# 摘　要

随着国内电信市场竞争的日趋激烈，电信运营商的经营模式逐渐从“业务驱动”向“客户驱动”转化、从“粗放式经营”向“精确化管理”转变。为了更好地做到对企业的决策支持，经营分析系统孕育而生。本文的目标是在电信经营分析系统下，研究数据的处理流程，重点研究数据挖掘理论与技术在系统中的应用。

本文首先简要概述数据挖掘的概念、挖掘过程模型、数据预处理、数据挖掘分类、常用技术、热点研究方向，以及数据挖掘和数据仓库、数据挖掘和 OLAP 之间的关系；接着介绍经营分析系统中客户细分所应用的数据挖掘技术——聚类算法，包括聚类的定义、算法的要求、主要的聚类方法以及 k-means 算法原型和改进方向，并着重介绍基于 K-D 树的改进 k-means 算法；然后对电信经营分析系统进行概述，主要涉及该系统的建设背景、功能架构、数据挖掘技术在系统中的实际应用，并结合数据仓库、OLAP、数据挖掘在系统中的定位对系统中的核心数据处理流程进行详细介绍。

本文第 5 章（核心章节）根据标准的数据挖掘过程模型 CRISP-DM、应用改进的初始聚类中心选取方法和基于 K-D 树的改进 k-means 算法对某地电信公司经营分析系统的客户细分过程进行详细的描述。首先是电信客户信息的组成、数据挖掘的基础－宽表结构等的介绍，接着给出聚类模型的整体结构、各主要模块功能及处理流程，并从业务上对细分的结果进行解释和特征描述，给出相关的营销建议，起到决策支持的作用。最后结合实验数据，分析改进算法的参数设置问题，验证改进的 k-means 算法相比标准的 k-means 算法在效率上和稳定性上都有较大提升；同时针对本文算法中的不足提出进一步的改进意见。

关键词：数据挖掘、CRISP-DM、电信经营分析系统、客户细分、K-D 树，k-means 算法

# ABSTRACT

As the enhancement of competition in telecom market, the management pattern of China Telecom has changed from "business- oriented" to "custom- oriented", from "extensive management" to "accurate management". In order to support decision-making effectively, the Telecom Manage-Analysis System has been build up .The paper research how data is deal with in the system and the most important one is the application of data mining in the system..

Firstly, some background knowledge is summarized briefly in the front of the paper, including the conception of data mining、the process of data mining model、the classification of technology about data mining、the relationship of data mining and data warehouse、the relationship of data mining and OLAP; and then something about one technology of data mining - cluster analysis is introduced, such as definition、the requirement of the algorithm、some methods of cluster analysis、k-means algorithm which is used frequently in cluster analysis and a more effective k-means algorithm based on K-D tree; subsequently, the Telecom Manage-Analysis System is recommended briefly, including the background 、structure of the system、the application of data mining in the system, the emphases of the part is the core process of dealing with data.

Lastly, according to the standard process model of data mining - CRISP-DM, the process of customer segmentation which uses k-means algorithm based on K-D tree is researched deeply, and the details involve the structure of clustering model、the function、realization of each module. The result about the compare between standard k-means algorithm and k-means algorithm based on K-D tree、the explanation of customer segmentation result and some advice based on analysis are given out in the end of the paper and k-means algorithm based on K-D tree is proved more effective according to the result.

Keywords: Data Mining、 CRISP-DM、Telecom Manage-Analysis System、Customer Segmentation、K-D Tree、K-means

# 缩略词

| 缩略词 | 英文全称 | 译文 |
| --- | --- | --- |
| OLAP | On-Line Analytical Processing | 联机分析处理 |
| DM | Data Mining | 数据挖掘 |
| DW | Data Warehouse | 数据仓库 |
| ETL | Extraction、Transformation、Loading | 抽取、转换和加载 |
| ODS | Operational Data Store | 操作数据仓储 |
| KPI | Key Performance Indicator | 关键绩效指标 |
| GUI | Graphical User Interface | 图形用户接口 |
| API | Application Programming Interface | 应用程序接口 |
| ARPU | Average Revenue Per User | 每用户平均收入 |
| BSS | Business Support System | 业务支撑系统 |
| OSS | Operating Support System | 运营支撑系统 |
| MSS | Management support system | 管理与经营支撑系统 |

# 图表清单

# 南京邮电大学学位论文独创性声明

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

研究生签名：肖云宾 日期：2007.4.12

# 南京邮电大学学位论文使用授权声明

南京邮电大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权南京邮电大学研究生部办理。

研究生签名：肖云宾 导师签名：[illegible] 日期：2007.4.12

# 第1章 引言

## 1.1 中国电信发展现状

中国电信行业经过近十多年的发展，取得了举世瞩目的成就。其中，电话用户规模世界排名第一，信产部公布了2006年电信业务数据，中国电话用户总数已突破8亿户大关；互联网用户仅次于美国，世界排名第二。“八五”和“九五”时期电信业保持30%～40%的高增长，近几年趋于平稳，只有12%～15%的增长率，但都远远高于同期GDP的增幅[1]。

从国内市场看，目前我国电信业已形成中国电信、移动、联通、网通、铁通等多家公司竞争的格局，在所有业务领域都有两家或两家以上的公司经营，电信市场竞争日趋激烈。价格战、服务战表现最为突出，各运营商为争夺客户、留住客户，纷纷采取各种名目繁多的套餐，变相或直接降低资费，吸引用户，移动通信市场竞争尤为突出。

从国际市场看，2001年11月10日我国正式加入WTO，标志着我国电信市场将融入国际市场，逐步对外开放。如今，5年的过渡期已经过去，从2006年开始，我国全面实现对外承诺，取消地域限制，国外跨国电信公司逐步进入中国市场，电信市场竞争将更加激烈，尤其是增值业务市场竞争更趋白热化。

在激烈的市场竞争中，中国电信面临着增量市场日趋减缓、市场竞争愈演愈烈、移动分流不断加剧、传统语音业务增长下降、缺乏新的业务增长点等诸多难题。同时，3G时代的即将来临，也给中国电信一个新的发展契机。如何抓住机遇、应对挑战，则是中国电信发展过程中的重要研究课题。

## 1.2 电信客户细分的必要性

随着中国电信市场的对外开放和3G时代的临近，电信市场的竞争愈加激烈。在中国电信用户持续快速增长的同时，电信运营商不得不面对ARPU值不断降低、增量不增收的现象，特别是国内电信运营商所推出的比较简单的价格比拼和优惠活动。面对这种恶性循环，各电信运营商从简单的价格竞争过渡到电信品牌、业务、服务的竞争，提高用户的满意度和忠诚度[2]。

电信客户数量巨大，每个客户都用不同的需求，如何最大限度地满足不同客户的需求？实践证明，客户细分是一种行之有效的方法。所谓的客户细分就是根据消费者之间通

信需求的差异以及消费者的自身情况，如客户对电信产品的需求、偏好、消费行为、消费能力等方面的差异，把一个电信市场划分成多个通信客户群体，针对不同的群体实施不同的服务，提供差异化营销服务。

## 1.3 本文的工作以及思路

本文结合数据挖掘的理论和电信经营分析系统的实际应用，介绍数据挖掘在经营分析中的应用，详细描述客户细分过程。内容安排如下：

第1章：引言。介绍我国电信行业的发展现状，引出电信客户细分的必要性。

第2章：数据挖掘概述。介绍数据挖掘的概念，数据挖掘与数据仓库、OLAP的关系，数据挖掘过程模型CRISP-DM以及数据挖掘技术分类、热点研究方向等。

第3章：数据挖掘中的聚类算法。由于客户细分中应用了聚类算法，因此本章重点介绍了数据挖掘中的聚类算法的定义、要求、主要的聚类算法，并详细介绍了k-means算法的实现，在这个基础上提出了基于K-D树的改进的k-means算法。

第4章：电信经营分析系统概述。本章介绍电信经营分析系统的建设背景、功能框架，着重介绍了系统中的核心数据处理模块。最后介绍数据挖掘在电信经营分析系统中的实际应用。

第5章：应用数据挖掘技术进行客户细分。本章根据数据挖掘过程模型CRISP-DM介绍客户细分的详细过程，应用改进的k-means算法改善客户细分，并通过对比分析，直观地体现出改进算法在效率和稳定性方面要大大优于标准的k-means算法。

第6章：总结。对已完成工作的总结和对未来的展望。

# 第2章　数据挖掘技术概述

## 2.1　数据挖掘的概念

随着社会信息化的飞速发展，各行各业都积累了大量的生产和管理数据。数据的极大丰富是否就意味着信息的极大丰富呢？事实说明没有经过整理和分析的大量数据就像“坟墓”，根本无法为决策者提供决策依据。如何把“数据坟墓”转变成为“知识金库”，就是数据挖掘需要做的事情。

数据挖掘[3][4] (Data Mining)作为数据库知识发现的核心技术，就是从大量的、不完全的、有噪声的、模糊的、随机的数据中，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，提取的知识一般可以表示为概念、规则、规律、模式等形式。确切地说，数据挖掘过程就是一种决策支持过程，主要基于人工职能、机器学习、统计学等技术，高度自动化地分析生产业务中原有的数据，做出归纳性的推理，从中挖掘出潜在的模式，预测客户的行为，帮助企业的决策者调整市场策略，减少风险，做出正确的决策。

还有很多和数据挖掘相近似的术语，如从数据库中发现知识(KDD)、数据分析、数据融合(Data Fusion)以及决策支持等。数据挖掘是一门很广义的交叉学科，图 2-1 形象地表现出数据挖掘与很多学科之间的相互渗透的关系。数据挖掘汇聚了不同领域的研究者，尤其是数据库、人工智能、数理统计、可视化、并行计算等方面的学者和工程技术人员。

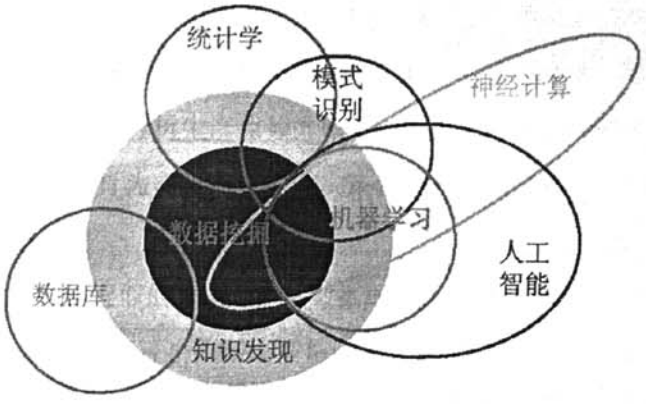


图 2-1　数据挖掘和各学科之间相互渗透的关系

## 2.2　DW 项目的知识技能需求

在实施一个具体的数据挖掘项目时，需要具备的知识和技能包括：

- 某个领域的业务知识（业务专家）：能够理解业务问题的细节和特殊性、背景业务知识、内容含义、术语，知道对该业务问题的当前处理方法和优劣。
- 数据知识和处理能力（数据专家）：理解数据的结构、格式，数据源的状况，数据量的大小，有对数据操作的能力。
- 分析方法和技能（分析专家）：理解和该业务问题相关的分析方法的特点和局限，有使用相关算法进行数据分析和建模的能力。

## 2.3 数据挖掘与数据仓库的关系

### 2.3.1 什么是数据仓库

国内外很多学者都提出数据仓库的描述，但很难给出数据仓库严格的定义。宽松地讲，数据仓库就是一个数据库，但是在这个数据库中存储的数据内容和数据组织方式以及维护方式和一般的操作型数据库不同。数据仓库存储大量的历史数据，允许将各种应用集成在一起，对各种历史数据进行分析挖掘，为分析人员提供信息处理平台。

WH.H.Inmon这位数据仓库系统构造方面的领头设计师给数据仓库的定义是："数据仓库是一个面向主题的、集成的、时变的、非易失的数据集合，支持管理决策制定"[5]。

从这四个方面能看出数据仓库与普通的操作型数据库的区别。

(1) 面向主题的：数据仓库围绕着一些特定的主题建立，例如电信业务中的客户、收入、业务使用量等，数据仓库关注的是对决策支持有用的数据，排除无用的数据，提供特定主题的简明视图。

(2) 集成的：数据仓库的数据通常来自多个数据源，可能是多个不同厂商的关系数据库，也可能是一般的文件。使用数据清理和数据集成技术，将不同数据源的数据导入数据仓库中，确保命名规则、编码结构以及属性度量等的一致性。

(3) 时变的：数据仓库从历史的角度提供信息。数据仓库中的关键结构，隐式或显式地包含有时间元素。

(4) 非易失的：数据仓库在物理上分离存放数据，这些数据源自操作环境下的应用数据。通常数据仓库需要两种数据访问：数据的初始装入和数据访问，数据仓库与面向生产的操作型数据库相分离，不需要事务处理、恢复和并发控制机制。

### 2.3.2 数据挖掘与数据仓库的关系

大部分的情况下，数据挖掘要在数据仓库的基础上实现，先把数据从数据源（各种操作型数据库及手工录入数据）加载到数据仓库中，再将数据从数据仓库提取到专门的数据挖掘库或者数据集市中。在数据仓库的基础上进行数据挖掘有很多好处，原因是数据仓库的数据清理和数据挖掘的数据清理差不多，如果数据在导入数据仓库的过程中已经进行了清理，那么在数据挖掘过程中就可以省略这一步骤。

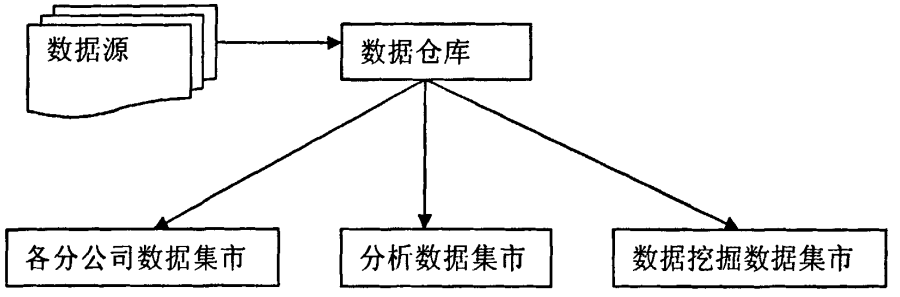


图 2-2 数据挖掘库从数据仓库中得出

数据挖掘库可以是数据仓库上的一个逻辑子集，而不一定非要是物理上单独的数据库。如果数据仓库的资源比较紧张，从挖掘效率的角度考虑，建议最好还是单独建立一个数据挖掘库。

当然数据挖掘也不是非要经过建立数据仓库这一阶段，数据仓库不是必需的。建立一个巨大的数据仓库，把各种数据源的数据整合到一起，解决所有的数据一致性问题，并把所有的数据导到数据仓库中，是一项巨大的工程，需要花费巨大的人力、物力、财力。一个便于实现的方法是把操作型数据库的数据导到一个只读数据库中，就把它当成数据集市或数据挖掘库，然后在上面进行数据挖掘工作。

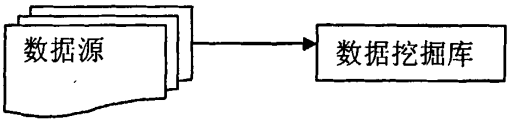


图 2-3 数据挖掘库从操作型数据库中得出

## 2.4 数据挖掘和 OLAP 的关系

### 2.4.1 什么是 OLAP

OLAP[6] (On-Line Analytical Processing，联机分析处理)和数据仓库都是决策支持领域的一部分。OLAP 是数据仓库系统的主要应用，支持复杂的分析操作，侧重于决策支持，提供直观易懂的查询结果。

OLAP 能够从多种角度对从原始数据中转化出来的、能够真正为用户所理解的、并真实反映企业经营维度特性的信息进行快速、一致、交互地存取，技术核心是“维”这个概念。

### 2.4.2 OLAP 和传统报表工具的区别

传统的查询和报表工具只能告诉你数据库中有什么，OLAP 则进一步告诉你下一步会怎么样、以及如果采取这样的措施又会怎么样。OLAP 分析过程本质上是一个演绎推理的过程。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成立。比如，分析人员想找出是什么原因导致电信用户欠费，他可能会先假设低收入的用户容易产生欠费，然后用 OLAP 来验证这个假设。可能情况并不是他假设的这样，因为低收入的用户相应的消费额也较低。那么这个假设就没有被证实，他就可能转而去分析那些高收入的用户。一直到找到他想要的结果或者放弃[7]。

### 2.4.3 数据挖掘和 OLAP 的区别

OLAP 是基于“维”的分析，如果在维度较少的情况下，分析人员可以通过 OLAP 工具简明直观地看到分析结果，但如果分析的变量（即维）达到几十个甚至上百个，那么再用 OLAP 手动分析验证这些假设就变得非常困难。

数据挖掘和 OLAP 不同的地方是，数据挖掘不是用来验证分析人员的某个假定是否成立，而是在数据库中自己寻找模型。他在本质上是一个归纳的过程。比如分析人员想找到哪些客户容易产生欠费，数据挖掘工具可以帮他归纳出产生欠费的客户群体的特征，例如收入特征、信用度特征，还有可能是一些分析人员没有注意到的因素，比如说年龄。OLAP 侧重于与用户的交互、快速的响应以及提供数据的多维视图，而数据挖掘则注重自动发现隐藏在数据中的模式和有用的信息，用户可以指导这一过程[7]。

数据挖掘和 OLAP 有一定的互补性。例如，通过定义合适的“维”（更进一步，通过确定在维中如何断开连续值），数据挖掘能够帮助 OLAP 建立更好的立方体。而 OLAP 提供了强大的可视化能力，可以帮助用户更好地理解数据挖掘的结果，如聚类和神经网络。联合使用 OLAP 和数据挖掘，二者优势互补，为数据开发提供更多的机会[8]。

## 2.5 数据挖掘过程模型 CRISP-DM

业界流行的数据挖掘过程模型很多，典型的如：SPSS 的 CRISP-DM，SAS 的 SEMMA 等等，

其中CRISP-DM是事实上的工业标准。

CRISP-DM[9]模型定义的一个数据挖掘项目的生命周期包括六个阶段。如图2-4所示。各个阶段的顺序不是僵硬不变的，有时需要在不同阶段之间向前和向后移动。这取决于每一个阶段的成果和下一个阶段的具体任务。

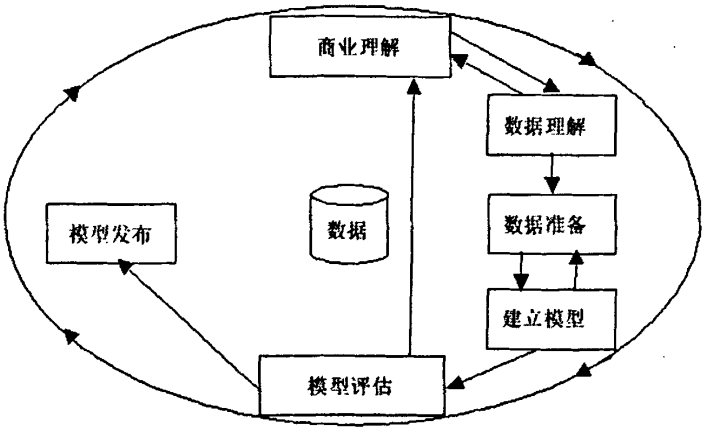


图 2-4 CRISP-DM模型

(1) 商业理解

这一初始阶段集中在从商业角度理解项目的目标和要求，然后把理解转化为数据挖掘问题的定义和一个旨在实现目标的初步计划。

(2) 数据理解

数据理解阶段开始于原始数据的收集，然后是熟悉数据，表明数据质量问题，探索对数据的初步理解，发觉有趣的子集以形成对隐藏信息的假设。

(3) 数据准备

数据准备阶段包括所有从原始的未加工的数据构造最终数据集的活动（这些数据集指将要嵌入建模工具中的数据）。数据准备任务可能要被实施多次，而且没有任何规定的顺序。这些任务包括表格、记录和属性选择以及按照建模工具的要求，对数据的转换和清洗。

(4) 建模

在此阶段，主要是选择和应用各种建模技术，同时对它们的参数进行校准以达到最优值。通常对于同一个数据挖掘问题类型，会有多种模型技术。一些技术对数据格式有特殊的要求。因此常常需要返回到数据准备阶段。

(5) 评估

进入项目中的这个阶段时，你已经建立了一个模型（或者多个），从数据分析的角度来看，该模型似乎有很高的质量。在模型最后发布前，有一点是很重要的一更为

彻底地评估模型和检查建立模型的各个步骤，从而确保它真正达到商业目标。此阶段关键目的是决定是否存在一些重要的商业问题仍未得到充分的考虑。关于数据挖掘结果的使用决定应该在此阶段结束时确定下来。

(6) 发布

模型的创建通常不是项目的结尾。即使模型的目的是增加对数据的了解，所获得的了解也需要进行组织并以一种客户能够使用的方式呈现出来。

## 2.6 数据挖掘过程中的数据预处理

在实际应用系统中收集到的原始数据往往是杂乱、重复和不完整的，因此数据预处理是数据挖掘中的一个重要环节。数据预处理应该包括以下几方面的功能：[10]

(1) 数据集成：数据集成主要是将多文件或多数据库运行环境中的异构数据进行合并处理，解决语义的模型性。该部分主要涉及数据的选择、数据的冲突问题以及不一致数据的处理问题。

(2) 数据清洗：数据清洗要去除源数据集中的噪声数据和无关数据，处理遗漏数据和清洗脏数据，去除空白数据域和知识背景上的白噪声，考虑时间顺序和数据变化等。主要包括重复数据处理和缺值数据处理，并完成一些数据类型的转换。

(3) 数据变换：数据变换主要是找到数据的特征表示，用维变换或转换方式减少有效变量的数目或找到数据的不变式，包括格式化、归纳、切换、旋转和投影等操作。

(4) 数据简化：数据简化是在对发现任务和数据本身内容理解的基础上，寻找依赖于发现目标的表达数据的有用特征，以缩减数据规模，从而在尽可能保持数据原貌的前提下最大限度地精简数据量。它主要有两个途径：属性选择和数据抽样，分别针对数据库中的属性和记录。

## 2.7 数据挖掘常用算法

(1) 决策树

决策树提供了一种展示类似在什么条件下会得到什么值这类规则的方法。决策树是一个类似流程图的树型结构，建立决策树的过程，即树的生长过程是不断的把数据进行切分的过程，每次切分对应一个问题，也对应着一个节点。对每个切分都要求分成的组之间的“差异”最大。各种决策树算法之间的主要区别就是对这个“差异”衡量方式的区别。[7]

决策树的优点是效率高、容易理解，并且很擅长处理非数值型数据，这与神经网络只能处理数值型数据比较起来，免去了很多数据预处理的工作。

(2) 神经网络

就是一组相互连接的输入输出单元，这些单元之间都关联一个权重。在网络学习阶段，通过调整权重来实现输入样本与其相应类别的对应。神经网络可以很容易的解决具有上百个参数的问题。神经网络常用于两类问题：分类和回归。在结构上，可以把一个神经网络划分为输入层、输出层和隐含层。输入层的每个节点对应一个个的预测变量。输出层的节点对应目标变量，可有多个。在输入层和输出层之间是隐含层（对神经网络使用者来说不可见），隐含层的层数和每层节点的个数决定了神经网络的复杂度。调整节点间连接的权重就是在建立（也称训练）神经网络时要做的工作。[11]

(3) 遗传算法

基于进化理论，并采用遗传结合、遗传变异、以及自然选择等设计方法的优化技术。遗传算法模拟进化/适者生存的过程，以随机的形式将最适合特定目标函数的种群通过重组产生新的一代，在进化过程中通过选择、重组和突变逐渐产生优化的问题解决方案。它通过选择、交叉和变异等进化概念，产生出解决问题的新方法和策略。选择是指挑出好的解决方案，交叉是将各个好的方案中的部分进行组合连接，而变异则是随机地改变解决方案的某些部分，这样当提供了一系列可能的解决方案后，遗传算法就可以得出最优解决方案。[12]

(4) 近邻算法：将数据集合中每一个记录进行分类的方法。

(5) 规则推导：从统计意义上对数据中的“如果-那么”规则进行寻找和推导。

## 2.8 数据挖掘技术的分类

数据挖掘技术基本上分为两大类：描述型数据挖掘和预测型数据挖掘，下面就这两种挖掘类型进行说明[4] [7]。

### 2.8.1 描述型数据挖掘

描述型数据挖掘是用来了解数据中潜在的规律。主要包括：

(1) 统计和可视化

统计：了解自己的数据的最基本的方法就是计算各种统计变量，如平均值、方差、标准差等。尽管统计分析需要专业的技能，但它却是所有数据挖掘技术中发展最成熟

同时也是最容易理解的一种技术[13]。

可视化：帮助快速地、直观地分析数据。

(2) 聚类（分群）

聚类是把整个数据集划分成不同的群组。它的目的是要群和群之间的差别很明显，而同一个群内的数据尽量相似。

聚类与分类是不同的，聚类在开始之前并不知道要把数据集分成几类，也不知道依据哪些变量来分，而分类之前是知道要分成哪几类的，每个类的特征是什么。

通过聚类得到的分群结果需要有一个很熟悉业务的人来解释这些分群的意义，对每个具体的分群给出特征描述。聚类是一个反复的过程，很多时候一次聚类的结果对业务来说可能并不好，这时就需要增加或者删除变量以影响分群的方式，最终得到理想的结果。神经元网络和K-均值是比较常用的聚类方法。

(3) 关联分析

关联规则是寻找数据库中值的相关性。关联规则最早提出的动机是针对购物篮分析问题提出的，其目的是为了发现交易数据库中不同商品之间的联系规则。

### 2.8.2 预言型数据挖掘

预言型数据挖掘是用历史来预测将来。主要包括：

(1) 分类挖掘

按照分类对象的属性分门别类加以定义，建立分组。换句话说，分类要解决的问题是为一个事件或者对象归类。在实际使用中，既可以用分类来分析已有的数据，也可以用它来预测未来的数据。例如在电信业务应用中，用分类来区分不同属性的客户，预测哪些客户可能会使用电信新业务等等。

(2) 回归挖掘

回归是通过具有已知值的变量来预测其他变量的值，如果此变量随事件变化，可成为时间序列预测。在最简单的情况下，回归采用的是像线性回归这样的标准统计技术。但在大多数现实世界中，很多问题是无法用简单的线性回归来预测的。如电信业务的价格、使用量，很难找到简单有效的方法来预测，因为要描述这些事件的变化需要数以百计的变量，且这些变量本身往往是非线性的。为此，人们又发明了许多新的手段来试图解决这个问题，如逻辑回归、决策树、神经网络等。

(3) 时序挖掘

时间序列是用变量过去的值来预测未来的值。与回归一样，它也是用已知的值来预测未来的值，区别在于这些值的变量所处的时间不同，存在时间上的先后关系。时间序列采用的方法一般是在连续的时间流中截取一个时间窗口(一个时间段)，窗口内的数据作为一个数据单元，然后让这个时间窗口在时间流上滑动，以获得建立模型所需要的训练集。比如用前六天的数据来预测第七天的值，这样就可以建立一个区间大小为七的时间窗口。

## 2.9 数据挖掘热点研究方向

就目前来看，数据挖掘将来的热点包括：文本挖掘、WEB 挖掘、生物信息或基因的数据挖掘、多媒体挖掘等。下面就这几个方面加以简单介绍。

(1) 文本挖掘[14]

文本挖掘是从大量文本数据中提取以前未知的、有用的、可理解的、可操作的知识的过程。文本数据包括技术报告、文本集、新闻、电子邮件、网页、用户手册等。文本挖掘对单个文本或文本集（如 Web 搜索中返回的结果集）进行分析，从中提取概念，并按照指定的方案组织、概括文本，发现文本集中重要的主题。它除了从文本中提取关键词外，还要提取事实、作者的意图、期望和主张等。这些知识对许多应用目标，如市场营销、趋势分析、需求处理等，都是很有用的。

相对于一般的数据挖掘，文本挖掘面临的主要问题在于挖掘的对象是半结构化或非结构化的，而且自然语言文本中包含多层次的歧义（如词汇、句法、语义、语用等）等。

(2) Web 挖掘[14]

Web 挖掘是从 WWW 的资源和行为中抽取感兴趣的、有用的模式和隐含的信息，一般可以分为三类：Web 内容挖掘、Web 结构挖掘和 Web 应用挖掘。

- Web 内容挖掘：用来提取文字、图片或者其他组成网页内容成分的信息和知识。
- Web 结构挖掘：用来提取网络的拓扑信息，即网页之间的链接信息。从 WWW 的组织结构和链接关系中挖掘知识。
- Web 应用挖掘：用来提取关于客户如何运用浏览器浏览和使用页面链接的信息。从 Web 的访问记录中抽取感兴趣的模式。

(3) 生物信息或基因的数据挖掘

生物信息或基因数据挖掘则完全属于另外一个领域，在商业上很难讲有多大的价

值，但对于人类却受益非浅。例如，基因的组合千变万化，得某种病的人的基因和正常人的基因到底差别多大？能否找出其中不同的地方，进而对其不同之处加以改变，使之成为正常基因？这都需要数据挖掘技术的支持。

对于生物信息或基因的数据挖掘和通常的数据挖掘相比，无论在数据的复杂程度、数据量还有分析和建立模型的算法而言，都要复杂得多。从分析算法上讲，更需要一些新的和好的算法。现在很多厂商正在致力于这方面的研究。但就技术和软件而言，还远没有达到成熟的地步。[15]

(4) 多媒体挖掘

多媒体挖掘就是从大量多媒体数据集中，通过综合分析视听特性和语义，发现隐含的、有效的、有价值的、可理解的模式，得出事件的趋向和关联，为用户提供问题求解层次的决策支持能力。[16]

## 2.10 本章小结

本章对数据挖掘技术作了简要的概述，是全文的理论基础部分。其中涉及到数据挖掘的概念、挖掘过程模型、数据预处理、数据挖掘分类、常用算法、热点研究方向，以及数据仓库介绍、OLAP介绍、数据挖掘和数据仓库、数据挖掘和OLAP之间的关系等。

# 第3章　数据挖掘中的聚类算法

## 3.1　聚类的定义

聚类[17][18]（clustering）是一个将数据集划分成若干组（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相似度较低。相似或不相似的描述是基于数据对象属性的取值来确定的，通常是利用各对象间的距离来进行表示。

## 3.2　聚类算法的典型要求

聚类分析是一个富有挑战的研究领域，每一个应用都有自己独特的要求。以下就是对数据挖掘中的聚类分析的一些典型要求[17]。

(1) 可扩展性。许多聚类算法在小数据集（少于200个数据对象）时可以工作地很好，随着数据对象的增加，这些聚类算法的处理能力就会下降；但一个大的数据库可能会包含数以百万的对象。利用采样方法进行聚类分析可能得到一个有偏差的结果，这时就需要可扩展的聚类分析算法。

(2) 处理不同类型属性的能力。许多算法是针对基于区间的数值属性而设计的。但是有些应用需要对其它类型的数据，如：二值类型、符号类型、顺序类型，或这些数据类型的组合进行分析。

(3) 发现任意形状的聚类。许多聚类算法是根据欧氏距离和Manhattan距离来进行聚类的。基于这类距离的聚类方法一般只能发现具有类似大小和密度的圆形或球状聚类。而实际上一个聚类是可以具有任意形状的，因此设计出能够发现任意形状类集的聚类算法是非常重要的。

(4) 需要（由用户）决定的输入参数最少。许多聚类算法需要用户输入聚类分析中所需要的一些参数（如：期望所获聚类的个数）。而聚类结果通常都与输入参数密切相关；而这些参数常常也很难决定，特别是包含高维对象的数据集。这不仅构成了用户的负担；也使得聚类质量难以控制。

(5) 处理噪声数据的能力。大多数现实世界的数据库均包含异常数据、不明数据、数据丢失和噪声数据，有些聚类算法对这样的数据非常敏感并会导致获得质量较差

的聚类结果。

(6) 对输入记录的顺序不敏感。一些聚类算法对输入数据的顺序敏感，也就是不同的数据输入顺序会导致获得非常不同的结果。因此设计对输入数据顺序不敏感的聚类算法也是非常重要的。

(7) 处理高维对象的问题。一个数据库或一个数据仓库或许包含若干维或属性。许多聚类算法在处理低维数据时（仅包含二到三个维）时表现很好。人的视觉也可以帮助判断多至三维的数据聚类分析质量。然而设计对高维空间中的数据对象，特别是对高维空间稀疏和怪异分布的数据对象，能进行较好聚类分析的聚类算法已成为聚类研究中的一项挑战。

(8) 基于约束的聚类。现实世界中的应用可能需要在各种约束之下进行聚类分析。假设需要在一个城市中确定一些新加油站的位置，就需要考虑诸如：城市中的河流、高速路，以及每个区域的客户需求等约束情况下居民住地的聚类分析。设计能够发现满足特定约束条件且具有较好聚类质量的聚类算法也是一个重要聚类研究任务。

(9) 可解释性和可用性。用户往往希望聚类结果是可理解的、可解释的，以及可用的。这就需要聚类分析要与特定的解释和应用联系在一起。因此研究一个应用的目标是如何影响聚类方法选择也是非常重要的。

## 3.3 主要的聚类方法

(1) 划分方法[19]

给定一个包含 n 个对象的数据集，划分方法就是将数据集划分为 k 个子集。其中每个子集都代表一个聚类（k<=n）。也就是说把数据分成 k 组，这些组要满足以下要求：每组应至少包含一个对象；且每个对象必须只能属于某一组。需要注意的是后一个要求在一些模糊划分方法中可以放宽。

(2) 层次方法[20]

层次方法就是通过分解所给定的数据对象集来创建一个层次.根据层次分解形成的方式，可以将层次方法分为自下而上和自上而下两种类型。自下而上的层次方法从每个对象均为一个单独的组开始，逐步将这些（对象）组进行合并，直到组合并在层次顶端或满足终止条件为止。自上而下层次方法从所有对象均属于一个组开始，逐步将组划分成为更小的组，直到每个对象构成一组或满足终止条

件为止。大多数层次聚类采用自下而上的方法[21]。

(3) 密度方法[20]

基于密度概念的聚类方法实际上就是不断增长所获得的聚类直到“邻近”(数据对象或点)密度超过一定的阀值(如：一个聚类中的点数，或一个给定半径内必须包含至少的点数)为止。这种方法可以用于消除数据中的噪声(异常数据)，以及帮助发现任意形状的聚类。

(4) 网格方法[22]

基于网格方法将对象空间划分为有限数目的单元以形成网格结构。所有聚类操作均是在这一网格上进行的。这种方法主要优点就是处理时间由于与对象个数无关而仅与划分对象空间的网格数相关，从而显得相对较快。

(5) 模型方法[23]

基于模型方法就是为每个聚类假设一个模型，然后再去发现符合相应模型的数据对象。一个基于模型的算法可以通过构造一个描述数据点空间分布的密度函数来确定具体聚类。它根据标准统计方法并考虑到噪声或异常数据，可以自动确定聚类个数，因此它可以产生很鲁棒的聚类算法。

目前，虽然聚类算法已被广泛深入地研究，产生了许多不同的适用于数据挖掘的聚类算法，但这些算法仅适用于特定的问题及用户。多数电信行业软件针对客户细分都采用划分方法，而层次方法、密度方法、模型方法通常使用较少。

## 3.4　K-means 算法简介

### 3.4.1　K-means 算法流程

k-means 算法由 MacQueen[24][25] 首先提出，是解决聚类问题的经典算法。K-means 是一种无监督的学习方法，它不是去预测某一结果，而是从给定的属性中发现特征。

k-means 算法叙述如下：

算法：根据聚类中的均值进行聚类划分的 k-means 算法。

输入：聚类个数 k, 以及包含 n 个数据对象的数据集。

输出：满足方差最小标准的 k 个聚类。

处理流程：

1) 从 n 个数据对象中任意选择 k 个对象作为初始聚类中心；

2) 循环3)到5)直到错误函数E不再明显改变或者每个聚类不再发生变化为止；

3) 根据每个聚类对象的均值（中心对象），计算每个对象与这些中心对象的距离；并根据最小距离重新对相应对象进行划分；

4) 重新计算每个（有变化）聚类的均值（中心对象）；

5) 计算错误函数E。

如上算法所示，k-means算法要求输入参数k；然后将n个数据对象划分成为k个聚类以便使得所获得的聚类满足以下条件：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。聚类相似度是以对象之间的距离来表示，距离越小代表对象之间的相似度越高。由于欧氏距离的直观性，本文采用欧氏距离来衡量对象之间的相似度。对象$X(x_1,x_2,...,x_n)$，$Y(y_1,y_2,...,y_n)$之间的欧氏距离按下列的公式计算：

$$d(X,Y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_n-y_n)^2} \qquad (3\text{-}1)$$

k-means算法的工作过程说明如下：首先从n个数据对象中任意选择k个对象作为初始聚类中心；而对于所剩下的其他对象，则根据它们与这些聚类中心的相似度（距离），分别将它们分配给与其最相似的（聚类中心所代表的）聚类；然后再计算每个所获新聚类的聚类中心（该聚类中所有对象的均值）；不断重复这个过程直到标准测度函数开始收敛到某个允许条件为止。一般都采用均方差作为标准测度函数，具体定义如下：

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-m_i|^2 \qquad (3\text{-}2)$$

其中E为数据集中所有对象的均方差之和；p为代表对象的空间中的一个点；$m_i$为聚类$C_i$的均值（p和$m_i$均是多维的）。该公式所示的聚类标准旨在使所获得的k个聚类具有以下特点：各聚类本身尽可能的紧凑，而各聚类之间尽可能的分开。

从以上k-means算法的流程可以看出，k-means算法的计算复杂度可以分为以下几个部分（其中：n为对象个数，k为聚类个数，d为维度个数）：

(1) 流程中的第3)步骤，所需要的时间复杂度为O(nkd)。

(2) 流程中的第4)步骤，更新聚类中心所需要的时间复杂度为O(nd)。

(3) 流程中的第5)步骤，计算错误函数所需要的时间复杂度为O(nd)。

因此，整个算法总的复杂度为O(nkd)[26]，需要循环的次数取决于聚类个数以及输入数据的数量和维数。k-means算法的时间复杂度可以精确地计算出来，这在处理大数据量的聚类要求时是非常重要的。

### 3.4.2 k-means 算法的优缺点

k-means 算法的优点是简单、容易理解、运算速度较快。而 k-means 算法的缺点也很容易看出：

(1) 用户必须事先指定聚类个数 k。不同的聚类个数会导致差异很大的聚类结果。如果对业务和数据理解不够，很难事先选择合适的聚类个数。

(2) k-means 算法的聚类质量依赖于初始聚类中心的选择，它的执行结果与数据的输入次序有关[27]。

(3) k-means 算法只适用于聚类均值有意义的情况。因此在某些应用中，诸如：数据集包含符号属性时，直接应用 k-means 算法就有困难了。

(4) k-means 算法不适合用于发现非凸形状的聚类，或具有各种不同大小的聚类。

(5) k-means 算法对噪声和异常数据比较敏感，因为这些数据可能会影响到各聚类的均值（计算结果）。

### 3.4.3 k-means 算法的变化版本

k-means 算法还有一些变化（版本）[17]。它们主要在初始 k 个聚类中心的选择、差异程度计算、聚类均值的计算方法等方面有所不同。一个常常有助于获得好的结果的策略就是首先应用自下而上的层次算法来获得聚类个数，并发现初始分类；然后再应用循环再定位（聚类方法）来帮助改进分类结果[28]。例如两步聚类法：这种方法首先需要确定一个最大群数（比如说 n），并把数据按照一定的规则分为 n 群，这是该方法的第一步。接着按照一定的规则把 n 群中最接近的群进行归并，当达到一定的标准时，这种归并停止，归并停止时的实际的群数，就是该算法最终确定的聚类群数，这是第二步。两步聚类法一个显著优点就是它可以不需要事先指定聚类群数，而可以根据数据结构本身自动确定应该把数据分为多少个群。

另一个 k-means 算法的变化版本就是 k-modes 算法。该算法通过模来替换聚类均值、采用新差异性计算方法来处理符号量、以及利用基于频率对各聚类模进行更新方法，从而将 k-means 算法的应用范围从数值量扩展到符号量。将 k-means 算法和 k-modes 算法结合到一起，就可以对采用数值量和符号量描述的对象进行聚类分析，从而构成了 k-prototypes 算法。

而 EM(期望最大值)算法又从多个方面对 k-means 算法进行扩展。其中包括：它根

据描述聚类所属程度的概率权值，将每个对象归类为一个聚类，不是将一个对象仅归类为一个聚类（所拥有）；也就是说在个聚类之间的边界并不是非常严格。因此可以根据概率权值计算相应的聚类均值。

此外通过识别数据中所存在的三种类型区域，即可压缩区域、必须存入内存区域、可以丢弃区域，来改善 k-means 算法的可扩展性。若一个对象归属某个聚类的隶属值是不确定的，那它就是可丢弃的；若一个对象不是可丢弃的且属于一个更紧密的子聚类，那么它就是可压缩的。利用一个被称为聚类特征的数据结构来对所压缩或所丢弃数据进行综合（summarize），若一个对象既不是可丢弃的也不是可压缩的，那它就需要保持在内存里（在聚类过程中）。为实现可扩展性，循环聚类算法仅需对可压缩和可丢弃数据的聚类特征，以及保持在内存中的对象进行分析处理即可。

## 3.5 K-means 算法的改进

### 3.5.1 K-means 算法的优化方向

当前，k-means 算法的优化主要在三个方向：

(1) 聚类个数（k）值的选定。K-means 算法要求把聚类个数值作为整个算法的输入参数，不同的聚类个数对聚类的效果影响很大。在很多复杂应用中，影响聚类的属性值可能有几十个甚至上百个之多，很难在一开始就选择一个合适的聚类个数值。目前的很多改进方法是在运用 k-means 聚类前，先用别的方法（如层次聚类）得到合适的聚类个数值，然后再用 k-means 算法去调整这些聚类。

(2) 算法初始聚类中心的选择。初始聚类中心对算法的运行效率有较大影响。K-means 算法采用的方法是随机地选择 k 个点作为初始的聚类中心，因此整个算法的效率也就变得随机。目前已经有很多学者对初始聚类中心的选择提出改进想法。

(3) 算法本身的改进，主要以提高效率、降低计算复杂度为目标。K-means 算法中每一次重新聚类都涉及到大量的数据对象和质心点之间的距离计算，当需要聚类的数据集非常大，比如上百万、上千万时，k-means 算法的效率就变得不能接受。

根据以上三个优化方向，本文选择如下改进策略：

(1) 对于优化方向 1，电信运营商在客户细分方面已经有比较成熟的 k 经验值可以参考，聚类个数一般选择在[7-2, 7+2]之间。本文研究的客户细分取聚类个数为 5。

(2) 对于优化方向 2，根据随机函数的分布知识，聚类的数据应主要分布在所有数据

的均值和标准差所构成的区间$(\mu-\sigma,\mu+\sigma)$之间，将这个区间进行k+1等分，得到的k个等分点作为初始的聚类中心。多维的情况也类似可得。采用这种初始聚类中心选择方法，不仅可以得到分布比较合理的初始聚类中心，提高算法效率，还可以克服随机选取初始聚类中心的不确定性。

(3) 对于优化方向3，主要考虑如何减少计算复杂度，提高算法效率。有两种方法可以用来减少k-means算法的计算复杂度。1)一种是利用上一次循环计算的信息来减少距离计算的数量。这种方法利用了对象对聚类的分配在k-means算法第一个循环执行几次后相对变化很少的事实，使用一个启发式来判断一个对象q所属的最近聚类中心是否已经改变，若没有改变，就不需要进一步的距离计算。这种方法还利用了另外一个事实：对于连续的循环（尤其是在几个循环之后），聚类的质心点移动较少。2)另一种用来减少算法计算复杂度的思想是：将原型向量组织在一个恰当的数据结构中，其结果是对给定的一个原型，发现最近的原型变得更富有效率。使用这种方法的距离计算的数量与每一个循环的$n\times f(k,d)$成正比（一般情况下有$k<<n, d<<n$）。对于许多应用来说，向量的数量及原型向量的数量都是固定的，对于一个给定的输入测试模式，构造优化的数据结构可以发现最近的向量[29]。本文采用后一种改进思想。

### 3.5.2　K-D树介绍

K-D树是把二叉搜索树推广到多维数据空间的一种主存数据结构，即K-D树是一棵多维二叉搜索树[30]，其中k表示搜索空间的维数，它的内部节点根据一个相关联的维度a和一个值V，它将数据点分成两个部分：a维值小于V的部分和a维值大于等于V的部分，这两部分分别构成该节点的左右子树[31]。不断对K-D树的内部节点进行分裂，直到每个节点只包含一个数据对象或者K-D树的深度满足要求为止。

一般情况下，所有的维度在K-D树的各层节点划分中被循环使用，因此树的不同层上的属性是不同的；但同一个层上依据相同的维度进行划分。例如：第一层（根节点）的划分依据第一个维度，第二层所有节点的划分根据第二个维度，以此类推在各个维度之间反复地进行划分。

K-D树的构建过程也就是不断地划分多维空间的过程。例如[32]：给定一组取值范围在100×100的二维数据集，假设构成的K-D树如图3-1所示：

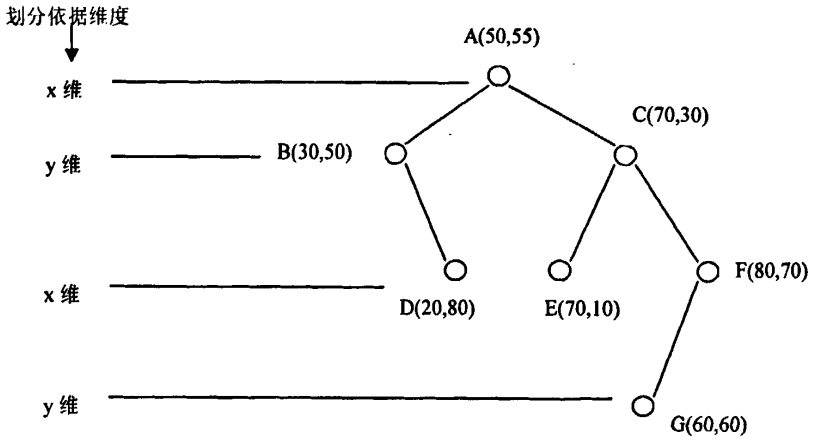


图 3-1 K-D 树结构示例

则 K-D 树的每一个内部节点的分裂对二维空间的划分过程如图 3-2 所示。

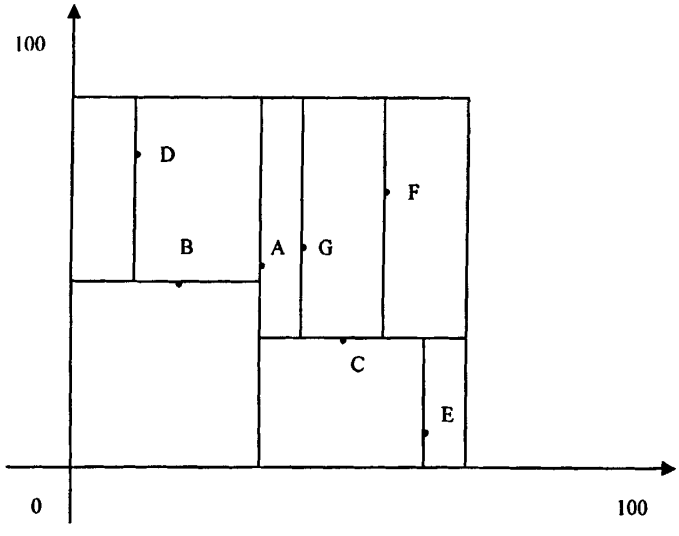


图 3-2 K-D 树对二维空间的划分

从图 3-2 中可以看出 K-D 树对空间划分的特点：

(1) 根节点将空间划分成两部分，其子节点进一步将子空间划分成更小的部分，以此类推进行空间划分。

(2) 子节点的划分线不会穿过其父节点的划分线。

(3) K-D 树中的内部节点最终将空间划分成矩形，所有的数据对象则分布在这些矩形空间中。

需要注意的是，在选择维度 a 的值 V 进行划分时，应尽可能地使被划分的节点大约一半的数据落在左子树，另一半的数据落在右子树，以保持 K-D 树的平衡。

### 3.5.3 基于K-D树的改进k-means算法

借助K-D树这一数据结构来改进k-means算法，核心思想在于把K-D树的一个内部节点所代表的子空间作为一个整体，来考察这个整体到哪个（哪些）候选者（聚类中心）距离更接近，而不是像k-means算法那样考察每个对象到候选者的距离远近。具体如下：1)将所有的数据对象组织在K-D树节点内。2)将初始聚类中心组织在一个候选集内。3)对于根节点来说，候选集内所有的原型都是潜在候选者。4)对于非根节点来说，可以通过采用简单的物理度量方面的限制对候选集进行修剪。5)这种修剪方法一直递归地应用直到所有节点满足：候选集只剩下一个候选者或者节点是叶节点时结束。如果某一节点的候选集只剩下一个候选者，那么该节点所代表的子空间内所有的数据对象都属于这个唯一候选者所代表的聚类。

由于K-D树内的每一个节点在遍历时都需要进行候选集的修剪计算，因此改进算法的效率和修剪算法的优劣紧密相关。修剪算法可以采用下面的策略：

(1) 对于每一个候选者，找出到节点所代表的子空间内任意一点的最大距离与最小距离；

(2) 在这些最大距离中找出最小的那个距离，称为MinMax；

(3) 修剪掉所有的最小距离大于MinMax的候选者。

这种策略可以保证当一个候选者到一个给定的子空间的距离比其他的候选者都近时不会被修剪掉。

在更新质心点的坐标以及计算误差函数时，需要用到每个对象在各个维度上的信息。为了防止对这些维度信息的重复访问和计算，在建立K-D树的过程中，可以在每个节点存储以下信息：

(1) 该节点所包含的对象个数（m）。

(2) 对象各个维度上的线性和（LS），即$\sum_{i=1}^{m} p_i$。

(3) 对象各个维度上的平方和（SS），即$\sum_{i=1}^{m} p_i^2$。

设维度为d的K-D树的深度为D，则在每一个节点维护上面的信息所需要的额外的时间和空间开销为$O(nd)^2$；在第D层，中心点的计算时间复杂度为$O(nD)$，这些中心点被用来分裂树的内部节点。因此，当在给定的层上每一个内部节点代表相同数目

的数据对象时，建立这样的K-D树所需要的整个时间开销是$O(n(d+D))^3$。[33]

同时，在构建K-D树时，存在以下两个影响整个树结构的不同选择：

(1) 用于分支的维的选择。第一种选择是依次循环地利用所有维进行节点划分。同一层上的节点依据相同的维划分，而不同层上划分依据的维各不相同。当自上而下构建K-D树时，对不同的层选择维是以循环的方式进行的。第二种选择是依据具有最长长度的维进行划分。在实际应用中，数据对象的数据在每个维上的分布各不相同，有的维度的数据差异性较大、区分度较高，而有的维度的数据可能大部分相同甚至全部一致。如果循环地应用这些维度对树的内部节点进行划分，当遇到数据区分度较差的维度时，会导致树的层数增加了但并没有很合理地划分空间，并且对保持K-D树的平衡不利。而具有最长长度的维通常也具有较高的数据区分度，依据这样的维划分空间是比较合适的。因此本文根据第二种方法选择用于节点划分的维。

(2) 分支维的分支点的选择也有不同的方法：一种是选择中心分支点，按照分支维的宽度分为两个部分；另一种是选择中间分支点，按照节点包含对象的数量进行等分。很显然基于中心的方法在应用中更容易实现。

本文基于K-D树的改进的k-means算法描述如下：

1) 构建K-D树。从根节点开始每层选择具有最长长度的维进行节点划分，以数据对象在该维上的中心点作为划分点。划分过程一直进行到叶子节点所包含的对象数量小于一定值为止。

2) 根据全部数据对象的均值和标准差选择初始聚类中心。

3) 遍历K-D树，在每个节点上运行修剪算法。如果节点找到唯一候选者，则认为该节点包含的所有对象均属于这个唯一候选者所代表的聚类并更新聚类集中的聚类中心；如果遍历到叶子节点但候选集中仍然有多于一个候选者，则在这个叶子节点的所有对象中应用标准的k-means算法进行聚类。

4) 考察误差函数，如果还没达到要求，重复执行步骤3)，直到满足误差要求。

## 3.6　本章小结

本章首先介绍聚类的定义、要求和主要的聚类方法。由于本文运用到聚类算法中的

k-means 算法，因此本章对标准的 k-means 算法做了详细的介绍。在此基础上，结合对 K-D 树这种数据结构的分析，引出利用 K-D 树来改进 k-means 算法的思想，并详细叙述这种改进算法的特点、过程及注意事项。另外，本章提出的对 k-means 算法的改进还包括在初始聚类中心选择上的改进。

# 第4章 电信经营分析系统概述

## 4.1 电信经营分析的建设背景

随着电信市场的开放以及3G时代的临近，国外的运营商对中国电信市场这块巨大的“蛋糕”虎视眈眈，国内的各大运营商则纷纷筹备全业务运营，电信市场“垄断”经营已经成为历史，竞争日益激烈。一方面，为了能够在竞争中生存和持续发展，各电信运营商都对企业的经营和管理提出了更高的要求；另一方面，电信运营商原来为支撑各种业务运营所建立的各种计算机系统，主要面向生产，分散且功能比较单一，无法全面满足企业经营管理工作的需要，突出表现在：单一系统产生的报表无法满足企业管理的要求；相对固定的报表不能跟上市场形势的变化；庞大的数据库系统不能有效地利用以产生可以指导企业经营管理的知识。电信企业迫切需要寻找到一种新的经营管理支撑手段，使得管理人员能够及时准确地了解市场竞争、业务发展和资源使用情况，以便及时地发现问题和解决问题，这正是经营分析系统产生的原因[34]。

电信企业通过建设经营分析系统，目标是建立一个同一的数据信息平台，采用先进的数据仓库技术和分析挖掘工具，提取企业数据中的有价值信息，为企业的客户服务、市场营销等工作提供科学有效的支撑，提升企业的运营水平和竞争能力，体现以客户为中心的经营理念。简而言之就是“用数据说话，用理性决策”。

## 4.2 电信经营分析系统功能架构

经营分析系统功能架构图4-1所示[35]：

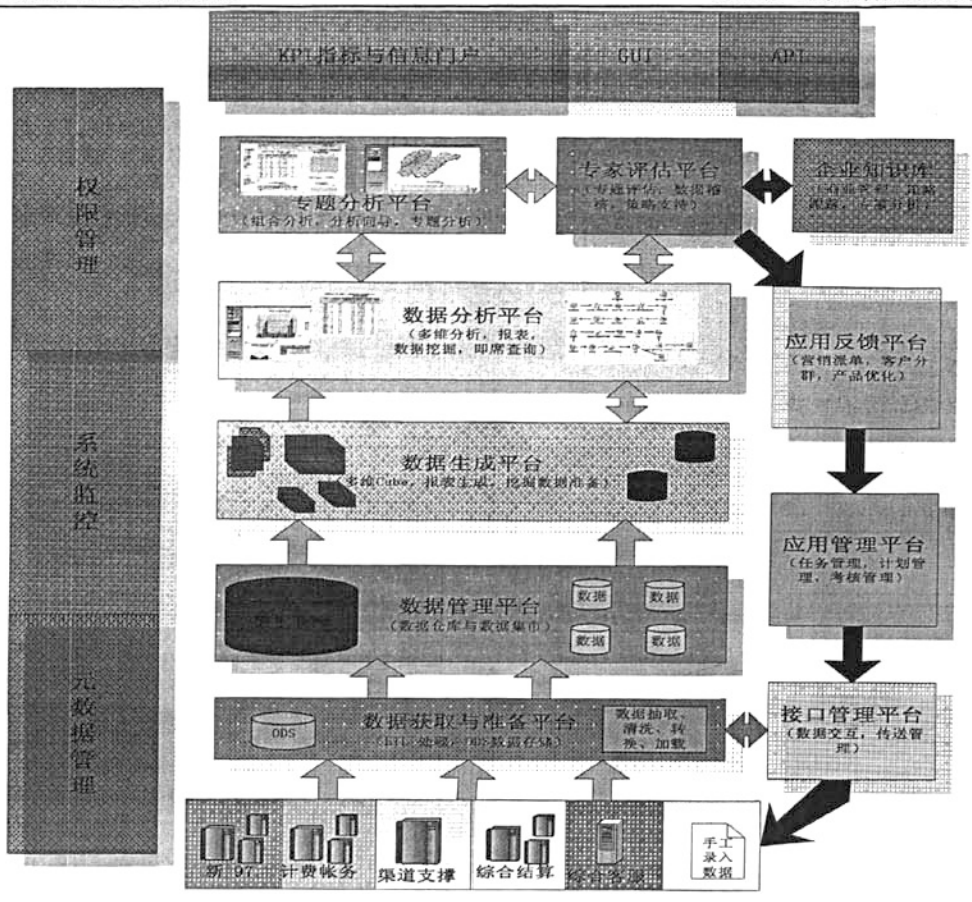


图 4-1　经营分析系统架构图

按照功能可以将经营分析系统划分为以下几个模块：

(1) 核心数据处理模块

数据处理模块是经营分析系统的主体部分。分析的基础是数据，因此全面的数据来源、优质的数据质量、科学的数据存储、高效的数据访问是经营分析系统的关键所在。该模块又进一步分为数据获取层、数据存储层、数据访问层。具体内容在下一节中介绍。

(2) 闭环反馈流程模块

经营分析是一个闭环的、可持续运作的过程，即，由管理层制定政策，各生产部门根据管理层制定的政策组织业务管理，并把相关信息反馈给管理层，管理层分析反馈的信息，进行政策的调整，从而完成企业管理的闭环化。

该模块在数据分析的基础上，通过专题分析平台和专家评估平台的分析和总结之后，制定营销方案；利用应用反馈平台实施营销派单、产品优化等；利用应用管理平台，进行营销的任务管理、计划管理和考核管理；利用接口管理平台实施数据和生产操作系统之间的交互、传送管理。在实施营销方案之后，各生产操作系统的数据又通

过核心数据处理流程到达数据分析平台，应用对比分析，可以观察到营销方案的效果，进而调整和优化营销方案，进一步促进生产。

(3) 知识管理模块

包括专题分析平台、专家评估平台、企业知识库。专题分析平台根据具体应用提供专题分析，例如客户发展分析、业务发展分析、数据业务与新业务分析等，利用分析工具从多角度直观地分析数据。专家评估平台结合数据分析、挖掘的结果，由业内专家对结果进行业务上的描述和评估，并给出相关的建议或策略，例如，对于运用聚类挖掘方法得到的客户细分结果，业务专家应能给出每个分群的大致特征，对这些特征进行进一步分析并给出营销或市场推广方面的建议。企业知识库的目的建立知识积累和共享的平台，与商业管理、策略跟踪、专家分析相关的知识和经验都可以存储在企业知识库中。

(4) 辅助管理模块

系统提供必要的管理和监控功能，实现经营分析系统各级层面的管理和监控，保障系统的稳定运行。包括元数据管理、系统监控、权限管理。其中元数据管理模块存放系统中的所有元信息，包括应用功能元信息，模板元信息，报表元信息，知识元信息，数据仓库元信息，ODS 元信息，ETL 规则等。系统监控模块实现各项统计数据的审核监控功能，系统通过多个角度对相应指标进行监控，可以发现指标数据的异常波动情况，帮助系统使用人员及时进行分析和调整。权限管理内容包含：1) 根据角色授权，实现权限、角色的有效周期维护；2) 报表查询权限（地域、时间、级别），以及报表数据范围控制；3) 授权管理采用分级授权机制。操作员只能使用上级业务主管授权使用的应用模块；4) 操作员日志的跟踪。

(5) 其他

- KPI 指标和信息门户：KPI 是企业运作的晴雨表，包含全面而灵活的主题分析和专题分析功能。
- GUI（图形用户接口）：操作直观简便，操作者只需借助鼠标点选就可以完成大部分工作。
- API 接口

## 4.3 核心数据处理流程

经营分析系统核心数据处理流程分为三个层次：数据获取层、数据存储层、数据访问

层。其中数据获取层又分为数据来源、数据抽取/转换/加载和ODS数据存储两个子层；数据存储层根据应用的范围可以分为数据仓库和数据集市两种方式；数据访问层可以根据不同的应用提供多样化的访问机制。为了使论述更有条理，本论文将ODS放在数据存储层介绍。

### 4.3.1 数据获取层

数据获取层的功能是将数据从数据源加载到经营分析的数据仓库中。

(1) 数据源分析：

由于生产和管理的需要，电信在建设过程中形成了多个分散的独立运行的系统，包括BSS系统、计费帐务系统、营销支撑系统、综合结算系统、10000客户系统以及内部管理的系统，例如OA等MSS系统。另外电信的数据还包括一些外部的数据，例如市场情报数据、人口统计数据、市场调查数据等。经营分析系统通过与计费系统、渠道系统等建立标准的接口进行数据抽取。

(2) 接口分析

接口中规定了具体的文件传送方式、传送格式、传送时间、传送内容、交接方式、验证方式等内容。接口数据传送多采用固定格式的平面文件或表的形式。内容包括客户（用户）档案信息、消费信息、缴费/欠费信息、通话量信息等。不同的内容传送的频率不同，比如长市话详单信息按天传送，更粗粒度的量收信息按月传送等。传送的方式有全量和增量两种，对于客户（用户）档案信息，由于其更新量少，可采用增量方式，防止重复劳动，减少工作量。

(3) ETL数据抽取、转换和加载

由于历史的原因，各生产系统可能采用的是不同厂商的数据库，即使是同一厂商的数据库，版本也不尽相同，不同系统在数据的组织和存储方式方面各不相同，要把这些数据加载到数据仓库中，需要进行一定的处理。

ETL是Extraction、Transformation、Loading的缩写，指的是数据抽取、转换和加载，是数据仓库实现过程中，将数据由业务系统向数据仓库加载的主要方法，是数据仓库建设的关键部分[36]。

ETL过程管理：ETL是一个复杂的过程，需要进行过程管理。ETL的过程管理包括ETL的调度、ETL程序管理、ETL出错处理以及故障恢复[37]。

- ETL的调度：ETL是一批定时运行的后台过程，需要通过一个合理的规划进行

自动运行，只有在ETL过程出现异常时，进行人工干预或自动处理。经营分析系统提供系统管理员的控制和管理界面。

- ETL程序管理：ETL过程包括许多程序，这些程序在运行期间对数据进行处理，将这些ETL程序管理起来，可以保证ETL过程的正确及稳定。
- ETL出错处理及故障恢复：在ETL过程中由于数据接口、网络、主机或数据质量等问题，造成ETL过程出现错误，每次数据的处理和装载需要将非法的数据或处理失败的数据输送到专门的出错数据处理区中，一些错误可以通过自动处理进行恢复，一些错误需要人工进行处理。还可以自动将错误信息通过短信、mail等方式通知系统管理员或相关人员。

同样，数据在从数据仓库到数据挖掘库中也要运用到ETL技术。只不过由于前面的工作，数据在数据仓库中基本上不存在不一致的情况，因此这个阶段的ETL工作要轻松许多。

业务系统的数据通过ETL技术，形成统一的信息层—ODS(操作数据存储)层，在逻辑和存储上对业务处理系统与数据仓库进行隔离。

### 4.3.2 数据存储层

在数据存储层中，数据根据不同的层次进行组织：按照数据的粒度粗细和面向的分析应用，数据存储层分为ODS部分、数据仓库部分和数据挖掘部分。ODS部分数据粒度最细，遵循各生产系统数据的本质特征，按照一定的分类原则来组织数据，该部分数据最稳定；数据仓库部分是在ODS的基础上经过轻度和深度汇总产生，主要是为面向主题分析服务；数据挖掘部分指的是数据挖掘库，是在ODS和DW的基础上经过深层分析和整理产生，主要是为专题分析服务。数据的层次组织既能满足分析需求的变化，同时也能保证数据仓库的结构稳定。

(1) ODS (Operational Data Store)

ODS即为操作数据仓储，引入ODS的目的是降低目前经营分析系统与各业务系统的数据处理压力，提高数据生成效率，提升数据生成质量，打造真正意义上的统一数据平台，实现对本地应用的全面支持。

ODS和数据仓库互有优劣，数据仓库中存储的是历史数据，虽然能解决企业的决策需求，但是无法满足企业实时监控的需求，而ODS存储当前的综合数据，企业可以利用ODS把握实时的企业运作情况，及时采取应对措施。把ODS和数据仓库结合应用，

就能既满足企业决策需求，又可对企业进行实时监控。

(2) 数据仓库和数据集市

数据仓库提供面向整个企业的数据支撑，是经营分析系统的数据中心，按照企业完整的数据信息主题来组织数据，为企业提供一个完整的经营分析信息视图。而数据集市是具有特定应用的数据仓库，主要是面向部门和重要的领导，根据其关心的主题建立不同的数据集市，满足高性能、高灵活性的需求。

经营分析根据不同的主题将数据仓库划分为：客户基本资料信息、客户订购资料信息（服务信息）、客户帐户资料信息、计费系统配置表信息资料、计费系统提取帐单资料、计费系统提取销帐资料、计费系统提取长途话单资料、计费系统提取本地网话单资料、网间结算系统提取网间通话数据资料、智能网系统卡类业务通话数据资料等。数据集市又从数据仓库中提取出更有针对性的主题信息，例如：业务发展主题、收益情况主题、缴费/欠费主题、市场竞争主题等[38]。

数据仓库按照信息粒度的粗细综合了从细节级、轻度综合、中度综合直至高度综合的各级数据，例如客户话单数据可以从每条详单数据到按天、按月甚至按年的综合数据。

数据在数据仓库和数据集市中多以星型或雪花型模式存储，例如客户基本资料信息的逻辑模型如图 4-2 所示：

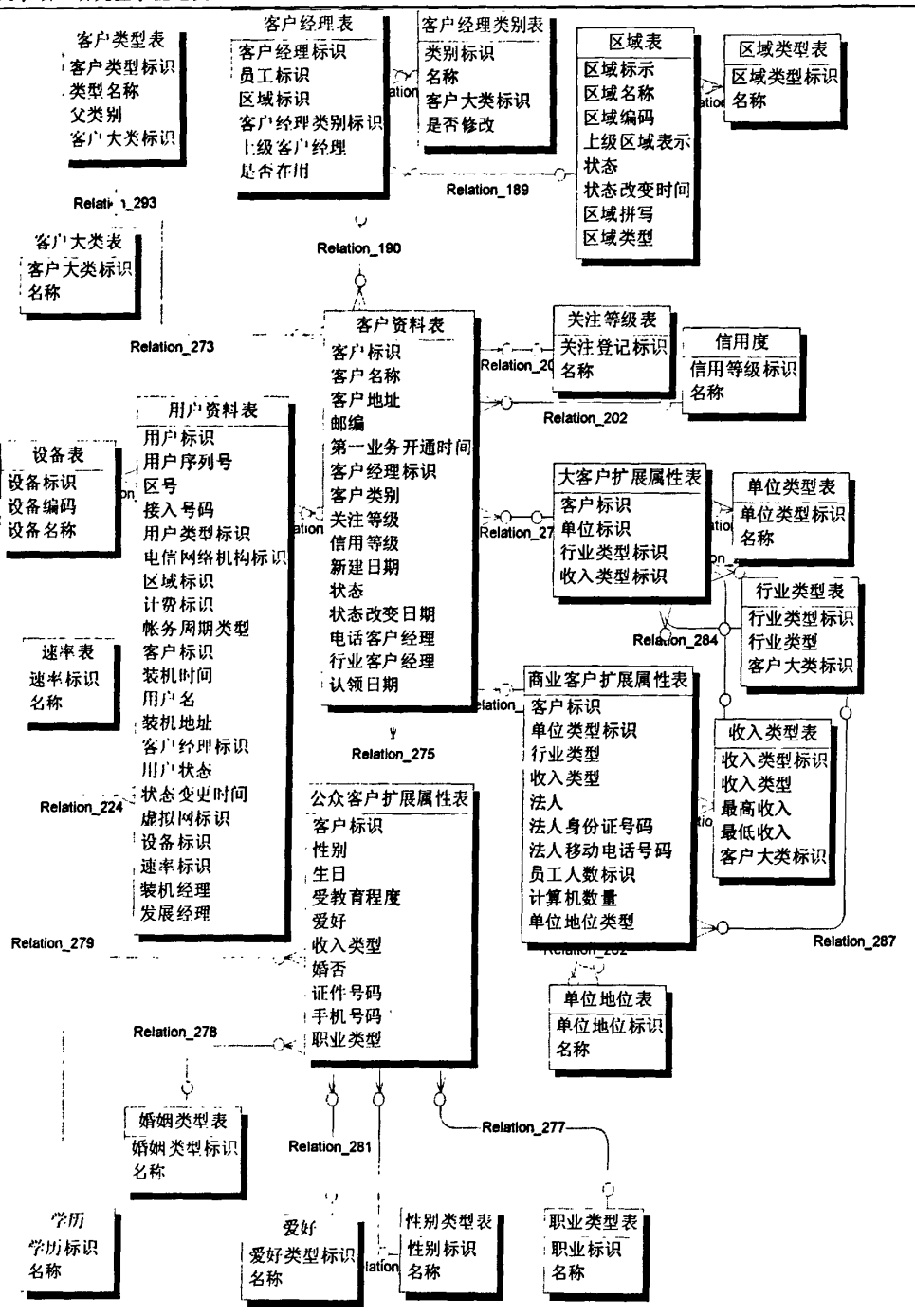


图 4-2　数据仓库中的客户模型

(3) 数据挖掘库

数据挖掘库中数据以宽表的形式组织存储。宽表是数据挖掘需要的一种数据结构，将客户相关的信息都记录在一行上，以便进行分析。在数据仓库中，数据是分不同主题组织的，并且由于生产系统的实际应用需求，客户数据是分客户、帐户、用户这三个不同层面存放的。宽表就是要将与一个客户相关的所有数据都有机地以客户 ID

为主键合并组织为一条记录，所有客户记录的集合就组成宽表。

宽表也是面向应用的，不同的应用对应不同的宽表结构。例如在客户细分的这一应用中，就可以选择对聚类有意义的、重要的信息来构建宽表的总体框架，这些信息如图 4-3 所示[39]：

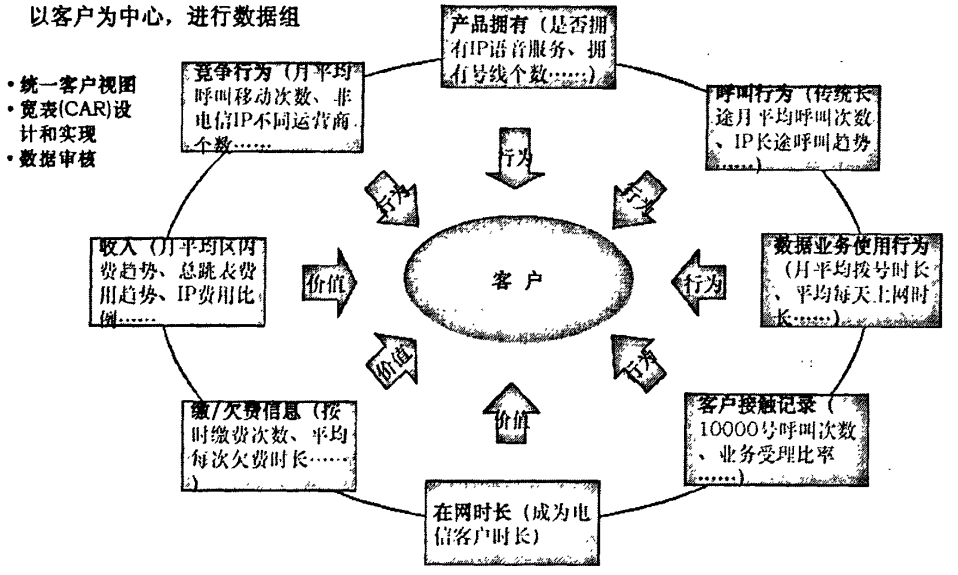


图 4-3　客户信息构成图

主要包括：

- 客户基本信息：了解客户基本人口统计信息，如客户籍贯、客户年龄、成为电信客户时长等。
- 客户产品拥有信息：了解客户使用了哪些电信业务，如国内长途、国际长途等主产品以及各种程控新业务。
- 客户帐单费用信息：了解客户各种产品分别的消费额、总消费额、消费趋势、消费波动、各种消费占的消费比重等。
- 客户话务量信息：了解客户的各种业务需求，如本地通话量、普通长途通话量、IP 长途通话量以及相应的趋势、波动等情况。
- 客户使用竞争业务信息：了解客户使用竞争业务的消费量和消费习惯，从而分析客户的实际需求，挖掘销售机会。
- 客户卡类消费信息：分析客户卡类使用的消费习惯和使用量。
- 客户使用单价信息

### 4.3.3　数据访问层

数据访问层的功能主要包括查询报表、OLAP分析、统计分析、数据挖掘等。通过对数据存储层中的数据进行加工、整理、分析、预测等操作，然后将获得的数据、以及隐含在数据中的趋势、规律等以文字、报表、曲线和各种图形的方式，简便、快捷地展现出来。例如，在OLAP分析中，利用工具Cognos可以轻松实现收入主题的分析，如图4-4所示。

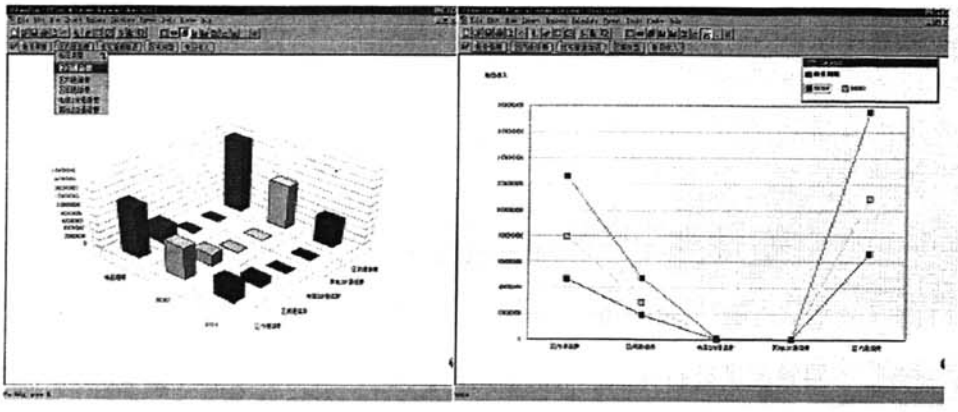

图 4-4　Cognos中收入分析展现

## 4.4　数据挖掘在经营分析中的应用

数据挖掘方法在经营分析系统的应用越来越得到电信行业的重视。目前主要运用的技术如图4-5所示，包括：决策树、聚类分析、神经网络等。不同的业务需求要采用不同的技术实现。

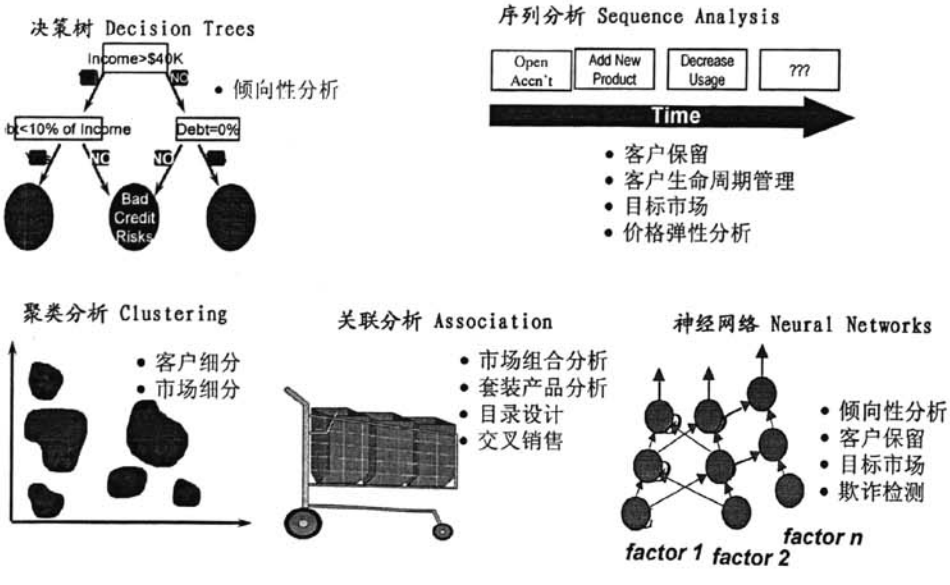


图 4-5　数据挖掘在经营分析中的应用

下面简单介绍数据挖掘在经营分析系统中的几个主要的应用：

(1) 客户细分

客户细分就是依据客户的基本信息、客户的行为信息、客户的消费信息以及客户与电信之间的交互信息等把客户划分为不同的群，针对每个群的具体特征选择合适的营销方案，提供更贴切的个性化服务，提高客户的满意度和忠诚度。

客户细分摆脱了以往简单以客户的消费水平作为依据的“分类”思想。例如，按照以往的标准，ARPU值在80元到150元的公众客户可能被定义为中端客户，但用分群的观点来看，这些客户并不一定都属于一个群，因为客户的ARPU值虽然都在80元到150元之间，但是消费构成可能差距很多，比如有的客户主要消费在传统语音业务方面，而有的客户可能主要消费在数据业务方面。显然这是两个消费习惯有较大差异的客户群，应该区分开来并采用不同的管理和营销方法。客户细分就可以实现比较科学合理的分群。

(2) 客户消费模式分析

客户消费模式分析（如固话话费行为分析）是对客户历年来长话、市话、信息台的大量详单、数据以及客户档案资料等相关数据进行关联分析，结合客户的分类，可以从消费能力、消费习惯、消费周期等诸方面对客户的话费行为进行分析和预测，从而为电信运营商的相关经营决策提供依据。

(3) 客户市场推广分析

客户市场推广分析（如优惠策略预测仿真）是利用数据挖掘技术实现优惠策略的仿真，根据数据挖掘模型进行模拟计费和模拟出账，其仿真结果可以揭示优惠策略中存在的问题，并进行相应的调整优化，以达到优惠促销活动的收益最大化。

(4) 客户欠费分析和动态防欺诈

通过数据挖掘，总结各种骗费、欠费行为的内在规律，并建立一套欺诈和欠费行为的规则库。当客户的话费行为与该库中规则吻合时，系统可以提示运营商相关部门采取措施，从而降低运营商的损失风险。

(5) 客户流失分析

随着电信业竞争的激烈升级，客户流失问题也越来越严重。实践证明挽留一个客户比发展一个客户或者客户流失后再“策反”的成本小的多。而客户流失分析则是实施客户挽留的基础。根据已有的客户流失数据，建立客户属性、服务属性、客户消费情况等数据与客户流失概率相关联的数学模型，找出这些数据之间的关系，并给出明确的数学公式。然后根据此模型来监控客户流失的可能性，如果客户流失的可能性过高，则通过促销等手段来提高客户忠诚度，防止客户流失的发生。这就彻底改变了以往电信运营商在成功获得客户以后无法监控客户流失、无法有效实现客户关怀的状况[40]。

## 4.5 本章小结

本章主要介绍电信经营分析系统的概况，包括系统的建设背景、系统的功能架构、核心的数据处理流程。最后介绍数据挖掘技术在经营分析系统中的应用。其中数据处理流程包含：

- 数据获取层：涉及数据源分析、接口分析、ETL（数据抽取、转换和加载）；
- 数据存储层：包括ODS、数据仓库、数据集市、数据挖掘库；
- 数据访问层：包括查询报表、OLAP分析、统计分析、数据挖掘等。

# 第5章　应用数据挖掘技术进行客户细分

在“以客户为中心”的指导原则下，客户细分所应用的数据模型从分离的、以各个产品为中心的模式，转向为以客户为中心的整合的同一客户视图，基于数据分析/挖掘平台，处理客户信息实现客户细分和相关预测，为客户理解和客户关怀提供更有力的依据。

以下按照标准的数据挖掘过程CRISP-DM（商业理解—>数据理解—>数据准备—>建模—>评估），详细叙述应用聚类方法进行客户细分的全过程。

另外，本文叙述中涉及到“细分”和“分群”这两个概念，实际上这两个概念并无本质区别，只是在理论研究中常常用“细分”这个概念，而在实际应用中则常常用“分群”这个概念。本文在用词方面采用习惯用法。

## 5.1　商业理解

挖掘目标的定义要求非常明确，任何不明确的定义都会严重影响模型建立的准确性和应用时的效果。数据挖掘的目的是数据挖掘的重要一步。经营分析客户细分的目的，就是根据客户的价值和行为特点，把目标客户准确的定位于某一个客户群中，再针对这个客户群的特征实行精确化管理和营销。

一个客户的相关信息非常广泛，包括：人口统计信息（如年龄、职业、性别）、组织信息（如行业、公司规模、收入）、态度/意向（如一般的购买态度、购买心理）等等，要想把这些信息都收集齐全往往费时费力，并且很多信息不容易获得。实际上有些信息对电信业务的营销并没有很大的作用。因此首先我们要选择那些对营销分析起重要作用的并且是可得的信息，如客户价值信息（如客户帐单费用信息、客户卡类消费信息、客户使用单价信息）、客户行为信息（客户产品拥有信息、客户话务量信息、客户使用竞争业务信息），从这两个维度开始再向其他的维度蔓延。依据客户的价值信息和行为信息对客户进行分群称为V-B分群（价值—Value，行为—Behavior），本文主要介绍V-B分群的V分群。

本文研究的客户群范围：某电信公司客户中的小型商业客户（所有的一户一机商业客户和一户两机客户中没有宽带业务的商业客户）。因此在客户细分中，业务问题定义为：依据客户价值维度对小型商业客户进行细分，了解不同客户群的特征和需求，以便实施精确化管理和营销。

## 5.2　数据理解和数据准备

数据理解部分要解决数据的具体来源、数据组成结构和说明、建立数据处理流程，具体如下：

### 5.2.1　数据来源

客户细分所依据的信息数据来源于各生产系统，包括BSS系统、计费帐务系统、套餐系统、10000号客户系统、网间结算系统、智能网系统等。

### 5.2.2　客户信息组成

客户细分模型根据客户的价值/行为信息对客户进行分群，涉及到的客户信息包括：客户基本信息、客户产品拥有信息、客户帐单费用信息、客户使用竞争业务以及卡消费信息。埃森哲公司把常见的客户信息归纳为：（本文只需提取其中客户价值信息部分）

表 5-1　客户信息组成表

| 数据类型 | 细项 |
|---|---|
| 客户基本信息 | 了解客户基本人口统计信息，如客户籍贯、客户年龄、成为电信客户时长等 |
| 产品拥有信息 | 了解客户使用了哪些电信业务，如国内长途、国际长途等主产品以及各种程控新业务 |
| 月帐单 | 了解客户各种产品分别的月消费额、月总消费额、消费趋势、消费波动、各种消费占的消费比重等 |
| 新业务 | 分析客户使用的新业务种类、新业务消费额度、新业务消费趋势、研究客户对电信新业务的感兴趣程度及使用程度 |
| 付费信息 | 考察客户常用的付费方式以及付费习惯 |
| 欠费信息 | 分析客户的欠费记录，从而推断客户的信用情况 |
| 本地通话 | 分析客户的本地通话需求，如月均本地费用额度、本地费用趋势、本地费用波动等 |
| 呼出信息 | 考察客户的呼出量、呼往的运营商分布，通过呼往的运营商的不同分析客户的交际圈 |
| 呼入信息 | 通过呼入量的多少考察客户与外界交往的频繁程度，从而分析客户异质流失的可能 |
| 传统长途 | 分析客户的传统长途需求及消费习惯，如长途呼叫单次时长、呼叫时间、呼叫频率、长途使用需求等 |
| 电信IP长途 | 分析客户的电信IP长途需求及消费习惯，如IP长途呼叫单次时长、呼叫时间、呼叫频率、长途使用需求等 |
| 200、300、201等卡类 | 分析客户卡类使用的消费习惯和使用量 |
| 竞争运营商IP长途 | 分析客户的竞争运营商IP长途需求及消费习惯，如竞争运营商IP长 |

| | |
|---|---|
| | 途呼叫单次时长、呼叫时间、呼叫频率、长途使用需求等 |
| 竞争运营商 ISP | 分析客户使用竞争运营商 ISP 拨号上网的使用量和使用习惯，从而分析客户的实际窄带需求，挖掘销售机会 |
| 电信窄带接入信息 | 分析客户使用电信窄带接入使用量，使用习惯 |
| 宽带客户基本信息 | 了解宽带安装地址、费率以及宽带用户基本信息 |
| 宽带使用记录 | 分析宽带使用入流量、出流量、特殊时段宽带使用量、接入方式等 |
| 宽带信息更新记录 | 了解客户宽带信息更新频率及更改类型 |
| 10000 号呼入 | 通过 10000 号的呼入，分析客户的投诉、报修、话费查询等客户发起的交互 |
| 10000 号呼出 | 通过 10000 号呼出分析电信主动发起的客户交互 |
| 渠道信息 | 分析客户通过渠道与电信进行的交互 |
| 其他 | 包括市场调查、市场营销活动反馈等信息 |

### 5.2.3　客户信息处理流程

客户信息处理流程如图 5-1 所示，每个阶段都要涉及到 ETL 操作，可见 ETL 操作在整个信息处理流程中的重要作用。

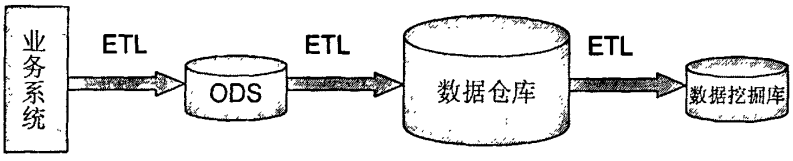


图 5-1　客户信息处理流程

数据准备工作应具体到各业务细节问题，例如：

(1) 确定取数时间窗，研究客户价值，至少需要六个月的历史数据。

(2) 剔除特殊的客户信息，例如局用客户、公免客户数据。

在进行数据收集和整理时要注意：

(1) 收集的数据尽量全面，以保证最全面、客观的客户视图。

(2) 要考虑数据的一致性，因为数据来自不同的业务系统，需要排除各个系统之间可能存在的不一致情况。

数据经过 ETL 过程到数据仓库以后，还要利用系统的数据审核功能检查数据的准确性，发现数据处理过程中因人为的或者系统的原因导致的异常数据。例如根据客户的竣工时间计算出客户的在网时长长达数百年，或者客户总费用为负的情况，这就要求数据分析人员和业务人员一起查找原因并采取解决办法。

### 5.2.4　宽表的设计

为了满足数据挖掘建模要求，针对不同的分析目的，数据仓库的数据需要进一步的选择、处理并导入数据挖掘库的宽表中。由于聚类分析涉及到客户以往六个月的历史数据，如何将这六个月的历史数据进行整合加工，使之更能体现出客户的消费特征是挖掘前必须要考虑的问题。属性的衍生变量如均值、比例、趋势、波动、产品使用率等能很好的体现出客户的消费特征变化，为分析提供新的数据视点，这些新的视点对客户细分有着及其重要的作用，因此宽表设计中需要引入对业务具有指导意义的衍生变量。通过变量的转换之后，一方面可以减少参与建模的变量数量，另一方面能更好的表达变量的意义，便于模型解释。

衍生变量中的波动和趋势的计算方法如下：

(1) 波动：六个月的最大值减去最小值再除以六个月的平均值。

(2) 趋势的算法采用了较为准确的 slope 算法：客户六个月费用的拟合直线的斜率，其中具体计算公式如下：

$$slope = \frac{n\sum xy - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \qquad (5\text{-}1)$$

其中：n 为数据对象个数；

x 为时间段标识；

y 为每个时间段内的观察值。

宽表的设计是数据准备阶段最重要的部分，是分析建模的基础。经过业务人员和技术人员的反复讨论，在小型商业客户细分这个应用中，宽表的价值（V）字段部分如表 5-2 所示，附录一列出宽表的所有价值字段（由于篇幅限制，宽表的行为（B）属性不再做详细介绍）。

表 5-2　宽表部分价值字段

| 字段名称 | 类型 | 字段说明 |
|---|---|---|
| CUST_ID | NUMBER(10) | 客户 ID |
| MB_AVG_TOTAL_FEE | NUMBER(10) | 总费用均值 |
| MB_AVG_TOTAL_FEE_FACT | NUMBER(10) | 实缴总费用均值 |
| MB_AVG_LOCAL_FEE | NUMBER(10) | 市话费均值 |
| MB_AVG_LONG_FEE | NUMBER(10) | 长话费均值 |
| MB_AVG_DATA_FEE | NUMBER(10) | 数据费用均值 |

## 5.3　建模

在客户 V-B 分群中，先根据客户的 V 特征和 B 特征分别进行细分，在得到的结果的基础上再进行交叉结合，获得更有战略意义的分群。本文在建模部分仅以客户 V 分群为例子

进行说明。

### 5.3.1　客户分群程序总流程

聚类算法包括以下几个主要的模块：1)宽表数据上载主机内存；2)数据标准化；3)构建K-D树；4)产生k个初始聚类中心；5)K-D树遍历；6) 修剪候选中心集；7)聚类结果回写数据库。整个聚类算法的流程图如图5-2所示：

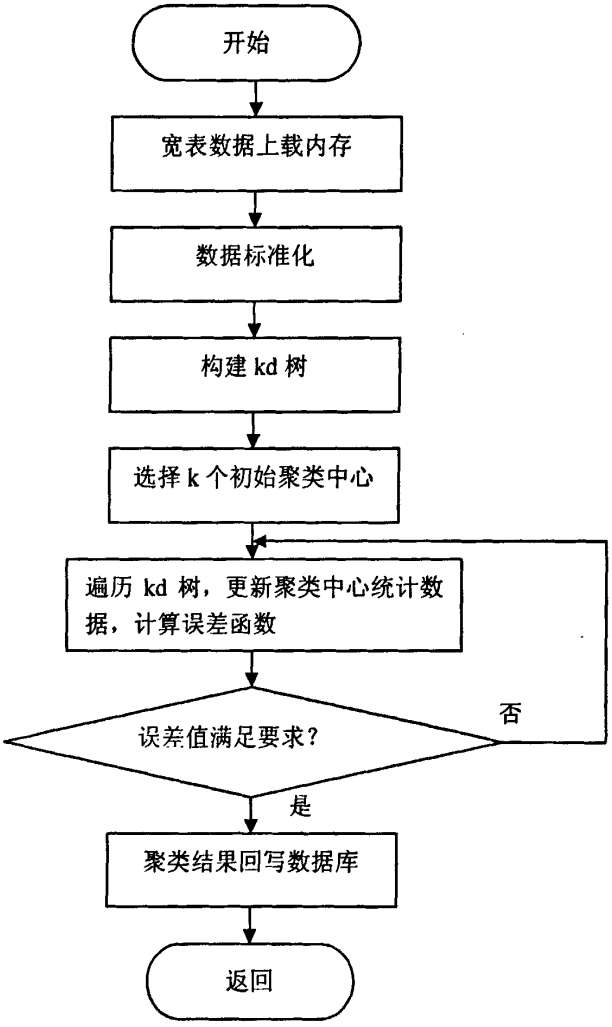


图 5-2　聚类算法流程图

值得注意的是，对K-D树的遍历是循环数次的，直到满足规定的误差需求。流程中的每一阶段都由不同的函数实现相应的功能。算法中运用的函数列表如下：

表 5-3　算法中涉及的函数列表

| 函数名称 | 函数描述 | 备注 |
| --- | --- | --- |

| | | |
|---|---|---|
| MainProcessMgr | 聚类算法总进程 | |
| Format | 数据标准化 | 将各维数据标准化到区间[0,1]之间，使得各维具有相同的权重 |
| SelClusterSet | 初始聚类中心选择 | 按照随机数据的分布特征，更合理地选择初始聚类中心 |
| BuildKdTree | 构建K-D树 | 叶子节点容量的大小(LeafSize)对K-D树的高度及整个算法的效率影响重大，要选择一个合适的值 |
| ClusterNodeCpy | 候选集聚类中心点赋值 | |
| TreeValueCpy | 数据对象赋值 | |
| UpdateTreeNode | 更新K-D树节点信息 | 包括：节点空间包含的数据对象个数、每个维度的线性和LS、平方和SS、上下界 |
| TraverseTree | K-D树遍历 | |
| Pruning | 修剪树节点的候选中心集 | 对K-D树的每个节点运行修剪算法，直到节点的候选集中只有一个候选者 |
| GetMinDist | 计算候选集中的候选者到节点所代表的子空间中的所有点的最短距离 | |
| GetMaxDist | 计算候选集中的候选者到树节点所代表的子空间中的所有点的最大距离 | |
| UpdateClusterSet | 更新聚类集信息 | |
| TraverseKmeas | 标准的k-means算法 | 当叶子节点的候选集中的候选者个数大于1时，就要应用标准的k-means算法将叶子节点空间中的对象分别归入不同的候选者所代表的聚类 |
| GetCurDate | 获取系统当前时间 | |
| TraverseToDb | 聚类结果回写数据库 | |

图 5-3 描述出以上各函数之间的调用关系：

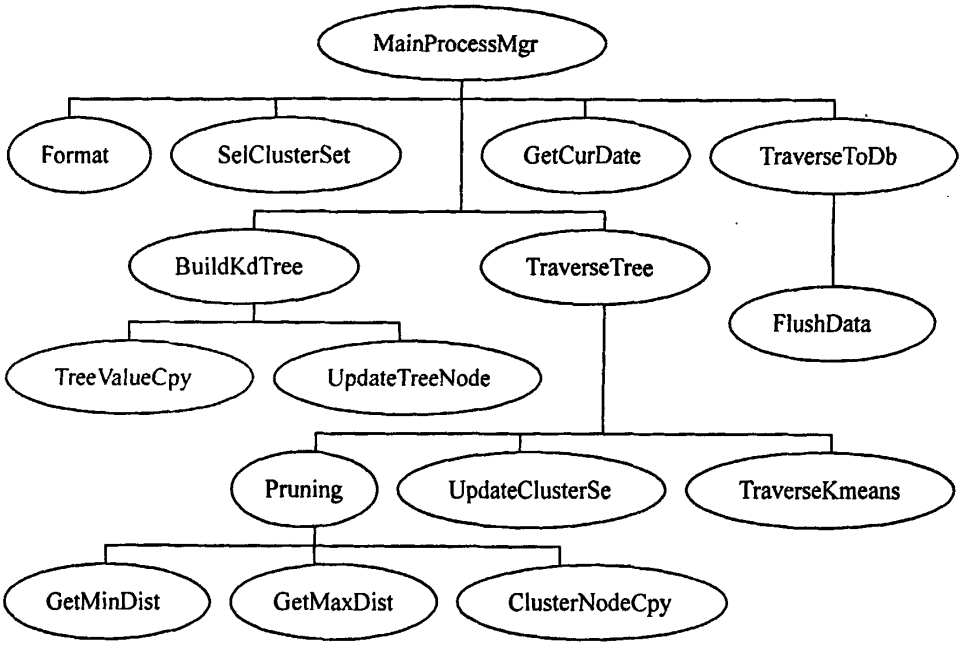


图 5-3　各函数之间的调用关系

### 5.3.2　模块详细设计

(一)　数据标准化

客户细分中的 k-means 算法根据客户数据之间的相似度（距离）来进行分群。而客户的属性有很多种，例如费用，通话时长，单位不同，数值上也有很大的差距。如果直接用原始数据来实现算法，势必造成数值上大的属性对聚类结果的影响大，而数值小的属性对聚类结果的影响小甚至可以忽略。这样对各个属性来说是不公平的。为了避免这个问题，使得聚类结果更具有实际意义，需要在聚类之前对这些数据进行标准化，确保各个属性对聚类的结果有相同的权重。本文采用较简单的最小一最大标准化方法，通过线性变换将各维数据标准化到区间[0, 1]之间。设 $\min_A$ 和 $\max_A$ 分别为属性 A 的最小值和最大值，最小一最大标准化通过计算：

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A \qquad (5\text{-}2)$$

将属性 A 的值 v 映射到区间[$new\_\max_A$, $new\_\min_A$]中的 $v'$。这种方法对原始数据进行了线性变换，保持了原始数据值之间的线性关系[41]。

标准化函数说明入下：

函数：void Format(TreeNode *pRoot, TreeValue *pTotalBuf)

输入：K-D 树的根节点 pRoot、储存所有数据对象的内存空间 pTotalBuf

输出：标准化后的各维数据

功能：标准化各维度数据

```
1) For 数据对象的每个维度  do
   {
      计算维度的最大值最小值
   }
2) For 所有数据对象  do
   {
      For 数据对象的每个维度  do
      {
         根据最小一最大标准化方法标准化维度值
      }
   }
3) Return
```

(二)　构建 K-D 树

K-D 树节点应包含的内容：节点所代表的子空间包含的数据对象个数、SS（对象各个维度上的平方和）、LS（对象各个维度上的线性和）、各维度的上下限、数据对象指针、左右孩子指针。K-D 树构造函数如下：

函数：TreeNode *BuildKdTree(TreeNode *pRoot, TreeValue *pBuf, const long lDepth, const long lLeafSize, TreeValue *pTempBuf)

输入：D 维的数据对象集合 pBuf, K-D 树当前的深度 lDepth 和叶节点容量 lLeafSize、pTempBuf 临时缓冲区，pBuf 集合分成两个集合 P1 和 P2

输出：存储集合 P 的 K-D 树的根部 pRoot

功能：构建 K-D 树

```
1) If 集合 pBuf 中的数据对象个数小于 Leafsize
   {
      pRoot 的数据对象指针指向存储数据对象的集合 P
      Return 叶节点 pRoot
   }
```

2) else

{

选择当前待划分的维 lDim=当前最长长度维

lDim 维的中点 SplitPoint=pRoot->lDimLs[lDim]/pRoot->lNum

根据 SplitPoint 将数据对象划成集合 P1、P2，储存在 pTempBuf 中

如果对象在 lDim 维上的值小于 SplitPoint，则将对象归到左集合 P1,同时计算 P1 中数据对象个数、SS、LS、维度的上下限

如果对象在 lDim 维上的值大于等于 SplitPoint，则将对象归到右集合 P2，同时计算 P2 中数据对象个数、SS、LS、维度的上下限

}

3) If 集合 P1 非空

{

Leftchild= BuildKDTree(pLeftChild, pTempBuf(P1 部分), lDepth+1, lLeafSize, pBuf)

}

4) If 集合 P2 非空

{

Rightchild= BuildKDTree(pRightChild, pTempBuf(P2 部分), lDepth+1, lLeafSize, pBuf)

}

5) Return K-D 树根节点 pRoot

LeafSize 的大小设置在构建 K-D 树算法中至关重要，并且直接影响到整个聚类算法的效率。LeafSize 设置的两种极端的情况是：1)LeafSize=1，这种情况表示 K-D 树的每个叶子节点只能存储一个数据对象，对于一个数十万甚至上百万的数据对象集来说，意味着要建一棵高度非常高的 K-D 树，很显然在树的遍历上要花费大量的时间；2) LeafSize>=所有数据对象个数，这种情况表示 K-D 树的高度为 1，所有的数据对象都存储在 K-D 树的根节点中，运用标准的 k-means 算法把根节点空间中的所有数据对象进行聚类。显然上面两种极端情况都不可取。好的 LeafSize 值的选择要满足：1)树的深度 D 合适；2)在 K-D 树遍历和叶子节点标准 k-means 聚类之间取得一个平衡。设置小了，树的深度 D 变大，对 K-D 树遍历的时间增加；设置大了，对节点的修剪得不到很好的效果，最终还要对叶节点进行标准 k-means 聚类。

创建 K-D 树的流程图如图 5-4 所示：

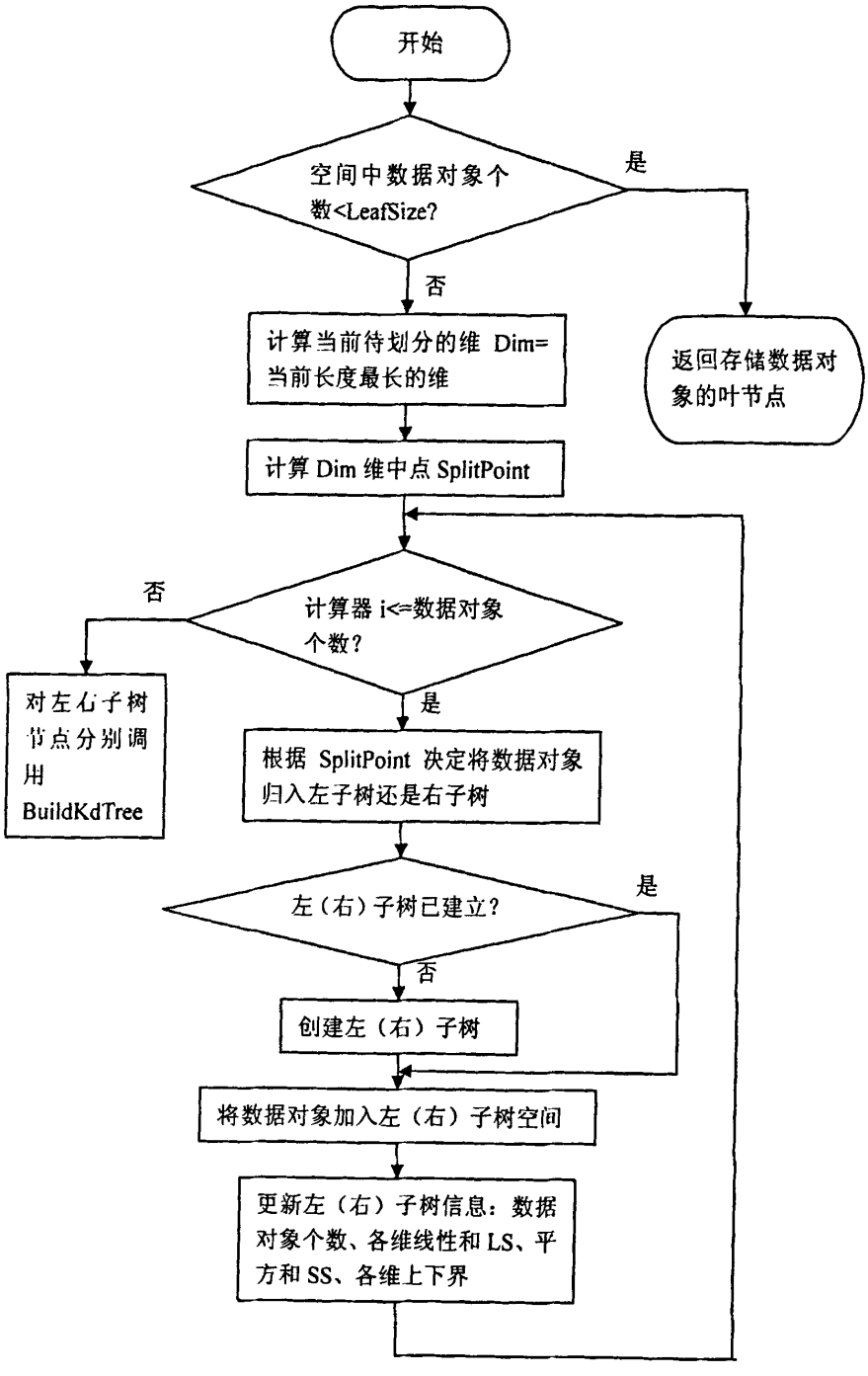


图 5-4　K-D 树构造流程图

(三)　生成初始聚类中心

本文选取初始聚类中心的方法是根据随机数据的分布特点，绝大部分的数据应分布在由均值$\mu$和方差$\sigma$构成的区间$[\mu-\sigma, \mu+\sigma]$内。本算法首先计算所有数据对象(D 维)在第

d维度上的均值为$\mu_d$和标准差为$\sigma_d$，然后选择（$\mu_d-\sigma_d$，$\mu_d+\sigma_d$）之间的K个等分点作为各初始聚类中心在第d维上的坐标值。设第i个初始聚类中心在第d维上的坐标值为$m_{id}$，则：

$$m_{id}=(\mu_d-\sigma_d)+\frac{2i\sigma_d}{K}，\ i=1,2,\cdots,k; \qquad (5\text{-}3)$$

因此第i个聚类中心的坐标为$(m_{i1},m_{i2},...,m_{id})$，d=1,2,…,D。生成初始聚类中心的具体函数如下：

函数：ClusterSet SelClusterSet(TreeNode *pRoot,TreeValue *pTotalBuf,long iSetCnt)

输入：K-D树的根节点pRoot、储存所有数据对象的内存空间pTotalBuf、聚类个数iSetCnt

输出：包含iSetCnt个初始聚类中心集

功能：根据随机数据分布特征更合理地选择iSetCnt个初始聚类中心

1) For 数据对象的每个维度 do

```
{
    根据K-D树根节点中存储的信息计算各维均值和方差
}
```

2) 创建空聚类中心集

3) For 每一个初始聚类中心 do

```
{
    For 数据对象的每个维度  do
    {
        计算初始聚类中心在各维度上的坐标
    }
}
```

4) Return 初始聚类中心集

(四)　遍历K-D树

函数：TraverseTree(TreeNode *pRoot,ClusterSet *pSet,ClusterNode *pNode,const long lNodeLen)

输入：K-D树的根节点pRoot,聚类中心集pSet、候选集pNode、候选集个数lNodeLen

输出：聚类结果，即K个聚类

功能：遍历K-D树，将所有数据对象划分为K个聚类

1) Alive=Pruning(node,P,1,d)

2) If |Alive=1| then //如果节点的候选集中只有一个候选者

3) {

根据存储在节点中的信息（数据对象个数、各维度线性和LS、平方和SS），更新聚类中心集的统计数据

}

4) 不再遍历该节点的子节点，继续按深度遍历K-D树中剩余的节点

5) Else if node 是叶节点 then

{

For node 的每一个对象 //运行标准的k-means算法

{

找到最近的聚类中心 $p_i$

把这个对象分配给 $p_i$

更新聚类中心集的统计数据

}

}

6) Else //继续遍历该节点的子节点

{

TraverseKDTree(lchild,Alive,|Alive|,d)

TraverseKDTree(rchild,Alive,|Alive|,d)

}

该算法流程图如图5-5所示：

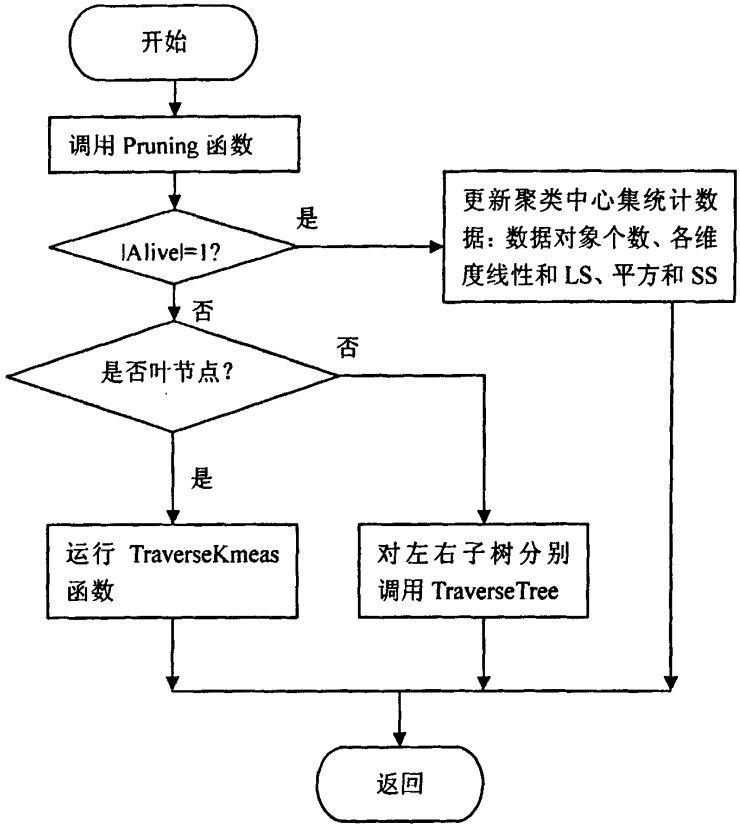


图 5-5　K-D 树遍历流程图

(五)　修剪函数

函数：ClusterNode *Pruning(TreeNode *subTree, ClusterNode *P, const long lNodeLen, long &lAliveLen)

输入：子树的根节点 subTree, 父节点候选集 P、父节点候选集个数 lNodeLen

输出：子树节点 subtree 的候选集 pAlive、候选集个数 lAliveLen

功能：生成 K-D 树的节点候选集

1) pAlive 指针置空

2) For 每一个原型 $p_i \in P$ do

{

计算 $p_i$ 到 K-D 树节点 subtree 所代表的子空间里的每一个点的最大距离($max_i$)和最小距离($min_i$)

}

3) 找出所有 $max_i$ 中的最小者，称为 MinMax

4) For 每一个原型 $p_i \in P$ do

```
{
    If  min_i <=MinMax
    {
        pAlive=pAlive+{p_i}
        //如果某个原型 p_i 满足条件，则添加到子节点的候选集中
    }
}
```

5) Return pAlive

其中：最大值最小值的算法思想如下：

(1) 最大值

可以证明，候选者到子空间的最大距离是到子空间的一个角的距离。设聚类中心 $p_i$ 到子空间的最远的那个角的坐标为 $edge_i(edge_{i1}, edge_{i1}, \Lambda, edge_{id})$，其第 j 维的坐标值的计算方法为：

$$edge_{ij} = \begin{cases} B_j^l : |B_j^l - p_{ij}| > |B_j^u - p_{ij}| \\ B_j^u : otherwise \end{cases} \qquad (5\text{-}4)$$

其中：$B_j^u$ 和 $B_j^l$ 分别是子空间第 j 维的上界和下界。由此可以得出 $p_i$ 到子空间的最大距离 $\max_i$ 的计算公式：

$$\max_i = \sqrt{\sum_{j=1}^{d}(p_{ij} - edge_{ij})^2} \qquad (5\text{-}5)$$

(2) 最小值

最小值相比最大值略为复杂。可以借助二维平面来考察最小值的取法。

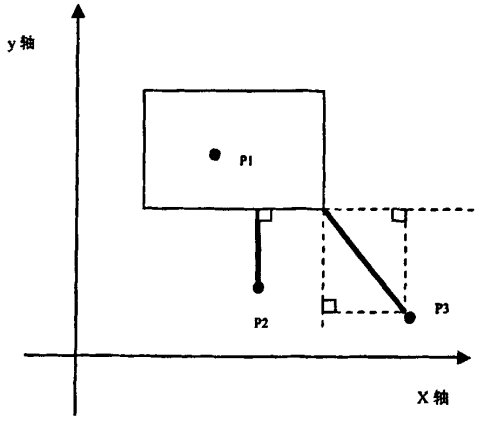


图 5-6　在二维坐标中最小距离计算的三种不同情况

如图 5-6 所示，聚类中心到子空间的最短距离可以分成三种情况：

(a) 对于点 P1，由于它被包含在子空间中，因此最短距离为 0；

(b) 对于点 P2，它到子空间的最短距离应该是点到子空间 y 维下限的垂直距离；

(c) 对于点 P3，它到子空间的最短距离应该是到离它最近的子空间的角的距离。

设聚类中心到子空间最短距离的点的坐标是 $dot_i(dot_{i1},dot_{i1},...,dot_{id})$，把这几种不同的情况归纳到一起，可以得出该点在 j 维上的坐标值：

$$dot_{ij}=\begin{cases} p_{ij}:B_j^l\le p_{ij}\le B_j^u \\ B_j^l:|B_j^l-p_{ij}|\lhd B_j^u-p_{ij}\ |\ and((p_{ij}\le B_j^l)or(p_{ij}\ge B_j^u)) \\ B_j^u:otherwise \end{cases} \quad (5\text{-}6)$$

因此，聚类中心到子空间的最小距离的计算公式如下：

$$\min_i=\sqrt{\sum_{j=1}^{d}(p_{ij}-dot_{ij})^2} \quad (5\text{-}7)$$

从上述最大值和最小值的计算可以看出，每一个候选者都是独立地对每个节点所代表的子空间计算最大值和最小值，而子空间的盒子的坐标除了在父节点用于分支的一个维不同之外，其他的则完全相同。利用这个特点，借助与父节点的信息来计算子节点的最大最小距离，可以降低计算的复杂度。当然，要存储这些信息需要额外的开销。这是本文算法可以继续尝试改进的一个地方。

K-D 树中的每一个节点都需要保存自己的候选集信息。Pruning 函数在对各个节点候选集的处理上有两种思路：1)在遍历到子节点时，先把父节点的候选集拷贝到子节点的候选集中作为初始的候选集，然后根据计算修剪这个候选集，最后得到子节点自己的候选集；2)把子节点的初始候选集设置为空，然后根据计算把父节点候选集中满足条件的候选者加入到子节点的候选集中，最后得到子节点自己的候选集。本文中采用了效率明显更高的第 2)种方法。

Pruning 算法流程图如图 5-7 所示：

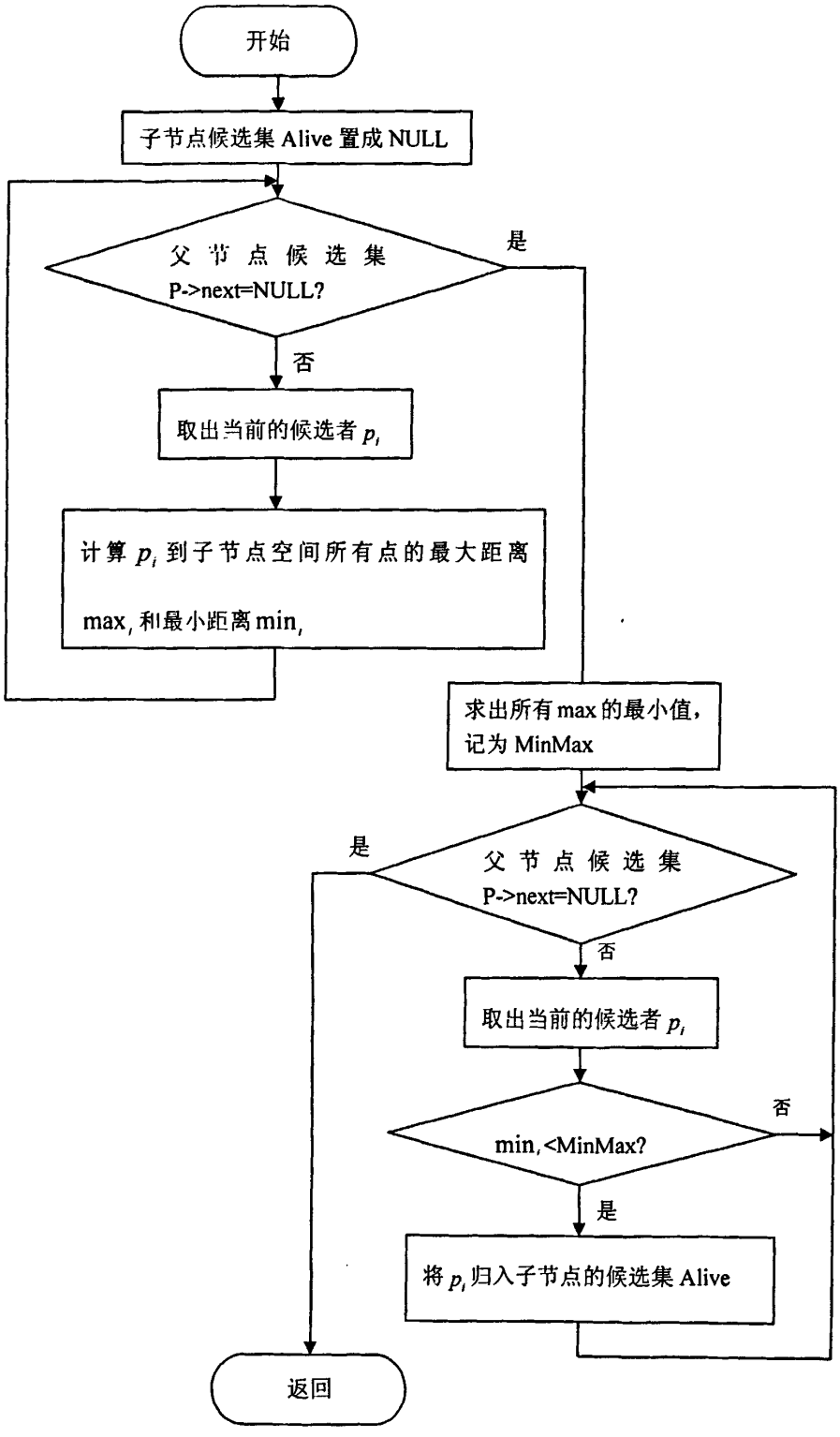


图 5-7　Pruning 函数流程图

(六)　叶节点聚类函数（标准 k-means 函数）

当遍历 K-D 树到达叶子节点时，对叶节点应用修剪函数仍然无法把候选集的候选者个

数减少至1个，这时就需要运用标准的k-means算法对叶节点所包含的所有数据对象进行聚类。实现算法如下：

函数：TraverseKmeas(TreeNode *pRoot,ClusterSet *pSet,ClusterNode *pNode)

输入：叶节点指针*pRoot，候选集*pNode，聚类中心集*pSet

输出：叶子节点所包含的数据对象的聚类结果

功能：将叶子节点中所包含的数据对象进行聚类

```
1) For 叶子节点中的每一个数据对象  do
   {
      计算数据对象到候选集第一个候选者的距离，并设置成数据对象到候选集的初始
      最短距离CurMin，将第一个候选者作为数据对象的初始归属聚类中心
      While(候选集中剩余的候选者)
      {
         For 数据对象的每个维度  do
         {
            计算数据对象到候选者的距离NextMin
         }
         If NextMin<CurMin then
         {
            修改数据对象所属的聚类
            CurMin= NextMin  //更新数据对象到候选集中候选者的最短距离
         }
      }
      更新聚类集数据对象个数信息
      For 数据对象的每个维度  do
      {
         更新聚类集平方和信息SS
         更新聚类集线性和信息LS
      }
   }
```

TraverseKmeas算法流程图如图5-8所示：

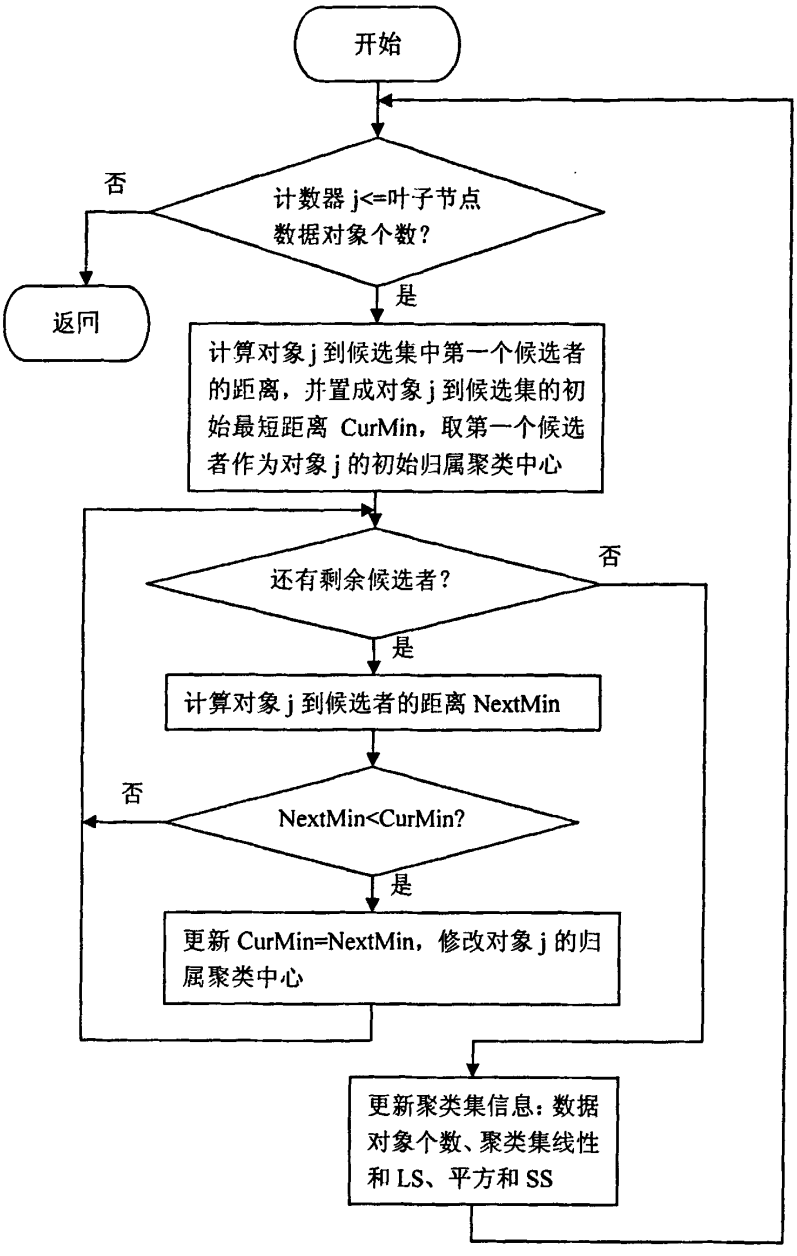


图 5-8　叶节点聚类流程图

### 5.3.3　聚类结果分析、营销建议

在实际的营销活动中，客户细分并不是最终目的，在细分的结果上归纳整理出每一客户群的显著特征，抓住目标客户群，挖掘客户潜在的需求，实施精确化营销才是电信实施客户细分的根本目的。

这一阶段需要分析人员将业务知识和对数据的理解紧密地结合。为了使聚类的结果

中隐藏的规律更加明了，可以借助一些辅助手段，如报表、统计、可视化等来表现聚类的结果。

本次聚类分析的小型商业客户经过筛选后得到的有效总数是113122，对分群的结果进行分析统计，得到表5-4：

表 5-4　分群结果的ARPU值、趋势、客户数及相应比例

| V分群 | ARPU | 趋势 | 趋势比例 | 客户数 | 客户数占比 | 收入占比 |
|---|---|---|---|---|---|---|
| V5 | 307 | -28.1 | -9.1% | 23333 | 20.6% | 36.6% |
| V4 | 268 | 19.2 | 7.2% | 23152 | 20.5% | 31.6% |
| V2 | 161 | -6.5 | -4.1% | 12902 | 11.4% | 10.6% |
| V3 | 115 | -2.3 | -2.0% | 17877 | 15.8% | 10.4% |
| V1 | 59 | -0.8 | -1.3% | 35858 | 31.7% | 10.8% |
| 总体 | 173 | -3.2 | -1.9% | 113122 | 100.0% | 100.0% |

根据上表，我们可以得到对比图5-9：

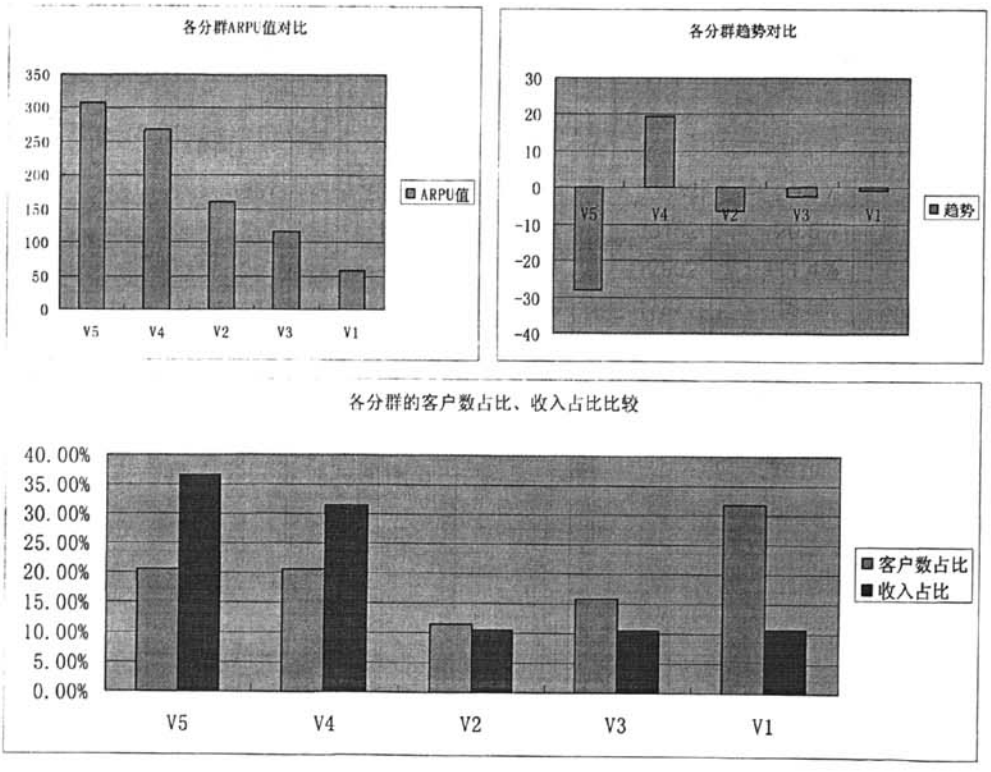


图 5-9　聚类结果中各个分群的对比图

从图5-9中，我们可以很直观地得到每个分群的一些总体特征。我们以比较典型的V5分群和V4分群为例：

(1) V5分群

- ARPU值最高，属于小型商业客户中的高值客户
- 总费用下降速度很快，月均下降28.1元，月下降率达到9.1%

- 20.6%的客户创造了36.6%的收入
- 根据以上特征，定义V5为高值高危型

(2) V4分群

- ARPU值第二高，属于小型商业客户中的高值客户
- 总费用上升速度很快，月均上升19.2元，月下降率达到7.2%
- 20.5%的客户创造了31.6%的收入
- 根据以上特征，定义V4为高值增长型

为了更细致地刻画客户分群，满足市场分析人员的需要，我们还可以对各个分群的费用构成（分项费用）、每一分项费用的占比、趋势进行深入分析。以各个分群的费用构成为例，具体数据见表5-5：

表5-5　分群结果的各种费用占比

| 价值分群 | V1 | V2 | V3 | V4 | V5 |
|---|---|---|---|---|---|
| 市话费占比 | 27.26% | 66.50% | 38.16% | 43.70% | 42.79% |
| 数据费占比 | 1.75% | 2.99% | 1.79% | 1.31% | 1.67% |
| 租费占比 | 59.03% | 25.37% | 34.44% | 16.32% | 15.26% |
| 长话费占比 | 10.84% | 2.68% | 23.72% | 37.46% | 39.37% |

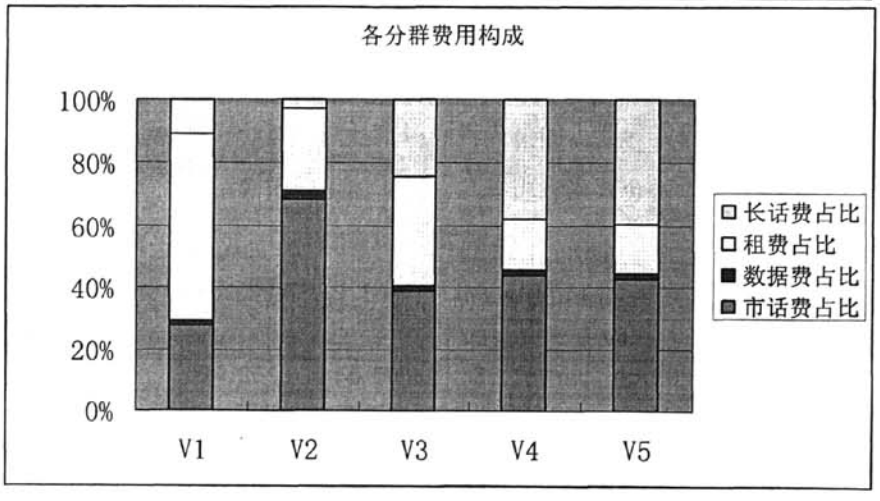


图5-10　各分群的费用组成

从图5-10可以看出，V1和V2分群（中低值）的长话费在总费用中的比重明显比V4和V5分群（高值）小。

根据以上的分析和特征刻画，我们给出对各个特征明显的客户群的营销建议方案，仍以V5和V4分群为例：

- V5(高值流失型):做好一对一营销和客户关怀，在客户流失之前挽留客户；该分群客户使用竞争对手IP占相当比重，赢回这部分市场；鼓励使用宽带，提升总体价值等。

- V4(高值增长型):稳定客户，保护已有收入，增加优惠；同样，要赢回对手 IP 市场，鼓励使用宽带等。

## 5.4　改进算法的评估

### 5.4.1　算法运行环境

表 5-6　程序运行环境说明

| 环境 | 要求 |
|---|---|
| 操作系统（OS） | UNIX |
| 数据库（Database） | ORACLE9i |
| 开发工具 | C++、OCI |

### 5.4.2　分析数据范围及要求

本文分析的数据对象是：商业客户中的小型客户；其中数据对象在每一维度上的数据均映射到区间[0,1]之间。

本文所有的聚类结果均满足相同的误差函数 E 收敛条件，即满足下列条件聚类结束：

$$E(\text{本次迭代})-E(\text{前一次迭代})<0.01 \qquad (5\text{-}8)$$

### 5.4.3　算法效率对比与分析

本文对标准的 k-means 算法的改进主要有两个方面：1)初始聚类中心选取的改进；2)对距离运算的改进（基于 K-D 树的 k-means 算法），为了分清这两个方面改进方法的效果，本文将分别进行阐述。

(一)　改进的初始聚类中心选取方法对标准 k-means 算法的影响

在聚类算法中，初始聚类中心的选取直接影响到整个算法运行的效率和聚类结果的优劣。标准 k-means 算法在选取初始聚类中心的时候采用的是随机的方法，因此使得聚类的效率和质量都带有随机性。为了验证这种随机性的影响，本文分别对大小为 10000、100000 和 1000000 的数据集进行测试（如表 5-7 所示），得到结果如表 5-8 所示（为了直观地表现初始聚类中心和最终聚类中心的偏移，本文选取数据对象所有属性中的一个属性来做测试）。

表 5-7　初始聚类中心选取方法测试集

| 数据集编号 | 数据对象个数 | 均值 | 标准差 |
|---|---|---|---|
| 1 | 10000 | 0.077239 | 0.064891 |
| 2 | 100000 | 0.041871 | 0.037655 |
| 3 | 1000000 | 0.136323 | 0.075603 |

表 5-8　随机选取初始聚类中心测试结果

| 编号 | 数据集编号 | 初始聚类中心 | 聚类结果中心 | 迭代次数 | 时间(ms) |
|---|---|---|---|---|---|
| 1 | 1 | {0.010130, 0.006938, 0.013557, 0.007156, 0.010935} | {0.136383, 0.035104, 0.485508, 0.075358, 0.233034} | 40 | 25 |
| 2 | 1 | {0.340147, 0.280763, 0.240340, 0.225224, 0.208735} | {0.784471, 0.378808, 0.187150, 0.094312, 0.040882} | 16 | 36 |
| 3 | 1 | {0.121002, 0.113328, 0.094395, 0.093220, 0.088897} | {0.485487, 0.232841, 0.136273, 0.075341, 0.035104} | 23 | 57 |
| 4 | 2 | {0.010130, 0.006938, 0.013557, 0.007156, 0.010935} | {0.082225, 0.019385, 0.338419, 0.044258, 0.156020} | 45 | 1034 |
| 5 | 2 | {0.210890, 0.218282, 0.261456, 0.202526, 0.278239} | {0.048426, 0.091363, 0.180750, 0.020578, 0.401045} | 40 | 853 |
| 6 | 2 | {1.000000, 0.501410, 0.425563, 0.321084, 0.210890} | {null, 0.301592, 0.124413, 0.059505, 0.023233} | 44 | 837 |
| 7 | 3 | {0.010130, 0.006938, 0.013557, 0.007156, 0.010935} | {0.137421, 0.030199, 0.249632, 0.082901, 0.192939} | 39 | 8611 |
| 8 | 3 | {0.210890, 0.218282, 0.261456, 0.202526, 0.278239} | {0.103868, 0.163197, 0.219994, 0.039779, 0.281148} | 37 | 7789 |
| 9 | 3 | {1.000000, 0.501410, 0.425563, 0.321084, 0.210890} | {0.281139, 0.219991, 0.163194, 0.103866, 0.039778} | 48 | 10392 |

从表 5-8 中，我们可以发现三个问题：

(1) 算法迭代次数存在随机性，如对大小为 10000 的数据集随机选择 3 个初始聚类中心，在相同的误差函数要求条件下，迭代的次数分别为 40、16、23；

(2) 聚类运行时间存在随机性，如对大小为 1000000 的数据集随机选择 3 个初始聚类中心，在相同的误差函数要求条件下，算法的运行时间分别为 8611ms、7789ms、10392ms；

(3) 如果随机选择了比较极端的点作为初始聚类中心，最终可能导致聚类结果中一些集合数据极少甚至没有数据，如编号为 6 的测试结果数据，第一个聚类就是 NULL。这种现象也就导致了聚类结果的不合理。

对于很多大数据量的聚类要求（电信行业中很多数据集都超过上千万），很重要的一点就是效率可以量化并且性能稳定，显然随机选取初始聚类中心的方法因为其效率存在随机性而无法满足要求。实际上我们可以在分析数据分布的基础上，通过简单的方法得到更合理的初始聚类中心。本文采用均值-标准差方法选取的初始聚类中心，得到的测试结果如表5-9所示。

表 5-9　改进的初始聚类中心选取方法测试结果

| 编号 | 数据集编号 | 初始聚类中心 | 聚类结果中心 | 递归次数 | 时间(ms) |
|---|---|---|---|---|---|
| 1 | 1 | {0.038305, 0.064261, 0.090218, 0.116174, 0.142131} | {0.034997, 0.075081, 0.135741, 0.231700, 0.482218} | 25 | 30 |
| 2 | 2 | {0.019278, 0.034340, 0.049402, 0.064464, 0.079525} | {0.019364, 0.044200, 0.082097, 0.155606, 0.337580} | 32 | 736 |
| 3 | 3 | {0.090961, 0.121203, 0.151444, 0.181685, 0.211926} | {0.030262, 0.083032, 0.137584, 0.193093, 0.249733} | 18 | 3953 |

从表5-9中，我们可以看到与随机选取初始聚类中心方法相比，改进的初始聚类中心选取方法在递归次数和运行时间上都有了明显减少，更重要的是得到性能上的稳定；通过对比初始聚类中心和最终的聚类中心，我们还可以看出在改进的算法中聚类中心的偏移量相对较小，例如对1000000的数据集进行的测试，改进的方法聚类中心从{0.090961, 0.121203, 0.151444, 0.181685, 0.211926}偏移到{0.030262, 0.083032, 0.137584, 0.193093, 0.249733}，而随机选择初始聚类中心方法在比较差的条件下，聚类中心从{1.000000, 0.501410, 0.425563, 0.321084, 0.210890}偏移到{0.281139, 0.219991, 0.163194, 0.103866, 0.039778}，显然后者的偏离量要大的多。

(二)　基于K-D树的k-means算法的效率分析

标准的k-means算法在每一次迭代过程中都涉及到大量的距离运算，随着聚类数据量、维度数、聚类个数k的增加，标准的k-means算法在运行时间上也成倍增加。而基于K-D树的k-means算法则利用K-D树这种数据结构减少算法中的距离运算量，达到提高效率的目的。本文根据大小分别为10000、100000、1000000的3个数据集，测试在不同维度数、不同的聚类个数和相同的误差收敛要求的条件下，标准的k-means算法和基于K-D树的改进的k-means算法在每次迭代平均运行时间、总的运行时间方面的比较，最后给出改进算法的效率提高倍数，测试结果如表5-10所示。

表 5-10　标准的 k-means 算法和基于 K-D 树的改进的 k-means 算法的比较结果

| 编号 | 数据容量 | 维度 | k | 迭代次数 | 标准的 k_means 算法 | | 本文的算法 | | | 效率比 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 每次迭代时间(ms) | 总的运行时间(ms) | LeafSize | 每次迭代时间(ms) | 总的运行时间(ms) | |
| 1 | 10000 | 4 | 4 | 34 | 5 | 200 | 100 | 2 | 90 | 2.22 |
| 2 | 10000 | 4 | 16 | 87 | 17 | 1730 | 100 | 6 | 645 | 2.68 |
| 3 | 10000 | 16 | 4 | 42 | 19 | 828 | 100 | 11 | 536 | 1.54 |
| 4 | 10000 | 16 | 16 | 31 | 50 | 1578 | 100 | 26 | 902 | 1.75 |
| 5 | 10000 | 64 | 4 | 29 | 70 | 2064 | 100 | 66 | 1988 | 1.04 |
| 6 | 10000 | 64 | 16 | 43 | 191 | 8240 | 100 | 179 | 8010 | 1.03 |
| 7 | 100000 | 4 | 4 | 63 | 57 | 3638 | 100 | 8 | 747 | 4.87 |
| 8 | 100000 | 4 | 16 | 100 | 154 | 15427 | 100 | 28 | 3047 | 5.06 |
| 9 | 100000 | 16 | 4 | 13 | 186 | 2480 | 100 | 29 | 1306 | 1.90 |
| 10 | 100000 | 16 | 16 | 30 | 509 | 15351 | 100 | 66 | 2911 | 5.27 |
| 11 | 100000 | 64 | 4 | 37 | 718 | 26818 | 100 | 265 | 14515 | 1.85 |
| 12 | 100000 | 64 | 16 | 24 | 1920 | 46293 | 100 | 642 | 20043 | 2.31 |
| 13 | 1000000 | 4 | 4 | 98 | 539 | 54119 | 100 | 39 | 6827 | 7.93 |
| 14 | 1000000 | 4 | 16 | 95 | 1582 | 158436 | 100 | 135 | 16544 | 9.57 |
| 15 | 1000000 | 16 | 4 | 23 | 3950 | 91914 | 100 | 154 | 16241 | 5.66 |
| 16 | 1000000 | 16 | 16 | 42 | 5233 | 220367 | 100 | 510 | 33992 | 6.48 |
| 17 | 1000000 | 64 | 4 | 27 | 21225 | 579843 | 100 | 1990 | 115011 | 5.04 |
| 18 | 1000000 | 64 | 16 | 31 | 29569 | 919621 | 100 | 5052 | 217137 | 4.24 |

其中，效率提高(%)＝标准 k-means 算法运行时间/改进的算法运行时间。

为了直观表现测试结果，本文将表 5-10 中两种算法的运行时间数据按照测试编号的顺序绘制成图 5-11，随着测试编号的增加，影响算法效率的总的距离运算量也随之增大。

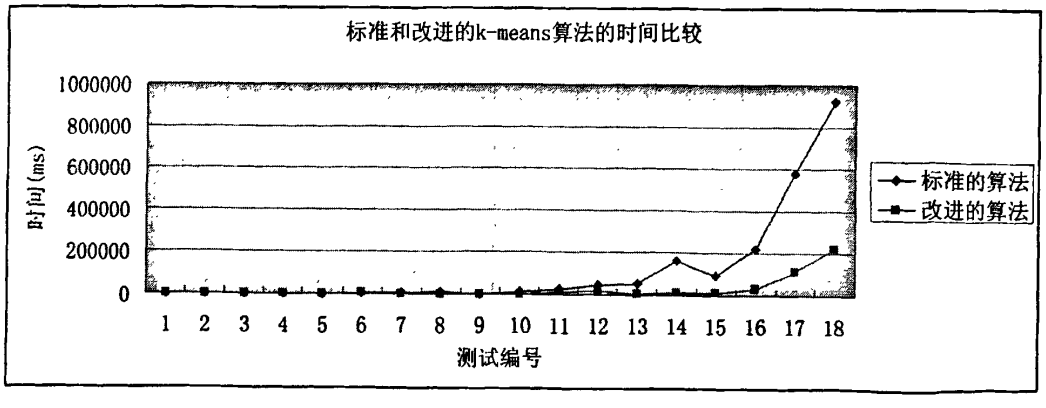


图 5-11 标准和改进的 k-means 算法的时间比较

从表 5-10 和图 5-11 中可以看出：

(1) 不论是标准的 k-means 算法还是改进的算法，算法的运行时间都随着数据集的增大、维数和 k 值的增加而增长。例如：在改进的 k-means 算法中，4 维、聚类个数为 4、大小为 10000 的数据集，聚类时间只需 90ms，而相同条件下对于 100000、1000000 的数据量，聚类时间则分别增加到 747ms 和 6827ms。

(2) 基于 K-D 树的 k-means 算法相对于标准的 k-means 算法在效率上有了明显的提高，平均提高 3-5 倍，最高的可提高一个数量级。从图 5-10 可以看出，当数据量较小的时候，标准的 k-means 算法的和改进的算法在运行时间上相差较少；当距离运算量成数量级增加时，改进的算法能节省相当可观的运行时间。

(3) 大多数情况下，迭代次数不超过 50 次即可满足误差函数的要求。

对于基于 K-D 树的 k-means 算法，LeafSize 大小的选择决定了算法的效率，如果 LeafSize 的值大于数据集大小，基于 K-D 树的 k-means 算法则退化成标准的 k-means 算法；如果 LeafSize 的值为 1，虽然叶子节点不再需要距离计算，但是建立 K-D 树、遍历 K-D 树以及内部节点修剪的代价也达到最大，算法的效率不一定是最高的，表 5-11 的测试结果也说明了 LeafSize 为 1 并不能取得最好的效率。合适的 LeafSize 必须在叶子节点的距离计算量和 K-D 树的建立、遍历和修剪之间取得平衡。图 5-12 列出在大小为 1000000 的数据集下，不同的 LeafSize 的选择对算法的影响。

表 5-11　不同 LeafSize 的选择对比

| 编号 | 维度 | k | LeafSize | 内部节点距离计算量 | 叶节点距离计算量 | 总的距离计算量 | 建 K-D 树时间(ms) | 运行时间(ms) |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 4 | 1000000 | 800 | 400151600 | 400152400 | 132 | 54119 |
| 2 | 4 | 4 | 1000 | 550644 | 44669954 | 45220598 | 2196 | 10063 |
| 3 | 4 | 4 | 100 | 1903928 | 15288313 | 17192241 | 2892 | 6827 |
| 4 | 4 | 4 | 1 | 13741972 | 0 | 13741972 | 5760 | 14820 |
| 5 | 16 | 16 | 1000000 | 1344 | 672254688 | 672256032 | 499 | 220367 |
| 6 | 16 | 16 | 1000 | 1963540 | 138012051 | 139975591 | 9693 | 69268 |
| 7 | 16 | 16 | 100 | 5864928 | 30084164 | 35949092 | 12526 | 33992 |
| 8 | 16 | 16 | 1 | 19737920 | 0 | 19737920 | 22859 | 48298 |

从表 5-11 可以看出，在相同的数据集条件下，维数值越大、聚类个数越多，总的距离计算量也成倍增加。这里总的距离计算量包括 K-D 树内部节点距离计算量和叶节点距离计算量两个部分，内部节点距离计算指的是修剪算法中从候选集到内部节点所代表的空间的最大值最小值计算，而叶节点距离计算和标准的 k-means 距离计算一样，即计算叶节点

中所有数据对象到各候选集的距离。

LeafSize 的选择需要反复的比较。对于大小为 1000000 的测试集，LeafSize 设置为 100 时，距离的计算量最少，算法的运行时间也最少。例如在维度为 4、聚类个数为 4 的条件下，不同的 LeafSize 的选择对算法效率的影响如图 5-12 所示。其中：LeafSize 为 1000000 的距离计算量为 $4\times10^7$，运行时间为 54s；随着 LeafSize 的减小，算法运行时间先是缓慢地减少，而后减少的速度变快；当 LeafSize 为 100 时，距离计算量为 $0.17\times10^7$，运行时间为 6.83s，这时算法的运行时间达到最短；随着 LeafSize 的继续减少，算法的运行时间又缓慢增加。因此，找到合适的 LeafSize 使得算法的运行时间最短至关重要。

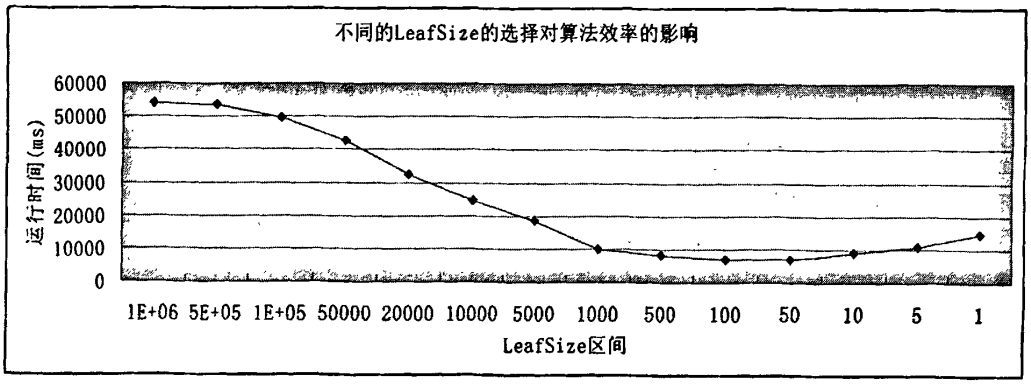


图 5-12 不同的 LeafSize 的选择对算法效率的影响

表 5-11 还列出不同的 LeafSize 条件下建立 K-D 树所需要的时间。可以看出在基于 K-D 树的 k-means 算法中，K-D 树的建立时间在算法总运行时间中的比重有的高达 1/3 甚至 1/2，如果能减少建立 K-D 树的时间，基于 K-D 树的 k-means 算法的效率还可以大大提高。

在测试过程中，我们还发现数据的分布对基于 K-D 树的 k-means 算法的效率也有很大影响。数据的跨度越大、区分度越高，基于 K-D 树的 k-means 算法效率也就越高。发现这些规律以后，我们在数据标准化过程中，就可以把数据标准化映射到比较大的区间，减少算法的运行时间。

## 5.5　本章小结

本章以理论联系实际，研究数据挖掘技术中的聚类方法在电信经营分析系统中的实际应用，是全文的核心章节。

本章按照标准的数据挖掘过程模型 CRISP-DM（商业理解—>数据理解—>数据准备—>建模—>评估），详细叙述客户细分的全过程。每一阶段的内容总结如下：

(1) 商业理解：介绍客户细分的目的；

(2) 数据理解和数据准备：介绍客户信息数据的来源、组成、数据处理流程，以及宽表的结构和用途等。

(3) 建模：包括客户细分总的流程、相关函数介绍，并详细分析每个模块（数据标准化、初始聚类中心选取、K-D 树创建、K-D 树遍历、修剪函数、叶节点聚类函数）的具体设计、实现、流程等。最后给出聚类的结果分析、各个分群的特征刻画以及相应分群的营销建议。

(4) 评估：本文对标准 k-means 算法的改进分为两个方面，1)初始聚类中心选取方法的改进，2)运用 K-D 树结构对标准 k-means 算法中距离计算的改进。为了区分出这两种改进的作用，本文对改进算法的评估也相应的分两个部分进行。这部分结合大量的实验数据，通过对比的方式，直观地体现出改进算法在效率和稳定性方面要大大优于标准的 k-means 算法。

# 第6章 总结和展望

电信经营分析系统的建设是一个逐步完善的过程。经过几年时间的发展，经营分析系统已经在企业的营销决策中发挥重要的作用。目前系统在数据仓库构建和OLAP分析方面具备很成熟的经验。

相对而言，数据挖掘技术在经营分析中的应用还处在起步阶段。有的地区还只是在实验期，结果并没有真正地运用来指导生产，有的地区甚至还没开展数据挖掘的工作。在电信行业已经实现的数据挖掘应用主要包括客户聚类分析、客户流失分析、客户价值分析、欺诈分析、交叉销售等。

电信客户聚类分析最常用的方法是k-means。本文提出的初始聚类中心的选取方法和基于K-D树的改进的k-means算法比标准的k-means算法在效率和稳定性上都有很大提升，更加适合于电信企业中大数据量的分析应用。

K-D树是二叉树，在实际应用中，我们还可以尝试用三叉树或四叉树代替K-D树来组织数据对象，目的都是为了减少k-means算法中的距离的计算次数，提高效率。另外，利用标准的k-means算法在循环几次后聚类中心相对变化较小的事实，可以利用上一次循环的信息来减少距离计算的数量，从而也实现提高效率、加快收敛的目的。

另外，在k-means算法的聚类个数k的选择上，可以运用以下方法：首先把数据压缩成可控制的小子集，而后运用统计聚类方法逐步地使小类合并成大类，接着再使这些类合并成更大的类，如此反复直至所希望的最小类数到达为止。这种方法的优点在于可以自动估计训练数据的最佳分类数目。这些都是我们今后的研究需要进行的工作。

随着竞争的激烈，在追求个性化的今天，用户的需求日新月异。要想提高客户的忠诚度，防止客户流失，就要以客户为中心，对不同类别的客户提供不同的服务，因此，客户细分的重要性就不言而喻了。随着数据挖掘技术的更加成熟和不断应用，经营分析系统必定会在企业的生产和决策中发挥更重要的作用。

# 附录一

宽表字段（价值部分）

| 字段名称 | 类型 | 字段说明 |
|---|---|---|
| CUST_ID | NUMBER(10) | 客户 ID |
| MB_AVG_TOTAL_FEE | NUMBER(10) | 总费用均值 |
| MB_AVG_TOTAL_FEE_FACT | NUMBER(10) | 实缴总费用均值 |
| MB_AVG_LOCAL_FEE | NUMBER(10) | 市话费均值 |
| MB_AVG_LONG_FEE | NUMBER(10) | 长话费均值 |
| MB_AVG_DATA_FEE | NUMBER(10) | 数据费用均值 |
| MB_AVG_HOUR_DATA_FEE | NUMBER(10) | 包时数据费用均值 |
| MB_AVG_EN_TOTAL_FEE | NUMBER(10) | 平均每号线总费用均值 |
| MB_AVG_EN_TOTAL_FEE_FACT | NUMBER(10) | 平均每号线实缴总费用均值 |
| MB_AVG_EN_LOCAL_FEE | NUMBER(10) | 平均每号线市话费均值 |
| MB_AVG_EN_SUBURB_FEE | NUMBER(10) | 平均每号线区间话费均值 |
| MB_AVG_EN_LONG_FEE | NUMBER(10) | 平均每号线长话费均值 |
| MB_AVG_EN_DATA_FEE | NUMBER(10) | 平均每号线数据费用均值 |
| MB_AVG_REMARK_FEE | NUMBER(10) | 优惠费均值 |
| MB_RATIO_REMARK_FEE | NUMBER(10,4) | 优惠费比例 |
| MB_AVG_INCITY_FEE | NUMBER(10) | 区内费均值 |
| MB_AVG_SUBURB_FEE | NUMBER(10) | 区间费均值 |
| MB_AVG_PREPAY | NUMBER(10) | 预付费市话费均值 |
| MB_AVG_OTHER_IP_FEE | NUMBER(10) | 非电信 IP 市话费均值 |
| MB_AVG_OTHER_LOCAL_FEE | NUMBER(10) | 他网固话区内+区间费均值 |
| MB_AVG_ISP_FEE | NUMBER(10) | 上网记次费均值 |
| MB_AVG_ISP_OTHER_FEE | NUMBER(10) | 他网记次费均值 |
| MB_AVG_ISP_163_FEE | NUMBER(10) | 163 记次费均值 |
| MB_AVG_ICP_FEE | NUMBER(10) | 上网信息费均值 |
| MB_AVG_MSG_FEE | NUMBER(10) | 信息费均值 |
| MB_AVG_MONTHLY_CHARGE | NUMBER(10) | 基本租费均值 |
| MB_AVG_VAS_FEE | NUMBER(10) | 程控新业务费均值 |
| MB_AVG_LD_FEE | NUMBER(10) | 直拨长话费均值 |
| MB_AVG_IP_FEE | NUMBER(10) | IP 长话费均值 |
| MB_AVG_LOCAL_IP_FEE | NUMBER(10) | 17909IP 长话费均值 |
| MB_AVG_INTRA_FEE | NUMBER(10) | 直拨国内费均值 |
| MB_AVG_INTER_FEE | NUMBER(10) | 直拨国际费均值 |
| MB_AVG_GAT_FEE | NUMBER(10) | 直拨港澳台费均值 |
| MB_AVG_IP_INTRA_FEE | NUMBER(10) | IP 国内费均值 |
| MB_AVG_IP909_INTRA_FEE | NUMBER(10) | 17909IP 国内费均值 |
| MB_AVG_IP_INTER_FEE | NUMBER(10) | IP 国际费均值 |

| | | |
|---|---|---|
| MB_AVG_IP909_INTER_FEE | NUMBER(10) | 17909IP 国际费均值 |
| MB_AVG_IP_GAT_FEE | NUMBER(10) | IP 港澳台费均值 |
| MB_AVG_IP909_GAT_FEE | NUMBER(10) | 17909IP 港澳台费均值 |
| MB_AVG_BROADBAND_FEE | NUMBER(10) | 宽带费用均值 |
| MB_AVG_ADSL_FEE | NUMBER(10) | Adsl 费用均值 |
| MB_AVG_LAN_FEE | NUMBER(10) | lan 费用均值 |
| MB_AVG_IN_FEE | NUMBER(10) | 城域网费用均值 |
| MB_AVG_NETELEMENT_FEE | NUMBER(10) | 网元费用均值 |
| MB_TREND_TOTAL_FEE | NUMBER(10) | 总费用趋势 |
| MB_TREND_LOCAL_FEE | NUMBER(10) | 市话费趋势 |
| MB_TREND_LONG_FEE | NUMBER(10) | 长话费趋势 |
| MB_TREND_DATA_FEE | NUMBER(10) | 数据费趋势 |
| MB_OPPO_TREND_TOTAL_FEE | NUMBER(10) | 总费用相对趋势 |
| MB_OPPO_TREND_LOCAL_FEE | NUMBER(10) | 市话费相对趋势 |
| MB_OPPO_TREND_SUBURB_FEE | NUMBER(10) | 区间费相对趋势 |
| MB_OPPO_TREND_LONG_FEE | NUMBER(10) | 长话费相对趋势 |
| MB_OPPO_TREND_DATA_FEE | NUMBER(10) | 数据费相对趋势 |
| MB_TREND_INCITY_FEE | NUMBER(10) | 区内费趋势 |
| MB_TREND_SUBURB_FEE | NUMBER(10) | 区间费趋势 |
| MB_TREND_ISP_FEE | NUMBER(10) | 上网记次费趋势 |
| MB_TREND_ISP_OTHER_FEE | NUMBER(10) | 他网记次费趋势 |
| MB_TREND_MSG_FEE | NUMBER(10) | 信息费趋势 |
| MB_TREND_OTHER_IP_FEE | NUMBER(10) | 非电信 IP 市话费趋势 |
| MB_TREND_LD_FEE | NUMBER(10) | 直拨长话费趋势 |
| MB_TREND_IPLD_FEE | NUMBER(10) | IP 长话费费趋势 |
| MB_TREND_NATION_FEE | NUMBER(10) | 直拨国内费趋势 |
| MB_TREND_INTER_FEE | NUMBER(10) | 直拨国际费趋势 |
| MB_TREND_GAT_FEE | NUMBER(10) | 直拨港澳台费趋势 |
| MB_TREND_IP_NATION_FEE | NUMBER(10) | IP 国内费趋势 |
| MB_TREND_IP_INTER_FEE | NUMBER(10) | IP 国际费趋势 |
| MB_TREND_IP_GAT_FEE | NUMBER(10) | IP 港澳台费趋势 |
| MB_TREND_HOUR_DATA_FEE | NUMBER(10) | 包时数据费用趋势 |
| MB_FLUCT_TOTAL_FEE | NUMBER(10,4) | 总费用波动 |
| MB_FLUCT_LOCAL_FEE | NUMBER(10,4) | 市话费波动 |
| MB_FLUCT_SUBURB_FEE | NUMBER(10,4) | 区间费波动 |
| MB_FLUCT_LONG_FEE | NUMBER(10,4) | 长话费波动 |
| MB_RATIO_LOCAL_FEE | NUMBER(10,4) | 市话费比例 |
| MB_RATIO_INCITY_FEE | NUMBER(10,4) | 区内费占市话费比例 |
| MB_RATIO_SUBURB_FEE | NUMBER(10,4) | 区间费占市话费比例 |
| MB_RATIO_PREPAY | NUMBER(10,4) | 预付费市话费占市话费比例 |
| MB_RATIO_ISP_FEE | NUMBER(10,4) | 上网记次费占市话费比例 |
| MB_RATIO_ISP_OTHER_FEE | NUMBER(10,4) | 他网记次费占上网记次费比例 |
| MB_RATIO_MSG_FEE | NUMBER(10,4) | 信息费比例 |
| MB_RATIO_MONTHLY_CHARGE | NUMBER(10,4) | 租费比例 |
| MB_RATIO_LONG | NUMBER(10,4) | 长话费比例 |

| | | |
|---|---|---|
| MB_RATIO_LD_FEE | NUMBER(10,4) | 直拨长话费占长话费比例 |
| MB_RATIO_IPLD_FEE | NUMBER(10,4) | IP 长话费费占长话费比例 |
| MB_RATIO_NATION_FEE | NUMBER(10,4) | 直拨国内费占长话费比例 |
| MB_RATIO_INTER_FEE | NUMBER(10,4) | 直拨国际费占长话费比例 |
| MB_RATIO_GAT_FEE | NUMBER(10,4) | 直拨港澳台费占长话费比例 |
| MB_RATIO_IP_NATION_FEE | NUMBER(10,4) | IP 国内费占长话费比例 |
| MB_RATIO_IP_INTER_FEE | NUMBER(10,4) | IP 国际费占长话费比例 |
| MB_RATIO_IP_GAT_FEE | NUMBER(10,4) | IP 港澳台费比例占长话费比例 |
| MB_RATIO_DATA_FEE | NUMBER(10,4) | 数据费用比例 |
| MB_RATIO_BROADBAND_FEE | NUMBER(10,4) | 宽带费用占数据费用比例 |
| MB_RATIO_NETELEMENT_FEE | NUMBER(10,4) | 网元费用占数据费用比例 |
| COM_AVG_CARD_FEE | NUMBER(10) | 卡类费均值 |
| COM_AVG_CARD_DIANXIN_FEE | NUMBER(10) | 电信卡类费均值 |
| COM_AVG_CARD_OTHER_FEE | NUMBER(10) | 非电信卡类费均值 |
| COM_RATIO_CARD_OTHER_FEE | NUMBER(10,4) | 非电信卡类费占卡类费比例 |
| COM_RATIO_CARD_FEE | NUMBER(10,4) | 卡类费用占总费用（含卡类）比例 |
| PRI_UNIT_INTRA | NUMBER(10) | 直拨国内长话单价（应收款/时长） |
| PRI_UNIT_INTER | NUMBER(10) | 直拨国际长话单价 |
| PRI_UNIT_GAT | NUMBER(10) | 直拨港澳台长话单价 |
| PRI_UNIT_IP_INTRA | NUMBER(10) | IP 国内长话单价 |
| PRI_UNIT_IP_INTER | NUMBER(10) | IP 国际长话单价 |
| PRI_UNIT_IP_GAT | NUMBER(10) | IP 港澳台长话单价 |

# 攻读硕士学位期间的学术论文

[1] 肖玉容, 奚建春, 廖国栋. 数据挖掘在电信经营分析中客户分群研究的应用[J]. 全国计算机新科技与计算机教育论文集, 2006(14), 317-320.

# 参考文献

以下按照正文引用的顺序列出参考文献：

[1]电信发展趋势及其对电信企业成功转型的启示[EB/OL].
http://www.gd-emb.com/addetail/id-2387.html,2006-03-22.

[2]舒华英.电信客户关怀业务的全过程设计[EB/OL].
http://news.xinhuanet.com/newmedia/2005-09/27/content_3551857.htm,2005-09-27.

[3]吴志勇,吴跃.数据挖掘在电信业中的应用研究[J].计算机应用,2005,12(增):213-214.

[4]Jiawei Han,Micheline Kamber.数据挖掘概念与技术[M].北京:机械工业出版社,2001,188-194.

[5]WH.H.Inmon.Building the Data Warehouse[M].New York:John Wiley & Sons,1996.

[6]亚信.数据挖掘交流探讨及 IM 应用初阶[Z]，2006.

[7]126 网站.数据挖掘入门（Ver 0.9）.http://datamining.126.com,2000.

[8]Michael J.A.Berry, Gordon.S.Linoff. Data Mining Techniques – For Marketing,Sales,and Customer Relationship Management[M]. 北京:机械工业出版社,2006-07,339.

[9]CRISP-DM 协会.CRISP-DM1.0 数据挖掘方法论指南[Z],2000.

[10]张祖昶. 一个基于数据挖掘技术的电信反欺诈系统的研究和实现[D].南京:南京邮电大学,2004.

[11](美)Olivia Parr rud.数据挖掘实践[M].北京:机械工业出版社,2003.

[12]张云涛,龚玲.数据挖掘原理与技术[M].北京:电子工业出版社,2004-04,123.

[13]George M.Marakas. Modern Data Warehousing,Mining,and Visualization – Core Concepts[M].北京:清华大学出版社,2004-10,78.

[14]陈文伟.数据仓库与数据挖掘教程[M].北京:清华大学出版社,2006-01,245-252.

[15]上海科技在线学习,数据挖掘教程[EB/OL].
http://www.stcsm.gov.cn/learning/lesson/xinxi/20021125/lesson-8.asp,2007-02.

[16]计算机世界,多媒体挖掘[EB/OL].http://www.ccw.com.cn/htm/center/topic/02-7-26.asp. 2007-02.

[17]朱明.数据挖掘[M].合肥:中国科学技术大学出版社,2002,129-140.

[18]Bing Liu, Wynne Hsu and Yiming Ma. Integrity Classification and Association Rule Mining[J], In Proc. Of the fouth Int. Conference on Knowledge Discovery & Data Mining. New York, New York.1998:80-86.

[19]U.M.Fayyad，G.Piatetsky-Shairo，and P.smith，R.Uthurusamy.Advances in knowledge discovery and data mining[M].AAAI/MIT Press，1996:440-443.

[20]J.S.Park，P.S.Yu and M.S.Chen.Mining association rules with adjustable Accuracy. In Proceedings of the sixth International Conference on Information and Knowledge management [J](CIKM'97).1997:151-160.

[21]R.J.Bayardo. Efficiently mining long patterns from databases[J]. In Proceedings ACM SIGMOD International Conference on Management of Data(SIGMOD'98), 1998:85-93.

[22]R.Agrawal,H.Mannila,R.Srikant,H.Toivonen,and A.Inkeri Verkamo.Fast discovery of association rules.In U.Fayyad,G.Piatetsky-Shapiro,P.Smith,and R.Uthurusamy,editors,Advances in knowledge discovery and data mining[M].AAAI/MIT Press,Menlo Park,CA,1996:307-328.

[23]丁继承.基于聚类分析的电信客户细分系统研究与设计[D].哈尔滨:哈尔滨工业大学,2006.

[24]Wu D, Hou Y T, Zhang Y Q. Transporting Real-time video over the Internet Challenges and Approaches[J].Proceeding of the IEEE ,2000,88(12):1855-1875.

[25]Fine Granularity Scalable.MPEG4 Standards[S].Beijing,2000,07.

[26]Moore A W. The anchors hierarchy: Using the triangle inequality to survive high dimensional data[J] In:Proc.UAI-2000;The Sixteenth Conference on Uncertainty in Artificial Intelligence,2000.

[27]Pena J M, Lozano J A, Larranaga P. An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm[J]. Pattern Recognition letters,1999,20:1027-1040.

[28]汪亚鸥.论电信客户细分模型的建设[D].四川:四川大学,2004.

[29]邵峰晶,于忠清.数据挖掘原理与算法[M].北京:中国水利水电出版社,2003,219-224.

[30]Bentley J L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM,1975,18(9):509-517.

[31]Hertor Garcia-Molina, Jeffrey D.Ullman, Jennifer Widom. Database Systems[M].北京:机械工业出版社,2003.8,441-444.

[32]张铭,赵海燕,王腾蛟.数据结构与算法[Z],北京大学信息科学与技术学院“数据结构与算法”教学小组.2007-02.

[33]Cormen T H, Leiserson C E, Rivest R L. Introduction to algo-[M].McGRaw-Hill Book Company,1990.

[34]张巧英.大唐电信经营分析系统解决方案[J/OL].
http://comm.ccidnet.com/art/1907/20030927/65655_1.html,2003-09-27.

[35]联创科技（南京）有限公司.江苏电信经营分析及决策支持系统解决方案建议[Z],2005-09.

[36]张宁,贾自艳,史忠植.数据仓库中 ETL 技术的研究[J].计算机工程与应用,2002,24:213-216.

[37]王成. DW、OLAP 和 DM 在电信渠道建设系统中的研究与应用[D].南京:南京邮电大学,2004.

[38]联创科技.江苏电信省级经营分析概要设计（数据库）[Z],2005-12-20.

[39]联创科技.江苏省内 MR 培训（数据准备）[Z],2005-05-15.

[40]龙志勇.数据挖掘在电信行业关系管理中的应用[EB/OL].
http://www.dmresearch.net/bbs/viewthread.php?tid=3582&extra=page%3D3,2005-09-20.

[41]李雄飞,李军.数据挖掘与知识发现[M].北京:高等教育出版社,2003-11,21.

# 致　谢

三年的研究生学习时间让我收获了很多。回首往事，我衷心地感谢所有关心我、帮助我的老师、同学和朋友。

感谢我的导师奚建春老师，奚老师不仅在学习和研究方面给我指导和帮助，更重要的是教会了我很多为人处世的道理，让我对诚实、责任、无私这些优秀品德有了更深刻的体会，这些都足以让我受用终生！

感谢我的家人、朋友一直以来对我默默的关心和支持！

感谢我的师兄王成、张祖昶、刘向军，给了我许多的帮助！

感谢审阅本论文的专家教授在百忙之中抽出时间给我提出宝贵的意见！