扬州大学

硕士学位论文

数据挖掘在高校固定资产管理中的应用研究

姓名：朱锡亮

申请学位级别：硕士

专业：计算机技术

指导教师：李斌

20100401

摘 要

高校的固定资产是教学、科研、生活后勤及产业的物资保障。随着高校固定资产规模的扩大，固定资产管理也处在不断的改进之中，管理模式由原来的手工管理发展到现在的计算机辅助管理。近年来由于学校合并使得学校分区办校、校区分散，同时学校的学科体系调整较大，固定资产的变动较大，再则随着政府投资力度的加大以及科研经费的增多，使得学校固定资产尤其是仪器设备急剧增加，因而在固定资产管理中常常需要处理大量的数据信息。另外，随着学校实验室的开放，教学、科研工作量与对外交流的增加，越来越多的校内外师生与研究人员希望了解学校教学、科研仪器设备的简要情况，为教学科研工作提供帮助，从而对学校的设备资产信息共享提出了日益迫切的要求。

高校的快速发展促使学校内部收集了大量的数据，并且迫切需要将这些数据转换成有用的信息和资料，为资产的管理提供有效的保障，本文就数据挖掘技术中决策树学习算法运用于资产的优化管理分配问题进行了探讨。

第一部分概述了高校资产管理课题研究的意义及当前国内外研究的现状，并简要评述了我国高校资产管理的现状；第二部分描述了数据挖掘技术；第三部分对固定资产管理系统进行需要分析，设计并实现了固定资产管理系统；第四部分对固定资产管理的资产数据进行预处理，描述了决策树算法及在我校资产管理的应用。

关键词：数据挖掘；资产管理；管理信息系统；决策树算法

Abstract

The fixed assets of university are the material support of teaching, scientific research, logistics and industrial life. With the expansion of the scale of fixed assets, its management is also in continuous improvement, and the management model is from the manual to the computer-aided management. In recent years, the school merger makes the campus scattered and disciplinary system adjust more. Meanwhile, the government increases the investment and the research funding so that school equipment has increased sharply, and a great deal of data needs analyzing. In addition, with the opening of school laboratories and the increasing of the exchange in teaching, research and with foreign countries, more and more teachers, students and researchers hope to understand the teaching and scientific research equipment to provide help for research and urgent demands for sharing information.

The rapid development of universities promotes the collection of a great deal of data and urgently need to convert these data into useful information so as to secure for asset management. This paper discusses the data mining technology used in the decision tree in learning algorithm for optimal management of distribution assets.

The first part outlines the significance of asset management at universities and the current status of domestic and foreign research, and briefly reviews the status of asset management in Chinese universities. The second part describes the data mining. The third part makes analysis on the fixed assets management system, designs and implements the fixed assets management system. The fourth part makes a preprocessing to the fixed asset data management, and describes the decision tree algorithm and asset management in the application of our school.

Keywords: data mining; asset management; management information systems; decision tree algorithm

扬州大学学位论文原创性声明和版权使用授权书

学位论文原创性声明

本人声明：所呈交的学位论文是在导师指导下独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含其他个人或集体已经发表的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者签名：

签字日期：　　年　月　日

学位论文版权使用授权书

本人完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交学位论文的复印件和电子文档，允许论文被查阅和借阅。本人授权扬州大学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》，并通过网络向社会公众提供信息服务。

学位论文作者签名：　　导师签名：

签字日期：　　年　月　日　　签字日期：　　年　月　日

第1章 引言

随着经济全球化进程的加速和高等教育的大众化，教育越来越成为人们关注的焦点，高校绩效水平的提高对经济和文化的基础作用日益显著。随着国家公共财政体制改革的进行，市场经济体条件下所有权和经营权的分离，经过大量的理论研究和实践探索，我国国有资产管理理论基本确立，传统的资产管理已经不能适应实际需要，在管理工作中存在着体制不顺、配置不公、产权不清、监管不严、隐性流失的问题。研究高校资产管理问题，建立健全资产管理体制，创新资产管理模式，探索现代化的管理方式，成为高校财务管理的重要课题。

1.1 研究背景

目前，各普通高职校都经历着教育部的校区合并重组工作，在固定资产的管理过程中存在着诸多实际问题. 由于我国高校大部分是公办性质，属于事业单位管理体制，高校的固定资产主要通过国家财政投资建设而成。长期以来高校无偿地使用固定资产，不计提固定资产折旧. 也不需要进行成本核算. 导致了部分高校、职业院校负责人对财务管理特别是固定资产管理和教育成本核算的重要性认识不足；其次,不少高校领导忙于学校的教学、科研和行政工作。忽视了高校固定资产管理工作，导致高校固定资产管理机制不健全、管理手段落后、管理工作缺乏科学性. 从而造成高校普遍存在着重钱轻物、重购建轻管理、公物私用、重复建设、资源浪费的现象。使固定资产难以发挥其在高校教学科研中应有的作用，影响了学校的可持续发展。在我国，国有资产流失不仅是国有企业存在的个重大问题，在高校中也有类似现象. 特别是随着近几年高校扩招，办学规模的不断扩大，使得这一问题变得越来越严重。

由于固定资产管理体制没有理顺．管理制度不完善和执行监督不力，高校固定资产流失状况令人担忧。某些部门对转为经营性资产的固定资产不按有关规定进行评估检查，对出租、出借、转让的固定资产不按规定办理资产出租、出借、转人手续，甚至也不收取使用费用，使得固定资产被经营单位无偿占有和使用，高校资产投入得不到合理的补偿，资产的安全性、完整性也得不到保证，更谈不上资产的保值增值，造成国有资产的流失。目前，大部分高校对固定资产管理实行的是以账物分管为原则的分类归口管理模式。如后勤管理部门管理房屋、建筑物；图书馆管理图书杂志；设备处管理设备器：财务处负责固定资产价值核算。而各职能部门又分别归不同学校领导分管，学校的固定资产管理缺乏一个统一领导、统一管理、权力集中的综合监督协调部门，造成固定资产实物流动与财务核算相脱节，职能管理部门与各教学、科研等占有使用部门相分离的状态。这种条块分割的管理模式．最终导致高校固定资产普遍存在账实不符、家底不清、资源浪费、资产流失等问题。目前．我国大部分高校仍用收付实现制会计核算方法，在固定资产核算中还存在着一些不合理的地方，造成高校财务报表中普遍存在着虚增资产、成本核算不真、资产更新资金不足等问题。根据高等学校会计制度规定，固定资产只核算原值，不计提折旧。固定资产的账面价值除了清理报废外，入账后数据一直不变。由于固定资产的账面价值只反映历史成本，使固定资产的账面价值与实际价值相背离。并使资产负债表中的账面余额不能反映其客观情况，从而导致虚增净资产。这种会计核算方法既违背了会计核算的配比原则．也违背了会计核算的真实性原则。

二十世纪，数据库技术取得了决定性的成果并且已经得到广泛的应用。这表明，我们已具备将这些“数据洪流”转换为“整齐有序”但却“堆积如山”数据集合的能力。但是，面对“堆积如山”的数据集合，数据库所能做到的只是对数据库中已有的数据进行存取和简单的操作，通过这些数据所获得的信息量仅仅是

整个数据库中信息量的很少一部分，隐藏在这些数据之后的关于这些数据的更重要的整体特征的描述及对其发展趋势预测的信息却无法得到，而这些信息在决定制定过程中具有重要的参考价值。在需要对大量数据分析之后才能做出正确决策的领域中，这已是普遍存在的问题。这样，快速的数据产生与搜索技术和拙劣的数据分析方法之间形成了鲜明的对照，需要新的技术来“智能地”和“自动地”分析这些原始数据，面对这一挑战，数据挖掘技术应运而生，并显示出强大的生命力。数据挖掘技术可以高度自动地和智能地分析原有的数据，从大量的数据中发现隐藏于其后的规律或数据间的关系，从中挖掘出潜在的模式获取有意义的信息，归纳出有用的结构，帮助决策者做出正确的决策，它通常采用机器自动识别的方式，不需要更多的人工干预，是目前国际上在数据库、数据仓库和信息决策领域最前沿的研究方向之一，也是计算机科学和技术应用的一大研究热点。如今，越来越多的研究投向了数据挖掘。在现有技术中，数据挖掘主要应用于科学研究、市场营销、金融投资、真假甄别、产品制造、通信网络管理以及Internet应用等方面。从以上应用来看，数据挖掘的研究主要是面向商业应用尤其是电子商务的，很少应用于非商业机构，尤其是与校园信息网的结合…还不够广泛。本课题将数据挖掘技术应用到固定资产管理系统中.

1.2 主要研究现状

由于历史原因，学校的数据库不少是分布、异构的。大量信息必须通过数据库系统才能有效管理。那么，如何建立合理高效的数据库，成为我校迫切需要解决的问题。

而数据仓库[1,2]和数据挖掘技术[3,4]正好为上述问题提供了一种很好的解决方法[5-9]，我们可以用资产数据系统中的各种类型数据集中建立起资产数据仓库

（DWS），主管部门的人员可以通过联机分析（OLAP）[10-13]中灵活多变的多维分析查询，从不同角度分析整个校区的资产情况，预测未来的资产出入情况；监察人员通过报表（Reporting）工具获得其需要的报表。资产管理各个部门通过数据仓库和数据挖掘技术，以真正实现数据的共享，实现全面有效的分析和预测。

1.3 研究的主要内容

本文研究工作的目的是以学校固定资产数据为分析对象，采用已有数据挖掘算法进行适应性研究开发，并为今后的研究工作打下举实的基础，我们将在以下几方面开展具体工作：

（1）对学校现有固定资产管理系统相关数据进行集成，并进行必要的预处理，形成用于挖掘的数据仓库系统。

（2）剖析固定资产管理系统相关数据可能挖掘的知识[14]。

（3）重点探讨决策树ID3[15-20]算法，以及对资产分类、资产采购、资产折旧、资产报损等各项特征等给出分析结果，对资产管理部门与学校主管部门给予辅助性决策。

1.4 论文组织结构

论文以下章节的组织结构如下：

第一章，引言。首先介绍了本文的课题背景，阐述了本课题的研究目的以及意义，然后简单介绍了本课题研究的主要内容，以及本文的组织结构。

第二章，基本理论。本章主要介绍数据仓库以及数据挖掘技术技术，包含数据仓库概念，数据仓库的组成，数据挖掘的方法与步骤等。

第三章，对固定资产管理系统进行需要分析以及总体框架设计。

第四章，固定资产管理系统的设计与实现，构建具体功能，分析其功能与操作流程。

第五章，固定资产管理系统的数据预处理分析，合理构建数据仓库。

第六章，探讨决策树 ID3 算法在固定资产管理中的应用。

第七章，总结与未来的工作。介绍本文讨论的主要成果，以及未来的研究方向。

第二章 相关理论技术

本章主要讨论数据仓库的理论知识，包括数据仓库的定义[21-22]、数据仓库（Data Warehouse）与数据库（Database）的区别、数据仓库的组成、数据仓库的设计方法，数据仓库的建立步骤等相关理论。

2.1 数据仓库

2.1.1 数据仓库的定义

自从数据仓库[21]概念出现以来，不同学者从不同的角度为数据仓库下了不同的定义。Informix公司的定义：数据仓库将分布在企业网络中不同信息岛上的业务数据集成到一起，存储在一个单一的集成关系型数据库[22-26]中，利用这种集成信息，可方便用户对信息的访问，更可使决策人员对一段时间内的历史数据进行分析，研究事务发展走势。

SAS软件研究所的定义：数据仓库是一种管理技术，旨在通过通畅、合理、全面的信息管理，达到有效的决策支持。

斯坦福大学数据仓库研究小组的定义：数据仓库是集成信息的存储中心，这些信息可用于查询或分析。数据仓库公司RedBrickSystem的定义是：数据仓库是特别为信息检索而设计的关系数据库管理系统。

我国著名数据库专家王珊将其定义为：数据仓库是一个用以更好地支持企业或组织的决策分析处理的、面向主题的、集成的、不可更新的、随时间不断变化的数据集合。

目前，大家公认的数据仓库之父 W. H. Inmon 在 1992 年所著{Building the DataWarehouse}一书中对数据仓库的定义最具权威性，他认为数据仓库是一个面向

主题的(Subject oriented)、集成的(Integrated)、非易失的(Nonvolati le)且随时间变化(Time—variant)的数据集合，用来支持管理人员的决策分析。对于数据仓库的概念我们可以从两个层次予以理解，首先，数据仓库用于支持决策，面向分析型数据处理，它不同于企业现有的操作型数据库；其次，数据仓库是对多个异构的数据源有效集成，集成后按照主题进行了重组，并包含历史数据，而且存放在数据仓库中的数据一般不再修改。

随着人们对数据系统研究、管理、维护等方面的深刻识认和不断完善，在总结、丰富、集中多行企业信息的经验之后，为数据仓库给出了更为精确的定义，即“数据仓库是在企业管理和决策中面向主题的、集成的、与时间相关的、不可修改的数据集合”。数据仓库并没有严格的数据理论基础，也没有成熟的基本模式，且更偏向于工程，具有强烈的工程性。通常按其关键技术分为数据的抽取、存储与管理以及数据的表现等三个基本方面。

数据仓库的重点与要求是能够准确、安全、可靠地从数据库中取出数据，经过加工转换成有规律信息之后，再供管理人员进行分析使用。数据仓库主要是应用于决策支持系统，其主要目的是“提取”信息并加以扩展，用来进行处理基于数据仓库的决策支持系统(DSS) 的应用。

2.1.2 数据仓库的特征

从W.H.Inmon对数据仓库的定义中，我们可以发现数据仓库具有这样一些重要的特征：面向主题性、集成性、时变性、非易失性、集合性。

1．面向主题性

面向主题性是数据仓库中数据组织的基本原则，数据仓库中的所有围绕着某一主题组织、展开的。主题是与传统数据库的面向应用相对应的，是一个抽象的

概念，是在较高层次上将企业信息系统中的数据综合归类并进行分析利用的抽象。在逻辑意义上，它是对应企业中某一宏观分析领域所涉及的分析对象。从信息管理角度看，主题就是在一个较高的管理层次上对信息系统中的数据按照某一具体的管理对象进行综合、归类所形成的分析对象。

2. 集成性

数据仓库的集成性是指根据决策分析的要求，将分散于各处的源数据进行抽取、筛选、清理、综合等集成工作，使数据仓库中的数据具有集成性。数据仓库所需要的数据通常来源于不同的数据源(如关系数据库、一般文件和联机事务处理记录)，这些数据只为业务的日常处理服务，而不是为决策分析服务。所以，首先要从源数据库中挑选出数据仓库所需要的数据，将这些数据按照标准进行统一，确保命名约定。编码结构。属性度量的一致性，然后在将原始数据结构做一个从面向应用向面向主题的转变。

3. 非易失性

在数据仓库中，数据是从事务操作型数据中抽取出来，反映一段相当长时间内的历史数据，是不同时间点的数据库快照的集合，以及基于快照的统计、综合和重组。数据仓库中的数据主要提供企业决策分析之用，所涉及的数据操作主要是数据查询，一旦数据进入数据仓库，只要数据没有超过数据仓库的数据存储期限，一般不对数据进行更新操作，只进行查询。

4. 时变性

数据仓库中的数据不可更新是针对应用来说的，也即数据仓库的用户在进行分析处理时是不进行数据更新操作的。但是，数据仓库的数据是随时间的变化而不断变似’这一特征表现在以下三个方面：

（1）数据仓库随时间变化不断增加新的数据内容。

（2）数据仓库随时间变化不断删除旧的数据内容。

（3）数据仓库中包含大量的综合数据，这些综合数据中很多是与时间有关，并要随时间的变化不断地进行重新综合。

5．集合性

数据仓库的集合性意味着数据仓库必须以某种数据集合的形式存储起来。目前数据仓库所采用的数据集合方式是以多维数据库方式进行存储的多维模式，以关系数据库方式进行存储的关系模式或是以两者结合的方式进行存储的混合模式。

概言之，数据仓库是一种语义上一致的数据集合，它是决策支持数据模型的物理实现，并存放企业战略决策所需信息。它也常常被看作决策支持系统的一种体系结构，通过将异种数据源中的数据集成在一起，支持结构化的和专门的查询与分析，支持决策过程。

2.1.3 数据仓库的体系结构

数据仓库系统体系结构[27-29]如图 2-1 所示。

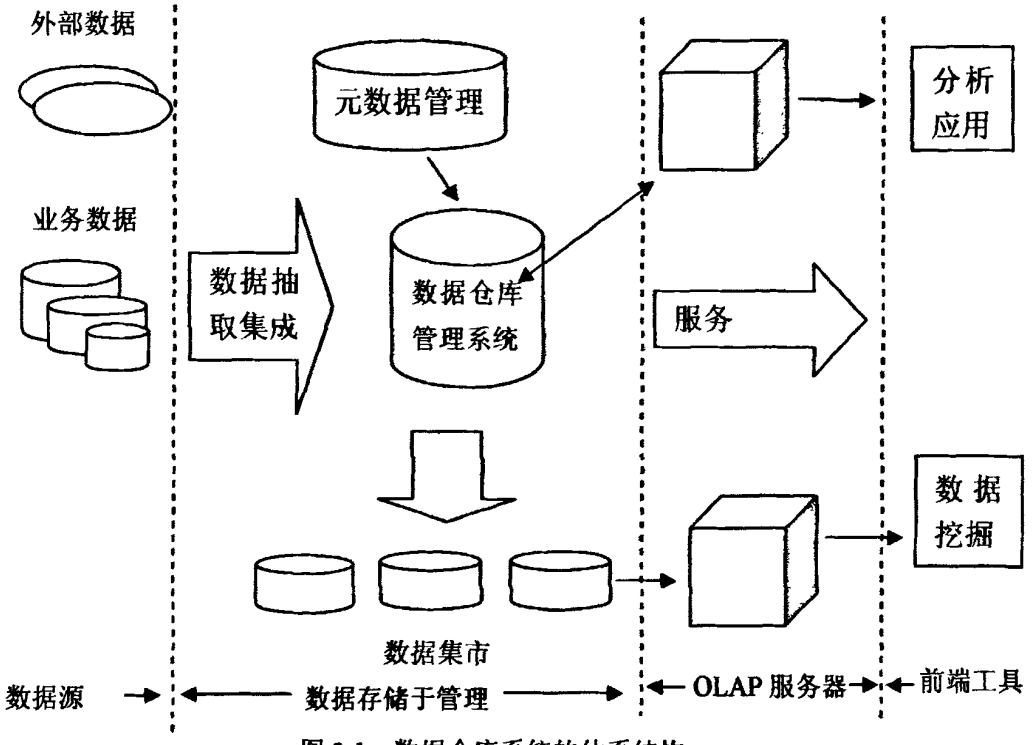


图 2-1 数据仓库系统的体系结构

1．中心数据仓库

是整个数据仓库系统的核心，是数据存放的地方。其突出的特点是对海量数据的支持和快速的检索技术。

2．数据抽取工具

把数据从各种各样的数据源，进行必要的转化、整理，再存放到数据仓库内。对各种不同数据存储方式的访问能力是数据抽取工具[30]的关键，应能生成 COBOL 程序、MVS 作业控制语言（JCL）、UNIX 脚本和 SQL 语句等，以访问不同的数据。

3．元数据

元数据[31]是描述数据仓库内数据的结构和建立方法的数据。可将其按用途的不同分为两类，技术元数据和商业元数据。

技术元数据是数据仓库的设计和管理人员用于开发和日常管理数据仓库是用的数据。包括：数据源信息；数据转换的描述；数据仓库内对象和数据结构的定义；数据清理和数据更新时用的规则；源数据到目的数据的映射；用户访问权限，数据备份历史记录，数据导入历史记录，信息发布历史记录等。

商业元数据从商业业务的角度描述了数据仓库中的数据。包括：业务主题的描述，包含的数据、查询、报表。

元数据为访问数据仓库提供了一个信息目录（information directory），这个目录全面描述了数据仓库中都有什么数据、这些数据怎么得到的、和怎么访问这些数据。是数据仓库运行和维护的中心，数据仓库服务器利用他来存贮和更新数据，用户通过他来了解和访问数据。

4．数据仓库分析工具

数据仓库分析工具[32-34]是为用户分析数据仓库中数据提供手段。主要有数据查询和报表工具、应用开发工具、管理信息系统（EIS）工具、在线分析（OLAP）工具、数据挖掘工具等。

5．数据集市（DataMarts）

为了特定的应用目的或应用范围，而从数据仓库中独立出来的一部分数据，也可称为部门数据或主题数据（subject area）[35]。在数据仓库的实施过程中往往可以从一个部门的数据集市着手，以后再用几个数据集市组成一个完整的数据仓库。需要注意的就是再实施不同的数据集市时，同一含义的字段定义一定要相容，这样再以后实施数据仓库时才不会造成大麻烦。

6．数据仓库管理

数据仓库管理主要包括安全和特权管理，跟踪数据的更新，数据质量检查，

管理和更新元数据，审计和报告数据仓库的使用和状态，删除数据，复制、分割和分发数据，备份和恢复，存储管理等功能。

7. 信息发布系统

把数据仓库中的数据或其他相关的数据发送给不同的地点或用户。

2.1.4 数据仓库的建立步骤

数据仓库的设计是一个循环反复的过程，大体上可以分为以下几个步骤：

1. 概念模型设计

A. 界定系统边界

虽然无法在数据仓库设计的初期就得到详细而明确的需求，但有些方向性的需求需要解决，比如要做的决策类型有哪些，决策者感兴趣的是什么问题，这些问题需要什么样的信息，要得到这些信息需要包含哪些数据源

B. 确定主要的主题域及其内容

要确定系统所包含的主题，即数据仓库的分析对象，然后对每个主题的内容进行较明确的描述，包括：确定主题及其属性信息，确定主题的公共码键，主题间联系及其属性等等。

C. OLAP 设计

根据用户的分析处理要求，设计系统所采用的 OLAP 数据模型，如：星型模型、雪花模型、数据立方体[36]等。

2. 逻辑模型设计

本阶段的任务主要是对每个当前要装载的主题的逻辑实现进行定义，并将相关内容记录在数据仓库的元数据中。

由于目前的数据仓库系统的实现一般采用关系数据库[37]系统，所以数据仓库

的逻辑设计就是将在概念设计阶段得到的E-R图转换成关系模式。

3．物理模型设计

该阶段的任务是确定数据仓库中数据的存储结构，确定索引策略，确定数据存放位置，确定存储分配。

4．数据仓库生成

根据数据仓库元数据中的定义信息，利用相关的数据抽取工具收取生成数据仓库中的数据，并将其加载到数据仓库中去；统计生成OLAP数据。在这个阶段，可能也需要设计和编制一些数据抽取程序。

这一步的工作成果是：数据已经装载到数据仓库中，可以在其上建立数据仓库的应用，如OLAP分析处理、数据挖掘、DSS应用等。

5．数据仓库运行与维护

这个阶段的任务是建立数据仓库的应用，并在应用过程中理解需求，改善和完善系统，维护数据仓库中的数据。

由于数据仓库主题的不稳定性，因此数据仓库系统的建立与使用有一个稳定的过程，在应用过程中根据用户的反馈信息来修改与完善数据仓库的需求。

在系统的运行过程中，随着数据源中数据的不断变化，需要通过数据刷新操作来维护数据仓库中数据的一致性，即重新生成数据仓库中的数据。

2.2 数据挖掘技术

2.2.1 数据挖掘的定义

数据挖掘(DataMining)[38]，也叫数据开采，数据采掘等，就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。现存的信息系统的数据量非

常大，而其中真正有价值的信息却很少，因此从大量的数据中经过深层分析，获得有利于业务运作、提高竞争力的信息。这种新式的信息处理技术，可以按既定业务目标，对大量的数据进行探索和分析，揭示隐藏的、未知的或验证已知的规律性，并进一步将其模型化。

在较浅的层次上，它利用现有数据库管理系统的查询、检索及报表功能，与多维分析、统计分析方法相结合，进行联机分析处理(OLAP)，从而得出可供决策参考的统计分析数据。在深层次上，则从数据库中发现前所未有的、隐含的知识。OLAP的出现早于数据挖掘，它们都是从数据库中抽取有用信息的方法，就决策支持的需要而言两者是相辅相成的。

数据挖掘基于的数据库类型主要有：关系型数据库、面向对象数据库、事务数据库、演绎数据库、多媒体数据库、主动数据库、空间数据库、异质数据库、文本型、Internet信息库以及新兴的数据仓库(DataWarehause)等。而挖掘后获得的知识包括关联规则、特征规则、区分规则、分类规则、总结规则、偏差规则、聚类规则、模式分析及趋势分析等。

数据挖掘是一门交叉学科，它把人们对数据的应用从低层次的简单查询，提升到从数据中挖掘知识，提供决策支持。随着DMKD研究逐步走向深入，数据挖掘和知识发现的研究已经形成了三根强大的技术支柱：数据库、人工智能和数理统计。

2.2.2 数据挖掘与传统分析方法的区别

数据挖掘与传统的数据分析(如查询、报表、联机应用分析OLAP)的本质区别是数据挖掘是在没有明确假设的前提下去挖掘信息、发现知识。数据挖掘所得到的信息应具有先前未知，有效和可实用三个特征。先前未知的信息是指该信息是

预先未曾预料到的，即数据挖掘是要发现那些不能靠直觉发现的信息或知识，甚至是违背直觉的信息或知识，挖掘出的信息越是出乎意料，就可能越有价值。最典型的案例就是通过数据挖掘发现了小孩尿布和啤酒之间有着惊人的联系。

数据挖掘和OLAP是完全不同的工具，所基于的技术也大相径庭。OLAP是决策支持领域的一部分。传统的查询和报表工具是告诉你数据库中都有什么(Whathappened)，OLAP则更进一步告诉你下一步会怎么样(Whatnext)、和如果我采取这样的措施又会怎么样(Whatif)。用户首先建立一个假设，然后用OLAP检索数据库来验证这个假设是否正确，数据挖掘与OLAP不同的地方是，数据挖掘不是用于验证某个假定的模式(模型)的正确性，而是在数据库中自己寻找模型。它在本质上是一个归纳的过程。

数据挖掘和OLAP具有一定的互补性。在利用数据挖掘出来的结论采取行动之前，你也许要验证一下如果采取这样的行动会给公司带来什么样的影响，那么OLAP工具能回答你的这些问题。而且在知识发现的早期阶段，OLAP工具还有其他一些用途。可以帮你探索数据，找到哪些是对一个问题比较重要的变量，发现异常数据和互相影响的变量。这都能帮你更好的理解你的数据，加快知识发现的过程。

2.2.3 数据挖掘的特点

数据挖掘技术具有以下特点：

1．处理的数据规模十分庞大，达到GB、TB数量级，甚至更大。

2．查询一般是决策制定者(用户)提出的即时随机查询，往往不能形成精确的查询要求，需要靠系统本身寻找其可能感兴趣的东西。

3．在一些应用中，由于数据变化迅速，因此要求数据挖掘能快速做出相应反

应以随时提供决策支持。

4. 数据挖掘中，规则的发现基于统计规律。因此，所发现的规则不必适用于所有数据，而是当达到某一临界值时，即认为有效。因此，利用数据挖掘技术可能会发现大量的规则。

5. 数据挖掘所发现的规则是动态的，它只反映了当前状态的数据库具有的规则，随着不断地向数据库中加入新数据，需要随时对其进行更新。

2.2.4 描述型数据挖掘

1. 统计和可视化

要想建立一个好的预言模型，必须了解自己的数据。最基本的方法是计算各种统计

变量(平均值、方差等)和察看数据的分布情况。也可以用数据透视表察看多维数据。数据的种类可分为连续的，有一个用数字表示的值(比如销售量)或离散的，分成一个个的类别(如红、绿、蓝)。离散数据可以进一步分为可排序的，数据间可以比较大小(如，高、中、低)和标称的，不可排序(如邮政编码)：图形和可视化工具在数据准备阶段尤其重要，它能让使用者快速直观的分析数据，而不是只给出枯燥乏味的文本和数字。它不仅使用者看到整个森林，还允许使用者拉近每一棵树来察看细节。在图形模式下我们很容易找到数据中可能存在的模式、关系、异常等，直接看数字则很难。

可视化工具的问题是模型可能有很多维或变量，但是我们只能在2维的屏幕或纸上展示它。比如，我们可能要看的是信用风险与年龄、性别、婚姻状况、参加工作时间的一关系。因此，可视化工具必须用比较巧妙的方法在两维空间内展示n维空间的数据。虽然目前有了一些这样的工具，但它们都要用户“训练”过他们的眼睛后才能理解图中画的到底是什么东西。在使用这些工具时可能会遇到困难。对于眼睛有色盲或空间感不强的人，在使用这些工具时可能会遇到困难。

2. 聚类

聚类(Clustering)[39-41]是一个将数据集划分为若干组(class)或类(Cluster)的过

程，并使得同一个组内的数据对象具有较高的相似度；而不同组中的数据对象是不相的。相似或不相似的描述是基于数据描述属性的取值来确定的。通常就是利用(各对象间)距离来进行表示的。许多领域，包括数据挖掘、统计学和机器学习都有聚类研究和应用。

目前在文献中存在大量的聚类算法。算法的选择取决于数据的类型，聚类的目的和应用。如果聚类分析被用作描述或探查的工具，可以对同样的数据尝试多种算法，以发现数据可能揭示的结果。

大体上，主要的聚类算法可以划分为如下几类：

划分方法(Partitioning Methods)：给定一个n个对象或元组的数据库，一个划分方法构建数据的k个划分，每个划分表示一个聚类，并且k<=n。也就是说，它将数据划分为k个组，同时满足如下的要求；(1)每个组至少包含一个对象；(2)每个对象必须属于且只属于一个组。注意在某些模糊划分技术中第二个要求可以放宽。在参考文献中列出了对于该类技术的参照。

层次的方法(Hierarchical Methods)：层次的方法对给定数据集合进行层次的分解。根据层次的分解如何形成，层次的方法可以被分为凝聚的或分裂的方法。凝聚的方法，也称为自底向上的方法，一开始将每个对象作为单独的一个组，然后继续地合并相近的对象或组，直到所有的组合并为一个(层次的最上层)，或者达到一个终止条件。分裂的方法，也称为自顶向下的方法，一开始将所有的对象置于一个簇中。在迭代的每一步中，一个簇被分裂为更小的簇，直到最终每个对象在单独的一个簇中，或者达到一个终止条件。

基于密度的方法：绝大多数划分方法基于对象之间的距离进行聚类。这样的方法只能发现球状的簇，而在发现任意形状的簇上遇到了困难。随之提出了基于密度的另一类聚类方法，其主要思想是：只要临近区域的密度(对象或数据点的数目)超过某个阈值，就继续聚类。也就是说，对给定类中的每个数据点，在一个给定范围的区域中必须包含至少某个数目的点。这样的方法可以用来过滤“噪音”数据，发现任意形状的簇。

基于网格的方法(Grid-based Methods)：基于网格的方法把对象空间量化为有限数目的单元，形成了一个网格结构。所有的聚类操作都在这个网格结构(即量化的空间)上进行。这种方法的主要优点是它的处理速度很快，其处理时间独立于数据对象的数目，只与量化空间中每一维的单元数目有关。STING是基于网格方法的一个典型例子。CLIQUE和WaveCluster这两种算法既是基于网格的，又是基于密度的。

基于模型的方法(Model-based Methods)：基于模型的方法为每个簇假定了一个模型，寻找数据对给定模型的最佳匹配。一个基于模型的算法可能通过构建反映数据点空间分布的密度函数来定位聚类。它也基于标准的统计数字自动决定聚类的数目，考虑“噪音”数据和孤立点，从而产生健壮的聚类方法。

3．关联分析

关联规则[42-45]挖掘就是从大量的数据中挖掘出有价值描述数据项之间相互联系的有关知识。随着收集和存储在数据库中的数据规模越来越大，人们对从这些数据中挖掘相应的关联知识越来越有兴趣。例如：从大量的商业交易记录中发现有价值的关联知识就可帮助进行商品目录的设计、交叉营销或帮助进行其它有关的商业决策。关联分析是寻找数据库中值的相关性。两种常用的技术是关联规则和序列模式。

关联规则是寻找在同一个事件中出现的不同项的相关性，比如在一次购买活动中所买不同商品的相关性。序列模式与此类似，它找的是事件之间时间上的相关性，如对分析某一知识点掌握成绩与学生测试成绩的关联关系。

关联规则可记为A=>B，A称为前提和左部(LHS)，B称为后续或右部(RHS)。如关联规则“上课讲话的人同一天肯定会打瞌睡”，左部是“上课讲话”，右部是“打瞌睡”。有些软件产品用图形的方式显示项之间的相关性。

2.2.5 序列模式挖掘

序列模式挖掘(Sequence Pattern Mining)是指挖掘频繁出现的有序事件或子序列。一个序列模式的例子比如是“9个月以前购买奔腾PC的客户很可能在一个

月内订购新的CPU芯片”。由于很多商业交易、电传记录、天气数据和生产过程都是时间序列数据，在针对目标市场、客户吸引、气象预报等的数据分析中，序列模式挖掘是很有用途的。

1．序列模式挖掘的情形和参数

许多有关序列模式挖掘的研究主要针对符号模式(Symbolic Pattern)，因为数字曲线模式通常属于统计时序分析中的趋势分析和预测范畴。

对序列模式挖掘，存在一些参数，其取值如何，将严重影响挖掘结果。

第一个参数是时间序列的持续时间(Duration)T。持续时间可以是数据库中的整个序列，或由用户选择的一个子序列，如对应于1999年的子序列。序列模式挖掘因此是限制在特定的持续时间内的挖掘。持续时间还可定义为一组分割的序列，如每年，或股票暴跌后的每周，或火山喷发前后的每两周等。在这些情形中，可以发现周期模式(Periodic Pattern)。

第二个参数是事件重叠窗口(Event Folding Window)w。在指定时间周期内出现的一组事件，可以视为某一分析中一起出现的事件。若w设为与持续时间T相同的值，则找出的是与时间无关的模式——即是一些基本的相关模式，如“在1999，购买PC的顾客也购买数字相机”(这里不反映先购买哪一个)。若W取值为0(即没有事件序列折叠)，则找出的序列模式中的每个事件出现在不同的时间值，如“购买了PC的顾客，可能接着买内存芯片，再买CD—ROM”。若w设为之间的值(如同一月内发生的交易，或24小时滑动窗口内)，则考虑同一周期内出现的交易，分析中序列被折叠。第三个参数是被发现的模式中时间之间的时间间隔(Interval)int。

2．序列模式挖掘的方法

关联规则挖掘中采用的Apriori特性可以用于序列模式的挖掘，因为着长度为k的序列模式是非频繁的，其超集(长度为k+1)不可能是频繁的。因此，序列模式挖掘的大部分方法都采用了类Apriorl算法的变种，虽然所考虑的参数设置和约束都

有所不同。另一种挖掘此类模式的方法是基于数据库投影的序列模式生长(Databaseproject—based Sequential Pattern Growth)技术，类似用于无候选生成的频繁模式挖掘(Frequent Pattern)的，频繁模式增长(FP—growth)法。

2.2.6 数据挖掘算法与选择

针对每一种特定的应用，建立起了相关的数据挖掘的模型后，会有多种算法可供选择。大多数数据挖掘使用的算法都是在计算机科学或统计数学杂志上技表过的成熟算法，所不同的只是算法的实现和对性能的优化。

几乎所有的数据挖掘技术都可称为是数据驱动的，而不是用户驱动的．也就是说用户在使用这些算法时，只要给出数据，不用告诉算法程序怎么做和期待得到什么结果，一切都是算法自身从给定的数据中自己找出来。应注意的是大部分算法都不是专为解决某个问题而特制的，算法之间也并不互相排斥。不能说一个问题一定要采用某种算法，别的就不行。一般来说并不存在所谓的最好的算法，在最终决定选取那种模型或算法之前，可能各种模型都试一下，然后再选取一个较好的。

2.2.7 数据挖掘的步骤

对数据挖掘过程模型的研究很多，根据这些过程模型，设计和实现了许多相应的数据挖掘原型系统和商业系统。大致可以将数据挖掘模型划分为两种类型，一种是Fayyad总结出的过程模型，如图2．4。另一种是遵循CRISP-DM标准的过程模型，如图2．5。

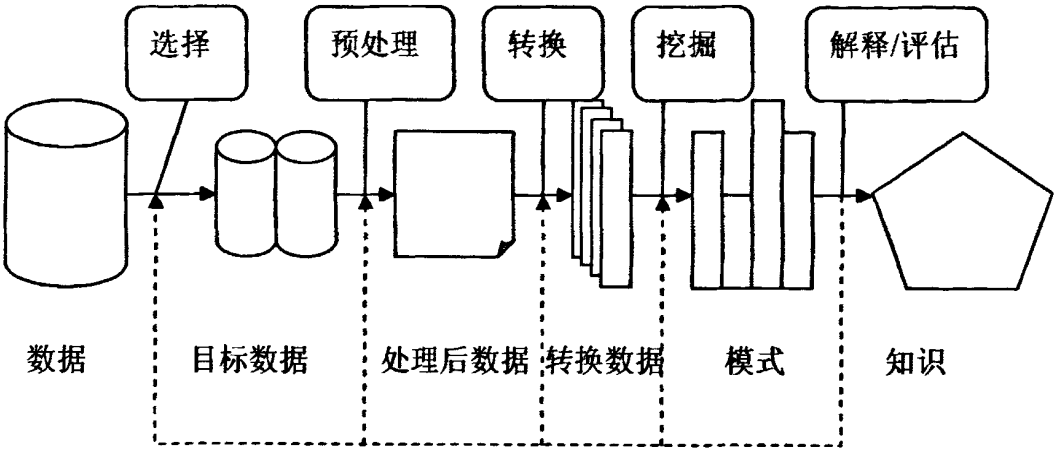


图 2.4 Fayyad的过程模型

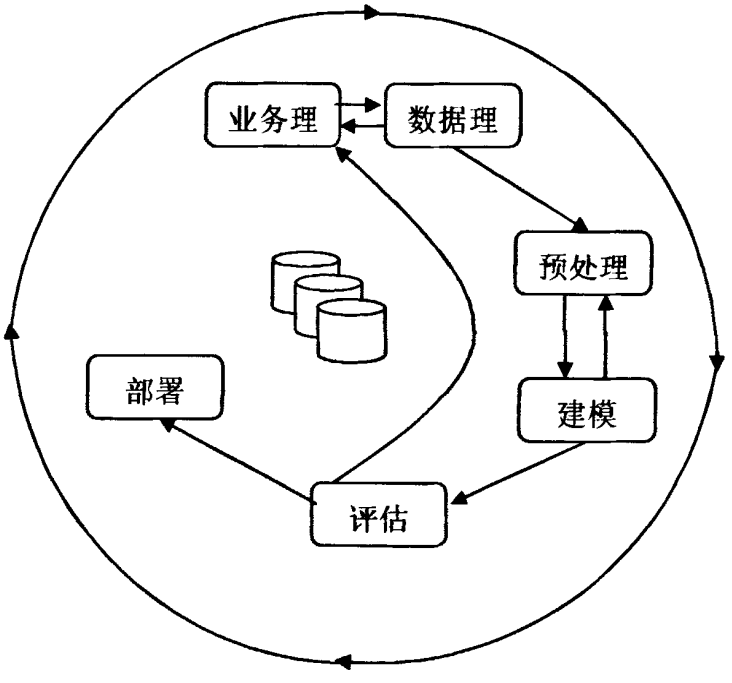


图 2.5 CRISP-DM的过程模型

数据挖掘的过程理论上包括以下五个阶段

1．确定挖掘目标：清晰地定义出业务问题，认清数据挖掘的目的是数据挖掘的重要一步。

2．数据准备：包括数据的选择和数据的预处理。在确定数据挖掘的业务对象

后，需要搜索所有与业务对象有关的内部和外部数据，从中选择出适合数据挖掘应用的数据。选定数据后，还需要对数据进行预处理，对数据清洗，解决数据中的缺值、冗余、数据值不一致、数据定义不一致 过时的数据等问题。

3．数据挖掘：这个阶段就是应用合适的数据挖掘技术对对经过转换清理的数据进行挖掘和知识发现的过程。

4．结果分析：结果分析就是由分析人员或通过算法根据发现知识的领域重要性、可信度和支持度等阀值来对发现结果进行评价，并以用户能理解和观察的方式将发现的知识呈现给用户。

5．知识的应用：将分析所得到的知识集成到业务信息系统的组织结构中去。

数据挖掘的过程如图 2.6所示。

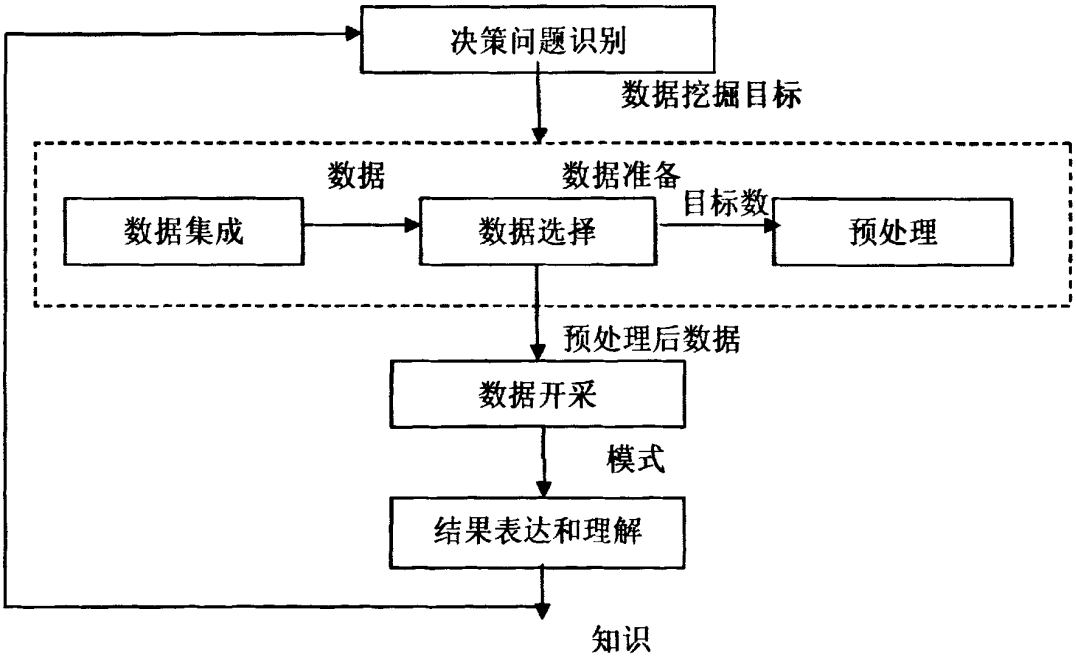


图 2.6 数据挖掘过程

2.2.8 数据挖掘的方法

1．神经网络方法

神经网络由于本身良好的鲁棒性、自组织自适应性、并行处理、分布存储和高度容错等特性非常适合解决数据挖掘的问题，因此近年来越来越受到人们的关注。典型的神经网络模型主要分 3 大类：以感知机、bp 反向传播模型、函数型网络为代表的，用于分类、预测和模式识别的前馈式神经网络模型；以 hopfield 的离散模型和连续模型为代表的，分别用于联想记忆和优化计算的反馈式神经网络模型；以 art 模型、koholon 模型为代表的，用于聚类的自组织映射方法。神经网络方法的缺点是"黑箱"性，人们难以理解网络的学习和决策过程。

2．遗传算法

遗传算法是一种基于生物自然选择与遗传机理的随机搜索算法，是一种仿生全局优化方法。遗传算法具有的隐含并行性、易于和其它模型结合等性质使得它在数据挖掘中被加以应用。

sunil 已成功地开发了一个基于遗传算法的数据挖掘工具，利用该工具对两个飞机失事的真实数据库进行了数据挖掘实验，结果表明遗传算法是进行数据挖掘的有效方法之一[4]。遗传算法的应用还体现在与神经网络、粗集等技术的结合上。如利用遗传算法优化神经网络结构，在不增加错误率的前提下，删除多余的连接和隐层单元；用遗传算法和 bp 算法结合训练神经网络，然后从网络提取规则等。但遗传算法的算法较复杂，收敛于局部极小的较早收敛问题尚未解决。

3．决策树方法

决策树是一种常用于预测模型的算法，它通过将大量数据有目的分类，从中找到一些有价值的，潜在的信息。它的主要优点是描述简单，分类速度快，特别适合大规模的数据处理。最有影响和最早的决策树方法是由 quinlan 提出的著名的基于信息熵的 id3 算法。它的主要问题是：id3 是非递增学习算法；id3 决策树是单变量决策树，复杂概念的表达困难；同性间的相互关系强调不够；抗噪性差。针对上述问题，出现了许多较好的改进算法，如 schlimmer 和 fisher 设计了 id4 递增式学习算法;钟鸣，陈文伟等提出了 ible 算法等。

4．粗集方法

粗集理论是一种研究不精确、不确定知识的数学工具。粗集方法有几个优点：不需要给出额外信息；简化输入信息的表达空间；算法简单，易于操作。粗集处理的对象是类似二维关系表的信息表。目前成熟的关系数据库管理系统和新发展起来的数据仓库管理系统，为粗集的数据挖掘奠定了坚实的基础。但粗集的数学基础是集合论，难以直接处理连续的属性。而现实信息表中连续属性是普遍存在的。因此连续属性的离散化是制约粗集理论实用化的难点。现在国际上已经研制出来了一些基于粗集的工具应用软件，如加拿大 regina 大学开发的 kdd-r;美国 kansas 大学开发的 lers 等。

5．覆盖正例排斥反例方法

它是利用覆盖所有正例、排斥所有反例的思想来寻找规则。首先在正例集合中任选一个种子，到反例集合中逐个比较。与字段取值构成的选择子相容则舍去，相反则保留。按此思想循环所有正例种子，将得到正例的规则(选择子的合取式)。比较典型的算法有 michalski 的 aq11 方法、洪家荣改进的 aq15 方法以及他的 ae5 方法。

6．统计分析方法

在数据库字段项之间存在两种关系：函数关系(能用函数公式表示的确定性关系)和相关关系(不能用函数公式表示，但仍是相关确定性关系)，对它们的分析可采用统计学方法，即利用统计学原理对数据库中的信息进行分析。可进行常用统计(求大量数据中的最大值、最小值、总和、平均值等)、回归分析(用回归方程来表示变量间的数量关系)、相关分析(用相关系数来度量变量间的相关程度)、差异分析(从样本统计量的值得出差异来确定总体参数之间是否存在差异)等。

7. 模糊集方法

即利用模糊集合理论对实际问题进行模糊评判、模糊决策、模糊模式识别和模糊聚类分析。系统的复杂性越高，模糊性越强，一般模糊集合理论是用隶属度来刻画模糊事物的亦此亦彼性的。李德毅等人在传统模糊理论和概率统计的基础上，提出了定性定量不确定性转换模型--云模型，并形成了云理论。

第三章 固定资产管理系统的需求分析与总体框架

3.1 固定资产管理系统的需求分析

无锡旅游商贸高等职业技术学校是国家级重点职业学校，2003 年由原无锡旅游职中和商业职中合并，后又合并原无锡房管中专与建工中专，学生办学规模不断扩大，招生人数也在不断增加，教学、科研设备的投入大幅度提高。学校建立了完善的校园网络，连接了办公区、教学区、实训楼、食堂、宿舍等区域，网络遍布校园各个角落，同时成立了信息系、经贸系、旅游系、房管系四大系部，学生人数近 5000 人，实验室设备由原来的 1 千万增加到近 3 千万。为满足教学、科研的需求、提高工作效率，迫切需要一个建立在校园网环境下的、功能齐全的固定资产管理系统。

系统运行在校园网环境中，用户可以通过校园内网的任意一台Pc登陆系统：能够实现远程计划申报、远程查询(部分数据)；增加仓库管理功能，入、出库管理、库存设备查询、零库存历史记录查询等；提供对内资料文档的下载服务；对外采购信息及报废设备处理信息的发布：要求对数据的安全有相应的保护措施；对不同的使用者，设置不同的操作权限，如仓库管理人员对库存管理系统只具有录入、修改等权限，其他授权用户只有浏览权；整个界面友好，操作简单方便，易学易懂。

设备管理需要如下图3-1。

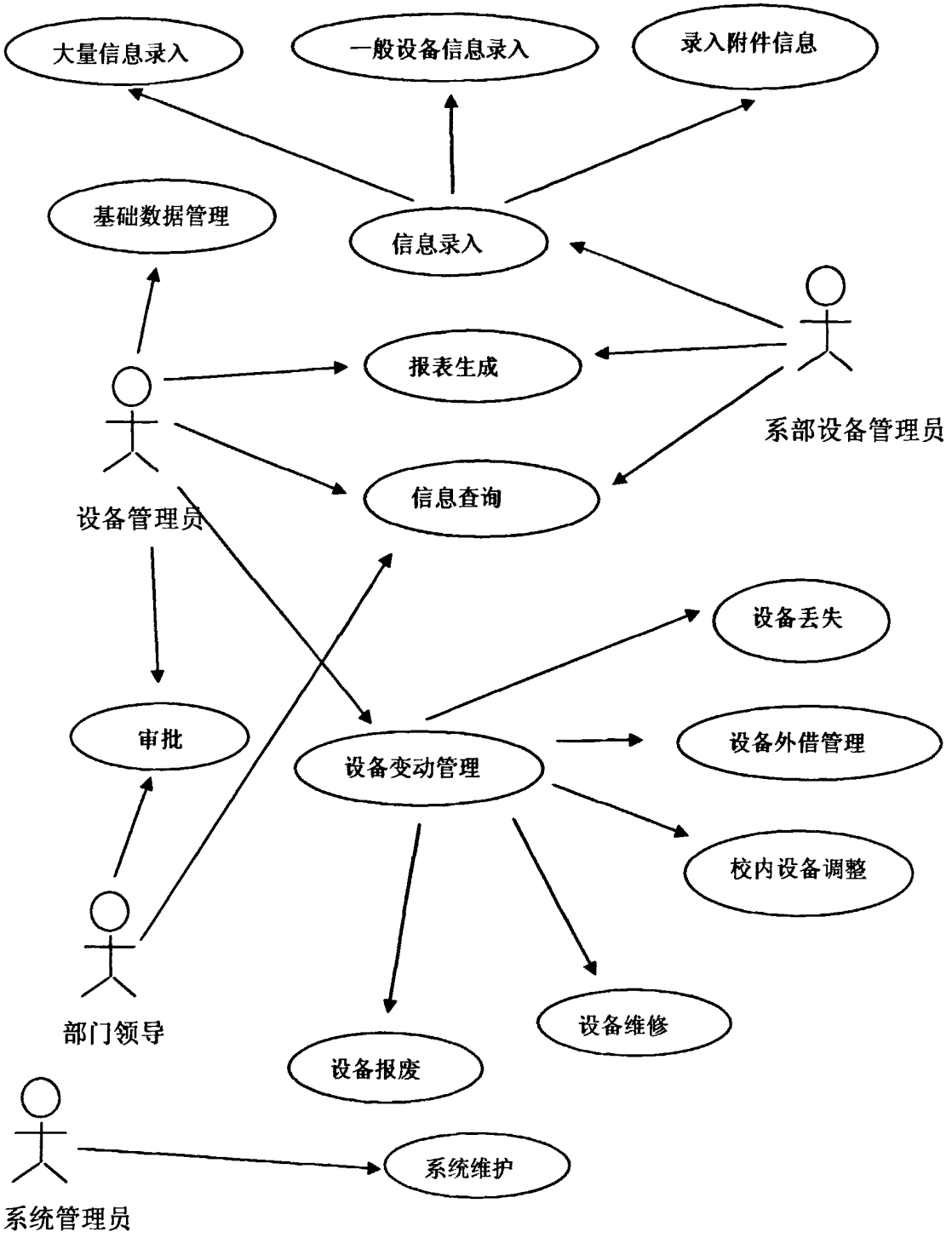


图3-1 设备管理需求图

设备管理的主要流程如下：

本系统分为资料维护、查询审计、申报审批、账务统计和系统管理五大模块

1. 资料维护：对系统的基础数据进行维护
2. 查询审计：对系统的所有人员信息进行查询；按申请生成相应的报表。增加审计功能预防重复申请。
3. 申报审批：包含所有设备的申请的申报、复核、审批、签收流程。
4. 账务统计：统计收入、支出，资金账务类报表。
5. 系统管理：管理所有系统使用人信息，并对其进行系统使用权限设定。

3.2 系统的总体框架设计

系统框架设计采用C/S结构，即客户机与服务器模式。整个系统的用户覆盖全校任何处室部门。在这种模式下，用户只要身处校园网络的覆盖范围，都可以通过浏览器来进行设备的管理。简化了系统对客户端的硬件要求，同时也简化了系统维护和升级的成本和工作量。

系统整体框架如下图3-2。

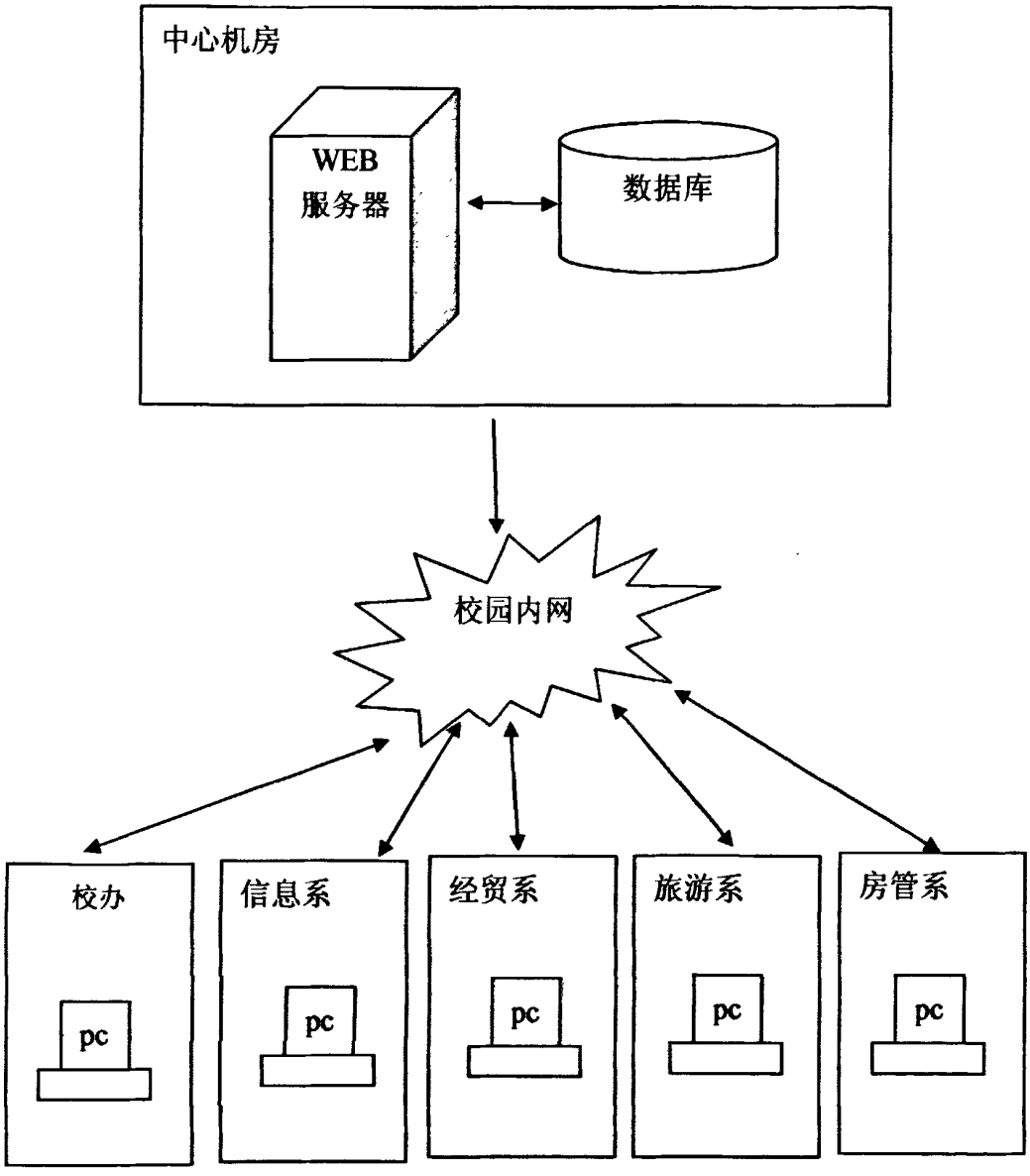


图3-2 系统整体框架图

3.3 系统运行环境

高校资产管理系统部署在LINUX(操作系统)+ORACEL(数据库)应用平台上。LINUX操作系统具有高稳定性和安全性的特点，并拥有十分强大的网络管理功能，

适合用来做服务器。数据库采用ORACLE数据库，它适合企业级数据管理。

3.4 主要设计流程

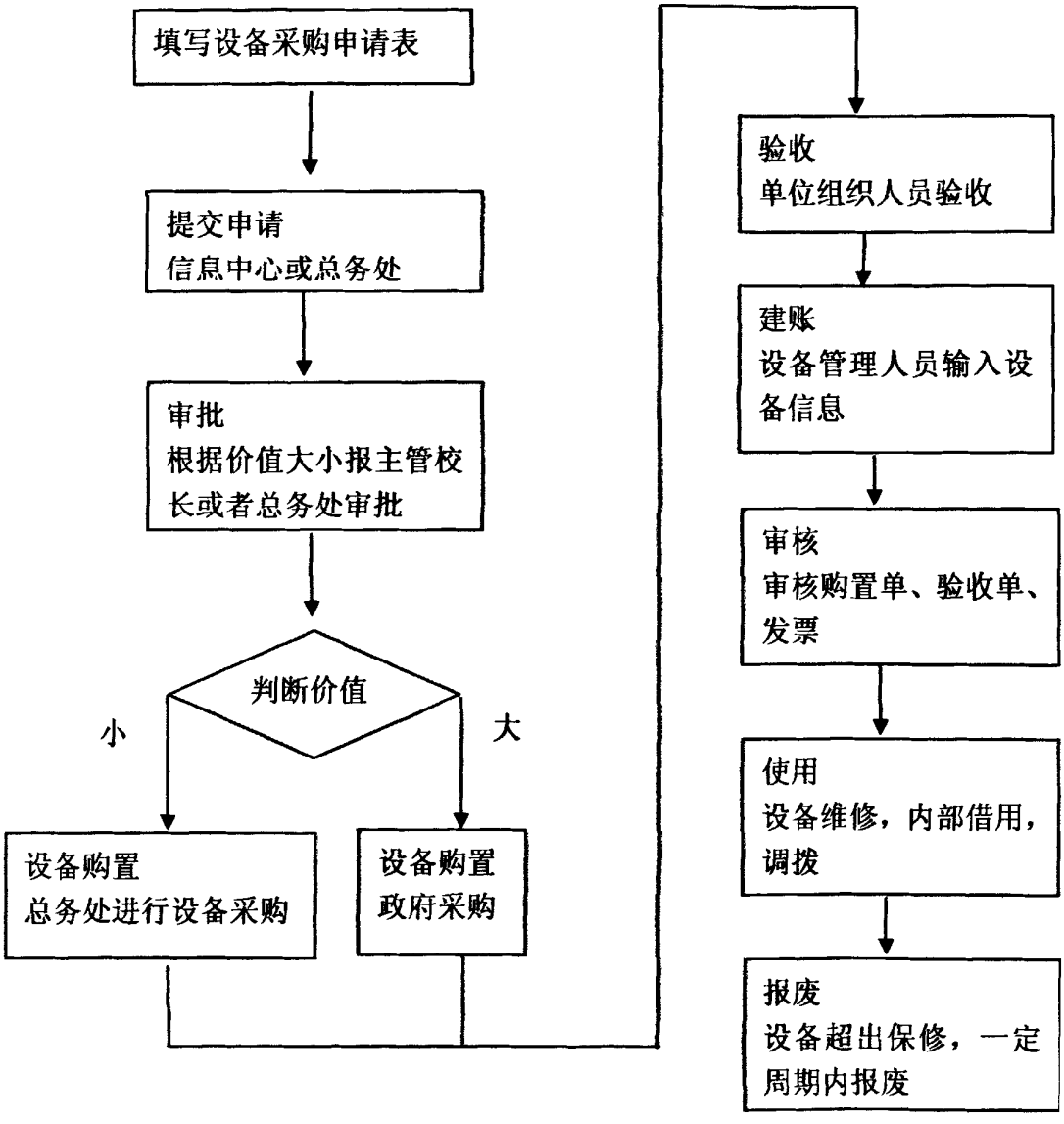


图3-3 主要设计流程图

第四章 固定资产管理系统的设计与实现

4.1 固定资产管理系统的功能结构

该系统功能主要包括数据管理、设备的采购管理、申报审批管理、验收管理、设备变动管理、低值耐用品管理等，每一部分具相对独立和完整的功能。

4.1.1 数据管理

数据管理主要包含用户管理、报表统计和系统维护3个功能模块。报表统计实现对维修、调拨、报废数据定期汇总，生成相应的统计报表。用户管理主要负责用户注册，用户注销、用户口令的修改、用户分配权限。系统数据维护完成使用单位、使用专业方向、设备分类等数据的维护。功能如下图4-1

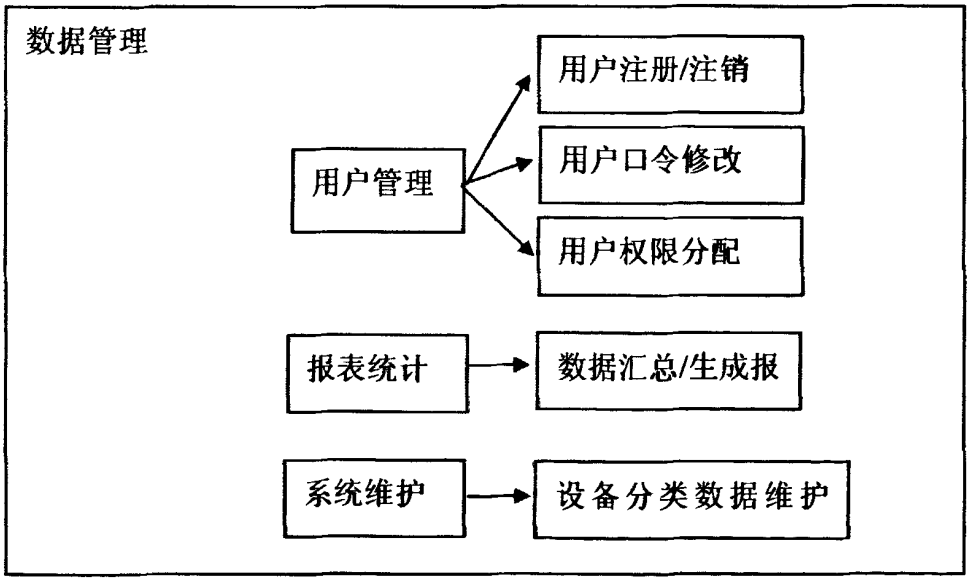


图4-1 数据管理图

4.1.2 设备的采购管理

设备采购管理是指申请系部上报设备采购计划，由相关审批领导审核签字，然后总务处汇总整理数据，并根据设备价格决定招标购买或直接自行购买。

业务流程图如图4-2所示。

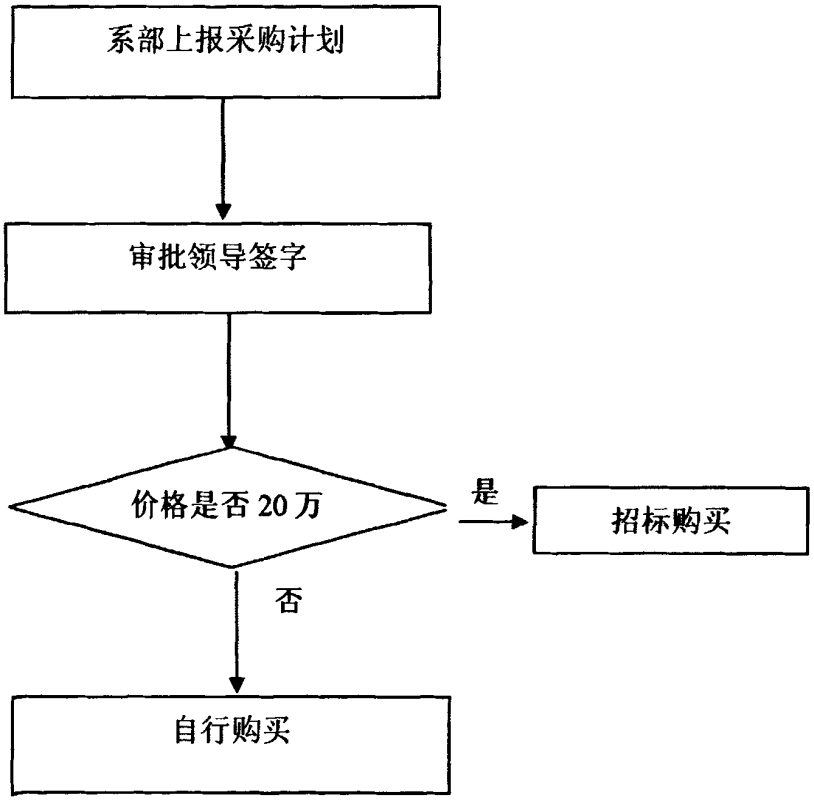


图4-2 设备采购流程图

4.1.3 设备验收

设备验收是指对单位购买的仪器设备进行审核、验收、建账和报销。

其业务流程如下图所示。

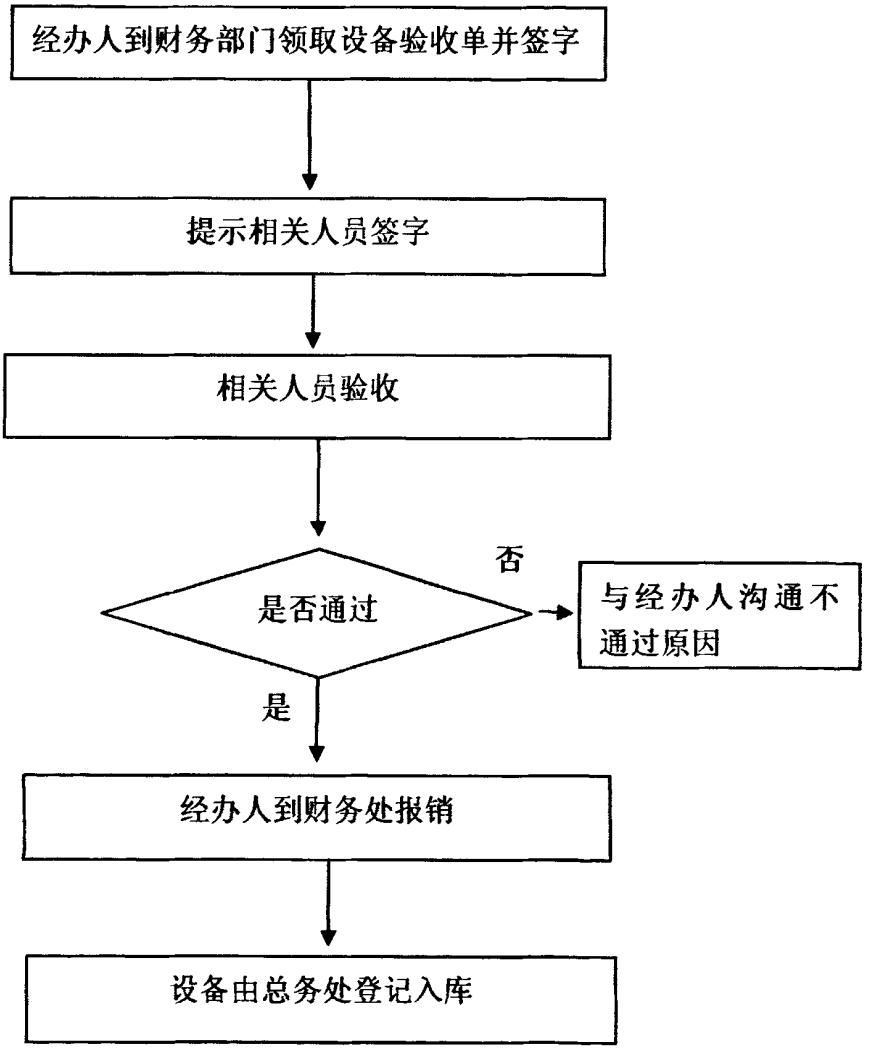


图4-3 设备验收流程图

4.1.4 设备变动管理

设备的变动管理主要包括设备的借出/归还、调拨/调剂、维修/报损、报废等

1. 设备的借出与归还

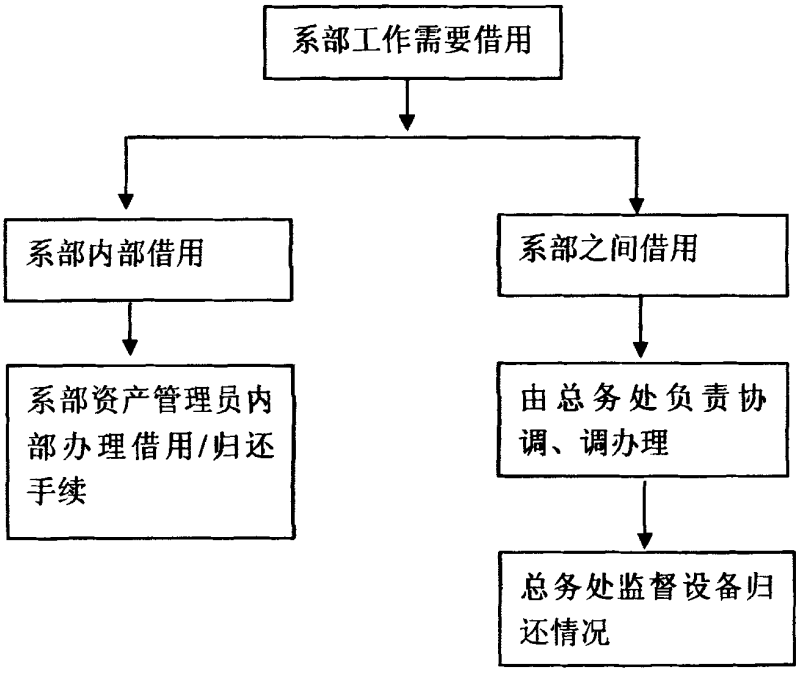


图 4-4 设备借出/归还流程图

4.1.5 设备调剂/调拨

设备的调剂调拨是指校属行政、教学、科研设备等校内使用单位发生变更或调至校外。

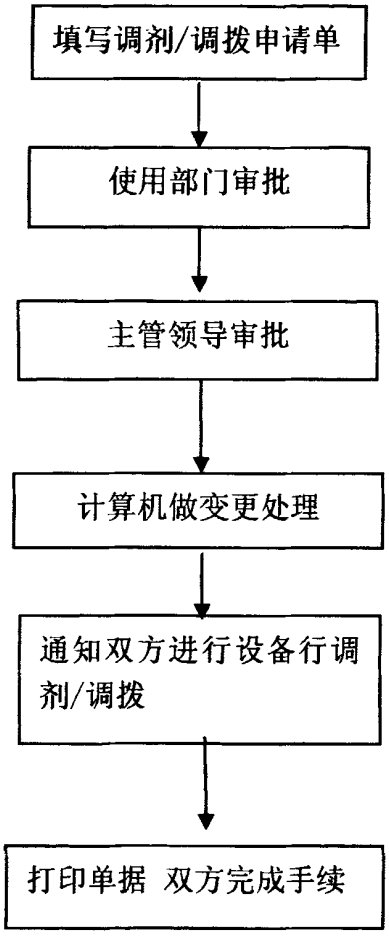


图4-5 设备调剂/调拨流程图

4.1.6 设备维修

固定资产的维修需要填写维修单。固定资产管理检索到需要维修的设备，将该设备的使用状况修改为维修，撤销单据时，再将该设备的使用状况修改为正常。

1．一般设备维修

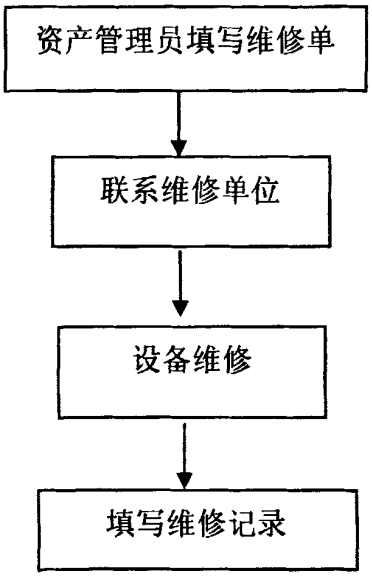


图4-6 设备维修流程图

2. 贵重设备维修

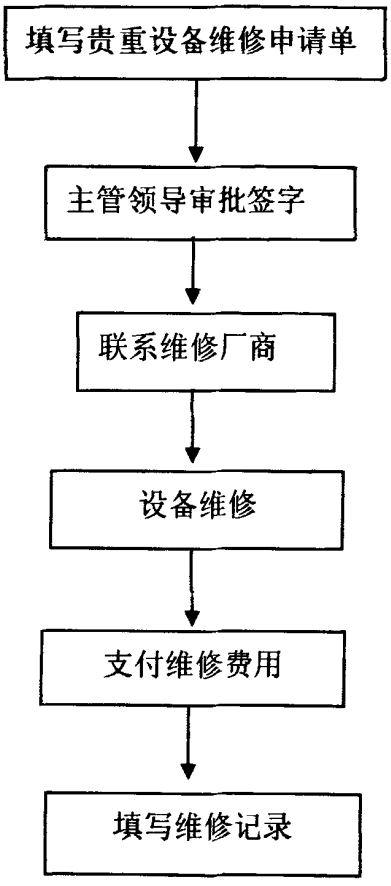


图4-7 贵重设备维修流程图

4.1.7 设备报损

设备管理员提出设备报损申请，由分管校长审批，资产管理处审批后打印报损清单。

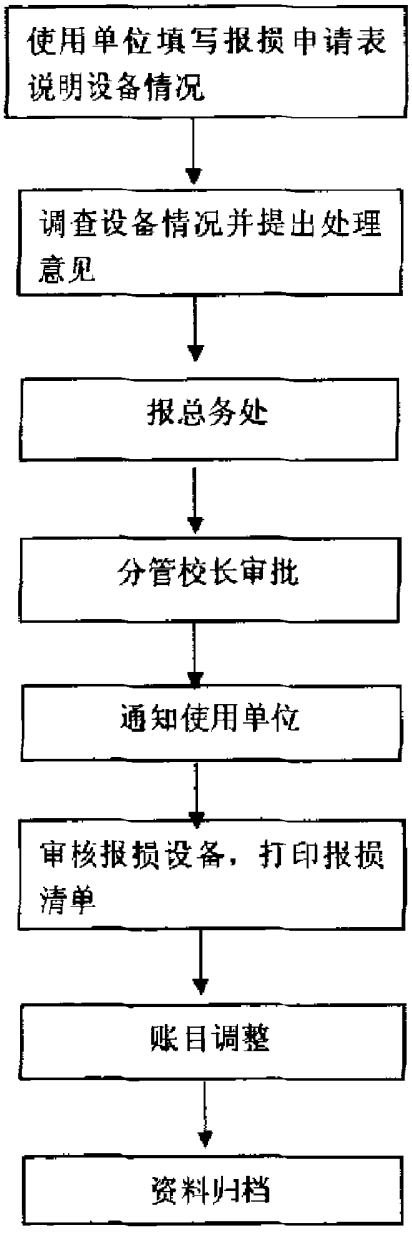


图4-8设备报损流程图

2. 低值耐用设备管理

低值耐用设备指教学、科研使用的单价在200-800之间的仪器设备和行政、后勤使用的单价在200-500之间的仪器设备。其设备的验收、调拨、报损与仪器设备管理的过程基本相同，由于其价值较低，故进行各项处理时无需校领导的审核签字，由总务处负责人直接管理。

4.2 资产数据库设计

主机表 4-1。

order	name	data_type	length	numeric_precision	numeric_scale	is_nullable	ispk	Isldently	Default	desc
1	Id(标识)	bigint		19	0	NO	NO	YES		
2	Lydwh(领用单位号)	char	10			NO	NO	NO		
3	Yqbh(仪器编号)	char	8			NO	YES	NO		
4	Flh(分类号)	char	8			NO	NO	NO		
5	Ymch(分类名称)	varchar	30			NO	NO	NO		
6	Xh(型号)	varchar	30			NO	NO	NO		
7	Gg(规格)	varchar	40			NO	NO	NO		
8	Dj(单价)	money		19	4	NO	NO	NO		
9	Code(国别号)	char	3			NO	NO	NO		
10	pCountry(国别)	varchar	10			YES	NO	NO		
11	Chj(厂家)	varchar	30			NO	NO	NO		
12	Chuchh(出厂号)	varchar	20			YES	NO	NO		
13	Chcrq(出厂日期)	datetime		23	3	NO	NO	NO		
14	Gzhrz(购置日期)	datetime		23	3	NO	NO	NO		
15	Fjshl(附件数量)	int		10	0	YES	NO	NO		
16	Fjzj(附件总价)	money		19	4	YES	NO	NO		
17	Xzh(现状)	char	1			NO	NO	NO		
18	Gljb(管理级别)	char	1			NO	NO	NO		
19	Lyr(领用人)	varchar	10			YES	NO	NO		
20	Jfdm(经费代码)	char	1			NO	NO	NO		
21	Syfx(使用方向代码)	char	1			NO	NO	NO		
22	Jshr(经手人)	varchar	10			YES	NO	NO		
23	Bdrq(变动日期)	datetime		23	3	YES	NO	NO		
24	Shydwh(使用单位号)	char	10			YES	NO	NO		
26	Gbflh(国标分类号)	char	6			YES	NO	NO		
27	Zchlb(资产类别)	char	2			YES	NO	NO		
28	Rkshj(入库时间)	datetime		23	3	YES	NO	NO		
29	Kyh(科研号)	varchar	20			YES	NO	NO		
30	Shbh(设备号)	varchar	20			YES	NO	NO		

31	Djh(单据号)	varchar	20			YES	NO	NO		
32	Jzhr(建帐人)	varchar	10			YES	NO	NO		
33	varchar1(可选字符)	varchar	50			YES	NO	NO		
34	varchar2	varchar	50			YES	NO	NO		
35	varchar3	varchar	50			YES	NO	NO		
36	num1(可选字符)	numeric				YES	NO	NO		
37	num2	numeric				YES	NO	NO		
38	Shh(审核)	char	1			YES	NO	NO		
39	Xuhao(序号)	datetime				YES	NO	NO		
40	Bzh(标志)	char	1			YES	NO	NO		
41	Ghsh(供货商)	varchar	30			YES	NO	NO		
42	Cfdd(存放地点)	varchar	30			YES	NO	NO		
43	Jkj(进口价)	money				YES	NO	NO		
44	Dlf(代理费)	money				YES	NO	NO		
45	Befzhu(备注)	varchar	200			YES	NO	NO		
46	Xiaoqu(校区)	varchar	1			YES	NO	NO		
47	Shenhedate(审核日期)	datetime		23	3	YES	NO	NO		
48	Shendanren(审单人)	varchar	10			YES	NO	NO		
49	Shuliang(数量)	int		10	0	NO	NO	NO		
50	Jkdj(进口单价)	money		19	4	YES	NO	NO		
51	Zhongzhibianhao(终止编号	varchar	8			YES	NO	NO		

表 4-2 附件表

order	name	date type	length	number precision	numeric scale	is nullable	ispk	isidentity	default	desc
1	accessoryid(标识)	bigint		19	0	NO	NO	NO		
2	fjbh(附件编号)	char	8			NO	YES	NO		
3	yqbh(主机编号)	char	8			NO	NO	NO		
4	lydwh	char	10			NO	NO	NO		
5	flh	char	8			NO	NO	NO		
6	flmch	varchar	30			NO	NO	NO		
7	fjmch	varchar	30			NO	NO	NO		
8	xh	varchar	30			NO	NO	NO		
9	gg	varchar	50			NO	NO	NO		
10	dj	money		19	4	NO	NO	NO		
11	chj	varchar	30			NO	NO	NO		
12	gzhrq	datetime		23	3	NO	NO	NO		
13	chchrq	datetime		23	3	NO	NO	NO		
14	chchbh	varchar	20			NO	NO	NO		
15	lyr	varchar	10			NO	NO	NO		
16	jfkm	char	1			NO	NO	NO		
17	Shyfx(使用方法)	char	1			NO	NO	NO		
18	xzh	char	1			NO	NO	NO		
19	Code	char	3			NO	NO	NO		
20	dif	varchar	20			NO	NO	NO		
21	kyh	varchar	20			NO	NO	NO		
22	shydwh	char	10			NO	NO	NO		
23	jshr	varchar	10			NO	NO	NO		
24	ghsh	varchar	30			NO	NO	NO		
25	yshf	money		19	4	NO	NO	NO		
26	cfdd	varchar	30			NO	NO	NO		
27	rzhshj	datetime		23	3	NO	NO	NO		
28	jzhr	varchar	10			NO	NO	NO		
29	fjjkdj	money		19	4	NO	NO	NO		
30	beizhu	varchar	200			NO	NO	NO		
31	bzh	char	1			NO	NO	NO		
32	num1	numeric		18	0	NO	NO	NO		
33	num2	numeric	50	18	0	NO	NO	NO		
34	varchar1	varchar	50			NO	NO	NO		
35	varchar2	varchar	50			NO	NO	NO		
36	varchar3	varchar				NO	NO	NO		
37	shh	char	1			NO	NO	NO		
38	shendanren	varchar	10			NO	NO	NO		
39	shDate	datetime		23	3	NO	NO	NO		

4.3 固定资产数据仓库系统主要实现界面

本系统适合无锡旅游商贸高等职业技术学校的需求。软件提供了完善的资产档案管理，支持固定资产的增加、删除、修改等基本管理环节，同时还提供了资产的借出与归还管理、资产内部调拨管理、资产维修登记管理。

系统主要界面如下：

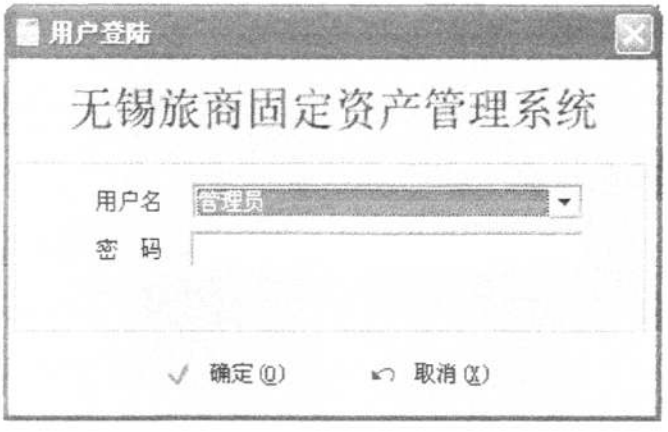


图 4-9 系统登录界面

图 4-10 系统主界面图

进入资产管理菜单下选择资产增加子菜单，弹出如下资产增加窗口：

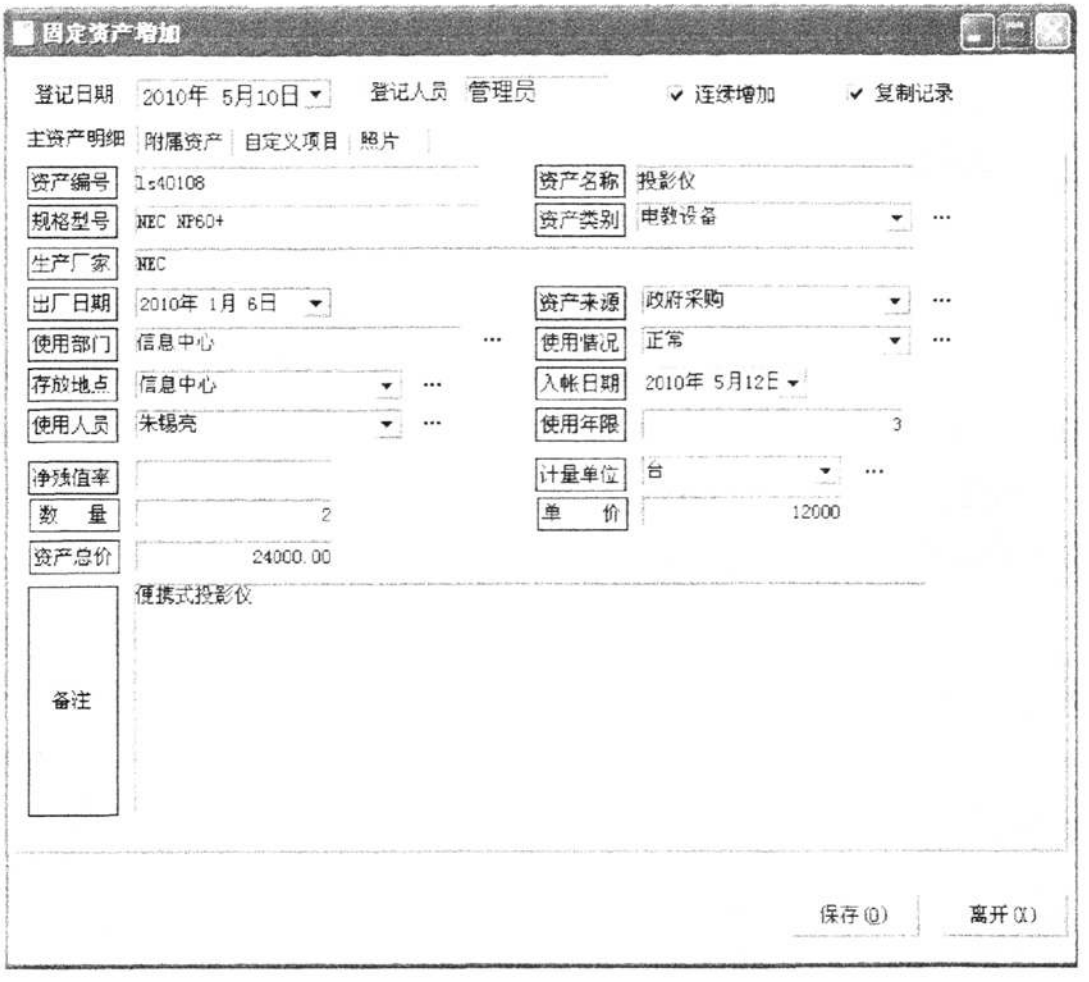


图 4-11 固定资产新增记录界面

在此窗口中，可以实现固定资产的新增，通过输入产品的编号、名称、规格型号、类别、生产厂家、使用部门等情况来新增记录。

图 4-12 固定资产借出界面

在固定资产表中选定一项资产，然后选取资产管理菜单下的资产借出，进入资产借出界面，填入借出的日期、部门、借出人、批准人 、拟归还日期、借用人、借用数量、借用部门、备注等信息按确定按钮。

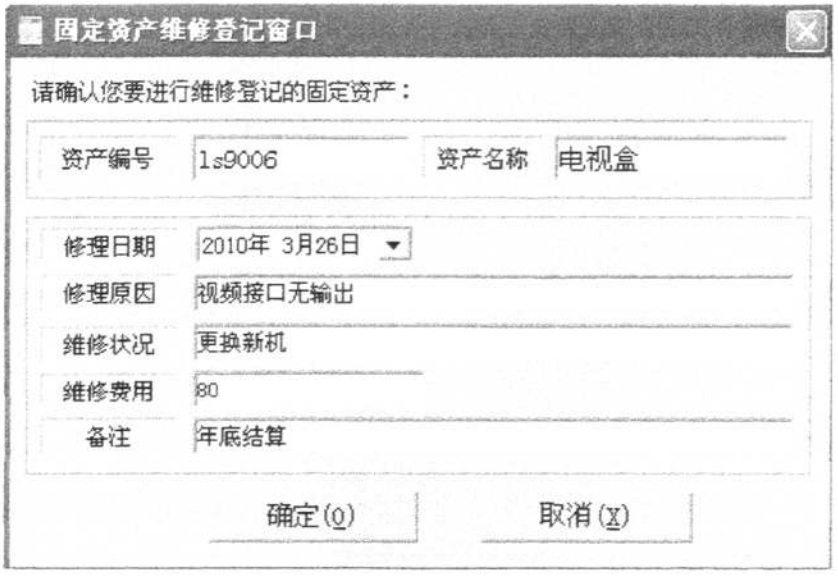


图 4-13 固定资产维修界面

在固定资产表中选定资产，然后选取资产管理菜单下的资产维修项目，进入

资产维修界面：填入修理日期、修理原因、维修状况、费用、备注等资料后按“确定”按钮完成修理登记。

图 4-14 固定资产调拨界面

在固定资产表中选定资产，然后选取资产管理菜单下的资产调拨项目，进入资产调拨界面：填入调拨日期、批准人、调出部门、调入部门、存放地点、使用人等资料后按“调拨”按钮完成调拨登记。

第五章 固定资产数据的预处理

由于数据库系统所获数据量的迅速膨胀（已达G或T数量级），从而导致了现实世界数据库中常常包含许多含有噪声、不完整、甚至是不一致的数据。显然对数据挖掘所涉及的数据对象必须进行预处理。

数据预处理主要包括:数据清洗（data clening）、数据集成（data integration）、数据转换（data transformation）和数据消减（data reduction）。

5.1 数据清洗

固定资产管理系统数据库中的数据往往是不完整的和不一致的。数据清洗过程通过填写空缺的值，平滑噪声数据，识别、删除孤立点，并解决不一致来“清理"数据。尽管大部分挖掘例程都有一些过程，处理不完整或噪声数据，但它们并非总是强壮的。相反，它们更致力于避免数据过分适合所建的模型。这样，一个有用的预处理步骤是使用某些清理例程清理你的数据。如图5-1。

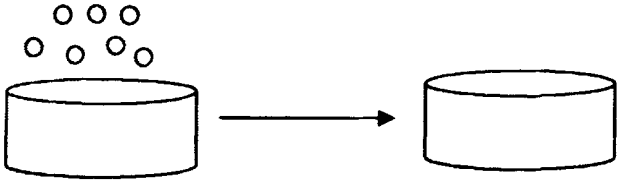


图5-1 数据清理

一般数据仓库中的数据来源于异质操作数据库。这些异质操作数据库中的数据并不都是正确的，常常不可避免地存在着不完整、不一致、不精确和重复的数

据，这些数据统称为“脏”数据。在将数据装入到数据仓库时，要对脏数据进行清洗。数据清洗可以在数据装入数据仓库之前进行，也可以在装入之后进行。数据清洗技术一般可分为基于规则的方法、可视化方法和统计学法方法。基于规则的方法根据字段定义域的知识、约束和与其它字段的关系对该字段的每一数据项进行评估：可视化方法以图形方式显示数据集的有效轮廓，从而很容易辨别脏数据；统计学法方法通过统计技术填补丢失的数据和更正错误的数据。数据选择实际上是在两个维上进行的。首先是列或参数维的选择，它是数据挖掘过程的一部分；其次是行或记录维的选择，这个选择基于各个字段的值。无论是列还是行的选择，都可以通过SQL语言进行，也可通过数据库前端工具进行。数据选择要求对问题域和基础数据有详细而深入了解。将数据选择好之后，在进行挖掘之前还需对数据进行预处理。由于原有数据库是为OLTP服务的，所以有许多便于实时数据处理的数据字段等信息对于OLAP是无用的，留在数据库中反而会严重影响数据仓库建立以及数据挖掘进程的效率。

5.1.1 遗漏数据处理

在数据预处理研究过程中，一个重要的问题就是处理数据集中的空缺值，如分析资产中某一录像机属性值中的报损日期值为空，对于为空的属性值，可以采用以下方法进行遗漏数据（missing data）处理：

1. 忽略该条记录。若一条记录中有属性值被遗漏了，则将此条记录排除在数据挖掘过程之外，尤其当类别属性（class label）的值没有而又要进行分类数据挖掘时。当然这种方法并不很有效，尤其是在每个属性遗漏值的记录比例相差较大时。

2. 手工填补遗漏值。一般讲这种方法比较耗时，而且对于存在许多遗漏情

况的大规模数据集而言，显然可行较差。

3．利用缺省值填补遗漏值。对一个属性的所有遗漏的值均利用一个事先确定好的值来填补。如：都用%& 来填补。但当一个属性遗漏值较多值，若采用这种方法，就可能误导挖掘进程。因此这种方法虽然简单，但并不推荐使用，或使用时需要仔细分析填补后的情况，以尽量避免对最终挖掘结果产生较大误差。

4．利用均值填补遗漏值。计算一个属性（值）的平均值，并用此值填补该属性所有遗漏的值。如：若一个录像机的报损日期（bsdate）为2005年9月，则用此值填补bsdate属性中所有被遗漏的值。

5．利用同类别均值填补遗漏值。这种方法尤其在进行分类挖掘时使用。如：若要对固定资产中投影仪灯泡使用时间进行分类挖掘时，就可以用在同一品牌类型投影仪下正常使用的时间属性的平均值，来填补同一投影仪类别下使用时间属性的遗漏值。

6．利用最可能的值填补遗漏值。。可以利用回归分析、贝叶斯计算公式或决策树推断出该条记录特定属性的最大可能的取值。例如：利用数据集中其它录象机的属性值，可以构造一个决策树来预测属性bsdate的遗漏值

最后一种方法是一种较常用的方法，与其他方法相比，它最大程度地利用了当前数据所包含的信息来帮助预测所遗漏的数据。通过利用其它属性的值来帮助预测属性bsdate的值。

5.1.2 噪声数据处理

噪声是指一个测量变量中的随机错误或偏差。如：物品价格，有以下几种平滑数据去掉噪声的技术：

排序后价格：4,8,15,21,21,24,25,28,34

划分为等高度 bins:

-bin 1:4,8,15

-bin 2:21,21,24

-bin 3:25,28,34

根据bin均值进行平滑：

-bin 1:9,9,9

-bin 2:22,22,22

-bin 3:29,29,29

根据bin边界进行平滑：

-bin 1:4，4,4

-bin 2:21,21,24

-bin 3:25,25,34

图5-2 利用bin方法进行平滑描述

1．Bin方法：Bin方法通过考察“邻居"(即周围的值)来平滑存储数据的值。存储的值被分布到一些“桶"或箱中。

2．聚类方法：。通过聚类分析可帮助发现异常数据（outliers），道理很简单，相似或相邻近的数据聚合在一起形成了各个聚类集合，而那些位于这些

聚类集合之外的数据对象，自然而然就被认为是异常数据。

3．计算机和人工检查结合：可以通过人工检查和计算机结合的办法来识别孤立点。例如，在一种应用中，使用信息理论度量，帮助识别手写体字符数据库中的孤立点。度量值反映被判断的字符与已知的符号相比的差异度。其差异度大于

某个阀值的模式输出到一个表中。人可以审查表中的模式，识别真正的垃圾。这比人工地搜索整个数据库快得多。在其后的数据挖掘应用时，垃圾模式将由数据库清除掉。

4. 回归方法。可以利用拟合函数对数据进行平滑。如：借助线性回归方法，包括多变量回归方法，就可以获得的多个变量之间的一个拟合关系，从而达到利用一个（或一组）变量值来帮助预测另一个变量取值的目的。利用回归分析方法所获得的拟合函数，能够帮助平滑数据及除去其中的噪声。

5.1.3 不一致数据处理

有些数据库常出现所记录的数据存在不一致的情况。其中有些数据不一致可以使用其他材料人工地加以更正。例如，输入数据时的错误可以使用原始数据上的记录加以更正。这可以与用来帮助纠正编码不一致的例程一块使用。知识工程工具也可以用来检测违反限制的数据。例如，知道属性间的函数依赖，可以查找违反函数依赖的值。由于数据集成，也可能产生不一致：一个给定的属性在不同的数据库中可能拥有不同的名字，也可能存在冗余。

5.2 数据集成和数据转换

5.2.1 数据集成

在数据集成过程中，需要考虑解决以下几个问题：

1. 模式集成问题，即如何使来自多个数据源的现实世界的实体相互匹配，这其中就涉及到实体识别问题。例如：如何确定以个数据库中的“shebei_id”与另一个数据库中的”shebei_number”是否表示同一实体。通过数据库或者数据仓库包含

的元数据可以帮助避免在模式集成时发生错误。

2．属性冗余问题。，这是数据集成中经常发生的另一个问题。若一个属性可以从其它属性中推演出来，那这个属性就是冗余属性。如：一个投影仪灯泡的平均使用时间就是冗余属性，显然它是通过投影使用时间属性推算出来的。

3．数据值冲突检测与消除。如对于一个现实世界实体，其来自不同数据源的属性值或许不同。产生这样问题原因可能是表示的差异、比例尺度不同、或编码的差异等

5.2.2 数据转换

所谓数据转换就是将数据转换或归并已构成一个适合数据挖掘的描述形式。数据转换包含以下处理内容：

1．平滑处理。帮助除去数据中的噪声，主要技术方法有：bin方法、聚类方法和回归方法。

2．合计处理。对数据进行总结或合计操作。

3．数据泛化处理。所谓泛化处理就是用更抽象（更高层次）的概念来取代低层次或数据层的数据对象。

4．规格化。规格化就是将有关属性数据按比例投射到特定小范围之中。如将投影仪采购价格属性值映射到-1.0 到1.0范围内。

5．属性构造。根据已有属性集构造新的属性，以帮助数据挖掘过程。平滑是一种数据清洗方法。合计和泛化也可以作为数据消减的方法。这些方法前面已分别作过介绍，因此下面将着重介绍规格化和属性构造方法。

根据上述几种方法，我们对所取得的不同校区固定资产的采购数据库、使用记录数据库、资产出入数据库、资产报损数据库进行了数据转换。共分以下几步进行：

第一、将采购数据库、使用记录数据库、资产出入数据库、资产报损数据库并成固定资产数据库，然后分割成小的数据表。

第二、通过每个设备的具有唯一性的属性编号，我们将存在于固定资产数据库数据搜索出来，由于数据量庞大，我们利用第一步中分割后的数据库，从中寻找数据，并将数据变成典型数据，该数据中包括：采购数据库、使用记录数据库、资产出入数据库、资产报损数据库等，为数据挖掘提供数据源。

第三、计算属性：由于集成数据库的得到的数据依然反映的是固定资产综合信息的记录，为对资产管理进行综合评定，将不同类别的设备总数进行求和。

第四、归一化处理：在固定资产数据库中的设备使用记录项，记录很多，进行处理时，会掩盖其它数据项在数据处理中的作用，因此需要将其作归一化处理。

第五、均值处理：采购数据库中的价格项也可以做均值处理，即将各类资产采购价格相加后除以记录数，这样就可以合并多项数据记录为一项了，大大减少了数据量，在做某些总量统计时可以有效得提高速度。

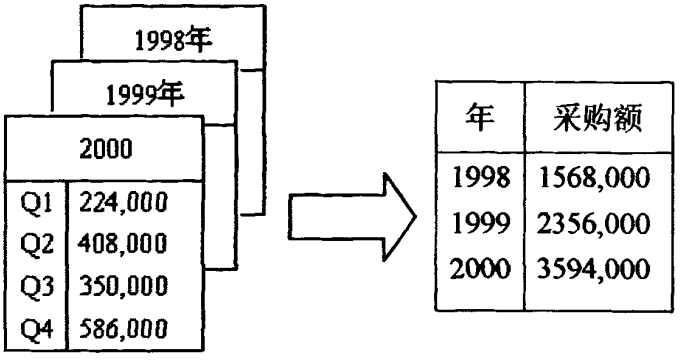
经过集成形成的固定资产管理系统数据仓库中，生成了几种可供数据挖掘的数据源，包括：典型设备数据，通过归一化和均值处理的减约数据等。

5.3 数据消减

在大规模的数据库上进行复杂的数据分析和挖掘将需要很长的时间，尤其是需要交互式数据挖掘时这样的分析变的不现实和不可行，而数据消减技术可以从原有庞大的数据集中或得一个精简的数据集合，并保持原有数据的完整性，这样在精简数据集上进行的数据挖掘效率明显更高，并且挖掘出的结果与原有数据集所获结果基本一致。

数据消减的策略主要由以下几种：

1．数据立方体合计：聚集操作用于数据立方体中的数据。如图



如图 5-3 数据合计示意图

2．位数削减：主要用于检测和消除无关，弱相关、或冗余的属性或维（数据仓库中属性）。

3．数据压缩：使用编码技术压缩数据集的大小。

4．数据块削减：利用更简单的数据表达形式，如：参数模型、非参数模型（聚类、采样、值方图等），来取代原有数据。

5．离散化与概念层次产生：离散化就是利用取值范围或者更高层次概念来替换初始数据。利用概念层次可以帮助挖掘不同抽象层次的模式知识。当我们为分析投影仪情况收集了相关数据，这些数据由各个校区统计数据组成。然后，我们又对台投影仪使用情况感兴趣，继而对所有投影仪感兴趣。可以对已经得到的这些数据再聚集，从而得到更宏观的知识。结果数据量小得多，并不丢失分析任务所需的信息。

第六章 固定资产管理系统的决策树算法

6.1 决策树介绍

决策树算法是目前应用广泛的推理归纳算法之一，是一种接近离散值函数的算法，可以把它看作是一个布尔函数。它通常用来形成分类器和预测模型，着眼于从一组无规则、无次序的事例中推理出决策树表示形成的分类规则。它采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较并根据不同的属性值判断从该结点向下的分支，最后在决策树的叶结点得到结论。因此从根到叶结点的一条路径就对应着一条合取规则，而整棵决策树就对应着一组析取表达式规则。

6.2 决策树 ID3 算法在固定资产管理系统中的应用

无锡旅游商贸高等职业技术新购买了一批HP台式机，拟分配给各个系部机房和公用机房使用，因为数量不多不能满足各个部门的需求，所以需要按照各部门对台式机的使用效率、需要程度做一个定位，通过决策树的分析将台式机分配给最需要的部门使用。通过对数据仓库中的各部门实验室台式机的使用情况的纪录进行整理，得出如下表格：

表6-1 台式机状况统计表

数量	机房	规模	电脑档次	使用率
160	系部机房	大	一般	高
80	公用机房	大	高	高
50	系部机房	一般	高	低
40	公用机房	一般	高	高
70	系部机房	一般	高	低
130	公用机房	一般	一般	低
100	系部机房	一般	一般	高

以ID3算法为例，给无锡旅游商贸高等职业技术学校的设备分配方案提出一个可行的决策树分析方法。

ID3算法处理流程：

1．从训练集中随机选择一个既含正例又含反例的子集(称为窗口)。

2．用建树算法使当前窗口形成一棵决策树。

3．对训练集(窗口除外)中例子用所得决策树进行类别判定，找出错判的例子。

4．若存在错判的例子，把它们插入窗口，转2，否则结束。主算法流程如图1所示，其中PE、NE分别表示正例集和反例集，它们共同组成训练集。PE ，PE"，和NE ，NE”，分别表示正例集和反例集的子集。

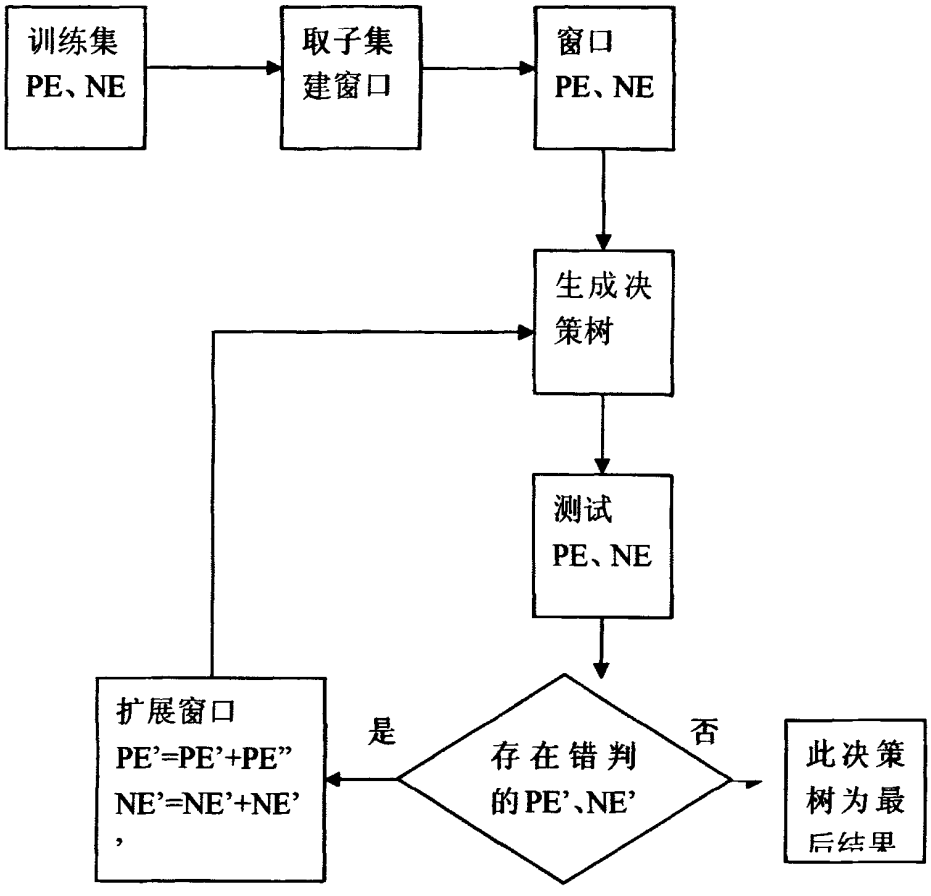


图 6-1 ID3主算法流程

ID3 算法流程：

```
// id3.cpp
#pragma warning (disable: 4996)

#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <math.h>
```

```
#include <list>
using namespace std;

    #define LN_2 0.693147180559945309417
#define ID3_ERROR 9999999

    #define A_CHAR_MAX 16
#define A_VALUE_MAX 16
#define A_NUM_MAX   32

    #define SAMPLES_MAX 256
#define ALL -1
#define NUL -2

    #define YES     1
#define NO      0
#define NUKOWN -1

    #define VALID '*'
#define INVALID '-'

    // 属性
struct Attribute
{
char name[A_CHAR_MAX];          // 属性名称
int num;                        // 属性值个数
```

```
char att[A_VALUE_MAX][A_CHAR_MAX]; // 属性值
};

    // 假设
struct Hypothesis
{
int num;                    // 属性个数
Attribute an[A_NUM_MAX];        // 属性集合
};

    // 假设值
struct HypoValue
{
int value[A_NUM_MAX];
};

    // 样本
struct Sample
{
HypoValue ev;                // 假设
int result;                // 正例/反例
};

    Hypothesis g_Hypo;                // 假设集合
Sample g_sa[SAMPLES_MAX];            // 样本空间
int g_sn;                    // 样本数
```

```
    bool ReadHypothesis(const char* filename);
bool ReadSamples(const char* filename);
int CheckAllPositive();
int CheckAllNegative();
int CreateTree(char samplevalid[SAMPLES_MAX], char attvalid[A_NUM_MAX],
FILE*, int);
int NotAllSame(char samplevalid[SAMPLES_MAX]);
int FindAtt(char attvalid[A_NUM_MAX], char samplevalid[SAMPLES_MAX]);
void Disaster(int);
int ID3(const char* filename);

    // 从文件中读取假设集合
/*/ 文件格式
[集合个数 n]
[属性名称 1] [属性值个数] [属性值 1] [属性值 2] [属性值 3] .....
[属性名称 2] [属性值个数] [属性值 1] [属性值 2] [属性值 3] .....
.....
[属性名称 n] [属性值个数] [属性值 1] [属性值 2] [属性值 3] .....
/*/
bool ReadHypothesis(const char* filename)
{
FILE* file;
if (fopen_s(&file, filename , "r"))
  return false;
```

```
int i,j,k;
fscanf(file, "%d\n", &g_Hypo.num);
for (i=0; i<g_Hypo.num ; i++)
{
   fscanf(file, "%s%d\n", g_Hypo.an[i].name, &k);
   g_Hypo.an[i].num = k;
   for (j=0; j<k; j++)
   {
    fscanf(file, "%s", g_Hypo.an[i].att[j]);
   }
   fscanf(file, "\n");
}
fclose(file);
return true;
}
```

// 从文件中读取样本

/*/ 文件格式

[样本个数 m]

[样本 1 属性 1 的值的序号] [样本 1 属性 2 的值的序号] ..... [样本 1 属性 n 的值的序号] [1（正例）或者 0（反例）]

[样本 2 属性 1 的值的序号] [样本 2 属性 2 的值的序号] ..... [样本 2 属性 n 的值的序号] [1（正例）或者 0（反例）]

```
[样本 m 属性 1 的值的序号] [样本 m 属性 2 的值的序号] ..... [样本 m 属性 n 的值
的序号] [1（正例）或者 0（反例）]
/*/
bool ReadSamples(const char* filename)
{
FILE* file;
if (fopen_s(&file, filename , "r"))
   return false;
int i,j;
fscanf(file, "%d\n", &g_sn);
for (i=0; i<g_sn ; i++)
{
   for (j=0; j<g_Hypo.num; j++)
   {
    fscanf(file, "%d", &g_sa[i].ev.value[j]);
   }
   fscanf(file, "%d\n",&g_sa[i].result);
}
fclose(file);
return true;
}

      int CheckAllPositive()
{
```

```
int i;
for(i=0;i<g_sn;i++)
{
  if (g_sa[i].result == NO)
  {
   return 0;
  }
}
return 1;
}

     int CheckAllNegative()
{
int i;
for(i=0;i<g_sn;i++)
{
  if (g_sa[i].result == YES)
  {
   return 0;
  }
}
return 1;
}
```

```
    int NotAllSame(char samplevalid[SAMPLES_MAX])
{
int i, y_tot = 0, n_tot = 0;
for (i=0; i<g_sn; i++)
{
  if (samplevalid[i] == VALID)
  {
   if (g_sa[i].result == YES)
    ++y_tot;
   if (g_sa[i].result == NO)
    ++n_tot;
  }
}
if (n_tot == 0)
  return 2; /* all yes */
else if (y_tot == 0)
  return 3; /* all no */
else
  return 1;
}

    int FindAtt(char attvalid[A_NUM_MAX], char samplevalid[SAMPLES_MAX])
{
int i, j, l, y_tot = 0, n_tot = 0, y_tot_2, n_tot_2;
```

```
int tot_diff_atts;
int att_no = 0;
double max_inf_gain = -1.0;
double entropy, entropy_2, r_entropy_tot;
double att_entropy[A_NUM_MAX];
char valid_2[SAMPLES_MAX];

for (i=0; i<g_Hypo.num; i++)
{
   att_entropy[i] = -2.0;
}

for (i=0; i<g_sn; i++)
{
   if (samplevalid[i] == VALID)
   {
    if (g_sa[i].result == YES)
     ++y_tot;
    if (g_sa[i].result == NO)
     ++n_tot;
   }
}
if (y_tot == 0 || n_tot == 0)
   entropy = 0.0;
```

```
else
{
   entropy = 0.0 - ((y_tot/(double)(y_tot+n_tot))*log((y_tot/(double)(y_tot+n_tot))))
    - ((n_tot/(double)(y_tot+n_tot))*log((n_tot/(double)(y_tot+n_tot))));
}
for (i=0; i<g_Hypo.num; i++)
{
   if (attvalid[i] == VALID)
   {
    r_entropy_tot = 0.0;
    tot_diff_atts = g_Hypo.an[i].num;
    for (j=0; j<tot_diff_atts; j++)
    {
     memset (valid_2, INVALID, g_sn);
     for (l=0; l<g_sn; l++)
     {
      if ((g_sa[l].ev.value[i] == j+1) && (samplevalid[l] == VALID))
       valid_2[l] = VALID;
     }
     y_tot_2 = 0;
     n_tot_2 = 0;
     for (l=0; l<g_sn; l++)
     {
      if (valid_2[l] == VALID)
```

```
    {
     if (g_sa[l].result == YES)
      ++y_tot_2;
     if (g_sa[l].result == NO)
      ++n_tot_2;
    }
   }
   if (n_tot_2 == 0 || y_tot_2 == 0)
    entropy_2 = 0.0;
   else
   {
                entropy_2 = 0.0 -
((y_tot_2/(double)(y_tot_2+n_tot_2))*log((y_tot_2/(double)(y_tot_2+n_tot_2))))
                                                                    -
((n_tot_2/(double)(y_tot_2+n_tot_2))*log((n_tot_2/(double)(y_tot_2+n_tot_2))));
   }
       r_entropy_tot = r_entropy_tot + (entropy_2 *
((n_tot_2+y_tot_2)/(double)(n_tot+y_tot)));
  }
  att_entropy[i] = entropy - r_entropy_tot;
 }
}
for (l=0; l<g_Hypo.num; l++)
{
```

```
    if (att_entropy[l] >= max_inf_gain)
    {
     max_inf_gain = att_entropy[l];
     att_no = l;
    }
}
if (max_inf_gain == 0.0)
{
   return ID3_ERROR;
}
return att_no;
}

      void Disaster(int num)
{
switch(num)
{
case 1: printf("** ID3 failure **\n");
   break;
case 2:
   break;
default:
   break;
```

```
}
}

    int CreateTree(char samplevalid[SAMPLES_MAX], char attvalid[A_NUM_MAX],
FILE *opf, int tab_cnt)
{
char samplevalid2[SAMPLES_MAX];
char attvalid2[A_NUM_MAX];
int j, l, i, ret, tot_diff_atts, att_no;
for (i=0; i<tab_cnt+tab_cnt; i++)
{
  fprintf(opf,"\t");
}
tab_cnt++;
if ((att_no = FindAtt(attvalid, samplevalid)) == ID3_ERROR)
{
  return ID3_ERROR;
}
strncpy(attvalid2, attvalid, g_Hypo.num);
attvalid2[att_no] = INVALID;
fprintf(opf, "[%s]\n", g_Hypo.an[att_no].name);
tot_diff_atts = g_Hypo.an[att_no].num;
for (j=0; j<tot_diff_atts; j++)
{
```

```
//valid[M1-1] = '\0';
strncpy(samplevalid2, samplevalid, g_sn);
for (l=0; l<g_sn; l++)
{
 if (g_sa[l].ev.value[att_no] != j+1)
 {
  samplevalid2[l] = INVALID;
 }
}
if ((ret = NotAllSame(samplevalid2)) == 1)
{
 for (i=0; i<tab_cnt+tab_cnt-1; i++)
 {
  fprintf(opf, "\t");
 }
 fprintf(opf, " %s\n", g_Hypo.an[att_no].att[j]);
 if (CreateTree(samplevalid2, attvalid2, opf, tab_cnt) == ID3_ERROR)
 {
  return ID3_ERROR;
 }
}
else
{
 for (i=0; i<tab_cnt+tab_cnt-1; i++)
```

```
    {
     fprintf(opf, "\t");
    }
    if (ret == 2)
    {
     fprintf(opf, " %s\t - YES\n", g_Hypo.an[att_no].att[j]);
    }
    else
    {
     fprintf(opf, " %s\t - NO\n", g_Hypo.an[att_no].att[j]);
    }
   }
}
return 1;
}


// ID3 算法，结果保存在 filename 文件中
int ID3(const char* filename)
{
int tab_cnt = 0;
char samplevalid[SAMPLES_MAX];
char attvalid[A_NUM_MAX];
FILE *opf;
```

```
if ((opf = fopen(filename, "w")) == NULL)
{
   printf("File error : Cannot create output file TREE\n");
   return 0;
}
fprintf(opf, "\n");
if (CheckAllPositive())
{
   fprintf(opf,"HALT:all_positive\n");
   fclose(opf);
   return 1;
}
if (CheckAllNegative())
{
   fprintf(opf,"HALT:all_negative\n");
   fclose(opf);
   return 1;
}
memset (samplevalid, VALID, g_sn);
memset (attvalid, VALID, g_Hypo.num);
if (CreateTree(samplevalid, attvalid, opf, tab_cnt) == ID3_ERROR)
{
   return 0;
}
```

```
fclose(opf);
return 1;
}

    int main(int arc, char** argv)
{
// 读取假设和样本
if (!ReadHypothesis(argv[1]))
{
  printf("read hypothesis file error");
  return 0;
}
if (!ReadSamples(argv[2]))
{
  printf("read samples file error");
  return 0;
}
if (ID3(argv[3]))
  printf ("Decision Tree has been created with ID3 and stored in %s\n", argv[3]);
else
  Disaster(1);

    getchar();
return 0;
```

}

得出如下实验结果：

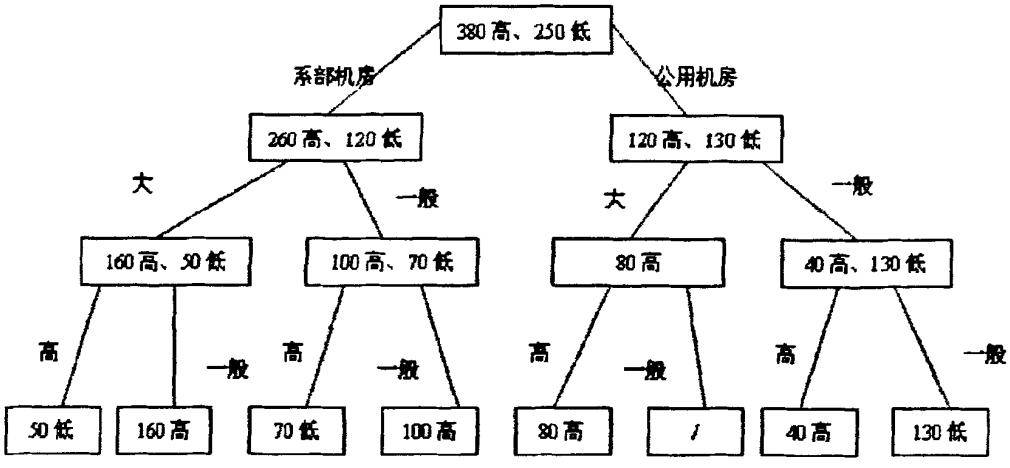


图 6-2 分配决策树

决策树分析首先针对上表计算各个属性的信息，并将属性从大到小重新排列，我们先假设以上各属性的信息相等。然后建立决策树：首先按位置(任选)建立决策树，得到数据的第一次分组，然后依次按同样方法按规模、档次分组，得到数据的第二次分组和数据的第三次分组。完整的决策树如图2所示：根据上面的决策树，我们可以得出以下决策规则：

1．档次一般的台式机在规模大的系部机房使用率高

2．档次一般的台式机在规模一般的系部机房使用率高

3．档次高的台式机在规模大的公用机房使用率高

4．档次高的台式机在规模一般的公用机房使用率高

从上述规则我们可以得出结论：台式机的分配可以把档次一般的大部分都分配给系部机房使用，把档次高的分配给公用机房使用。最后我们可以利用决策树和导出的规则对计划分配的台式机的部门是否合适作出评估。

第七章 总结与展望

7.1 全文总结

目前，数据挖掘在高校资产管理中的应用仅仅处于初步应用阶段，对于数据挖掘的方法和算法还需要进一步研究，限于本人知识有限，在本文中就资产管理过程中设备的分配问题提出的IDC3决策树模型进行了探讨，在数据挖掘的可靠性和算法优化方面还有待进一步研究。数据仓库与数据挖掘在高等教育管理领域的应用是一个综合复杂的系统工程，可以涵盖招生、就业、科研、人事、财务等方方面面。构建开发一套基于数据仓库的高校综合信息决策支持管理系统，将有效地推动高校的深化改革，使高校管理逐步走向信息化、科学化，这是作为高校管理工作者努力的目标。

7.2 研究展望

因为实验数据的限制，本文所得出的结论仅仅建立在一些小型标准测试数据上。而现实生活中的企业数据库则往往包含成千上万条客户数据，且数据特点复杂多变，所以本课题仍存在很大的可研究空间。

参考文献

[1] 数据仓库[Z]. 维基百科, http://www.wikipedia.org

[2] E. F. Codd, S. B. Codd and C. T. Sally. Beyond decision support[J]. Computer World, 1993,27(30):553-556.

[3] 张菽,刘春红,敬卿. 数据仓库的建设与数据挖掘技术浅析[J]. 高校图书馆工作. 2000(03):27-30

[4] J. Han, M. Kamber. 数据挖掘概念与技术[M]. 机械工业出版社, 2001

[5] 朱丽. 浅谈利用数据仓库技术构建环境数据中心[J]. 环境科学导刊, 2008(03):89-91

[6] 潘海芸. 浅谈数据仓库在环境保护工程中的应用及意义[J]. 治淮, 2005(10): 39-40

[7] 李爱红, 胡平, 林宣雄. 数据仓库在水环境质量评价系统中的应用前景[J]. 环境科学与技术, 2004.3: 52-54

[8] 崔晓军,薛永生. 数据仓库集成环境研究与实现[J]. 计算机应用研究, 2006(12):184-186+190.

[9] 汪彦云. 市级环境信息数据仓库开发与应用研究[J]. 环境科学导刊, 2008(S1):37-39

[10]林杰斌, 刘明德, 陈湘等. 数据挖掘与 OLAP 理论与实务[M]. 北京:清华大学出版社, 2003

[11]廖剑岚. 决策支持系统中的数据挖掘与 OLAP——数据仓库环境下的信息分析 华东师范大学, 2002

[12]李菁菁, 邵培基, 黄亦潇. 数据挖掘在中国的现状和发展研究[J]. 管理工程学[13]

[13]马志军. 基于数据仓库的生态环境监测与管理决策支持系统[J]. 电脑开发

与应用，2002.15(12):10-11

[14] 史忠植. 知识发现. 清华大学出版社. 2002.

[15] Han Jiawei. 数据挖掘概念与技术. 北京:机械工业出版社,2001.3～5.

[16] Fayyad U. Data mining and knowledge discovery in databases implications for scientific databases [A]. Scientific and Statistical Database Management, Proceedings, Ninth International Conference on [C]. IEEE, 1997.2～11.

[17] Cheng QM, Jason TL, Wang. DNA sequence classification via an expectation maximization algorithm and neural networks: a case study. Systems, Man and Cybernetics, Part C: Applications and Reviews [J]. IEEE Transactions on, 2001,31(4):468～475.

[18] Adomavicius G, Tuzhilin A. Using data mining methods to build customer profiles[J]. Computer, 2001, 34(2):74～82.

[19] Syeda M, Yan Q Z, Pan Y. Parallel granular neural networks for fast credit card fraud detection. Fuzzy Systems[A]. Proceedings of the 2002 IEEE International Conference[C], 2002, 1:572～577.

[20] Bhandari, Inderpal, Colet. Advanced Scout: data mining and knowledge discovery in NBA

data[J]. Data Mining and Knowledge Discovery, 1997, 1(1):121～125.

[21] 李菁菁，邵培基，黄亦潇. 数据挖掘在中国的现状和发展研究[J]. 管理工程学报,2004,18(3):10～15.

[22] 慕春棣，戴剑彬，叶俊.用于数据挖掘的贝叶斯网络[J] .软件学报，2000，11(5)：660～666.

[23] 宫秀军，刘少辉，史忠植.一种增量贝叶斯分类模型[J].计算机学

报, 2002, 25(6):645～650.

[24] 季文赟, 周傲英, 张亮, 等. 一种基于遗传算法的优化分类器的方法[J]. 软件学报, 2002, 13(2):245～249.

[25] 时施仁, 史忠植. 基于 CBR 的中心渔场预报[J]. 高技术通讯, 2001, 5:64～68.

[26] 毛国君, 刘椿年. 基于项目序列集操作的关联规则挖掘算法[J]. 计算机学报, 2002, 25(4):417～422.

[27]谷和启. 数据仓库创建、设计与开发 [J]. 中文信息, 2003(04): 32-34

[28]龙志勇. 数据挖掘在电信行业客户关系管理中的应用[J]. 信息网络, 2003(12): 26-28

[29]李新荣, 米新江. 数据仓库的研究与发展现状[J]. 廊坊师范学院学报, 2001(04): 68-72

[30]徐立臻, 谢鸿强, 董逸生. 数据仓库系统中源数据的提取与集成[J]. 小型微型计算机系统, 2003, 24(05): 869-873

[31]连立贵, 沈彦南, 蔡家楣. 数据仓库的填充与访问[J]. 计算机工程与应用, 2002(07):188-189

[32]王珊等. 数据仓库技术与联机分析处理[M]. 北京:科学出版社, 1998

[33]M. Riedewald, D. Agrawal, A. El Abbadi. Fiexible data Cubes for online aggregation[C]. Proc of the 8th Int' l Conf on Database Theory, 2001:159-173.

[34]王祥. 数据仓库在电信经营分析系统中的应用研究[D]. 武汉理工大学, 2006

[35]薛惠忠, 庄晓青, 董逸生. 数据仓库中的数据集成转换[J]. 现代计算机, 2003(12): 78-82

[36樊明辉, 陈崇成, 涂建东. 环境专题科学数据仓库及WEB联机分析处理的设计与实现[J]. 福州大学学报(自然科学版), 2004(05):20-25

[37] 程岩,黄梯云.粗糙集中定量关联规则的发现及其规则约简的方法研究[J].管理工程学报,2001,15(3):73～77.871.

[38] 倪志伟，蔡庆生，方瑾. 用神经网络来挖掘数据库中的关联规则[J]. 系统仿真学报,2000,12(6):685～687.

[39] 程继华，施鹏飞.多层次关联规则的有效挖掘算法[J].软件学报,1998,9(12):937～941.

[40] 苑森淼,程晓青.数量关联规则发现中的聚类方法研究[J].计算机学报,2000

[41] 肖利，金远平，徐宏炳，等. 基于多维标度的快速挖掘关联规则[J]. 软件学报,1997,10(7):749～753.

[42] 陆建江，宋自林，钱祖平.挖掘语言值关联规则[J].软件学报,2001,12(4):607～611.

[43] 陆建江，钱祖平，宋自林. 正态云关联规则在预测中的应用[J]. 计算机研究与发展,2000,37(11):1317～1320.

[44] 程继华,施鹏飞.概念指导的关联规则的挖掘[J].计算机研究与发展,1999,36(9)

[45] 谢志鹏，刘宗田.概念格与关联规则发现[J].计算机研究与发展,2000,37(12):1415～1421.

[46] 肖利,王能斌,徐宏炳,等.挖掘转移规则:一种新的数据挖掘技术[J].计算机研究与发展,1998,35(10):902～906.1096.

[47] Codd E F, Codd S B, Salley C T.Beyond decision support [N]. Computer World,27,July 1993.

[48] Cai Y, Cercine N, Han J W. Attribute oriented induction in relational database[J].Knowledge Discovery in Database, MA:AAA/MIT press,1991.213～228.

[49] 陈红梅，王丽珍. 面向属性的量化归纳[J]. 计算机研究与发展, 2001, 38(2):150～156.

[50] 周生炳，张钹，成栋. 基于规则面向属性的数据库归纳的无回溯算法[J]. 软件学报, 1999, 10(7):673～678.

致 谢

在论文完成之际，我要特别感谢我的指导老师李斌教授的热情关怀和悉心指导。在我撰写论文的过程中，李斌教授倾注了大量的心血和汗水，无论是在论文的选题、构思和资料的收集方面，还是在论文的研究方法以及成文定稿方面，我都得到了李斌教授悉心细致的教诲和无私的帮助，特别是他广博的学识、深厚的学术素养、严谨的治学精神和一丝不苟的工作作风使我终生受益，在此表示真诚地感谢和深深的谢意。

在论文的写作过程中，也得到了许多同学的宝贵建议，同时在工作过程中还得到许多同事的支持和帮助，在此一并致以诚挚的谢意。

最后，向在百忙中抽出时间对本文进行评审并提出宝贵意见的各位专家表示衷心地感谢！

攻读学位期间发表的学术论文

[1] 《计算机网络实训》教材. 高教出版社, 2007 年 9 月, 编写第二章.

[2] 《浅谈职业学校的计算机教学》 2008 无锡职教论文三等奖