顺序号（硕）：SS021601
姓名：刘晓宇
学号：'21110233052
学院：信息科学与工程学院
专业：软件工程

中国海洋大学
OCEAN UNIVERSITY OF CHINA

# 硕士学位论文

MASTER DISSERTATION

论文题目：C4.5 算法的一种改进及其应用

英文题目：**An Improved C4.5 Algorithm and Application**

作　　者：刘晓宇

指导教师：魏志强 教授

学位类别：全日制专业学位

专业名称：软件工程

研究方向：现代软件工程学

2013 年 5 月 23 日

# C4.5 算法的一种改进及其应用

学位论文答辩日期：2013.5.23

指导教师签字：

答辩委员会成员签字：

## 独 创 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。据我所知，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，也不包含未获得_____（注：如没有其他需要特别声明的，本栏可空）或其他教育机构的学位或证书使用过的材料。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。

学位论文作者签名：刘晓宇　签字日期：2013 年 5 月 23 日

---

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，并同意以下事项：

1、学校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。

2、学校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。同时授权清华大学“中国学术期刊(光盘版)电子杂志社”用于出版和编入 CNKI《中国知识资源总库》，授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：刘晓宇　　导师签字：

签字日期：2013 年 5 月 23 日　　签字日期：2013 年 5 月 23 日

谨以此论文献给我的导师和家人们

---------- 刘晓宇

# C4.5 算法的一种改进及其应用

## 摘　要

随着科学技术的不断发展，人们的生活节奏不断加快，迫切需要从海量的数据中快速提取有用信息的技术，这项技术就是数据挖掘。数据挖掘已成为当今最热门的信息技术之一。C4.5 算法是数据挖掘十大经典算法中最经典的算法，在数据挖掘技术中起着非常重要的作用，使用率非常高。C4.5 算法属于决策树算法，分类规则以树的形式呈现。C4.5 算法改进于 ID3 算法，它在 ID3 算法的基础上，用信息增益率代替信息增益作为选取根属性的标准，克服了用信息增益选择属性时偏向选择取值多的属性的不足，能够完成对连续属性的离散化处理。C4.5 算法的最大特点是建树规则易于理解，建树者不需要了解任何挖掘对象所在领域的专业知识，并且分类速度快，分类器准确率高。C4.5 算法现在已经被广泛应用到经济、工业、医药、农业等各个领域，因此对 C4.5 算法研究是十分重要的。但是 C4.5 算法在很多地方存在不足，本文针对 C4.5 算法在数据冗余时可能导致算法复杂度过大，效率低等问题，对 C4.5 算法进行改进，并命名为 R-C4.5 算法。

算法的具体改进：计算每个属性中的元素的信息熵，比较同一属性下每个信息熵的值，如果数值相近，再计算元素集合的相似度。如果相似度系数很高，那么说明两个元素性质相同或相近，对两种元素进行合并形成一个新的元素。而相似度的计算采用了改进的 Jaccard 系数，将两个集合其中的一个集合的每个元素的个数乘以他们两个集合的总元素个数比，这样改进的目的不仅仅简单地比较两个集合元素个数的相近度，而是比较集合中元素所占比例的相近度。

通过对 C4.5 算法的改进，增强算法的预处理机制。改进的原理利用了信息熵属性的约简，将冗余属性剔除，减少了算法的复杂度，从而大大提高了准确度。本文不仅对 C4.5 算法进行了改进，同时在计算集合相似度时对 Jaccard 系数进行了改进，使相似度计算的标准不再是集合中元素个数之比，而改为集合中元素比例之比。这样做的目的是避免由于选取的总数量不同，而导致判断错误。

**关键词： 数据挖掘；决策树；C4.5；R-C4.5**

**An Improved C4.5 Algorithm and Application**

**Abstract**

With the continuous development of science and technology, there is an urgent need to extract useful information from the vast amounts of data technology. Data mining has become one of the most popular information technologies. C4.5 algorithm is the most classical of ten classical algorithms for data mining algorithms. Data mining technology plays a very important role with the high utilization rate. C4.5 algorithm is a decision tree algorithm based on classification rules, which is presented in the form of a tree. C4.5 algorithm improves ID3 algorithm, based on information gain ration instead of information gain as the standards of the selected root attribute, overcoming the deficiencies of the bias select value attribute when the attribute is selected using information gain, which is useful to discretize continuous attributes. The most important feature of the C4.5 algorithm is the contribution rules easier to understand, the achievements of those who do not need to know any mining objects in your field of expertise, and fast classification classifier with high accuracy. C4.5 algorithm has now been widely applied to various fields of economy, industry, medicine, agriculture, etc., so the C4.5 algorithm research is significantly important.

C4.5 algorithm inadequacies exist in many places. C4.5 algorithm in data redundancy may result in the complexity of the algorithm is too large. In this paper C4.5 algorithm has been improved in these aspect, and renamed R-C4.5 algorithm.

The algorithm specific improvements: calculate the elements in each attribute information entropy, compare the same property value of each information entropy. If values are similar, then calculate the similarity of the set of elements; if the similarity coefficient is high, then the description of the nature of the two elements of the same or similar, the two elements merge to form a new element. Similarity calculation uses improved JACCARD coefficient. The aim of such the change is not the simple comparison of two similar degrees on the number of elements in the collection, but compares similar degrees of collection elements in proportion.

The improvement of the C4.5 algorithm enhanced the procession mechanism. With the attribute of information entropy reduction, this removed redundant attributes

to reduce the complexity of the algorithm, which greatly improving the accuracy. This paper not only improved C4.5 algorithm, but also improved the calculation of JACCARD coefficient in similarity collections. The similarity calculation is no longer the ratio of the same number elements in collections, which changed to the ratio of elements proportions in the collection. The purpose of such improvement is to avoid due to the total number of selected, which led to an error of judgment.

**Key words: Data mining; Decision tree; C4.5; R-C4.5**

# 目　录

# 1 引言

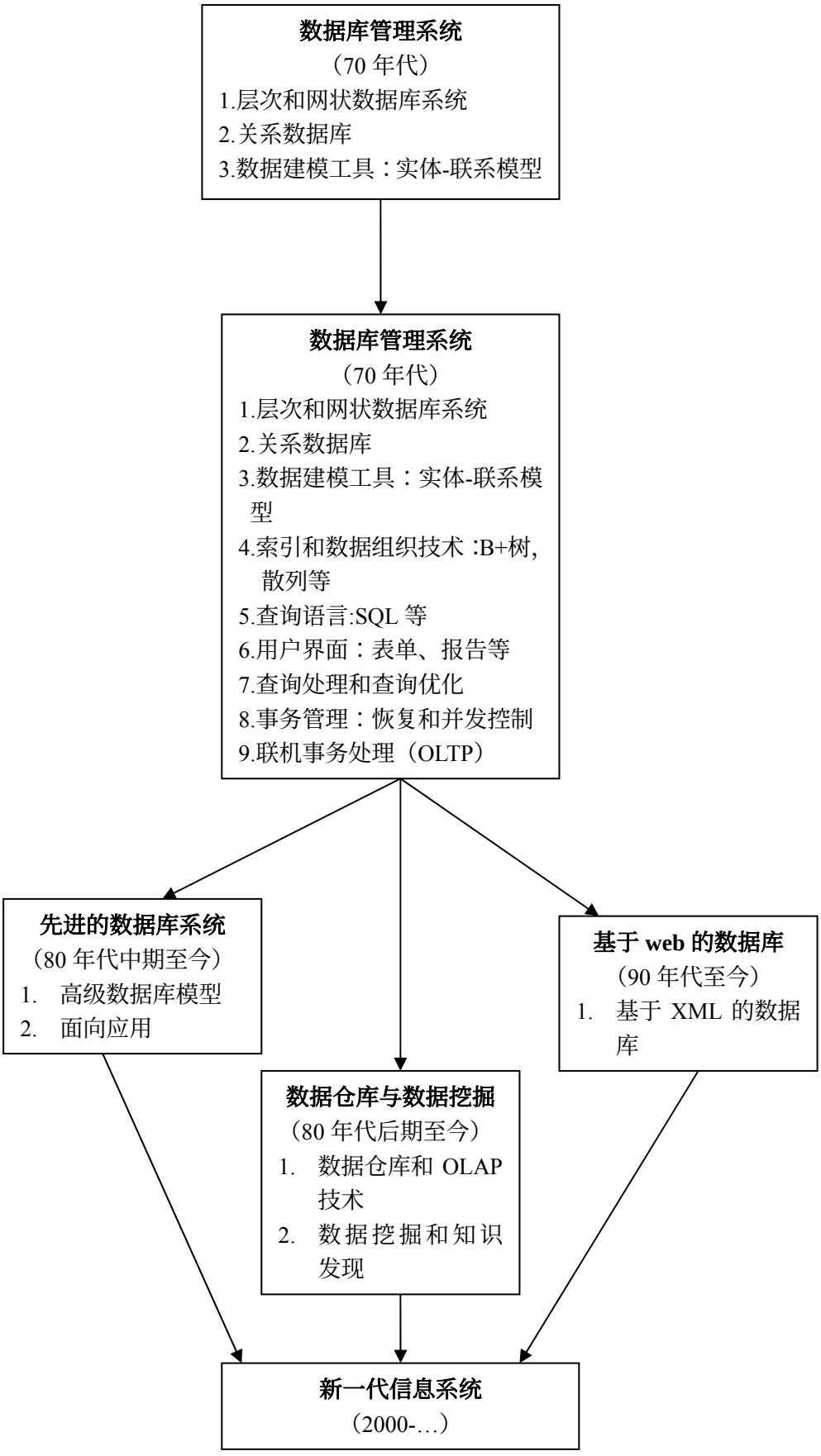
## 1.1 数据挖掘重要性

随着科技的不断进步，数据挖掘逐渐成为 21 世纪最热门的信息技术之一，引起了信息产业界的极大关注，人们迫切需要在海量的数据中挖掘出有用的信息，正是这种需要大大推动了数据挖掘技术的发展。如今数据挖掘已经被用于多个领域，如股票市场的走势分析，再比如产品市场的销售预测，这些领域由于数据量非常大使得数据分析极度困难，所以数据挖掘技术成了必备的工具。数据挖掘是信息技术自然进化的结果。数据挖掘技术很多都用于工业的发展，比如将数据库上升为数据仓库，提高了数据库的使用效率，而数据仓库正是数据挖掘所需要的样本训练集存储的地方。随着数据库中的数据不断增多，数据分析就成了人们关注的热点问题。

## 1.2 数据挖掘概念

数据挖掘[1]其实就是一种在海量的数据中获得重要信息的技术，这就像在海水中提纯盐一样，剔除其余不相关的元素，只是得到海水中的 NaCl 元素。很多时候数据挖掘又称为知识发现，因为数据挖掘就是在海量的样本空间中寻找需要的信息或规律，即知识，但是知识发现不能体现我们挖掘的样本训练集的数据之大的特点，这样，“ 数据 ”和“ 挖掘 ”成了流行的选择。数据挖掘这个概念也被更广泛的人接受。

数据挖掘发展历程如下：



## 1.3 数据挖掘的分类

数据挖掘不是一门独立的学科，它是很多学科的综合体现。我们在数据挖掘的过程中经常需要用到概率论、机器学习、数据库、数据结构以及信息论等其他学科的理论知识。数据挖掘的数据类型和的数据挖掘应用是非常复杂的，数据挖掘系统可以将数理统计与计算机的数据结构结合起来构建数据模型，也可以将信息论的知识用于系统中，更可以将经济、商业、以及心理学等不同领域的应用集合起来。因此，数据挖掘的系统标准并不是唯一的，它会针对不同的领域有不同的变化。

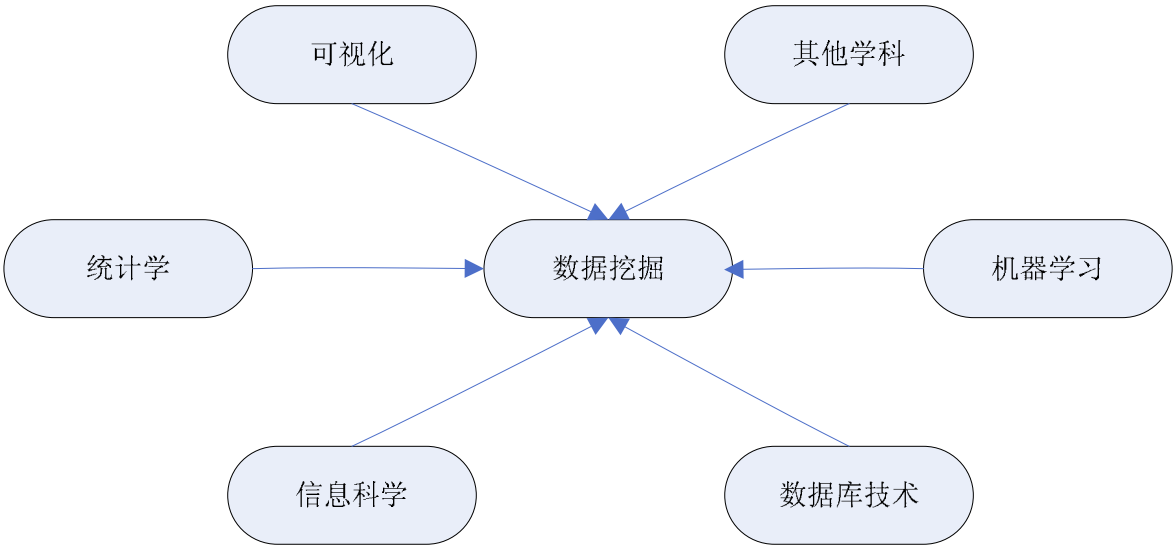



图 1-1 学科交叉图

数据挖掘系统分类如下：

### 1.3.1 数据挖掘技术分类

数据挖掘系统可以根据系统中不同的算法描述和数据分析，可以分为神经网络、遗传算法等系统。

### 1.3.2 数据挖掘的数据库类型分类

由于有多媒体数据库、空间数据库、关系数据库等不同类型的数据库的存在，因此相应的数据挖掘系统也可分为多媒体数据挖掘系统、空间数据挖掘系统以及关系数据挖掘系统。不同类型的数据库或数据仓库，数据挖掘系统构成也往往不相同。

## 1.4 数据挖掘的目的和任务

数据挖掘的主要目标是通过各种数据之间的相互关联，提取数据中最有用的信息。

数据挖掘其实就是挖掘人员将已经制定的数据挖掘规则从系统外部录入到数据的决策系统中心。在海量的数据中寻找到还未被发现的信息及规律是数据挖掘的主要任务，数据挖掘其实就是自动从海量数据中生成重要信息的过程，有些明确的信息决策者是直接获取得到的，比如说他们可以通过查询、联机分析处理(OLAP)[2]或其他方式得到他们所需要的信息；数据挖掘是针对那些蕴藏在大量数据中的非常隐含的关联以及趋势，这些信息往往是那些行业资深人士也看不出的，但是可能这些信息又是非常重要的。根据不同的目的对数据挖掘任务做分类如下：

### 1.4.1 探索性数据分析

探索性数据分析就是没有想法的去探索数据，随着探索的不断加深，知识一点点的积累，从而得到重要信息。

### 1.4.2 描述建模

很多时候我们需要多所建立的数据模型进行描述，描述建模指的是将数据挖掘的所需要的数学模型，用文字或图像解释清楚，是用户很清楚的了解整个数据模型。

### 1.4.3 预测模型分类和回归

预测模型是根据已知的知识建立的一个数据模型，通过这个模型可以对未知的数据值进行预测。分类中，被预测的变量时范畴型的，回归中被预测的变量时数量型的。

1.4.4 寻找模式和规则

很多数据挖掘应用并不是为了数据模型的建立, 它们更多的时候是为了模型的探索。

1.4.5 根据内容检索

根据用户所提供的信息建立相似的数据模型, 往往这种模型非常直观, 用户看起来易懂, 模型多采用文字叙述, 有时会用现实模型如树、网等构造。算法根据用户提供信息的关键词找到与其关键词最相关的文件或者图片, 这就是搜索算法。

## 1.5 数据挖掘系统组成

(1) 数据库与数据仓库在整个数据挖掘系统中主要是用于存储与整理数据, 一般样本训练集都来自于数据库。

(2) 在数据挖掘系统中, 有一个模块存储着很多已知的方法用于对数据的前期处理分析, 它就是知识库。知识库在数据挖掘中非常重要, 如果数据预处理得当, 那后面的数据建模就会非常轻松与准确, 反之, 如果处理不当将会使整个系统低效, 大大降低系统的准确性。

(3) 数据挖掘系统的核心就是数据挖掘引擎, 它为数据挖掘提供核心算法, 包括很多的分类算法与聚类算法。数据的特征分类与聚类以及数据模型的建立都在此模块中完成。

(4) 在数据挖掘的过程中我们需要对数据模型进行评估预测, 这就需要系统中要有评估模块。它主要是用于评估挖掘过程中所建立的数据模型对未知的数据样本分析的结果是否准确。

(5) 数据挖掘系统的最后部分是用户界面, 它的作用是更好地与用户进行交互, 使用户能够更好的操作系统, 完成自己的需求。为了使用户对系统更容易理解, 用户界面往往以图形界面的形式出现。

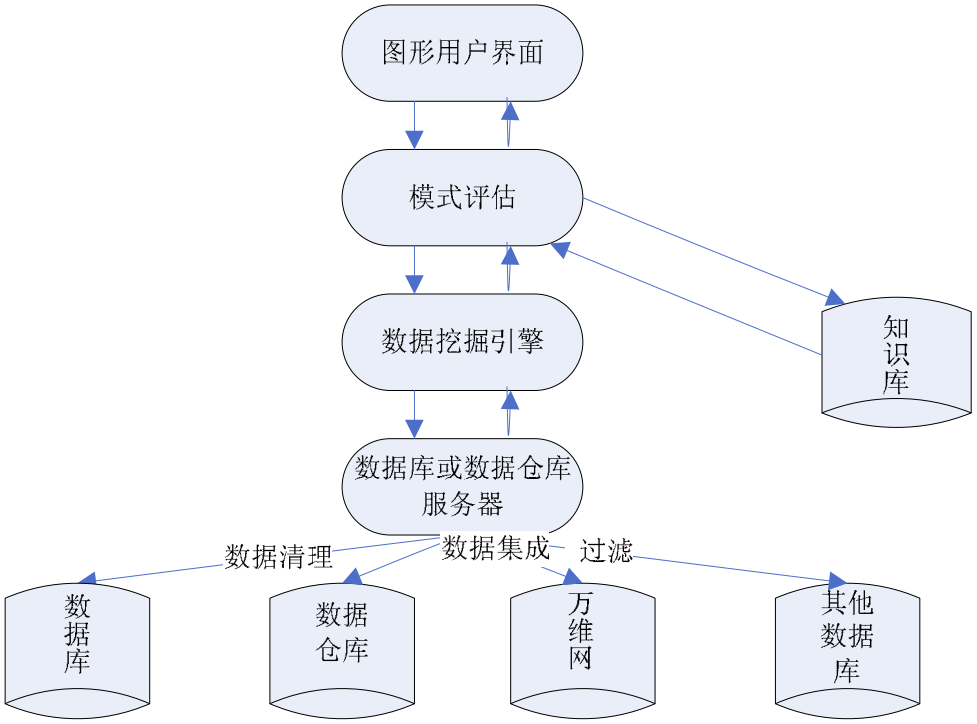



图 1-2 数据库挖掘系统结构

## 1.6 数据挖掘基本步骤

在不同的领域及其应用上，针对不同的数据类型，数据挖掘的步骤是完全不一样的，数据挖掘技术会根据用户的不同需要制定不同的方案。数据库里的数据的完备性以及数据挖掘者在挖掘对象的专业知识的储备量，同样对挖掘的结果产生重要的影响。不用的领域不同的挖掘对象他们的数据挖掘过程是不一样的，即便面对同一领域甚至面对同一挖掘对象，不同的人也会制定出不一样的挖掘方案。因此，数据挖掘是多变的，所以需要将数据挖掘的规则进行标准化，让整个数据挖掘更加的系统化，制定统一的规则，这样就可以使数据挖掘在跨平台跨领域时更加方便，增加程序的可移植性。

数据挖掘完整的步骤如下：

(1) 理解数据和数据的来源（understanding）

(2) 获取相关知识与技术（acquisition）

(3) 整合与检查数据（integration and checking）

(4) 去除错误或不一致的数据（data cleaning）

(5) 建立模型和假设（model and hypothesis development）

(6) 实际数据挖掘工作（data mining）

(7) 测试和验证挖掘结果（testing and verification）

## 1.7 国内外研究现状

### 1.7.1 国外研究现状

人们一开始在庞大的数据库中寻找数据间的关联，期望得到相关规律，这就是早期的发现知识（Knowledge Discovery in Database）[3]。在 1989 年，国际人工智能会议上，将发现知识上升为学术概念，即数据挖掘。这也是数据挖掘首次被提出。到目前为止，数据挖掘已经有了很大的发展，尤其在数据仓库以及事物关系库中的挖掘取得了多项突破，其中较为突出的是：将面向对象的方法用到数据挖掘中，提出面向属性这一概念，用这一方法对数据进行特征挖掘以及区分；大大提高了聚类算法的效率，使类别的相似度证明更加的严谨，从而提高了聚类算法的准确率；发现了数据库中各种事物之间的关联，找到数据间的关联规则。在整个数据挖掘中存在很多的不确定性，这就大大推动了粗糙集的发展，将概率论的概念加入到数据挖掘中。除了以上这些，在机器学习和知识发现中大量使用决策树、神经网络、遗传算法、可视化等方法，使这些方法的研究有了很大的发展。在数据挖掘领域，最有影响力的发现算法有加拿大 Simon Fraser 大学 J.Han 教授的概念树提升算法(Hanetal,1992)、IBM 的 R.Agrawal 的关联算法 Apriori、澳大利亚的 J.R.Quinlan 教授的分类算法 C4.5/C5.0、Zhang 等的 BIRCH 聚类算法、密歇根州立大学 Erick Goodman 的遗传算法[4]。

目前国外主要研究的数据挖掘算法为贝叶斯算法，因为贝叶斯算法的数学理论依据是最多的，所以它的准确率也是最高的；经过不断的研究知识发现与数据库的紧密结合；传统的统计学回归方法在知识发现中得到应用。比如现在的知识发现中，不再是简简单单的针对某一问题而解决问题，更多的是根据这一问题而建立一个体系，从而以后在遇到相似问题时同样可解，增加了算法的共用性。现在很多的公司以及政府银行等单位已经在广泛的使用这些数据挖掘的应用。美国的数据挖掘技术是世界领先的，他们的很多著名的计算机公司如微软、谷歌等都在数据挖掘领域投入了大量的人力物力，并相继成立了数据挖掘研究中心，足可

见数据挖掘技术的重要性。

现在许多国际上知名的大学以及相关的计算机公司都在大力开发自己的数据挖掘系统。目前，最有影响力的数据挖掘系统有：SGI 的 MinerSet、SAS 公司的 Enterprise Miner、IBM 公司的 Intelligent Miner、SPSS 公司的 Clementime、Sysbase 的 WareHouse Studio、Stanford System 的 CART、Thinking Machines 公司的 Darwin、Rulequest Reserch 公司的 See5 等[5]。

空间数据挖掘近十几年来刚刚被提出的，因此他的研究对于实体数据库以及关系数据库来说要晚很多，但随着科技的不断的进步信息量不断增大，传统的数据库已经不能满足一些用户的需求，所以空间数据库引起了很多专家学者的关注。空间数据挖掘作为新兴的数据挖掘技术，它的出现目的在于，将二维数据关系上升为三维数据，从而使整个数据关联更加准确，扩展了存储空间大大提高了传统数据挖掘的效率。

国际上，数据挖掘技术已经应用于保险业、金融和银行业、市场营销、交通、电信等众多领域中。随着数据挖掘方法和理论的研究不断加深，不仅是应用广度不断拓展，同样大大推动了数据库的发展，从而出现了像空间数据库、网页数据库、多媒体数据库等新型数据库。数据挖掘技术必将成为 21 世纪初信息技术领域中热门的课题。

### 1.7.2 国内研究现状

我国在数据挖掘方面的研究比西方国家要晚，并且在很多方面上技术不够成熟。目前我国正大力发展该项技术，并且取得了很多学术成果：提高了分类算法在处理海量数据以及连续数据时的能力，将集合理论与分类算法相融合；将粗糙集和模糊集理论二者融合用于知识发现；构造模糊系统辨识方法与模糊系统知识模型；构造智能专家系统；研究中文文本挖掘的理论模型与实现技术；利用概念进行文本挖掘。现状态空间理论和云模型对在数据挖掘中的应用、模糊方法在数据挖掘中的应用、数据立方体代数的研究、关联规则发掘算法的优化和改造、非结构化数据以及 Web 数据挖掘。

## 1.8 国内外常用软件介绍

### 1.8.1 Knowledge Studio

这款软件由著名的 Angoss 公司开发，它能够灵活的将外部的模型与产生的规则导入软件中，它的最大优点是响应速度非常快，文档与模型非常易于理解，并在 SDK 中增加了新的算法。

### 1.8.2 SPSS Clementine

SPSS Clementine 是 ISL ( Integral Solutions Limited) 公司开发的数据挖掘工具平台[6]。SPSS Clementine 主要针对的商业领域，它的最大作用是可以快速准确的预测市场走势，提供相应的走势图帮助决策人进行正确的决策，所以 SPSS Clementine 在商业领域内应用的非常广泛。SPSS Clementine 提供了 K-Means 和 Two Step 种聚类分析模型。聚类分析是将样本训练集中相似度高的数据集聚集在一起形成新的数据集，聚类完成后每个数据集之间的相似度非常低，这样就可以将整个数据集分成小的数据集。在聚类过程中，我们只需要计算数据集的相似度，并不需要其他的相关知识以及他人的指导，所以聚类学习也可以叫做无监督学习。

### 1.8.3 Enterprise Miner

Enterprise Miner 软件在我国的商业领域中得到非常广泛的应用，我国很多公司采用的数据挖掘软件都是 Enterprise Miner。SAS Enterprise Miner 可以应对不同的数据集，它的运算模式是固定的，首先他要在总训练集中抽取相应的样本空间，然后分析所抽取的样本空间，转换样本空间中的数据从而建立相关的模型，最后根据建立的模型做出相应的结果预测。Enterprise Miner 软件可以嵌入到数据仓库中与仓库完美融合，大大降低了数据传输速度，使整个数据挖掘过程变得非常高效。

## 1.9 数据挖掘的发展趋势

数据挖掘面对的数据是千变万化的，它需要应用到不同的专业领域。这就需要不断地完善数据挖掘技术，从而能够解决更多的现实中的问题。现在数据挖掘有许多问题解决起来非常困难，比如如何使数据挖掘算法变得更加高效简单，如何提供更好的挖掘环境，能否建立一个模型解决多种问题使整个数据挖掘算法更加统一更加标准化，能否针对数据挖掘建立统一的挖掘语言等等，这都是需要我

们解决的技术问题。

然而，毫无疑问数据挖掘具有无穷的潜力。数据挖掘与人工智能正在飞速的发展，它们是未来工业发展的主要推动力，对整个科技发展有着深远的影响。现在很多公司研究机构都将大量的人力物力财力投入到数据挖掘这个新兴的领域中去，足可见数据挖掘在未来生活中的重要性。数据挖掘的发展趋势可以归结如下：

### 1.9.1 应用的探索

早期数据挖掘应用主要集中在帮助企业提升竞争能力，而现在整个数据挖掘已经应用到各行各业，其中在工业、商业、医学领域内应用的最多。并且现在电子商务与互联网技术正在飞速的发展，这增加了对数据挖掘技术的需求，推动了数据挖掘技术的发展。但是目前许多数据挖掘系统的设计仅是针对某一特定的问题，无法将整个系统共用。

### 1.9.2 可伸缩的数据挖掘方法

数据挖掘的最大特点就是需要处理海量数据，这和简单的数据分析是不一样的。所以对于海量的数据处理，数据挖掘算法的可伸缩性是必不可少的，它可以使整个过程处理起来更加容易。

### 1.9.3 数据挖掘与数据库系统、数据仓库系统的集成

在整个数据挖掘体系中需要将数据挖掘技术与数据仓库紧密的结合起来，将处理数据与分析数据放入一个体系中，提高算法的效率。通过系统的集合可以提高算法的移植性，使同一种数据挖掘算法能够解决多种问题，从而提高了数据挖掘算法的共用性。

### 1.9.4 数据挖掘语言的标准化

由于面对数据以及问题的不同，数据挖掘系统是多样的。因此制定相同的语言标准可以增加算法的通用型，增加算法之间的交互，有时我们在整个挖掘过程中需要用到多种挖掘算法。

### 1.9.5 可视化数据挖掘

可视化数据挖掘时从大量数据中发现知识的有效途径。可视化数据挖掘增加了对图像图片等样本集的处理，增加了数据挖掘可处理的数据类型，使数据挖掘

更加完善。

### 1.9.6 复杂数据类型挖掘的新方法

在数据挖掘中我们需要面对各种各样的数据类型，所以如何处理复杂数据类型成了很多专家学者关注的焦点。比如在对网页进行挖掘时我们需要同时处理文字与图片，所以对复杂数据类型的挖掘在现实生活中是必不可少的，但是在这方面数据挖掘技术还需要提高。

### 1.9.7 Web 挖掘[7]

互联网已经成为人们生活中必不可少的生活工具，如网上购物，浏览新闻，点击视频等，所以对互联网的挖掘则显得尤为重要，其中最为代表的算法就是搜索算法。

### 1.9.8 数据挖掘中的隐私保护和信息安全

数据挖掘技术虽然给我们带来了很多方便，但是由于它需要提取数据信息，而往往在提取的信息中可能涉及到个人的隐私，如姓名、电话号码、年龄、身份证号等私密信息。因此如何在提取数据时保护个人隐私是目前数据挖掘研究的新课题。

### 1.9.9 空间数据挖掘[8]

最近十几年刚刚提出的概念，它主要是针对传统的数据挖掘技术在面对海量数据时通常处理起来很困来这一不足之处。空间数据挖掘所获得信息更加准确并且效率更高，是当今数据挖掘领域内研究的热点问题。

## 1.10 课题意义

随着科技的不断进步，数据挖掘逐渐成为 21 世纪最热门的信息技术之一，引起了信息产业界的极大关注。C4.5 作为数据挖掘十大经典算法中排名第一的算法，更是很多专家学者的研究重点。

C4.5 算法继承了早期决策树算法中分类规则形象易于理解、建树者不需要了解任何挖掘对象所在领域的专业知识、分类速度快以及分类器准确率高等优点。C4.5 算法在继承决策树算法优点的同时，还做出了很多改进。例如它用信息增益率代替 ID3 算法中的信息增益来选择属性，增加了算法的准确率，它还增强了对数据的预处理能力，减少了算法复杂度，同时也增强了处理连续数据的能

力。C4.5 算法现在已经被广泛应用到经济、工业、医药、农业等各个领域，因此对 C4.5 算法研究是十分重要的。

## 1.11 选题缘由

C4.5 算法虽然已经被广泛应用到各个行业，但是算法还是有些许不足之处。例如虽然增强了数据预处理能力，但是效果有时候依然不好，再比如处理海量数据时比较困难，往往会造成决策树过大，算法低效。所以本文针对 C4.5 算法的不足，做出了改进并提出了一种改进的 C4.5 算法。对 C4.5 算法的主要改进是进一步加强了算法的预处理能力，在决策树构建之前会对原始数据进行信息熵的比较，将相似度高的数据集进行合并，减少训练样本的类别属性，从而使整棵树的算法复杂度减小，已达到提高算法效率的目的。在合并相似属性的过程，本文还对 JACCARD 系数做了改进。

## 1.12 文章结构

第一章 数据挖掘技术部分主要介绍了有关数据挖掘的概念、数据挖掘技术的重要性以及不同种类的数据挖掘技术,并介绍了本文的课题意义以及选题缘由，C4.5 算法的优点以及重要性。

第二章 主要是介绍了当今主流的数据挖掘算法，对每个算法的优点、缺点以及特点做了重点描述，是算法的比较一目了然。

第三章 主要介绍了决策树算法，对决策树算法中的分类规则、核心思想、构造方法等方面做了重点介绍。

第四章 主要介绍了 C4.5 算法，重点介绍了 C4.5 算法的核心思想、算法的优点以及不足之处。

第五章 介绍了一种改进的 C4.5 算法，并以具体实验为例，给出了改进的 C4.5 算法与 C4.5 算法实验对比。

第六章 介绍了改进的 C4.5 算法的不足之处。

# 2 数据挖掘常用算法介绍

数据挖掘算法简单来说就是数据挖掘的具体计算过程。算法通过分析样本训练集得到相关规则，数据挖掘系统会根据这些规则建立与之对应的模型。因此数据挖掘算法是整个数据挖掘体系的核心。算法使用对训练集分析的结果来定义用于创建挖掘模型的最佳参数，从而确定整个模型的参数，而数据挖掘的模型是多种多样的，很多时候根据样本训练集的实际情况可以建立混合数据模型，数据模型包括以下形式：

(1) 说明数据集中的事例如何相关的一组分类。

(2) 预测结果并描述不同条件是如何影响该结果的决策树。

(3) 预测销量的数学模型。

(4) 阐述样本训练集的具体分类规则，通过对已知数据分析得到的结果推测未知数据。

## 2.1 C4.5 算法

C4.5 算法是决策树算法中最常用的算法之一，它是基于决策树(决策树为一颗倒树，最上面为根部，下面用线的形式将各个节点连接起来)核心算法 ID3 的改进算法。很多时候我们不要掌握全部的决策树方法就可以用 C4.5 算法建立一棵树。决策树构造方法是指通过属性分类条件，将每个单一属性作为一个树的子节点，然后再将各个子节点连接起来。相比 ID3 算法，C4.5 算法在如下地方进行了改进的地方：

ID3 在属性的选择上用的是属性的信息增益(在信息论中对信息有很多的量标准，C4.5 算法与 ID3 算法都以信息熵为度量标准(熵来源于物理热力学定理))，信息增益描述的是信息熵值变化的大小。而不同于 ID3 算法，C4.5 算法采用的是信息增益率，他描述的是单位属性上信息量的变化。增益率实际上描述的信息属性的平均值，增益率与增益是完全不同的两个概念，这就像我国的国民总收入是世界第一的，但是由于我国人口众多导致人均收入就非常低，所以有时候信息增益与信息增益率差别还是很大的。信息增益率克服了用信息增益选择属性时偏

向选择取值多的属性的不足[9]。在 C4.5 算法中剪枝机制是必不可少的，很多时候会出现空枝或数据冗余导致树过于庞大，这样会导致整棵树过度拟合，因而剪枝在 C4.5 算法中是必不可少的。C4.5 算法在面对连续值数据值时通常会采用对数据进行泛化处理，将连续数值离散化，增加了算法在面对复杂数据时处理的能力。

## 2.2 THE APRIORI ALGORITHM 算法

Apriori 算法是一种挖掘数据间关系的算法，它主要是寻找数据间的关联，建立新的数据集，即频繁项集[13]，通常在建立的新的数据集中的数据可信度都大于在挖掘系统中设定的最小可信度。Apriori 算法的主要作用就是找到数据间的布尔关联规则。Apriori 算法核心是对两段频繁数集的递推演算，这是整个算法最重要的地方。

在 Apriori 算法的递推演算中，存在三个非常重要的运算参数，他们都需要提前统计。首先要在所有样本空间中，找到含有最多样本数量的数据集，统计它所含有的样本数量，记为最大物件数。期次需要设定最小支持度，它的作用是规定符合样本特征中样本数量最低个数。在 Apriori 算法执行之前，我门还需要统计最小信息水准数，算法中所有的分类规则必须符合它，否则，这个分类规则会被弃用。

基本思想：第一要建立支持度大于最小支持度（最小支持度为提前设定）的频集，通过频繁项集建立强关联规则，所有集合项中的规则都会依据强关联规则建立，强关联规则必须满足支持度大于系统设定的最低支持度。在制定集合规则的过程中采用中规则制定的方法，将制定的只包含集合项的规则的右部分只确定一项。这样如果可信度大于系统设定的最小可信度的规则才会被保留下来，否则就会被淘汰。通常会采用递推的方法，建立需要的频繁项集。在 Apriori 算法中有两大缺点：(1)有时候在算法执行中会产生大量的候选集，导致数据过于庞大处理起来非常困难；(2)很多时候需要重复扫描数据库，导致算法低效。

## 2.3 PAGE RANK 算法

PageRank 算法是 Google 算法中的一个分步，是谷歌创始人之一的拉里•佩奇

（Larry Page）发明，2001 年 9 月被美国授予专利[15]。因此 PageRank 算法以佩奇命名。PageRank 算法是用来评估一个网站的好坏，他的评估标准是网站内外部链接的数量以及质量。PageRank 算法的核心思想是，统计外面其他网页链接该网页的个数，外面链接该网页的链接次数越多，就说明该页面质量就越好。它采用链接流行度作为衡量网站好坏的标准，即如果该网站的链接次数越多，说明该网站的认可度就越高，也就说明网站质量越好。PageRank 来源于学术论文的引用频度，当你的论文被其他人的论文引用的多了，说明你的论文技术含量就很高，从而学术水平也高。

在 Google 算法中有专门记录链接次数的程序。Google 将 PageRank 算法分为十个等级，其中十级代表的网站质量最高，但是这是非常不多见的。PageRank 算法的每个等级差别程度是不一样的，没有一个固定的差别标准表示级别之间到底差多少，但是顺序是固定的，也就是 10 级一定会比 9 级强，但是强多少则不固定。有时候 3 级可能比 2 级要强 1 倍，但是 4 级与 3 级可就相差 4，5 倍甚至更多，所以 PageRank 算法的等级划分在数学思想上是非常独特的，他没有给定一个准确的等级划分标准，我们也无法单从级别上推算出两者的差值。PageRanks 的等级划分，只是等级的排序。

由于 PageRank 算法衡量标准是外部网页的链接数，这就导致了很多人为了获取更高的 PageRank 算法等级，与别人交换链接甚至出售自己网站的链接（PageRank 等级越高，Google 搜索位置就会越靠前）。面对这种现象的出现 Google 修改了其算法规则，对网站的链接进行筛选，删除一些网站的链接数。例如，Google 删除了一些空网站的链接（很多为了增加链接次数建立的网站），这些数据将不会记录在 PageRank 算法的等级划分制度内。再比如增加本网站与链接网站的关联度检测，如果两个网站内容是完全不同的（例如在在体育网站链接的是唱歌类网站），这些网站的链接次数同样会被剔除，PageRank 算法也不会记录这些数据。Google 更新 PageRank 等级的速度非常慢，这样做的目的在于减少网站工作者对等级的关注度。PageRank 算法的优缺点如下：

优点：PageRank 算法获得数据非常直观也非常方便，获取数据速度非常快，效率极高。

缺点：PageRank 算法在用户搜索主题性的相关性上做得不足，很多时候查

询的结果与用户查询主题差别比较大；另外，PageRank 算法衡量的数据是外部网站的链接数，这样对一些刚刚建立的新网站非常不利，因为刚刚建立所以其链接次数几乎为 0，但它不一定就比很多老网站质量差，因而 PageRank 算法在这方面仍需要改进。

## 2.4 ADABOOST 算法

Adaboost 算法是分类算法的一种，它主要依靠于对数据的迭代。它的核心思想是在需要进行数据挖掘的训练集中建立很多小的弱分类器。Adaboost 算法利用这些小的分类器对原始数据进行分析，随着信息的不断积累，就可以建立一个更加完善的分类器，这就是 Adaboost 算法的强分类器[16]，并且算法中只会含有一个强分类器。Adaboost 算法改变整个数据的结构，对数据进行重新划分，Adaboost 算法需要判断样本划分的准确率以及总体分类的准确性，通过这两个标准来确定样本数量平均值。将新划分的数据的平均值传递给弱分类器中继续划分，划分到最后将所有的弱分类器分类结果综合起来，产生一个强分类器，最后数据结果是由强分类器计算得出。Adaboost 数据分类的所有重点都是只在乎数据的本身，而对数据外的一些条件 Adaboost 算法考虑的非常少甚至忽略这些条件，所以 Adaboost 分类器是纯粹的数据分类器。

到目前为止，很多专家学者对 Adaboost 算法中的分类器构造做了大量的研究与改进，Adaboost 算法不仅可以应用到分类算法中，同时像 SV 机算法一样已经广泛于很多回归问题中。Adaboost 算法对于类别数据很多的样本训练集有着很好的效用。

Adaboost 算法的具体步骤如下：

1 将原始的数据训练集分为正负两个不同的样本空间，对样本数据集进行不间断的分类。

2.计算初始样本的概率分布；

3. 对样本进行迭代：

(1) 根据训练样本的权重分析， 生成若干初始分类器即弱分类器

(2) 根据已知规律判断分类器的准确率

(3) 选取准确率最高的分类器对样本空间进行分类

(4) 重新计算样本的概率分布：

(5)根据以上过程生成系统唯一的分类器，即强分类器

## 2.5 KNN: K-NEAREST NEIGHBOR CLASSIFICATION（邻近算法）

邻近算法英文名为 k-Nearest Neighbor[17]，简称 KNN 算法，是分类算法的一种，该算法提出的时间非常早，因此在技术上非常成熟。KNN 算法是一种非常简单的算法，算法的思想与 K-Means 算法非常相近，都是以计算数据间相似度为主，不同的是 K-Means 算法是将相似的数据聚在一起，而 KNN 算法则与之相反属于分类算法。

KNN 的具体思想 :在所确定的样本空间中划分出 K 个不同的相邻的数据集，如果一个数据集中的数据大部分与特征数据相近, 那么就把该数据集划到该特征的样本空间。KNN 算法中，相邻的数据集都是已经分类的，所以算法准确度上有所提高。KNN 算法上在确定样本类别时是根据该样本最邻近的一个数据集或多个数据集，通过数据集的特征确立整个样本的特征值。 KNN 算法也需要对没有读数据进行相似度计算。与 K-Means 不同的是 KNN 算法只需要计算邻近样本空间的相似度，计算量远小于 K-Means 算法。KNN 算法的类域非常的小，它通常会忽略与邻样本无关的其他类域，所以在 KNN 算法的计算过程中只包含对邻近样本的计算，也就是说 KNN 算法更加适合数据集相似度高、重复度高的样本训练集的分类。

KNN 算法与 Adaboost 算法以及 SV 机算法一样不仅可以应用于分类，同时可以应用到回归分析中。以某个样本为中心，通过分类算法找到与其相近的 K 个数据集，计算这些数据集的平均值，而这个平均值表征整个样本的特征值。我们可以将不同样本对中心样本的影响度进行量化表示, 比较各不同距离的样本的对中心样本的影响度。

KNN 算法的不足之处：如果在整个样本训练集中样本分布不均匀，例如一个中心样本的邻居中有数量特别大的数据集, 那么该样本的特征值选取有可能偏向该数据集，这就类似于 ID3 算法中采用的信息增益为分类标准的不足之处一样。因此在 KNN 算法中引进了类似于信息增益率的标准权值来解决这个问题。KNN 算法的另一个缺点就是虽然思想非常简单，但是往往计算量过于庞大，导

致整个算法非常低效。KNN 算法的复杂在于它要求确定离中心最近的 K 个邻点，而邻点的求得需要对所有的待分类的数据集进行距离计算，从而得到邻点。针对 KNN 算法的这点不足，很多专家学者都提出了不同的改进方案，其主要改进思想就是先对样本进行化简，剔除数据中冗余数据，将整个数据集化为最简。KNN 算法对数据容量大的样本训练集有很好的分类效果，即样本数量越多算法的分类就越准确。

## 2.6 NAIVE BAYES 算法

贝叶斯算法[18]是最常用的两大分类算法之一（另一个常用算法是决策树算法，决策树算法是通过在训练集中建立树模型来分类）。很多时候分类算法中使用决策树算法而非贝叶斯算法，主要原因在于：贝叶斯与决策树相比，决策树算法分类更加简单与高效，由于模型是一颗倒装的树，数据描述更加形象，分类规则也更加容易理解；决策树与数据仓库是两个独立的体系，不会因为数据过大而导致无法建树；决策树算法与贝叶斯相比在处理多属性多层次的数据集中算法更加高效。当然在决策树算法中也存在许多不足之处，比如由于冗余数据过导致树建立的过于庞大从而使整个算法过于的复杂，再比如决策树上的很多节点上的样本是空的，即为空节点，这种空节点对分类毫无作用，但却使整个算法变得低效。

贝叶斯算法比决策树算法有优势的地方是，贝叶斯算法有强大数学理论做依据，有许多的概率理论作支撑，准确度要远远高于决策树算法，决策树算法在很多时候缺乏数学理论支持，往往树的建立需要建树人在该领域有丰富的经验才可提高决策树算法的准确率。贝叶斯算法的思想非常容易理解，在制定分类规则时所需要的参数并不多，但是贝叶斯算法在遇到如在挖掘过程中出现数据缺失的问题，算法往往检测不到，从而造成算法准确率下降。贝叶斯算法因为有强大的数学理论依据作支撑，它要比其他分类算法的准确率要高。但这只是理论上的概念，很多时候贝叶斯算法并非那么准确，原因是在贝叶斯模型中，它要求每个分类的属性是相互独立的，这是不符合实际的，很多时候属性之间是相互关联的，而贝叶斯算法忽略的这种属性的关联性导致算法的准确率下降。所以，当一个样本训练集中个属性相关度很大时，贝叶斯算法的效率以及准确率远不及决策树算法，相反当面对属性相关性很小的样本训练集，相对于决策树算法具有很大优势。

贝叶斯模型与决策树模型不同的是, 贝叶斯模型采用的是有向无环图的网状结构, 整个网状结构中的任意节点都代表着一个属性参数, 如果两个节点之间有弧线相连接, 说明两个随机变量是相互关联的, 否则说明两个参数是相互独立的。结构的每个节点都能够计算在其父节点中对所有样本可能获取的数值时的条件概率, 因为每个节点的内部都会储存一张 Conditional Probability Table, 即条件概率表, 简称为 CPT[19]。假设如果一个节点没有它的父节点, 它依然存在条件概率表, 只不过这张表为该节点自己的条件概率分布。通过对贝叶斯网状图的分析, 我们可以清楚的得到整个参数的概率分布图。

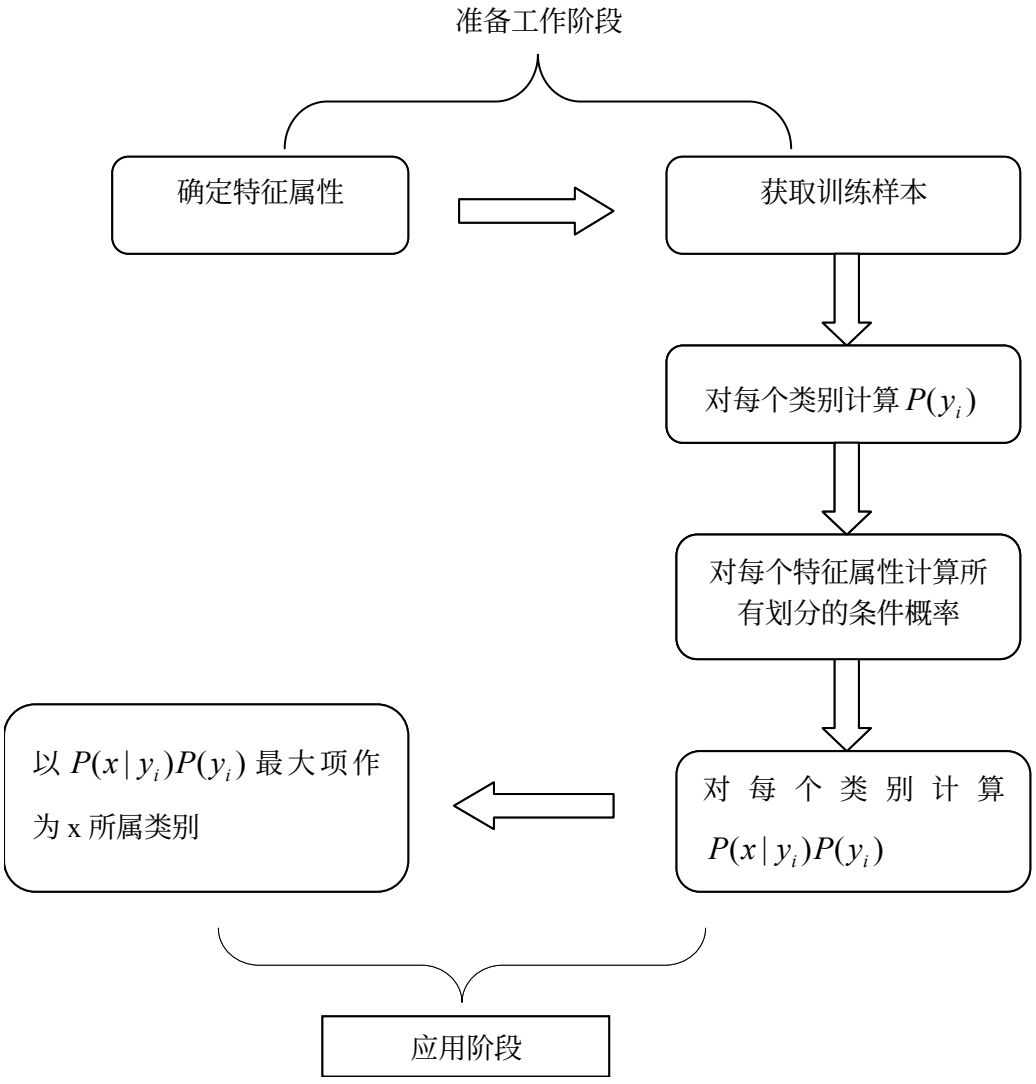



图 2-1 贝叶斯算法

# 3 决策树算法

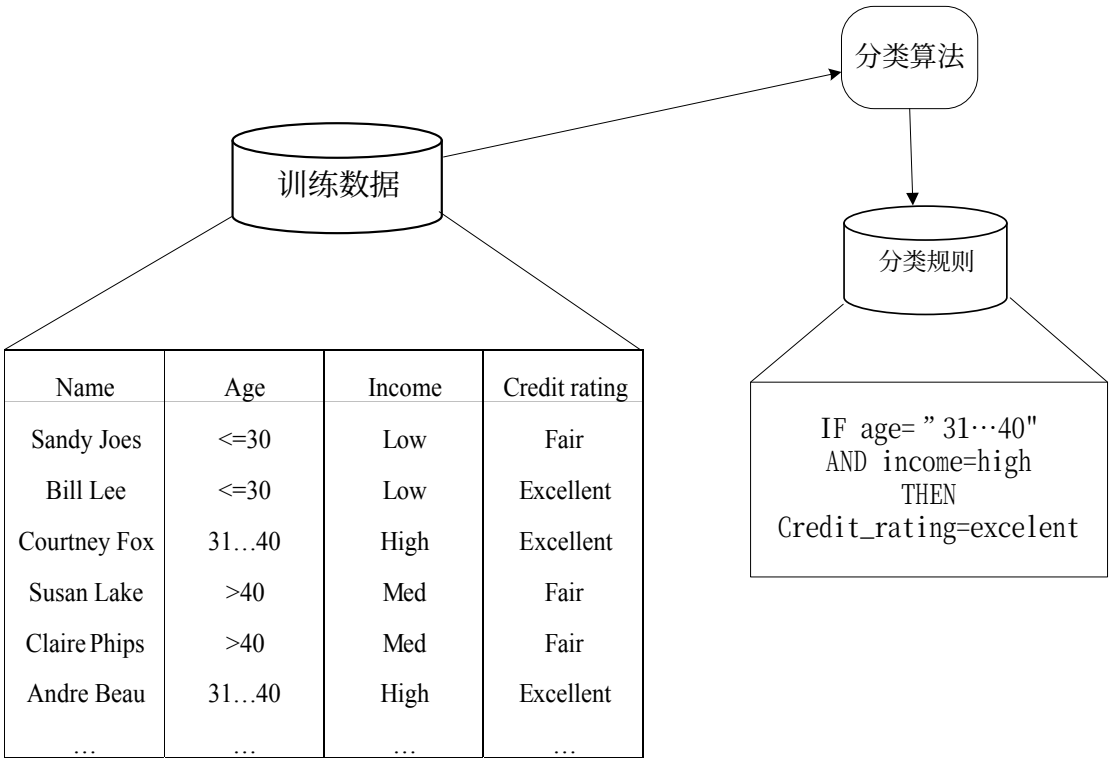
## 3.1 分类与预测

决策树是分类算法[22]的一种，数据分类由两大步骤组成。首先，确定数据类或概念集的意义，依据样本属性的描述建立合理的数据模型。样本集中的每一个属性元素就是一个类标号。被模型分析的数据被称为训练数据集。在训练数据集中，数据又被划分为小的训练样本，并被样本群随机的选取。由于对类进行标号，这就是对数据分类有指导性的作用，通常称为指导学习，即很清楚的知道具体的哪个类在指导着训练样本的数据分析，与有指导的学习相对应的是无指导的学习，又被称作聚类，因为在聚类过程中并不知道类的标号，并且很多时候类的集合与数量也无从知道。

在数据分析的过程中往往需要有具体的分类规则和相关的数学理论以及判定树。如图 3.1(a)该图描述了在一个顾客信用信息的数据库中，通过分类规则判断一个顾客是否具有良好的信用度。通过对数据的分类，使整个数据库更加容易理解，使数据表现得更加直观。

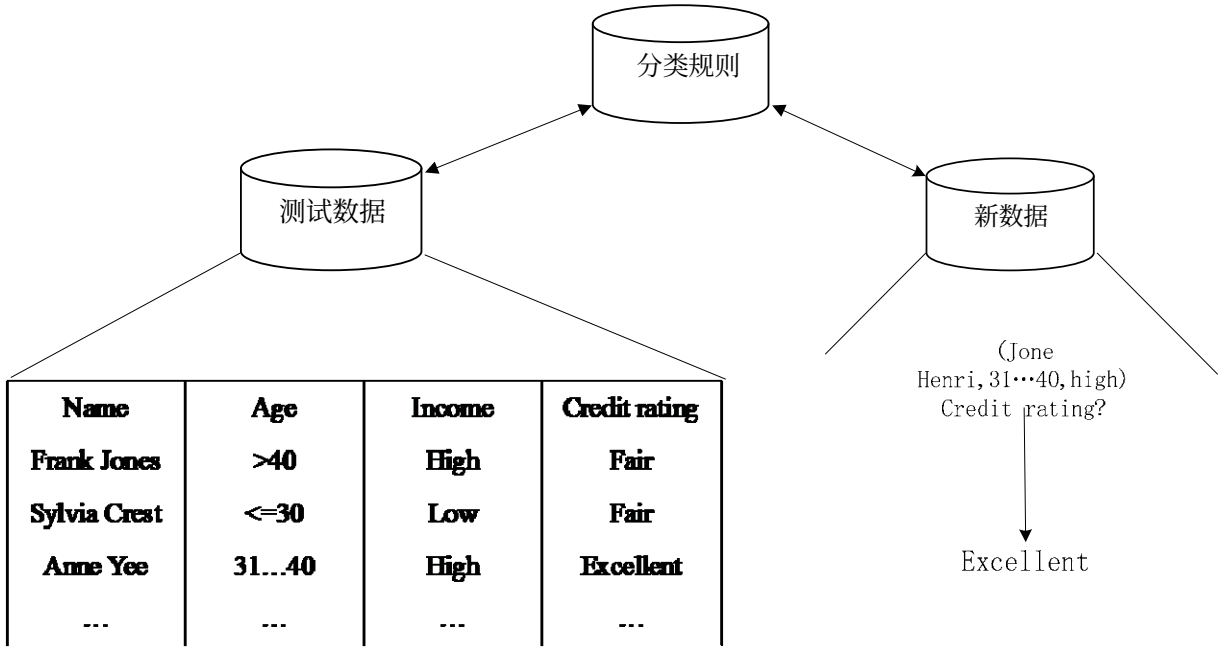
数据分类的第二步就是利用数据模型进行数据分析，如图 3.1(b)。首先对分类方法的准确率进行预估计，测量准确方法有很多，其中保持方法是其中较为简单的一种。所有抽取的样本都是独立于训练样本而随机选出。在分类算法中准确率是很重要的衡量标准，它代表着数据被正确划分的比例。在整个数据挖掘的过程中我们需要将已知的信息规则与数据模型推测的数据结果进行比较，他们的差异越大说明算法准确率越低。如果准确率依据的是训练数据集的评估，这样的结果往往过于乐观。

如果该分类算法的准确率非常高，就可以使用这种方法对未知的样本训练集进行分类。例如在图 3.1(a)中，通过对已知数据进行分析预测顾客的信誉度。




| Name | Age | Income | Credit rating |
|---|---|---|---|
| Sandy Joes | <=30 | Low | Fair |
| Bill Lee | <=30 | Low | Excellent |
| Courtney Fox | 31…40 | High | Excellent |
| Susan Lake | >40 | Med | Fair |
| Claire Phips | >40 | Med | Fair |
| Andre Beau | 31…40 | High | Excellent |
| … | … | … | … |

图 3-1（a）




| Name | Age | Income | Credit rating |
|---|---|---|---|
| Frank Jones | >40 | High | Fair |
| Sylvia Crest | <=30 | Low | Fair |
| Anne Yee | 31…40 | High | Excellent |
| … | … | … | … |

图 3-1（b）

预测是通过建立模型对未知样本进行预估，或者对给定的样本的属性值或区间进行预估，因而分类与回归就成为了预测的最主要的问题。回归是针对连续属

性值的预测，回归更多的是针对离散数据值的预测。

分类与预测在数据挖掘中是十分重要的，这种思想在很多领域得到广泛的应用，例如市场风险判断、医疗诊断、性能评估等。分类与预测更是决策树算法的核心思想与目的。

### 3.1.1 数据准备

很多时候为了提高数据的分类结果与预测结果的准确性，减少分类算法的复杂度，需要对现有数据进行预处理。具体处理步骤如下：

(1) 首先需要对数据进行清理, 从而减少数据的噪音防止很多常用值被遗漏。处理遗漏数值与噪音是很多算法必不可少的机制，这样能降低数据的混乱性，提高分类的准确性。

(2) 需要对样本属性之间的相关性进行分析，这是相当关键的。比如学生交费成功与不成功可能与他具体几号交的是无关的。很多时候许多属性是冗余存在的，这样我们可以通过对属性之间的相关性进行分析剔除冗余属性，避免冗余属性对预测结果产生影响。

(3) 在很多情况下需要对数据进行转换处理，例如数据的泛化。我们可以将温度这样的连续值泛化为冷、适中和热，将连续的数据离散化，这可以大大减少算法的复杂度，同时也使数据看起来更加的直观与简单。

### 3.1.2 分类方法之间的比较

分类与预测方法可以通过比较预测的准确率、速度、强壮型、可规模性以及可解释性来判断什么方法是最适合该数据分类的。

### 3.1.3 判定树分类归纳

判定树[23]简单意义上讲就是一颗类似于流程图的树。树的叶点就是类属性的分布, 而每一个属性测试都在节点内部完成, 每一个分支都会有属性结果的输出。树的最顶层为根节点，在不同的决策树算法中，对根属性的选择的标准是不一样的，ID3 算法选取的是信息增益最大的属性为根节点，而 C4.5 算法依据的是则是信息增益率, CART 算法是依据的是 Gini 系数[24]。图 3.2 是一颗典型的决策树，表示对顾客是否购买计算机的预测。

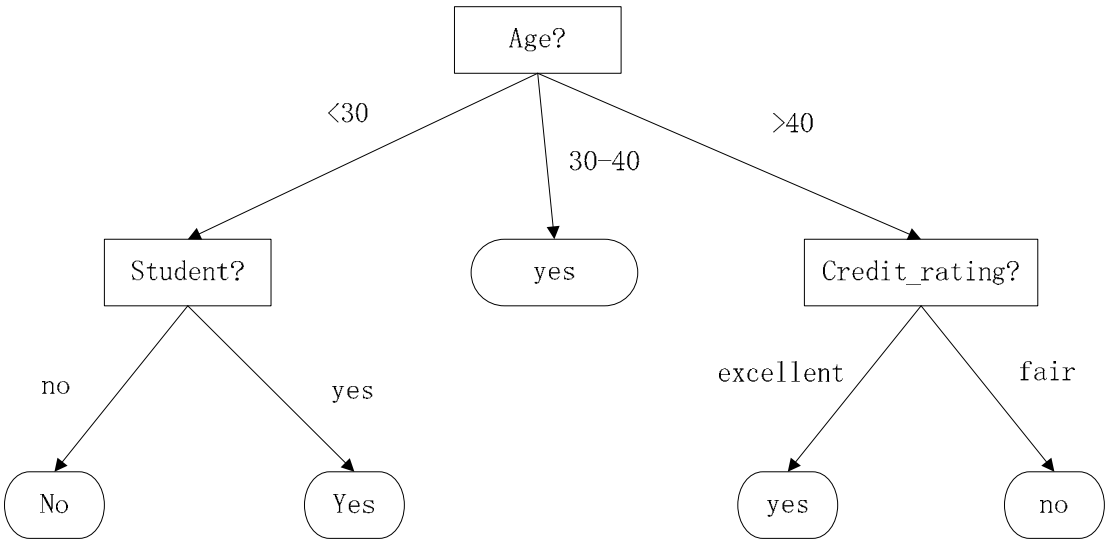



图 3-2 预测顾客是否购买计算机的判定树

## 3.2 决策树算法概念

决策树算法是一种逼近离散函数值的方法。它是众多分类算法中的一种，它的步骤是先要预处理原始数据，通过对原始数据的初步分析建立分类规则，分类规则一般以树的形式出现，通过建立的树对样本训练集进行实质的分析。决策树算法简单来说就是通过一定规则对样本训练集进行分类，只不过分类规则以树的形式出现。决策树模型往往是用于对未知样本的预测，其叶子节点为属性的类别，每一棵树枝都是一组逻辑判断。

## 3.3 决策树基本思想

决策树算法产生于 20 世纪六十、七十年代，J Ross Quinlan 提出了著名的决策树算法 ID3 算法[25]，该算法大大减少了树的深度，使建树规则更加容易被人理解。随后又提出了 C4.5 算法，C4.5 算法对 ID3 算法的不足之处做出了重大改进，比如克服了信息增益偏向属性样本数量多的属性集合，从而大大提高了算法的准确度，扩大了算法适用范围。

决策树算法将分类规则以树的形式表现，它虽然思路非常简单，但是难点在于如何降低决策树算法的复杂度以及提高算法的准度。决策树构造可以分两步进行：

首先，建造决策树：由训练样本集生成决策树的过程。一般情况下我们需要对数据进行预处理以防止树的规模过于庞大。

然后，在生成决策树后需要对树的树枝进行修剪，剔除那些影响算法准确率的叶子节点，例如某些叶子节点的样本数为 0，具体步骤如下[26]：

(1) 树以代表训练样本的单个结点开始。

(2) 如果样本都在同一个类．则该结点成为树叶，并用该类标记。

(3) 否则，算法选择最有分类能力的属性作为决策树的当前结点。

(4) 根据决策树中的叶子节点，将样本训练集划分为不同的训练子集，每一个结果的选取过程都是一棵分枝，分枝的节点标注分类规则。决策树的节点建造是重复的，如果一个属性存在于决策树的末端，表明已经出现结果，所以可以剔除该数据

(5) 如果出现下列情况递归自动停止：

①给定结点的所有样本属于同一类。

②如果出现没有剩余属性可以用来进一步划分样本，那么就把该结点转化为叶子节点，而叶子结点的属性标注为该数据集内数量最多的样本属性，或者将样本的类别分布储存在叶子节点中。

③如果发现叶子节点中没有样本，即为空样本那么就选择类别较多的属性作为新的叶子节点。

3.3.1 决策树算法的伪代码

(1)建立树的根节点，命名为 M；

(2)如果所有的样本训练集属于相同的类 F；

(3)那么就回到整个树木的根节点 M,并将 M 的特征属性表为 F；

(4) 如果样本训练集中的候选属性集合为空集；

(5)那么回到根节点 M，将为空的候选属性集合剔除；

(6) 选择属性集合中“最好”的属性，并且命名为 test_attribute；

(7)将根节点 M 的属性标记为 test_attribute；

(8)如果根属性 test_attribute 可以划分;

(9)那么根据划分值建立树枝；

(9)遍历每一个由根节点 M 生成的枝干；

(10 设集合 a 为根属性 test_attribute 中某一元素的数据集；

(11) 如果 a 为空集；

(12) 那么在根节点 M 出加一个树叶，标记为训练样本集中的多数类；

(13) 否则加上除去根节点属性 test_attribute 的决策树；

返回的结点；

## 3.4 构造方法

在决策树构造过程中，所有属性类都会被标记，决策时既可以是一棵严格的二叉树，也可以是一棵多叉树[27]。构造的结果是一棵二叉树或多叉树。在决策树中每一条数据都代表一个逻辑判断过程，例如对 c 与 $c_j$ 进行逻辑判断，$c_j$ 是属性 c 集合中所有可能的取值，而 c 为属性元素，那么就可以通过决策树的一条树枝来判断最后的结果。决策树还可以生成多叉树，例如 ID3 算法与 C4.5 算法生成的决策树都不一定是严格二叉树。在多叉树中，每一条枝干依然代表着一个逻辑判断过程。

很多时候由于对原始数据的预处理做得少，就会使整个决策树过于复杂，导致算法效率低下。因此需要对决策树进行剪枝，在保证准确率的情况下力求使整个决策树规模最小。对决策树的优化主要集中在以下两个方面：

(1)使叶子节点数最少。

(2)减少整棵树的深度。

## 3.5 举例阐述决策树思想：

例如我们对是否去旅游做一个决策树，决策条件如果放假、经费 2000 以下、景区人不多、旅游时间为 3 天以下就去旅游根据以上条件可以建树：

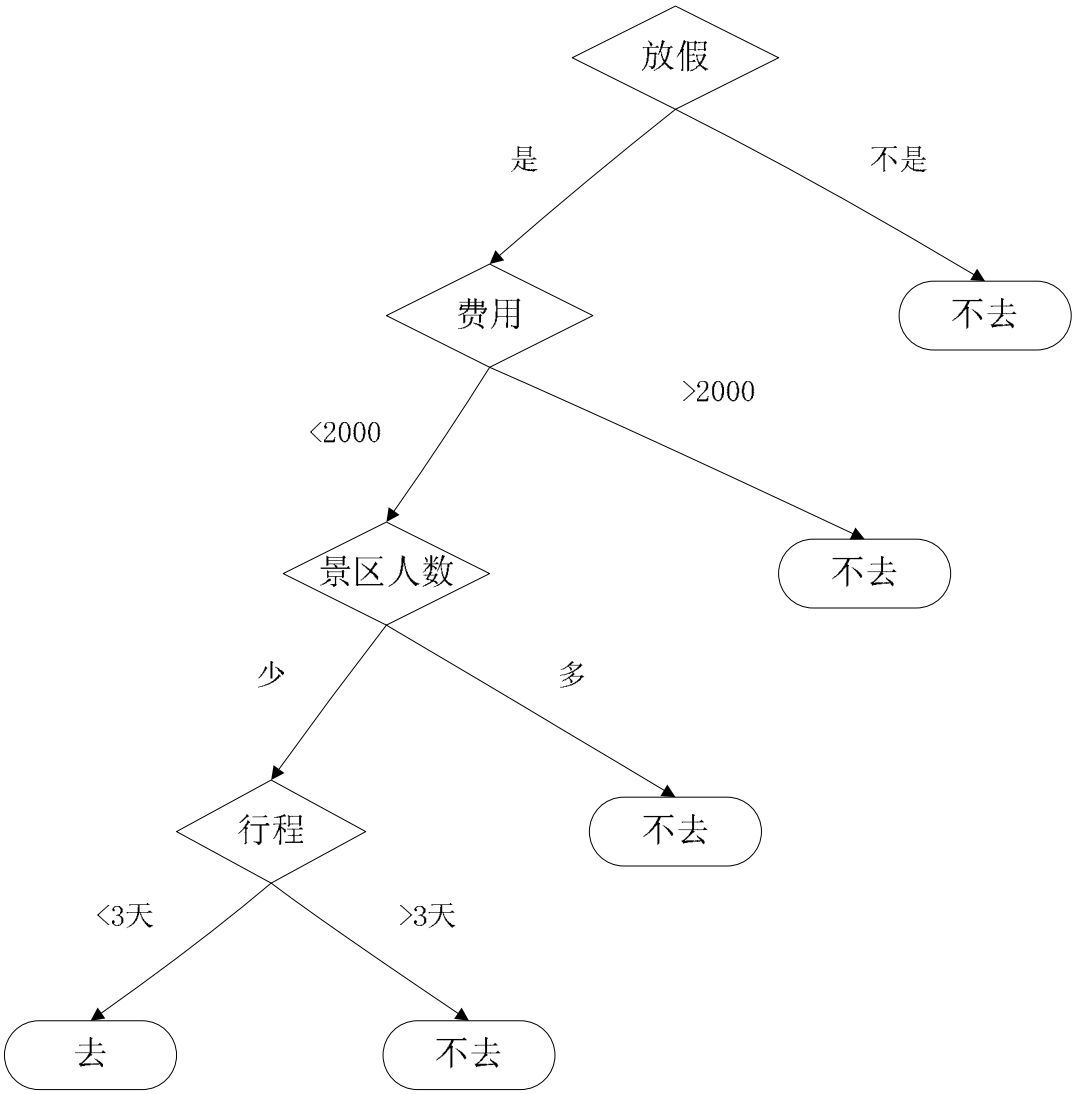



图 3-1 是否旅游的决策树

## 3.6 决策树剪枝

很多时候当决策树创建时，许多分支所反映的训练数据异常，这是由于数据中的噪音或者其余外部因素造成的。面对这样的问题通常的做法是通过对数据进行统计分析，将树中不可靠的分枝减去。这可以大大提高分类的效率以及准确性。

剪枝又分为前枝和后剪枝两种方法。前剪枝是对树先剪枝，停止对树的构造。构造一旦停止就可以对树中的节点评估分裂的优劣，而评估的标准则比如信息增益，方差等数值。之后需要在分裂过程中设定一个阀值[28]，阀值的选取是非常困难的。如果由于样本划分的过于细致产生了一个低于预定义的阀值的分裂，那么将停止指定子集的划分。如果一个阀值设定的过低，那么有可能使整棵树过于

的复杂，如果阀值设定的过高，有可能树就过于简单，从而对最后决策产生不利的影响。

剪枝的第二种方法就是后剪枝，与前剪枝不同的是，后剪枝并不像前者随着树的成长不断地在剪树，而是再等一颗树长完整后，再对整棵树剪枝。通过对数据的分析，减掉多余的分枝，减少树的节点，其中最具代表性的后剪枝算法就是代价性剪枝算法。

## 3.7 决策树提取规则

我们在决策树中提取我们想要的信息时一般采用“IF-THEN”的形式表示[29]，在树中每一个从根部到分枝末端的节点都是一个分类规则，IF 表示的是前提条件，而 THEN 表示的是最后的结果 。“IF-THEN”的表达形式非常容易理解，尤其对复杂度较大的树时更加方便

根据图 3.2 中的决策树，我们可以将从中提取的信息转换为“IF-THEN”的形式如下：

(1)IF AGE=“<=30” AND STUDENT =“NO” THEN BUYS_COMPUTER=“NO”

(2)IF AGE=“<=30” AND STUDENT =“YES” THEN BUYS_COMPUTER=“YES”

(3)IF AGE=“30-40” THEN BUYS_COMPUTER=“YES”

(4)IFAGE=“>40” AND CREDIT_RATING =“EXCELLENT” THEN BUYS_COMPUTER=“NO”

(5)IF AGE=“>40” AND CREDIT_RATING =“FAIR” THEN BUYS_COMPUTER=“YES”

## 3.8 决策树归纳的可规模性

决策树算法对于规模性小的数据库是非常有效的, 无论是 ID3 还是后来改进的 C4.5 算法在小数据库上的表现是非常优秀的。但是当遇到拥有数以百万的数以千万甚至上亿数据的数据库时则显得非常困难。在我们的日常生活中，这种大型的数据库是非常常见的, 因此决策树算法的规模性和有效型已经引起了很多专家学者的关注。决策树算法影响可规模性最大的问题是由于计算时样本训练集需要在高速缓存与主存之间进行频繁的交换，而当训练集过大时，处理速度就会过

于缓慢，从而导致算法低效。

为了针对可规模性这一问题，又提出了新的一些决策树算法例如 SPRNT 算法与 SLLQ 算法。这些算法都是对决策树算法在处理数据量大、数据连续时对算法进行了有效的改进。两种算法最大的改进就是在对数据处理之前，采用预先排序的方法，被排序的对象主要是针对那是数据流量非常大一时无法放入主存的样本训练集。SPRNT 算法与 SLLQ 算法为了使树的建造各家便利，都重新产生了新的数据结构。在 SLLQ 算法中新的数据结构为单个驻留主存的类表以及存在于磁盘空间的属性表。如图 3.3，对于表 3.2 中的数据我们在记录标识中建立索引，为每一个属性建立一个属性表。将样本训练集中的元组在属性表中的具体数值变为类表表目中的一个链接。类表的表目则与决策树中他所对应的叶子节点进行链接。由于决策树在建造和剪枝时需要经常访问类表，所以类表往往存在与主存中。训练集中元组数目越大类表就会越长，如果类表过大导致其无法在主存中存储时，SLLQ 算法效率则会非常低下。因此 SLLO 算法依然会被内存所限制。

| RID | Credit_rating | Age | Buys_computer |
|---|---|---|---|
| 1 | Excellent | 38 | Yes |
| 2 | Excellent | 26 | Yes |
| 3 | Fair | 35 | No |
| 4 | Excellent | 49 | No |

表 3-2 数据表

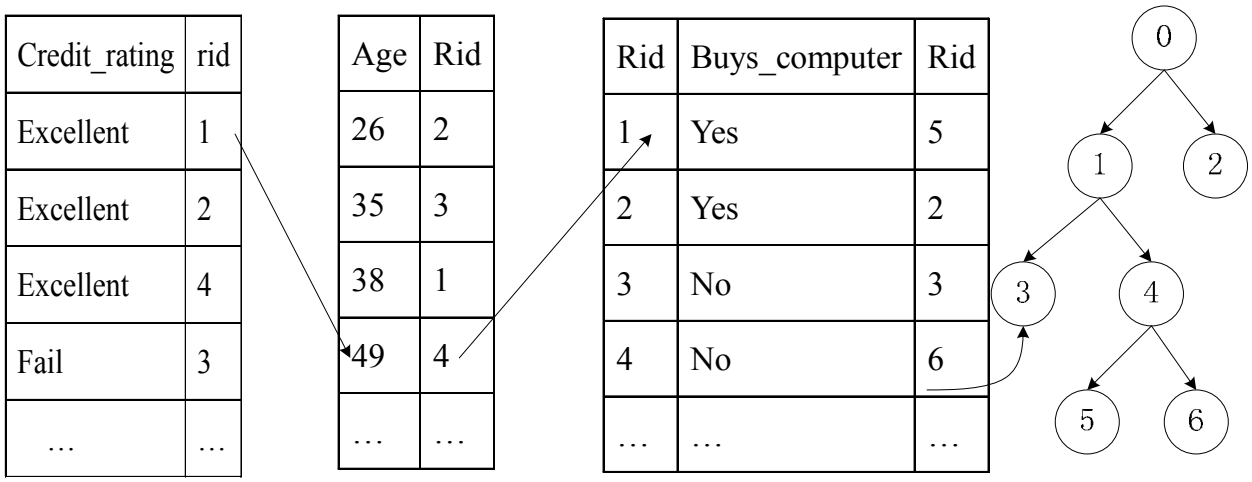



图 3-3 属性结构表

与 SLLQ 算法不同的是，SPRINT 采用完全不一样的属性表结构，RID 信息与存放类如图 3.4 表示，SPPINT 算法中的属性表会随着节点的分裂与之对应分裂，将分裂的信息在结果的子女中分布表示，而表中的记录顺序是保持不变的，这就避免了对表进行重新排列。SPPINT 算法增加了数据处理时的并行运算，所以增加了可规模性。

| Credit_rating | Buys_computer | Rid |
|---|---|---|
| Excellent | Yes | 1 |
| Excellent | Yes | 2 |
| Excellent | No | 4 |
| fair | No | 3 |
| … | … | … |

| Age | Buys_computer | Rid |
|---|---|---|
| 26 | yes | 2 |
| 35 | No | 3 |
| 38 | Yes | 1 |
| 49 | No | 4 |
| … | … | … |

图 3-4 属性表结构

SPRINT 算法虽然解决了内存限制的问题，但是它的算法的复杂度依然与训

练集中元组数目成正比，当数据过于庞大时，算法所消耗的资源也是巨大的。

# 4 C4.5 算法介绍

## 4.1 C4.5 算法概念

C4.5 算法[30]是一种非常重要的分类算法，被评为十大经典算法之首，它是决策树(决策树是一棵倒着树，最上面的节点为根，将决策的节点连接起来像一棵树)的核心算法，并在 ID3 算法的基础上做出了重大改进。C4.5 算法由 Quinlan 提出，用信息增益率代替信息增益度来选择属性，克服了用信息增益选择属性时偏向选择取值多的属性的不足，能够完成对连续属性的离散化处理。

C4.5 算法相对于 ID3 算法改进之后有如下优点：

（1）C4.5 算法的最大改进就是不在用信息增益来选择属性，而是采用信息增率，这么做就可以避免那些样本数量多但却对分类贡献少的属性作为根节点，提高了算法准确率[31]。

（2）对树进行前剪枝，发现数据有问题可以及时处理，不用等到树建完后在对其剪枝，这样就大大提高了算法效率。

（3）能够通过对数据进行泛化，使连续数据离散化，从而增加了对连续数据的处理能力。

（4）在面对有缺失的数据时，C4.5 算法依然能够有效处理。

## 4.2 信息熵

信息熵[32]是由 C. E. Shannon 也就是信息论之父在 1948 年发表的论文通信的数学理论（A Mathematical Theory of Communication）”中提出，文中说明任何信息都存在冗余，冗余大小与信息中每个符号（数字、字母或单词）的出现概率或者说不确定性有关。C. E. Shannon 将热力学的熵引入到信息论中目的就是将信息量化，把信息中排除了冗余后的平均信息量称为“ 信息熵 ”，信息熵计算的数学表达式[33]。

计算公式：

$$H(X)=E[I(x_i)]=E[\log_2(1/p(x_i))]=-\sum p(x_i)\log_2(p(x_i))(i=1,2,\cdots)$$

## 4.3 信息增益

信息增益[34]（Kullback–Leibler divergence）又被称为 Information Divergence, Information Gain，Relative Entropy 或者 KLIC。

信息增益在信息论与概率论中的分布是非对称的, 它的作用是度量两种不同的概率分布间的差异性。例如存在两个完全不同的概率分布 P 和 Q, 信息增益描述的是分别对 P 和 Q 进行编码时，比较两者编码后的结果，两个结果的差异值为信息增益。在很多数据集中，P 往往代表的是样本的观察值，而 Q 代表的是一种理论模型，或者类似于 P 的样本观察值。

信息增益不仅是对属性间信息量的一种简单度量, 就比如信息增益它的非对称性，一个物体从 A 转化为 B 与它从 B 转化为 A 所带的信息增益量是不同的。信息增益是 f 增益（f-divergences）的一种特殊情况。在 1951 年由 Solomon Kullback 和 Richard Leibler 首先提出作为两个分布的直接增益（directed divergence）。它与微积分中的增益不同，但可以从 Bregman 增益（Bregman divergence）推导得到。

信息增益的大小衡量标准是计算每个属性在分类过程中所携带的信息量, 属性信息量越大说明对分类的贡献也就越大, 信息增益也就越大, 反之信息增益就越小。对单一的每个属性, 其信息量的计算是比较该属性存在时整个系统所含信息量与剔除该属性时系统所含信息量, 两者的差值就为该属性的信息量, 也就是信息熵。

## 4.4 信息增益率

信息增益是偏向含有大量值的属性，也就是谁的属性值多谁的信息增益就大，而往往这种划分对分类没有什么作用。比如在一组数据中以 ID 号为属性标识，每一个 ID 号都对应唯一的元组，因此 ID 分裂会使数据的到大量划分，其划分得到的信息量最大，因此信息增益也最大，但是很显然 ID 属性标识对分类起不到任何作用。

因此为了解决这个问题, 又提出了一个新的划分标准就是信息增益率来代替信息增益。信息增益率[35]用分裂信息值将信息增益规范化，简单意义上说就是信息量与属性个数的比值，也就是比较的是单位属性上的信息量，而不在比较信

息总量。

## 4.5 C4.5 算法研究现状

C4.5 算法已经在各个领域有了广泛的应用并且有许多非常成熟的系统，如医疗诊断、语音识别、模式识别、和专家系统等[36]。目前，C4.5 技术面临的挑战表现在以下几个方面：

(1)可扩展性需要提高。在大型数据集中，能快速而准确地发现隐藏于其中的主要分类规则，是算法具有良好的可扩展性的表现。数据挖掘面临的数据一般都是海量的，对于很多实时性要求很高的决策场所，数据挖掘算法的的主动性和快速性显得日益重要。

(2)适应多数据类型和容噪性[37]。随着计算机网络和信息的社会化，数据挖掘的对象已经不再单是关系数据库模型，而是多类型数据库，如今数据的非结构化程度、噪声等现象越来越多，这需要 C4.5 算法能够应对各种数据，这也是 C4.5 算法急需解决的难题。

(3)C4.5 的递增性。数据挖掘出来的规律，可能只是相对于某一特定时间的某些数据，新的数据可能使发现的新知规律与原来的冲突。因此，设计具有递增性 C4.5 算法，也是实用化的基本要求之一。

目前 C4.5 算法已经升级到了 C5.0 算法，C5.0 算法在 C4.5 算法的基础上增加了 Boosting 分类器，但是 C5.0 算法是以产品出现的，现在还不知道其内部的数学逻辑关系。

## 4.6 C4.5 算法描述

(1) 设$T$为数据集，其中的类别集合为$\{C_1, C_2, \cdots C_k\}$，选择其中一个属性 $V$ 把$T$分为多个子集。$V$有互不重合的$n$个取值$\{v_1, v_2, \cdots v_n\}$，则 $T$ 被划分为 $n$ 个子集$\{T_1, T_2, \cdots T_n\}$，其中$T_i$ 中所有实例的取值均为$v_i$。令$|T_i|$为数据集$T$的例子数，$|T_i|$为 $V = v_i$的例子数，$|C_j| = freq(C_j, T)$为$C_j$的例子数，$|C_{jv}|$是 $V = v_i$ 例子中具有类别$C_j$的例子数。则有：

(2) $C_j$发生概率：$P(C_j)=\frac{|C_j|}{|T|}=freq(C_j,T)$。

(3) $V=v_i$发生概率：$P(v_i)=\frac{|T_i|}{|T|}$。

(4) 属性$V=v_i$的例子中，具有类别条件$C_j$的条件概率为：

$P(C_j\,|\,v_j)=\frac{|C_{jv}|}{|T_i|}$。

(5) 类别信息熵计算：

$$Info(C)=-\sum_j P(C_j)\lg P(C_j)=-\sum_{j=1}^{k}\frac{freq(C_j,T)}{|T|}\lg\frac{freq(C_j,T)}{|T|}=Info(T)$$

(6) 类别条件熵计算：

$$Info\left(\frac{C}{V}\right)=-\sum_j P(v_j)\sum_i P\left(\frac{C_j}{v_j}\right)\lg P\left(\frac{C_j}{v_i}\right)=-\sum_{i=1}^{n}\frac{|T_i|}{|v_i|}Info(T_i)=Info(T)$$。

(7) 信息增益：

$$I(C,V)=H(C)-H\left(\frac{C}{V}\right)=Info(T)-Info_v(T)=gain(v)$$。

(8) 属性$v$的信息熵：

$$Info(V)=-\sum_i P(v_i)\lg P(v_i)=-\sum_{i=1}^{n}\frac{|T_i|}{|T|}\lg\frac{|T_i|}{|T|}=split\_Info(v)$$。

(8)信息增益率：

$$gain\_ration(v)=\frac{I(C,V)}{H(V)}=\frac{gain(v)}{split\_Info(v)}$$

## 4.7 C4.5 算法构造决策树过程

（1） 设总样本训练集为$E$；

（2） 如果 E 训练集为空集；

（3） 那么返回一个失败值为 $F$ 的单节点；

（4） 如果训练集 E 由相同的属性类 $C$ 的数据集组成；

（5） 那么返回带有 $C$ 类标记的单节点；

（6） 如果一个无类别分类的含有连续属性的集合 $A$ 为空值；

（7） 那么返回一个样本训练集 $E$ 中样本数量最多的属性值；

（8） 遍历集合 $A$ 中所有元素；

（9） 如果集合 $A$ 中的元素 $A_j$ 为连续属性；

（10） 那么令 $A_j$ 中的最大值为 B1，最小值为 B2；

（11） 执行 *For* 循环，$j$ 初始值为 2, 每次执行完毕 $j$ 加 1, 一直执行到 $j = n - 2$ ；

（12） $B_h = B_1 + j * (B_1 B_n) / n$ ；

（13） 令 B 等于 $A_i$ 元素中最大的信息增益属性值；

（14） 令集合 $A$ 元素中信息增益最大的属性值为 $X$ ；

（15） 建立集合 $X$ 的映射关系集合 $\{X_j \mid j = 1,2,3,4...m\}$ ；

（16） 根据 $\{X_j \mid j = 1,2,3,4...m\}$ 中的数值建立节点，节点标记为 $x_1, x_2, x_{3,...} x_m$ ；

（17） 重复上述过程建立其余子树；

# 5 改进的 C4.5 算法

虽然 C4.5 算法具有简单灵活，易于理解的优点，并且在用信息增益率代替了信息增益度后，避免了因为属性取值多造成实验结果的误差，但是在处理多数据的实验中依然存在算法过于复杂并且误差比较大的情况。因此，本文对 C4.5 算法做了改进，提出了一个新的 C4.5 算法命名为 R-C4.5 算法，目的就在于减少误差以及算法的复杂度

## 5.1 改进方法

设训练集 S 中属性集合为{J}，J 中每一个元素又含有一个集合{V}将训练集 S 分为不同的集合。计算同一属性下不同值的信息熵，如果信息熵值相近即 $|Info(S)_{v1}-Info(s)_{v2}|\leq\varepsilon$，比较两个集合的相似度，如果两个集合相似度>0.8，那么将两个值合并产生一个新的属性值。其中$\varepsilon$的取值需要根据训练集的大小来设定，不同的$\varepsilon$值会影响准确率。$\varepsilon$值越小分类的准确率就越高，一般不应该大于 0.1。

### 5.1.1 思想来源

由于很多时候在数据统计的过程中过于细致，造成了很多数据重复产生冗余，通过对属性熵值以及相似度比较去掉重复数据，减少算法的复杂度，并且可以减少由于数据重复产生的误差。

### 5.1.2 改进的相似系数 Jaccard 系数

在离散集合相似度的计算中，很多时候都会用到 Jaccard 系数，它是计算样本相似性与分散性的一个概率，具体公式如下：

$$J(A,B)=\frac{|A\bigcap B|}{|A\bigcup B|}$$

其中 A，B 为两个离散集合，改进的 Jaccard 系数是将集合 A 或 B 中的每一个元素的个数乘以一个系数$\alpha$，这个系数就是两个集合元素总个数的比值，假设现在 A 中有 m 个元素，B 中有 n 个元素，改进的 Jaccard 系数为：

$$J^*(A,B)=\frac{|A\bigcap \alpha B|}{|A\bigcup \alpha B|}$$

这么做的原因是改进的 Jaccard 系数比较的事集合中元素所占比例的相似度，而不再是以前的元素的个数的相似度，举例说明：集合 A 为{1，1，2，2，3，3}，而集合 B 为{1，2，3}，其中 A 集合中含有一个两个 1，两个 2，两个 3，而集合 B 中则是一个 1，一个 2，一个 3 用 Jaccard 系数计算：

$$J(A,B)=\frac{1+1+1}{1+1+1+1+1+1}=0.5$$

而改进的 Jaccard 的系数由于集合 B 中每一个元素都乘以了 A，B 元素总个数之比，即集合 B 中元素个数都扩大了两倍集合变为{1，1，2，2，3，3}根据改进的系数计算可得：

$$J^*(A,B)=\frac{2+2+2}{2+2+2}=1$$

即两个集合相同。很多时候我们比较的是元素的比例而不简简单单的比较元素的个数。比如在比较两个物质是否是同一物质时，并不能因为 A 物质量多，测出的各个属性含量都高，而 B 物质数量少，测出的物质属性含量少，比较测出的属性值不相等而判断两种物质不属于同一物质。

## 5.2 R-C4.5 算法对用户下载行为的实验分析

本文用改进的 C4.5 算法对天翼宽媒软件的用户下载数据进行了分析，并建立了决策树。

### 5.2.1 天翼宽媒软件

天翼宽媒是电信公司推出的手机软件下载平台, 用户可以在平台内下载喜欢的软件应用 ，系统还会推荐用户可能喜欢的应用。


图 5-1 天翼宽媒界面

5.2.2 服务器端抽取数据

天翼宽媒采用用户注册制度，用户在注册时可以自愿填写出生年月、学历、收入、性别、工作种类（区分学生还是工作人员）等相关信息，其中学历、收入、学生、工作种类采用下拉表格的形式，使用户选择式的填写。本次实验是仅对用户手机游戏下载的行为的分析，我们从天翼宽媒数据库随机抽取 20 个样本并以年龄的大小为顺序进行排列组成以下样本：

| 序号 | 年龄 | 性别 | 收入 | 学历 | 工作种类 | 最喜欢的游戏类型 |
|---|---|---|---|---|---|---|
| 1 | 17-24 岁 | 男 | 2000 以下 | 高中及高中以下 | 学生 | 体育射击类 |
| 2 | 17-24 岁 | 男 | 2000-7000 | 专科-本科 | 学生 | 休闲娱乐智力类 |
| 3 | 17-24 岁 | 女 | 2000 以下 | 专科-本科 | 学生 | 动作角色扮演类 |
| 4 | 17-24 岁 | 女 | 7000 以上 | 专科-本科 | 学生 | 手机网游 |
| 5 | 17-24 岁 | 男 | 7000 以上 | 高中及高中以下 | 在职 | 动作角色扮演类 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | 17-24 岁 | 女 | 2000-7000 | 专科-本科 | 在职 | 手机网游 |
| 7 | 17-24 岁 | 女 | 7000 以上 | 专科-本科 | 学生 | 手机网游 |
| 8 | 17-24 岁 | 男 | 7000 以上 | 专科-本科 | 在职 | 休闲娱乐智力类 |
| 9 | 17-24 岁 | 男 | 2000-7000 | 高中及高中以下 | 在职 | 动作角色扮演类 |
| 10 | 17-24 岁 | 男 | 2000 以下 | 硕士以上 | 学生 | 动作角色扮演类 |
| 11 | 25-30 岁 | 男 | 2000-7000 | 硕士以上 | 学生 | 休闲娱乐智力类 |
| 12 | 25-30 岁 | 女 | 2000 以下 | 高中及高中以下 | 在职 | 体育射击类 |
| 13 | 25-30 岁 | 女 | 2000-7000 | 高中及高中以下 | 学生 | 体育射击类 |
| 14 | 25-30 岁 | 女 | 2000-7000 | 硕士以上 | 学生 | 手机网游 |
| 15 | 25-30 岁 | 男 | 7000 以上 | 专科-本科 | 学生 | 休闲娱乐智力类 |
| 16 | 30 岁以上 | 女 | 7000 以上 | 硕士以上 | 在职 | 手机网游 |
| 17 | 30 岁以上 | 女 | 2000-7000 | 专科-本科 | 学生 | 休闲娱乐智力类 |
| 18 | 30 岁以上 | 男 | 7000 以上 | 硕士以上 | 学生 | 休闲娱乐智力类 |
| 19 | 30 岁以上 | 女 | 2000-7000 | 硕士以上 | 学生 | 休闲娱乐智力类 |
| 20 | 30 岁以上 | 女 | 7000 以上 | 硕士以上 | 学生 | 手机网游 |

表 5-1 随机抽取的用户信息中剔除了用户年龄在 17 岁以下 55 岁以上的数据以及用户信息不完整的数据，即如遇以上两种条件随机在抽取另外的数据代替，并且表中所有属性集合均为左右闭集合

5.2.3 泛化数据

由于以上表格中数据为数据库提取的原始数据，缺乏直观性，并且在数据挖掘过程中属性分析过于繁琐，所以对以上数据进行泛化。年龄方面将 17-24 岁年龄段用青年代替，将 25-30 岁用中青年代替，将 30 岁以上用中年代替 。收入方面将 2000 以下用低来代替，2000-7000 用中来代替，7000 以上用高来代替。同样在学历属性方面，将高中及高中以下学历用低来代替，将专科-本科学历用中来代替，将硕士以上学历用高来代替。分别用 A、B、C、D 来代替体育射击类、动作角色扮演类、休闲娱乐智力类和手机网游，重新整合数据得到如下新的表 5-2：

| 序号 | 年龄 | 性别 | 收入 | 学历 | 工作种类 | 最喜欢的游戏类型 |
|---|---|---|---|---|---|---|
| 1 | 青年 | 男 | 低 | 低 | 学生 | A |
| 2 | 青年 | 男 | 中 | 中 | 学生 | C |
| 3 | 青年 | 女 | 低 | 中 | 学生 | B |
| 4 | 青年 | 女 | 高 | 中 | 学生 | D |
| 5 | 青年 | 男 | 高 | 低 | 在职 | B |
| 6 | 青年 | 女 | 中 | 中 | 在职 | D |
| 7 | 青年 | 男 | 高 | 低 | 学生 | A |
| 8 | 青年 | 男 | 高 | 中 | 在职 | C |
| 9 | 青年 | 男 | 中 | 低 | 在职 | B |
| 10 | 青年 | 男 | 低 | 高 | 学生 | B |
| 11 | 中青年 | 男 | 中 | 高 | 学生 | C |
| 12 | 中青年 | 女 | 低 | 低 | 在职 | A |
| 13 | 中青年 | 女 | 中 | 低 | 学生 | A |
| 14 | 中青年 | 女 | 中 | 高 | 学生 | D |
| 15 | 中青年 | 男 | 高 | 中 | 学生 | C |
| 16 | 中年 | 女 | 高 | 高 | 学生 | D |
| 17 | 中年 | 女 | 中 | 中 | 学生 | D |

| | | | | | | |
|---|---|---|---|---|---|---|
| 18 | 中年 | 男 | 高 | 高 | 学生 | C |
| 19 | 中年 | 女 | 中 | 高 | 在职 | C |
| 20 | 中年 | 女 | 高 | 高 | 在职 | D |

表 5-2 泛化后的数据表

5.2.4 计算属性集合中每个元素的信息熵：

（1）年龄：在年龄这个属性集合中有青年、中青年、中年三个元素，它们分别含有 10 个样本、5 个样本以及 5 个样本，其中青年样本中含有 1 个 A，4 个 B，2 个 C 以及 3 个 D。中青年样本中则是 2 个 A,2 个 C 以及 1 个 D，中年样本中是 2 个 C 以及 3 个 D，根据以上信息可得：

$$Info(S)_{青年} = -\frac{1}{10}\log_2\frac{1}{10} - \frac{4}{10}\log_2\frac{4}{10} - \frac{2}{10}\log_2\frac{2}{10} - \frac{3}{10}\log_2\frac{3}{10}$$

$$=1.846$$

$$Info(S)_{中青年} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{2}{5}\log_2\frac{2}{5} - \frac{1}{5}\log_2\frac{1}{5}$$

$$=1.522$$

$$Info(S)_{中年} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$=0.971$$

（2）性别：在性别属性集合中分别含有男、女两个元素，它们分别含有 9 个样本、11 个样本，其中男样本中含有 1 个 A，3 个 B 以及 5 个 C。女样本中则是 2 个 A,1 个 B,1 个 C 以及 7 个 D，根据以上信息可得：

$$Info(S)_{男} = -\frac{1}{9}\log_2\frac{1}{9} - \frac{3}{9}\log_2\frac{3}{9} - \frac{5}{9}\log_2\frac{5}{9} = 1.351$$

$$Info(S)_{女} = -\frac{2}{11}\log_2\frac{2}{11} - \frac{1}{11}\log_2\frac{1}{11} - \frac{1}{11}\log_2\frac{1}{11} - \frac{7}{11}\log_2\frac{7}{11} = 1.490$$

（3）收入：在收入这个属性集合中有低、中、高 3 个元素，它们分别含有 4 个样本、8 个样本以及 8 个样本，其中低样本中含有 2 个 A，2 个 B。中样本中则是 1 个 A,1 个 B，3 个 C 以及 3 个 D，高样本中是 1 个 A,1 个 B，3 个 C 以及 3 个 D，根据以上信息可得：

$$Info(S)_{高} = -\frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} - \frac{3}{8}\log_2\frac{3}{8} - \frac{3}{8}\log_2\frac{3}{8}$$

$$=1.811$$

$$Info(S)_{低} = -\frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} - \frac{3}{8}\log_2\frac{3}{8} - \frac{3}{8}\log_2\frac{3}{8}$$
$$= 1.811$$
$$Info(S)_{低} = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$
$$= 1.000$$

根据上述计算可知 $Info(S)_{高} = Info(S)_{中}$，这说明数据收入高与收入低中集合中元素比例有可能是相等的，所以进一步计算两集合的相似度。计算集合相似度时采用改进的 Jacccard 系数计算，由于两集合元素个数相等，所以集合收入中的元素所乘系数为 1，即收入高中包括 1 个 A、1 个 B、3 个 C 和 3 个 D，而收入中同样包括 1 个 A、1 个 B、3 个 C 和 3 个 D，可以判定两个集合相等，根据 R-C4.5 算法将中、高这两个属性元素合并，形成新的属性元素命名为中高，两个集合的相似度计算公式如下：

$$J^*(\text{高，中}) = \frac{1+1+3+3}{1+1+3+3} = 1$$

（4）学历：在学历这个属性集合中有低、中、高 3 个元素，它们分别含有 6 个样本、7 个样本以及 7 个样本，其中低样本中含有 4 个 A，1 个 B。中样本中则是 1 个 B，3 个 C 以及 3 个 D，高样本中是 1 个 B，3 个 C 以及 3 个 D，根据以上信息可得：

$$Info(S)_{低} = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$$
$$Info(S)_{中} = -\frac{1}{7}\log_2\frac{1}{7} - \frac{3}{7}\log_2\frac{3}{7} - \frac{3}{7}\log_2\frac{3}{7} \quad = 1.449$$
$$Info(S)_{高} = -\frac{1}{7}\log_2\frac{1}{7} - \frac{3}{7}\log_2\frac{3}{7} - \frac{3}{7}\log_2\frac{3}{7} = 1.449$$

根据上述计算可知者 $|Info(S)_{中} - Info(S)_{高}|$ 的值都小于设定的 $\varepsilon$，所以用改进的 Jaccard 系数计算相似度。

计算集合学历中与集合学历高的相似度，根据改进的 Jaccard 系数将集合学历高中的每个元素的个数乘以 1 可知集合学历高中含有 1 个 B、3 个 C 和 3 个 D，集合学历中含有 1 个 B、3 个 C 和 3 个根据改进的 Jaccard 系数公式：

$$J^*(\text{中，高}) = \frac{1+3+3}{1+3+3} = 1$$

根据 R-C4.5 算法，由于$J^*$(中，高)大于 0.8，所以可以将属性学历中的元素高与中合并，同样形成新的元素中高。

（5）工作种类：在工作种类属性集合中分别含有学生与在职两个元素，它们分别含有 13 个样本、7 个样本，其中学生样本中含有 3 个 A，2 个 B，4 个 C 以及 4 个 D。在职样本中则是 1 个 A，2 个 B，2 个 C 以及 2 个 D。根据以上信息可得：

$$Info(S)_{学生} = -\frac{3}{13}\log_2\frac{3}{13} - \frac{2}{13}\log_2\frac{2}{13} - \frac{4}{13}\log_2\frac{4}{13} - \frac{4}{13}\log_2\frac{4}{13} = 1.349$$

$$\begin{aligned} Info(S)_{在职} &= -\frac{1}{7}\log_2\frac{1}{7} - \frac{2}{7}\log_2\frac{2}{7} - \frac{2}{7}\log_2\frac{2}{7} - \frac{2}{7}\log_2\frac{2}{7} \\ &= 1.350 \end{aligned}$$

根据上述公式可知$|Info(S)_{学生} - Info(S)_{在职}| < \varepsilon$，所以根据改进的 Jaccard 系数计算两者相似度，集合学生中每个元素个数乘以$\frac{13}{7}$，可得集合含有$\frac{13}{7}$个 A、$\frac{26}{7}$个 B、$\frac{26}{7}$个 C 与$\frac{26}{7}$个 D，而集合在职含有 1 个 A、2 个 B、3 个 C 和 2 个 D，根据相似度计算公式：

$$J^*(学生，在职) = \frac{\frac{13}{7} + 2 + \frac{26}{7} + \frac{26}{7}}{\frac{13}{7} + \frac{2}{7} + \frac{26}{7} + \frac{26}{7} + \frac{2}{7} + 2 + \frac{12}{7} + \frac{8}{7}} = 0.766$$

因为$J^*$(学生，在职) 小于 0.8，虽然两属性元素的信息熵近似依然不能合并。

根据以上对所有属性元素信息熵值的计算，我们可以将属性收入中的元素高和中合并，同样我们将属性学历中的元素高与低合并，从而得到一张新的数据表 5-3：

| 序号 | 年龄 | 性别 | 收入 | 学历 | 工作种类 | 最喜欢的游戏类型 |
|---|---|---|---|---|---|---|
| 1 | 青年 | 男 | 低 | 低 | 学生 | A |
| 2 | 青年 | 男 | 中 | 中高 | 学生 | C |
| 3 | 青年 | 女 | 低 | 中高 | 学生 | B |
| 4 | 青年 | 女 | 中高 | 中高 | 学生 | D |
| 5 | 青年 | 男 | 中高 | 低 | 在职 | B |
| 6 | 青年 | 女 | 中高 | 中高 | 在职 | D |
| 7 | 青年 | 男 | 中高 | 低 | 学生 | A |
| 8 | 青年 | 男 | 中高 | 中高 | 在职 | C |
| 9 | 青年 | 男 | 中高 | 低 | 在职 | B |
| 10 | 青年 | 男 | 低 | 中高 | 学生 | B |
| 11 | 中青年 | 男 | 中高 | 中高 | 学生 | C |
| 12 | 中青年 | 女 | 低 | 低 | 在职 | A |
| 13 | 中青年 | 女 | 中高 | 低 | 学生 | A |
| 14 | 中青年 | 女 | 中高 | 中高 | 学生 | D |
| 15 | 中青年 | 男 | 中高 | 中高 | 学生 | C |
| 16 | 中年 | 女 | 中高 | 中高 | 学生 | D |
| 17 | 中年 | 女 | 中高 | 中高 | 学生 | D |
| 18 | 中年 | 男 | 中高 | 中高 | 学生 | C |
| 19 | 中年 | 男 | 中高 | 中高 | 在职 | C |
| 20 | 中年 | 女 | 中高 | 中高 | 在职 | D |

表 5-3 经过属性合并后的数据表

5.2.5 生成决策树

（1）计算样本的信息熵根据公式可得：

$$Info(S) = -\frac{4}{20}\log_2\frac{4}{20} - \frac{4}{20}\log_2\frac{4}{20} - \frac{6}{20}\log_2\frac{6}{20} - \frac{6}{20}\log_2\frac{6}{20}$$

$$= 2.100$$

（2）计算年龄各属性的信息增益率:

属性年龄信息增益率计算：

$$Info(S)_{青年/中青年/中年}=\frac{10}{20}Info(S)_{青年}+\frac{5}{20}Info(S)_{中青年}+\frac{5}{20}Info(S)_{中年}$$

$$=1.546$$

$$Gain(S)_{青年/中青年/中年}=Info(S)-Info(S)_{青年/中青/年中年}$$

$$=0.554$$

$$Splite_{青年/中青年/中年}=-\frac{10}{20}\log_2\frac{10}{20}-\frac{5}{20}\log_2\frac{5}{20}-\frac{5}{20}\log_2\frac{5}{20}$$

$$=2.0$$

$$GainRatio_{年龄}=\frac{Gain(s)_{青年/中青年/中年}}{Splite_{青年/中青/年中年}}$$

$$=0.277$$

属性性别信息增益率计算：

$$Info(S)_{男/女}=\frac{9}{20}Info(S)_{男}+\frac{11}{20}Info(S)_{女}=1.427$$

$$Gain(S)_{男/女}=Info(S)-Info(S)_{男/女}=0.673$$

$$Splite_{男/女}=-\frac{11}{20}\log_2\frac{11}{20}-\frac{9}{20}\log_2\frac{9}{20}$$

$$=0.992$$

$$GainRatio_{性别}=\frac{Gain(s)_{男/女}}{Splite_{男/女}}=0.678$$

属性收入的信息增益率计算：

对属性收入中的元素进行划分，所以需要重新计算新元素的信息熵，根据公式得：

$$Info(S)_{中高}=-\frac{2}{14}\log_2\frac{2}{14}-\frac{2}{14}\log_2\frac{2}{14}-\frac{5}{14}\log_2\frac{5}{14}-\frac{5}{14}\log_2\frac{5}{14}$$

$$=1.863$$

$$Info(S)_{低/中高}=\frac{6}{20}Info(S)_{低}+\frac{14}{20}Info(S)_{中高}$$

$$=1.604$$

$$Gain(S)_{低/中高}=Info(S)-Info(S)_{低/中高}=0.496$$

$$Splite_{低/中高}=-\frac{6}{20}\log_2\frac{6}{20}-\frac{14}{20}\log_2\frac{14}{20}$$

$$=0.881$$

$$GainRatio_{收入}=\frac{Gain(s)_{低/中高}}{Splite_{低/中高}}=0.562$$

属性学历的信息增益率计算：

同样由于对属性学历中的元素进行了合并，需要重新计算元素的信息熵值，根据公式可得：

$$Info(S)_{中高}=-\frac{2}{15}\log_2\frac{2}{15}-\frac{6}{15}\log_2\frac{6}{15}-\frac{7}{15}\log_2\frac{7}{15}$$
$$=1.918$$

$$Info(S)_{低/中高}=\frac{5}{20}Info(S)_{低}+\frac{15}{20}Info(S)_{中高}$$
$$=1.689$$

$$Gain(S)_{低/中高}=Info(S)-Info(S)_{低/中高}=0.411$$

$$Splite_{低/中高}=-\frac{5}{20}\log_2\frac{5}{20}-\frac{15}{20}\log_2\frac{15}{20}$$
$$=0.811$$

$$GainRatio_{学历}=\frac{Gain(s)_{低/中高}}{Splite_{低/中高}}=0.507$$

属性工作种类的信息增益率计算：

$$Info(S)_{学生/在职}=\frac{7}{20}Info(S)_{在职}+\frac{13}{20}Info(S)_{学生}=1.349$$

$$Gain(S)_{学生/在职}=Info(S)-Info(S)_{学生/在职}=0.751$$

$$Splite_{学生/在职}=-\frac{13}{20}\log_2\frac{13}{20}-\frac{7}{20}\log_2\frac{7}{20}$$
$$=0.933$$

$$GainRatio_{工作种类}=\frac{Gain(s)_{学生/在职}}{Splite_{学生/在职}}=0.804$$

根据以上计算可知工作种类的信息增益率最大，因此根据规则建树可得：

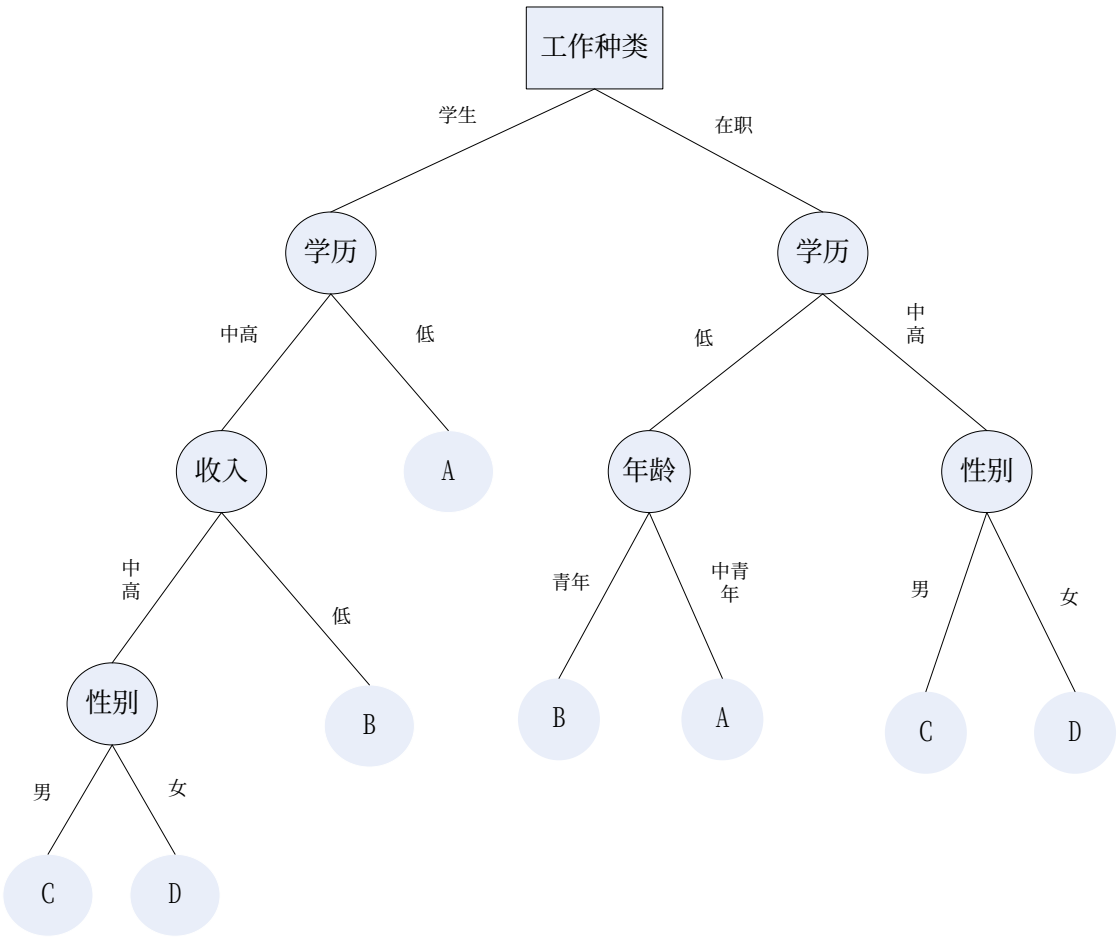



图 5-2 R-C4.5 算法决策树

根据上图路径我们可得以下结论：

1. IF:“工作种类=学生” AND“学历高中及高中以下”THEN 用户喜欢体育射击类游戏
2. IF:“工作种类=学生” AND“收入<2000”AND“专科以上学历”AND“年龄=青年”THEN 用户喜欢动作角色扮演类游戏
3. IF:“工作种类=学生” AND“收入>2000”AND“专科以上学历”AND“性别=男” THEN 喜欢休闲智力类游戏
4. IF:“工作种类=学生” AND“收入>2000”AND“专科以上学历”AND“性别=女” THEN 喜欢手机游戏类
5. IF:“工作种类=在职” AND“学历高中及高中以下”AND“年龄在 17-24 岁”THEN 用户喜欢角色扮演类游戏
6. IF:“工作种类=在职” AND“学历高中及高中以下”AND“年龄在 24-30

岁”THEN 用户喜欢体育射击类

7. IF:“工作种类=在职” AND“专科以上学历”AND“性别=男”THEN 用户喜欢休闲智力类游戏
8. IF:“工作种类=在职” AND“专科以上学历”AND“性别=男”THEN 用户喜欢手机游戏类

5.2.6 与 C4.5 算法比较

由于属性年龄、性别、工作性质的信息增益率已经计算出，现在只需要计算学历与收入的信息增益率，计算过程如下：

（1）收入的信息增益率计算

$$Info(S)_{低/中/高} = \frac{4}{20} Info(S)_{低} + \frac{8}{20} Info(S)_{中} + \frac{8}{20} Info(S)_{高}$$
$$= 1.948$$

$$Gain(S)_{低/中/高} = Info(S) - Info(S)_{低/中/高}$$
$$= 0.151$$

$$Splite_{低/中/高} = -\frac{4}{20}\log_2\frac{4}{20} - \frac{8}{20}\log_2\frac{8}{20} - \frac{8}{20}\log_2\frac{8}{20}$$
$$= 1.521$$

$$GainRatio_{收入} = \frac{Gain(s)_{低/中/高}}{Splite_{低/中/高}} = 0.100$$

（2）属性学历的信息增益率计算：

$$Info(S)_{低/中/高} = \frac{6}{20} Info(S)_{低} + \frac{7}{20} Info(S)_{中} + \frac{7}{20} Info(S)_{高}$$
$$= 1.290$$

$$Gain(S)_{低/中/高} = Info(S) - Info(S)_{低/中/高}$$
$$= 0.810$$

$$Splite_{低/中/高} = -\frac{7}{20}\log_2\frac{7}{20} - \frac{6}{20}\log_2\frac{6}{20} - \frac{7}{20}\log_2\frac{7}{20}$$
$$= 1.581$$

$$GainRatio_{学历} = \frac{Gain(s)_{低/中/高}}{Splite_{低/中/高}} = 0.512$$

根据上述信息建树可得：

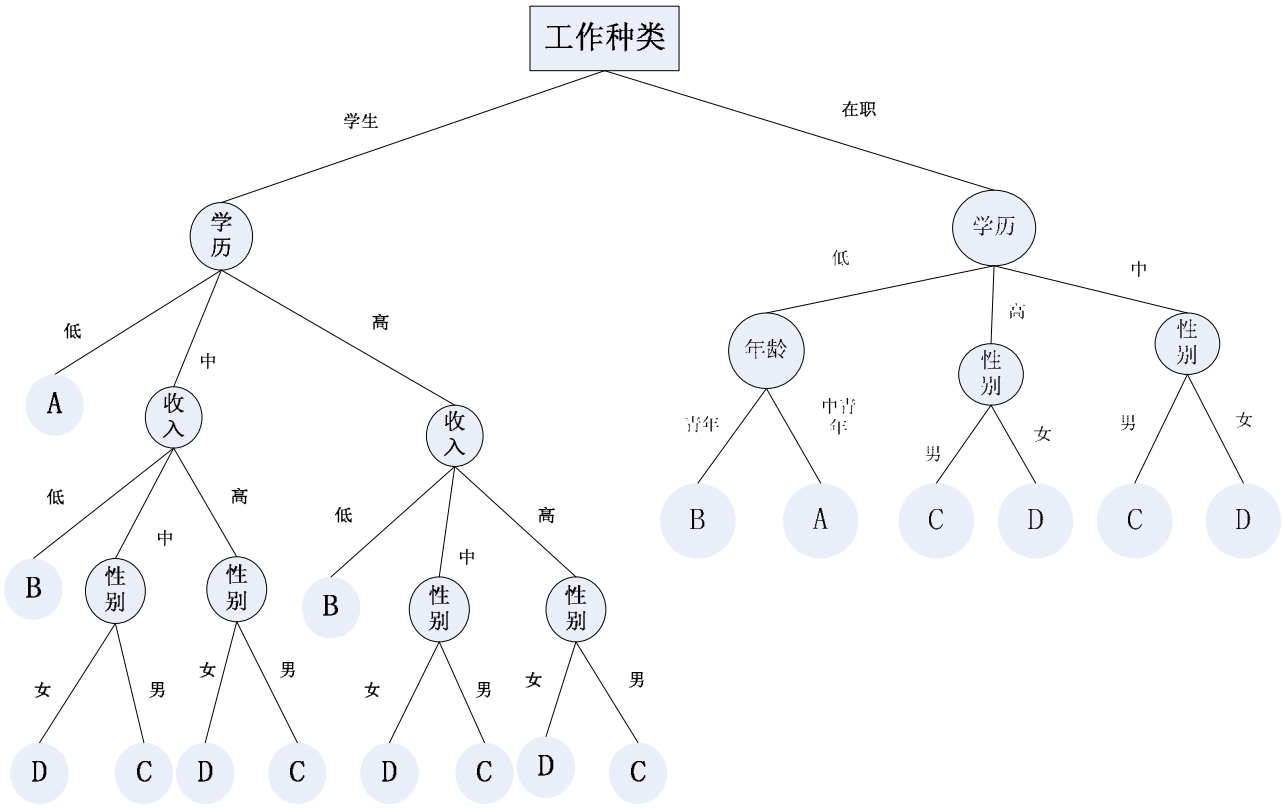



图 5-3　C4.5 算法决策树

R-C4.5 算法与 C4.5 算法实验对比：

| 模型 | 树的层次 | 叶子节点数 | 叶子节点与样本的比例值 |
|---|---|---|---|
| C4.5 算法 | 5 | 8 | (20/8)/20*100%=12.50% |
| R-C4.5 算法 | 5 | 17 | (20/17)/20*100%=5.88% |

表 5-4 实验结果

根据表 5-4 对比，我们可以发现改进后的 C4.5 算法树的复杂程度要小于传统的 C4.5 算法，叶子节点与样本的比例值也要大于传统的 C4.5 算法建立的决策树，其中叶子节点数减少了 9 个，是整棵树更加精简。由于剔除了原数据中的冗余属性，算法的准确率也比 C4.5 算法要高，从而根据以上结果可知，改进的 C4.5

算法更加高效。

# 6 结束语

C4.5 算法在数据挖掘中应用的非常广，所以对它研究是必要的，本文是针对 C4.5 算法不足，通过引入信息熵约简的理论做出了些许改进，减少了算法的复杂度，根据实验对比证明改进是有效的。但是改进的 C4.5 算法依然有很多问题：

(1) 面对连续数据时处理起来依然比较困难。

(2) 改进的算法中参数$\varepsilon$的值不是固定值，它的设定需要有一定的经验，如果$\varepsilon$选的过大或过小都会影响整个数据结果，选取合适的$\varepsilon$值是不易的。

(3) 算法建树之前需要对比属性的信息熵，从而进一步对比属性的相似度，很多时候虽然信息熵相同但是属性并不相似，但是无论是否相似都需要进行计算过程，所以有可能导致算法低效。

# 参考文献


[1] Jiawei Han, Mincherline Kamber. 数据挖掘概念与技术（范明，孟晓峰译），北京：机械工业出版社，2006.15~16 页

[2] 徐义峰,徐云青,诸葛理绣. 基于 SQL 的 OLAP 多维数据分析[J]. 微机发展. 2005(07)

[3] Jan M. Żytkow,Robert Zembowicz. Database exploration in search of regularities [J],1993.

[4] Yang Bingru.Double-Base Cooperating Mechanism in KDD. International Symposium on Computer . 1998

[5] 姜丽萍. 组件式数据挖掘系统的研究与实现 [D].哈尔滨工程大学. 2006

[6] 谢秋丽. 基于关联规则的教学质量评价数据挖掘[J]. 现代计算机(专业版). 2008(06)

[7] 石建,孔祥成,苏春萍. 论个性化信息提取中的 Web 挖掘技术[J]. 情报杂志. 2003(02)

[8] 胡彩平,秦小麟. 空间数据挖掘研究综述[J]. 计算机科学. 2007(05)

[9] Supratik B,Sue M.Network Performance Monitoring and Measurement: Techniques and Experience. MMNS Tutorial . 2002

[10] Chiang J,Yin Z.Unsupervised minor prototype detection using an adaptive population partitioning algorithm. Pattern Recognition . 2007

[11] 段丹青,陈松乔,杨卫平. 网络入侵检测中的支持向量机主动学习算法[J]. 计算机工程与应用. 2006(01)

[12] University of California. Knowledge discovery in databases DARPA archive.Task description. http://www.kdd.ics.uci.edu/databases/kddcup99/task.html . 2010

[13] AU WH,CHAN KCC,YAO X.A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction. IEEE Transactions on Evolutionary Computation. 2003

[14] Hui Tak Kee,Wan David,Ho Alvin.Tourists satisfaction, recommendation and revisiting Singapore. Tourism Management. 2007

[15] EYTAN B.Facebook research report: the importance of social network of weak ties. http://tech.sina.com.cn/i/2012-01-18/13286651169.shtml . 2012

[16] Gundimada S, Asari V K. Facial Recognition Using Multisensory Images Based on Localized Kernel Eigen Spaces. IEEE Transactions on Image Processing. 2009

[17] 张鹏,于静泊,郑雄飞. Deformation Behaviors of Laser Forming of Ring Sheet Metals [J]. Tsinghua Science and Technology. 2009(S1).

[18] 李鹏. 基于贝叶斯理论的神经网络算法研究[J]. 光机电信息. 2011(01)

[19] 赵建立,高会生,赵生岗. 贝叶斯网络在可靠性评估中的应用[J]. 电力科学与工程. 2008(02)

[20] Zhang Tian.BIRCH:a new data clustering algorithm and its applications. 2002

[21] 艾芳菊. 基于实例推理系统中的权重分析[J]. 计算机应用. 2005(05)

[22] 王宏威. 基于决策树的分类算法研究[J]. 软件导刊. 2007(17)

[23] 马廷斌,徐芬. 判定树归纳分类研究[J]. 科技信息. 2009(13)

[24] 冯振华,李金叶,崔道忠. 基于曲率为结构权重的对基尼系数几何算法的改进[J]. 数量经济技术经济研究. 2012(01)

[25] 韩少锋,陈立潮. 数据挖掘技术及应用综述[J]. 机械管理开发. 2006(02)

[26] 蔡科玉.基于改进决策树的网络入侵检测[D].西安电子科技大学.2008

[27] 丁治国,朱学永,郭立,古今. 自适应多叉树防碰撞算法研究[J]. 自动化学报. 2010(02)

[28] 刘汉中. 平稳阈值自回归下的伪回归研究[J]. 统计研究. 2011(01)

[29] 杜来红. 语义 Web 知识表示方法及应用[J]. 价值工程. 2011(16)

[30] Kuwazawa Y, Bashir W,Pope MH, et al.Biomechanical aspects of the cervical cord: effects of postural changes in healthy volunteers using positional magnetic resonance imaging. Journal of Spinal Disorders. 2006

[31] Aplan L,Bronstein Y,Barzilay Y, et al.Canal expansive laminoplasty in the management of cervical spondylotic myelopathy. The Israel Medical Association Journal. 2006

[32] Shannon, Claude E. (July/October 1948). "A Mathematical Theory of Communication". Bell System Technical Journal 27 (3): 379–423.

[33] Chengyi Liu,Hong Guo,Wei Hu,Ximing Deng. A Schrödinger formulation research for light beam propagation [J], 2000

[34] 许朝阳. 文本分类中特征选择方法的分析和改进[J]. 计算机与现代化. 2010(04)

[35] 李强. 创建决策树算法的比较研究——ID3,C4.5,C5.0 算法的比较[J]. 甘肃科学学报. 2006(04)

[36] 决策树 C4.5 算法在数据挖掘中的分析及其应用[J]. 计算机与现代化. 2008(12)

[37] 吴磊,邓涛,陈元奇. 基于 LM 算法的 BP 神经网络在 NAO 模型运动学求逆解中的应用[J]. 软件导刊. 2011(07)

# 致谢

本研究及学位论文在我的导师魏志强教授的亲切关怀和悉心指导下完成的。他严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深地感染和激励着我。魏老师不仅在学业上给我以精心指导，同时还在思想、生活上给我以无微不至的关怀，在此谨向魏老师致以诚挚的谢意和崇高的敬意。我还要感谢在一起愉快的度过毕业论文小组的同学们，正是由于你们的帮助和支持，我才能克服一个一个的困难和疑惑，直至本文的顺利完成。

最后感谢我的母校中国海洋大学，海洋大学浓厚的学术氛围，舒适的学习环境我将终生难忘！祝母校蒸蒸日上，永创辉煌！

# 个人简历

1988 年 10 月 20 日出生于山东省德州市。

2007 年 9 月考入山东财经大学会计学院会计专业，2011 年 7 月本科毕业并获得管理学学学士学位。

2011 年 9 月考入中国海洋大学信息科学与工程学院软件工程专业攻读硕士学位至今。

中国海洋大学 OCEAN UNIVERSITY OF CHINA 硕士学位论文