而且其多类属性和多属性值的特点也是不容忽视的。由于大数据拥有数据的量大、数据多样、增长速度非常快、数据的价值信息密度低等特点，所以在实际应用中，特别是在需要高效实时地处理大数

据的行业中，大大增加了对数据处理和分析的难度。我们需要进一步开展研究工作来应对这种形势，主要的研究工作计划具体为以下几点：

RPC4.5 算法的建模速度和分类正确率还需进一步提高，应该继续深入研究，开发出具有更好性能的改进算法。

算法需要进行多重判断，不能对任何一个数据集进行高效准确地处理，这会增加算法的时间复杂度。针对这个问题，结合大数据的特点进一步研究数据挖掘算法，改进算法的性能，并让数据挖掘算法与数据分析平台充分结合，开发出可以快速高效地处理杂乱繁多的算法。

算法只能处理单个类属性的数据集，缺乏对于具有多个类属性的数据集进行处理的能力。可处理多个类属性数据集的决策树分类算法有待深入研究。研发出一种有效公平的测评机制，在评估的工作过程中，具有更准确的属性选择和更合理的评估结果，提高企业或学校的评估工作的公平性与合理性；同时保证算法的高效准确的性能。

将数据挖掘算法与云计算并行处理的原理相结合，开发出能高效处理大数据的数据挖掘算法，提高算法处理的承受能力；同时是实现算法对数据源的处理能力，可以让客户更加方便快捷地使用平台对数据进行处理。

致　谢

首先，我要诚挚地感谢我的导师孙力教授！在这三年的研究生求学生涯中，我的导师对我的学术研究工作进行了无私的帮助与细心的指导。是导师把迷茫的我带入了有方向的研究中；是导师在我一次次失败的时候给予我鼓励与指导；是导师在我困惑的时候给我解答……是导师以自身严谨的治学态度，教会了我在搞研究工作时应有的态度。不仅如此，在撰写论文之时，孙力导师以渊博的科学知识和认真负责态度，给我了很多建设性的意见。孙教授的工作精神和治学态度以及敏锐的科学思维让我受益终身。在此衷心感谢孙老师的指导！

同时，非常感谢颜丽燕，赵莉莉，高晓梅，周袅等同学，在我的学术研究中给予我的建议与帮助。在这里也要感谢王建龙同学，感谢他在我的研究工作中给予的大力支持，也感谢他帮助我解决了很多难题。

最后我要感谢我的家人，谢谢他们在这期间给予我的关爱，感谢他们给予我的精神上的鼓励和物质上的支持，让我可以静心地完成研究工作！

参考文献

1. Han, J, Kamber, M. Data Mining: Concepts and Techniques,Third EditionM]. 机械工业出版社, 2012.
2. Petr P. Data Mining [J]. Tools and Techniques, 1994, 29(1):47.
3. Szafron D, Greiner R, Lu P, et al. Explaining naïve Bayes classifications[D]. University of Alberta, 2003.
4. Wu X, Kumar V, Ross Quinlan J, et al. Top 10 algorithms in data mining[J]. Knowledge & Information Systems, 2007, 14(1):1-37.
5. Li F. The Information Content of Forward‐Looking Statements in Corporate Filings—A Naïve Bayesian Machine Learning Approach[J]. Journal of Accounting Research, 2010, 48(5):1049-1102.
6. Manickam S A V, Sharmila K. Survey on Data Mining Algorithm and its Application in Healthcare Sector Using Hadoop Platform[J]. Solid State Technology, 2015, 43(9):567-571.
7. Du J, Cui H, Li W, et al. Fault diagnosis of vacuum circuit breakers based on ID3 method[C]// Electric Power Equipment - Switching Technology (ICEPE-ST), 2011 1st International Conference on. IEEE, 2011:283-286..
8. Wang Q, De chun B A, Wang X D, et al. Diagnosis of RH-KTB Vacuum System Based on Decision Tree ID3 Theory[J]. Journal of Iron & Steel Research, 2006, 18(4):59-62.
9. Vishwakarma U, Jain A. Reduces Unwanted Attribute in Intruder File Based on Feature Selection and Feature Reduction Using ID3 Algorithm[J]. International Journal of Computer Science & Information Technolo, 2014, 5(1):896-900.
10. Meichun H U, Tian D, School B. Improved ID3 Algorithm Based on Parameter Correction and Simplified Standard[J]. Computer & Digital Engineering, 2015.
11. Quinlan J R. C4.5: programs for machine learning[C]. Morgan Kaufmann Publishers Inc. 1993.
12. Hashim H, Talab A A, Satty A, et al. Data Mining Methodologies to Study Student's Academic Performance Using the C4.5 Algorithm[J]. 2015, 5(2):59-68.
13. Mantas C J, Abellán J, Castellano J G. Analysis of Credal-C4.5 for classification in noisy domains[J]. Expert Systems with Applications, 2016, 61:314-326..
14. Elomaa B T. In Defense of C4.5: Notes on Learning One-Level Decision Trees[C]. Proc. of the 11th Int. Conf. on Machine Learning. 2015.
15. Chen C, He B, Zeng Z. A method for mineral prospectivity mapping integrating C4.5 decision tree, weights-of-evidence and m-branch smoothing techniques: a case study in the eastern Kunlun Mountains, China[J]. Earth Science Informatics, 2014, 7(1):13-24.
16. Wei D, Wei J. A MapReduce Implementation of C4.5 Decision Tree Algorithm[J]. International Journal of Database Theory & Application, 2014, 7(1):49-60.
17. Haddadi F, Runkel D, Zincir-Heywood A N, et al. On botnet behaviour analysis using GP and C4.5[C]. Companion Publication of the 2014 Conference on Genetic and Evolutionary Computation. ACM, 2014:1253-1260..
18. Qian H, Qiu Z. Feature selection using C4.5 algorithm for electricity price prediction[J]. 2014, 1:175-180.
19. Soliman S A, Abbas S, Salem A B M. Classification of thrombosis collagen diseases based on C4.5 algorithm[C]. IEEE Seventh International Conference on Intelligent Computing and Information Systems. 2015.
20. De'Ath G, Fabricius K E. Classification and Regression Trees: A Powerful Yet Simple Technique for Ecological Data Analysis [J]. Ecology, 2008, 81(81):3178-3192.
21. Zarei K. Prediction of Infinite Dilution Activity Coefficients of Halogenated Hydrocarbons in Water Using Classification and Regression Tree Analysis and Adaptive Neuro-Fuzzy Inference Systems[J]. Journal of Solution Chemistry, 2013, 42(3):516-525.
22. Domańska U, Papis P, Szydłowski J. Thermodynamics and activity coefficients at infinite dilution for

organic solutes, water and diols in the ionic liquid choline bis(trifluoromethylsulfonyl)imide[J]. Journal of Chemical Thermodynamics, 2014, 77:63-70.
23. Russell S J, Norvig P. Instructor's Manual: Exercise Solutions for Artificial Intelligence A Modern Approach Second Edition[J]. Artificial Intelligence A Modern Approach, 2015, 15(96):217-218.
24. El-said S H. Image quantization using improved artificial fish swarm algorithm[J]. Soft Computing, 2015,19(9):2667-2679.
25. Lebanon G, Lafferty J. Riemannian geometry and statistical machine learning[J]. Dissertation Abstracts International, 2015, 66(1):0367-0398.
26. Islam M R, Habib M A. A Data Mining Approach to Predict Prospective Business Sectors for Lending in Retail Banking Using Decision Tree[J]. Eprint Arxiv, 2015, 5(2):13-22.
27. Shakil K A, Anis S, Alam M. Dengue disease prediction using weka data mining tool[J]. Computer Science, 2015, 30(10):105018.
28. Guo Q, Jiang D. Method for Walking Gait Identification in a Lower Extremity Exoskeleton based on C4.5 Decision Tree Algorithm[J]. International Journal of Advanced Robotic Systems, 2015, 12(30):1-11.
29. Cetinkaya S, Basaraner M. Characterisation of Building Alignments With New Measures Using C4.5 Decision Tree Algorithm[J]. Geodetski Vestnik, 2014, 58(3):552-567.
30. Pattanapairoj S, Silsirivanit A, Muisuk K, et al. Improve discrimination power of serum markers for diagnosis of cholangiocarcinoma using data mining-based approach[J]. Clinical Biochemistry, 2015, 48(10-11):668-673..
31. Adhatrao K, Gaykar A, Dhawan A, et al. Predicting Students' Performance Using ID3 And C4.5 Classification Algorithms[J]. International Journal of Data Mining & Knowledge Management Proc, 2013, 3(5):39-52.
32. Chen Y, Cheng F, Lu S, et al. Computational models to predict endocrine-disrupting chemical binding with androgen or oestrogen receptors[J]. Ecotoxicology & Environmental Safety, 2014, 110:280-287.
33. Saeh I S, Mustafa M W, Mohammed Y S, et al. Static Security classification and Evaluation classifier design in electric power grid with presence of PV power plants using C-4.5[J]. Renewable & Sustainable Energy Reviews, 2016, 56:283-290.
34. Podolsky M D, Barchuk A A, Kuznetcov V I, et al. Evaluation of Machine Learning Algorithm Utilization for Lung Cancer Classification Based on Gene Expression Levels.[J]. Asian Pacific Journal of Cancer Prevention Apjcp, 2016, 17(2):835-838.
35. Alickovic E, Subasi A. Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier.[J]. Journal of Medical Systems, 2016, 40(4):1-12.
36. Huang S H, Teng N C, Wang K J, et al. Use of oximetry as a screening tool for obstructive sleep apnea: a case study in Taiwan.[J]. Journal of Medical Systems, 2015, 39(3):1-10.
37. Canul-Reich J, Hernández-Torruco J, Frausto-Solis J, et al. A Kernel-Based Predictive Model for Guillain-Barré Syndrome[M]. Advances in Artificial Intelligence and Its Applications. 2015.
38. Keretna S, Lim C P, Creighton D, et al. Enhancing medical named entity recognition with an extended segment representation technique.[J]. Computer Methods & Programs in Biomedicine, 2015, 119(2):88-100.
39. Jegadeeshwaran R, Sugumaran V. Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines[J]. Mechanical Systems & Signal Processing, 2015, 52(1):436-446.
40. Abdi L, Hashemi S. To combat multi-class imbalanced problems by means of over-sampling and boosting techniques[J]. Soft Computing, 2016, 19(12):3369-3385.
41. Dietterich T, Bakiri G. Solving multiclass learning problems via error-correcting output codes[J]. Journal of Artificial Intelligence Research, 1995, 2(2):263-286.
42. Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computer and System Sciences, 1997 55(7):119-139.
43. Liu KY, Lin J, Zhou X, Wong ST. Boosting alternating decision trees modeling of disease trait

information[J]. BMC Genetics,2005, 6(6): 1-6.
44. Rocach L. Pattern Classification Using Ensemble Methods [M]. World Scientific Publishing Co, Inc. 2009.
45. Mai Q. A review of discriminant analysis in high dimensions[J]. Wiley Interdisciplinary Reviews Computational Statistics, 2013, 5:190-197.
46. Harsiti, Munandar T, Sigit H. Implementation Of Fuzzy-C4.5 Classification As a Decision Support For Students Choice Of Major Specialization[J]. Computer Science, 2013, 11(2):1577-1581.
47. Lee B K, Jeong EH, Lee S S. Context-Awareness Healthcare for Disease Reasoning Based on Fuzzy Logic[J]. Electr Eng Technol, 2016,11(1):247-256.
48. Cintra M E, Monard M C, Camargo H A. FuzzyDT -- A Fuzzy decision tree algorithm based on C4.5[J]. Mathware & Soft Computing, 2013, 20(1):56–62
49. Povoas d L H, De A C H. A Methodology for Building Fuzzy Rule-Based Systems Integrating Expert and Data Knowledge[C]. 2014 Brazilian Conference on Intelligent Systems (BRACIS). IEEE Computer Society, 2014:300-305.
50. Ribeiro M V, Cunha L M S, Camargo H A, et al. Applying a Fuzzy Decision Tree Approach to Soil Classification[M]. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Springer International Publishing, 2014, 442:87-96.
51. Wang X, Liu X, Pedrycz W, et al. Fuzzy rule based decision trees[J]. Pattern Recognition, 2015, 48(1):50-59.
52. Liu X, Feng X, Pedrycz W. Extraction of fuzzy rules from fuzzy decision trees: An axiomatic fuzzy sets (AFS) approach[J]. Data & Knowledge Engineering, 2013, 84(3):1-25.
53. Kumar A, Hanmandlu M, Gupta H M. Fuzzy binary decision tree for biometric based personal authentication[J]. Neurocomputing, 2013, 99(1):87-97.
54. Pashaei E, Ozen M, Aydin N. Improving medical diagnosis reliability using Boosted C5.0 decision tree empowered by Particle Swarm Optimization.[C]. Conf Proc IEEE Eng Med Biol Soc, 2015.
55. Sivapriya T R, Arnb K, Prj T. Ensemble Merit Merge Feature Selection for Enhanced Multinomial Classification in Alzheimer's Dementia.[J]. Computational & Mathematical Methods in Medicine, 2015, 2015(3):1-11.
56. Chen K H, Wang K J, Wang K M, et al. Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data[J]. Applied Soft Computing, 2014, 24(3):773-780.
57. Güraksın G E, Haklı H, Uğuz H. Support vector machines classification based on particle swarm optimization for bone age determination[J]. Applied Soft Computing, 2014, 24:597-602.
58. Chen H L, Yang B, Wang S J, et al. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy[J]. Applied Mathematics & Computation, 2014, 239(8):180–197.
59. Vosooghifard M, Ebrahimpour H. Applying Grey Wolf Optimizer-based decision tree classifer for cancer classification on gene expression data[C]. International Conference on Computer and Knowledge Engineering. IEEE, 2015.
60. Kar S, Sharma K D, Maitra M. Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K -nearest neighborhood technique[J]. Expert Systems with Applications, 2015, 42(1):612-627.
61. Devi B R, Rao K N, Setty S P. Towards better classification using improved particle swarm optimization algorithm and decision tree for dengue datasets[J]. International Journal of Soft Computing, 2016, 11:18-25.
62. 唐华松, 姚耀文. 数据挖掘中决策树算法的探讨[J]. 计算机应用研究, 2001, 18(8):18-19.
63. 姜欣, 徐六通, 张雷. C4.5 决策树展示算法的设计[J]. 计算机工程与应用, 2003(4):93-97.
64. 杨舒晴. 基于粗糙集的决策树分类算法研究[D]. 江西理工大学, 2009.

65. 郭磊，王亚弟，陈庶樵,等. 一种基于信息度量的流特征遴选算法[J]. 计算机工程，2012, 38(16):96-99.

66. 周剑峰，阳爱民，刘吉财. 基于改进的 C4.5 算法的网络流量分类方法[J]. 计算机工程与应用，2012,48:71-74.

67. 周剑峰，阳爱民，周咏梅,等. 基于二元搭配词的微博情感特征选择[J]. 计算机工程，2014, 40(6):162-165.

68. Yan-yan SONG, Ying LU. 用于分类与预测的决策树分析 [J]. 上海精神医学, 2015, 2 (27):55-59.

69. Borgelt C, Meinl T, Berthold M R. Advanced pruning strategies to speed up mining closed molecular fragments[J], 2004 IEEE International Conference, 2004, 5(5):4565-4570.

70. 郑伟，马楠. 一种改进的决策树后剪枝算法[J]. 计算机与数字工程, 2015,6(6):960-966..

71. Krishnamoorthy S. Pruning strategies for mining high utility itemsets[J]. Expert Systems with Applications, 2015, 42(5):2371-2381.

72. Kennedy J, Eberhart R. Particle swarm optimization[J].Proceedings of the IEEE International Conference on Neural Networks,1995,4(1):1942-1948.

73. James, S, Sochacki. The Modified Picard Method for Solving Arbitrary Ordinary and Initial Value Partial Differential Equations[D]. Virginia:James Madison University, 2008.

74. 任国锋，李德华，潘莹. 一种改进的基尼指数特征权重算法[J]. 计算机与数字工程，2010, 38(12):8-13.

75. 王苗，柴瑞敏. 一种改进的决策树分类属性选择方法[J]. 计算机工程与应用, 2010, 46(8):127-129.

76. 刘文军，谷云东. 属性依赖性及重要性度量[J]. 数学的实践与认识, 2009, 39(7):148-156.

77. Shlens J. A Tutorial on Principal Component Analysis[J]. Eprint Arxiv, 2014, 58(3):219-226.

附录：作者在攻读硕士学位期间发表的论文

1. 黄秀霞，孙力. C4.5 算法优化研究[J]. 计算机工程与设计.2016, 37(5): 1265-1270.
2. 黄秀霞，孙力. 基于属性依赖度计算和 PCA 的 C4.5 算法 [J]. 传感器与微系统. （已录用）