中南财经政法大学

本科生毕业论文（设计）

论文题目	：	关联规则和决策树组合算法在学生成绩分析中的研究
姓　　名	：	廖娴静
学　　号	：	1409030231
班　　级	：	1402 班
年　　级	：	14 级
专　　业	：	信息管理与信息系统
学　　院	：	信息与安全工程学院
指导教师	：	XXX　教 授
完成时间	：	2018 年 4 月 16 日

作者声明

本毕业论文是在导师的指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。因本论文引起的法律结果完全由本人承担。

毕业论文成果归中南财经政法大学所有。

特此声明。

作者专业：

作者学号：

作者姓名：

年　　月　　日

关联规则和决策树组合算法在学生成绩分析中的研究

廖娴静

Research on Association Rules and Decision Tree Combination Algorithm in Student Performance Analysis

Liao, Xian Jing

2018 年 4 月 16 日

摘　要

近年来，大数据引领的风潮不断向各大领域推进，推动了各个领域的发展和变革，教育领域也毫不例外地迈入了大数据时代。教育信息化的快速发展积累了大量的数据，而其中最为重要的就是考试成绩。因此，我们如何利用科学的方法对这些数据进行挖掘与分析，不断革新学生的学习模式、教师的教学模式以及教育政策制定的方法，这个问题引起了广大教育工作者们的关注。

目前，大部分学生成绩分析的研究方向一方面是基于关联规则算法展开的，另一方面是基于决策树算法展开的。尽管关联规则能够挖掘出课程之间的关联性，但是却没有考虑学生、教师自身等个性化因素对学生成绩造成的影响，因此关联规则得到的结果通常在个体分析时有失偏颇。而决策树算法虽然能够实现个体学生成绩的预测，但是由于没有考虑到个体学生的课程间的关联性，因而决策树算法得到的预测结果准确度不高。如果能结合关联规则和决策树两者的优势，弥补两者的劣势，就能实现高效的挖掘分析。

针对目前研究方向的缺陷，本文提出一种高效的关联规则和决策树组合算法，综合考虑学生课程间的关联性和学生、教师自身等个性化因素，以期提高学生成绩分析结果的准确性。

首先，本文基于本校信息与安全工程学院信管专业的学生课程成绩设计以学生成绩为主题的数据仓库，为后续的成绩分析提高可靠的数据支持。其次，利用关联规则算法挖掘分析课程间的关联性，并生成用于构造决策树的新属性。最后，通过信息增益率的思想将生成的新属性和原有属性构造成决策树，实现学生成绩分析预测。

综上所述，本文提出的组合算法应用于学生成绩分析是可行的，其得到的分析预测结果更全面，个体课程成绩预测更为准确，具有一定的实用性。

关键词：数据挖掘；关联规则；决策树；成绩分析；数据仓库

Abstract

In recent years, the trend which big data leads has continued to advance in all major areas, promoting development and changes in various fields, and the educational field has entered the era of big data without exception. The rapid development of educational informatization has accumulated a large amount of data, and the most important one is the test scores. Therefore, how do we use scientific methods to mine and analyze these data, and constantly innovate students' learning modes, teachers' teaching modes, and educational policy-making methods? This issue has become the focus of educators.

At present, most of the research on student achievement data mining is based on association rule algorithm or decision tree algorithm. Although the association rules can excavate the relevance between courses, it does not consider the impact of individual factors such as students and teachers on scores. Therefore, the results obtained by association rules often have an inaccuracy in individual analysis. However, although the decision tree algorithm can achieve the prediction of individual scores, the accuracy of the prediction results obtained by the decision tree algorithm is not high because the relevance between individual students' courses is not taken into account. If we can combine the advantages of both the association rules and the decision tree to make up for the disadvantages of both, we can achieve efficient mining analysis.

In view of the shortcomings of the current research direction, this paper proposes an efficient combination algorithm of association rules and decision trees, which comprehensively considers the correlation between students' courses and individual factors such as students and teachers themselves, in order to improve the accuracy of student performance analysis results.

First of all, this paper designs a data warehouse based on students' scores as a result of the student's course performance of the College of Information and Security Engineering College of Information Management so that we can provide reliable data support for subsequent scores' analysis. Secondly, the association rule algorithm is used to mine and analyze the correlation between courses and generate new attributes for constructing a decision tree. Finally, through the idea of information gain rate, the generated new attributes and original attributes are constructed into a decision tree to achieve students' scores analysis and prediction.

In summary, the combination algorithm proposed in this paper is applicable to the analysis of students' scores. The analysis and prediction results obtained by it are more comprehensive, and the individual course performance is more accurate and has certain practicality.

Key words: Data mining; Association rules; Decision tree algorithm; Performance analysis; Data warehouse

目　录

一、绪论

（一）选题背景

近年来，大数据引领的风潮不断向各大领域推进，推动了各个领域的发展和变革，教育领域也毫不例外地迈入了大数据时代。教育信息化的快速发展积累了大量的数据，这些数据不仅包括考试成绩，还包括学生生活、社会实践、教学过程中产生的多方面数据。在教育领域中，数据是教学改进中最为显著的指标，而其中最为重要的就是考试成绩。因此，我们如何利用科学的方法对这些海量数据进行挖掘与分析，不断革新学生的学习模式、教师的教学模式以及教育政策制定的方法，这个问题引起了广大教育工作者们的关注。

随着大数据时代的到来，对海量数据进行处理并获取新知识的数据挖掘技术也成为了热门话题。数据挖掘是指，运用基于计算机的新方法对大型数据库中的信息数据进行挖掘分析，得到未知的、有价值的结果或模式的过程。数据挖掘本身融合了统计学、数据库技术和机器学习等多门学科，目前主要包括统计技术、概念描述、关联规则、决策树、基于历史的分析、遗传算法、聚集检测、连接分析、回归分析、差别分析、神经网络、粗糙集、模糊集等十三种常用的数据挖掘的技术。数据挖掘就是通过运用以上所述的某种或某几种技术相结合的方法实现对海量数据的处理。

通过科学的数据挖掘或更深层次的挖掘，我们可以获得每一个学生的学习需求、风格、态度甚至是学习模式，从而提供适合不同学生的课程与学习指导，实现个性化教育。比如说，每一个学生都会有为其量身定做的学习课程系列与学习指导模式，并且通过对学生前期课程的成绩的分析可以预测后期课程的成绩优劣，从而可以创建一个成绩预警系统，每当学生预测成绩出现下降时，就对其进行提示，以敦促其加强学习。

然而，一方面，目前大部分高校依然采用的是教师对学生成绩数据的简单人工分析的方法来进行科学教育改进，这显然没有充分发挥成绩的分析评估对教学改进的作用。另一方面，由于影响学生成绩的因素众多，不仅包括课程间的关联性，还包含学生、教师自身等个性化因素，致使目前即便某些高校采用了大数据分析方法也无法达到成绩预测应有的准确度。在信息化不断革新的今天，我们必须要学会使用科学的数据分析找到影响学生成绩的主要因素与潜藏因素，实现对学生成绩的更为准确的预测与预警，进而推动课程的优化设置，提高教学质量，这就对一种创新且有效的成绩分析方法提出了要求。

（二）国内外研究现状

1. 国外研究现状

国外对于数据挖掘的研究起步较早，大概在 20 世纪 60 年代就开始了相关研究。并且，在国外，本文即将展开讨论的基于数据挖掘算法的学生成绩管理这一方向也是研究的热点方向。数据挖掘中很多经典算法例如关联规则、逻辑回归和决策树算法等均已应用于国外各大高校的教学管理研究中，比如说

预测大学入学比例、预测学生毕业状况、学生选课推荐等。

在关联规则的研究方向上，1994 年相关研究人员关联规则的经典算法，即 Apriori 算法[1]。随后，国外的研究重点逐渐集中在频繁项集[2]上，S. Bria 提出了动态项集算法，该算法思想为动态评估已被记录的项集支持度，动态体现在不断检查项集的所有子集是否均为频繁项集，若是的话则被确立为新的候选集。S. S. gua[3]等人在频繁项集的求解上进行了创新，创新之处在于利用了仿生学中的遗传算法原理。R. J. Ku[4]等相关研究人员使用粒子群算法优化关联规则，并且利用蚁群算法寻找事务集中的频繁项集，支持度阀值和置信度阀值的选取[5]。

在决策树算法的研究方向上，1967 年，Gordon B. K 博士提出用于分类性数据挖掘的 CHART 算法[6]。1984 年，可处理连续型变量的 CART 算法[7]得到推广。1993 年，Quinlan 在专著《Machine learning specification》中提出对 ID3 算法的改进，即 C4.5 算法[8]。

在学生成绩分析应用的研究方向上，美国哈佛大学的研究人员 Jody Clarke 与 Chris Daye 为了评估学生学习效果，利用数据挖掘技术分析与研究了学习行为相关的海量数据。亚利桑那州立大学的相关研究人员采用决策树算法研究了影响调查问卷回收率的主要因素。近些年，国外在学生成绩研究方向上的研究核心技术还是在关联规则和决策树算法上。比如，J. S. Park[9]等人提出了利用并行计算进行数据挖掘的 PDM 算法。R. Agrawal[10]等人在算法空间开支和执行效率上进行了考虑，提出了关联算法并行挖掘的三个算法，分别为候选集分布算法、数据分布模式算法、频度分布算法。这些改进后的算法使得学生成绩分析技术的功能范围不断扩大，结果预测准确度也大大提高。

2. 国内研究现状

直到 20 世纪 90 年代中期，我国才逐步开始研究数据挖掘技术，相较于西方发达国家，不仅时间上落后，而且国内科研力量也更薄弱，难以整合在一起，这是因为国内科研工作者缺乏交叉学科背景，通常都是学习背景较为单一的高校计算机老师，而数据挖掘技术却是一门多学科交叉的技术。此外，比起应用研究，国内更注重与理论的研究，当然，国内的实际应用工作也在逐渐起步。

在关联规则的研究方向上，有柴华昕等研究人员提出了 Napriori 算法，其算法思想为压缩事务集和候选集，从而达到减小算法空间开支的效果，是 Apriori 算法的改进算法。吴振光[11]等研究人员提出 Apriori 改进算法，其创新之处在于减少事务集中项集的重复扫描。盛立[12]等研究人员也提出了 Apriori 改进算法，其算法思想为改进 Apriori 算法剪枝过程，加入限制条件，也即项集中的元素必须大于某个特定的 K 值。

在决策树算法的研究方向上，2012 年，王章恩[13]等人采用决策树算法提高对学生成绩的分析能力。同年，邝继红[14]采用 C4.5 算法分析学生成绩并对其准确度进行评估。2013 年，傅亚莉[15]利用据册书算法分析学生成绩中各课程的内在关联性。2014 年，龙钧宇[16]采用 k-均值聚类算法划分成绩等级并且运用 C4.5 算法构建决策树的过程中，得出了学生的单个课程成绩与总评成绩的关联性。丁勇、武玉艳通过分类预测实验，证明决策树有较高的预测准确率。

在学生成绩分析应用的研究方向上，董欢[17]、刘志妩[18]等研究人员利用决策树算法建立分析预测模型完成了对成绩较为准确的预测。付希[19]、刘美玲[20]等研究人员根据聚类挖掘结果对学生成绩等级

进行动态划分，使得评价结果更为客观。

（三）论文的主要研究内容与意义

国内目前在学生成绩分析研究方向上的欠缺，主要是由两方面造成的。一方面，尽管国内的学生成绩应用研究也在逐渐起步，但比起应用研究，国内还是更注重与理论的研究，因而尽管有大量的理论支持，实现上还是有一定的困难，但这方面的欠缺只能通过不断发展应用研究来弥补，短时间内无法攻克。另一方面，影响学生成绩的因素众多，不仅包括课程间的关联性，还包含学生、教师自身等个性化因素，因此即便某些高校采用了大数据分析方法也无法达到成绩预测应有的准确度，这方面的欠缺则可以通过提出一种创新且有效的成绩分析方法来解决。

本论文通过对决策树算法和关联规则算法的深入研究，提出一种将两者结合的解决方案。一则可以获取学生各课程间的关联性，扩宽分析预测结果的覆盖面；二则可以提高成绩预测的准确度，这是因为各课程间的关联度对某单科成绩必然存在影响。同时，考虑到对学生课程成绩的数据处理宇存储，作者决定采用数据仓库技术，既可以实现有效储存，又可以为后续的成绩分析提高可靠的数据支持。

本论文的主要工作围绕以下三个方面：

（1）在对学生成绩进行数据挖掘分析之前，进行数据仓库的设计与建立，其中包括数据的处理过程。

（2）利用关联规则算法建立挖掘模型，主要用于挖掘课程间的关联性。

（3）采用关联规则和决策树算法相结合的方法，实现学生成绩的分析预测。

在信息化不断革新的今天，我们必须要学会使用科学的数据分析找到影响学生成绩的主要因素与潜藏因素，实现对学生成绩的更为准确的预测与预警，进而为高校教育的各个方面提供服务技术和支持，比如说推动课程的优化设置，提高教学质量等等。

（四）论文的架构

全文总共五章，其具体结构与内容如下：

第一章为绪论。本章首先介绍论文的选题背景，然后简单阐述该课题的国内外研究现状，主要介绍关联规则、决策树算法和学生成绩分析研究三个研究方向，最后介绍本论文的主要研究内容与意义以及论文的结构安排。

第二章为数据仓库的基础内容介绍和学生成绩数据仓库的设计。首先对数据仓库的相关概念、特点、体系结构、设计步骤做了具体介绍，然后再围绕学生成绩分析这一主题完成对学生成绩数据仓库的设计，包括概念模型设计、逻辑模型设计和物理模型设计，最后实现数据加载。

第三章为数据挖掘技术综述。首先介绍数据挖掘的相关概念、功能和流程，然后分别具体介绍关联规则与决策树算法，其中包含概念、经典算法思想以及优缺点。

第四章为关联规则和决策树组合算法在学生成绩分析中的应用。首先简单介绍关联规则和决策树组合算法，然后阐述关联规则在学生成绩分析的研究与应用，即产生决策树算法中的属性，最后阐述关联

规则与决策树组合算法在学生课程成绩分析中的应用研究。

第五章为总结与展望。总结论文所做的科研工作以及仍存在的问题，最后提出未来的改进和研究方向。

二、数据仓库在成绩分析中的应用与设计

本章首先介绍数据仓库的基础内容，包括数据仓库的相关概念、特点、体系结构、设计步骤。然后再围绕学生成绩分析这一主题完成对学生成绩数据仓库的设计，包括概念模型设计、逻辑模型设计和物理模型设计。最后完成对数据仓库的数据加载。

（一）数据仓库

1. 基本概念

数据仓库是指，一个面向主题的、集成的、随时间变化的、非易失性的数据集合，用于支持经营管理中的决策制订过程。这是 1991 年数据仓库之父 Bill Inmon 在《Building the Data Warehouse》一书中提到的。

在作者的理解中，数据仓库是一个过程而不是一个项目，是一个环境而不是一个产品。数据仓库技术是为了用户能更方便查询到需要的信息以支持其决策，将有效数据集成到一个统一的稳定的环境中。

（1）主题（Subject）

主题就是指用户进行数据挖掘分析时需要的关键信息，主题有两个元素，一是分析角度，也称维度，二是要分析的具体量度。举例来说，假设用户需要某年某月某学校学生某科不合格的数据，那么维度应包括时间、地点、课程，而要分析的具体量度应为该校某科不合格学生数，这是因为量度最通常的表现形式就是数值。

（2）维度（Dimension）

维度是主题两个关键元素之一，指的是分析角度，也就是从不同角度考察事物的特征。通常来说，维度都有多个层级，而每一个层级都具有属性，这些属性可以是公有的，也可以是特有的。举例来说，家庭地址这一维度会有国家、省、市、县等层级，而这些层级都会有编号、名称、描述这几个公共属性。当然这些公共属性不局限于地址的维度，也可以用比如学生、商品等维度。

（3）量度

量度就是我们要分析的具体的技术指标，一般为数值型数据，比如某学科学生不及格数。这些用来汇总、求独立出现次数或者取最值等的一类数据称为量度。

（4）事实表

在通常的关系型数据库中，不存在事实表和维度表的区别，这是因为关系型数据库的用途主要是记录、查询和增删改数据，它对数据本身所含有的业务含义并不在意。而在数据仓库中要区别事实表

和维度表是为了分析数据，具体来说是为了关注数据与数据之间的业务含义。在数据仓库中，每个维度表通常通过一个索引字段与事实表相连。

事实表是数据仓库主题描述信息的集合表，包含了主题事实的度量信息、联系维度表的键以及用于标识唯一数据的主键 ID，其实事实表一般是没有主键的，但是为了方便统一管理数据，可以设置数据的序号 ID 为主键。综上所述，事实表中存储的信息是用来挖掘分析的目标数据的全量信息，简单来说就是用户关注的内容。

（5）维度表

维度表则是对事实的某一分析角度的描述信息的集合，这些描述信息也可称之为属性，简单来说不同维度表对应着用户分析该事务的不同角度。

2. 数据仓库的特点

（1）面向主题性，即数据仓库都是围绕某个明确主题来建立的，因此，其他与主题无关的细节数据会被排除掉。

（2)集成性,即数据仓库的数据来自多个不同的数据源,此过程中涉及到数据仓库的 ETL(extract、transform、load)过程。

（3）反映历史变化，即数据仓库中的关键数据会显式或隐式地随时间变化，用户可以利用仓库中从过去某一时点到现在的数据进行分析与预测工作。

（4）非易失性，即信息本身相对稳定，由于决策分析过程中使用的数据必须是大量积累的历史数据，而存储这些海量历史数据的数据仓库需要一个稳定的存储环境。具体来说是指，一旦数据装入这个环境后，一般只具有查询的权限而不具有增删改的权限，并且，这个环境通常只要定期的加载、刷新。

3.数据仓库的 ETL 过程

ETL 过程是建立面向某特定主题的数据仓库环境的重要环节,它包括抽取(extract)、转换(transform)及加载(load)三部分。数据抽取，是指将来自多个不同的数据源集成起来。数据转换，是指对集成的数据进行一系列处理，比如数据合并、转换、过滤、清洗等方式。数据加载，是指将数据转换后的数据统一存储到数据仓库中。

4. 数据仓库的体系结构

数据仓库的体系结构如图 2-1 所示，分为信息获取层、信息存储层、信息传递层[1]。其中，信息获取层上实现 ETL，信息存储层上实现存储功能，信息传递层上实现用户与数据仓库的交互功能。

[1]注：图中的数据集市，也称为部门数据、主题数据，是指为了特定的应用而从数据仓库中独立出来的一部分数据。

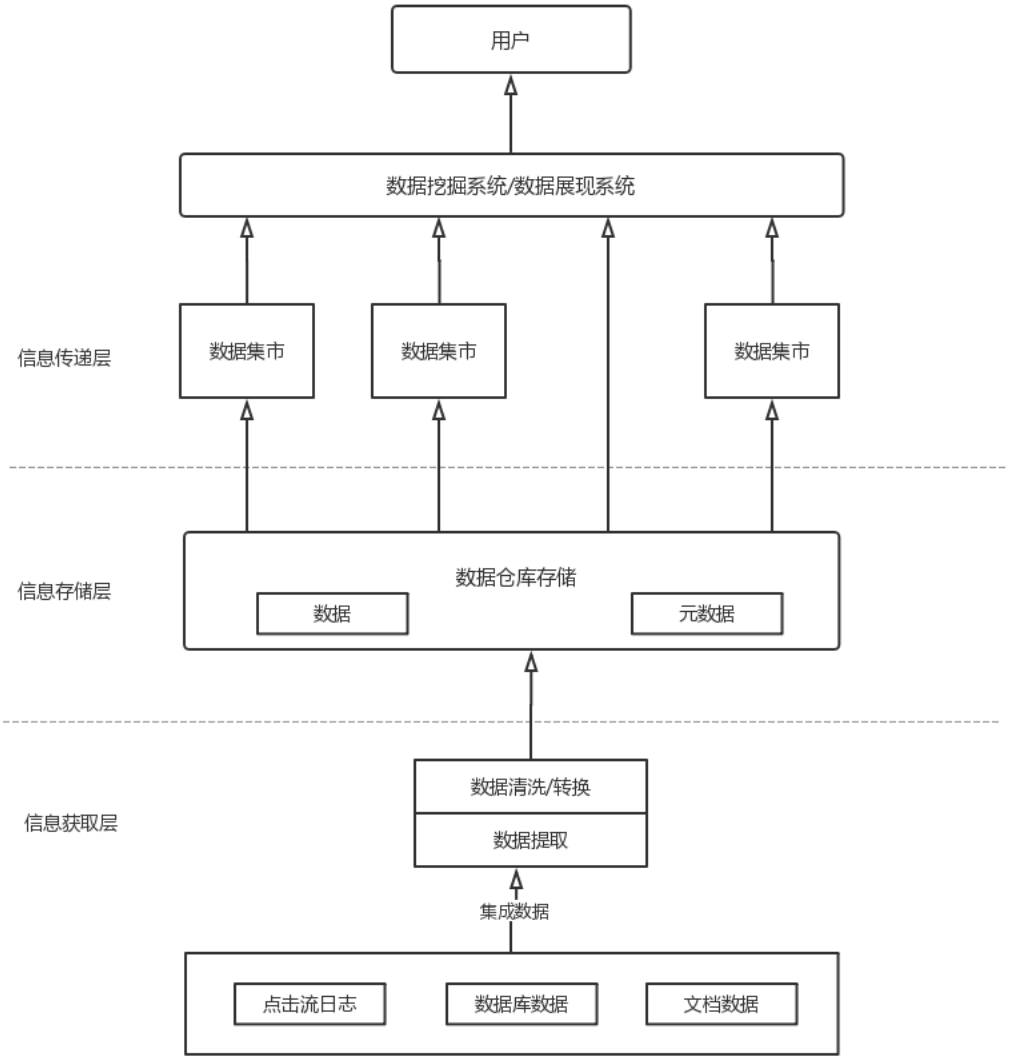


图 2-1 数据仓库体系结构图

5. 数据仓库的设计

数据仓库的设计流程如下：

（1）收集和分析业务需求来确定主题

（2）建立数据模型（概念模型、逻辑模型）和数据仓库的物理设计

（3）定义数据源

（4）选择数据仓库技术和平台

（5）对数据源进行 ETL 操作并将数据存储到数据仓库中

（6）选择访问和报表工具

（7）选择数据库连接软件

（8）选择数据分析和数据展示软件

（9）更新数据仓库

（二）学生成绩数据仓库概述

本论文决定采用数据仓库支持下文的数据挖掘，原因有三。一是考虑到数据仓库的面向主题性与非易失性，面向主题便于用户快速查询其需求的信息，而非易失性则保证了数据在数据库中的稳定存储。二是为了充分利用本校信管专业历年来积累的海量学生、教师、课程和成绩的相关数据，这是因为，一方面，数据仓库能够反映历史变化，有利于随时间变化的历史数据的存储；另一方面，学生成绩不但能够反映学生的学习状态，而且能够反映课程安排得合理与否以及教师的授课水平等多方面的情况。三是采用数据仓库技术能够实现对数据的 ETL 操作，即数据抽取、数据转换、数据加载等操作，这样能够实现多方面的数据集成。综上所述，采用数据仓库技术，能够充分挖掘学生成绩中的价值信息，为本校的教务管理提供数据支持。

前文已经介绍了数据仓库技术的基本概念，那么接下来将介绍的是如何实现学生成绩数据仓库的搭建，为下文的数据挖掘提供合适的数据集。众所周知，数据仓库设计最为核心的部分就是模型设计，包括概念模型设计、逻辑模型设计和物理模型设计。下文本论文将从这几个方面一一设计，最后再完成数据加载。

（三）学生成绩数据仓库的概念模型设计

我们知道，概念模型设计是现实世界到信息世界的第一层抽象，它是面向用户的数据模型，也是面向现实世界的数据模型，而并未涉及 DBMS 的一些技术性问题，典型的概念模型有 E-R 模型，本论文也采用的是 E-R 模型。因此，在概念模型设计过程中，我们要做的工作包含两个部分，一是确定主题域并划分各个主题域的边界，二是构建 E-R 模型。

1. 确定主题域

前文对数据仓库的介绍中已经提到，数据仓库是面向主题的，即按照主题域模型进行组织的[2]。一方面，考虑到学生成绩不但能够反映学生的学习状态，而且能够反映课程安排得合理与否以及教师的授课水平等多方面的情况；另一方面，考虑到本校信息与安全工程学院信管专业历年来积累的学生成绩的相关数据量庞大，作者决定利用学生、教师、课程、专业等多方面的数据构建学生成绩数据仓库。

（1）事实

前文对数据仓库的介绍中已经提到，事实表用来挖掘分析的目标数据的全量信息，故事实指的是用户决策时使用的目标数据。因此，在学生成绩数据仓库中，事实指的是学生成绩。

（2）维度

前文对数据仓库的介绍中已经提到，维表则是对事实表中事件的要素的描述信息，或者称之为属性，也就是用户观察该事务的角度，记录的一般为基本属性信息。因此，在学生成绩数据仓库中，维度指的是事实的不同考察角度，即学生信息维、专业信息维、课程信息维和教师信息维。具体见下表 2-1。

---

[2] 说法引用自 ClaudiaImhoff, NicholasGalemmo, JonathanG.Geiger. 数据仓库设计:relational and dimensional techniques[M]. 机械工业出版社，2004.

表 2-1 学生成绩数据仓库维度信息表

	维度			
类别	学生维度	专业维	课程维	教师维
	性别	专业分类	课程名称	性别
	成绩分组	班级名称	课程性质	职称
	上机时间		课程类型	学历分组
			学年学期	教龄分组
度量指标：分数等级				

综合以上的分析，本论文确定了学生成绩数据仓库的主题域，它们分别为学生、班级、课程、教师、成绩。

2. E-R 模型

对已确定的主题域进行细化，明确出实体相互之间的关系，从而得出学生成绩分析的 E-R 模型，如图 2-2 所示。

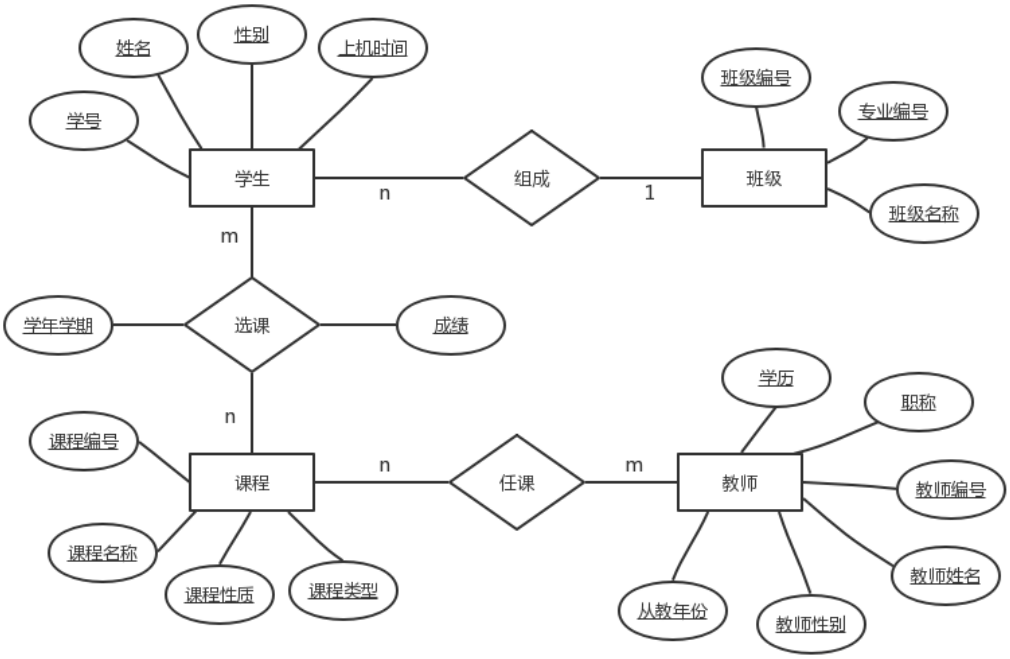


图 2-2 学生成绩分析的 E-R 模型

（四）学生成绩数据仓库的逻辑模型设计

分析第三节得到的 E-R 模型，可以得到如表 2-2 所示的关系模式：

表 2-2 学生成绩分析关系模式

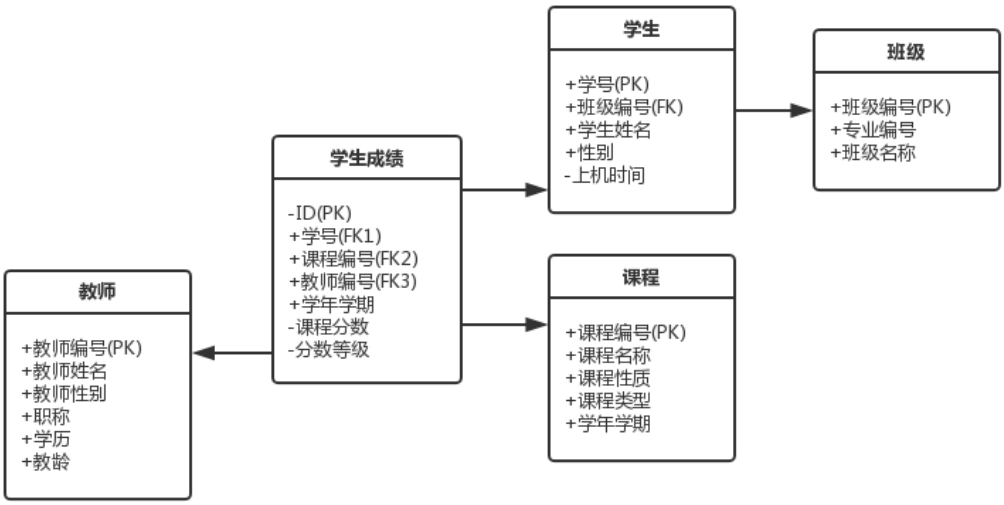
关系名	属性组	码	外码
学生	学号，姓名，性别，上机时间，班级编号	学号	班级编号
班级	班级编号，专业编号，班级名称	班级编号	-
课程	课程编号，课程名称，课程性质，课程类型[3]	课程编号	-
教师	教师编号，教师姓名，教师性别，职称，学历，从教年份	教师年份	-
选课	课程编号，学号，学年学期，成绩	（课程编号，学号）	课程编号，学号

由于学生数据仓库是面向学生成绩这个主题的，所以事实表肯定是学生成绩表，并且事实表是由其余四个表集成的，故在关系模式上进一步进行数据集成得到学生成绩表。综上，得到学生成绩分析主题的详细描述如表 2-3 所示：

表 2-3 学生成绩分析主题的描述

关系名	属性组	公共键
学生	学号，姓名，性别，上机时间，班级编号	学号
班级	班级编号，专业编号，班级名称	班级编号
课程	课程编号，课程名称，课程性质，课程类型	课程编号
教师	教师编号，教师姓名，教师性别，职称，学历，教龄	教师年份
学生成绩	ID，课程编号，教师编号，学号，学年学期，分数，分数等级[4]	ID

综上，得出基于学生成绩分析为主题的数据仓库逻辑模型，如图 2-3 所示。



[3] 课程性质指的是必修和专业限选的不同，课程类型指的是基础理论课程、专业课程和通识课程的不同。

[4] 学生成绩表这一事实表是由其他表集成而来，故拥有其余表的主键作为外键。同时为了便于 C4.5 算法的运行，该表也对分数进行了离散化处理。

图 2-3 学生成绩分析逻辑模型

（五）学生成绩数据仓库的物理模型设计

数据仓库物理模型设计实际上就是确定数据存储的具体实现形式。论文的物理模型设计包括两个阶段，一是选择开发工具，二是数据表设计。以下将对这两个阶段一一介绍并设计。

1. 选择开发工具

本论文选择 Microsoft SQL server2008 R2 作为数据仓库开发的工具。

2. 数据表设计

根据图 2-3 的逻辑模型得知，学生成绩数据仓库主要包括学生信息维度表(student)、班级信息维度表(class)、课程信息维度表(course)、教师信息维度表(teacher)和学生成绩事实表(score)。如下：

表 2-4 学生信息维度表(student)

字段名称	数据类型	描述
学号	char(20)	主键
姓名	char(10)	-
性别	char(2)	男、女
上机时间	char(10)	每周大于三小时、每周小于三小时
班级编号	char(10)	-

表 2-5 班级信息维度表(class)

字段名称	数据类型	描述
班级编号	char(10)	主键
班级名称	char(10)	-
专业编号	char(10)	-

表 2-6 课程信息维度表(course)

字段名称	数据类型	描述
课程编号	char(10)	主键
课程名称	char(10)	-
课程性质	char(4)	必修课、限选
课程类型	char(10)	专业课、基础理论课程、通识课
学年学期	date	来源于学生成绩数据库

表 2-7 教师信息维度表(teacher)

字段名称	数据类型	描述
教师编号	char(10)	主键
教师姓名	char(10)	-

教师性别	char(2)	男、女
教龄	int	-
职称	char(4)	讲师、副教授 、教授
学历	char(4)	本科、硕士、博士

表 2-8 学生成绩事实表(score)

字段名称	数据类型	来源	描述
ID	char(40)	-	主键
学号	char(20)	来自 student 表	-
课程编号	char(10)	来自 course 表	-
教师编号	char(10)	来自 teacher 表	-
学年学期	date	来源于学生成绩数据库	-
课程分数	float	来源于学生成绩数据库	-
分数等级	char(1)	根据课程成绩进行分类	A/B/C/D/E

（六）学生成绩数据仓库的数据加载

设计好数据仓库后，接下来的工作就是对数据进行 ETL 过程。

1. 数据抽取

数据抽取，指的是将来自多个不同的数据源集成起来的过程。由于本论文构建的是学生成绩数据仓库，因此抽取的是本校信息与安全工程学院教务系统中学生、教师、课程等相关数据。

2. 数据清洗和数据转换

数据清洗和数据转换的主要用途在于当数据集成的过程中出现数据不一致的问题时，可用该方法来进行数据修正。本论文中主要的处理如下：

（1）丢失或错位数据的处理

在数据集成时可能会出现字段丢失或者错位的问题，在本论文中，主要处理方法为先进行关联匹配，无法匹配时按照缺失处理。比如说，数据源中的学生某科成绩丢失数据，若通过关联方式匹配无果，则按照缺失处理，删去该行成绩。

（2）格式修正

格式修正主要是为了确保不同表中的同一属性这一类数据的规范性。举例来说，在数据源中，课程信息表中的“课程编号”，在学生成绩表中可能是“课程代码”，这时可将两表中的字段统一为“课程编号”。

（3）数据离散化

本论文采用的 C4.5 算法无法处理连续型数据，而课程分数却是连续型的参数，因此本论文需要对课程分数进行一般的离散化，成绩分组分为 A>=90，90>B>=80，80>C>=70，70>D>=60，E<60 这五个

档，如图 2-4 所示。

c++		离散数学		数据库系统原理	
87	B	70	C	80	B
80	B	73	C	66	D
85	B	83	B	95	A
88	B	66	D	80	B
90	A	80	B	86	B
68	D	77	C	80	B
88	B	80	B	82	B
73	C	92	A	96	A
78	C	73	C	81	B
90	A	67	D	81	B
65	D	58	E	66	D
80	B	77	C	96	A
82	B	86	B	80	B
62	D	60	D	77	C
74	C	92	A	88	B
82	B	88	B	94	A
94	A	89	B	91	A
79	C	83	B	79	C
75	C	72	C	80	B
76	C	73	C	92	A

图 2-4 数据离散化实例

3. 数据加载

数据加载，是指将数据转换后的数据统一存储到数据仓库中。本文中，数据加载过程是将处理后的数据统一存储到学生成绩数据仓库中。

三、数据挖掘技术综述

（一）数据挖掘

1. 基本定义

数据挖掘(Data Mining)，是指从海量模糊的、有噪声的随机数据中，分析提取出隐含在其中的有用的信息与知识的过程。这种知识发现的过程必须通过一定的算法才能实现，因此在数据挖掘的核心就是算法或者模型。

2. 数据挖掘与数据仓库的关系

在第二章中已经提到，数据仓库是一个由多个数据源集成的海量数据集合，并且具有面向主题性和非易失性。数据挖掘所用到的海量数据正是依赖于这样一个稳定的历史数据存储环境，换句话说，建立在数据仓库上的数据挖掘才能提供更准确、有效的分析结果。因此，数据仓库与数据挖掘之间的关系可以简单地概括为被依赖与依赖或者支持与被支持。

3.数据挖掘的功能

数据挖掘的功能可以分为 6 类，对比分析见下表 3-1。

表 3-1 数据挖掘的功能简述表

功能	简述
分类	分类是指，通过事例的其余属性对类别的模式进行确定的方法，即对物品或抽象事物分类。决策树算法就是典型的分类算法之一。
聚类	聚类也是把物品划分至不同的类别，但与分类不同，它的目标类别是未知的，即从数据中挖掘未知的类别并添加至已挖掘出的类别中。
关联	关联是指，从海量数据中挖掘各项之间的关联性。典型的关联算法有 Apriori 算法。
回归	回归也类似于分类，但它是借助查找来确定数值的方法。典型的回归有逻辑回归和线性回归。
序列分析	序列是指一系列事件，因而序列分析则是用来发现这一系列事件中的模式的方法。注：序列可以包含离散的状态。
偏差分析	偏差分析是指，寻找观测结果与参照值之间有意义的差别的方法。

4.数据挖掘的流程

数据挖掘不是一个从模型到结果的简单数据流程，而是一个周期性和逐步细化的过程。一般来说，数据挖掘由以下步骤组成：

（1）确认研究对象，即理解研究对象的目标与需求，设计出达到目标的初步计划。

（2）模型的选择，即选择一个合适的数据挖掘算法。

（3）数据采集，即收集相关数据。

（4）数据预处理，即将原始数据构造成最终适用于数据挖掘分析的数据，包括数据抽取、数据清洗、数据转换、数据格式化等方式。

（5）数据挖掘，即根据选择的挖掘算法对处理后的数据进行挖掘分析。

（6）结果分析，即解释并评估结果。评价的方法通常因应用类型而异。

（7）知识的使用，即将已开发的知识应用于实际的决策过程。 这个阶段可以是简单生成一个分析报告的过程，也可以是一个复杂的、可重复的数据挖掘过程。

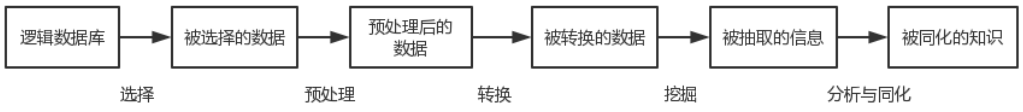


图 3-1 数据挖掘流程图

（二）关联规则

关联分析，是指从一个数据集中发现各个项之间的隐藏关联性与规律性。举例来说，我们平时逛淘宝的时候，如果将一条腰部松紧设计条纹衬衫裙子加入购物车，那么它很快会推荐你可能还会感兴趣的

连衣裙，比如说印花系带连衣裙，这其实就是著名的购物篮分析。

1.基本概念

关联规则，是指从海量数据中挖掘出满足置信度的关联关系，形如蕴涵式$A \rightarrow B$，其中，A、B 均为数据集合，A 代表蕴涵式先导，B 代表蕴涵式后继，且 A、B 的交集不为空。关联规则的典型数学模型是“支持度—置信度模型”。

在关联规则中，有四个重要的元素值得介绍一下。

（1）项集

项集，简单来说，就是项的集合。设$A = \{A_1 + A_2 + \cdots + A_s\}$，其中 A 表示项集，$A_s$表示项。

（2）事务集

事务集，简单来说，是若干个事物的集合。而事物是由几个相关项的集合组成的集合。关联规则由事务集组成。设$D = \{T_1 + T_2 + \cdots + T_s\}$，则 D 表示由若干个事物 T 组成的一个事务集。

（3）支持度

设 A、B 都事务集 D 中的一个事件，那么在事务集中出现 A 的概率叫做 A 的支持度，表示为$Sup(A)$，在事务集中同时出现 A、B 的概率称之为$A \rightarrow B$关联关系支持度，表示为$Sup(A \rightarrow B)$。为便于理解，介绍一下表达公式：

$$Sup(A) = P(A) = \left|\left\{\frac{t|t \in A}{|D|}\right\}\right| \tag{3-1}$$

$$Sup(A \rightarrow B) = P(AB) = \left|\left\{\frac{t|t \in A \cap t \in B}{|D|}\right\}\right| \tag{3-2}$$

注：$|D|$表示集合中的元素个数。

（4）置信度

满足（3）的假设，在事务集中，在出现 A 的条件下， B 也会出现的概率称之为置信度。表达公式如下：

$$conf(A \rightarrow B) = P(B|A) = \frac{P(AB)}{P(A)} = \frac{|\{t|t \in A \cap t \in B\}|}{|\{t|t \in A\}|} \tag{3-3}$$

2.强关联规则

强关联规则，实际上是指支持度和置信度不小于最小支持度和最小置信度水平的关联规则。

关联规则挖掘的执行步骤一般分为两个步骤：

（1）找出所有支持度大于等于最小支持度的频繁项集合。

（2）找出置信度大于等于最小置信度的强关联规则。

3.Apriori 算法

Apriori 算法支持“支持度—置信度模型”，故也包含数据集、事务集、支持度和置信度四个要素。值得注意的是，Apriori 算法是采用 K 项集来产生 K+1 项集，即通过自连接 K 项集获取 K+1 项集，但是两个 K 项集进行连接有一个条件，是它们至少有 K-1 项相同。

Apriori 算法基本思想为：

（1）从事务集中找出符合最小支持度的频繁项集。

（2）基于前一步的频繁集进行挖掘，得到一个候选 2 项集，除去其中不满足最小支持度的项，得到 2 项频繁项集，迭代挖掘，直到最终生成的频繁项集为空时停止，最终得到所有的频繁项集。

4.Apriori 算法缺点

本论文并没有对 Apriori 算法进行改进，故只对其缺点简单阐述。

（1）生成候选集时重复扫描数据库

（2）生成的候选集庞大

（3）算法挖掘模式固定

（三）决策树算法

1.决策树的基本定义

在众多数据挖掘算法中，决策树算法是一种典型的分类算法。它可以从一组无次序、无规则的样本数据中推出决策树的表示形式。目前，比较典型的决策树算法有 ID3 算法、C4.5 算法、CHAID 算法、CART 算法。由于本论文采用的是 C4.5 算法，因此仅对 C4.5 算法进行阐述。在 C4.5 算法中有几个重要的要素：

（1）信息熵

1948 年，香农引入信息熵这一概念，并将其定义为离散随机事件出现的概率。设一个随机变量 X 的取值为$\mathrm{X}=\{x_1,x_2,\cdots x_n\}$，而每一种取值的概率又分别是$\{p_1,p_2,\cdots p_n\}$，则 X 的熵定义为

$$\mathrm{H(X)}=-\sum_{i=1}^{n}p_i\log_2 p_i \tag{3-4}$$

（2）信息增益

信息增益是针对单个特征而言的，对比系统有它和没有它时的信息量的差值，这个差值就是这个单个特征给系统带来的信息量，即信息增益。设 D 为一个数据集，D_i为属于第 i 类的样本，S 表示数据的类属性个数，则表达公式如下：

$$\mathrm{G(D,S)}=\mathrm{H(D)}-\sum P(D_i)H(D_i) \tag{3-5}$$

（3）信息增益率

信息增益率，实际上是 C4.5 算法对 ID3 算法中的信息增益计算的改进。改进之处在于，信息增益率用来计量相同的划分所获得的信息，克服了信息增益选择特征时偏向于特征值个数较多的不足。属性 S 的信息增益率的表达公式如下：

$$\text{GainRatio}(\text{D},\text{S}) = \frac{Gain(D,S)}{SplitInfo(D,S)} \tag{3-6}$$

其中 Gain(S,A)就是 ID3 算法中的信息增益，而

$$SplitInfo(D,S) = -\sum_{i=1}^{n} \frac{|S_i|}{|D|} \log_2 \frac{|D|}{|S_i|} \tag{3-7}$$

其中，S_i到S_n是特征 S 的 n 种不同取值构成的样本集。

2.C4.5 算法

由于 C4.5 算法用文字表述起来比较复杂，本论文采用流程图的方式展示 C4.5 算法。

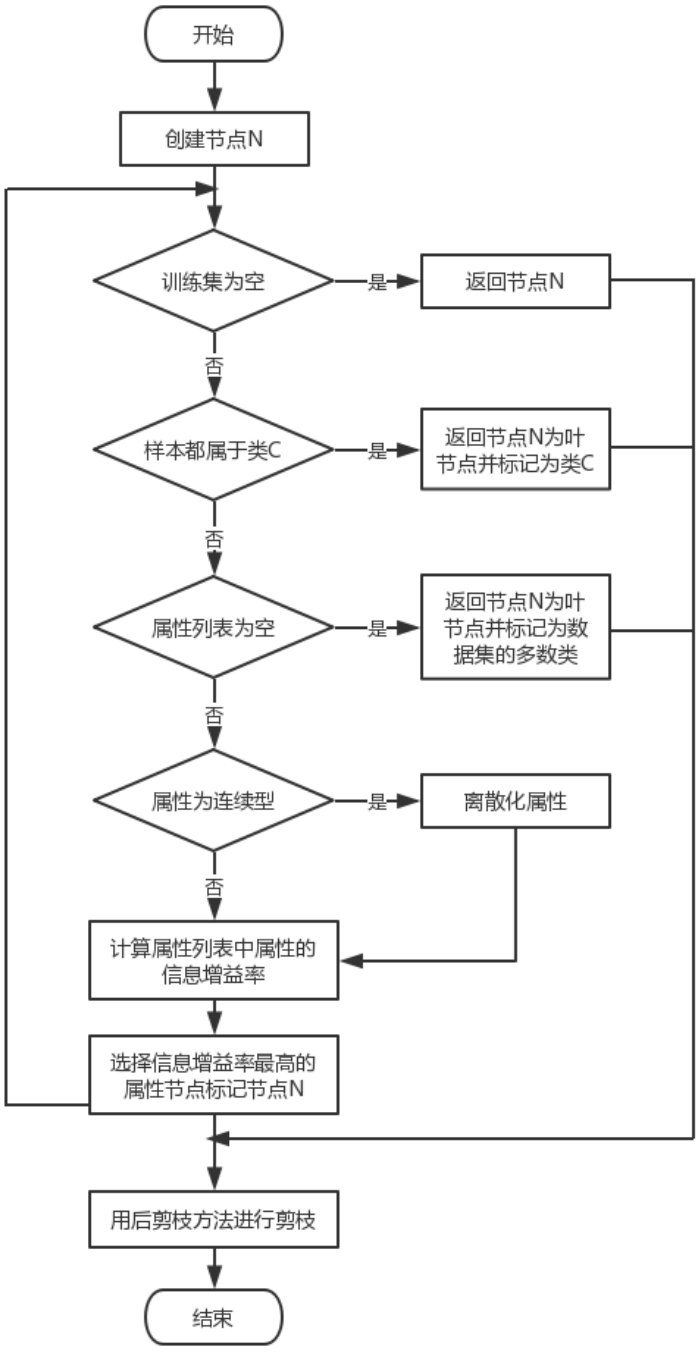


图 3-2 C4.5 算法流程图

3.C4.5 算法优缺点

（1）优点

C4.5 算法可以处理数据不完整和连续型属性（通过离散化）的数据集，而且建模快，分类正确率比较高。

（2）缺点

从公式 3-7 可以看出，C4.5 算法涉及大量的对数运算，算法时间开销增大，而且从流程图 3-2 可以看出，在建立决策树的过程中，计算机必须重复依次扫描对应的数据集并逐个排序。

（四）其他数据挖掘算法

数据挖掘算法除了关联规则和决策树算法，还包括许多其他算法，比如聚类分析、回归分析、神经网络、遗传算法等挖掘算法。但本论文用到的算法只有关联规则和决策树算法，对其它挖掘算法就不再做阐述。

四、关联规则和决策树组合算法在学生成绩分析中的应用

众所周知，学生的课程成绩的优异是反映一所学校教学质量好坏的重要指标，因而学生成绩分析能为高校教育的改进提供服务和数据支持。但是，高校通常存在庞大的学生历史成绩数据库，缺乏关联性和逻辑性的数理统计方法难以实现对海量数据的挖掘分析。因此，对于数据挖掘在学生课程成绩的应用这一研究方向成了教育工作者们关注的焦点。

然而，尽管学生成绩分析对高校教学改进意义非凡，但是目前学生成绩分析算法却不够理想。这是因为，一方面，目前大部分高校依然采用的是缺乏关联性和逻辑性的数理统计方法来分析学生成绩，而即便采用了数据挖掘算法，也只是简单的利用关联规则进行分析课程之间的关联性或者利用决策树进行学生成绩的预测。这样的简单应用远远达不到学生成绩分析应有的效果，无法充分发挥成绩的分析对教学改进的作用。另一方面，由于影响学生成绩的因素众多，不仅包括课程间的关联性，还包含学生、教师自身等个性化因素，现有的一些数据挖掘算法得到分析结果在准确性上还有待提高。

因此，本文经过对学生成绩分析的深入研究，提出了一种高效且针对性强关联规则和决策树组合算法，以期实现高校教育改进上三个维度的作用。第一维度是学生，学生可根据自己现有的成绩预测后续课程中有不合格风险的课程，提高后续学习中的针对性。第二维度是教师，教师可根据学生成绩分析结果对不同学生制定合适的教学模式与方法，提高教学质量，比如说某学生的某一学科出现不合格现象，那么教师要注重该生的后续课程中与该不合格学科关联的学科的教学，可以适当多布置课后作业等。第三维度是教务管理人员，教务管理人员可根据学生成绩分析结果了解课程之间的关联性，制订更为优良的教学计划，从而提高学生成绩和教学质量。

（一）关联规则和决策树组合算法概述

1. 提出组合算法的理论依据

目前，大部分学生成绩数据挖掘的研究一方面是基于关联规则算法展开的，另一方面是基于决策树算法展开的。尽管关联规则能够挖掘出课程之间的关联性，但是却没有考虑学生、教师自身等个性化因素对学生成绩造成的影响，因此关联规则得到的结果通常在个体分析时有失偏颇。而决策树算法虽然能够实现个体学生成绩的预测，但是由于没有考虑到个体学生的课程间的关联性，因而决策树算法得到的预测结果准确度不高。如果能结合关联规则和决策树两者的优势，弥补两者的劣势，就能实现高效的挖掘分析。

综上，本文提出一种高效的关联规则和决策树组合算法，综合考虑学生课程间的关联性和学生、教师自身等个性化因素，提高学生成绩分析结果的准确性。

2. 算法设计思路概述

该组合算法设计思路就是利用关联规则生成高可信度的强关联规则作为新属性，然后通过对新属性的判别后，最终得到新的属性与原有属性合并来构造决策树。该组合算法采用关联规则的经典算法——Apriori 算法和决策树算法的典型算法——C4.5 算法。该组合算法思想如下：

（1）通过 Apriori 算法挖掘学生成绩数据内隐藏的关联规则，再选取其中的强关联规则，即支持度、置信度都大于等于最小支持度、最小置信度的关联规则。

（2）判别（1）得到的关联规则，将其中不合理的关联规则去除，得到新的关联规则集合。注：这里判别方法主要为判别关联规则的课程之间的学年学期是不是逆序，若是则要去除掉该条关联规则。

（3）将（2）所得的关联规则作为新的分类属性与原有属性合并，利用 C4.5 算法构造决策树。

（二）关联规则在学生成绩分析中的应用

这一节介绍的关联规则在学生成绩分析中的应用，是根据前文算法思想概述中提到的，通过 Apriori 算法挖掘学生成绩数据内隐藏的关联规则，再选取其中的强关联规则，即支持度、置信度都大于等于最小支持度、最小置信度的关联规则。同时，本文根据第三章的数据挖掘综述中提到的数据挖掘流程进行分析。具体的数据挖掘实施流程图如图 4-1。

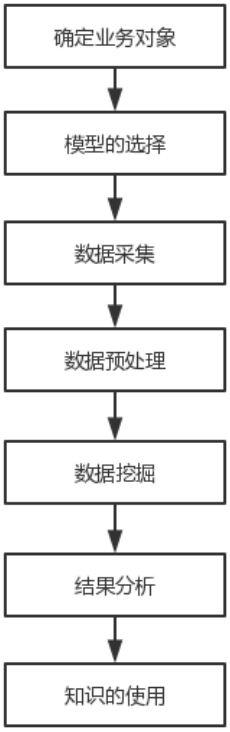


图 4-1 数据挖掘实施流程图

1.确定业务对象

本次数据挖掘的分析对象就是第二章中设计的学生成绩数据仓库。

2. 模型的选择

本次数据挖掘采用 Apriori 算法。

3. 数据采集

本次数据挖掘中用到的数据来源于本校信息与安全工程学员信管专业的学生成绩，如图 4-2 所示。

1	xh	xm	bj	xl	nj	kcmc	xqmc	kclbmc	ksxzmc	zcj	xf	kcxzmc
3575	1409030231	廖娴静	信管1402	女	2014	信息管理导论	2014-2015-1	必修	正常考试	75	2	必修
3576	1409030231	廖娴静	信管1402	女	2014	公共体育（1）	2014-2015-1	必修	正常考试	86	1	必修
3577	1409030231	廖娴静	信管1402	女	2014	军事理论	2014-2015-1	必修	正常考试	78	2	必修
3578	1409030231	廖娴静	信管1402	女	2014	军训	2014-2015-1	必修	正常考试	通过	2	必修
3579	1409030231	廖娴静	信管1402	女	2014	大学英语（1）	2014-2015-1	必修	正常考试	83	4	必修
3580	1409030231	廖娴静	信管1402	女	2014	实用信息技术	2014-2015-1	必修	正常考试	85	4	必修
3581	1409030231	廖娴静	信管1402	女	2014	形势与政策（1）	2014-2015-1	必修	正常考试	90	0.5	必修
3582	1409030231	廖娴静	信管1402	女	2014	程序设计语言	2014-2015-1	必修	正常考试	92	5	必修
3583	1409030231	廖娴静	信管1402	女	2014	高等数学（上）	2014-2015-1	必修	正常考试	74	5	必修
3584	1409030231	廖娴静	信管1402	女	2014	公共体育（2）	2014-2015-2	必修	正常考试	86	1	必修
3585	1409030231	廖娴静	信管1402	女	2014	大学英语（2）	2014-2015-2	必修	正常考试	85	4	必修
3586	1409030231	廖娴静	信管1402	女	2014	大学语文	2014-2015-2	必修	正常考试	75	3	必修
3587	1409030231	廖娴静	信管1402	女	2014	形势与政策（2）	2014-2015-2	必修	正常考试	92	0.5	必修
3588	1409030231	廖娴静	信管1402	女	2014	微机原理	2014-2015-2	必修	正常考试	81	4	必修
3589	1409030231	廖娴静	信管1402	女	2014	思想道德修养与法律基础	2014-2015-2	必修	正常考试	90	2	必修
3590	1409030231	廖娴静	信管1402	女	2014	数据结构	2014-2015-2	必修	正常考试	80	5	必修
3591	1409030231	廖娴静	信管1402	女	2014	毛泽东思想和中国特色社会	2014-2015-2	必修	正常考试	95	3	必修
3592	1409030231	廖娴静	信管1402	女	2014	离散数学	2014-2015-2	必修	正常考试	91	4	必修
3593	1409030231	廖娴静	信管1402	女	2014	马克思主义基本原理概论	2014-2015-2	必修	正常考试	81	3	必修
3594	1409030231	廖娴静	信管1402	女	2014	高等数学（下）	2014-2015-2	必修	正常考试	73	5	必修
3595	1409030231	廖娴静	信管1402	女	2014	Windows编程	2015-2016-1	限选	正常考试	86	3	限选

图 4-2 课程成绩部分原始数据

4. 数据预处理

数据预处理是一个工作量大且关键的一步。只有精准的数据预处理，才能为后续数据挖掘提供有效的数据集。此处预处理的详细流程为：

（1）数据集成

数据集成是指，围绕学生成绩这一主题汇集来自多个不同数据源的数据，如图 4-3 所示。

1	学号	性别	上机时间	c++	高等数学（上）	数据结构	离散数学	高等数学（下）
2	1409030101	女	每周大于三小时	87	70	87	70	80
3	1409030102	男	每周大于三小时	80	76	77	73	79
4	1409030103	女	每周大于三小时	85	61	87	83	75
5	1409030104	男	每周不大于三小时	88	87	83	66	77
6	1409030106	女	每周大于三小时	90	75	84	80	70
7	1409030107	女	每周不大于三小时	68	71	79	77	76
8	1409030109	女	每周大于三小时	88	89	93	80	66
9	1409030111	男	每周大于三小时	73	90	90	92	86
10	1409030112	女	每周大于三小时	78	81	74	73	70
11	1409030113	女	每周大于三小时	90	81	76	67	76
12	1409030115	男	每周不大于三小时	65	61	67	58	61
13	1409030116	女	每周大于三小时	80	68	83	77	81

图 4-3 集成后的数据

（2）数据清洗

数据清洗是指，对原始数据进行筛选检查，包括检查数据一致性，处理残缺数据、错误数据、重复数据。由于本次采集的数据来源于教务部，缺失、错误和重复数据很少，因而数据清洗工作量不大。

（3）数据消减

数据消减是指，在不影响最终的数据挖掘结果的前提下，缩小所挖掘数据的规模。举例来说，从图 4-2 可看出，原始数据中包括军训这一项数据，但这与学生成绩分析没有关联，像这种类型的数据就是数据消减的对象。

（4）数据转换

由于 Apriori 算法是 bool 类型的算法，而学生成绩却是连续型参数，因此本文考虑将课程成绩转换为离散数据形式，即 bool 类型数据（0、1 形式）。具体方法为，首先求出课程的平均成绩，若大于该课程平均成绩则离散化为 1，若小于该课程平均成绩则离散化为 0，这种离散化的方式考虑到了各个课程评判标准不一致的问题。图 4-4 是转换后的数据。

学号	c++	高等数学（上）	数据结构	离散数学	高等数学（下）	数据库系统原理	概率论与数理统计	线性代数
1409030101	1	0	1	0	1	0	0	1
1409030102	0	0	0	0	1	0	0	0
1409030103	1	0	1	1	0	1	1	1
1409030104	1	1	1	0	1	0	1	0
1409030106	1	0	1	1	0	1	0	0
1409030107	0	0	1	1	0	0	0	0
1409030109	1	1	1	1	0	0	1	0
1409030111	0	1	1	1	1	1	1	0
1409030112	0	1	0	0	0	0	0	1
1409030113	1	1	0	0	0	0	0	1
1409030115	0	0	0	0	0	0	0	0
1409030116	0	0	1	1	1	1	1	1
1409030118	0	1	1	1	0	0	1	0
1409030119	0	0	1	0	0	0	1	0
1409030120	0	1	1	1	0	1	1	0
1409030121	0	1	1	1	1	1	1	1

图 4-4 转换后的数据

5. 数据挖掘

此处以部分课程与专业综合设计这门课程的关联性为例做详细阐述。在具体实现时可以挖掘任意多门课程之间的关联性。

令 L1 为课程程序设计语言（C++）中高于平均分的成绩记录的集合，L2 为课程高等数学（上）中高于平均分的成绩记录的集合，L3 为课程数据结构中高于平均分的成绩记录的集合，L4 为课程离散数学中高于平均分的成绩记录的集合，L5 为课程高等数学（下）中高于平均分的成绩记录的集合，L6 为课程数据库系统原理中高于平均分的成绩记录的集合，L7 为课程概率论与数理统计中高于平均分的成绩记录的集合，L8 为课程线性代数中高于平均分的成绩记录的集合，L9 为课程计算机网络原理中高于平均分的成绩记录的集合，L10 为课程建模语言中高于平均分的成绩记录的集合，L11 为课程网络编程（java）中高于平均分的成绩记录的集合，L12 为课程专业综合设计中高于平均分的成绩记录的集合。

假设最小支持度为 28，最小置信度为 0.55，最小支持度和置信度都是人定的，可以根据实验结果的优劣对这两个参数进行调整。

第一轮候选集和剪枝的结果为：

Candidate set:

L1：56

L2：52

L3：51

L4：55

L5：46

L6：54

L7：44

L8：51

L9：51

L10：49

L11：55

L12：49

After pruning:

L1：56

L2：52

L3：51

L4：55

L5：46

L6：54

L7：44

L8：51

L9：51

L10：49

L11：55

L12：49

可以看到，第一轮时，其实就是用的数据集中的项。而因为最小支持度是 28 的缘故，所以没有被剪枝的，所以得到的频繁集就与候选集相同。

第二轮这里做了简化，因为本例中只探讨单科与专业综合设计这门课程的关联性，故只需考虑 Li（i=l,…,l1）与 L12 合集的频度，即 P（Li∪L12）。

第二轮候选集和剪枝的结果为：

Candidate set:

L1，L12：37

L2，L12：28

L3，L12：29

L4，L12：37

L5，L12：27

L6，L12：33

L7，L12：29

L8，L12：30

L9，L12：28

L10，L12：26

L11，L12：27

After pruning:

L1，L12：37

L2，L12：28

L3，L12：29

L4，L12：37

L6，L12：33

L7，L12：29

L8，L12：30

L9，L12：28

可以看到，第二轮的候选集就是第一轮的频繁集自连接得到的（进行了去重），然后根据数据集频度计算得到支持度，与最小支持度比较，过滤了一些记录，频繁集已经与候选集不相同了。通过这两轮候选集的支持度计算，就可以得出置信度：

Confidence(L1->L12)=support(L1∪L12)/support(L1)=37/56=0.6607

Confidence(L2->L12)=support(L2∪L12)/support(L2)=28/52=0.5385

Confidence(L3->L12)=support(L3∪L12)/support(L3)=29/51=0.5686

Confidence(L4->L12)=support(L4∪L12)/support(L4)=37/55=0.6727

Confidence(L6->L12)=support(L6∪L12)/support(L6)=33/54=0.6111

Confidence(L7->L12)=support(L7∪L12)/support(L7)=29/44=0.6591

Confidence(L8->L12)=support(L8∪L12)/support(L8)=30/51=0.5882

Confidence(L9->L12)=support(L9∪L12)/support(L9)=28/51=0.5490

由于最小置信度为 0.55，故筛选得到高可信度强关联规则，如表 4-1 所示。

表 4-1 高可信度强关联规则

序号	规则	支持度	置信度
1	C++->专业综合设计	1.0000	0.6607
2	离散数学->专业综合设计	1.0000	0.6727
3	数据库系统原理->专业综合设计	1.0000	0.6111
4	概率论与数理统计->专业综合设计	1.0000	0.6591
5	线性代数->专业综合设计	1.0000	0.5882

6. 结果分析

结果分析是指，通过对数据挖掘得到的信息进行进一步研究分析，将挖掘结果解释为易理解的理论

结果。对于第五步中得到的高可信度强关联规则，可得出以下分析结果：

（1）在以上 11 门专业课程中，课程《C++》、《离散数学》、《数据库系统原理》、《概率论与数据统计》和《线性代数》对课程《专业综合设计》的学习影响较大，同时这些课程的学习在前有利于《专业综合设计》的学习。这说明教学计划中将这些课程安排在课程《专业综合设计》之前是十分正确的。

（2）课程《概率论与数理统计》、《线性代数》虽然是公共课，但是置信度却高于一些专业课程的置信度，因此说明《概率论与数理统计》、《线性代数》也是本专业的基础知识储备课程，不但要早学，还要学好，为下面课程的学习打下良好的基础。

7. 知识的使用

本次数据挖掘得到的知识主要用于本文中决策树的构建，同时也能说明课程之间的关联性，为本校信息与安全工程学院的教学计划提供数据支持。

（三）关联规则与决策树组合算法在学生课程成绩分析中的应用

1.确定业务对象

本次数据挖掘的分析对象就是第二章中设计的学生成绩数据仓库。

2. 模型的选择

该组合算法采用关联规则的经典算法——Apriori 算法和决策树算法的典型算法——C4.5 算法。该组合算法思想如下：

（1）通过 Apriori 算法挖掘学生成绩数据内隐藏的关联规则，再选取其中的强关联规则，即支持度、置信度都大于等于最小支持度、最小置信度的关联规则。

（2）判别（1）得到的关联规则，将其中不合理的关联规则去除，得到新的关联规则集合。

（3）将（2）所得的关联规则作为新的分类属性与原有属性合并，利用 C4.5 算法构造决策树。

该组合算法的第一步在第二节已经完成。为了便于理解第二、三步的文字，本文使用流程图进行阐述，如图 4-5 所示。从图 4-5 可以看出，Apriori 算法得出的强关联规则只要符合规则前件课程学期小于规则后件课程的学期，即可生成新属性，并与原有属性合并生成决策树。

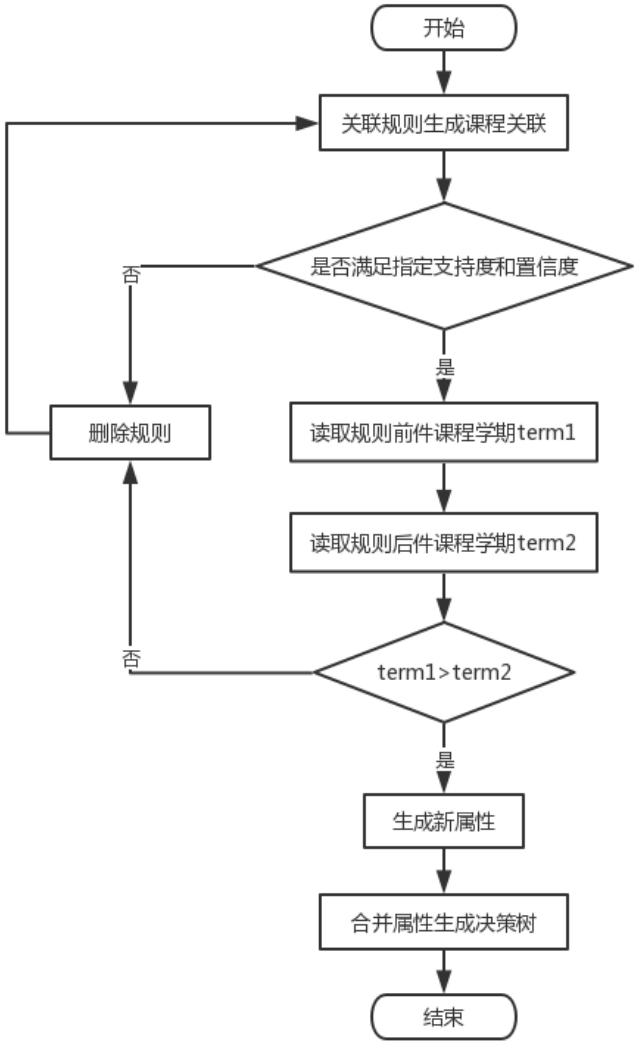


图 4-5 关联规则与决策树结合流程图

3. 数据采集和预处理

本次数据挖掘中用到的数据来源于本校信息与安全工程学员信管专业的学生成绩，在第二节已经完成。而数据预处理主要是指课程分数离散化，专业综合设计是要判断类别的属性，故只有两种取值，即合格与不合格，如图 4-6 所示。

c++←>专业综合设计	离散数学->专业综合设计	数据库系统原理->专业综合设计	概率论与数理统计->专业综合设计	线性代数->专业综合设计	专业综合设计
B	C	B	C	B	合格
B	C	D	D	D	不合格
B	B	A	C	B	合格
B	D	B	C	D	不合格
A	B	B	D	C	合格
D	C	B	D	C	不合格
B	B	B	C	D	合格
C	A	A	C	C	合格
C	C	B	D	B	合格
A	D	B	D	B	不合格
D	E	D	D	E	不合格
B	C	A	B	B	合格
B	B	B	B	C	不合格
D	D	C	C	C	不合格
C	A	B	C	C	不合格

图 4-6 学生课程成绩离散化

4. 数据挖掘

（1）确定分析属性，原有属性包括性别、上机时间；关联规则生成的新属性包括 C++->专业综合设计、离散数学->专业综合设计、数据库系统原理->专业综合设计、概率论与数理统计->专业综合设计、线性代数->专业综合设计。为了体现课程关联性，生成的新属性名称会变成“xx->专业综合设计”，但取值其实仍是原课程成绩离散值。

（2）利用候选属性生成决策树

通过 Python 编写 C4.5 算法[5]对预处理后的数据集进行处理并生成图 4-7 所示的结果。该结果显示的是决策树的每条路径，故将所有路径合并后可还原得到决策树，如图 4-8。

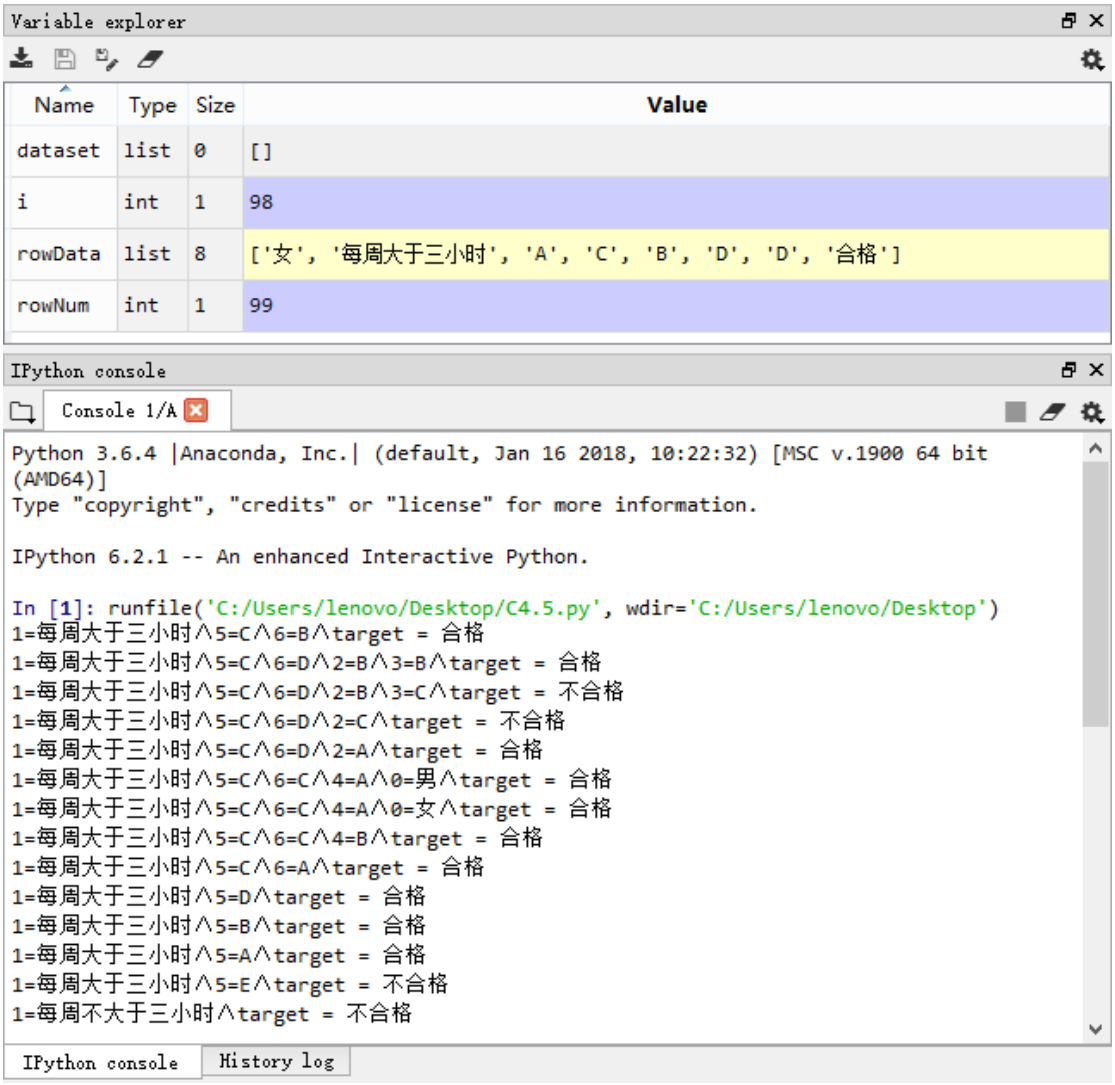


图 4-7 运行结果

[5] 代码见附录。

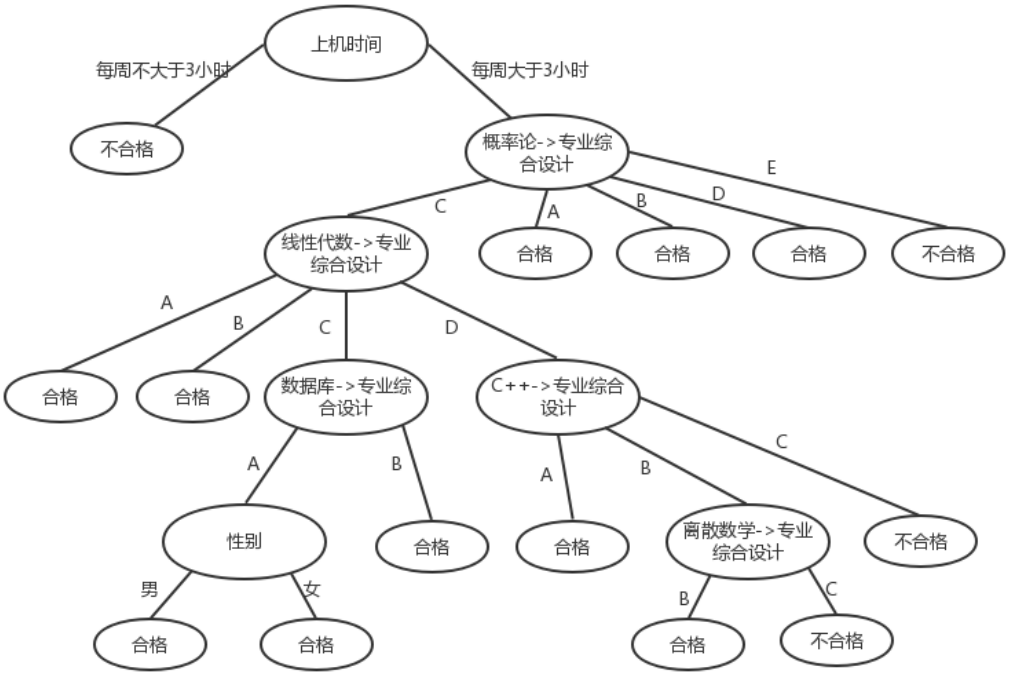


图 4-8 生成的决策树

5. 结果分析

对于第四步数据挖掘得到的结果进行分析，得到以下分析结果：

（1）决策树的第一个选择属性是上机时间的分类，说明上机时间是第一个主要的影响因素，这与平常的经验判断所得结果是吻合的。对于信管专业的学生来说，上机时间至少要保证每周大于三小时，这样才能提高自身实践能力，才能保证专业课程达到合格的水平。

（2）对于信管专业来说，性别对专业课程的影响不大，可以看到无论男生女生，当数据库系统原理这门课程达到 A 时，专业综合设计均是合格的。

（3）对于信管专业来说，数学类的课程对专业课程影响很大，比如概率论与数理统计、线性代数和离散数学对专业综合设计的影响。这说明数学公共课和计算机专业基础数学（离散数学）是信管专业的基础知识储备课程，不但要早学，还要学好，这样才能为后续课程的学习打下良好的基础。

由于数据挖掘的样本不够大、组合算法不够完善，分析结果一定会存在一定的误差，但是相较于单纯基于关联规则或者决策树算法的学生成绩分析来说，本文的准确度还是有一定的提高。

6. 知识的使用

本次数据挖掘得到的知识可以用于预测学生成绩，当输入决策树属性的取值时，即可预测该生某一课程是否能够合格。本文这一节只是拿《专业综合设计》这门课程举例，但在建立好学生成绩数据仓库后，关联规则和决策树组合算法的应用可以通过任何目前所学课程进行预测任何后续课程合格与否。因此在学生这一维度，学生可根据自己现有的成绩预测后续课程中有不合格风险的课程，提高后续学习中

的针对性；在教师这一维度，教师可根据学生成绩分析结果对不同学生制定合适的教学模式与方法，提高教学质量，比如说某学生的某一学科出现不合格现象，那么教师要注重该生的后续课程中与该不合格学科关联的学科的教学，可以适当多布置课后作业等；在教务管理人员这一维度，教务管理人员可根据学生成绩分析结果了解课程之间的关联性，制订更为优良的教学计划，从而提高学生成绩和教学质量。

五、总结与展望

本论文通过对决策树算法和关联规则算法的深入研究，提出一种将两者结合的算法，即关联规则和决策树组合算法，并探讨了该算法在学生成绩分析中的应用与研究。一则可以获取学生各课程间的关联性，扩宽分析预测结果的覆盖面；二则可以提高成绩预测的准确度，这是因为各课程间的关联度对个体某单科成绩必然存在影响。同时，考虑到对学生课程成绩的数据处理宇存储，本文决定采用数据仓库技术，既可以实现有效储存，又可以为后续的成绩分析提高可靠的数据支持。

本论文的研究工作主要围绕以下三个方面：

（1）在对学生成绩进行数据挖掘分析之前，设计与搭建以学生成绩为主题的数据仓库，其中包括数据的预处理过程。

（2）利用关联规则算法挖掘分析课程间的关联性，并生成用于构造决策树的新属性。

（3）通过信息增益率的思想将生成的新属性和原有属性构造成决策树，实现学生成绩分析预测。

本论文预期实现三个维度的作用。在学生这一维度，学生可根据自己现有的成绩预测后续课程中有不合格风险的课程，提高后续学习中的针对性；在教师这一维度，教师可根据学生成绩分析结果对不同学生制定合适的教学模式与方法，提高教学质量，比如说某学生的某一学科出现不合格现象，那么教师要注重该生的后续课程中与该不合格学科关联的学科的教学，可以适当多布置课后作业等；在教务管理人员这一维度，教务管理人员可根据学生成绩分析结果了解课程之间的关联性，制订更为优良的教学计划，从而提高学生成绩和教学质量。

通过对学生成绩分析这一研究方向的研究，本论文初步提出了数据仓库和关联规则、决策树算法这两个数据挖掘算法在该研究方向中的应用。然而，由于本人能力上的不足，本文的研究还存在需要进一步完善的地方。一是，本文的实例样本不够大，仅限于信管专业 14 级的学生，因此分析结果存在一定误差，但这并不意味着理论上的错误。二是，本文仅采用了关联规则和决策树两种数据挖掘算法，存在一定的局限性，后期希望能加入更多的挖掘算法，不断提高分析结果的准确性。三是，本文的研究范围不够大，只局限于课程成绩的分析，虽然加入了上机时间这一属性，但研究范围还是远远达不到本校信息与安全工程学院教育领域的范围，后期希望能加入更多的属性，提高分析结果的准确度，并且挖掘出更多有价值的规则。

主要参考文献

[1] J.Brachmaa, T.Anand.The Process of Knowledge Discovery in Databases. A Human-entered Approach, 1998:37-58.

[2] 梁盾.数据挖掘算法与应用[M]. 北京:北京大学出版社，2006,158-170.

[3] S.S Gua. Application of genetic algorithm and weighted item set for association rule mining. Department of Industrial Engineering and Management, 2002:2007-2016.

[4] R.J.Kuo, C.W.Shih. Association rule ming through the ant colony system for National Health Insurance research Databases in Taiwan. Computers and Mathematic with Apllication,2007:1303-1318.

[5] R.J.Kuo, C.W.Chao. Application of particle swam optimization to association rule ming. Aplied Soft Computer.2009:326-336.

[6] U. M. Fayyad, K. B.Irani, On the handing of continuous. Valued attributes in decision tree generation. Machine Learning, 1992, 8:87-105.

[7] L. Breiman, J. Friedman, R. Olshen. Classification and Regression Trees. Monterey, CA:Wadsorth International Group, 1984.

[8] Quinlan J R. C4.5: Programs for machine learning .Morgan Kauffman. 1993.

[9] Kuu-Ming Yu, Jia yi Zhou. Parallel Tid-based frequent Pattern ming algorithm on a PC Cluster and grid computer system. Expert System with Application, 2010:2486-2494.

[10] 尚学群，沈君毅. 并行关联规则挖掘综述[J]. 计算机工程, 2004,30(14):1-3

[11] 吴振光. 一个改进的关联规则频繁项集数据挖掘算法[J]. 计算机科学, 2007,34(9):145-147

[12] 盛立, 刘希玉. 挖掘关联规则中 Apriori-Tid 算法改进[J]. 山东师范大学学报(白然科学版),2005, 20(4): 20-22

[13] 王章恩. 学生成绩管理系统的设计与实现[D]. 成都:电子科技大学, 2014.

[14] 邝继红. 数据挖掘在教务系统成绩分析中的应用研究[D]. 长春:吉林大学, 2012.

[15] 傅亚莉. 数据挖掘技术 C4.5 算法在成绩分析中的应用[J]. 重庆理工大学学报, 2013, 11:35-36

[16] 龙钧宇. 基于均值聚类和决策树算法的学生成绩分析[J]. 计算机与现代化，2014, 6(11):40-42

[17] 董欢. 决策树技术在高校学生成绩分析中的应用研究[D]. 西安电了科技大学，2012.

[18] 刘志妩. 基于决策树算法的学生成绩的预测分析[J]. 计算机应用与软件，2012(11):312-314

[19] 付希. 基于蚁群算法的聚类分析在学生成绩评价中的应用研究[D]. 西南交通大学，2013.

[20] 刘美玲, 李熹, 李永胜. 数据挖掘技术在高校教学与管理中的应用[J]. 计算机工程与设计, 2010, 31(5):1130-1133.

[21] 胡在林. 关联规则和决策树组合算法在学生成绩分析中的应用与研究[D]. 青岛理工大学, 2017.

[22] 梁啸. 基于数据挖掘的高校学生成绩预警技术的研究[D]. 武汉理工大学, 2014.

[23] 陈国林. 基于决策树的高校成绩管理与分析系统的设计与实现[D]. 河北科技大学, 2016.

[24] 黄秀霞. C4.5 决策树算法优化及其应用[D]. 江南大学, 2017.

[25] 黄爱辉. 基于决策树算法的考试成绩分析系统的研究与开发[D]. 湖南大学, 2008.

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

[27] 马丹. 基于数据挖掘技术的学生成绩分析系统的设计与实现[D]. 吉林大学, 2015.

[28] 姚文迪. 基于关联规则算法的数据挖掘在高校成绩中的研究与应用[D]. 西南交通大学, 2015.

[29] Mitidieri E, Pokhozhaev S I. Apriori estimates and blow-up of solutions to nonlinear partial differential equations and inequalities[J]. Proceedings of the Steklov Institute of Mathematics, 2014, 4(8):3-383.

[30] ClaudiaImhoff, NicholasGalemmo, JonathanG.Geiger. 数据仓库设计:relational and imensional techniques[M]. 机械工业出版社, 2004.

[31] 李岚. 基于数据仓库的学生成绩分析与研究[D]. 北京交通大学, 2014.

附录

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr  4 16:06:13 2018

@author: Administrator
"""

class Node:
    '''''Represents a decision tree node.

    '''
    def __init__(self, parent = None, dataset = None):
        self.dataset = dataset # 落在该结点的训练实例集
        self.result = None # 结果类标签
        self.attr = None # 该结点的分裂属性ID
        self.childs = {} # 该结点的子树列表，key-value pair: (属性attr的值，对应的子树)
        self.parent = parent # 该结点的父亲结点


def entropy(props):
    if (not isinstance(props, (tuple, list))):
        return None

    from math import log
    log2 = lambda x:log(x)/log(2) # 计算经验熵
    e = 0.0
    for p in props:
        e -= p * log2(p)
    return e


def info_gain(D, A, T = -1, return_ratio = False):
    '''''特征A对训练数据集D的信息增益 g(D,A)

    g(D,A)=entropy(D) - entropy(D|A)
            假设数据集D的每个元组的最后一个特征为类标签
    T为目标属性的ID，-1表示元组的最后一个元素为目标'''
```

```
    if (not isinstance(D, (set, list))):
        return None
    if (not type(A) is int):
        return None
    C = {} # 结果计数字典
    DA = {} # 属性A的取值计数字典
    CDA = {} # 结果和属性A的不同组合的取值计数字典
    for t in D:
        C[t[T]] = C.get(t[T], 0) + 1  #统计目标属性各种取值下的个数，用户经验熵的计算
        DA[t[A]] = DA.get(t[A], 0) + 1  #统计属性列下各种取值的个数，用于计算经验条件熵
        CDA[(t[T], t[A])] = CDA.get((t[T], t[A]), 0) + 1
        #统计（属性列，目标列）下各种组合取值的个数，例如（女，合格）（男、合格）（）

    PC = map(lambda x : x / len(D), C.values()) # 类别的概率列表
    entropy_D = entropy(tuple(PC)) # map返回的对象类型为map，需要强制类型转换为元组


    PCDA = {} # 特征A的每个取值给定的条件下各个类别的概率（条件概率）
    for key, value in CDA.items():
        a = key[1] # 特征A的取值
        pca = value / DA[a]
        PCDA.setdefault(a, []).append(pca)

    condition_entropy = 0.0
    for a, v in DA.items():
        p = v / len(D)
        e = entropy(PCDA[a])
        condition_entropy += e * p  #计算经验条件熵

    if (return_ratio):
        return (entropy_D - condition_entropy) / entropy_D  #C4.5的信息增益比
    else:
        return entropy_D - condition_entropy  #ID3的信息增益

```

```
72 def get_result(D, T = -1):
73     '''获取数据集D中实例数最大的目标特征T的值'''
74     if (not isinstance(D, (set, list))):
75         return None
76     if (not type(T) is int):
77         return None
78     count = {}
79     for t in D:
80         count[t[T]] = count.get(t[T], 0) + 1
81     max_count = 0
82     for key, value in count.items():
83         if (value > max_count):
84             max_count = value
85             result = key
86     return result
87 
88 
89 def devide_set(D, A):
90     '''根据特征A的值把数据集D分裂为多个子集'''
91     #判断D的数据类型是set和list类型
92     if (not isinstance(D, (set, list))):
93         return None
94     #判断A的数据类型是否是int型
95     if (not type(A) is int):
96         return None
97     subset = {}
98     '''根据特征A的结果划分数据集'''
99     for t in D:
100         subset.setdefault(t[A], []).append(t)
101     return subset
102 
103 
104 def build_tree(D, A, threshold = 0.0001, T = -1, Tree = None, algo = "C4.5"):
105     '''根据数据集D和特征集A构建决策树.
106 
107     T为目标属性在元组中的索引 . 目前支持ID3和C4.5两种算法'''
108     #判断Tree是否存在和Tree是否是节点
```

```
    if (Tree != None and not isinstance(Tree, Node)):
        return None
    #判断数据集D的类型是否是set集合和list集合的一种，如果不是直接返回
    if (not isinstance(D, (set, list))):
        return None
    #判断特征集A的类型是否是一个set集合
    if (not type(A) is set):
        return None

    if (None == Tree):
        Tree = Node(None, D)
    subset = devide_set(D, T)   #根据特征T的取值拆分数据集
    if (len(subset) <= 1):  #如果该特征T的取值为一个时，则这个唯一取值为这个节点的结果
        for key in subset.keys():
            Tree.result = key
        del(subset)
        return Tree
    if (len(A) <= 0):  #当特征个数小于等于0的时候，返回
        Tree.result = get_result(D)
        return Tree
    use_gain_ratio = False if algo == "ID3" else True
    #是要实现ID3还是C4.5算法，如果是ID3算法，use_gain_ratio为false，否则为true
    max_gain = 0.0
    for a in A:
        gain = info_gain(D, a, return_ratio = use_gain_ratio)
        if (gain > max_gain):
            max_gain = gain
            attr_id = a # 获取信息增益最大的特征
    if (max_gain < threshold):
        #判断信息增益比是否小于阈值，如果小于，返回数据集D中实例数最大的目标特征T的值
        Tree.result = get_result(D)
        return Tree
    Tree.attr = attr_id

    subD = devide_set(D, attr_id)
    del(D[:]) # 删除中间数据,释放内存
    Tree.dataset = None
    A.discard(attr_id) # 从特征集中排查已经使用过的特征
    for key in subD.keys():
        tree = Node(Tree, subD.get(key))
        Tree.childs[key] = tree
        build_tree(subD.get(key), A, threshold, T, tree)
    return Tree


def print_brance(brance, target):  #输出结果
    odd = 0
    for e in brance:
        print(e, end = ('=' if odd == 0 else '^'))
        odd = 1 - odd
    print("target =", target)


def print_tree(Tree, stack = []):
    if (None == Tree):
        return
    if (None != Tree.result):
        print_brance(stack, Tree.result)
        return
    stack.append(Tree.attr)
    for key, value in Tree.childs.items():
        stack.append(key)
        print_tree(value, stack)
        stack.pop()
    stack.pop()
```

```
174
175 # ==============================================================================
176 # #根据决策树产生数据结果
177 # def classify(Tree, instance):
178 #     if (None == Tree):
179 #         return None
180 #     if (None != Tree.result):
181 #         return Tree.result
182 #     return classify(Tree.childs[instance[Tree.attr]], instance)
183 # ==============================================================================
184
185 #导入操作excel文件的xlrd库
186 import xlrd
187 #读取文件
188 bk = xlrd.open_workbook("C:\\Users\\lenovo\\Desktop\\decisiontree_data.xls")
189 try:
190     sh = bk.sheet_by_name("Sheet1")#假设数据在该文件的sheet1下，读取sheet1
191 except:
192     print("当前文件不存在Sheet1")#如果不存在sheet1，则输出当前文件不存在sheet1
193 rowNum = sh.nrows #读取改文件sheet1下数据的行数
194 dataset = [] #定义存储数据的列表dataset
195 #按行循环读取数据
196 for i in range(1,rowNum):
197     rowData = sh.row_values(i)
198     dataset.append(rowData)
199 #开始建立决策树
200 T = build_tree(dataset, set(range(0, len(dataset[0]) - 1)))
201 #打印输出决策树
202 print_tree(T)
203 # ==============================================================================
204 # #根据决策树产生结果
205 # print(classify(T, ('女', '好', '每周大于三小时', 'A', 'C', 'B', 'D', 'D')))
206 # ==============================================================================
207
```