HZ BOOKS
华章IT

10余位数据挖掘领域资深专家和科研人员，10余年大数据挖掘咨询与实施经验结晶。

从数据挖掘的应用出发，以电力、航空、医疗、互联网、生产制造以及公共服务等行业真实案例为主线，深入浅出介绍Python数据挖掘建模过程，实践性极强。

大数据技术丛书

Python Practice of Data Analysis and Mining

Python数据分析与挖掘实战

张良均 王路 谭立云 苏剑林◎等著

机械工业出版社
China Machine Press

大数据技术丛书

Python Practice of Data Analysis and Mining

Python数据分析与挖掘实战

张良均 王 路 谭立云 苏剑林 云伟标 刘名军
杨 坦 肖 刚 樊 哲 廖晓霞 周 龙 焦正升 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Python 数据分析与挖掘实战 / 张良均等著 . —北京：机械工业出版社，2015.12
（大数据技术丛书）

ISBN 978-7-111-52123-5

I. P… II. 张… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字（2015）第 264170 号

Python 数据分析与挖掘实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：高婧雅 责任校对：殷 虹
印 刷：北京诚信伟业印刷有限公司 版 次：2016 年 1 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：21.75
书 号：ISBN 978-7-111-52123-5 定 价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

Preface 前　言

为什么要写这本书

LinkedIn 对全球超过 3.3 亿用户的工作经历和技能进行分析后得出，目前最炙手可热的 25 项技能中，数据挖掘排名第一。那么数据挖掘是什么？

数据挖掘是从大量数据（包括文本）中挖掘出隐含的、先前未知的、对决策有潜在价值的关系、模式和趋势，并用这些知识和规则建立用于决策支持的模型，提供预测性决策支持的方法、工具和过程。数据挖掘有助于企业发现业务的趋势，揭示已知的事实，预测未知的结果，因此“数据挖掘”已成为企业保持竞争力的必要方法。

但跟国外相比，由于我国信息化程度不太高，企业内部信息不完整，零售业、银行、保险和证券等对数据挖掘的应用并不理想。但随着市场竞争的加剧，各行业对数据挖掘技术的需求越来越强烈，可以预计，未来几年各行业的数据分析应用一定会从传统的统计分析发展到大规模数据挖掘应用。在大数据时代，数据过剩、人才短缺，数据挖掘专业人才的培养又需要专业知识和职业经验积累。本书注重数据挖掘理论与项目案例实践相结合，可以让读者获得真实的数据挖掘学习与实践环境，更快、更好地学习数据挖掘知识与积累职业经验。

总的来说，随着云时代的来临，大数据技术将具有越来越重要的战略意义。大数据已经渗透到每一个行业和业务职能领域，逐渐成为重要的生产要素，人们对于海量数据的运用预示着新一轮生产率增长和消费者盈余浪潮的到来。大数据分析技术将帮助企业用户在合理时间内攫取、管理、处理、整理海量数据，为企业经营决策提供帮助。大数据分析作为数据存储和挖掘分析的前沿技术，广泛应用于物联网、云计算和移动互联网等战略性新兴产业。虽然大数据目前在国内还处于初级阶段，但是其商业价值已经显现出来，特别是有实践经验的大数据分析人才更是各企业争夺的热门。为了满足日益增长的大数据分析人才需求，很多大学开始尝试开设不同程度的大数据分析课程。“大数据分析”作为大数据时代的核心技术，必将成为高校数学与统计学专业的重要课程之一。

本书特色

本书从实践出发，结合大量数据挖掘工程案例及教学经验，以真实案例为主线，深入浅出地介绍数据挖掘建模过程中的有关任务：数据探索、数据预处理、分类与预测、聚类分析、时序预测、关联规则挖掘、智能推荐和偏差检测等。因此，图书的编排以解决某个应用的挖掘目标为前提，先介绍案例背景提出挖掘目标，再阐述分析方法与过程，最后完成模型构建。在介绍建模过程的同时穿插操作训练，把相关的知识点嵌入相应的操作过程中。为方便读者轻松地获取真实的实验环境，本书使用目前在数据科学领域非常热门的 Python 语言对样本数据进行处理以进行挖掘建模。

根据读者对案例的理解，本书配套提供真实的原始样本数据文件，读者可以从“泰迪杯”全国大学生数据挖掘竞赛网站（http://www.tipdm.org/ts/661.jhtml）免费下载。另外，为方便教师授课，本书还特意提供了建模阶段的过程数据文件、Python 语言代码程序和 PPT 课件，以及基于 Python、SAS、SPSS Modeler 等上机实验环境下的数据挖掘各阶段程序 / 模型及相关代码，读者可通过本书“勘误和支持”中提供的联系方式咨询获取。

本书适用对象

（1）开设数据挖掘课程的高校教师和学生

目前，国内不少高校将数据挖掘引入本科教学中，在数学、计算机、自动化、电子信息和金融等专业开设了数据挖掘技术相关课程，但目前这一课程的教学仍然主要限于理论介绍。单纯的理论教学过于抽象，学生理解起来往往比较困难，教学效果也不甚理想。本书提供的基于实战案例和建模实践的教学，能够使教师充分发挥互动性和创造性，理论联系实际，使教师获得最佳的教学效果。

（2）需求分析及系统设计人员

需求分析及系统设计人员可以在理解数据挖掘原理与建模过程的基础上，结合数据挖掘案例完成精确营销、客户分群、交叉销售、流失分析、客户信用记分、欺诈发现和智能推荐等数据挖掘应用的需求分析和设计。

（3）数据挖掘开发人员

数据挖掘开发人员可以在理解数据挖掘应用需求和设计方案的基础上，结合本书提供的基于第三方接口快速完成数据挖掘应用的编程实现。

（4）进行数据挖掘应用研究的科研人员

许多科研院所为了更好地对科研工作进行管理，纷纷开发了适应自身特点的科研业务管理系统，并在使用过程中积累了大量的科研信息数据。但是，这些科研业务管理系统一般没有对数据进行深入分析，并没有对数据所隐藏的价值进行充分挖掘和利用。科研人员需要通过数据挖掘建模工具及有关方法论来深挖科研信息的价值，从而提高科研水平。

（5）关注高级数据分析的人员

业务报告和商业智能解决方案对了解过去和现在的状况可能是非常有用的。但是，数据挖掘的预测分析解决方案还能使关注高级数据分析的人员预见未来的发展状况，使他们的机构能够先发制人，而不是处于被动。因为数据挖掘的预测分析解决方案将复杂的统计方法和机器学习技术应用到数据之中，通过使用预测分析技术来揭示隐藏在交易系统或企业资源计划（ERP）、结构数据库和普通文件中的模式与趋势，从而为这类人员的决策提供科学依据。

如何阅读本书

本书共 15 章，分两篇：基础篇和实战篇。基础篇介绍了数据挖掘的基本原理，实战篇介绍了一个个真实案例，通过对案例深入浅出的剖析，使读者在不知不觉中通过案例实践获得数据挖掘项目经验，同时快速领悟看似难懂的数据挖掘理论。读者在阅读过程中，应充分利用随书配套的案例建模数据，借助相关的数据挖掘建模工具，通过上机实验快速理解相关知识与理论。

基础篇（第 1 ～ 5 章），第 1 章的主要内容是数据挖掘概述；第 2 章对 Python 以及本书所用到的数据挖掘建模库进行了简明扼要的说明；第 3 章、第 4 章和第 5 章对数据挖掘的建模过程，包括数据探索、数据预处理及挖掘建模的常用算法与原理进行介绍。

实战篇（第 6 ～ 15 章），重点对数据挖掘技术在电力、航空、医疗、互联网、生产制造以及公共服务等行业的应用进行分析。在案例结构组织上，本书是按照先介绍案例背景与挖掘目标，再阐述分析方法与过程，最后完成模型构建的顺序进行的，在建模过程的关键环节穿插程序实现代码。最后通过上机实践，加深对数据挖掘技术在案例应用中的理解。

勘误和支持

除封面署名外，参加本书编写工作的还有杨坦、肖刚、刘名军、樊哲、廖晓霞、周龙、焦正升等。由于笔者的水平有限，加之编写时间仓促，书中难免会出现错误或者不准确的地方，恳请读者批评指正。为此，读者可通过作者微信公众号 TipDM（微信号：TipDataMining）、TipDM 官网（www.tipdm.com）反馈有关问题。也可通过热线电话（40068-40020）或企业 QQ（40068-40020）进行在线咨询。

读者可以将书中的错误及遇到的任何问题反馈给我们，我们将尽量在线上为读者提供最满意的解答。本书的全部建模数据文件及源程序，可以从“泰迪杯”全国大学生数据挖掘竞赛网站（www.tipdm.org）下载，我们会将相应内容的更新及时发布出来。如果您有更多的宝贵意见，欢迎发送邮件至邮箱 13560356095@qq.com，期待能够得到您的真挚反馈。

致谢

在本书编写过程中，得到了广大企事业单位及科研人员的大力支持！在此谨向中国电力科学研究院、广东电力科学研究院、广西电力科学研究院、广东电信规划设计院、珠江 / 黄海水产研究所、轻工业环境保护研究所、华南师范大学、广东工业大学、广东技术师范学院、南京中医药大学、华南理工大学、湖南师范大学、韩山师范学院、广东石油化工学院、中山大学、广州泰迪智能科技有限公司、武汉泰迪智慧科技有限公司等单位给予支持的专家与师生致以深深的谢意。

本书得到华北科技学院“应用数学”校级重点学科建设项目资助（项目编号 hkxjzd 201402），同时在本书的编辑和出版过程中还得到了参与“泰迪杯”全国大学生数据挖掘建模竞赛（http://www.tipdm.org）的众多师生，以及机械工业出版社杨福川、高婧雅等人的无私帮助与支持，在此一并表示感谢。

张良均

Contents 目　录

前　言

基　础　篇

第 1 章　数据挖掘基础 ····· 2

1.1　某知名连锁餐饮企业的困惑 ····· 2
1.2　从餐饮服务到数据挖掘 ····· 3
1.3　数据挖掘的基本任务 ····· 4
1.4　数据挖掘建模过程 ····· 4
1.4.1　定义挖掘目标 ····· 4
1.4.2　数据取样 ····· 5
1.4.3　数据探索 ····· 6
1.4.4　数据预处理 ····· 7
1.4.5　挖掘建模 ····· 7
1.4.6　模型评价 ····· 7
1.5　常用的数据挖掘建模工具 ····· 7
1.6　小结 ····· 9

第 2 章　Python 数据分析简介 ····· 10

2.1　搭建 Python 开发平台 ····· 12
2.1.1　所要考虑的问题 ····· 12
2.1.2　基础平台的搭建 ····· 12
2.2　Python 使用入门 ····· 13
2.2.1　运行方式 ····· 14
2.2.2　基本命令 ····· 15
2.2.3　数据结构 ····· 17
2.2.4　库的导入与添加 ····· 20
2.3　Python 数据分析工具 ····· 22
2.3.1　Numpy ····· 23
2.3.2　Scipy ····· 24
2.3.3　Matplotlib ····· 24
2.3.4　Pandas ····· 26
2.3.5　StatsModels ····· 27
2.3.6　Scikit-Learn ····· 28
2.3.7　Keras ····· 29
2.3.8　Gensim ····· 30
2.4　配套资源使用设置 ····· 31
2.5　小结 ····· 32

第 3 章　数据探索 ····· 33

3.1　数据质量分析 ····· 33
3.1.1　缺失值分析 ····· 34
3.1.2　异常值分析 ····· 34
3.1.3　一致性分析 ····· 37
3.2　数据特征分析 ····· 37
3.2.1　分布分析 ····· 37
3.2.2　对比分析 ····· 40
3.2.3　统计量分析 ····· 41

3.2.4 周期性分析 ····· 44
3.2.5 贡献度分析 ····· 45
3.2.6 相关性分析 ····· 47
3.3 Python 主要数据探索函数 ····· 50
3.3.1 基本统计特征函数 ····· 50
3.3.2 拓展统计特征函数 ····· 53
3.3.3 统计作图函数 ····· 54
3.4 小结 ····· 59

第 4 章 数据预处理 ····· 60

4.1 数据清洗 ····· 60
4.1.1 缺失值处理 ····· 60
4.1.2 异常值处理 ····· 64
4.2 数据集成 ····· 64
4.2.1 实体识别 ····· 64
4.2.2 冗余属性识别 ····· 65
4.3 数据变换 ····· 65
4.3.1 简单函数变换 ····· 65
4.3.2 规范化 ····· 66
4.3.3 连续属性离散化 ····· 68
4.3.4 属性构造 ····· 70
4.3.5 小波变换 ····· 71
4.4 数据规约 ····· 74
4.4.1 属性规约 ····· 74
4.4.2 数值规约 ····· 77
4.5 Python 主要数据预处理函数 ····· 80
4.6 小结 ····· 81

第 5 章 挖掘建模 ····· 83

5.1 分类与预测 ····· 83
5.1.1 实现过程 ····· 83
5.1.2 常用的分类与预测算法 ····· 84
5.1.3 回归分析 ····· 85
5.1.4 决策树 ····· 89
5.1.5 人工神经网络 ····· 95
5.1.6 分类与预测算法评价 ····· 100
5.1.7 Python 分类预测模型特点 ····· 103
5.2 聚类分析 ····· 104
5.2.1 常用聚类分析算法 ····· 104
5.2.2 K-Means 聚类算法 ····· 105
5.2.3 聚类分析算法评价 ····· 111
5.2.4 Python 主要聚类分析算法 ····· 111
5.3 关联规则 ····· 113
5.3.1 常用关联规则算法 ····· 114
5.3.2 Apriori 算法 ····· 114
5.4 时序模式 ····· 119
5.4.1 时间序列算法 ····· 120
5.4.2 时间序列的预处理 ····· 120
5.4.3 平稳时间序列分析 ····· 122
5.4.4 非平稳时间序列分析 ····· 124
5.4.5 Python 主要时序模式算法 ····· 132
5.5 离群点检测 ····· 134
5.5.1 离群点检测方法 ····· 135
5.5.2 基于模型的离群点检测方法 ····· 136
5.5.3 基于聚类的离群点检测方法 ····· 138
5.6 小结 ····· 141

实 战 篇

第 6 章 电力窃漏电用户自动识别 ····· 144

6.1 背景与挖掘目标 ····· 144
6.2 分析方法与过程 ····· 147
6.2.1 数据抽取 ····· 148
6.2.2 数据探索分析 ····· 148
6.2.3 数据预处理 ····· 151
6.2.4 构建专家样本 ····· 156
6.2.5 模型构建 ····· 157

6.3 上机实验……161
6.4 拓展思考……162
6.5 小结……163

第 7 章 航空公司客户价值分析……164

7.1 背景与挖掘目标……164
7.2 分析方法与过程……166
7.2.1 数据抽取……168
7.2.2 数据探索分析……168
7.2.3 数据预处理……169
7.2.4 模型构建……173
7.3 上机实验……177
7.4 拓展思考……178
7.5 小结……179

第 8 章 中医证型关联规则挖掘……180

8.1 背景与挖掘目标……180
8.2 分析方法与过程……181
8.2.1 数据获取……183
8.2.2 数据预处理……186
8.2.3 模型构建……190
8.3 上机实验……193
8.4 拓展思考……194
8.5 小结……194

第 9 章 基于水色图像的水质评价……195

9.1 背景与挖掘目标……195
9.2 分析方法与过程……195
9.2.1 数据预处理……197
9.2.2 模型构建……199
9.2.3 水质评价……201
9.3 上机实验……202
9.4 拓展思考……202
9.5 小结……203

第 10 章 家用电器用户行为分析与事件识别……204

10.1 背景与挖掘目标……204
10.2 分析方法与过程……205
10.2.1 数据抽取……206
10.2.2 数据探索分析……207
10.2.3 数据预处理……207
10.2.4 模型构建……217
10.2.5 模型检验……219
10.3 上机实验……220
10.4 拓展思考……221
10.5 小结……222

第 11 章 应用系统负载分析与磁盘容量预测……223

11.1 背景与挖掘目标……223
11.2 分析方法与过程……225
11.2.1 数据抽取……226
11.2.2 数据探索分析……226
11.2.3 数据预处理……227
11.2.4 模型构建……229
11.3 上机实验……235
11.4 拓展思考……236
11.5 小结……237

第 12 章 电子商务网站用户行为分析及服务推荐……238

12.1 背景与挖掘目标……238
12.2 分析方法与过程……240
12.2.1 数据抽取……242
12.2.2 数据探索分析……244
12.2.3 数据预处理……251
12.2.4 模型构建……256
12.3 上机实验……266

12.4 拓展思考……267
12.5 小结……269

第 13 章 财政收入影响因素分析及预测模型……270

13.1 背景与挖掘目标……270
13.2 分析方法与过程……272
13.2.1 灰色预测与神经网络的组合模型……273
13.2.2 数据探索分析……274
13.2.3 模型构建……277
13.3 上机实验……294
13.4 拓展思考……295
13.5 小结……296

第 14 章 基于基站定位数据的商圈分析……297

14.1 背景与挖掘目标……297
14.2 分析方法与过程……299
14.2.1 数据抽取……299
14.2.2 数据探索分析……299
14.2.3 数据预处理……301
14.2.4 模型构建……304
14.3 上机实验……308
14.4 拓展思考……309
14.5 小结……309

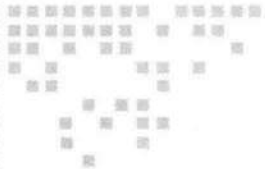
第 15 章 电商产品评论数据情感分析……310

15.1 背景与挖掘目标……310
15.2 分析方法与过程……310
15.2.1 评论数据采集……311
15.2.2 评论预处理……314
15.2.3 文本评论分词……320
15.2.4 模型构建……320
15.3 上机实验……333
15.4 拓展思考……334
15.5 小结……335

参考文献……336

基 础 篇

- 第 1 章　数据挖掘基础
- 第 2 章　Python 数据分析简介
- 第 3 章　数据探索
- 第 4 章　数据预处理
- 第 5 章　挖掘建模



Chapter 1 第 1 章

数据挖掘基础

1.1 某知名连锁餐饮企业的困惑

国内某餐饮连锁有限公司（以下简称 T 餐饮）成立于 1998 年，主要经营粤菜，兼顾湘菜、川菜等综合菜系。至今已经发展成为在国内具有一定知名度、美誉度，多品牌、立体化的大型餐饮连锁企业。员工 1000 多人，拥有 16 家直营分店，经营总面积近 13 000 平方米，年营业额近亿元。其旗下各分店均坐落在繁华市区主干道，雅致的装潢，配之以精致的饰品、灯具、器物，出品精美，服务规范。

近年来餐饮行业面临较为复杂的市场环境，与其他行业一样，餐饮企业都遇到了原材料成本升高、人力成本升高、房租成本升高等问题，这也使得整个行业的利润急剧下降。人力成本和房租成本的上升是必然趋势，如何在保持产品质量的同时提高企业效率，成为了 T 餐饮企业急需解决的问题。从 2000 年开始，T 餐饮企业通过加强信息化管理来提高效率，目前已上线的管理系统如下。

（1）客户关系管理系统

客户关系管理系统详细记录了每位客人的喜好，为顾客提供个性化服务，满足客户个性化需求。通过客户关怀，提高客户的忠诚度。例如，企业能随时查询今天哪位客人过生日或其他纪念日，根据客人的价值分类进行相应关怀，如送鲜花、生日蛋糕和寿面等。通过本系统，还可对客户行为进行深入分析，包括客户价值分析、新客户分析与发展，并根据其价值情况提供给管理者，为企业提供决策支持。

（2）前厅管理系统

前厅管理系统通过掌上电脑无线点菜方式，改变了传统“饭店点菜、下单、结账一支笔、一张纸，服务员来回跑的局面”，快速完成点菜过程。通过厨房自动送达信息，服务员的写

菜速度加快，不需要再通过手写，同时传菜部也轻松不少，菜单会通过电脑自动打印出来，差错率降低，也不存在厨房人员看不懂服务员字迹而搞错的问题。

（3）后厨管理系统

信息化技术可实现后厨与前厅沟通无障碍，客人菜单瞬间传到厨房。服务员只需单击掌上电脑的发送键，客人的菜单即被传送到收银管理系统中，由系统的电脑发出指令，设在厨房等处的打印机立即打印出相应的菜单，厨师按单做菜。与此同时，收银台也打印出一张同样的菜单放在客人桌上，以备客人查询以及作结账凭据，使客人明明白白地消费。

（4）财务管理系统

财务管理系统完成销售统计、销售分析、财务审计，实现对日常经营销售的管理。通过报表，企业管理者很容易掌握前台的销售情况，从而达到对财务的控制。通过表格和图形显示餐厅的销售情况，如菜品排行榜、日客户流量、日销售收入分析等；通过统计每天的出菜情况，我们可以了解哪些是滞销菜，哪些是畅销菜，从而了解顾客的品位，有针对性地制定出一套既适合餐饮企业发展又能迎合顾客品位的菜肴体系和定价策略。

（5）物资管理系统

物资管理系统主要完成对物资的进销存，实际上就是一套融采购管理（入库、供应商管理、账款管理）、销售（通过配菜卡与前台销售联动）、盘存为一体的物流管理系统。对于连锁企业，还涉及统一配送管理等。

通过以上信息化的建设，T 餐饮已经积累了大量的历史数据，有没有一种方法可帮助企业从这些数据中洞察商机，提取价值？在同质化的市场竞争中，怎样找到一些市场以前并不存在的“捡漏”和“补缺”呢？

1.2　从餐饮服务到数据挖掘

企业经营最大的目的就是盈利，而餐饮业企业盈利的核心就是其菜品和顾客，也就是其提供的产品和服务对象。企业经营者每天都在想推出什么样的菜系和种类能吸引更多的顾客，究竟顾客各自的喜好是什么，在不同的时段是不是有不同的菜品畅销，当把几种不同的菜品组合在一起推出时是不是能够得到更好的效果，未来一段时间菜品原材料应该采购多少……

T 餐饮的经营者想尽快地解决这些疑问，使自己的企业更加符合现有顾客的口味，吸引更多的新顾客，又能根据不同的情况和环境转换自己的经营策略。T 餐饮在经营过程中，通过分析历史数据，总结出一些行之有效的经验。

- 在点餐过程中，由有经验的服务员根据顾客特点进行菜品推荐，一方面可提高菜品的销量，另一方面可减少客户点餐的时间和频率，提高用户体验。
- 根据菜品历史销售情况，综合考虑节假日、气候和竞争对手等影响因素，对菜品销量进行预测，以便餐饮企业提前准备原材料。

- 定期对菜品销售情况进行统计，分类统计出好评菜和差评菜，为促销活动和新菜品推出提供支持。
- 根据就餐频率和金额对顾客的就餐行为进行评分，筛选出优质客户，定期回访和送去关怀。

上述措施的实施都依赖于企业已有业务系统中保存的数据，但是目前从这些数据中获得有关产品和客户的特点以及能够产生价值的规律更多依赖于管理人员的个人经验。如果有一套工具或系统，能够从业务数据中自动或半自动地发现相关的知识和解决方案，这将极大地提高企业的决策水平和竞争能力。这种从数据中“淘金”，从大量数据（包括文本）中挖掘出隐含的、未知的、对决策有潜在价值的关系、模式和趋势，并用这些知识和规则建立用于决策支持的模型，提供预测性决策支持的方法、工具和过程，就是**数据挖掘**；它是利用各种分析工具在大量数据中寻找其规律和发现模型与数据之间关系的过程，是统计学、数据库技术和人工智能技术的综合。

这种分析方法可避免“人治”的随意性，避免企业管理仅依赖个人领导力的风险和不确定性，实现精细化营销与经营管理。

1.3 数据挖掘的基本任务

数据挖掘的基本任务包括利用分类与预测、聚类分析、关联规则、时序模式、偏差检测、智能推荐等方法，帮助企业提取数据中蕴含的商业价值，提高企业的竞争力。

对餐饮企业而言，数据挖掘的基本任务是从餐饮企业采集各类菜品销量、成本单价、会员消费、促销活动等内部数据，以及天气、节假日、竞争对手以及周边商业氛围等外部数据；之后利用数据分析手段，实现菜品智能推荐、促销效果分析、客户价值分析、新店选点优化、热销 / 滞销菜品分析和销量趋势预测；最后将这些分析结果推送给餐饮企业管理者及有关服务人员，为餐饮企业降低运营成本、增加盈利能力、实现精准营销、策划促销活动等提供智能服务支持。

1.4 数据挖掘建模过程

从本节开始，将以餐饮行业的数据挖掘应用为例来详细介绍数据挖掘的建模过程，如图 1-1 所示。

1.4.1 定义挖掘目标

针对具体的数据挖掘应用需求，首先要明确本次的挖掘目标是什么？系统完成后能达到什么样的效果？因此，我们必须分析应用领域，包括应用中的各种知识和应用目标，了解相关领域的情况，熟悉背景知识，弄清用户需求。要想充分发挥数据挖掘的价值，必须对目标

有一个清晰明确的定义，即决定到底想干什么。

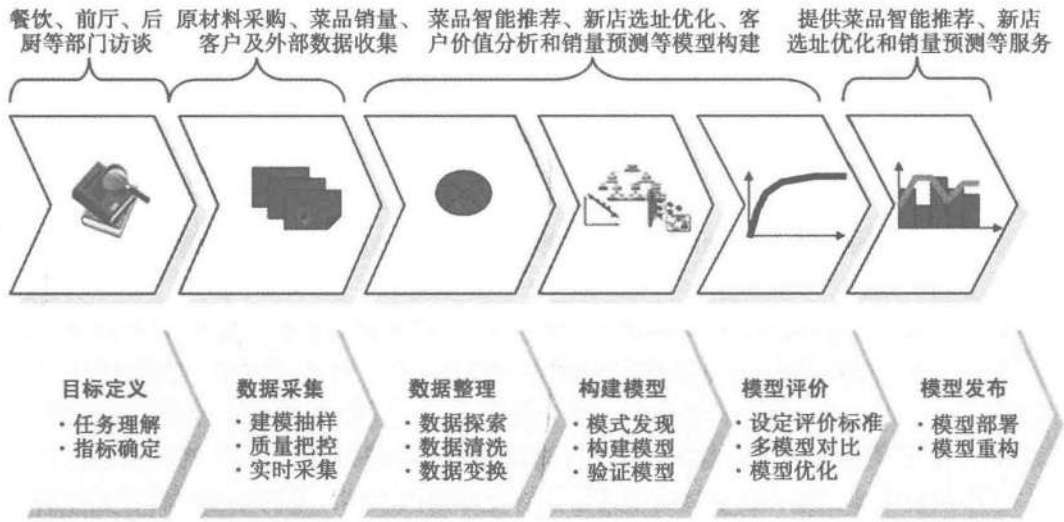


图 1-1　餐饮行业数据挖掘建模过程

针对餐饮行业的数据挖掘应用，可定义如下挖掘目标。

- 实现动态菜品智能推荐，帮助顾客快速发现自己感兴趣的菜品，同时确保推荐给顾客的菜品也是餐饮企业所期望的，实现餐饮消费者和餐饮企业的双赢。
- 对餐饮客户进行细分，了解不同客户的贡献度和消费特征，分析哪些客户是最有价值的，哪些是最需要关注的，对不同价值的客户采取不同的营销策略，将有限的资源投放到最有价值的客户身上，实现精准化营销。
- 基于菜品历史销售情况，综合考虑节假日、气候和竞争对手等影响因素，对菜品销量进行趋势预测，方便餐饮企业准备原材料。
- 基于餐饮大数据，优化新店选址，并对新店所在位置的潜在顾客口味偏好进行分析，以便及时进行菜式调整。

1.4.2　数据取样

在明确了需要进行数据挖掘的目标后，接下来就需要从业务系统中抽取出一个与挖掘目标相关的样本数据子集。抽取数据的标准，一是相关性，二是可靠性，三是有效性，而不是动用全部企业数据。通过对数据样本的精选，不仅能减少数据处理量，节省系统资源，还可以使我们想要寻找的规律性更加凸显出来。

进行数据取样，一定要严把质量关。在任何时候都不能忽视数据的质量，即使是从一个数据仓库中进行数据取样，也不要忘记检查其质量。因为数据挖掘是要探索企业运作的内在规律性，原始数据有误，就很难从中探索规律性。若真的从中还探索出来了什么“规律性”，

再依此去指导工作，则很可能会造成误导。若从正在运行的系统中进行数据取样，更要注意数据的完整性和有效性。

衡量取样数据质量的标准如下。

1）资料完整无缺，各类指标项齐全。

2）数据准确无误，反映的都是正常（而不是异常）状态下的水平。

对获取的数据，可再从中进行抽样操作。抽样的方式是多种多样的，常见的方式如下。

- ❑ 随机抽样：在采用随机抽样方式时，数据集中的每一组观测值都有相同的被抽样的概率。如按 10% 的比例对一个数据集进行随机抽样，则每一组观测值都有 10% 的机会被取到。
- ❑ 等距抽样：如按 5% 的比例对一个有 100 组观测值的数据集进行等距抽样，则有 100 / 5 = 20，等距抽样方式是取第 20、40、60、80 和第 100 这 5 组观测值。
- ❑ 分层抽样：在这种抽样操作时，首先将样本总体分成若干层次（或者说分成若干个子集）。在每个层次中的观测值都具有相同的被选用的概率，但对不同的层次可设定不同的概率。这样的抽样结果通常具有更好的代表性，进而使模型具有更好的拟合精度。
- ❑ 从起始顺序抽样：这种抽样方式是从输入数据集的起始处开始抽样。抽样的数量可以给定一个百分比，或者直接给定选取观测值的组数。
- ❑ 分类抽样：在前述几种抽样方式中，并不考虑抽取样本的具体取值。分类抽样则依据某种属性的取值来选择数据子集，如按客户名称分类、按地址区域分类等。分类抽样的选取方式就是前面所述的几种方式，只是抽样以类为单位。

基于上节定义的针对餐饮行业的挖掘目标，需从客户关系管理系统、前厅管理系统、后厨管理系统、财务管理系统和物资管理系统中抽取用于建模和分析的餐饮数据，主要内容如下。

1）餐饮企业信息：名称、位置、规模、联系方式，以及部门、人员、角色等。

2）餐饮客户信息：姓名、联系方式、消费时间、消费金额等。

3）餐饮企业菜品信息：菜品名称、菜品单价、菜品成本、所属部门等。

4）菜品销量数据：菜品名称、销售日期、销售金额、销售份数。

5）原材料供应商资料及商品数据：供应商姓名、联系方式、商品名称、客户评价信息。

6）促销活动数据：促销日期、促销内容、促销描述。

7）外部数据，如天气、节假日、竞争对手以及周边商业氛围等。

1.4.3 数据探索

前面所叙述的数据取样，多少是带着人们对如何实现数据挖掘目标的先验认识进行操作的。当我们拿到了一个样本数据集后，它是否达到我们原来设想的要求；样本中有没有什么明显的规律和趋势；有没有出现从未设想过的数据状态；属性之间有什么相关性；它们可区分成怎样一些类别……，这都是要探索的内容。

对所抽取的样本数据进行探索、审核和必要的加工处理，是保证最终的挖掘模型的质量所必需的。可以说，挖掘模型的质量不会超过抽取样本的质量。数据探索和预处理的目的是为了保证样本数据的质量，从而为保证模型质量打下基础。

针对 1.4.2 节采集的餐饮数据，数据探索主要包括：异常值分析、缺失值分析、相关分析和周期性分析等，有关介绍详见第 3 章。

1.4.4　数据预处理

当采样数据维度过大时，如何进行降维处理、缺失值处理等都是数据预处理要解决的问题。

由于采样数据中常常包含许多含有噪声、不完整，甚至不一致的数据，对数据挖掘所涉及的数据对象必须进行预处理。那么，如何对数据进行预处理以改善数据质量，并最终达到完善最终数据挖掘结果的目的呢?

针对采集的餐饮数据，数据预处理主要包括：数据筛选、数据变量转换、缺失值处理、坏数据处理、数据标准化、主成分分析、属性选择、数据规约等，有关介绍详见第 3 章。

1.4.5　挖掘建模

样本抽取完成并经预处理后，接下来要考虑的问题是：本次建模属于数据挖掘应用中的哪类问题（分类、聚类、关联规则、时序模式或者智能推荐），选用哪种算法进行模型构建?

这一步是数据挖掘工作的核心环节。针对餐饮行业的数据挖掘应用，挖掘建模主要包括基于关联规则算法的动态菜品智能推荐、基于聚类算法的餐饮客户价值分析、基于分类与预测算法的菜品销量预测、基于整体优化的新店选址。

以菜品销量预测为例，模型构建是对菜品历史销量，是综合考虑了节假日、气候和竞争对手等采样数据轨迹的概括，它反映的是采样数据内部结构的一般特征，并与该采样数据的具体结构基本吻合。模型的具体化就是菜品销量预测公式，公式可以产生与观察值有相似结构的输出，这就是预测值。

1.4.6　模型评价

从 1.4.5 节的建模过程中会得出一系列的分析结果，模型评价的目的之一就是从这些模型中自动找出一个最好的模型，另外就是要根据业务对模型进行解释和应用。

对分类与预测模型和聚类分析模型的评价方法是不同的，具体评价方法详见第 5 章相关章节介绍。

1.5　常用的数据挖掘建模工具

数据挖掘是一个反复探索的过程，只有将数据挖掘工具提供的技术和实施经验与企业的

业务逻辑和需求紧密结合，并在实施过程中不断地磨合，才能取得好的效果。下面简单介绍几种常用的数据挖掘建模工具。

（1）SAS Enterprise Miner

Enterprise Miner（EM）是SAS推出的一个集成的数据挖掘系统，允许使用和比较不同的技术，同时还集成了复杂的数据库管理软件。它的运行方式是通过在一个工作空间（workspace）中按照一定的顺序添加各种可以实现不同功能的节点，然后对不同节点进行相应的设置，最后运行整个工作流程（workflow），便可以得到相应的结果。

（2）IBM SPSS Modeler

IBM SPSS Modeler原名Clementine，2009年被IBM公司收购后对产品的性能和功能进行了大幅度改进和提升。它封装了最先进的统计学和数据挖掘技术来获得预测知识，并将相应的决策方案部署到现有的业务系统和业务过程中，从而提高企业的效益。IBM SPSS Modeler拥有直观的操作界面、自动化的数据准备和成熟的预测分析模型，结合商业技术可以快速建立预测性模型。

（3）SQL Server

Microsoft公司的SQL Server中集成了数据挖掘组件——Analysis Servers，借助SQL Server的数据库管理功能，可以无缝地集成在SQL Server数据库中。在SQL Server 2008中提供了决策树算法、聚类分析算法、Naive Bayes算法、关联规则算法、时序算法、神经网络算法、线性回归算法等9种常用的数据挖掘算法。但是，预测建模的实现是基于SQL Server平台的，平台移植性相对较差。

（4）Python

Python（Matrix Laboratory，矩阵实验室）是美国Mathworks公司开发的应用软件，具备强大的科学及工程计算能力，它不但具有以矩阵计算为基础的强大数学计算能力和分析功能，而且还具有丰富的可视化图形表现功能和方便的程序设计能力。Python并不提供一个专门的数据挖掘环境，但它提供非常多的相关算法的实现函数，是学习和开发数据挖掘算法的很好选择。

（5）WEKA

WEKA（Waikato Environment for Knowledge Analysis）是一款知名度较高的开源机器学习和数据挖掘软件。高级用户可以通过Java编程和命令行来调用其分析组件。同时，WEKA也为普通用户提供了图形化界面，称为WEKA Knowledge Flow Environment和WEKA Explorer，可以实现预处理、分类、聚类、关联规则、文本挖掘、可视化等。

（6）KNIME

KNIME（Konstanz InformationMiner，http://www.knime.org）是基于Java开发的，可以扩展使用Weka中的挖掘算法。KNIME采用类似数据流（data flow）的方式来建立分析挖掘流程。挖掘流程由一系列功能节点组成，每个节点有输入/输出端口，用于接收数据或模型、导出结果。

（7）RapidMiner

RapidMiner 也称为 YALE（Yet Another Learning Environment, https://rapidminer.com），提供图形化界面，采用类似 Windows 资源管理器中的树状结构来组织分析组件，树上每个节点表示不同的运算符（operator）。YALE 中提供了大量的运算符，包括数据处理、变换、探索、建模、评估等各个环节。YALE 是用 Java 开发的，基于 Weka 来构建，可以调用 Weka 中的各种分析组件。RapidMiner 有拓展的套件 Radoop，可以和 Hadoop 集成起来，在 Hadoop 集群上运行任务。

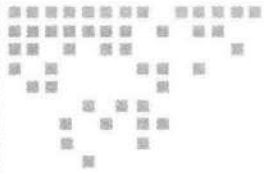
（8）TipDM

TipDM（顶尖数据挖掘平台）使用 Java 语言开发，能从各种数据源获取数据，建立多种数据挖掘模型。TipDM 目前已集成数十种预测算法和分析技术，基本覆盖了国外主流挖掘系统支持的算法。TipDM 支持数据挖掘流程所需的主要过程：数据探索（相关性分析、主成分分析、周期性分析）；数据预处理（属性选择、特征提取、坏数据处理、空值处理）；预测建模（参数设置、交叉验证、模型训练、模型验证、模型预测）；聚类分析、关联规则挖掘等一系列功能。

1.6　小结

本章从一个知名餐饮企业经营过程中存在的困惑出发，引出数据挖掘的概念、基本任务、建模过程及常用工具。

如何帮助企业从数据中洞察商机，提取价值，这是现阶段所有企业都关心的问题。通过发生在身边的案例，由浅入深地引出深奥的数据挖掘理论，让读者在不知不觉中感悟到数据挖掘的非凡魅力！本案例同时也贯穿到第 3 章至第 5 章的理论介绍中。



Chapter 2 第 2 章

Python 数据分析简介

Python 是一门简单易学且功能强大的编程语言。它拥有高效的高级数据结构，并且能够用简单而又高效的方式进行面向对象编程。Python 优雅的语法和动态类型，再结合它的解释性，使其在许多领域成为编写脚本或开发应用程序的理想语言。

要认识 Python，首先得明确一点，Python 是一门编程语言！这就意味着，原则上来说，它能够完成 Matlab 能够做的所有事情（因为大不了从头开始编写），而且在大多数情况下，同样功能的 Python 代码会比 Matlab 代码更加简洁、易懂；另一方面，因为它是一门编程语言，所以它能够完成很多 Matlab 不能做的事情，比如开发网页、开发游戏、编写爬虫来采集数据等。

Python 以开发效率著称，也就是说，它致力于以最短的代码完成任务。Python 通常为人诟病的是它的运行效率，而 Python 还被称为“胶水语言”，它允许我们把耗时的核心部分用 C/C++ 等更高效率的语言编写，然后由它来“黏合”，这很大程度上已经解决了 Python 的运行效率问题。事实上，在大多数数据任务上，Python 的运行效率已经可以媲美 C/C++ 语言。

本书致力于讲述用 Python 进行数据挖掘这一部分功能，而这部分功能，仅仅是 Python 强大功能中的冰山一角。随着 NumPy、SciPy、Matplotlib 和 Pandas 等众多程序库的开发，Python 在科学领域占据着越来越重要的地位，包括科学计算、数学建模、数据挖掘，甚至可以预见，未来 Python 将会成为科学领域的编程语言的主流。图 2-1 和图 2-2 是一些编程语言的使用排行榜图，它们可以证明 Python 越来越受欢迎。

Jun 2015	Jun 2014	Change	Programming Language	Ratings	Change
1	2	^	Java	17.822%	+1.71%
2	1	v	C	16.788%	+0.60%
3	4	^	C++	7.756%	+1.33%
4	5	^	C#	5.056%	+1.11%
5	3	v	Objective-C	4.339%	-6.60%
6	8	^	Python	3.999%	+1.29%
7	10	^	Visual Basic .NET	3.168%	+1.25%
8	7	v	PHP	2.868%	+0.02%
9	9		JavaScript	2.295%	+0.30%
10	17	^^	Delphi/Object Pascal	1.869%	+1.04%
11	-	^^	Visual Basic	1.839%	+1.84%
12	12		Perl	1.759%	+0.28%
13	23	^^	R	1.524%	+0.85%
14	-	^^	Swift	1.440%	+1.44%
15	19	^^	MATLAB	1.436%	+0.66%

图 2-1　2015 年 06 月的 TIOBE 编程语言排行榜，每月更新一次 [1]

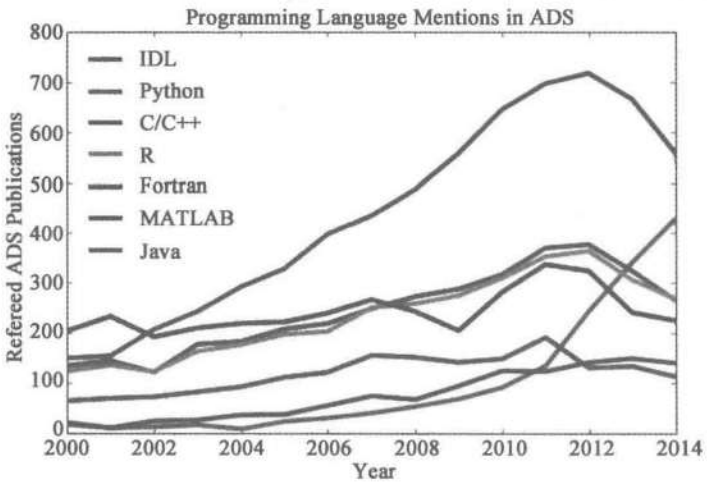


图 2-2　上图是近年天文学论文中所涉及的编程语言的趋势图，根据 ADS 中的论文致谢所提及的编程语言次数而制作 [2]

2.1 搭建 Python 开发平台

2.1.1 所要考虑的问题

Python 的官网：https://www.python.org/。

搭建 Python 开发平台有几个问题需要考虑，第一是选择什么操作系统，是 Windows 还是 Linux？第二是选择哪个 Python 版本，是 2.x 还是 3.x？

首先，来回答后一个问题。3.x 是对 2.x 的一个较大的更新，可以认为，Python 3.x 什么都好，就是它的部分代码不兼容 2.x 的，这使得不少好用的库都无法支持 3.x（值得庆幸的是，越来越多的主流库已经开始支持 3.x 了）。对于本书来说，本书使用 Python 2.7 版本，但是本书的代码尽可能地同时兼容 2.x 和 3.x，包括在各种第三方库也使用两个版本都兼容的扩展库。因此，在阅读本书的时候，不管你已经装了 2.x 还是 3.x，都无须在这个问题上太多纠结。

其次，就是选择操作系统的问题，主要是在 Windows 和 Linux 之间选择。Python 是跨平台的语言，因此脚本可以跨平台运行。然而，不同的平台运行效率不一样，一般来说，在 Linux 下的运行速度会比 Windows 快，而且是对于数据分析和挖掘任务。此外，在 Linux 下搭建 Python 环境相对来说容易一些，很多 Linux 发行版自带了 Python 程序，并且在 Linux 下更容易解决第三方库的依赖问题。当然，Linux 的操作门槛较高，入门的读者可以先在 Windows 环境下熟悉，然后再考虑迁移到 Linux 环境中。

2.1.2 基础平台的搭建

第一步是 Python 核心程序的安装，分为 Windows 和 Linux 介绍；最后介绍一个 Python 的科学计算发行版——Anaconda。

（1）Windows

在 Windows 系统中安装 Python 比较容易，直接到官方网站下载相应的 msi 安装包安装即可，和一般软件的安装无异，在此不赘述。安装包还分 32 位和 64 位版本，请读者自行选择适合的版本。

（2）Linux

大多数 Linux 发行版，如 CentOs、Debian、Ubuntu 等，都已经自带了 Python 2.x 的主程序，因此并不需要额外安装。

（3）Anaconda

安装 Python 核心程序只是第一步，为了实现更丰富的科学计算功能，还需要安装一些第三方的扩展库，这对于一般的读者来说可能显得比较麻烦，尤其是在 Windows 环境中还可能出现各种错误。幸好，已经有人专门将科学计算所需要的模块都编译好，然后打包以发行版

的形式供用户使用，Anaconda 就是其中一个常用的科学计算发行版。

Anaconda 的特点如下。

1）包含了众多流行的科学、数学、工程、数据分析的 Python 包。

2）完全开源和免费。

3）额外的加速、优化是收费的，但对于学术用途可以申请免费的 License。

4）全平台支持：Linux、Windows、Mac；支持 Python 2.6、2.7、3.3、3.4，可自由切换。

因此，推荐初级读者（尤其是 Windows 环境下的读者）安装此 Python 发行版。读者只需要到官方网站下载安装包安装，网址为：http://continuum.io/downloads。

安装好 Python 后，只需要在命令窗口输入 python 就可以进入 Python 环境，如图 2-3 是在 Windows 下启动 Python 2.7.8 的界面。

图 2-3　Python 2.7.8 在 Windows 下的启动

2.2　Python 使用入门

限于篇幅，本书不可能详细地讲解 Python 的使用，而只能是针对本书涉及的数据挖掘案例所用到的代码进行基本讲解。如果读者是初步接触 Python，并且用 Python 的目的就是数据挖掘，那么相信本节的介绍对你来说是比较充足的。如果读者需要进一步了解 Python，或者需要运行更加复杂的任务，那么本书是不够的（例如，本书没有谈及到面向对象编程），请读者自行阅读相应的 Python 教程。

2.2.1 运行方式

本节示例代码使用的是 Python 2.7。运行 Python 代码有两种方式，一种方式是启动 Python，然后在命令窗口下直接输入相应的命令；另外一种就是将完整的代码写成 .py 脚本，如 hello.py，然后通过 python hello.py 执行，如下所示。

```
# hello.py
print 'Hello World!'
```

执行结果如图 2-4 所示。

图 2-4 Hello.py

在编写脚本的时候，可以添加适当的注释。在每一行中，可以用井号“#”来添加注释，例如：

```
a = 2 + 3 #这句命令的意思是将2+3的结果赋值给a
```

如果注释有多行，可以在两个“”” 之间（3 个英文的单引号）添加注释内容。

```
a = 2 + 3
'''
这里是Python的多行注释。
这里是Python的多行注释。
'''
```

如果脚本中带有中文（中文注释或者中文字符串，中文字符串要在前面加 u），那么需要

在文件头注明编码，并且还要将脚本保存为 UTF-8 编码格式。

```
# -*- coding: utf-8 -*
print u'世界，你好！'
```

2.2.2　基本命令

（1）基本运算

认识 Python 的第一步，是可以把它当做一个方便的计算器来看待。读者可以打开 Python，试着输入以下命令。

```
a = 2
a * 2
a ** 2
```

以上是 Python 几个基本的运算，第一个是赋值运算，第二是乘法，最后是一个是幂（即 a^2），这些基本上是编程语言通用的。不过 Python 支持多重赋值。

```
a, b, c = 2, 3, 4
```

这句命令相当于

```
a = 2
b = 3
c = 4
```

Python 支持对字符串的灵活操作，如：

```
s = 'I like python'
s + ' very much' #将s与' very much'拼接，得到'I like python very much'
s.split(' ') #将s以空格分割，得到列表['I', 'like', 'python']
```

（2）判断与循环

显然判断和循环是所有编程语言的基本命令，Python 的判断语句如下。

```
if 条件1:
     语句2
elif 条件3:
     语句4
else:
     语句5
```

需要特别指出的是，Python 一般不用花括号 {}，也没有 end 语句，它是用缩进对齐作为语句的层次标记。同一层次的缩进量要一一对应，否则报错，如下面的语句是错误的。

```
if a==1:
   print a #缩进两个空格
else:
     print u'a不等于1' #缩进三个空格
```

不管是哪种语言，正确的缩进都是一个优雅的编程习惯。

Python 的循环也相应地有 for 循环和 while 循环，while 循环如下。

```
s,k = 0
while k < 101: #该循环过程就是求1+2+3+...+100
  k = k + 1
  s = s + k
print s
```

for 循环如下。

```
s = 0
for k in range(101): #该循环过程也是求1+2+3+...+100
  s = s + k
print s
```

这里，我们看到了 in 和 range 语法，in 是一个非常方便、而且非常直观的语法，用来判断一个元素是否在列表 / 元组中，range 用来生成连续的序列，一般语法为 range(a, b, c)，表示以 a 为首项、c 为公差且不超过 b-1 的等差数列，例如，

```
s = 0
if s in range(4):
    print u's在0, 1, 2, 3中'
if s not in range(1, 4, 1):
    print u's不在1, 2, 3中'
```

（3）函数

Python 用 def 来自定义函数。

```
def add2(x):
  return x+2
print add2(1) #输出结果为3
```

这很普通，没什么特别的，但是与一般编程语言不同的是，Python 的函数返回值可以是各种形式，比如返回列表，甚至返回多个值。

```
def add2(x = 0, y = 0): #定义函数，同时定义参数的默认值
  return [x+2, y+2] #返回值是一个列表
def add3(x, y):
  return x+3, y+3 #双重返回
a, b = add3(1,2) #此时a=4,b=5
```

有时候，像定义 add2() 这类简单的函数，用 def 来正式地写个命名、计算和返回显得有点麻烦了，Python 支持用 lambda 对简单的功能定义“行内函数”，这有点像 Matlab 中的“匿名函数”，如下。

```
f = lambda x : x + 2 #定义函数f(x)=x+2
g = lambda x, y: x + y #定义函数g(x,y)=x+y
```

2.2.3　数据结构

Python 有 4 个内建的数据结构——List（列表）、Tuple（元组）、Dictionary（字典）以及 Set（集合），它们可以统称为容器（container），因为它们实际上是一些“东西”组合而成的结构，而这些“东西”，可以是数字、字符甚至是列表，或者是它们之间几种的组合。通俗地讲，容器里是什么都行，而且容器里的元素类型不要求相同。

（1）列表 / 元组

列表和元组都是序列结构，它们本身很相似，但又有一点不同的地方。

从外形上看，列表与元组的区别是，列表是用方括号标记的，如 a = [1, 2, 3]，而元组是用圆括号标记的，如 b = (4, 5, 6)，访问列表和元组中的元素的方式都是一样的，如 a[0] 等于 1，b[2] 等于 6 等。上面已经谈及，容器里是什么都行，因此，下面的定义也是成立的。

```
c = [1, 'abc', [1, 2]]
'''
c是一个列表，列表的第一个元素是整型1，第二个是字符串'abc'，第三个是列表[1, 2]
'''
```

从功能上看，列表与元组的区别是，列表可以被修改，而元组不可以。比如，对于 a = [1, 2, 3]，那么语句 a[0] = 0，就会将列表 a 修改为 [0, 2, 3]，而对于元组 b = (4, 5, 6)，语句 b[0] = 1 就会报错。要注意的是，如果已经有了一个列表 a，同时想复制 a，命名为变量 b，那么 b = a 是无效的，这时候 b 仅仅是 a 的别名（或者说引用），修改 b 也会修改 a 的。正确的复制方法应该是 b = a[:]。

与列表有关的函数是 list，与元组有关的函数是 tuple，它们的用法和功能几乎一样，都是将某个对象转换为列表 / 元组，如 list('ab') 的结果是 ['a', 'b']，tuple([1, 2]) 的结果是 (1, 2)。表 2-1 是一些常见的与列表 / 元组相关的函数。

表2-1　列表/元组相关的函数

函　数	功　能	函　数	功　能
cmp(a, b)	比较两个列表 / 元组的元素	min(a)	返回列表 / 元组元素最小值
len(a)	列表 / 元组元素个数	sum(a)	将列表 / 元组中的元素求和
max(a)	返回列表 / 元组元素最大值	sorted(a)	对列表的元素进行升序排序

此外，作为对象，列表本身自带了很多实用的方法（元组不允许修改，因此方法很少），见表 2-2。

表2-2　列表相关的方法

函　数	功　能
a.append(1)	将 1 添加到列表 a 末尾
a.count(1)	统计列表 a 中元素 1 出现的次数

（续）

函　数	功　能
a.extend([1, 2])	将列表 [1, 2] 的内容追加到列表 a 的末尾中
a.index(1)	从列表 a 中找出第一个 1 的索引位置
a.insert(2, 1)	将 1 插入列表 a 的索引为 2 的位置
a.pop(1)	移除列表 a 中索引为 1 的元素

最后，不能不提的是“列表解析”这一功能，它能够简化我们对列表内元素逐一进行操作的代码，如下面的代码

```
a = [1, 2, 3]
b = []
for i in a:
  b.append(i + 2)
```

可以简化到

```
a = [1, 2, 3]
b = [i+2 for i in a]
```

这样的语法不仅方便，而且直观！充分体现了 Python 语法的人性化。在本书中，我们将会比较多地用到这样简洁的代码。

（2）字典

Python 引入了“自编”这一方便的概念。从数学上来讲，它实际上是一个映射。通俗来讲，它也相当于一个列表，然而它的“下标”不再是以 0 开头的数字，而是让自己定义的“键”（Key）开始。

创建一个字典的基本方法为：

```
d = {'today':20, 'tomorrow':30}
```

这里的 'today'、'tomorrow' 就是字典的键，它在整个字典中必须是唯一的，而 20、30 就是键对应的值，访问字典中元素的方法也很直观。

```
d['today'] #该值为20
d['tomorrow'] #该值为30
```

还有其他一些比较方便的方法来创建一个字典，如通过 dict() 函数转换，或者通过 dict.fromkeys 来创建，如下。

```
dict([['today', 20], ['tomorrow', 30]]) #也相当于{'today':20, 'tomorrow':30}
dict.fromkeys(['today', 'tomorrow'], 20) #相当于{'today':20, 'tomorrow':20}
```

很多字典的函数和方法与列表是一样的，因此在这里就不再赘述了。

（3）集合

Python 内置了集合这一数据结构，同数学上的集合概念基本上是一致的，它与列表的区别在于：1. 它的元素是不重复的，而且是无序的；2. 它不支持索引。一般我们通过花括号 {} 或者 set() 函数来创建一个集合。

```
s = {1, 2, 2, 3} #注意2会自动去重，得到{1, 2, 3}
s = set([1, 2, 2, 3]) #同样，它将列表转换为集合，得到{1, 2, 3}
```

由于集合的特殊性（特别是无序性），因此集合有一些特别的运算。

```
a = t | s #t和s的并集
b = t & s #t和s的交集
c = t - s #求差集（项在t中，但不在s中）
d = t ^ s #对称差集（项在t或s中，但不会同时出现在二者中）
```

在本书中，集合并不常用，所以这里仅仅简单地介绍它，并不进行详细说明，如果读者想深入了解集合对象，请自行搜索相关教程。

（4）函数式编程

函数式编程（Functional Programming）或者函数程序设计，又称泛函编程，是一种编程范型，它将计算机运算视为数学上的函数计算，并且避免使用程序状态以及易变对象。简单来讲，函数式编程是一种“广播式”的编程，一般结合前面提到过的 lambda 定义函数，用于科学计算中，会显得特别简洁方便。

在 Python 中，函数式编程主要由几个函数的使用构成：lambda()、map()、reduce()、filter()，lambda 前面已经介绍过，主要用来自定义“行内函数”，所以现在我们逐一介绍后三个。

首先介绍 map() 函数。假设有一个列表 a = [1, 2, 3]，要给列表中的每个元素都加 2 得到一个新列表，利用前面已经谈及过的“列表解析”，我们可以这样写：

```
b = [i+2 for i in a]
```

而利用 map 函数我们可以这样写：

```
b = map(lambda x: x+2, a)
b = list(b) #结果是[3, 4, 5]
'''
在3.x需要b = list(b)这一步，在2.x不需要这步，原因是在3.x中，map函数仅仅是创建一个待运行的命
    令容器，只有其他函数调用它的时候才返回结果。
'''
```

也就是说，我们首先定义一个函数，然后再用 map() 命令将函数逐一应用到（map）列表中的每个元素，最后返回一个数组。map() 命令也接受多参数的函数，如 map(lambda x,y: x*y, a, b) 表示将 a、b 两个列表的元素对应相乘，把结果返回给新列表。

也许有的读者会疑问，有了列表解析，为什么还要有 map() 命令呢？其实列表解析虽然代码简短，但是本质上还是 for 命令，而 Python 的 for 命令效率并不高，而 map() 函数实现

了相同的功能，并且效率更高，原则上来说，它的循环命令速度相当于 C 语言。

接着是 reduce() 函数。它有点像 map() 函数，但 map() 函数用于逐一遍历，而是 reduce() 函数用于递归计算。先给出一个例子，这个例子可以算出 n 的阶乘：

```
reduce(lambda x,y: x*y, range(1, n+1))
```

（注：在 2.x 中，上述命令可以直接运行，在 3.x 中，reduce 函数已经被移出了全局命名空间，它被置于 fuctools 库中，如需使用，则要通过 from fuctools import reduce 引入 reduce）

其中，range(1, n+1) 相当于给出了一个列表，元素是 1 ～ n 这 n 个整数。lambda x, y: x*y 构造了一个二元函数，返回两个参数的乘积。reduce 命令首先将列表的前两个元素作为函数的参数进行运算，然后将运算结果与第三个数字作为函数的参数，然后再将运算结果与第四个数字作为函数的参数……依此递推，直到列表结束，返回最终结果。如果用循环命令，那就要写成：

```
s = 1
for i in range(1, n+1):
  s = s * i
```

最后是 filter() 函数。顾名思义，它是一个过滤器，用来筛选出列表中符合条件的元素，例如，

```
b = filter(lambda x: x > 5 and x < 8, range(10))
b = list(b) #结果是[6, 7]，在3.x需要b = list(b)这一步，在2.x不需要这步，理由同map
```

使用 filter() 函数首先需要一个返回值为 bool 型的函数，如上述的 lambda x: x > 5 and x < 8 定义了一个函数，判断 x 是否大于 5 且小于 8，然后将这个函数作用到 range(10) 的每个元素中，如果为 True，则“挑出”那个元素，最后将满足条件的所有元素组成一个列表返回。

当然，上述 filter 语句，可以用列表解析写为：

```
b = [i for i in range(10) if i > 5 and i < 8]
```

它并不比 filter 语句复杂。但是要注意，我们使用 map()、reduce() 或 filter()，最终目的是兼顾简洁和效率，因为 map()、reduce() 或 filter() 的循环速度比 Python 内置的 for 或 while 循环要快得多。

2.2.4 库的导入与添加

上面我们已经讲述了 Python 基本平台的搭建和使用，然而在默认情况下它并不会将它所有的功能加载进来。我们需要把更多的库（或者叫作模块、包等）加载进来，甚至需要额外安装第三方的扩展库，以丰富 Python 的功能，实现我们的目的。

（1）库的导入

Python 本身内置了很多强大的库，如数学相关的 math 库，可以为我们提供更加丰富复杂的数学运算：

```
import math

math.sin(1) #计算正弦
math.exp(1) #计算指数
math.pi #内置的圆周率常数
```

导入库的方法，除了使用“import 库名”之外，还可以为库起一个别名：

```
import math as m
m.sin(1) #计算正弦
```

此外，如果并不需要导入库中的所有函数，可以特别指定导入函数的名字：

```
from math import exp as e #只导入math库中的exp函数，并起别名e
e(1) #计算指数
sin(1) #此时sin(1)和math.sin(1)都会出错，因为没被导入
```

直接地导入库中的所有函数：

```
from math import * #直接的导入，也就是去掉math.，但如果大量地这样引入第三库，就容易引起命名
    冲突。
exp(1)
sin(1)
```

我们可以通过 help('modules') 命令来获得已经安装的所有模块名。

（2）导入 future 特征（For 2.x）

Python 2.x 与 3.x 之间的差别不仅仅在内核上，也表现在代码的实现中。比如，在 2.x 中，print 是作为一个语句出现的，用法为 print a；但是在 3.x 中，它是作为函数出现的，用法为 print(a)。为了保证兼容性，本书的基本代数是使用 3.x 的语法编写的，而使用 2.x 的读者，可以通过引入 future 特征的方式兼容代码，如，

```
#将print变成函数形式，即用print(a)格式输出
from __future__ import print_function

#3.x的3/2=1.5，3//2才等于1；2.x中3/2=1
from __future__ import division
```

（3）添加第三方库

Python 自带了很多库，但不一定可以满足我们的需求。就数据分析和数据挖掘而言，还需要添加一些第三方的库来拓展它的功能。这里简单介绍一下第三方库的安装，以安装数据分析工具 Pandas 为例。

安装第三方库一般有以下几种思路，见表 2-3。

表2-3　常见的安装第三方库的方法

思　路	特　点
下载源代码自行安装	安装灵活，但需要自行解决上级依赖问题

（续）

思　路	特　点
用 pip 安装	比较方便，自动解决上级依赖问题
用 easy_install 安装	比较方便，自动解决上级依赖问题，比 pip 稍弱
下载编译好的文件包	一般是 Windows 系统才提供现成的可执行文件包
系统自带的安装方式	Linux 或 Mac 系统的软件管理器自带了某些库的安装方式

这些安装方式将在 2.3 节中实际展示。

2.3 Python 数据分析工具

Python 本身的数据分析功能不强，需要安装一些第三方扩展库来增强它的能力。本书用到的库有 Numpy、Scipy、Matplotlib、Pandas、Scikit-Learn、Keras 和 Gensim 等，下面将对这些库的安装和使用进行简单的介绍。

如果读者安装的是 Anaconda 发行版，那么它已经自带了以下库：Numpy、Scipy、Matplotlib、Pandas 和 Scikit-Learn。

本章主要是对这些库进行简单的介绍，在后面的章节中，会通过各种案例对这些库的使用进行更加深入的说明。本书的介绍是有所侧重的，读者可以到官网阅读更加详细的使用教程。值得一提的是，本书所介绍的扩展库，它们的官网上的帮助文档都相当详细。

用 Python 进行科学计算是很丰富的学问，本书只是用到了它的数据分析和挖掘相关的部分功能，所涉及的一些库如表 2-4 所示。读者可以参考书籍《用 Python 做科学计算》了解更多信息。

表2-4　Python数据挖掘相关扩展库

扩　展　库	简　介
Numpy	提供数组支持，以及相应的高效的处理函数
Scipy	提供矩阵支持，以及矩阵相关的数值计算模块
Matplotlib	强大的数据可视化工具、作图库
Pandas	强大、灵活的数据分析和探索工具
StatsModels	统计建模和计量经济学，包括描述统计、统计模型估计和推断
Scikit-Learn	支持回归、分类、聚类等的强大的机器学习库
Keras	深度学习库，用于建立神经网络以及深度学习模型
Gensim	用来做文本主题模型的库，文本挖掘可能用到

限于篇幅，我们仅仅介绍本书的案例中会用到的一些库，还有一些很实用的库并没有介绍，如涉及图片处理可以用 Pillow（旧版为 PIL，目前已经被 Pillow 代替）、涉及视频处理可

以用 OpenCV、涉及高精度运算可以用 GMPY2 等，而对于这些额外的知识，建议读者在遇到相应的问题时，自行到网上搜索相关资料。相信通过对本书的学习后，读者解决 Python 相关问题的能力一定会大大提高的。

2.3.1 Numpy

Python 并没有提供数组功能。虽然列表可以完成基本的数组功能，但它不是真正的数组，而且在数据量较大时，使用列表的速度就会慢得让人难以接受。为此，Numpy 提供了真正的数组功能，以及对数据进行快速处理的函数。Numpy 还是很多更高级的扩展库的依赖库，后面章节介绍的 Scipy、Matplotlib、Pandas 等库都依赖于它。值得强调的是，Numpy 内置函数处理数据的速度是 C 语言级别的，因此在编写程序的时候，应当尽量使用它们内置的函数，避免出现效率瓶颈的现象（尤其是涉及循环的问题）。

在 Windows 中，Numpy 安装跟普通的第三方库安装一样，可以通过 pip 安装：

```
pip install numpy
```

也可以自行下载源代码，然后用

```
python setup.py install
```

安装。在 Linux 下上述方面也是可行的。此外，很多 Linux 发行版的软件源中都有 Python 常见的库，因此还可以通过 Linux 自带的软件管理器进行安装，如在 Ubuntu 下可以用

```
sudo apt-get install python-numpy
```

安装。安装完成后，可以使用以下命令进行测试。

代码清单2-1　Numpy基本操作

```
# -*- coding: utf-8 -*
import numpy as np #一般以np作为numpy的别名
a = np.array([2, 0, 1, 5]) #创建数组
print(a) #输出数组
print(a[:3]) #引用前三个数字（切片）
print(a.min()) #输出a的最小值
a.sort() #将a的元素从小到大排序，此操作直接修改a，因此这时候a为[0, 1, 2, 5]
b= np.array([[1, 2, 3], [4, 5, 6]]) #创建二维数组
print(b*b) #输出数组的平方阵，即[[1, 4, 9], [16, 25, 36]]
```

Numpy 是 Python 中相当成熟和常用的库，因此关于它的教程有很多，最值得一看的是它官网的帮助文档，还有很多中英文教程，读者遇到相应的问题时，可以自行搜索对应的内容。

参考链接：

- http://www.numpy.org/。
- http://reverland.org/python/2012/08/22/numpy/。

2.3.2 Scipy

如果说 Numpy 让 Python 有了 Matlab 的味道，那么 Scipy 就让 Python 真正地成为了半个 Matlab 了。Numpy 提供了多维数组功能，但它只是一般的数组，并不是矩阵。例如，当两个数组相乘时，只是对应元素相乘，而不是矩阵乘法。Scipy 提供了真正的矩阵，以及大量基于矩阵运算的对象与函数。

Scipy 包含的功能有最优化、线性代数、积分、插值、拟合、特殊函数、快速傅里叶变换、信号处理和图像处理、常微分方程求解和其他科学与工程中常用的计算，显然，这些功能都是挖掘与建模必备的。

Scipy 依赖于 Numpy，因此安装它之前得先安装 Numpy。安装 Scipy 的方式与安装 Numpy 的方法大同小异，需要提及的是，在 Ubuntu 下也可以用类似的

```
sudo apt-get install python-scipy
```

安装 Scipy。安装好 Scipy 后，可以通过以下命令简单试用。

代码清单2-2 Scipy求解非线性方程组和数值积分

```
# -*- coding: utf-8 -*
#求解非线性方程组2x1-x2^2=1,x1^2-x2=2
from scipy.optimize import fsolve #导入求解方程组的函数
def f(x): #定义要求解的方程组
  x1 = x[0]
  x2 = x[1]
  return [2*x1 - x2**2 - 1, x1**2 - x2 -2]

result = fsove(f, [1,1]) #输入初值[1, 1]并求解
print(result) #输出结果，为array([ 1.91963957,  1.68501606])

#数值积分
from scipy import integrate #导入积分函数
def g(x): #定义被积函数
  return (1-x**2)**0.5

pi_2, err = integrate.quad(g, -1, 1) #积分结果和误差
print(pi_2 * 2) #由微积分知识知道积分结果为圆周率pi的一半
```

参考链接：

- http://www.scipy.org/。
- http://reverland.org/python/2012/08/24/scipy/。

2.3.3 Matplotlib

不论是数据挖掘还是数学建模，都免不了数据可视化的问题。对于 Python 来说，Matplotlib 是最著名的绘图库，它主要用于二维绘图，当然它也可以进行简单的三维绘图。它不但提供了一整套和 Matlab 相似但更为丰富的命令，让我们可以非常快捷地用 Python 可

视化数据，而且允许输出达到出版质量的多种图像格式。

Matplotlib 的安装并没有什么特别之处，可以通过 pip install matplotlib 安装或者自行下载源代码安装，在 Ubuntu 下也可以用类似的。

```
sudo apt-get install python-matplotlib
```

安装。Matplotlib 的上级依赖库相对较多，手动安装的时候，需要逐一把这些依赖库都安装好。安装完成后就可以牛刀小试了，下面是一个简单的作图例子，它基本包含了 Matplotlib 作图的关键要素，作图效果如图 2-5 所示。

代码清单2-3　Matplotlib作图的基本代码

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt #导入Matplotlib

x = np.linspace(0, 10, 1000) #作图的变量自变量
y = np.sin(x) + 1 #因变量y
z = np.cos(x**2) + 1 #因变量z

plt.figure(figsize = (8, 4)) #设置图像大小
plt.plot(x,y,label = '$\sin x+1$', color = 'red', linewidth = 2) #作图，设置标签、
    线条颜色、线条大小
plt.plot(x, z, 'b--', label = '$\cos x^2+1$')  #作图，设置标签、线条类型
plt.xlabel('Time(s) ') # x轴名称
plt.ylabel('Volt') # y轴名称
plt.title('A Simple Example') #标题
plt.ylim(0, 2.2) #显示的y轴范围
plt.legend() #显示图例
plt.show() #显示作图结果
```

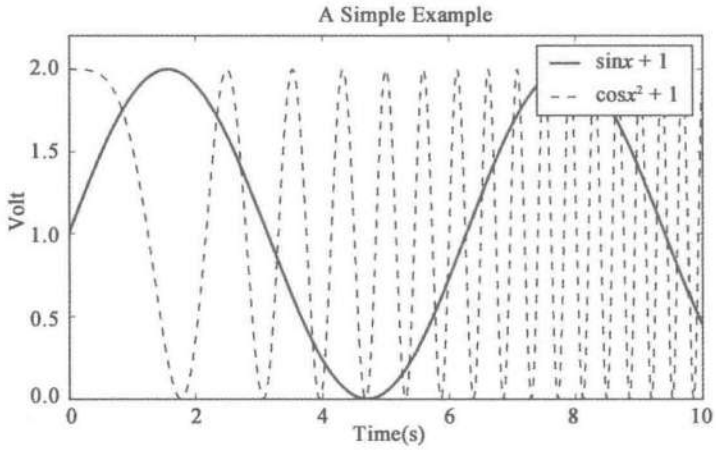


图 2-5　Matplotlib 基本的作图例子

如果读者使用的是中文标签，就会发现中文标签无法正常显示。这是由于 Matplotlib 的默认字体是英文字体所致，解决它的办法是在作图之前手动指定默认字体为中文字体，如黑体（SimHei）。

```
plt.rcParams['font.sans-serif'] = ['SimHei'] #这两句用来正常显示中文标签
```

另外，保存作图图像时，负号有可能显示不正常，可以通过以下代码解决：

```
plt.rcParams['axes.unicode_minus'] = False #解决保存图像是负号'-'显示为方块的问题
```

这里有一个小建议给读者：有时间多去 Matplotlib 提供的“画廊”欣赏它做出来的漂亮效果，也许你就慢慢地爱上 Matplotlib 作图了。（画廊：http://matplotlib.org/gallery.html ）

参考链接：

- http://matplotlib.org/。
- http://reverland.org/python/2012/09/07/matplotlib-tutorial/。

2.3.4 Pandas

终于谈到本书的主力工具——Pandas 了。Pandas 是 Python 下最强大的数据分析和探索工具（貌似没有之一）。它包含高级的数据结构和精巧的工具，使得在 Python 中处理数据非常快速和简单。Pandas 构建在 NumPy 之上，它使得以 NumPy 为中心的应用很容易使用。Pandas 的名称来自于面板数据（Panel Data）和 Python 数据分析（Data Analysis），它最初被作为金融数据分析工具而开发出来，由 AQR Capital Management 公司于 2008 年 4 月开发出来，并于 2009 年底开源。

Pandas 的功能非常强大，支持类似于 SQL 的数据增、删、查、改，并且带有丰富的数据处理函数；支持时间序列分析功能；支持灵活处理缺失数据等。事实上，单纯 Pandas 工具就足以写一本书，读者可以阅读 Pandas 的主要作者之一 Wes McKinney 写的《利用 Python 进行数据分析》一书，学习更详细的内容。

（1）安装

Pandas 的安装相对来说比较容易，安装好 Numpy 之后，就可以直接安装了，通过 pip install pandas 或下载源码后 python setup.py install 安装均可。由于我们频繁用到读取和写入 Excel，但默认的 Pandas 还不能读写 Excel 文件，需要安装 xlrd（读）和 xlwt（写）库才能支持 Excel 的读写，方法如下。

pip install xlrd # 为 Python 添加读取 Excel 的功能

pip install xlwt # 为 Python 添加写入 Excel 的功能

（2）使用

在后面的章节中，我们会逐步展示 Pandas 的强大功能，而在本节，我们先以简单的例子一睹为快。

Pandas 基本的数据结构是 Series 和 DataFrame。顾名思义，Series 就是序列，类似一维

数组；DataFrame 则是相当于一张二维的表格，类似二维数组，它的每一列都是一个 Series。为了定位 Series 中的元素，Pandas 提供了 Index 对象，每个 Series 都会带有一个对应的 Index，用来标记不同的元素，Index 的内容不一定是数字，也可以是字母、中文等，它类似于 SQL 中的主键。

类似地，DataFrame 相当于多个带有同样 Index 的 Series 的组合（本质是 Series 的容器），每个 Seiries 都带有唯一的表头，用来标识不同的 Series。

代码清单2-4　Pandas的简单例子

```
# -*- coding: utf-8 -*-
import pandas as pd #通常用pd作为pandas的别名。

s = pd.Series([1,2,3], index=['a', 'b', 'c']) #创建一个序列s
d = pd.DataFrame([[1, 2, 3], [4, 5, 6]], columns = ['a', 'b', 'c']) #创建一个表
d2 = pd.DataFrame(s) #也可以用已有的序列来创建表格

d.head() #预览前5行数据
d.describe() #数据基本统计量

#读取文件，注意文件的存储路径不能带有中文，否则读取可能出错。
pd.read_excel('data.xls') #读取Excel文件，创建DataFrame。
pd.read_csv('data.csv', encoding = 'utf-8') #读取文本格式的数据，一般用encoding指定编码。
```

由于 Pandas 是本书的主力工具，在后面将会频繁使用它，因此在这里就不进行详细介绍了，在后面的使用过程中将会更加详尽地讲解 Pandas 的使用方法。

参考链接：

- http://pandas.pydata.org/pandas-docs/stable/。
- http://jingyan.baidu.com/season/43456。

2.3.5　StatsModels

Pandas 着眼于数据的读取、处理和探索，而 StatsModels 则更加注重数据的统计建模分析，它使得 Python 有了 R 语言的味道。StatsModels 支持与 Pandas 进行数据交互，因此，它与 Pandas 结合，成为了 Python 下强大的数据挖掘组合。

安装 StatsModels 相当简单，既可以通过 pip 安装，又可以通过源码安装。对于 Windows 用户来说，官网上甚至已经有编译好的 exe 文件以供下载。如果手动安装的话，需要自行解决好依赖问题，StatModel 依赖于 Pandas（当然也依赖于 Pandas 所依赖的），同时还依赖于 pasty（一个描述统计的库）。

下面是一个用 StatsModels 来进行 ADF 平稳性检验的例子。

```
# -*- coding: utf-8 -*-
from statsmodels.tsa.stattools import adfuller as ADF #导入ADF检验
import numpy as np
```

```
ADF(np.random.rand(100)) #返回的结果有ADF值、p值等
```

参考链接：

- http://statsmodels.sourceforge.net/stable/index.html。
- http://jingyan.baidu.com/season/43456。

2.3.6 Scikit-Learn

从该库的名字可以看出，这是一个机器学习相关的库。不错，Scikit-Learn 是 Python 下强大的机器学习工具包，它提供了完善的机器学习工具箱，包括数据预处理、分类、回归、聚类、预测和模型分析等。

Scikit-Learn 依赖于 NumPy、SciPy 和 Matplotlib，因此，只需要提前安装好这几个库，然后安装 Scikit-Learn 就基本上没有什么问题了，安装方法和前几节一样，要不就是 pip install scikit-learn 安装，要不就是下载源码自己安装。

创建一个机器学习的模型很简单：

```
# -*- coding: utf-8 -*-
from sklearn.linear_model import LinearRegression #导入线性回归模型
model = LinearRegression() #建立线性回归模型
print(model)
```

1）所有模型提供的接口有：

model.fit()：训练模型，对于监督模型来说是 fit(X, y)，对于非监督模型是 fit(X)。

2）监督模型提供的接口有：

model.predict(X_new)：预测新样本

model.predict_proba(X_new)：预测概率，仅对某些模型有用（比如 LR）

model.score()：得分越高，fit 越好

3）非监督模型提供的接口有：

model.transform()：从数据中学到新的“基空间”。

model.fit_transform()：从数据中学到新的基并将这个数据按照这组“基”进行转换。

Scikit-Learn 本身提供了一些实例数据，比较常见的有安德森鸢尾花卉数据集、手写图像数据集等。我们有一百五十个鸢尾花的一些尺寸的观测值：萼片长度、宽度，花瓣长度和宽度。还有它们的亚属：山鸢尾（Iris setosa）、变色鸢尾（Iris versicolor）和维吉尼亚鸢尾（Iris virginica）。

```
# -*- coding: utf-8 -*-
from sklearn import datasets #导入数据集

iris = datasets.load_iris() #加载数据集
print(iris.data.shape) #查看数据集大小
```

```
from sklearn import svm #导入SVM模型

clf = svm.LinearSVC() #建立线性SVM分类器
clf.fit(iris.data, iris.target) #用数据训练模型
clf.predict([[ 5.0,  3.6,  1.3,  0.25]]) #训练好模型之后，输入新的数据进行预测
clf.coef_ #查看训练好模型的参数
```

参考链接：

http://scikit-learn.org/。

2.3.7 Keras

虽然 Scikit-Learn 足够强大，但是它并没有包含一种强大的模型——人工神经网络。人工神经网络是功能相当强大的、但是原理又相当简单的模型，在语言处理、图像识别等领域有着重要的作用。近年来逐渐火起来的“深度学习”算法，本质上也就是一种神经网络，可见在 Python 中实现神经网络是非常必要的。

本书用 Keras 库来搭建神经网络。事实上，Keras 并非简单的神经网络库，而是一个基于 Theano 的强大的深度学习库，利用它不仅仅可以搭建普通的神经网络，还可以搭建各种深度学习模型，如自编码器、循环神经网络、递归神经网络、卷积神经网络等。由于它是基于 Theano 的，因此速度也相当快。

有必要介绍一下 Theano，它也是 Python 的一个库，它是由深度学习专家 Yoshua Bengio 带领的实验室开发出来的，用来定义、优化和高效地解决多维数组数据对应数学表达式的模拟估计问题。它具有高效地实现符号分解、高度优化的速度和稳定性等特点，最重要的是它还实现了 GPU 加速，使得密集型数据的处理速度是 CPU 的数十倍。

用 Theano 就可以搭建起高效的神经网络模型，但是对于普通读者来说门槛还是相当高的，Keras 正是为此而生，它大大简化了搭建各种神经网络模型的步骤，允许普通用户轻松地搭建并求解具有几百个输入节点的深层神经网络，而且定制的自由度非常大，甚至可能惊呼：搭建神经网络可以如此简单！

（1）安装

安装 Keras 之前首先需要安装 Numpy、Scipy 和 Theano。安装 Theano 先要准备一个 C++ 编译器，这在 Linux 下是自带的。因此，在 Linux 下安装 Theano 和 Keras 非常简单，只需要下载源代码，然后用 python setup.py install 安装就行了，具体可以参考官方文档。

可是在 Windows 下就没有那么简单了，因为它没有现成的编译环境。一般而言是先安装 MinGW（Windows 下的 GCC 和 G++），然后再安装 Theano（提前装好 Numpy 等依赖库），最后安装 Keras。如果要实现 GPU 加速，还需要安装和配置 CUDA（天下没有免费的午餐，想要速度、易用两不误，那么就得花点心思）。限于篇幅，本书不详细介绍在 Windows 下 Theano 和 Keras 的安装配置方法。

值得一提的是，在 Windows 下 Keras 的速度会大打折扣，因此，想要在神经网络和深度

学习方面进行深入研究的读者，请在 Linux 下搭建相应的环境。

参考链接：

- http://deeplearning.net/software/theano/install.html#install。
- https://github.com/fchollet/keras。

（2）使用

用 Keras 搭建神经网络模型的过程相当简洁，也相当直观，就像搭积木一般。通过短短几十行代码，我们就可以搭建起一个非常强大的神经网络模型，甚至是深度学习模型。简单搭建一个 MLP（多层感知器），如下：

```
# -*- coding: utf-8 -*-
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import SGD

model = Sequential() #模型初始化
model.add(Dense(20, 64)) #添加输入层（20节点）、第一隐藏层（64节点）的连接
model.add(Activation('tanh')) #第一隐藏层用tanh作为激活函数
model.add(Dropout(0.5)) #使用Dropout防止过拟合
model.add(Dense(64, 64)) #添加第一隐藏层（64节点）、第二隐藏层（64节点）的连接
model.add(Activation('tanh')) #第二隐藏层用tanh作为激活函数
model.add(Dropout(0.5)) #使用Dropout防止过拟合
model.add(Dense(64, 1)) #添加第二隐藏层（64节点）、输出层（1节点）的连接
model.add(Activation('sigmoid')) #输出层用sigmoid作为激活函数

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True) #定义求解算法
model.compile(loss='mean_squared_error', optimizer=sgd) #编译生成模型，损失函数为平均
    误差平方和

model.fit(X_train, y_train, nb_epoch=20, batch_size=16) #训练模型
score = model.evaluate(X_test, y_test, batch_size=16) #测试模型
```

要注意的是，Keras 的预测函数与 Scikit-Learn 有所差别，Keras 用 model.predict() 方法给出概率，model.predict_classes() 方法给出分类结果。

参考链接：

- http://radimrehurek.com/gensim/。
- http://www.52nlp.cn/ 如何计算两个文档的相似度二。

2.3.8 Gensim

在 Gensim 的官网中，它对自己的简介只有一句话：topic modelling for humans!

Gensim 是用来处理语言方面的任务，如文本相似度计算、LDA、Word2Vec 等，这些领域的任务往往需要比较多的背景知识，通常的情况是：研究这方面的读者，已经不需要我再多说什么；不研究这方面的读者，在这里也说不清楚。（所以 Gensim 的介绍只有一句话也就不奇怪了。）

因此，在这一节中，只是提醒读者有这么一个库的存在，而且这个库很强大，如果用得到这个库，请读者去阅读官方帮助文档或参考链接。

需要一提的是，Gensim 把 Google 公司在 2013 年开源的著名的词向量构造工具 Word2Vec 编译好了作为它的子库，因此需要用到 Word2Vec 的读者也可以直接用 Gensim 而无需自行编译了。据说 Gensim 的作者对 Word2Vec 的代码进行了优化，据说它在 Gensim 下的表现比原生的 Word2Vec 还要快。(为了实现加速，需要准备 C++ 编译器环境，因此，建议用到 Gensim 的 Word2Vec 的读者在 Linux 下环境运行。)

下面是一个 Gensim 使用 Word2Vec 的简单例子。

```
# -*- coding: utf-8 -*-
import gensim, logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s',
    level=logging.INFO)
#logging是用来输出训练日志

#分好词的句子，每个句子以词列表的形式输入
sentences = [['first', 'sentence'], ['second', 'sentence']]

#用以上句子训练词向量模型
model = gensim.models.Word2Vec(sentences, min_count=1)

print(model['sentence']) #输出单词sentence的词向量。
```

参考链接：

- http://radimrehurek.com/gensim/。
- http://www.52nlp.cn/（如何计算两个文档的相似度二）。

2.4　配套资源使用设置

本书提供的下载资源按照章节组织，在资源的目录中会有 chapter2、chapter3、chapter4 等章节。在原理篇章节中其章节目录下只包含“ demo”文件夹（示例程序文件夹），包含 3 个子目录：code、data 和 tmp。其中，code 为章节正文中使用到的代码、data 为使用的数据文件、tmp 文件夹中存放临时文件或者示例程序运行的结果文件。

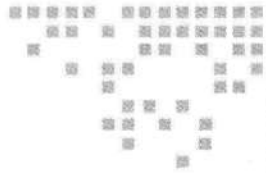
在实战篇章节如 chapter6 下则包含“ demo”“ test”“上机实验拓展”和“拓展思考”文件夹，分别对应于“示例程序”“上机实验”“上机实验拓展”和“拓展思考”。其中的“ demo”文件夹和原理篇一致；“ test”文件夹则主要针对上机实验部分的完整代码，其子目录结构和“示例程序”一致；“上机实验拓展”里面包含“SAS”“R”和“SPSS”文件夹，主要是使用不同的工具来解决上机实验问题；“拓展思考”则主要存储拓展思考部分的数据文件。

读者只需把整个章节（如 chapter2 ）复制到本地，注意用到 Pandas 的时候不要置于中文

路径，然后打开其中的示例程序即可运行程序并得到结果。需要注意，在示例程序中使用的一些自定义函数在对应的章节可以找到相应的 .py 文件。同时示例程序中的参数初始化可能需要根据具体设置进行配置，如果和示例程序不同，请自行修改。

2.5 小结

本章主要对 Python 进行简单介绍，包括软件安装、使用入门及相关注意事项和 Python 数据分析及挖掘相关工具箱。由于 Python 包含多个领域的扩展库，而且扩展库的功能也相当丰富，本章只介绍与数据分析及数据挖掘相关的一小部分，包括高维数组、数值计算、可视化、机器学习、神经网络和语言模型等。这些扩展库里面包含的函数在后续章节中会进行实例分析，通过在 Python 平台上完成实际案例来掌握数据分析和数据挖掘的原理，培养读者应用数据分析和挖掘技术解决实际问题的能力。



第3章 Chapter 3

数据探索

根据观测、调查收集到初步的样本数据集后，接下来要考虑的问题是：样本数据集的数量和质量是否满足模型构建的要求？是否出现从未设想过的数据状态？其中有没有什么明显的规律和趋势？各因素之间有什么样的关联性？

通过检验数据集的数据质量、绘制图表、计算某些特征量等手段，对样本数据集的结构和规律进行分析的过程就是数据探索。数据探索有助于选择合适的数据预处理和建模方法，甚至可以完成一些通常由数据挖掘解决的问题。

本章从数据质量分析和数据特征分析两个角度对数据进行探索。

3.1 数据质量分析

数据质量分析是数据挖掘中数据准备过程的重要一环，是数据预处理的前提，也是数据挖掘分析结论有效性和准确性的基础，没有可信的数据，数据挖掘构建的模型将是空中楼阁。

数据质量分析的主要任务是检查原始数据中是否存在脏数据，脏数据一般是指不符合要求，以及不能直接进行相应分析的数据。在常见的数据挖掘工作中，脏数据包括如下内容。

- 缺失值。
- 异常值。
- 不一致的值。
- 重复数据及含有特殊符号（如#、¥、*）的数据。

本小节将主要对数据中的缺失值、异常值和一致性进行分析。

3.1.1 缺失值分析

数据的缺失主要包括记录的缺失和记录中某个字段信息的缺失，两者都会造成分析结果的不准确，以下从缺失值产生的原因及影响等方面展开分析。

（1）缺失值产生的原因

1）有些信息暂时无法获取，或者获取信息的代价太大。

2）有些信息是被遗漏的。可能是因为输入时认为不重要、忘记填写或对数据理解错误等一些人为因素而遗漏，也可能是由于数据采集设备的故障、存储介质的故障、传输媒体的故障等非人为原因而丢失。

3）属性值不存在。在某些情况下，缺失值并不意味着数据有错误。对一些对象来说某些属性值是不存在的，如一个未婚者的配偶姓名、一个儿童的固定收入等。

（2）缺失值的影响

1）数据挖掘建模将丢失大量的有用信息。

2）数据挖掘模型所表现出的不确定性更加显著，模型中蕴涵的规律更难把握。

3）包含空值的数据会使建模过程陷入混乱，导致不可靠的输出。

（3）缺失值的分析

使用简单的统计分析，可以得到含有缺失值的属性的个数，以及每个属性的未缺失数、缺失数与缺失率等。

从总体上来说，缺失值的处理分为删除存在缺失值的记录、对可能值进行插补和不处理3种情况，将在4.1.1节详细介绍。

3.1.2 异常值分析

异常值分析是检验数据是否有录入错误以及含有不合常理的数据。忽视异常值的存在是十分危险的，不加剔除地把异常值包括进数据的计算分析过程中，对结果会产生不良影响；重视异常值的出现，分析其产生的原因，常常成为发现问题进而改进决策的契机。

异常值是指样本中的个别值，其数值明显偏离其余的观测值。异常值也称为离群点，异常值的分析也称为离群点分析。

（1）简单统计量分析

可以先对变量做一个描述性统计，进而查看哪些数据是不合理的。最常用的统计量是最大值和最小值，用来判断这个变量的取值是否超出了合理的范围。如客户年龄的最大值为199岁，则该变量的取值存在异常。

（2）3σ原则

如果数据服从正态分布，在3σ原则下，异常值被定义为一组测定值中与平均值的偏差超过3倍标准差的值。在正态分布的假设下，距离平均值3σ之外的值出现的概率为$P(|x-\mu|>3\sigma) \leqslant 0.003$，属于极个别的小概率事件。

如果数据不服从正态分布，也可以用远离平均值的多少倍标准差来描述。

（3）箱型图分析

箱型图提供了识别异常值的一个标准：异常值通常被定义为小于 $Q_L - 1.5IQR$ 或大于 $Q_U + 1.5IQR$ 的值。Q_L 称为下四分位数，表示全部观察值中有四分之一的数据取值比它小；Q_U 称为上四分位数，表示全部观察值中有四分之一的数据取值比它大；IQR 称为四分位数间距，是上四分位数 Q_U 与下四分位数 Q_L 之差，其间包含了全部观察值的一半。

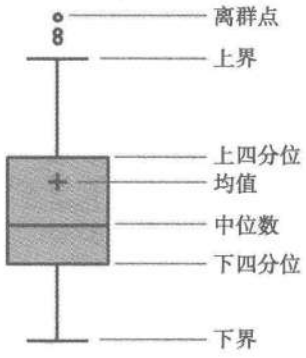


图 3-1 箱型图检测异常值

箱型图依据实际数据绘制，没有对数据作任何限制性要求（如服从某种特定的分布形式），它只是真实直观地表现数据分布的本来面貌；另一方面，箱型图判断异常值的标准以四分位数和四分位距为基础，四分位数具有一定的鲁棒性：多达 25% 的数据可以变得任意远而不会很大地扰动四分位数，所以异常值不能对这个标准施加影响。由此可见，箱型图识别异常值的结果比较客观，在识别异常值方面有一定的优越性，如图 3-1 所示。

在餐饮系统中的销量额数据可能出现缺失值和异常值，如表 3-1 中数据所示。

表3-1 餐饮日销额数据示例

时　间	2015/2/10	2015/2/11	2015/2/12	2015/2/13	2015/2/14
销量额（元）	2 742.8	3 014.3	865	3 036.8	

数据详见：demo/data/catering_sale.xls

分析餐饮系统日销量额数据可以发现，其中有部分数据是缺失的，但是如果数据记录和属性较多，使用人工分辨的方法就不切合实际，所以这里需要编写程序来检测出含有缺失值的记录和属性以及缺失率个数和缺失率等。

在 Python 的 Pandas 库中，只需要读入数据，然后使用 describe() 函数就可以查看数据的基本情况。

```
import pandas as pd
catering_sale = '../data/catering_sale.xls' #餐饮数据
data = pd.read_excel(catering_sale, index_col = u'日期') #读取数据，指定“日期”列为索引列
data.describe()
```

运行结果如下。

```
              销量
count   200.000000
mean   2755.214700
std     751.029772
min      22.000000
25%    2451.975000
50%    2655.850000
```

```
75%     3026.125000
max     9106.440000
```

其中 count 是非空值数，通过 len(data) 可以知道数据记录为 201 条，因此缺失值数为 1。另外，提供的基本参数还有平均值（mean）、标准差（std）、最小值（min）、最大值（max）以及 1/4、1/2、3/4 分位数（25%、50%、75%）。更直观地展示这些数据，并且可以检测异常值的方法是使用箱线图。其 Python 检测代码如代码清单 3-1 所示。

代码清单3-1 餐饮销额数据异常值检测代码

```
#-*- coding: utf-8 -*-
import pandas as pd

catering_sale = '../data/catering_sale.xls' #餐饮数据
data = pd.read_excel(catering_sale, index_col = u'日期') #读取数据，指定“日期”列为索
    引列

import matplotlib.pyplot as plt #导入图像库
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

plt.figure() #建立图像
p = data.boxplot() #画箱线图，直接使用DataFrame的方法
x = p['fliers'][0].get_xdata() # 'flies'即为异常值的标签
y = p['fliers'][0].get_ydata()
y.sort() #从小到大排序，该方法直接改变原对象

#用annotate添加注释
#其中有些相近的点，注解会出现重叠，难以看清，需要一些技巧来控制。
#以下参数都是经过调试的，需要具体问题具体调试。
for i in range(len(x)):
  if i>0:
    plt.annotate(y[i], xy = (x[i],y[i]), xytext=(x[i]+0.05 -0.8/(y[i]-y[i-
                          1]),y[i]))
  else:
    plt.annotate(y[i], xy = (x[i],y[i]), xytext=(x[i]+0.08,y[i]))

plt.show() #展示箱线图
```

代码详见：demo/code/ abnormal_check.py

运行上面的程序，其结果为“缺失值个数为：1”，同时可以得到如图 3-2 所示的箱型图。

从图 3-2 中可以看出，箱型图中的超过上下界的 7 个销售额数据可能为异常值。结合具体业务可以把 865、4060.3、4065.2 归为正常值，将 22、51、60、6607.4、9106.44 归为异常值。最后确定过滤规则为：日销量在 400 以下 5000 以上则属于异常数据，编写过滤程序，进行后续处理。

3.1.3 一致性分析

数据不一致性是指数据的矛盾性、不相容性。直接对不一致的数据进行挖掘，可能会产生与实际相违背的挖掘结果。

在数据挖掘过程中，不一致数据的产生主要发生在数据集成的过程中，这可能是由于被挖掘数据是来自于从不同的数据源、对于重复存放的数据未能进行一致性更新造成的。例如，两张表中都存储了用户的电话号码，但在用户的电话号码发生改变时只更新了一张表中的数据，那么这两张表中就有了不一致的数据。

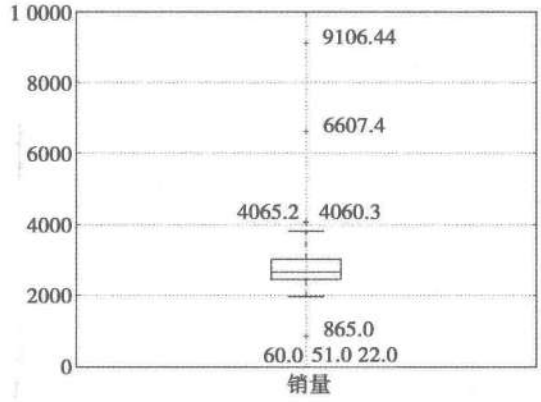


图 3-2 异常值检测箱型图

3.2 数据特征分析

对数据进行质量分析以后，接下来可通过绘制图表、计算某些特征量等手段进行数据的特征分析。

3.2.1 分布分析

分布分析能揭示数据的分布特征和分布类型。对于定量数据，欲了解其分布形式是对称的还是非对称的，发现某些特大或特小的可疑值，可通过绘制频率分布表、绘制频率分布直方图、绘制茎叶图进行直观地分析；对于定性分类数据，可用饼图和条形图直观地显示分布情况。

1. 定量数据的分布分析

对于定量变量而言，选择“组数”和“组宽”是做频率分布分析时最主要的问题，一般按照以下步骤进行。

1）求极差。

2）决定组距与组数。

3）决定分点。

4）列出频率分布表。

5）绘制频率分布直方图。

遵循的主要原则如下。

1）各组之间必须是相互排斥的。

2）各组必须将所有的数据包含在内。

3）各组的组宽最好相等。

下面结合具体实例，运用分布分析对定量数据进行特征分析。

表 3-2 是描述菜品“捞起生鱼片”在 2014 年第二个季度的销售数据，通过表中数据绘制销售量的频率分布表、频率分布图，对该定量数据做出相应的分析。

表3-2 “捞起生鱼片”的销售情况

日 期	销售额（元）	日 期	销售额（元）	日 期	销售额（元）
2014/4/1	420	2014/5/1	1770	2014/6/1	3960
2014/4/2	900	2014/5/2	135	2014/6/2	1770
2014/4/3	1290	2014/5/3	177	2014/6/3	3570
2014/4/4	420	2014/5/4	45	2014/6/4	2220
2014/4/5	1710	2014/5/5	180	2014/6/5	2700
…	…	…	…	…	…
2014/4/30	450	2014/5/30	2220	2014/6/30	2700
		2014/5/31	1800		

数据详见：demo/data/catering_sale.xls

（1）求极差

极差＝最大值－最小值＝ 3960 － 45 ＝ 3915

（2）分组

这里根据业务数据的含义，可取组距为 500。

组数＝极差 / 组距＝ 3915/500 ＝ 7.83 ⇒ 8

（3）决定分点

分布区间如表 3-3 所示。

表3-3 分布区间

[0，500）	[500，1000）	[1000，1500）	[1500，2000）
[2000，2500）	[2500，3000）	[3000，3500）	[3500，4000）

（4）绘制频率分布直方图[3]

根据分组区间得到如表 3-4 所示的频率分布表。其中，第 1 列将数据所在的范围分成若干组段，第 1 个组段要包括最小值，最后一个组段要包括最大值。习惯上将各组段设为左闭右开的半开区间，如第一个分组为 [0，500）。第 2 列组中值是各组段的代表值，由本组段的上、下限相加除以 2 得到。第 3 列和第 4 列分别为频数和频率。第 5 列是累计频率，是否需要计算该列视情况而定。

表3-4 频率分布表

组 段	组中值 x	频 数	频率 f	累计频率
[0，500）	250	15	16.48%	16.48%

（续）

组　段	组中值 x	频　数	频率 f	累计频率
[500，1000）	750	24	26.37%	42.85%
[1000，1500）	1250	17	18.68%	61.54%
[1500，2000）	1750	15	16.48%	78.02%
[2000，2500）	2250	9	9.89%	87.91%
[2500，3000）	2750	3	3.30%	92.31%
[3000，3500）	3250	4	4.40%	95.60%
[3500，4000）	3750	3	3.30%	98.90%
[4000，4500）	4250	1	1.10%	100.00%

（5）绘制频率分布直方图

若以2014年第二季度“捞起生鱼片”每天的销售额为横轴，以各组段的频率密度（频率与组距之比）为纵轴，表34的数据可绘制成频率分布直方图，如图3-3所示。

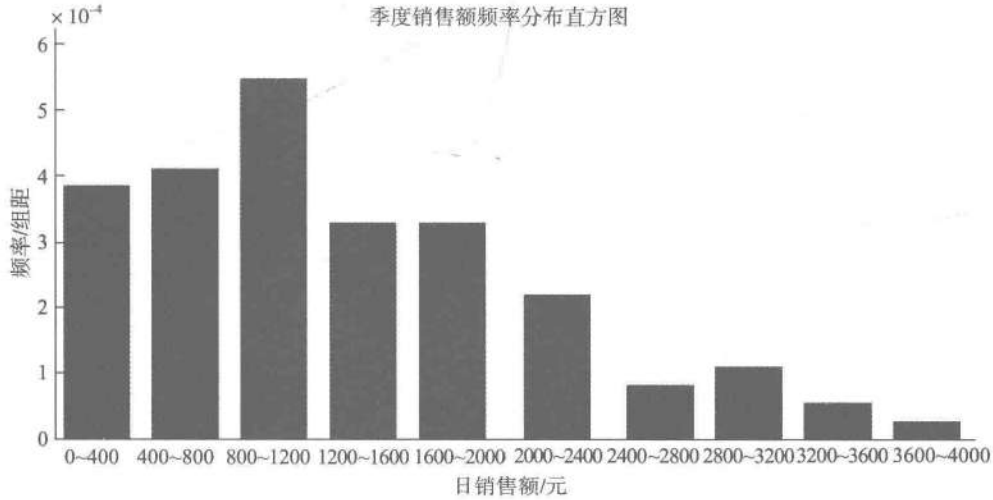


图3-3　销售额的频率分布直方图

2. 定性数据的分布分析

对于定性变量，常常根据变量的分类类型来分组，可以采用饼图和条形图来描述定性变量的分布。

饼图的每一个扇形部分代表每一类型的百分比或频数，根据定性变量的类型数目将饼图分成几个部分，每一部分的大小与每一类型的频数成正比；条形图的高度代表每一类型的百分比或频数，条形图的宽度没有意义。

图3-4和图3-5是菜品A、B、C在某段时间的销售量的分布图。

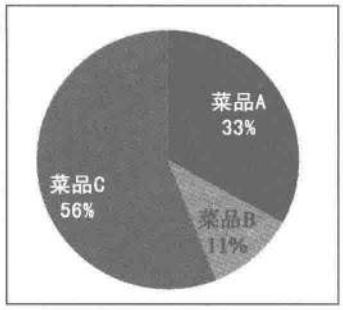


图 3-4 菜品销售量分布（饼图）

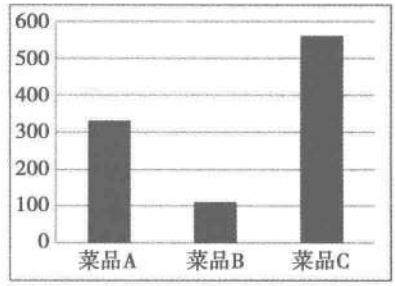


图 3-5 菜品的销售量分布（条形图）

3.2.2 对比分析

对比分析是指把两个相互联系的指标进行比较，从数量上展示和说明研究对象规模的大小，水平的高低，速度的快慢，以及各种关系是否协调。特别适用于指标间的横纵向比较、时间序列的比较分析。在对比分析中，选择合适的对比标准是十分关键的步骤，只有选择合适，才能做出客观的评价，选择不合适，评价可能得出错误的结论。

对比分析主要有以下两种形式。

（1）绝对数比较

绝对数比较是利用绝对数进行对比，从而寻找差异的一种方法。

（2）相对数比较

相对数比较是由两个有联系的指标对比计算的，用以反映客观现象之间数量联系程度的综合指标，其数值表现为相对数。由于研究目的和对比基础不同，相对数可以分为以下几种。

1）结构相对数：将同一总体内的部分数值与全部数值对比求得比重，用以说明事物的性质、结构或质量。如居民食品支出额占消费支出总额比重、产品合格率等。

2）比例相对数：将同一总体内不同部分的数值进行对比，表明总体内各部分的比例关系。如人口性别比例、投资与消费比例等。

3）比较相对数：将同一时期两个性质相同的指标数值进行对比，说明同类现象在不同空间条件下的数量对比关系。如不同地区商品价格对比，不同行业、不同企业间某项指标对比等。

4）强度相对数：将两个性质不同但有一定联系的总量指标进行对比，用以说明现象的强度、密度和普遍程度。如人均国内生产总值用“元 / 人”表示，人口密度用“人 / 平方公里”表示，也有用百分数或千分数表示的，如人口出生率用 ‰ 表示。

5）计划完成程度相对数：是某一时期实际完成数与计划数的对比，用以说明计划完成程度。

6）动态相对数：将同一现象在不同时期的指标数值进行对比，用以说明发展方向和变化的速度。如发展速度、增长速度等。

就各菜品的销售数据来看，从时间的维度上分析，可以看到甜品部A、海鲜部B、素菜部C三个部门之间的销售金额随时间的变化趋势，可以了解在此期间哪个部门的销售金额较高，趋势比较平稳，如图3-6所示。

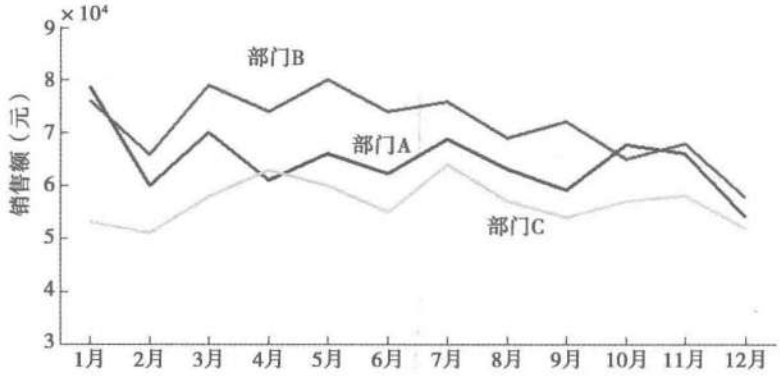


图3-6 部门之间销售金额的比较

从总体来看，三个部门的销售金额呈递减趋势；A部门和C部门的递减趋势比较平稳；B部门的销售金额下降的趋势比较明显，可以进一步分析造成这种现象的原因，可能是原材料不足造成的。

我们也可以对单一部门（如海鲜部）做分析，了解各月份的销售对比情况，如图3-7所示。

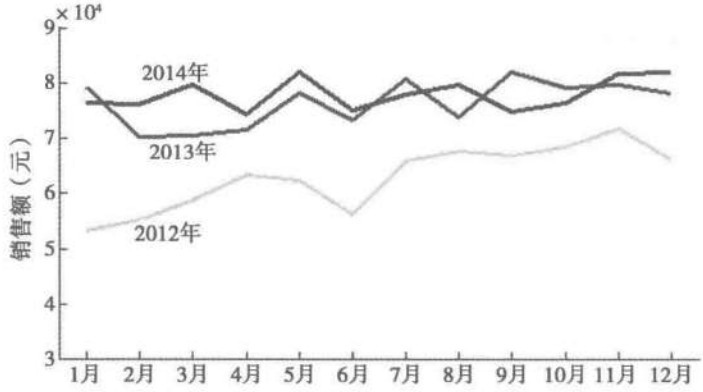


图3-7 海鲜部各年份之间月销售金额的比较

3.2.3 统计量分析

用统计指标对定量数据进行统计描述，常从集中趋势和离中趋势两个方面进行分析。

平均水平的指标是对个体集中趋势的度量，使用最广泛的是均值和中位数；反映变异程度的指标则是对个体离开平均水平的度量，使用较广泛的是标准差（方差）、四分位间距。

1. 集中趋势度量

（1）均值

均值是所有数据的平均值。

如果求 n 个原始观察数据的平均数，计算公式为：

$$\text{mean}(x) = \bar{x} = \frac{\Sigma x_i}{n} \tag{3-1}$$

有时，为了反映在均值中不同成分所占的不同重要程度，为数据集中的每一个 x_i 赋予 w_i，这就得到了加权均值的计算公式：

$$\text{mean}(x) = \bar{x} = \frac{\Sigma w_i x_i}{\Sigma w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n} \tag{3-2}$$

类似地，频率分布表（见表 3-4）的平均数可以使用下式计算：

$$\text{mean}(x) = \bar{x} = \Sigma f_i x_i = f_1 x_1 + f_2 x_2 + \cdots f_k x_k \tag{3-3}$$

式中，$x_1, x_2, \cdots, x_k$ 分别为 k 个组段的组中值；$f_1, f_2, \cdots, f_k$ 分别为 k 个组段的频率。这里的 f_i 起了权重的作用。

作为一个统计量，均值的主要问题是对极端值很敏感。如果数据中存在极端值或者数据是偏态分布的，那么均值就不能很好地度量数据的集中趋势。为了消除少数极端值的影响，可以使用截断均值或者中位数来度量数据的集中趋势。截断均值是去掉高、低极端值之后的平均数。

（2）中位数

中位数是将一组观察值按从小到大的顺序排列，位于中间的那个数。即在全部数据中，小于和大于中位数的数据个数相等。

将某一数据集 x:$\{x_1, x_2, \cdots, x_n\}$ 按从小到大排序：$\{x_{(1)}, x_{(2)}, \cdots, x_{(n)}\}$。

当 n 为奇数时

$$M = x_{(\frac{n+1}{2})} \tag{3-4}$$

当 n 为偶数时

$$M = \frac{1}{2}(x_{(\frac{n}{2})} + x_{(\frac{n+1}{2})}) \tag{3-5}$$

（3）众数

众数是指数据集中出现最频繁的值。众数并不经常用来度量定性变量的中心位置，更适用于定性变量。众数不具有唯一性。当然，众数一般用于离散型变量而非连续型变量。

2. 离中趋势度量

（1）极差

极差＝最大值－最小值

极差对数据集的极端值非常敏感，并且忽略了位于最大值与最小值之间的数据的分布情况。

（2）标准差

标准差度量数据偏离均值的程度，计算公式为：

$$s = \sqrt{\frac{\Sigma (x_i - \bar{x})^2}{n}} \tag{3-6}$$

（3）变异系数

变异系数度量标准差相对于均值的离中趋势，计算公式为：

$$\mathrm{CV} = \frac{s}{\bar{x}} \times 100\% \tag{3-7}$$

变异系数主要用来比较两个或多个具有不同单位或不同波动幅度的数据集的离中趋势。

（4）四分位数间距

四分位数包括上四分位数和下四分位数。将所有数值由小到大排列并分成四等份，处于第一个分割点位置的数值是下四分位数，处于第二个分割点位置（中间位置）的数值是中位数，处于第三个分割点位置的数值是上四分位数。

四分位数间距，是上四分位数 Q_U 与下四分位数 Q_L 之差，其间包含了全部观察值的一半。其值越大，说明数据的变异程度越大；反之，说明变异程度越小。

前面已经提过，DataFrame 对象的 describe() 方法已经可以给出一些基本的统计量，根据给出的统计量，可以衍生出我们所需要的统计量。针对餐饮销量数据进行统计量分析，其 Python 代码如代码清单 3-2 所示。

代码清单3-2 餐饮销量数据统计量分析代码

```
#-*- coding: utf-8 -*-
#餐饮销量数据统计量分析
from __future__ import print_function
import pandas as pd

catering_sale = '../data/catering_sale.xls' #餐饮数据
data = pd.read_excel(catering_sale, index_col = u'日期') #读取数据，指定“日期”列为索引列
data = data[(data[u'销量'] > 400)&(data[u'销量'] < 5000)] #过滤异常数据
statistics = data.describe() #保存基本统计量

statistics.loc['range'] = statistics.loc['max']-statistics.loc['min'] #极差
statistics.loc['var'] = statistics.loc['std']/statistics.loc['mean'] #变异系数
statistics.loc['dis'] = statistics.loc['75%']-statistics.loc['25%'] #四分位数间距

print(statistics)
```

代码详见：demo/code/statistics_analyze.py

运行上面的程序，可以得到下面的结果，此结果为餐饮销量数的统计量情况。

	销量
count	195.000000
mean	2744.595385
std	424.739407
min	865.000000
25%	2460.600000
50%	2655.900000
75%	3023.200000
max	4065.200000
range	3200.200000
var	0.154755
dis	562.600000

3.2.4 周期性分析

周期性分析是探索某个变量是否随着时间变化而呈现出某种周期变化趋势。时间尺度相对较长的周期性趋势有年度周期性趋势、季节性周期趋势，相对较短的有月度周期性趋势、周度周期性趋势，甚至更短的天、小时周期性趋势。

例如，要对某单位用电量进行预测，可以先分析该用电单位日用电量的时序图来直观地估计其用电量变化趋势。

图 3-8 是某用电单位 A 在 2014 年 9 月日用电量的时序图，图 3-9 是用电单位 A 在 2013 年 9 月日用电量的时序图。

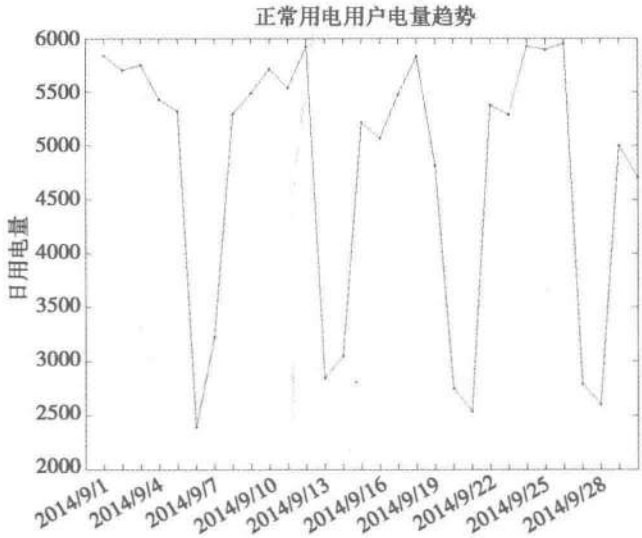


图 3-8 2014 年 9 月日用电量时序图

总体来看，用电单位 A 的 2014 年 9 月日用电量呈现出周期性，以周为周期，因为周六周日不上班，所以周末用电量较低。工作日和非工作日的用电量比较平稳，没有太大的波动。而 2013 年 9 月日用电量总体呈现出递减的趋势，同样周末的用电量是最低的。

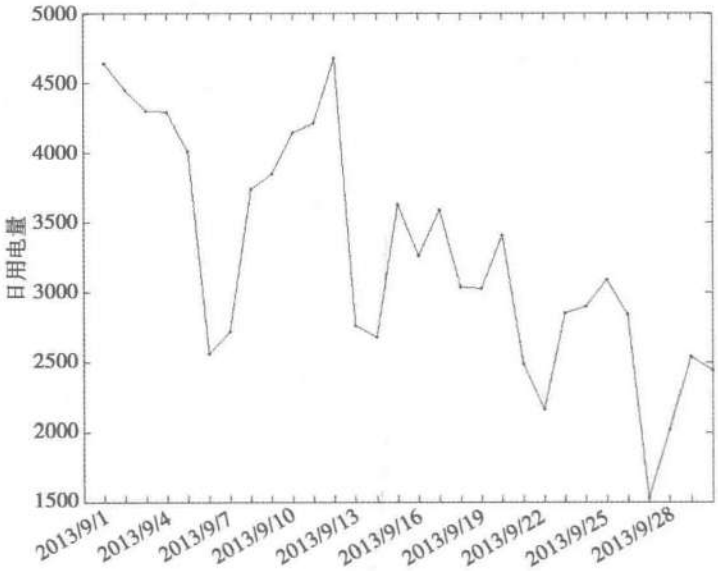


图 3-9 2013 年 9 月日用电量时序图

3.2.5 贡献度分析

贡献度分析又称帕累托分析，它的原理是帕累托法则，又称 20/80 定律。同样的投入放在不同的地方会产生不同的效益。例如，对一个公司来讲，80% 的利润常常来自于 20% 最畅销的产品，而其他 80% 的产品只产生了 20% 的利润。

对餐饮企业来讲，应用贡献度分析可以重点改善某菜系盈利最高的前 80% 的菜品，或者重点发展综合影响最高的 80% 的部门。这种结果可以通过帕累托图直观地呈现出来。图 3-10 是海鲜系列的 10 个菜品 A1 ～ A10 某个月的盈利额（已按照从大到小排序）。

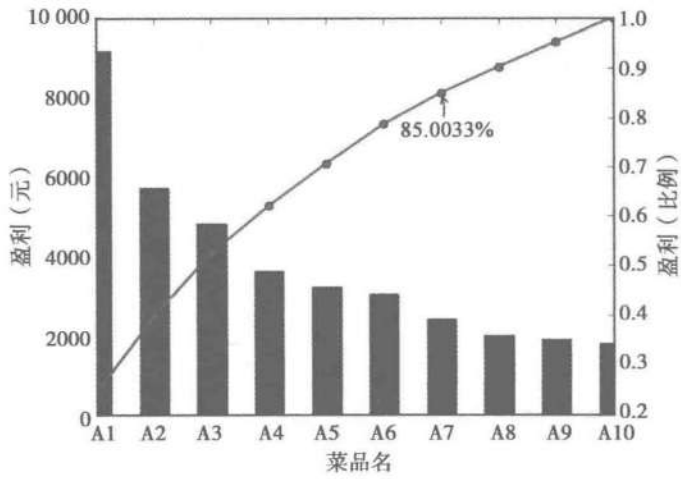


图 3-10 帕累托图

由上图可知，菜品 A1 ～ A7 共 7 个菜品，占菜品种类数的 70%，总盈利额占该月盈利额的 85.0033%。根据帕累托法则，应该增加对菜品 A1 ～ A7 的成本投入，减少对菜品 A8 ～ A10 的投入以获得更高的盈利额。

表 3-5 是餐饮系统对应的菜品盈利数据。

表3-5 餐饮系统菜品盈利数据

菜品 ID	17148	17154	109	117	17151
菜品名	A1	A2	A3	A4	A5
盈利：元	9173	5729	4811	3594	3195
菜品 ID	14	2868	397	88	426
菜品名	A6	A7	A8	A9	A10
盈利：元	3026	2378	1970	1877	1782

数据详见：demo/data/catering_dish_profit.xls

其 Python 代码如代码清单 3-3 所示。

代码清单3-3 菜品盈利帕累托图代码

```
#-*- coding: utf-8 -*-
#菜品盈利数据 帕累托图
from __future__ import print_function
import pandas as pd

#初始化参数
dish_profit = '../data/catering_dish_profit.xls' #餐饮菜品盈利数据
data = pd.read_excel(dish_profit, index_col = u'菜品名')
data = data[u'盈利'].copy()
data.sort(ascending = False)

import matplotlib.pyplot as plt #导入图像库
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

plt.figure()
data.plot(kind='bar')
plt.ylabel(u'盈利（元）')
p = 1.0*data.cumsum()/data.sum()
p.plot(color = 'r', secondary_y = True, style = '-o',linewidth = 2)
plt.annotate(format(p[6], '.4%'), xy = (6, p[6]), xytext=(6*0.9, p[6]*0.9), arro
    wprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2")) #添加注释，即85%处
    的标记。这里包括了指定箭头样式。
plt.ylabel(u'盈利（比例）')
plt.show()
```

代码详见：demo/code/dish_pareto.py

3.2.6 相关性分析

分析连续变量之间线性相关程度的强弱，并用适当的统计指标表示出来的过程称为相关分析。

1. 直接绘制散点图

判断两个变量是否具有线性相关关系的最直观的方法是直接绘制散点图，如图 3-11 所示。

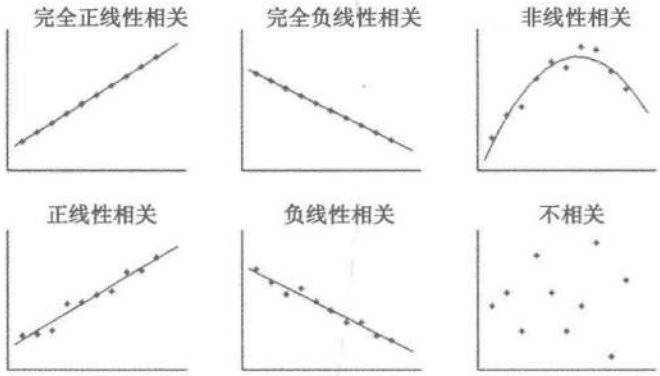


图 3-11 相关关系的图示

2. 绘制散点图矩阵

需要同时考察多个变量间的相关关系时，一一绘制它们间的简单散点图是十分麻烦的。此时可利用散点图矩阵同时绘制各变量间的散点图，从而快速发现多个变量间的主要相关性，这在进行多元线性回归时显得尤为重要。

散点图矩阵如图 3-12 所示。

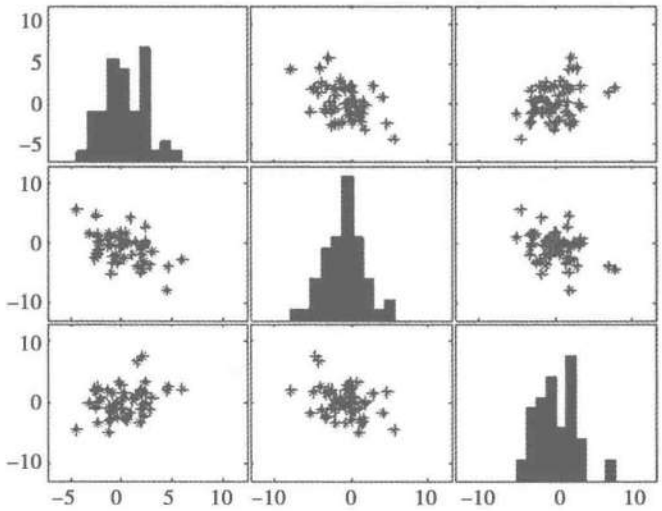


图 3-12 散点图矩阵

3. 计算相关系数

为了更加准确地描述变量之间的线性相关程度，可以通过计算相关系数来进行相关分析。在二元变量的相关分析过程中比较常用的有 Pearson 相关系数、Spearman 秩相关系数和判定系数。

（1）Pearson 相关系数

一般用于分析两个连续性变量之间的关系，其计算公式如下。

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3-8}$$

相关系数 r 的取值范围：$-1 \leqslant r \leqslant 1$

$$\begin{cases} r > 0 \text{ 为正相关}, r < 0 \text{ 为负相关} \\ |r| = 0 \text{ 表示不存在线性关系} \\ |r| = 1 \text{ 表示完全线性相关} \end{cases}$$

$0<|r|<1$ 表示存在不同程度线性相关：

$$\begin{cases} |r| \leqslant 0.3 \text{ 为不存在线性相关} \\ 0.3 < |r| \leqslant 0.5 \text{ 为低度线性相关} \\ 0.5 < |r| \leqslant 0.8 \text{ 为显著线性相关} \\ |r| > 0.8 \text{ 为高度线性相关} \end{cases}$$

（2）Spearman 秩相关系数

Pearson 线性相关系数要求连续变量的取值服从正态分布。不服从正态分布的变量、分类或等级变量之间的关联性可采用 Spearman 秩相关系数，也称等级相关系数来描述。

其计算公式如下。

$$r_s = 1 - \frac{6\sum_{i=1}^{n}(R_i - Q_i)^2}{n(n^2 - 1)} \tag{3-9}$$

对两个变量成对的取值分别按照从小到大（或者从大到大小）顺序编秩，R_i 代表 x_i 的秩次，Q_i 代表 y_i 的秩次，R_i–Q_i 为 x_i、y_i 的秩次之差。

表 3-6 给出一个变量 $x(x_1, x_2, \cdots x_i, \cdots x_n)$ 秩次的计算过程。

表3-6 变量x秩次计算过程

x_i 从小到大排序	从小到大排序时的位置	秩次 R_i
0.5	1	1
0.8	2	2
1.0	3	3

（续）

x_i 从小到大排序	从小到大排序时的位置	秩次 R_i
1.2	4	(4+5)/2 = 4.5
1.2	5	(4+5)/2 = 4.5
2.3	6	6
2.8	7	7

因为一个变量的相同的取值必须有相同的秩次，所以在计算中采用的秩次是排序后所在位置的平均值。

只要两个变量具有严格单调的函数关系，那么它们就是完全 Spearman 相关的，这与 Pearson 相关不同，Pearson 相关只有在变量具有线性关系时才是完全相关的。

在实际应用计算中，上述两种相关系数都要对其进行假设检验，使用 t 检验方法检验其显著性水平以确定其相关程度。研究表明，在正态分布假定下，Spearman 秩相关系数与 Pearson 相关系数在效率上是等价的，而对于连续测量数据，更适合用 Pearson 相关系数来进行分析。

（3）判定系数

判定系数是相关系数的平方，用 r^2 表示；用来衡量回归方程对 y 的解释程度。判定系数取值范围：$0 \leqslant r^2 \leqslant 1$。$r^2$ 越接近于 1，表明 x 与 y 之间的相关性越强；r^2 越接近于 0，表明两个变量之间几乎没有直线相关关系。

餐饮系统中可以统计得到不同菜品的日销量数据，数据示例如表 3-7 所示。

表3-7 菜品日销量数据

日 期	百合酱蒸凤爪	翡翠蒸香茜饺	金银蒜汁蒸排骨	乐膳真味鸡	蜜汁焗餐包	生炒菜心	铁板酸菜豆腐	香煎韭菜饺	香煎萝卜糕	原汁原味菜心
2015/1/1	17	6	8	24	13	13	18	10	10	27
2015/1/2	11	15	14	13	9	10	19	13	14	13
2015/1/3	10	8	12	13	8	3	7	11	10	9
2015/1/4	9	6	6	3	10	9	9	13	14	13
2015/1/5	4	10	13	8	12	10	17	11	13	14
2015/1/6	13	10	13	16	8	9	12	11	5	9

数据详见：demo/data/catering_sale_all.xls

分析这些菜品销售量之间的相关性可以得到不同菜品之间的关系，比如是替补菜品、互补菜品或者没有关系，为原材料采购提供参考。其 Python 代码如代码清单 3-4 所示。

代码清单3-4 餐饮销量数据相关性分析

```
#-*- coding: utf-8 -*-
#餐饮销量数据相关性分析
```

```
from __future__ import print_function
import pandas as pd

catering_sale = '../data/catering_sale_all.xls' #餐饮数据，含有其他属性
data = pd.read_excel(catering_sale, index_col = u'日期') #读取数据，指定“日期”列为索引列

data.corr() #相关系数矩阵，即给出了任意两款菜式之间的相关系数
data.corr()[u'百合酱蒸凤爪'] #只显示“百合酱蒸凤爪”与其他菜式的相关系数
data[u'百合酱蒸凤爪'].corr(data[u'翡翠蒸香茜饺']) #计算“百合酱蒸凤爪”与“翡翠蒸香茜饺”的
    相关系数
```

代码详见：demo/code/correlation_analyze.py

上面的代码给出了 3 种不同形式的求相关系数的运算。运行代码，可以得到任意两款菜式之间的相关系数，如运行“data.corr()[u' 百合酱蒸凤爪 ']”可以得到下面的结果。

```
>>> data.corr()[u'百合酱蒸凤爪']
百合酱蒸凤爪      1.000000
翡翠蒸香茜饺      0.009206
金银蒜汁蒸排骨     0.016799
乐膳真味鸡       0.455638
蜜汁焗餐包       0.098085
生炒菜心        0.308496
铁板酸菜豆腐      0.204898
香煎韭菜饺       0.127448
香煎萝卜糕      -0.090276
原汁原味菜心      0.428316
Name: 百合酱蒸凤爪, dtype: float64
```

从上面的结果可以看到，如果顾客点了“百合酱蒸凤爪”，则和点“翡翠蒸香茜饺”“金银蒜汁蒸排骨”“香煎萝卜糕”“铁板酸菜豆腐”“香煎韭菜饺”等主食类的相关性比较低，反而点“乐膳真味鸡”“生炒菜心”“原汁原味菜心”的相关性比较高。

3.3 Python 主要数据探索函数

Python 中用于数据探索的库主要是 Pandas（数据分析）和 Matplotlib（数据可视化）。其中，Pandas 提供了大量的与数据探索相关的函数，这些数据探索函数可大致分为统计特征函数与统计作图函数，而作图函数依赖于 Matplotlib，所以往往又会跟 Matplotlib 结合在一起使用。本节对 Pandas 中主要的统计特征函数与统计作图函数进行介绍，并举例以方便理解。

3.3.1 基本统计特征函数

统计特征函数用于计算数据的均值、方差、标准差、分位数、相关系数和协方差等，这些统计特征能反映出数据的整体分布。本小节所介绍的统计特征函数如表 3-8 所示，它们主要作为 Pandas 的对象 DataFrame 或 Series 的方法出现。

表3-8 Pandas主要统计特征函数

方 法 名	函 数 功 能	所 属 库
sum()	计算数据样本的总和（按列计算）	Pandas
mean()	计算数据样本的算术平均数	Pandas
var()	计算数据样本的方差	Pandas
std()	计算数据样本的标准差	Pandas
corr()	计算数据样本的 Spearman（Pearson）相关系数矩阵	Pandas
cov()	计算数据样本的协方差矩阵	Pandas
skew()	样本值的偏度（三阶矩）	Pandas
kurt()	样本值的峰度（四阶矩）	Pandas
describe()	给出样本的基本描述（基本统计量如均值、标准差等）	Pandas

（1）sum

- 功能：计算数据样本的总和（按列计算）。
- 使用格式：

D.sum()

按列计算样本 D 的总和，样本 D 可为 DataFrame 或者 Series。

（2）mean

- 功能：计算数据样本的算术平均数。
- 使用格式：

D.mean()

按列计算样本 D 的均值，样本 D 可为 DataFrame 或者 Series。

（3）var

- 功能：计算数据样本的方差。
- 使用格式：

D.var()

按列计算样本 D 的均值，样本 D 可为 DataFrame 或者 Series。

（4）std

- 功能：计算数据样本的标准差。
- 使用格式：

D.std()

按列计算样本 D 的均值，样本 D 可为 DataFrame 或者 Series。

（5）corr

- 功能：计算数据样本的 Spearman（Pearson）相关系数矩阵。
- 使用格式：

D.corr(method='pearson')

样本 D 可为 DataFrame，返回相关系数矩阵，method 参数为计算方法，支持 pearson（皮尔森相关系数，默认选项）、kendall（肯德尔系数）、spearman（斯皮尔曼系数）；

S1.corr(S2, method='pearson') S1、S2 均为 Series，这种格式指定计算两个 Series 之间的相关系数。

❑ 实例：计算两个列向量的相关系数，采用 Spearman 方法。

```
D = pd.DataFrame([range(1, 8), range(2, 9)]) #生成样本D，一行为1~7，一行为2~8
D.corr(method='spearson') #计算相关系数矩阵
S1 = D.loc[0] #提取第一行
S2 = D.loc[1] #提取第二行
S1.corr(S2, method='pearson') #计算S1、S2的相关系数
```

（6）cov

❑ 功能：计算数据样本的协方差矩阵。

❑ 使用格式：

D.cov()

样本 D 可为 DataFrame，返回协方差矩阵；

S1.cov(S2) S1、S2 均为 Series，这种格式指定计算两个 Series 之间的协方差。

❑ 实例：计算 6×5 随机矩阵的协方差矩阵。

```
import numpy as np
D = pd.DataFrame(np.random.randn(6, 5)) #产生6×5随机矩阵
D.cov() #计算协方差矩阵
          0         1         2         3         4
0  1.745257 -0.299968  0.850216 -0.484931  1.068187
1 -1.453670  1.460928  0.347299  1.585089  0.595347
2 -0.751128  0.504498 -1.244944 -0.672183 -0.595296
3 -0.423802 -1.086470  0.637264  0.873043 -0.506736
4  0.969907  0.721997 -0.550993  1.033300 -0.903234
5 -0.705159  0.385077  0.120580  0.347470  2.036798
D[0].cov(D[1]) #计算第一列和第二列的协方差
0.5
```

（7）skew/kurt

❑ 功能：计算数据样本的偏度（三阶矩）/ 峰度（四阶矩）。

❑ 使用格式：

D.skew() / D.kurt()

计算样本 D 的偏度（三阶矩）/ 峰度（四阶矩）。样本 D 可为 DataFrame 或 Series。

❑ 实例：计算 6×5 随机矩阵的偏度（三阶矩）/ 峰度（四阶矩）。

```
import numpy as np
D = pd.DataFrame(np.random.randn(6, 5)) #产生6×5随机矩阵
D.skew()
```

```
0   -0.210246
1   -0.348367
2   -1.152183
3   -0.378802
4   -0.859889
dtype: float64
D.kurt()
0   -0.191062
1   -1.831973
2    1.171797
3   -1.529854
4    1.494526
dtype: float64
```

（8）describe

- 功能：直接给出样本数据的一些基本的统计量，包括均值、标准差、最大值、最小值、分位数等。
- 使用格式：

D.describe()

括号里可以带一些参数，比如 percentiles = [0.2, 0.4, 0.6, 0.8] 就是指定只计算 0.2、0.4、0.6、0.8 分位数，而不是默认的 1/4、1/2、3/4 分位数。

- 实例：给出 6×5 随机矩阵的 describe。

```
import numpy as np
D = pd.DataFrame(np.random.randn(6, 5)) #产生6×5随机矩阵
D.describe()
              0         1         2         3         4
count  6.000000  6.000000  6.000000  6.000000  6.000000
mean   0.006958 -0.069822  0.113711 -0.168115 -0.584493
std    1.224979  1.017829  0.939980  1.173083  0.539911
min   -1.777763 -1.330542 -1.512842 -1.674685 -1.507229
25%   -0.669088 -0.937504 -0.202329 -1.109370 -0.721853
50%    0.176010  0.130924  0.472093  0.115791 -0.537366
75%    0.578993  0.650975  0.516907  0.538483 -0.305514
max    1.704960  1.119084  1.146215  1.272789  0.086585
```

3.3.2 拓展统计特征函数

除了上述基本的统计特征外，Pandas 还提供了一些非常方便实用的计算统计特征的函数，主要有累积计算（cum）和滚动计算（pd.rolling_），见表 3-8 和表 3-9。

表3-9 Pandas累积统计特征函数

方 法 名	函 数 功 能	所 属 库
cumsum()	依次给出前 1、2、…、n 个数的和	Pandas
cumprod()	依次给出前 1、2、…、n 个数的积	Pandas

（续）

方 法 名	函 数 功 能	所 属 库
cummax()	依次给出前 1、2、…、n 个数的最大值	Pandas
cummin()	依次给出前 1、2、…、n 个数的最小值	Pandas

表3-10 Pandas累积统计特征函数

方 法 名	函 数 功 能	所 属 库
rolling_sum()	计算数据样本的总和（按列计算）	Pandas
rolling_mean()	数据样本的算术平均数	Pandas
rolling_var()	计算数据样本的方差	Pandas
rolling_std()	计算数据样本的标准差	Pandas
rolling_corr()	计算数据样本的 Spearman（Pearson）相关系数矩阵	Pandas
rolling_cov()	计算数据样本的协方差矩阵	Pandas
rolling_skew()	样本值的偏度（三阶矩）	Pandas
rolling_kurt()	样本值的峰度（四阶矩）	Pandas

其中，cum 系列函数是作为 DataFrame 或 Series 对象的方法而出现的，因此命令格式为 D.cumsum()，而 rolling_ 系列是 pandas 的函数，不是 DataFrame 或 Series 对象的方法，因此，它们的使用格式为 pd.rolling_mean(D, k)，意思是每 k 列计算一次均值，滚动计算。

实例：

```
D=pd.Series(range(0, 20)) #构造Series，内容为0~19共20个整数
D.cumsum() #给出前n项和
0       0
1       1
2       3
3       6
.....
19    190
dtype: int32
pd.rolling_sum(D, 2) #依次对相邻两项求和
0     NaN
1       1
2       3
3       5
.....
19    37
dtype: float64
```

3.3.3 统计作图函数

通过统计作图函数绘制的图表可以直观地反映出数据及统计量的性质及其内在规律，如

盒图可以表示多个样本的均值，误差条形图能同时显示下限误差和上限误差，最小二乘拟合曲线图能分析两变量间的关系。

Python 的主要作图库是 Matplotlib，在第 2 章中已经进行了初步的介绍，而 Pandas 基于 Matplotlib 并对某些命令进行了简化，因此作图通常是 Matplotlib 和 Pandas 相互结合着使用。本小节仅对一些基本的作图函数做一下简介，而真正灵活地使用应当参考书中所给出的各个作图代码清单。我们要介绍的统计作图函数如表 3-8 所示。

表3-11 Python主要统计作图函数

作图函数名	作图函数功能	所属工具箱
plot()	绘制线性二维图，折线图	Matplotlib/Pandas
pie()	绘制饼型图	Matplotlib/Pandas
hist()	绘制二维条形直方图，可显示数据的分配情形	Matplotlib/Pandas
boxplot()	绘制样本数据的箱形图	Pandas
plot(logy = True)	绘制 y 轴的对数图形	Pandas
plot(yerr = error)	绘制误差条形图	Pandas

在作图之前，通常要加载以下代码。

```
import matplotlib.pyplot as plt #导入作图库
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
plt.figure(figsize = (7, 5)) #创建图像区域，指定比例
```

作图完成后，一般通过 plt.show() 来显示作图结果。

（1）plot

- 功能：绘制线性二维图、折线图。
- 使用格式：

plt.plot(x, y, S)

这是 Matplotlib 通用的绘图方式，绘制 y 对于 x（即以 x 为横轴的二维图形），字符串参量 S 指定绘制时图形的类型、样式和颜色，常用的选项有：'b' 为蓝色、'r' 为红色、'g' 为绿色、'o' 为圆圈、'+' 为加号标记、'–' 为实线、'--' 为虚线。当 x、y 均为实数同维向量时，则描出点 $(x(i), y(i))$，然后用直线依次相连。

D.plot(kind = 'box')

这里使用的是 DataFrame 或 Series 对象内置的方法作图，默认以 Index 为横坐标，每列数据为纵坐标自动作图，通过 kind 参数指定作图类型，支持 line(线)、bar(条形)、barh、hist(直方图)、box（箱线图）、kde（密度图）和 area、pie（饼图）等，同时也能够接受 plt.plot() 中接受的参数。因此，如果数据已经被加载为 Pandas 中的对象，那么以这种方式作图是比较简洁的。

❑ 实例：在区间（$0 \leqslant x \leqslant 2\pi$）绘制一条蓝色的正弦虚线，并在每个坐标点标上五角星。绘制图形如图 3-13 所示。

```
import numpy as np
x = np.linspace(0,2*np.pi,50) #x坐标输入
y = np.sin(x) #计算对应x的正弦值
plt.plot(x, y, 'bp--') #控制图形格式为蓝色带星虚线，显示正弦曲线
plt.show()
```

（2）pie

❑ 功能：绘制饼型图。

❑ 使用格式：

plt.pie(size)

使用 Matplotlib 绘制饼图，其中 size 是一个列表，记录各个扇形的比例。pie 有丰富的参数，详情请参考下面的实例。

❑ 实例：通过向量 [15, 30, 45, 10] 画饼图，注上标签，并将第 2 部分分离出来。绘制结果如图 3-14 所示。

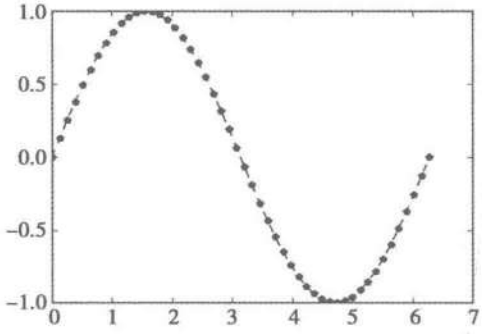


图 3-13 正弦曲线图

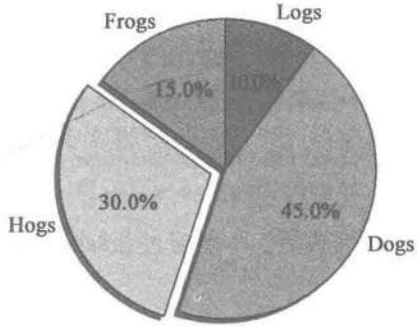


图 3-14 饼形图

```
import matplotlib.pyplot as plt

# The slices will be ordered and plotted counter-clockwise.
labels = 'Frogs', 'Hogs', 'Dogs', 'Logs' #定义标签
sizes = [15, 30, 45, 10] #每一块的比例
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral'] #每一块的颜色
explode = (0, 0.1, 0, 0) #突出显示，这里仅仅突出显示第二块（即'Hogs'）

plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',
    shadow=True, startangle=90)
plt.axis('equal') #显示为圆（避免比例压缩为椭圆）
plt.show()
```

（3）hist

❑ 功能：绘制二维条形直方图，可显示数据的分布情形。

- 使用格式：

Plt.hist(x, y)

其中，x 是待绘制直方图的一维数组，y 可以是整数，表示均匀分为 n 组；也可以是列表，列表各个数字为分组的边界点（即手动指定分界点）。

- 实例：绘制二维条形直方图，随机生成有 1000 个元素的服从正态分布的数组，分成 10 组绘制直方图。绘制结果如图 3-15 所示。

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.randn(1000) #1000个服从正态分布的随机数
plt.hist(x, 10) #分成10组进行绘制直方图
plt.show()
```

（4）boxplot

- 功能：绘制样本数据的箱形图。
- 使用格式：

D.boxplot() / D.plot(kind = 'box')

有两种比较简单的方式绘制 D 的箱形图，其中一种是直接调用 DataFrame 的 boxplot() 方法；另外一种是调用 Series 或者 DataFrame 的 plot() 方法，并用 kind 参数指定箱形图（box）。其中，盒子的上、下四分位数和中值处有一条线段。箱形末端延伸出去的直线称为须，表示盒外数据的长度。如果在须外没有数据，则在须的底部有一点，点的颜色与须的颜色相同。

- 实例：绘制样本数据的箱形图，样本由两组正态分布的随机数据组成。其中，一组数据均值为 0，标准差为 1，另一组数据均值为 1，标准差为 1。绘制结果如图 3-16 所示。

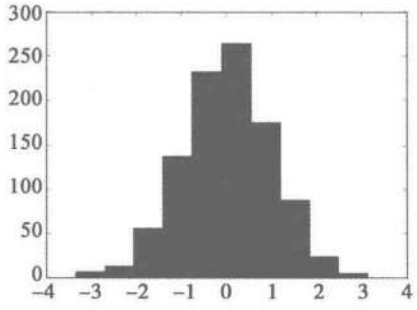


图 3-15 二维条形直方图

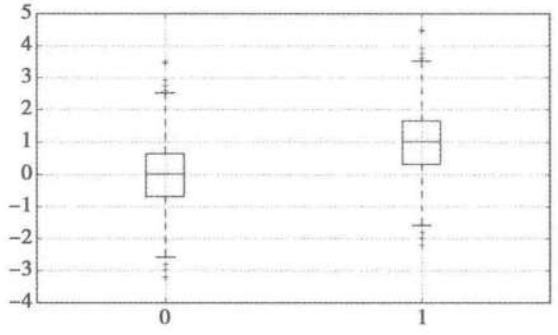


图 3-16 箱形图

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
x = np.random.randn(1000) #1000个服从正态分布的随机数
D = pd.DataFrame([x, x+1]).T #构造两列的DataFrame
D.plot(kind = 'box') #调用Series内置的作图方法画图，用kind参数指定箱形图box
plt.show()
```

（5）plot(logx = True) / plot(logy = True)

- 功能：绘制 x 或 y 轴的对数图形。
- 使用格式：

D.plot(log$x$ = True) / D.plot(log$y$ = True)

对 x 轴（y 轴）使用对数刻度（以 10 为底），y 轴（x 轴）使用线性刻度，进行 plot 函数绘图，D 为 Pandas 的 DataFrame 或者 Series。

- 实例：构造指数函数数据使用 plot(logy = True) 函数进行绘图，绘制结果如图 3-17 所示。

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
import numpy as np
import pandas as pd

x = pd.Series(np.exp(np.arange(20))) #原始数据
x.plot(label = u'原始数据图', legend = True)
plt.show()
x.plot(logy = True, label = u'对数数据图', legend = True)
plt.show()
```

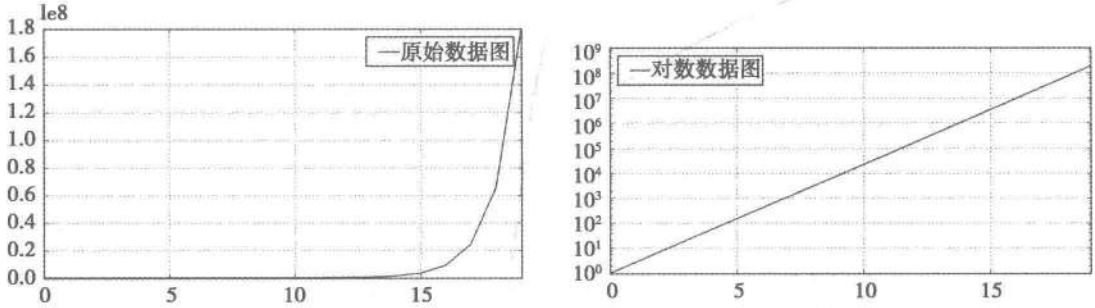


图 3-17 y 轴的对数图形对比图

（6）plot(yerr = error)

- 功能：绘制误差条形图。
- 使用格式：

D.plot(yerr = error)

绘制误差条形图。D 为 Pandas 的 DataFrame 或 Series，代表着均值数据列，而 error 则是误差列，此命令在 y 轴方向画出误差棒图；类似地，如果设置参数 xerr = error，则在 x 轴方向画出误差棒图。

- 实例：绘制误差棒图。绘制结果如图 3-18 所示。

```
import matplotlib.pyplot as plt
```

```
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
import numpy as np
import pandas as pd

error = np.random.randn(10) #定义误差列
y = pd.Series(np.sin(np.arange(10))) #均值数据列
y.plot(yerr = error) #绘制误差图
plt.show()
```

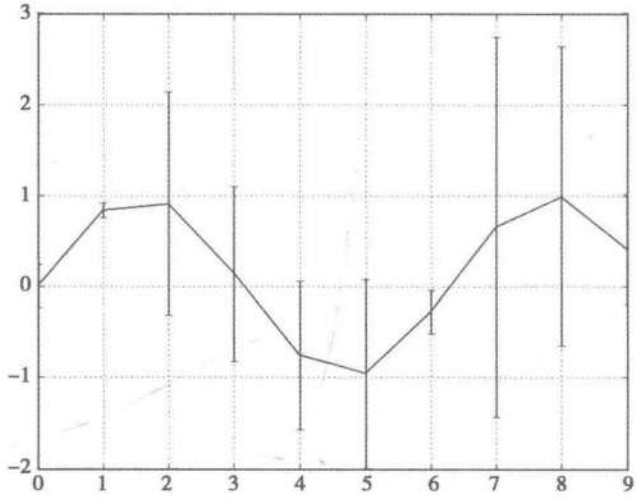
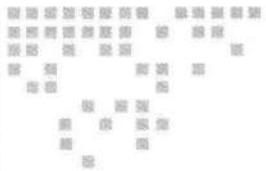


图 3-18 误差条形图

3.4 小结

本章从应用的角度出发，从数据质量分析和数据特征分析两个方面对数据进行探索分析，最后介绍了 Python 常用的数据探索函数及用例。数据质量分析要求我们拿到数据后先检测是否存在缺失值和异常值；数据特征分析要求我们在数据挖掘建模前，通过频率分布分析、对比分析、帕累托分析、周期性分析、相关性分析等方法，对采集的样本数据的特征规律进行分析，以了解数据的规律和趋势，为数据挖掘的后续环节提供支持。

要特别说明的是，在数据可视化中，由于主要使用 Pandas 作为数据探索和分析的工具，因此我们介绍的作图工具都是 Matplotlib 和 Pandas 结合使用。一方面，Matplotlib 是作图工具的基础，Pandas 作图依赖于它；另一方面，Pandas 作图有着简单直接的优势，因此，两者相互结合，往往能够以最高的效率作出符合我们需要的图。



Chapter 4 第 4 章

数据预处理

在数据挖掘中，海量的原始数据中存在着大量不完整（有缺失值）、不一致、有异常的数据，严重影响到数据挖掘建模的执行效率，甚至可能导致挖掘结果的偏差，所以进行数据清洗就显得尤为重要，数据清洗完成后接着进行或者同时进行数据集成、转换、规约等一系列的处理，该过程就是数据预处理。数据预处理一方面是要提高数据的质量，另一方面是要让数据更好地适应特定的挖掘技术或工具。统计发现，在数据挖掘的过程中，数据预处理工作量占到了整个过程的 60%。

数据预处理的主要内容包括数据清洗、数据集成、数据变换和数据规约。处理过程如图 4-1 所示。

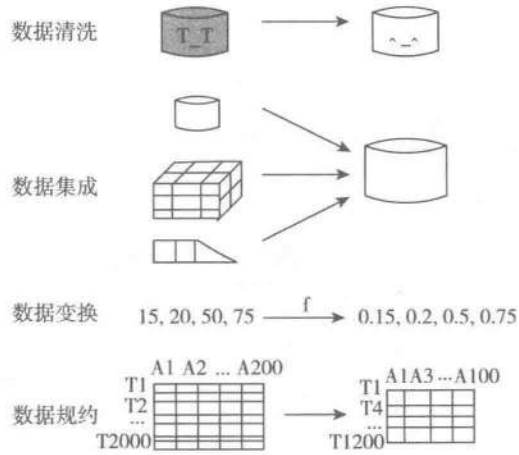


图 4-1 数据预处理过程示意图

4.1 数据清洗

数据清洗主要是删除原始数据集中的无关数据、重复数据，平滑噪声数据，筛选掉与挖掘主题无关的数据，处理缺失值、异常值等。

4.1.1 缺失值处理

处理缺失值的方法可分为 3 类：删除记录、数据插补和不处理。其中常用的数据插补方法见表 4-1。

表4-1　常用的插补方法

插补方法	方法描述
均值 / 中位数 / 众数插补	根据属性值的类型，用该属性取值的平均数 / 中位数 / 众数进行插补
使用固定值	将缺失的属性值用一个常量替换。如广州一个工厂普通外来务工人员的“基本工资”属性的空缺值可以用 2015 年广州市普通外来务工人员工资标准 1895 元 / 月，该方法就是使用固定值
最近临插补	在记录中找到与缺失样本最接近的样本的该属性值插补
回归方法	对带有缺失值的变量，根据已有数据和与其有关的其他变量（因变量）的数据建立拟合模型来预测缺失的属性值
插值法	插值法是利用已知点建立合适的插值函数 $f(x)$，未知值由对应点 x_i 求出的函数值 $f(x_i)$ 近似代替

如果通过简单的删除小部分记录达到既定的目标，那么删除含有缺失值的记录的方法是最有效的。然而，这种方法却有很大的局限性。它是以减少历史数据来换取数据的完备，会造成资源的大量浪费，将丢弃了大量隐藏在这些记录中的信息。尤其在数据集本来就包含很少记录的情况下，删除少量记录可能会严重影响到分析结果的客观性和正确性。一些模型可以将缺失值视作一种特殊的取值，允许直接在含有缺失值的数据上进行建模。

本节重点介绍拉格朗日插值法和牛顿插值法。其他的插值方法还有 Hermite 插值、分段插值、样条插值法等。

（1）拉格朗日插值法

根据数学知识可知，对于平面上已知的 n 个点（无两点在一条直线上）可以找到一个 $n-1$ 次多项式 $y = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}$，使此多项式曲线过这 n 个点。

1）求已知的过 n 个点的 $n-1$ 次多项式：

$$y = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} \tag{4-1}$$

将 n 个点的坐标 $(x_1, y_1), (x_2, y_2) \cdots (x_n, y_n)$ 代入多项式函数，得

$$\begin{aligned} y_1 &= a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_{n-1}x_1^{n-1} \\ y_2 &= a_0 + a_1x_1 + a_2x_2^2 + \cdots + a_{n-1}x_2^{n-1} \\ &\cdots\cdots \\ y_n &= a_0 + a_1x_n + a_2x_n^2 + \cdots + a_{n-1}x_n^{n-1} \end{aligned}$$

解出拉格朗日插值多项式为：

$$\begin{aligned} L(x) = {} & y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} \\ & + y_2 \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)} \\ & + \cdots\cdots \\ & + y_n \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} \end{aligned} \tag{4-2}$$

$$= \sum_{i=0}^{n} y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

2）将缺失的函数值对应的点 x 代入插值多项式得到缺失值的近似值 $L(x)$。

拉格朗日插值公式结构紧凑，在理论分析中很方便，但是当插值节点增减时，插值多项式就会随之变化，这在实际计算中是很不方便的，为了克服这一缺点，提出了牛顿插值法。

（2）牛顿插值法

1）求已知的 n 个点对 $(x_1, y_1), (x_2, y_2)\cdots(x_n, y_n)$ 的所有阶差商公式

$$f[x_1, x] = \frac{f[x] - f[x_1]}{x - x_1} = \frac{f(x) - f(x_1)}{x - x_1} \tag{4-3}$$

$$f[x_2, x_1, x] = \frac{f[x_1, x] - f[x_2, x_1]}{x - x_2} \tag{4-4}$$

$$f[x_3, x_2, x_1, x] = \frac{f[x_2, x_1, x] - f[x_3, x_2, x_1]}{x - x_3} \tag{4-5}$$

$$\cdots\cdots$$

$$f[x_n, x_{n-1}, \cdots, x_1, x] = \frac{f[x_{n-1}, \cdots, x_1, x] - f[x_n, x_{n-1}, \cdots, x_1]}{x - x_n} \tag{4-6}$$

2）联立以上差商公式建立如下插值多项式 $f(x)$

$$\begin{aligned} f(x) &= f(x_1) + (x-x_1)f[x_2, x_1] + (x-x_1)(x-x_2)f[x_3, x_2, x_1] + \\ &\quad (x-x_1)(x-x_2)(x-x_3)f[x_4, x_3, x_2, x_1] + \cdots + \\ &\quad (x-x_1)(x-x_2)\cdots(x-x_{n-1})f[x_n, x_{n-1}, \cdots x_2, x_1] + \\ &\quad (x-x_1)(x-x_2)\cdots(x-x_n)f[x_n, x_{n-1}, \cdots x_1, x] \\ &= P(x) + R(x) \end{aligned} \tag{4-7}$$

其中：

$$\begin{aligned} P(x) &= f(x_1) + (x-x_1)f[x_2, x_1] + (x-x_1)(x-x_2)f[x_3, x_2, x_1] + \\ &\quad (x-x_1)(x-x_2)(x-x_3)f[x_4, x_3, x_2, x_1] + \cdots + \\ &\quad (x-x_1)(x-x_2)\cdots(x-x_{n-1})f[x_n, x_{n-1}, \cdots, x_2, x_1] \end{aligned} \tag{4-8}$$

$$R(x) = (x-x_1)(x-x_2)\cdots(x-x_n)f[x_n, x_{n-1}, \cdots\ x_1, x] \tag{4-9}$$

$P(x)$ 是牛顿插值逼近函数，$R(x)$ 是误差函数。

3）将缺失的函数值对应的点 x 代入插值多项式得到缺失值的近似值 $f(x)$。

牛顿插值法也是多项式插值，但采用了另一种构造插值多项式的方法，与拉格朗日插值相比，具有承袭性和易于变动节点的特点。从本质上来说，两者给出的结果是一样的（相同次数、相同系数的多项式），只不过表示的形式不同。因此，在 Python 的 Scipy 库中，只提供了拉格朗日插值法的函数（因为实现上比较容易），如果需要牛顿插值法，则需要自行编写

函数。

下面结合具体案例介绍拉格朗日插值实现方法。

餐饮系统中的销量数据可能会出现缺失值，如表 4-2 为某餐厅一段时间的销量表，其中 2015 年 2 月 14 日的数据缺失，用拉格朗日插值对缺失值进行插补的 Python 程序实现如代码清单 4-1 所示。

表4-2　某餐厅一段时间的销量数据

时　间	2015/2/25	2015/2/24	2015/2/23	2015/2/22	2015/2/21	2015/2/20
销售额（元）	3442.1	3393.1	3136.6	3744.1	6607.4	4060.3
时　间	2015/2/19	2015/2/18	2015/2/16	2015/2/15	2015/2/14	2015/2/13
销售额（元）	3614.7	3295.5	2332.1	2699.3	空值	3036.8

数据详见：demo/data/catering_sale.xls

代码清单4-1　用拉格朗日法进行插补

```
#拉格朗日插值代码
import pandas as pd #导入数据分析库Pandas
from scipy.interpolate import lagrange #导入拉格朗日插值函数

inputfile = '../data/catering_sale.xls' #销量数据路径
outputfile = '../tmp/sales.xls' #输出数据路径

data = pd.read_excel(inputfile) #读入数据
data[u'销量'][(data[u'销量'] < 400) | (data[u'销量'] > 5000)] = None #过滤异常值，将
    其变为空值

#自定义列向量插值函数
#s为列向量，n为被插值的位置，k为取前后的数据个数，默认为5
def ployinterp_column(s, n, k=5):
  y = s[list(range(n-k, n)) + list(range(n+1, n+1+k))] #取数
  y = y[y.notnull()] #剔除空值
  return lagrange(y.index, list(y))(n) #插值并返回插值结果

#逐个元素判断是否需要插值
for i in data.columns:
  for j in range(len(data)):
    if (data[i].isnull())[j]: #如果为空即插值。
      data[i][j] = ployinterp_column(data[i], j)

data.to_excel(outputfile) #输出结果，写入文件
```

代码详见：demo/code/lagrange_newton_interp.m

应用拉格朗日插值法算对表 4-2 中的缺失值进行插补，使用缺失值前后各 5 个未缺失的数据参与建模，得插值结果如下所示。

时　间	原 始 值	插　值
2015/2/21	6607.4	4275.255
2015/2/14	空值	4156.86

在进行插值之前会对数据进行异常值检测，发现2015/2/21日的数据是异常的（数据大于5000），所以也把此日期数据定义为空缺值，进行补数。利用拉格朗日插值对这2015/2/21和2015/2/14的数据进行插补，结果是4275.255和4156.86，这两天都是周末，而周末的销售额一般要比周一到周五要多，所以插值结果比较符合实际情况。

4.1.2 异常值处理

在数据预处理时，异常值是否剔除，需视具体情况而定，因为有些异常值可能蕴含着有用的信息。异常值处理常用方法见表4-3。

表4-3 异常值处理常用方法

异常值处理方法	方 法 描 述
删除含有异常值的记录	直接将含有异常值的记录删除
视为缺失值	将异常值视为缺失值，利用缺失值处理的方法进行处理
平均值修正	可用前后两个观测值的平均值修正该异常值
不处理	直接在具有异常值的数据集上进行挖掘建模

将含有异常值的记录直接删除的方法简单易行，但缺点也很明显，在观测值很少的情况下，这种删除会造成样本量不足，可能会改变变量的原有分布，从而造成分析结果的不准确。视为缺失值处理的好处是可以利用现有变量的信息，对异常值（缺失值）进行填补。

在很多情况下，要先分析异常值出现的可能原因，再判断异常值是否应该舍弃，如果是正确的数据，可以直接在具有异常值的数据集上进行挖掘建模。

4.2 数据集成

数据挖掘需要的数据往往分布在不同的数据源中，数据集成就是将多个数据源合并存放在一个一致的数据存储（如数据仓库）中的过程。

在数据集成时，来自多个数据源的现实世界实体的表达形式是不一样的，有可能不匹配，要考虑实体识别问题和属性冗余问题，从而将源数据在最低层上加以转换、提炼和集成。

4.2.1 实体识别

实体识别是指从不同数据源识别出现实世界的实体，它的任务是统一不同源数据的矛盾之处，常见形式如下。

（1）同名异义

数据源 A 中的属性 ID 和数据源 B 中的属性 ID 分别描述的是菜品编号和订单编号，即描述的是不同的实体。

（2）异名同义

数据源 A 中的 sales_dt 和数据源 B 中的 sales_date 都是描述销售日期的，即 A. sales_dt= B. sales_date。

（3）单位不统一

描述同一个实体分别用的是国际单位和中国传统的计量单位。

检测和解决这些冲突就是实体识别的任务。

4.2.2　冗余属性识别

数据集成往往导致数据冗余，例如，

1）同一属性多次出现；

2）同一属性命名不一致导致重复。

仔细整合不同源数据能减少甚至避免数据冗余与不一致，从而提高数据挖掘的速度和质量。对于冗余属性要先分析，检测到后再将其删除。

有些冗余属性可以用相关分析检测。给定两个数值型的属性 A 和 B，根据其属性值，用相关系数度量一个属性在多大程度上蕴含另一个属性，相关系数介绍见 3.2.6 节。

4.3　数据变换

数据变换主要是对数据进行规范化处理，将数据转换成“适当的”形式，以适用于挖掘任务及算法的需要。

4.3.1　简单函数变换

简单函数变换是对原始数据进行某些数学函数变换，常用的变换包括平方、开方、取对数、差分运算等，即：

$$x' = x^2 \tag{4-10}$$

$$x' = \sqrt{x} \tag{4-11}$$

$$x' = \log(x) \tag{4-12}$$

$$\nabla f(x_k) = f(x_{k+1}) - f(x_k) \tag{4-13}$$

简单的函数变换常用来将不具有正态分布的数据变换成具有正态分布的数据。在时间序列分析中，有时简单的对数变换或者差分运算就可以将非平稳序列转换成平稳序列。在数据挖掘中，简单的函数变换可能更有必要，比如个人年收入的取值范围为 10 000 元到 10 亿元，

这是一个很大的区间，使用对数变换对其进行压缩是常用的一种变换处理方法。

4.3.2 规范化

数据规范化（归一化）处理是数据挖掘的一项基础工作。不同评价指标往往具有不同的量纲，数值间的差别可能很大，不进行处理可能会影响到数据分析的结果。为了消除指标之间的量纲和取值范围差异的影响，需要进行标准化处理，将数据按照比例进行缩放，使之落入一个特定的区域，便于进行综合分析。如将工资收入属性值映射到 [-1,1] 或者 [0,1] 内。

数据规范化对于基于距离的挖掘算法尤为重要。

（1）最小 - 最大规范化

最小 - 最大规范化也称为离差标准化，是对原始数据的线性变换，将数值值映射到 [0,1] 之间。

转换公式如下：

$$x^* = \frac{x - min}{max - min} \tag{4-14}$$

其中，*max* 为样本数据的最大值，*min* 为样本数据的最小值。*max-min* 为极差。离差标准化保留了原来数据中存在的关系，是消除量纲和数据取值范围影响的最简单方法。这种处理方法的缺点是若数值集中且某个数值很大，则规范化后各值会接近于 0，并且将会相差不大。若将来遇到超过目前属性 [*min*, *max*] 取值范围的时候，会引起系统出错，需要重新确定 *min* 和 *max*。

（2）零 - 均值规范化

零 - 均值规范化也称标准差标准化，经过处理的数据的均值为 0，标准差为 1。转化公式为：

$$x^* = \frac{x - \bar{x}}{\sigma} \tag{4-15}$$

其中 $\bar{x}$ 为原始数据的均值，σ 为原始数据的标准差，是当前用得最多的数据标准化方法。

（3）小数定标规范化

通过移动属性值的小数位数，将属性值映射到 [-1,1] 之间，移动的小数位数取决于属性值绝对值的最大值。

转化公式为：

$$x^* = \frac{x}{10^k} \tag{4-16}$$

下面通过对一个矩阵使用上面 3 种规范化的方法处理，对比结果。其程序如代码清单 4-2 所示。

代码清单4-2 数据规范化代码

```
#-*- coding: utf-8 -*-
#数据规范化
import pandas as pd

datafile = '../data/normalization_data.xls' #参数初始化
```

```
data = pd.read_excel(datafile, header = None) #读取数据

(data - data.min())/(data.max() - data.min()) #最小-最大规范化
(data - data.mean())/data.std() #零-均值规范化
data/10**np.ceil(np.log10(data.abs().max())) #小数定标规范化
```

代码详见：demo/code/data_normalization.py

执行上面的代码后，可以在命令行看到下面的输出：

```
>>> data
     0    1    2     3
0   78  521  602  2863
1  144 -600 -521  2245
2   95 -457  468 -1283
3   69  596  695  1054
4  190  527  691  2051
5  101  403  470  2487
6  146  413  435  2571
>>> (data - data.min())/(data.max() - data.min()) #最小-最大规范化
          0         1         2         3
0  0.074380  0.937291  0.923520  1.000000
1  0.619835  0.000000  0.000000  0.850941
2  0.214876  0.119565  0.813322  0.000000
3  0.000000  1.000000  1.000000  0.563676
4  1.000000  0.942308  0.996711  0.804149
5  0.264463  0.838629  0.814967  0.909310
6  0.636364  0.846990  0.786184  0.929571
>>> (data - data.mean())/data.std() #零-均值规范化
          0         1         2         3
0 -0.905383  0.635863  0.464531  0.798149
1  0.604678 -1.587675 -2.193167  0.369390
2 -0.516428 -1.304030  0.147406 -2.078279
3 -1.111301  0.784628  0.684625 -0.456906
4  1.657146  0.647765  0.675159  0.234796
5 -0.379150  0.401807  0.152139  0.537286
6  0.650438  0.421642  0.069308  0.595564
>>> data/10**np.ceil(np.log10(data.abs().max())) #小数定标规范化
       0      1      2       3
0  0.078  0.521  0.602  0.2863
1  0.144 -0.600 -0.521  0.2245
2  0.095 -0.457  0.468 -0.1283
3  0.069  0.596  0.695  0.1054
4  0.190  0.527  0.691  0.2051
5  0.101  0.403  0.470  0.2487
6  0.146  0.413  0.435  0.2571
```

对于一个含有 n 个记录 p 个属性的数据集，分别对每一个属性的取值进行规范化。对原始的数据矩阵分别用最小－最大规范化、零－均值规范化、小数定标规范化进行规范化后的数据如上所示。

4.3.3 连续属性离散化

一些数据挖掘算法，特别是某些分类算法（如 ID3 算法、Apriori 算法等），要求数据是分类属性形式。这样，常常需要将连续属性变换成分类属性，即连续属性离散化。

1. 离散化的过程

连续属性的离散化就是在数据的取值范围内设定若干个离散的划分点，将取值范围划分为一些离散化的区间，最后用不同的符号或整数值代表落在每个子区间中的数据值。所以，离散化涉及两个子任务：确定分类数以及如何将连续属性值映射到这些分类值。

2. 常用的离散化方法

常用的离散化方法有等宽法、等频法和（一维）聚类。

（1）等宽法

将属性的值域分成具有相同宽度的区间，区间的个数由数据本身的特点决定，或者由用户指定，类似于制作频率分布表。

（2）等频法

将相同数量的记录放进每个区间。

这两种方法简单，易于操作，但都需要人为地规定划分区间的个数。同时，等宽法的缺点在于它对离群点比较敏感，倾向于不均匀地把属性值分布到各个区间。有些区间包含许多数据，而另外一些区间的数据极少，这样会严重损坏建立的决策模型。等频法虽然避免了上述问题的产生，却可能将相同的数据值分到不同的区间以满足每个区间中固定的数据个数。

（3）基于聚类分析的方法

一维聚类的方法包括两个步骤，首先将连续属性的值用聚类算法（如 K-Means 算法）进行聚类，然后再将聚类得到的簇进行处理，合并到一个簇的连续属性值并做同一标记。聚类分析的离散化方法也需要用户指定簇的个数，从而决定产生的区间数。

下面使用上述 3 种离散化方法对“医学中中医证型的相关数据”进行连续属性离散化的对比，该属性的示例数据见表 4-4。

表4-4 中医证型连续属性离散化数据

肝气郁结证型系数	0.056	0.488	0.107	0.322	0.242	0.389

数据详见：demo/data/discretization_data.xls

具体可以参考第 8 章中相关章节，其 Python 代码如代码清单 4-3 所示。

代码清单4-3 数据离散化

```
#-*- coding: utf-8 -*-
#数据规范化
import pandas as pd

datafile = '../data/discretization_data.xls' #参数初始化
```

```
data = pd.read_excel(datafile) #读取数据
data = data[u'肝气郁结证型系数'].copy()
k = 4

d1 = pd.cut(data, k, labels = range(k)) #等宽离散化，各个类比依次命名为0,1,2,3

#等频率离散化
w = [1.0*i/k for i in range(k+1)]
w = data.describe(percentiles = w)[4:4+k+1] #使用describe函数自动计算分位数
w[0] = w[0]*(1-1e-10)
d2 = pd.cut(data, w, labels = range(k))

from sklearn.cluster import KMeans #引入KMeans
kmodel = KMeans(n_clusters = k, n_jobs = 4) #建立模型，n_jobs是并行数，一般等于CPU数较好
kmodel.fit(data.reshape((len(data), 1))) #训练模型
c = pd.DataFrame(kmodel.cluster_centers_).sort(0) #输出聚类中心，并且排序（默认是随机
    序的）
w = pd.rolling_mean(c, 2).iloc[1:] #相邻两项求中点，作为边界点
w = [0] + list(w[0]) + [data.max()] #把首末边界点加上
d3 = pd.cut(data, w, labels = range(k))

def cluster_plot(d, k): #自定义作图函数来显示聚类结果
  import matplotlib.pyplot as plt
  plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
  plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

  plt.figure(figsize = (8, 3))
  for j in range(0, k):
    plt.plot(data[d==j], [j for i in d[d==j]], 'o')

  plt.ylim(-0.5, k-0.5)
  return plt

cluster_plot(d1, k).show()
cluster_plot(d2, k).show()
cluster_plot(d3, k).show()
```

代码详见：demo/code/data_discretization.m

运行上面的程序，可以得到图 4-2 ～图 4-4 所示的结果。

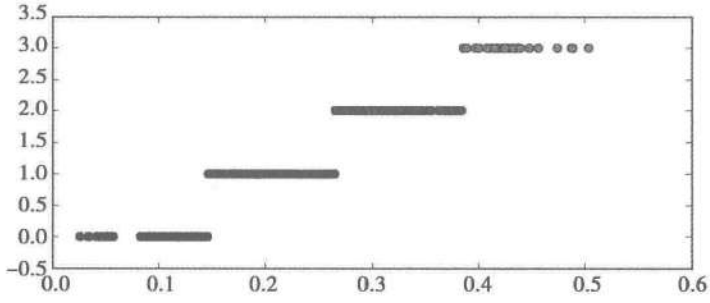


图 4-2　等宽离散化结果

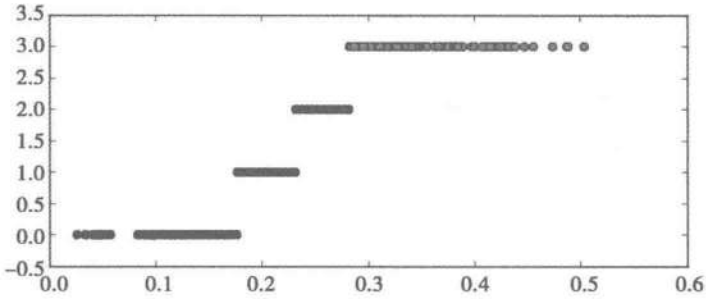


图 4-3 等频离散化方法

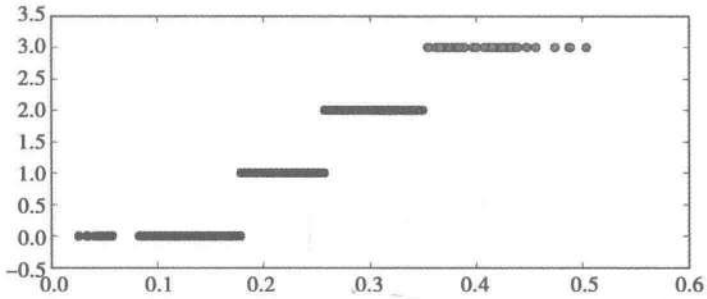


图 4-4 （一维）聚类离散化方法

分别用等宽法、等频法和（一维）聚类对数据进行离散化，将数据分成 4 类，然后将每一类记为同一个标识，如分别记为 A1、A2、A3、A4，再进行建模。

4.3.4 属性构造

在数据挖掘的过程中，为了提取更有用的信息，挖掘更深层次的模式，提高挖掘结果的精度，我们需要利用已有的属性集构造出新的属性，并加入到现有的属性集合中。

比如，进行防窃漏电诊断建模时，已有的属性包括供入电量、供出电量（线路上各大用户用电量之和）。理论上供入电量和供出电量应该是相等的，但是由于在传输过程中存在电能损耗，使得供入电量略大于供出电量，如果该条线路上的一个或多个大用户存在窃漏电行为，会使得供入电量明显大于供出电量（详见线户关系图 6-7）。反过来，为了判断是否有大用户存在窃漏电行为，可以构造出一个新的指标—线损率，该过程就是构造属性。新构造的属性线损率按如下公式计算。

$$\text{线损率} = \frac{\text{供入电量} - \text{供出电量}}{\text{供入电量}} \times 100\% \tag{4-17}$$

线损率的正常范围一般在 3% ～ 15%，如果远远超过该范围，就可以认为该条线路的大用户很可能存在窃漏电等用电异常行为。

根据线损率的计算公式，由供入电量、供出电量进行线损率的属性构造代码，如代码清

单 4-4 所示。

代码清单4-4　线损率属性构造

```
#-*- coding: utf-8 -*-
#线损率属性构造
import pandas as pd

#参数初始化
inputfile= '../data/electricity_data.xls' #供入供出电量数据
outputfile = '../tmp/electricity_data.xls' #属性构造后数据文件

data = pd.read_excel(inputfile) #读入数据
data[u'线损率'] = (data[u'供入电量'] - data[u'供出电量'])/data[u'供入电量']

data.to_excel(outputfile, index = False) #保存结果
```

代码详见：demo/code/line_rate_construct.py

4.3.5　小波变换

小波变换[4][5]是一种新型的数据分析工具，是近年来兴起的信号分析手段。小波分析的理论和方法在信号处理、图像处理、语音处理、模式识别、量子物理等领域得到越来越广泛的应用，它被认为是近年来在工具及方法上的重大突破。小波变换具有多分辨率的特点，在时域和频域都具有表征信号局部特征的能力，通过伸缩和平移等运算过程对信号进行多尺度聚焦分析，提供了一种非平稳信号的时频分析手段，可以由粗及细地逐步观察信号，从中提取有用信息。

能够刻画某个问题的特征量往往是隐含在一个信号中的某个或者某些分量中，小波变换可以把非平稳信号分解为表达不同层次、不同频带信息的数据序列，即小波系数。选取适当的小波系数，即完成了信号的特征提取。下面将介绍基于小波变换的信号特征提取方法。

（1）基于小波变换的特征提取方法

基于小波变换的特征提取方法主要有：基于小波变换的多尺度空间能量分布特征提取、基于小波变换的多尺度空间的模极大值特征提取、基于小波包变换的特征提取、基于适应性小波神经网络的特征提取，详见表 4-5。

表4-5　基于小波变换的特征提取方法

基于小波变换的特征提取方法	方 法 描 述
基于小波变换的多尺度空间能量分布特征提取方法	各尺度空间内的平滑信号和细节信号能提供原始信号的时频局域信息，特别是能提供不同频段上信号的构成信息。把不同分解尺度上信号的能量求解出来，就可以将这些能量尺度顺序排列，形成特征向量供识别用
基于小波变换的多尺度空间的模极大值特征提取方法	利用小波变换的信号局域化分析能力，求解小波变换的模极大值特性来检测信号的局部奇异性，将小波变换模极大值的尺度参数 s、平移参数 t 及其幅值作为目标的特征量

（续）

基于小波变换的特征提取方法	方 法 描 述
基于小波包变换的特征提取方法	利用小波分解，可将时域随机信号序列映射为尺度域各子空间内的随机系数序列，按小波包分解得到的最佳子空间内随机系数序列的不确定性程度最低，将最佳子空间的熵值及最佳子空间在完整二叉树中的位置参数作为特征量，可以用于目标识别
基于适应性小波神经网络的特征提取方法	基于适应性小波神经网络的特征提取方法可以把信号通过分析小波拟合表示，进行特征提取

（2）小波基函数

小波基函数是一种具有局部支集的函数，并且平均值为 0，小波基函数满足 $\psi(0)=\int\psi(t)dt=0$。常用的小波基有 Haar 小波基、db 系列小波基等。Haar 小波基函数如图 4-5 所示。

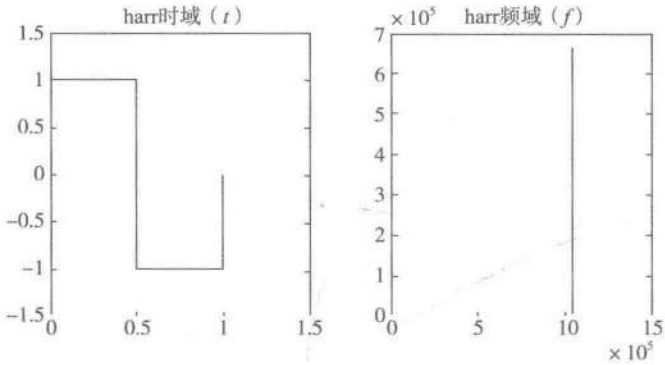


图 4-5 Haar 小波基函数

（3）小波变换

对小波基函数进行伸缩和平移变换：

$$\psi_{a,b}(t)=\frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \tag{4-18}$$

其中，a 为伸缩因子，b 为平移因子。

任意函数 $f(t)$ 的连续小波变换（CWT）为：

$$W_f(a,b)=|a|^{-1/2}\int f(t)\psi\left(\frac{t-b}{a}\right)dt \tag{4-19}$$

可知，连续小波变换为 $f(t)\rightarrow W_f(a,b)$ 的映射，对小波基函数 $\psi(t)$ 增加约束条件 $C_\psi=\int\frac{|\hat{\psi}(t)|^2}{t}dt<\infty$，就可以由 $W_f(a,b)$ 逆变换得到 $f(t)$。其中 $\hat{\psi}(t)$ 为 $\psi(t)$ 的傅里叶变换。

其逆变换为：

$$f(t)=\frac{1}{C_\psi}\iint\frac{1}{a^2}W_f(a,b)\psi\left(\frac{t-b}{a}\right)da\cdot db \tag{4-20}$$

下面介绍基于小波变换的多尺度空间能量分布特征提取方法。

（4）基于小波变换的多尺度空间能量分布特征提取方法

应用小波分析技术可以把信号在各频率波段中的特征提取出来，基于小波变换的多尺度空间能量分布特征提取方法是对信号进行频带分析，再分别以计算所得的各个频带的能量作为特征向量。

信号 $f(t)$ 的二进小波分解可表示为：

$$f(t) = A^j + \Sigma D^j \tag{4-21}$$

其中 A 是近似信号，为低频部分；D 是细节信号，为高频部分，此时信号的频带分布如图 4-6 所示。

信号的总能量为：

$$E = EA_j + \Sigma ED_j \tag{4-22}$$

选择第 j 层的近似信号和各层的细节信号的能量作为特征，构造特征向量：

$$F = [EA_j, ED_1, ED_2, \cdots, ED_j] \tag{4-23}$$

利用小波变换可以对声波信号进行特征提取，提取出可以代表声波信号的向量数据，即完成从声波信号到特征向量数据的变换。本例利用小波函数对声波信号数据进行分解，得到 5 个层次的小波系数。利用这些小波系数求得各个能量值，这些能量值即可作为声波信号的特征数据。

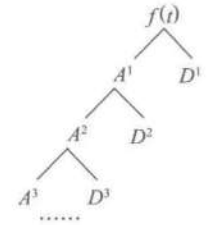


图 4-6　多尺度分解的信号频带分布

在 Python 中，Scipy 本身提供了一些信号处理函数，但不够全面，而更好的信号处理库是 PyWavelets（pywt）。PyWavelets 完成上述任务，程序实现如代码清单 4-5 所示。

代码清单4-5　小波变换特征提取代码

```
#-*- coding: utf-8 -*-
#利用小波分析进行特征分析

#参数初始化
inputfile= '../data/leleccum.mat' #提取自Matlab的信号文件

from scipy.io import loadmat #mat是Python专用格式，需要用loadmat读取它
mat = loadmat(inputfile)
signal = mat['leleccum'][0]

import pywt #导入PyWavelets
coeffs = pywt.wavedec(signal, 'bior3.7', level = 5)
#返回结果为level+1个数字，第一个数组为逼近系数数组，后面的依次是细节系数数组
```

代码详见：demo/code/wave_analyze.py

4.4 数据规约

在大数据集上进行复杂的数据分析和挖掘需要很长的时间，数据规约产生更小但保持原数据完整性的新数据集。在规约后的数据集上进行分析和挖掘将更有效率。

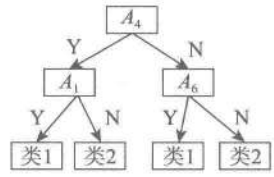
数据规约的意义在于：

- 降低无效、错误数据对建模的影响，提高建模的准确性；
- 少量且具代表性的数据将大幅缩减数据挖掘所需的时间；
- 降低储存数据的成本。

4.4.1 属性规约

属性规约通过属性合并来创建新属性维数，或者直接通过删除不相关的属性（维）来减少数据维数，从而提高数据挖掘的效率、降低计算成本。属性规约的目标是寻找出最小的属性子集并确保新数据子集的概率分布尽可能地接近原来数据集的概率分布。属性规约常用方法见表 4-6。

表4-6 属性规约常用方法

属性规约方法	方法描述	方法解析
合并属性	将一些旧属性合为新属性	初始属性集：$\{A_1, A_2, A_3, A_4, B_1, B_2, B_3, C\}$ $\{A_1, A_2, A_3, A_4\} \rightarrow$ A $\{B_1, B_2, B_3\} \rightarrow$ B ⇒规约后属性集：$\{A, B, C\}$
逐步向前选择	从一个空属性集开始，每次从原来属性集合中选择一个当前最优的属性添加到当前属性子集中。直到无法选择出最优属性或满足一定阈值约束为止	初始属性集：$\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\{\} \Rightarrow \{A_1\} \Rightarrow \{A_1, A_4\}$ ⇒规约后属性集：$\{A_1, A_4, A_6\}$
逐步向后删除	从一个全属性集开始，每次从当前属性子集中选择一个当前最差的属性并将其从当前属性子集中消去。直到无法选择出最差属性为止或满足一定阈值约束为止	初始属性集：$\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\} \Rightarrow \{A_1, A_4, A_5, A_6\}$ ⇒规约后属性集：$\{A_1, A_4, A_6\}$
决策树归纳	利用决策树的归纳方法对初始数据进行分类归纳学习，获得一个初始决策树，所有没有出现在这个决策树上的属性均可认为是无关属性，因此将这些属性从初始集合中删除，就可以获得一个较优的属性子集	初始属性集：$\{A_1, A_2, A_3, A_4, A_5, A_6\}$  ⇒规约后属性集：$\{A_1, A_4, A_6\}$
主成分分析	用较少的变量去解释原始数据中的大部分变量，即将许多相关性很高的变量转化成彼此相互独立或不相关的变量	详见下面计算步骤

逐步向前选择、逐步向后删除和决策树归纳是属于直接删除不相关属性（维）方法。主成分分析是一种用于连续属性的数据降维方法，它构造了原始数据的一个正交变换，新空间的基底去除了原始空间基底下数据的相关性，只需使用少数新变量就能够解释原始数据中的大部分变异。在应用中，通常是选出比原始变量个数少，能解释大部分数据中的变量的几个新变量，即所谓主成分，来代替原始变量进行建模。

主成分分析 [6] 的计算步骤如下。

1）设原始变量 $X_1, X_2, \cdots, X_p$ 的 n 次观测数据矩阵为：

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}=(X_1, X_2, \cdots, X_P) \tag{4-24}$$

2）将数据矩阵按列进行中心标准化。为了方便，将标准化后的数据矩阵仍然记为 X。

3）求相关系数矩阵 R, $R=(r_{ij})_{p\times p}$，r_{ij} 的定义为：

$$r_{ij}=\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)/\sqrt{\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)^2\sum_{k=1}^{n}(x_{kj}-\bar{x}_j)^2} \tag{4-25}$$

其中，$r_{ij}=r_{ji}$, $r_{ii}=1$。

4）求 R 的特征方程 $\det(R-\lambda E)=0$ 的特征根 $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_p > 0$。

5）确定主成分个数 m：$\dfrac{\sum_{i=1}^{m}\lambda_i}{\sum_{i=1}^{p}\lambda_i} \geqslant \alpha$，$\alpha$ 根据实际问题确定，一般取 80%。

6）计算 m 个相应的单位特征向量：

$$\beta_1=\begin{bmatrix}\beta_{11}\\ \beta_{21}\\ \vdots \\ \beta_{p1}\end{bmatrix}, \beta_2=\begin{bmatrix}\beta_{12}\\ \beta_{22}\\ \vdots \\ \beta_{p2}\end{bmatrix}, \cdots, \beta_m=\begin{bmatrix}\beta_{m2}\\ \beta_{m2}\\ \vdots \\ \beta_{m2}\end{bmatrix} \tag{4-26}$$

7）计算主成分：

$$Z_i=\beta_{1i}X_1+\beta_{2i}X_2+\cdots+\beta_{pi}X_p,\ i=1, 2, \cdots, m \tag{4-27}$$

在 Python 中，主成分分析的函数位于 Scikit-Learn 下：

```
sklearn.decomposition.PCA(n_components = None, copy = True, whiten = False)
```

参数说明：

（1）n_components

意义：PCA 算法中所要保留的主成分个数 n，也即保留下来的特征个数 n。

类型：int 或者 string，缺省时默认为 None，所有成分被保留。赋值为 int，比如 n_components=1，将把原始数据降到一个维度。赋值为 string，比如 n_components='mle'，将自动选取特征个数 n，使得满足所要求的方差百分比。

（2）copy

类型：bool，True 或者 False，缺省时默认为 True。

意义：表示是否在运行算法时，将原始训练数据复制一份。若为 True，则运行 PCA 算法后，原始训练数据的值不会有任何改变，因为是在原始数据的副本上进行运算；若为 False，则运行 PCA 算法后，原始训练数据的值会改，因为是在原始数据上进行降维计算。

（3）whiten

类型：bool，缺省时默认为 False。

意义：白化，使得每个特征具有相同的方差。

使用主成分分析降维的程序如代码清单 4-6 所示。

代码清单4-6 主成分分析降维代码

```
import pandas as pd

#参数初始化
inputfile = '../data/principal_component.xls'
outputfile = '../tmp/dimention_reducted.xls' #降维后的数据

data = pd.read_excel(inputfile, header = None) #读入数据

from sklearn.decomposition import PCA

pca = PCA()
pca.fit(data)
pca.components_ #返回模型的各个特征向量
pca.explained_variance_ratio_ #返回各个成分各自的方差百分比
```

代码详见：demo/code/principal_component_analyze.py

运行上面的代码可以，得到下面的结果。

```
>>> pca.components_ #返回模型的各个特征向量
array([[-0.56788461, -0.2280431 , -0.23281436, -0.22427336, -0.3358618 ,
        -0.43679539, -0.03861081, -0.46466998],
       [-0.64801531, -0.24732373,  0.17085432,  0.2089819 ,  0.36050922,
         0.55908747, -0.00186891, -0.05910423],
       [-0.45139763,  0.23802089, -0.17685792, -0.11843804, -0.05173347,
        -0.20091919, -0.00124421,  0.80699041],
       [-0.19404741,  0.9021939 , -0.00730164, -0.01424541,  0.03106289,
         0.12563004,  0.11152105, -0.3448924 ],
       [ 0.06133747,  0.03383817, -0.12652433, -0.64325682,  0.3896425 ,
         0.10681901, -0.63233277, -0.04720838],
       [-0.02579655,  0.06678747, -0.12816343,  0.57023937,  0.52642373,
        -0.52280144, -0.31167833, -0.0754221 ],
       [ 0.03800378, -0.09520111, -0.15593386, -0.34300352,  0.56640021,
        -0.18985251,  0.69902952, -0.04505823],
       [ 0.10147399, -0.03937889, -0.91023327,  0.18760016, -0.06193777,
         0.34598258,  0.02090066, -0.02137393]])
```

```
>>> pca.explained_variance_ratio_ #返回各个成分各自的方差百分比（贡献率）
array([  7.74011263e-01,   1.56949443e-01,   4.27594216e-02,
         2.40659228e-02,   1.50278048e-03,   4.10990447e-04,
         2.07718405e-04,   9.24594471e-05])
```

从上面的结果可以得到特征方程 $\det(R-\lambda E)=0$ 有 7 个特征根、对应的 7 个单位特征向量以及各个成分各自的方差百分比（也称为贡献率）。其中，方差百分比越大，说明向量的权重越大。

当选取前 4 个主成分时，累计贡献率已达到 97.37%，说明选取前 3 个主成分进行计算已经相当不错了，因此可以重新建立 PCA 模型，设置 n_components = 3，计算出成分结果。

```
#接代码清单4-5
pca = PCA(3)
pca.fit(data)
low_d = pca.transform(data) #用它来降低维度
pd.DataFrame(low_d).to_excel(outputfile) #保存结果
pca.inverse_transform(low_d) #必要时可以用inverse_transform()函数来复原数据
```

降维结果如下所示。

```
>>> low_d
array([[ -8.19133694, -16.90402785,   3.90991029],
       [ -0.28527403,   6.48074989,  -4.62870368],
       [ 23.70739074,   2.85245701,  -0.4965231 ],
       [ 14.43202637,  -2.29917325,  -1.50272151],
       [ -5.4304568 , -10.00704077,   9.52086923],
       [-24.15955898,   9.36428589,   0.72657857],
       [  3.66134607,   7.60198615,  -2.36439873],
       [-13.96761214, -13.89123979,  -6.44917778],
       [-40.88093588,  13.25685287,   4.16539368],
       [  1.74887665,   4.23112299,  -0.58980995],
       [ 21.94321959,   2.36645883,   1.33203832],
       [ 36.70868069,   6.00536554,   3.97183515],
       [ -3.28750663,  -4.86380886,   1.00424688],
       [ -5.99885871,  -4.19398863,  -8.59953736]])
```

原始数据从 8 维被降维到了 3 维，关系式由公式（4-27）确定，同时这 3 维数据占了原始数据 95% 以上的信息。

4.4.2　数值规约

数值规约指通过选择替代的、较小的数据来减少数据量，包括有参数方法和无参数方法两类。有参数方法是使用一个模型来评估数据，只需存放参数，而不需要存放实际数据，例如回归（线性回归和多元回归）和对数线性模型（近似离散属性集中的多维概率分布）。无参数方法就需要存放实际数据，例如直方图、聚类、抽样（采样）。

（1）直方图

直方图使用分箱来近似数据分布，是一种流行的数据规约形式。属性 A 的直方图将 A 的数据分布划分为不相交的子集或桶。如果每个桶只代表单个属性值 / 频率对，则该桶称为单桶。通常，桶表示给定属性的一个连续区间。

这里结合实际案例来说明如何使用直方图做数值规约。图 4-7 所示的数据是某餐饮企业菜品的单价表（按人民币取整）从小到大排序。

3，3，5，5，5，8，8，10，10，10，10，15，15，15，22，22，22，22，22，22，22，22，25，25，25，25，25，25，25，25，25，30，30，30，30，30，35，35，35，35，35，39，39，40，40，40。

图 4-7 使用单桶显示了这些数据的直方图。为进一步压缩数据，通常让每个桶代表给定属性的一个连续值域。在图 4-8 中每个桶代表长度为 13 元（人民币）的价值区间。

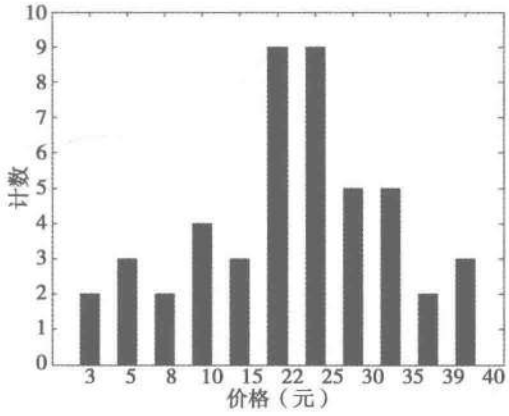


图 4-7 使用单桶的价格直方图——每个单桶代表一个价值 / 频率对

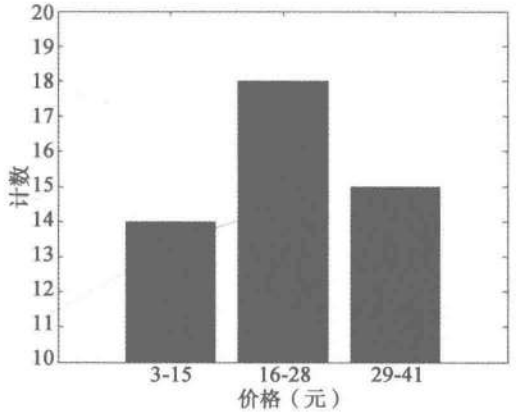


图 4-8 价格的等宽直方图——每个桶代表一个价格区间 / 频率对

（2）聚类

聚类技术将数据元组（即记录，数据表中的一行）视为对象。它将对象划分为簇，使一个簇中的对象相互“相似”，而与其他簇中的对象“相异”。在数据规约中，用数据的簇替换实际数据。该技术的有效性依赖于簇的定义是否符合数据的分布性质。

（3）抽样

抽样也是一种数据规约技术，它用比原始数据小得多的随机样本（子集）表示原始数据集。假定原始数据集 D 包含 N 个元组，可以采用抽样方法对 D 进行抽样。下面介绍常用的抽样方法。

s 个样本无放回简单随机抽样：从 D 的 N 个元组中抽取 s 个样本（$s < N$），其中 D 中任意元组被抽取的概率均为 $1/N$，即所有元组的抽取是等可能的。

s 个样本有放回简单随机抽样：该方法类似于无放回简单随机抽样，不同在于每次一个

元组从 D 中抽取后，记录它，然后放回原处。

聚类抽样：如果 D 中的元组分组放入 M 个互不相交的“簇”，则可以得到 s 个簇的简单随机抽样，其中 $s < M$。例如，数据库中元组通常一次检索一页，这样每页就可以视为一个簇。

分层抽样：如果 D 划分成互不相交的部分，称作层，则通过对每一层的简单随机抽样就可以得到 D 的分层样本。例如，可以得到关于顾客数据的一个分层样本，按照顾客的每个年龄组创建分层。

用于数据规约时，抽样最常用来估计聚集查询的结果。在指定的误差范围内，可以确定（使用中心极限定理）估计一个给定的函数所需的样本大小。通常样本的大小 s 相对于 N 非常小。而通过简单地增加样本大小，这样的集合可以进一步求精。

（4）参数回归

简单线性模型和对数线性模型可以用来近似描述给定的数据。（简单）线性模型对数据建模，使之拟合一条直线。以下介绍一个简单线性模型的例子，对对数线性模型只进行简单介绍。

把点对（2,5），(3,7），(4,9），(5,12），(6,11），(7,15），(8,18），(9,19），(11,22），(12,25），(13,24），(15,30），(17,35）规约成线性函数 $y = wx + b$。即拟合函数 $y = 2x + 1.3$ 线上对应的点可以近似看作已知点。如图 4-9 所示。

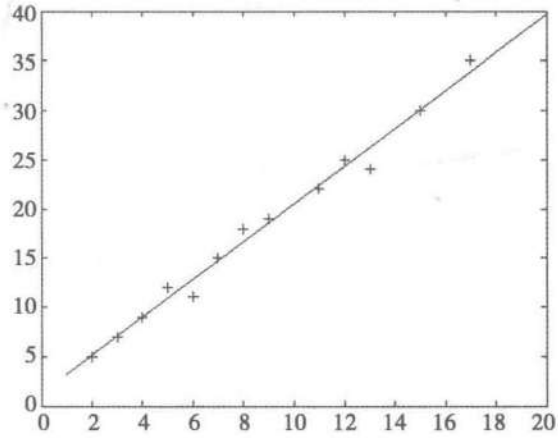


图 4-9　将已知点规约成线性函数 $y = wx + b$

其中，y 的方差是常量 13.44。在数据挖掘中，x 和 y 是数值属性。系数 2 和 1.3（称作回归系数）分别为直线的斜率和 y 轴截距。系数可以用最小二乘方法求解，它使数据的实际直线与估计直线之间的误差最小化。多元线性回归是（简单）线性回归的扩充，允许响应变量 y 建模为两个或多个预测变量的线性函数。

对数线性模型：用来描述期望频数与协变量（指与因变量有线性相关并在探讨自变量与因变量关系时通过统计技术加以控制的变量）之间的关系。考虑期望频数 m 取值在 0 到正无穷之间，故需要进行对数变换为 $f(m) = \ln m$，使它的取值在 $-\infty$ 与 ∞ 之间。

对数线性模型：

$$\ln m = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \tag{4-28}$$

对数线性模型一般用来近似离散的多维概率分布。在一个 n 元组的集合中，每个元组可以看作是 n 维空间中的一个点。可以使用对数线性模型基于维组合的一个较小子集，估计离散化的属性集的多维空间中每个点的概率，这使得高维数据空间可以由较低维空间构造。因

此，对数线性模型也可以用于维规约（由于低维空间的点通常比原来的数据点占据较少的空间）和数据光滑（因为与较高维空间的估计相比，较低维空间的聚集估计较少受抽样方差的影响）。

4.5 Python 主要数据预处理函数

表 4-7 给出了本节要介绍的 Python 中的插值、数据归一化、主成分分析等与数据预处理相关的函数。本小节对它们进行介绍。

表4-7 Python主要数据预处理函数

函 数 名	函 数 功 能	所属扩展库
interpolate	一维、高维数据插值	Scipy
unique	去除数据中的重复元素，得到单值元素列表，它是对象的方法名	Pandas/Numpy
isnull	判断是否空值	Pandas
notnull	判断是否非空值	Pandas
PCA	对指标变量矩阵进行主成分分析	Scikit-Learn
random	生成随机矩阵	Numpy

（1）interpolate

1）功能：interpolate 是 Scipy 的一个子库，包含了大量的插值函数，如拉格朗日插值、样条插值、高维插值等。使用前需要用 from scipy.interpolate import * 引入相应的插值函数，读者应该根据需要到官网查找对应的函数名。

2）使用格式：f = scipy.interpolate.lagrange(x, y)。这里仅仅展示了一维数据的拉格朗日插值的命令，其中 x, y 为对应的自变量和因变量数据。插值完成后，可以通过 f(a) 计算新的插值结果。类似的还有样条插值、多维数据插值等，此处不一一展示。

（2）unique

1）功能：去除数据中的重复元素，得到单值元素列表。它既是 Numpy 库的一个函数（np.unique()），也是 Series 对象的一个方法。

2）使用格式：

- np.unique(D)，D 是一维数据，可以是 list、array、Series；
- D.unique()，D 是 Pandas 的 Series 对象。

3）实例：求向量 A 中的单值元素，并返回相关索引。

```
>>> D = pd.Series([1, 1, 2, 3, 5])
>>> D.unique()
array([1, 2, 3, 5], dtype=int64)
>>> np.unique(D)
array([1, 2, 3, 5], dtype=int64)
```

（3）isnull/ notnull

1）功能：判断每个元素是否空值 / 非空值。

2）使用格式：D.isnull()/ D.notnull()。这里的 D 要求是 Series 对象，返回一个布尔 Series。可以通过 D[D.isnull()] 或 D[D.notnull()] 找出 D 中的空值 / 非空值。

（4）random

1）功能：random 是 Numpy 的一个子库（Python 本身也自带了 random，但 Numpy 的更加强大），可以用该库下的各种函数生成服从特定分布的随机矩阵，抽样时可使用。

2）使用格式：

- np.random.rand(k, m, n, ...)　生成一个 k × m × n × ... 随机矩阵，其元素均匀分布在区间（0,1）上；
- np.random.randn(k, m, n, ...)　生成一个 k × m × n × ... 随机矩阵，其元素服从标准正态分布。

（5）PCA

1）功能：对指标变量矩阵进行主成分分析。使用前需要用 from sklearn.decomposition import PCA 引入该函数。

2）使用格式：model = PCA()。注意，Scikit-Learn 下的 PCA 是一个建模式的对象，也就是说，一般的流程是建模，然后是训练 model.fit(D)，D 为要进行主成分分析的数据矩阵，训练结束后获取模型的参数，如 .components_ 获取特征向量，以及 .explained_variance_ratio_ 获取各个属性的贡献率等。

3）实例：使用 PCA() 对一个 10 × 4 维的随机矩阵进行主成分分析。

```
>>>from sklearn.decomposition import PCA
>>>D = np.random.rand(10,4)
>>>pca = PCA()
>>>pca.fit(D)
PCA(copy=True, n_components=None, whiten=False)
>>>pca.components_ #返回模型的各个特征向量
array([[-0.42899319, -0.69804397,  0.32876844, -0.46969221],
       [ 0.03680965, -0.0667248 ,  0.7848853 ,  0.61493733],
       [-0.62222716,  0.68499407,  0.28400153, -0.25091755],
       [-0.65379144, -0.19765007, -0.4418252 ,  0.58161989]])
>>>pca.explained_variance_ratio_ #返回各个成分各自的方差百分比
array([ 0.40836652,  0.32861061,  0.21894296,  0.0440799 ])
```

4.6　小结

本章介绍了数据预处理的 4 个主要任务：数据清洗、数据集成、数据变换和数据规约。数据清洗主要介绍了对缺失值和异常值的处理，延续了第 3 章的缺失值和异常值分析的内容，本章所介绍的处理缺失值的方法分为 3 类：删除记录、数据插补和不处理，处理异常值

的方法有删除含有异常值的记录、不处理、平均值修正和视为缺失值；数据集成是合并多个数据源中的数据，并存放到一个数据存储的过程，对该部分的介绍从实体识别问题和冗余属性两个方面进行；数据变换介绍了如何从不同的应用角度对已有属性进行函数变换；数据规约从属性（纵向）规约和数值（横向）规约两个方面介绍了如何对数据进行规约，使挖掘的性能和效率得到很大的提高。通过对原始数据进行相应的处理，将为后续挖掘建模提供良好的数据基础。

第 5 章 Chapter 5

挖掘建模

经过数据探索与数据预处理，得到了可以直接建模的数据。根据挖掘目标和数据形式可以建立分类与预测、聚类分析、关联规则、时序模式和偏差检测等模型，帮助企业提取数据中蕴含的商业价值，提高企业的竞争力。

5.1 分类与预测

就餐饮企业而言，经常会碰到如下问题。

1）如何基于菜品历史销售情况，以及节假日、气候和竞争对手等影响因素，对菜品销量进行趋势预测？

2）如何预测未来一段时间哪些顾客会流失，哪些顾客最有可能会成为 VIP 客户？

3）如何预测一种新产品的销售量，以及在哪种类型的客户中会较受欢迎？

除此之外，餐厅经理需要通过数据分析来帮助他了解具有某些特征的顾客的消费习惯；餐饮企业老板希望知道下个月的销售收入，原材料采购需要投入多少，这些都是分类与预测的例子。

分类和预测是预测问题的两种主要类型，分类主要是预测分类标号（离散属性），而预测主要是建立连续值函数模型，预测给定自变量对应的因变量的值。

5.1.1 实现过程

（1）分类

分类是构造一个分类模型，输入样本的属性值，输出对应的类别，将每个样本映射到预

先定义好的类别。

分类模型建立在已有类标记的数据集上，模型在已有样本上的准确率可以方便地计算，所以分类属于有监督的学习。图 5-1 是一个将销售量分为“高、中、低”3 分类问题。

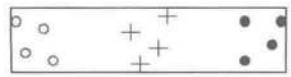


图 5-1 分类问题

（2）预测

预测是指建立两种或两种以上变量间相互依赖的函数模型，然后进行预测或控制。

（3）实现过程

分类和预测的实现过程类似，以分类模型为例，实现过程如图 5-2 所示。

分类算法有两步过程：第一步是学习步，通过归纳分析训练样本集来建立分类模型得到分类规则；第二步是分类步，先用已知的测试样本集评估分类规则的准确率，如果准确率是可以接受的，则使用该模型对未知类标号的待测样本集进行预测。

预测模型的实现也有两步，类似于图 5-2 描述的分类模型，第一步是通过训练集建立预测属性（数值型的）的函数模型，第二步在模型通过检验后进行预测或控制。

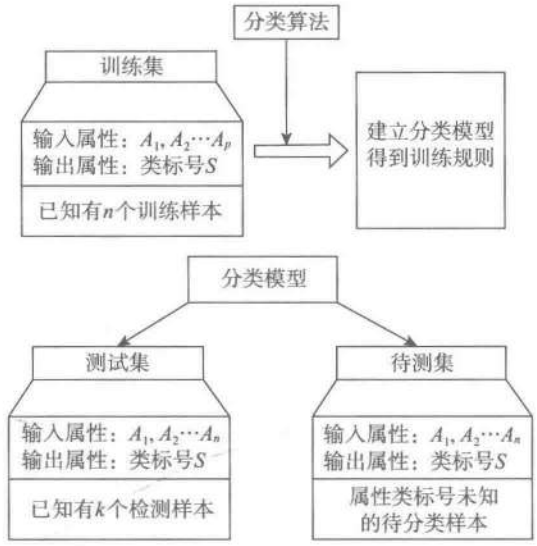


图 5-2 分类模型的实现步骤

5.1.2 常用的分类与预测算法

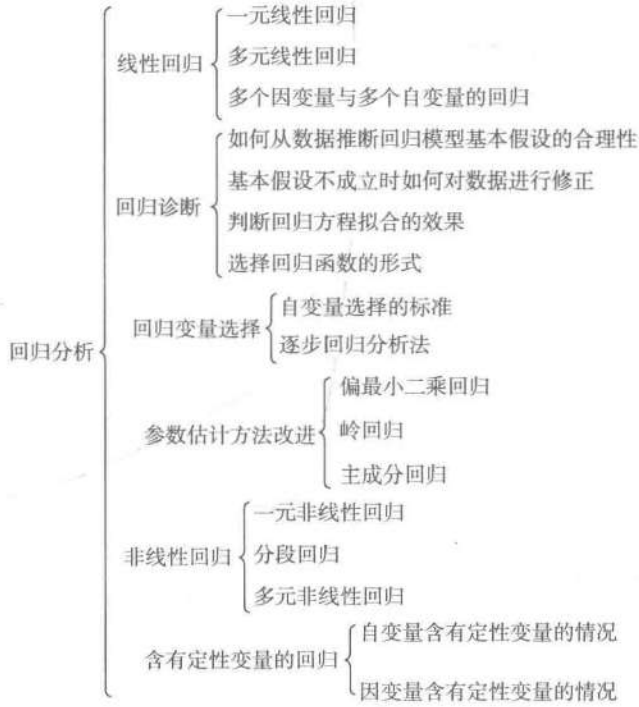
常用的分类与预测算法见表 5-1。

表5-1 主要分类与预测算法简介

算法名称	算法描述
回归分析	回归分析是确定预测属性（数值型）与其他变量间相互依赖的定量关系最常用的统计学方法。包括线性回归、非线性回归、Logistic 回归、岭回归、主成分回归、偏最小二乘回归等模型
决策树	决策树采用自顶向下的递归方式，在内部节点进行属性值的比较，并根据不同的属性值从该节点向下分支，最终得到的叶节点是学习划分的类
人工神经网络	人工神经网络是一种模仿大脑神经网络结构和功能而建立的信息处理系统，表示神经网络的输入与输出变量之间关系的模型
贝叶斯网络	贝叶斯网络又称信度网络，是 Bayes 方法的扩展，是目前不确定知识表达和推理领域最有效的理论模型之一
支持向量机	支持向量机是一种通过某种非线性映射，把低维的非线性可分转化为高维的线性可分，在高维空间进行线性分析的算法

5.1.3 回归分析

回归分析[7]是通过建立模型来研究变量之间相互关系的密切程度、结构状态及进行模型预测的一种有效工具，在工商管理、经济、社会、医学和生物学等领域应用十分广泛。从19世纪初高斯提出最小二乘估计起，回归分析的历史已有200多年。从经典的回归分析方法到近代的回归分析方法，按照研究方法划分，回归分析研究的范围大致如下。



在数据挖掘环境下，自变量与因变量具有相关关系，自变量的值是已知的，因变量是要预测的。

常用的回归模型见表5-2。

表5-2 主要回归模型分类

回归模型名称	适用条件	算法描述
线性回归	因变量与自变量是线性关系	对一个或多个自变量和因变量之间的线性关系进行建模，可用最小二乘法求解模型系数
非线性回归	因变量与自变量之间不都是线性关系	对一个或多个自变量和因变量之间的非线性关系进行建模。如果非线性关系可以通过简单的函数变换转化成线性关系，用线性回归的思想求解；如果不能转化，用非线性最小二乘方法求解
Logistic 回归	因变量一般有1和0(是否)两种取值	是广义线性回归模型的特例，利用Logistic函数将因变量的取值范围控制在0和1之间，表示取值为1的概率

（续）

回归模型名称	适 用 条 件	算 法 描 述
岭回归	参与建模的自变量之间具有多重共线性	是一种改进最小二乘估计的方法
主成分回归	参与建模的自变量之间具有多重共线性	主成分回归是根据主成分分析的思想提出来的，是对最小二乘法的一种改进，它是参数估计的一种有偏估计。可以消除自变量之间的多重共线性

线性回归模型是相对简单的回归模型，但是通常因变量和自变量之间呈现某种曲线关系，就需要建立非线性回归模型。

Logistic 回归属于概率型非线性回归，分为二分类和多分类的回归模型。对于二分类的 Logistic 回归，因变量 y 只有“是、否”两个取值，记为 1 和 0。假设在自变量 $x_1, x_2, \cdots, x_p$ 作用下，y 取“是”的概率是 p，则取“否”的概率是 $1-p$，研究的是当 y 取“是”发生的概率 p 与自变量 $x_1, x_2, \cdots, x_p$ 的关系。

当自变量之间出现多重共线性时，用最小二乘估计的回归系数将会不准确，消除多重共线性的参数改进的估计方法主要有岭回归和主成分回归。

下面就较常用的二分类 Logistic 回归模型的原理进行介绍。

1. Logistic 回归分析介绍

（1）Logistic 函数

Logistic 回归模型中的因变量的只有 1-0（如是和否、发生和不发生）两种取值。假设在 p 个独立自变量 $x_1, x_2, \cdots, x_p$ 作用下，记 y 取 1 的概率是 $p = P(y = 1|X)$，取 0 概率是 $1-p$，取 1 和取 0 的概率之比为 $\frac{p}{1-p}$，称为事件的优势比（odds），对 odds 取自然对数即得 Logistic 变换 $\text{Logit}(p) = \ln\left(\frac{p}{1-p}\right)$。

令 $\text{Logit}(p) = \ln\left(\frac{p}{1-p}\right) = z$，则 $p = \frac{1}{1+e^{-z}}$ 即为 Logistic 函数，如图 5-3 所示。

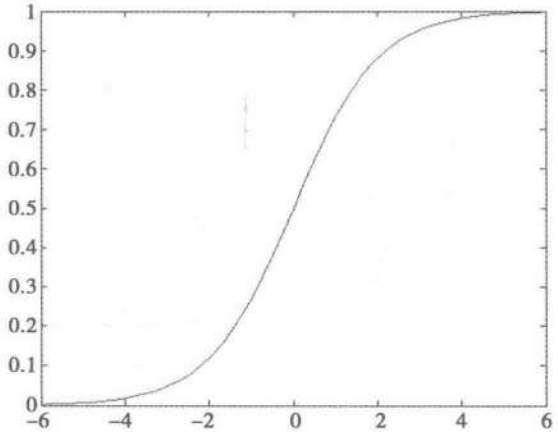


图 5-3 Logistic 函数

当 p 在 (0,1) 之间变化时，odds 的取值范围是 $(0, +\infty)$，则 $\ln\left(\frac{p}{1-p}\right)$ 的取值范围是 $(-\infty, +\infty)$。

（2）Logistic 回归模型

Logistic 回归模型是建立 $\ln\left(\frac{p}{1-p}\right)$ 与自变量的线性回归模型。

Logistic 回归模型为：

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon \tag{5-1}$$

因为 $\ln\left(\frac{p}{1-p}\right)$ 的取值范围是 $(-\infty, +\infty)$，这样，自变量 $x_1, x_2, \cdots, x_p$ 可在任意范围内取值。

记 $g(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$，得到：

$$p = P(y = 1 \mid X) = \frac{1}{1 + e^{-g(x)}} \tag{5-2}$$

$$1 - p = P(y = 0 \mid X) = 1 - \frac{1}{1 + e^{-g(x)}} = \frac{1}{1 + e^{g(x)}} \tag{5-3}$$

（3）Logistic 回归模型解释

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon} \tag{5-4}$$

β_0：在没有自变量，即 $x_1, x_2, \cdots, x_p$ 全部取 0，$y = 1$ 与 $y = 0$ 发生概率之比的自然对数；

β_i：某自变量 x_i 变化时，即 $x_i = 1$ 与 $x_i = 0$ 相比，$y = 1$ 优势比的对数值。

2. Logistic 回归建模步骤

Logistic 回归模型的建模步骤如图 5-4 所示。

1）根据分析目的设置指标变量（因变量和自变量），然后收集数据，根据收集到的数据，对特征再次进行筛选；

2）y 取 1 的概率是 $p = P(y = 1|X)$，取 0 概率是 $1-p$。用 $\ln\left(\frac{p}{1-p}\right)$ 和自变量列出线性回归方程，估计出模型中的回归系数；

3）进行模型检验。模型有效性的检验指标有很多，最基本的有正确率，其次有混淆矩阵、ROC 曲线、KS 值等。

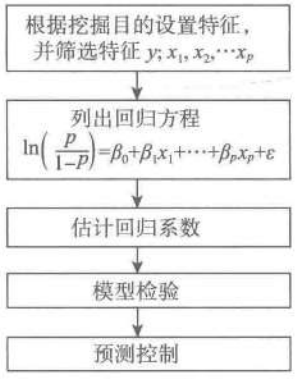


图 5-4 Logistic 回归模型的建模步骤

4）模型应用：输入自变量的取值，就可以得到预测变量的值，或者根据预测变量的值去控制自变量的取值。

下面对某银行在降低贷款拖欠率的数据进行逻辑回归建模，该数据示例如表 5-3 所示。

表5-3 银行贷款拖欠率数据

年　龄	教　育	工　龄	地　址	收　入	负债率	信用卡负债	其他负债	违　约
41	3	17	12	176.00	9.30	11.36	5.01	1

（续）

年　龄	教　育	工　龄	地　址	收　入	负债率	信用卡负债	其他负债	违　约
27	1	10	6	31.00	17.30	1.36	4.00	0
40	1	15	14	55.00	5.50	0.86	2.17	0
41	1	15	14	120.00	2.90	2.66	0.82	0
24	2	2	0	28.00	17.30	1.79	3.06	1

数据详见：示例程序 /data/bankloan.xls

利用 Scikit-Learn 对这个数据进行逻辑回归分析。首先进行特征筛选，特征筛选的方法有很多，主要包含在 Scikit_Learn 的 feature_selection 库中，比较简单的有通过 F 检验（f_regression）来给出各个特征的 F 值和 p 值，从而可以筛选变量（选择 F 值大的或者 p 值小的特征）。其次有递归特征消除（Recursive Feature Elimination，RFE）和稳定性选择（Stability Selection）等比较新的方法。这里使用了稳定性选择方法中的随机逻辑回归进行特征筛选，然后利用筛选后的特征建立逻辑回归模型，输出平均正确率，其代码如代码清单 5-1 所示。

代码清单5-1　逻辑回归代码

```
#-*- coding: utf-8 -*-
#逻辑回归 自动建模
import pandas as pd

#参数初始化
filename = '../data/bankloan.xls'
data = pd.read_excel(filename)
x = data.iloc[:,:8].as_matrix()
y = data.iloc[:,8].as_matrix()

from sklearn.linear_model import LogisticRegression as LR
from sklearn.linear_model import RandomizedLogisticRegression as RLR
rlr = RLR() #建立随机逻辑回归模型，筛选变量
rlr.fit(x, y) #训练模型
rlr.get_support() #获取特征筛选结果，也可以通过.scores_方法获取各个特征的分数
print(u'通过随机逻辑回归模型筛选特征结束。')
print(u'有效特征为：%s' % ','.join(data.columns[rlr.get_support()]))
x = data[data.columns[rlr.get_support()]].as_matrix() #筛选好特征

lr = LR() #建立逻辑货柜模型
lr.fit(x, y) #用筛选后的特征数据来训练模型
print(u'逻辑回归模型训练结束。')
print(u'模型的平均正确率为：%s' % lr.score(x, y)) #给出模型的平均正确率，本例为81.4%
```

代码详见：示例程序 /code/logistic_regression.py

运行代码清单 5-1，可以得到部分输出结果如下。

```
通过随机逻辑回归模型筛选特征结束。
有效特征为：工龄,地址,负债率,信用卡负债
逻辑回归模型训练结束。
模型的平均正确率为：0.814285714286
```

递归特征消除的主要思想是反复的构建模型（如SVM或者回归模型）然后选出最好的（或者最差的）的特征（可以根据系数来选），把选出来的特征放到一边，然后在剩余的特征上重复这个过程，直到遍历所有特征。这个过程中特征被消除的次序就是特征的排序。因此，这是一种寻找最优特征子集的贪心算法。Scikit-Learn提供了RFE包，可以用于特征消除，还提供了RFECV，可以通过交叉验证来对特征进行排序。

稳定性选择是一种基于二次抽样和选择算法相结合较新的方法，选择算法可以是回归、SVM或其他类似的方法。它的主要思想是在不同的数据子集和特征子集上运行特征选择算法，不断重复，最终汇总特征选择结果。比如，可以统计某个特征被认为是重要特征的频率（被选为重要特征的次数除以它所在的子集被测试的次数）。在理想情况下，重要特征的得分会接近100%。稍微弱一点的特征得分会是非0的数，而最无用的特征得分将会接近于0。Scikit-Learn在随机Lasso和随机逻辑回归中有对稳定性选择的实现。

从上面的结果可以看出，采用随机逻辑回归剔除变量，分别剔除了x2、x8、x1、x5，最终构建的模型包含的变量为常量x3、x4、x6、x7。在建立随机逻辑回归模型时，使用了默认的阈值0.25，读者可以用RLR(selection_threshold = 0.5)手动设置阈值。此外，在本例中，使用随机Lasso、甚至仅仅简单地采用F回归（f_rgression）也能够得到类似的结果。

逻辑回归本质上还是一种线性模型，因此这里的模型有效性检验本质上还是在做线性相关检验，因此，所筛选出来的变量，说明与结果具有比较强的线性相关性，然而被筛掉的变量并不一定就跟结果没有关系，因为它们之间有可能是非线性关系。因此，读者还需要根据问题的实际背景对筛选结果进行分析。对于非线性关系的变量筛选方法有决策树、神经网络等。

5.1.4 决策树

决策树方法在分类、预测、规则提取等领域有着广泛应用。20世纪70年代后期和80年代初期，机器学习研究者J.Ross Quinlan提出了ID3[8]算法以后，决策树在机器学习、数据挖掘领域得到极大的发展。Quinlan后来又提出了C4.5，成为新的监督学习算法。1984年，几位统计学家提出了CART分类算法。ID3和CART算法几乎同时被提出，但都是采用类似的方法从训练样本中学习决策树。

决策树是一树状结构，它的每一个叶节点对应着一个分类，非叶节点对应着在某个属性上的划分，根据样本在该属性上的不同取值将其划分成若干个子集。对于非纯的叶节点，多数类的标号给出到达这个节点的样本所属的类。构造决策树的核心问题是在每一步如何选择适当的属性对样本做拆分。对一个分类问题，从已知类标记的训练样本中学习并构造出决策树是一个自上而下，分而治之的过程。

常用的决策树算法见表 5-4。

表5-4 决策树算法分类

决策树算法	算 法 描 述
ID3 算法	其核心是在决策树的各级节点上，使用信息增益方法作为属性的选择标准，来帮助确定生成每个节点时所应采用的合适属性
C4.5 算法	C4.5 决策树生成算法相对于 ID3 算法的重要改进是使用信息增益率来选择节点属性。C4.5 算法可以克服 ID3 算法存在的不足：ID3 算法只适用于离散的描述属性，而 C4.5 算法既能够处理离散的描述属性，也可以处理连续的描述属性
CART 算法	CART 决策树是一种十分有效的非参数分类和回归方法，通过构建树、修剪树、评估树来构建一个二叉树。当终结点是连续变量时，该树为回归树；当终结点是分类变量，该树为分类树

本节将详细介绍 ID3 算法，也是最经典的决策树分类算法。

1. ID3 算法简介及基本原理

ID3 算法基于信息熵来选择最佳测试属性。它选择当前样本集中具有最大信息增益值的属性作为测试属性；样本集的划分则依据测试属性的取值进行，测试属性有多少不同取值就将样本集划分为多少子样本集，同时决策树上相应于该样本集的节点长出新的叶子节点。ID3 算法根据信息论理论，采用划分后样本集的不确定性作为衡量划分好坏的标准，用信息增益值度量不确定性：信息增益值越大，不确定性越小。因此，ID3 算法在每个非叶节点选择信息增益最大的属性作为测试属性，这样可以得到当前情况下最纯的拆分，从而得到较小的决策树。

设 S 是 s 个数据样本的集合。假定类别属性具有 m 个不同的值：$C_i(i = 1, 2, \cdots, m)$。设 s_i 是类 C_i 中的样本数。对一个给定的样本，它总的信息熵为

$$I(s_1, s_2, \cdots, s_m) = -\sum_{i=1}^{m} P_i \log_2(P_i) \tag{5-5}$$

其中，P_i 是任意样本属于 C_i 的概率，一般可以用 $\frac{s_i}{s}$ 估计。

设一个属性 A 具有 k 个不同的值 $\{a_1, a_2, \cdots, a_k\}$，利用属性 A 将集合 S 划分为个子集 $\{S_1, S_2, \cdots, S_k\}$，其中 S_j 包含了集合 S 中属性 A 取 a_j 值的样本。若选择属性 A 为测试属性，则这些子集就是从集合 S 的节点生长出来的新的叶节点。设 s_{ij} 是子集 S_j 中类别为 C_i 的样本数，则根据属性 A 划分样本的信息熵值为

$$E(A) = \sum_{j=1}^{k} \frac{s_{1j} + s_{2j} + \cdots + s_{mj}}{s} I(s_{1j}, s_{2j}, \cdots, s_{mj}) \tag{5-6}$$

其中，$I(s_{1j}, s_{2j}, \cdots, s_{mj}) = -\sum_{i=1}^{m} P_{ij} \log_2(P_{ij})$，$P_{ij} = \frac{s_{ij}}{s_{1j} + s_{2j} + \cdots + s_{mj}}$ 是子集 S_j 中类别为 C_i 的样本的概率。

最后，用属性 A 划分样本集 S 后所得的信息增益（Gain）为

$$Gain(A) = I(s_1, s_2, \cdots, s_m) - E(A) \tag{5-7}$$

显然 $E(A)$ 越小，$Gain(A)$ 的值越大，说明选择测试属性 A 对于分类提供的信息越大，选择 A 之后对分类的不确定程度越小。属性 A 的 k 个不同的值对应样本集 S 的 k 个子集或分支，通过递归调用上述过程（不包括已经选择的属性），生成其他属性作为节点的子节点和分支来生成整个决策树。ID3 决策树算法作为一个典型的决策树学习算法，其核心是在决策树的各级节点上都用信息增益作为判断标准进行属性的选择，使得在每个非叶节点上进行测试时，都能获得最大的类别分类增益，使分类后数据集的熵最小。这样的处理方法使得树的平均深度较小，从而有效地提高了分类效率。

2. ID3 算法具体流程

ID3 算法的具体详细实现步骤如下。

1）对当前样本集合，计算所有属性的信息增益；

2）选择信息增益最大的属性作为测试属性，把测试属性取值相同的样本划为同一个子样本集；

3）若子样本集的类别属性只含有单个属性，则分支为叶子节点，判断其属性值并标上相应的符号，然后返回调用处；否则对子样本集递归调用本算法。

下面将结合餐饮案例实现 ID3 的具体实施步骤。T 餐饮企业作为大型连锁企业，生产的产品种类比较多，另外涉及的分店所处的位置也不同，数目比较多。对于企业的高层来讲，了解周末和非周末销量是否有大的区别，以及天气、促销活动这些因素是否能够影响门店的销量等信息至关重要。因此，为了让决策者准确了解和销量有关的一系列影响因素，需要构建模型来分析天气、是否周末和是否有促销活动对销量的影响，下面以单个门店为例来进行分析。

对于天气属性，数据源中存在多种不同的值，这里将那些属性值相近的值进行类别整合。如天气为“多云”“多云转晴”“晴”这些属性值相近，均是适宜外出的天气，不会对产品销量有太大的影响，因此将它们归为一类，天气属性值设置为“好”。同理，对于“雨”“小到中雨”等天气，均是不适宜外出的天气，因此将它们归为一类，天气属性值设置为“坏”。

对于是否周末属性，周末则设置为“是”，非周末则设置为“否”。

对于是否有促销活动属性，有促销则设置为“是”，无促销则设置为“否”。

产品的销售数量为数值型，需要对属性进行离散化，将销售数据划分为“高”和“低”两类。将其平均值作为分界点，大于平均值的划分到类别“高”，小于平均值的划分为“低”类别。

经过以上的处理，我们得到的数据集合见表 5-5。

表5-5 处理后的数据集

序　号	天　气	是否周末	是否有促销	销　量
1	坏	是	是	高
2	坏	是	是	高

（续）

序　号	天　气	是否周末	是否有促销	销　量
3	坏	是	是	高
4	坏	否	是	高
…	…	…	…	…
32	好	否	是	低
33	好	否	否	低
34	好	否	否	低

数据详见：示例程序 /data/sales_data.xls

采用 ID3 算法构建决策树模型的具体步骤如下。

1）根据公式（5-5），计算总的信息熵。其中，数据中总记录数为 34，而销售数量为“高”的数据有 18，“低”的有 16。

$$I(18,16) = -\frac{18}{34}\log_2\frac{18}{34} - \frac{16}{34}\log_2\frac{16}{34} = 0.997\ 503$$

2）根据公式（5-5）和（5-6），计算每个测试属性的信息熵。

对于天气属性，其属性值有“好”和“坏”两种。其中，天气为“好”的条件下，销售数量为“高”的记录为 11，销售数量为“低”的记录为 6，可表示为 (11,6)；天气为“坏”的条件下，销售数量为“高”的记录为 7，销售数量为“低”的记录为 10，可表示为 (7,10)。则天气属性的信息熵计算过程如下。

$$I(11,6) = -\frac{11}{17}\log_2\frac{11}{17} - \frac{6}{17}\log_2\frac{6}{17} = 0.936\ 667$$

$$I(7,10) = -\frac{7}{17}\log_2\frac{7}{17} - \frac{10}{17}\log_2\frac{10}{17} = 0.977\ 418$$

$$E(\text{天气}) = \frac{17}{34}I(11,6) + \frac{17}{34}I(7,10) = 0.957\ 043$$

对于是否周末属性，其属性值有“是”和“否”两种。其中，是否周末属性为“是”的条件下，销售数量为“高”的记录为 11，销售数量为“低”的记录为 3，可表示为 (11，3)；是否周末属性为“否”的条件下，销售数量为“高”的记录为 7，销售数量为“低”的记录为 13，可表示为 (7,13)。则节假日属性的信息熵计算过程如下。

$$I(11,3) = -\frac{11}{14}\log_2\frac{11}{14} - \frac{3}{14}\log_2\frac{3}{14} = 0.749\ 595$$

$$I(7,13) = -\frac{7}{20}\log_2\frac{7}{20} - \frac{13}{20}\log_2\frac{13}{20} = 0.934\ 068$$

$$E(\text{是否周末}) = \frac{14}{34}I(11,3) + \frac{20}{34}I(7,13) = 0.858\ 109$$

对于是否有促销属性，其属性值有“是”和“否”两种。其中，是否有促销属性为

“是”的条件下，销售数量为“高”的记录为15，销售数量为“低”的记录为7，可表示为（15,7）；其中，是否有促销属性为“否”的条件下，销售数量为“高”的记录为3，销售数量为“低”的记录为9，可表示为(3，9)。则是否有促销属性的信息熵计算过程如下。

$$I(15,7) = -\frac{15}{22}\log_2\frac{15}{22} - \frac{7}{22}\log_2\frac{7}{22} = 0.902\ 393$$

$$I(3,9) = -\frac{3}{12}\log_2\frac{3}{12} - \frac{9}{12}\log_2\frac{9}{12} = 0.811\ 278$$

$$E(\text{是否有促销}) = \frac{22}{34}I(15,7) + \frac{12}{34}I(3,9) = 0.870\ 235$$

3）根据公式（5-7），计算天气、是否周末和是否有促销属性的信息增益值。

$$Gain(\text{天气}) = I(18,16) - E(\text{天气}) = 0.997\ 503 - 0.957\ 043 = 0.040\ 46$$

$$Gain(\text{是否周末}) = I(18,16) - E(\text{是否周末}) = 0.997\ 503 - 0.858\ 109 = 0.139\ 394$$

$$Gain(\text{是否有促销}) = I(18,16) - E(\text{是否有促销}) = 0.997\ 503 - 0.870\ 235 = 0.127\ 268$$

4）由第3）步的计算结果可以知道，是否周末属性的信息增益值最大，它的两个属性值“是”和“否”作为该根结点的两个分支。然后按照第1）步到第3）步所示步骤继续对该根结点的3个分支进行结点的划分，针对每一个分支结点继续进行信息增益的计算，如此循环反复，直到没有新的结点分支，最终构成一棵决策树。生成的决策树模型如图5-5所示。

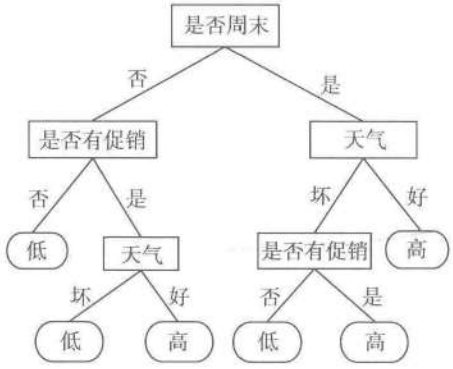


图5-5 ID3生成的决策树模型

从上面的决策树模型可以看出门店的销售高低和各个属性之间的关系，并可以提取出以下决策规则。

- 若周末属性为“是”，天气为“好”，则销售数量为“高”。
- 若周末属性为“是”，天气为“坏”，促销属性为“是”，则销售数量为“高”。
- 若周末属性为“是”，天气为“坏”，促销属性为“否”，则销售数量为“低”。
- 若周末属性为“否”，促销属性为“否”，则销售数量为“低”。
- 若周末属性为“否”，促销属性为“是”，天气为“好”，则销售数量为“高”。
- 若周末属性为“否”，促销属性为“是”，天气为“坏”，则销售数量为“低”。

由于ID3决策树算法采用了信息增益作为选择测试属性的标准，会偏向于选择取值较多的，即所谓高度分支属性，而这类属性并不一定是最优的属性。同时ID3决策树算法只能处理离散属性，对于连续型的属性，在分类前需要对其进行离散化。为了解决倾向于选择高度分支属性的问题，人们采用信息增益率作为选择测试属性的标准，这样便得到C4.5决策树算法。此外，常用的决策树算法还有CART算法、SLIQ算法、SPRINT算法和PUBLIC算法等。

使用 Scikit-Learn 建立基于信息熵的决策树模型，如代码清单 5-2 所示。

代码清单5-2 决策树算法预测销量高低代码

```
#-*- coding: utf-8 -*-
#使用ID3决策树算法预测销量高低
import pandas as pd

#参数初始化
filename = '../data/sales_data.xls'
data = pd.read_excel(filename, index_col = u'序号') #导入数据

#数据是类别标签，要将它转换为数据
#用1来表示“好” “是” “高”这三个属性，用-1来表示“坏” “否” “低”
data[data == u'好'] = 1
data[data == u'是'] = 1
data[data == u'高'] = 1
data[data != 1] = -1
x = data.iloc[:,:3].as_matrix().astype(int)
y = data.iloc[:,3].as_matrix().astype(int)

from sklearn.tree import DecisionTreeClassifier as DTC
dtc = DTC(criterion='entropy') #建立决策树模型，基于信息熵
dtc.fit(x, y) #训练模型

#导入相关函数，可视化决策树。
#导出的结果是一个dot文件，需要安装Graphviz才能将它转换为pdf或png等格式。
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
with open("tree.dot", 'w') as f:
  f = export_graphviz(dtc, feature_names = x.columns, out_file = f)
```

代码详见：示例程序 /code/decision_tree.py

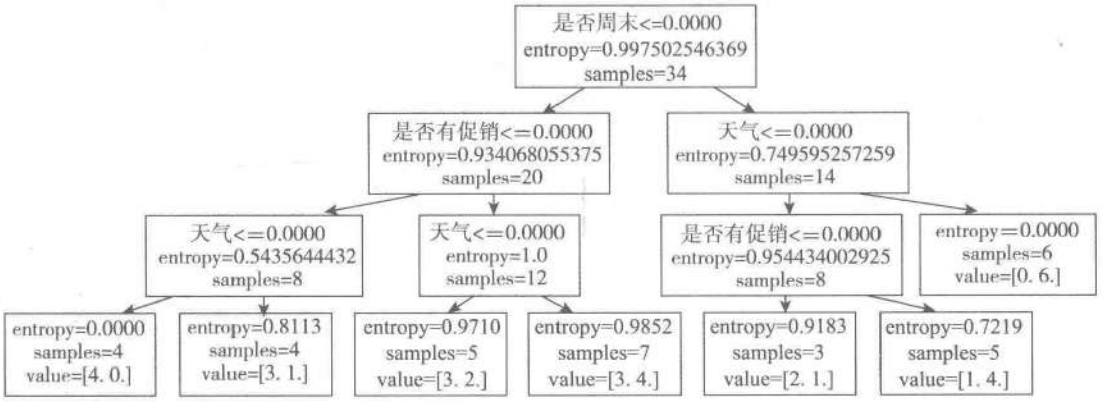
运行代码后，将会输出一个 tree.dot 的文本文件。由于我们输出中包含中文，需要用文本编辑器在文件中指定中文字体，如，

```
digraph Tree {
edge [fontname="SimHei"];  /*添加这两行，指定中文字体（这里是黑体）*/
node [fontname="SimHei"];  /*添加这两行，指定中文字体（这里是黑体）*/
0 [label="是否周末 <= 0.0000\nentropy = 0.997502546369\nsamples = 34", shape="box"] ;
1 [label="是否有促销 <= 0.0000\nentropy = 0.934068055375\nsamples = 20", shape="box"] ;
...
}
```

然后将它保存为 UTF-8 格式。为了进一步将它转换为可视化格式，需要安装 Graphviz（跨平台的、基于命令行的绘图工具），然后在命令行中以如下方式编译。

```
dot -Tpdf tree.dot -o tree.pdf
dot -Tpng tree.dot -o tree.pdf
```

生成的结果图如下，显然，它等价于图 5-5。



5.1.5 人工神经网络

人工神经网络[9][10]（Artificial Neural Networks，ANN），是模拟生物神经网络进行信息处理的一种数学模型。它以对大脑的生理研究成果为基础，其目的在于模拟大脑的某些机理与机制，实现一些特定的功能。

1943 年，美国心理学家 McCulloch 和数学家 Pitts 联合提出了形式神经元的数学模型 MP 模型，证明了单个神经元能执行逻辑功能，开创了人工神经网络研究的时代。1957 年，计算机科学家 Rosenblatt 用硬件完成了最早的神经网络模型，即感知器，并用来模拟生物的感知和学习能力。1969 年 M.Minsky 等仔细分析了以感知器为代表的神经网络系统的功能及局限后，出版了《Perceptron》(感知器）一书，指出感知器不能解决高阶谓词问题，人工神经网络的研究进入一个低谷期。20 世纪 80 年代以后，超大规模集成电路、脑科学、生物学、光学的迅速发展为人工神经网络的发展打下了基础，人工神经网络的发展进入兴盛期。

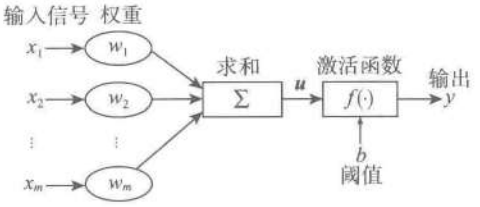


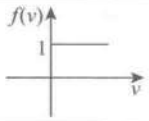
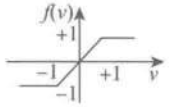
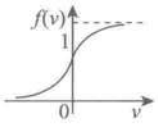
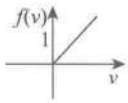
图 5-6 人工神经元模型

人工神经元是人工神经网络操作的基本信息处理单位。人工神经元的模型如图 5-6 所示，它是人工神经网络的设计基础。一个人工神经元对输入信号 $X = [x_1, x_2, \cdots x_m]^T$ 的输出 y 为 $y = f(u + b)$，其中 $u = \sum_{i=1}^{m} w_i x_i$，公式中各字符的含义如图 5-6 所示。

激活函数主要有以下 3 种形式，见表 5-6。

人工神经网络的学习也称为训练，指的是神经网络在受到外部环境的刺激下调整神经网络的参数，使神经网络以一种新的方式对外部环境作出反应的一个过程。在分类与预测中，人工神将网络主要使用有指导的学习方式，即根据给定的训练样本，调整人工神经网络的参数以使网络输出接近于已知的样本类标记或其他形式的因变量。

表5-6 激活函数分类表

激活函数	表达形式	图 形	解释说明
域值函数(阶梯函数)	$f(v)=\begin{cases}1 & v\geqslant 0\\0 & v<0\end{cases}$		当函数的自变量小于0时，函数的输出为0；当函数的自变量大于或等于0时，函数的输出为1，用该函数可以把输入分成两类
分段线性函数	$f(v)=\begin{cases}1, & v\geqslant 1\\v, & -1<v<1\\-1, & v\leqslant -1\end{cases}$		该函数在（-1，+1）线性区内的放大系数是一致的，这种形式的激活函数可以看作是非线性放大器的近似
非线性转移函数	$f(v)=\dfrac{1}{1+e^{-v}}$		单极性S型函数为实数域R到[0，1]闭集的连续函数，代表了连续状态型神经元模型。其特点是函数本身及其导数都是连续的，能够体现数学计算上的优越性
Relu函数	$f(v)=\begin{cases}v, & v\geqslant 0\\0, & v<0\end{cases}$		这是近年来提出的激活函数，它具有计算简单、效果更佳的特点，目前已经有取代其他激活函数的趋势。本书的神经网络模型大量使用了该激活函数

在人工神经网络的发展过程中，提出了多种不同的学习规则，没有一种特定的学习算法适用于所有的网络结构和具体问题。在分类与预测中，δ学习规则（误差校正学习算法）是使用最广泛的一种。误差校正学习算法根据神经网络的输出误差对神经元的连接强度进行修正，属于有指导学习。设神经网络中神经元i作为输入，神经元j为输出神经元，它们的连接权值为w_{ij}，则对权值的修正为$\Delta w_{ij}=\eta\delta_j Y_i$，其中$\eta$为学习率，$\delta_j=T_j-Y_j$为$j$的偏差，即输出神经元$j$的实际输出和教师信号之差，示意图如图5-7所示。

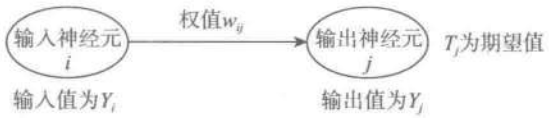


图5-7 δ学习规则示意图

神经网络训练是否完成常用误差函数（也称目标函数）E来衡量。当误差函数小于某一个设定的值时即停止神经网络的训练。误差函数为衡量实际输出向量Y_k与期望值向量T_k误差大小的函数，常采用二乘误差函数来定义为$E=\frac{1}{2}\sum_{k=1}^{N}[Y_k-T_k]^2$（或$E=\sum_{k=1}^{N}[Y_k-T_k]^2$）$k=1,2,\cdots,N$为训练样本个数。

使用人工神经网络模型需要确定网络连接的拓扑结构、神经元的特征和学习规则等。目前，已有近40种人工神经网络模型，常用的用来实现分类和预测的人工神经网络算法见表5-7。

表5-7 人工神经网络算法

算法名称	算法描述
BP神经网络	是一种按误差逆传播算法训练的多层前馈网络，学习算法是δ学习规则，是目前应用最广泛的神经网络模型之一

（续）

算法名称	算法描述
LM 神经网络	是基于梯度下降法和牛顿法结合的多层前馈网络，特点：迭代次数少，收敛速度快，精确度高
RBF 径向基神经网络	RBF 网络能够以任意精度逼近任意连续函数，从输入层到隐含层的变换是非线性的，而从隐含层到输出层的变换是线性的，特别适合于解决分类问题
FNN 模糊神经网络	FNN 模糊神经网络是具有模糊权系数或者输入信号是模糊量的神经网络，是模糊系统与神经网络相结合的产物，它汇聚了神经网络与模糊系统的优点，集联想、识别、自适应及模糊信息处理于一体
GMDH 神经网络	GMDH 网络也称为多项式网络，它是前馈神经网络中常用的一种用于预测的神经网络。它的特点是网络结构不固定，而且在训练过程中不断改变
ANFIS 自适应神经网络	神经网络镶嵌在一个全部模糊的结构之中，在不知不觉中向训练数据学习，自动产生、修正并高度概括出最佳的输入与输出变量的隶属函数以及模糊规则；另外，神经网络的各层结构与参数也都具有了明确的、易于理解的物理意义

BP 神经网络的学习算法是 δ 学习规则，目标函数采用 $E = \sum_{k=1}^{N} [Y_k - T_k]^2$，下面详细介绍 BP 神经网络算法。

反向传播（Back Propagation，BP）算法的特征是利用输出后的误差来估计输出层的直接前导层的误差，再用这个误差估计更前一层的误差，如此一层一层的反向传播下去，就获得了所有其他各层的误差估计。这样就形成了将输出层表现出的误差沿着与输入传送相反的方向逐级向网络的输入层传递的过程。这里我们以典型的三层 BP 网络为例，描述标准的 BP 算法。图 5-8 所示的是一个有 3 个输入节点，4 个隐层节点，1 个输出节点的一个三层 BP 神经网络。

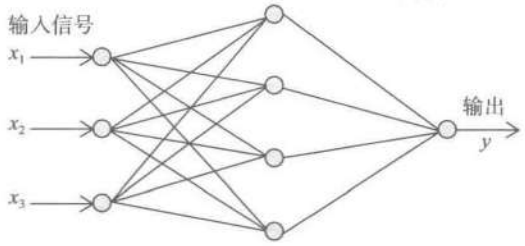


图 5-8 三层 BP 神经网络结构

BP 算法的学习过程由信号的正向传播与误差的逆向传播两个过程组成。正向传播时，输入信号经过隐层的处理后，传向输出层。若输出层节点未能得到期望的输出，则转入误差的逆向传播阶段，将输出误差按某种子形式，通过隐层向输入层返回，并“分摊”给隐层 4 个节点与输入层 x_1、x_2、x_3 三个输入节点，从而获得各层单元的参考误差或称误差信号，作为修改各单元权值的依据。这种信号正向传播与误差逆向传播的各层权矩阵的修改过程，是周而复始进行的。权值不断修改的过程，也就是网络的学习（或称训练）过程。此过程一直进行到网络输出的误差逐渐减少到可接受的程度或达到设定的学习次数为止，学习过程的流程图如图 5-9 所示。

算法开始后，给定学习次数上限，初始化学习次数为 0，对权值和阈值赋予小的随机数，一般在 [-1,1] 之间。输入样本数据，网络正向传播，得到中间层与输出层的值。比较输出层的值与教师信号值的误差，用误差函数 E 来判断误差是否小于误差上限，如不小于误差上限，则对中间层和输出层权值和阈值进行更新，更新的算法为 δ 学习规则。更新权值和阈值

后，再次将样本数据作为输入，得到中间层与输出层的值，计算误差 E 是否小于上限，学习次数是否到达指定值，如果达到，则学习结束。

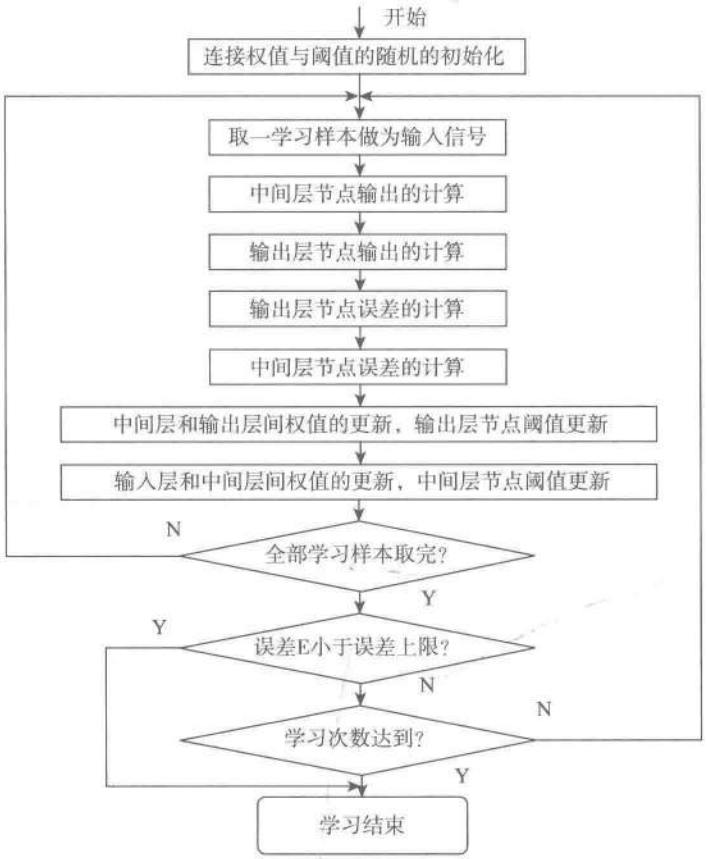


图 5-9 BP 算法学习过程流程图

BP 算法只用到均方误差函数对权值和阈值的一阶导数（梯度）的信息，使得算法存在收敛速度缓慢、易陷入局部极小等缺陷。为了解决这一问题，Hinton 等人于 2006 年提出了非监督贪心逐层训练算法，为解决深层结构相关的优化难题带来希望，并以此为基础发展成了如今脍炙人口的“深度学习”算法。本书中所建立的神经网络，结构跟传统的 BP 神经网络是类似的，但是求解算法已经用了新的逐层训练算法。限于篇幅，本文不对深度学习做进一步的讲解。有兴趣的读者，请自行搜索并阅读相关资料。

在第 2 章我们已经提过，Scikit-Learn 中并没有神经网络模型，我们认为 Python 中比较好的神经网络算法库是 Keras，这是一个强大而易用的深度学习算法库。在本书中，我们仅仅牛刀小试，把它当成一个基本的神经网络算法库来看待。

针对表 5-5 的数据应用神经网络算法进行建模，建立的神经网络有 3 个输入节点、10 个隐藏节点和 1 个输出节点，其 Python 代码如代码清单 5-3 所示。

代码清单5-3 神经网络算法预测销量高低

```
#-*- coding: utf-8 -*-
#使用神经网络算法预测销量高低

import pandas as pd

#参数初始化
inputfile = '../data/sales_data.xls'
data = pd.read_excel(inputfile, index_col = u'序号') #导入数据

#数据是类别标签，要将它转换为数据
#用1来表示“好” “是” “高”这3个属性，用0来表示“坏” “否” “低”
data[data == u'好'] = 1
data[data == u'是'] = 1
data[data == u'高'] = 1
data[data != 1] = 0
x = data.iloc[:,:3].as_matrix().astype(int)
y = data.iloc[:,3].as_matrix().astype(int)

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(3, 10))
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(10, 1))
model.add(Activation('sigmoid')) #由于是0-1输出，用sigmoid函数作为激活函数

model.compile(loss = 'binary_crossentropy', optimizer = 'adam', class_mode = 'binary')
#编译模型。由于我们做的是二元分类，所以我们指定损失函数为binary_crossentropy，以及模式为binary
#另外常见的损失函数还有mean_squared_error、categorical_crossentropy等，请阅读帮助文件。
#求解方法我们指定用adam，还有sgd、rmsprop等可选

model.fit(x, y, nb_epoch = 1000, batch_size = 10) #训练模型，学习一千次
yp = model.predict_classes(x).reshape(len(y)) #分类预测

from cm_plot import * #导入自行编写的混淆矩阵可视化函数
cm_plot(y,yp).show() #显示混淆矩阵可视化结果
```

代码详见：示例程序 /code/neural_network.py

运行上面的代码，可以得到下面的混淆矩阵图，如图 5-10 所示。

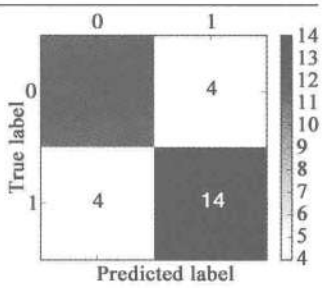


图 5-10 BP 神经网络预测销量高低混淆矩阵图

从图 5-10 可以看出，检测样本为 34 个，预测正确的个数为 26 个，预测准确率为 76.4%，预测准确率较低，是由于神经网络训练时需要较多样本，而这里是由于训练数据较少造成的。

需要指出的是，这里的案例比较简单，我们并没有考虑过拟合的问题。事实上，神经网络的拟合能力是很强的，容易出现过

拟合现象。跟传统的添加“惩罚项”的做法不同，目前神经网络（尤其是深度神经网络）中流行的防止过拟合的方法是随机地让部分神经网络节点休眠。

5.1.6 分类与预测算法评价

分类与预测模型对训练集进行预测而得出的准确率并不能很好地反映预测模型未来的性能，为了有效判断一个预测模型的性能表现，需要一组没有参与预测模型建立的数据集，并在该数据集上评价预测模型的准确率，这组独立的数据集叫作测试集。模型预测效果评价，通常用相对 / 绝对误差、平均绝对误差、均方误差、均方根误差等指标来衡量。

（1）绝对误差与相对误差

设 Y 表示实际值，$\hat{Y}$表示预测值，则称 E 为绝对误差（Absolute Error），计算公式如下。

$$E = Y - \hat{Y} \tag{5-8}$$

e 为相对误差（RelativeError），计算公式如下。

$$e = \frac{Y - \hat{Y}}{Y} \tag{5-9}$$

有时相对误差也用百分数表示。

$$e = \frac{Y - \hat{Y}}{Y} * 100\% \tag{5-10}$$

这是一种直观的误差表示方法。

（2）平均绝对误差

平均绝对误差（MeanAbsoluteError，MAE）定义如下。

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |E_i| = \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{Y}_i| \tag{5-11}$$

式中各项的含义如下。

- MAE：平均绝对误差。
- E_i：第 i 个实际值与预测值的绝对误差。
- Y_i：第 i 个实际值。
- $\hat{Y}_i$：第 i 个预测值。

由于预测误差有正有负，为了避免正负相抵消，故取误差的绝对值进行综合并取其平均数，这是误差分析的综合指标法之一。

（3）均方误差

均方误差（Mean Squared Error，MSE）定义如下。

$$MSE = \frac{1}{n}\sum_{i=1}^{n} E_i^2 = \frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{5-12}$$

在上式中，MSE 表示均方差，其他符号同前。

本方法用于还原平方失真程度。

均方误差是预测误差平方之和的平均数，它避免了正负误差不能相加的问题。由于对误差 E 进行了平方，加强了数值大的误差在指标中的作用，从而提高了这个指标的灵敏性，是

一大优点。均方误差是误差分析的综合指标法之一。

（4）均方根误差

均方根误差（Root Mean Squared Error，RMSE）定义如下。

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} E_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{5-13}$$

上式中，*RMSE* 表示均方根误差，其他符号同前。

这是均方误差的平方根，代表了预测值的离散程度，也称为标准误差，最佳拟合情况为 $RMSE = 0$。均方根误差也是误差分析的综合指标之一。

（5）平均绝对百分误差

平均绝对百分误差（Mean Absolute Percentage Error，MAPE）定义如下。

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |E_i/Y_i| = \frac{1}{n}\sum_{i=1}^{n} |(Y_i - \hat{Y}_i)/Y_i| \tag{5-14}$$

上式中，*MAPE* 表示平均绝对百分误差。一般认为 *MAPE* 小于 10 时，预测精度较高。

（6）Kappa 统计

Kappa 统计是比较两个或多个观测者对同一事物，或观测者对同一事物的两次或多次观测结果是否一致，以由于机遇造成的一致性和实际观测的一致性之间的差别大小作为评价基础的统计指标。Kappa 统计量和加权 Kappa 统计量不仅可以用于无序和有序分类变量资料的一致性、重现性检验，而且能给出一个反映一致性大小的“量”值。

Kappa 取值在［−1，+1］之间，其值的大小均有不同意义。

- Kappa = +1 说明两次判断的结果完全一致。
- Kappa = −1 说明两次判断的结果完全不一致。
- Kappa = 0 说明两次判断的结果是机遇造成。
- Kappa < 0 说明一致程度比机遇造成的还差，两次检查结果很不一致，在实际应用中无意义。
- Kappa > 0 此时说明有意义，Kappa 越大，说明一致性越好。
- Kappa ≥ 0.75 说明已经取得相当满意的一致程度。
- Kappa < 0.4 说明一致程度不够。

（7）识别准确度

识别准确度（Accuracy）定义如下。

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \times 100\% \tag{5-15}$$

式中各项说明如下。

- TP（True Positives）：正确的肯定表示正确肯定的分类数。
- TN（True Negatives）：正确的否定表示正确否定的分类数。
- FP（False Positives）：错误的肯定表示错误肯定的分类数。
- FN（False Negatives）：错误的否定表示错误否定的分类数。

（8）识别精确率

识别精确率（Precision）定义如下。

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{5-16}$$

（9）反馈率

反馈率（Recall）定义如下。

$$Recall = \frac{TP}{TP + TN} \times 100\% \tag{5-17}$$

（10）ROC 曲线

受试者工作特性（Receiver Operating Characteristic，ROC）曲线是一种非常有效的模型评价方法，可为选定临界值给出定量提示。将灵敏度（Sensitivity）设在纵轴，1- 特异性（1-Specificity）设在横轴，就可得出 ROC 曲线图。该曲线下的积分面积（Area）大小与每种方法优劣密切相关，反映分类器正确分类的统计概率，其值越接近 1 说明该算法效果越好。

（11）混淆矩阵

混淆矩阵（Confusion Matrix）是模式识别领域中一种常用的表达形式。它描绘样本数据的真实属性与识别结果类型之间的关系，是评价分类器性能的一种常用方法。假设对于 N 类模式的分类任务，识别数据集 D 包括 T_0 个样本，每类模式分别含有 T_i 个数据（$i = 1, \cdots, N$）。采用某种识别算法构造分类器 C，cm_{ij} 表示第 i 类模式被分类器 C 判断成第 j 类模式的数据占第 i 类模式样本总数的百分率，则可得到 $N \times N$ 维混淆矩阵 $CM(C, D)$。

$$CM(C, D) = \begin{pmatrix} cm_{11} & cm_{22} & \cdots & cm_{1i} & \cdots & cm_{1N} \\ cm_{21} & cm_{22} & \cdots & cm_{2i} & \cdots & cm_{2N} \\ & & \cdots & & \cdots & \\ cm_{i1} & cm_{i2} & \cdots & cm_{ii} & \cdots & cm_{iN} \\ & & \cdots & & \cdots & \\ cm_{N1} & cm_{N2} & \cdots & cm_{Ni} & \cdots & cm_{NN} \end{pmatrix} \tag{5-18}$$

混淆矩阵中元素的行下标对应目标的真实属性，列下标对应分类器产生的识别属性。对角线元素表示各模式能够被分类器 C 正确识别的百分率，而非对角线元素则表示发生错误判断的百分率。

通过混淆矩阵，可以获得分类器的正确识别率和错误识别率。

各模式正确识别率：

$$R_i = cm_{ii}, \qquad i = 1, \cdots, N \tag{5-19}$$

平均正确识别率：

$$R_A = \sum_{i=1}^{N} (cm_{ii} \cdot T_i) / T_0 \tag{5-20}$$

各模式错误识别率：

$$W_i = \sum_{j=1, j \neq i}^{N} cm_{ij} = 1 - cm_{ii} = 1 - R_i \tag{5-21}$$

平均错误识别率：

$$W_A = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (cm_{ii} \cdot T_i)/T_0 = 1 - R_A \tag{5-22}$$

对于一个二分类预测模型，分类结束后的混淆矩阵如表 5-8 所示。

表5-8 混淆矩阵

混淆矩阵表		预测类	
		类 =1	类 =0
实际类	类 =1	A	B
	类 =0	C	D

如有 150 个样本数据，这些数据分成 3 类，每类 50 个。分类结束后得到的混淆矩阵如下。

43	5	2
2	45	3
0	1	49

第 1 行的数据说明有 43 个样本正确分类，有 5 个样本应该属于第 1 类，却错误分到了第二类，有 2 个样本应属于第一类，却错误地分到第三类。

5.1.7 Python 分类预测模型特点

首先总结一下常见的分类 / 预测模型，见表 5-9。这些模型的使用方法大同小异，因此不再赘述，请读者参考本书相应的例子，以及对应的官方帮助文档。

表5-9 常见的模型评价和在Python中的实现

模 型	模 型 特 点	位 于
逻辑回归	比较基础的线性分类模型，很多时候是简单有效的选择	sklearn. linear_model
SVM	强大的模型，可以用来回归、预测、分类等，而根据选取不同的核函数。模型可以是线性的 / 非线性的	sklearn.svm
决策树	基于“分类讨论、逐步细化”思想的分类模型，模型直观，易解释，如前面 5.1.4 节中可以直接给出决策图	sklearn.tree
随机森林	思想跟决策树类似，精度通常比决策树要高，缺点是由于其随机性，丧失了决策树的可解释性	sklearn.ensemble
朴素贝叶斯	基于概率思想的简单有效的分类模型，能够给出容易理解的概率解释	sklearn.naive_bayes
神经网络	具有强大的拟合能力，可以用于拟合、分类等，它有很多个增强版本，如递神经网络、卷积神经网络、自编码器等，这些是深度学习的模型基础	Keras

经过对前面章节中的分类与预测的学习，我们应该基本认识到了 Python 建模的特点。首先，我们需要认识到：Python 本身是一门面向对象的编程语言，这就意味着很多 Python 的

程序是面向对象的。放到建模之中，我们就会发现，不管是在 Scikit-Learn 还是 Keras 中，建模的第一个步骤是建立一个对象，这个对象是空白的，需要进一步训练的，然后我们要设置模型的参数，接着就是通过 fit() 方法对模型进行训练，最后通过 predict() 方法预测结果。当然，还有一些方法有助于我们完成对模型的评估，如 score() 等。

Scikit-Learn 和 Keras 的功能都非常强大，我们能够做的仅仅是通过一些简单的例子介绍它们的基本功能，这仅仅是它们本身功能的冰山一角。因此，我们再次强调，如果遇到书本上没有讲解过的问题，应当尽可能地查阅官方的帮助文档。只有官方的帮助文章，才有可能全面地提供解决问题的答案。

5.2 聚类分析

餐饮企业经常会碰到这样的问题。

1）如何通过对餐饮客户消费行为的测量，进一步评判餐饮客户的价值和对餐饮客户进行细分，找到有价值的客户群和需关注的客户群？

2）如何合理对菜品进行分析，以便区分哪些菜品畅销毛利又高，哪些菜品滞销毛利又低？餐饮企业遇到的这些问题，可以通过聚类分析解决。

5.2.1 常用聚类分析算法

与分类不同，聚类分析是在没有给定划分类别的情况下，根据数据相似度进行样本分组的一种方法。与分类模型需要使用有类标记样本构成的训练数据不同，聚类模型可以建立在无类标记的数据上，是一种非监督的学习算法。聚类的输入是一组未被标记的样本，聚类根据数据自身的距离或相似度将其划分为若干组，划分的原则是组内距离最小化而组间（外部）距离最大化，如图 5-11 所示。

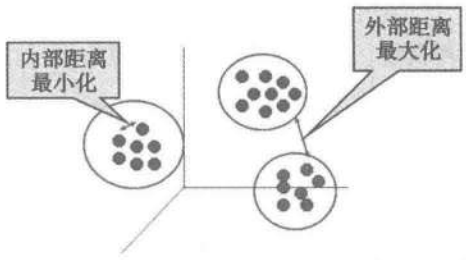


图 5-11 聚类分析建模原理

常用聚类方法见表 5-10。

表5-10 常用聚类方法

类 别	包括的主要算法
划分（分裂）方法	K-Means 算法（K- 平均）、K-MEDOIDS 算法（K- 中心点）、CLARANS 算法（基于选择的算法）
层次分析方法	BIRCH 算法（平衡迭代规约和聚类）、CURE 算法（代表点聚类）、CHAMELEON 算法（动态模型）
基于密度的方法	DBSCAN 算法（基于高密度连接区域）、DENCLUE 算法（密度分布函数）、OPTICS 算法（对象排序识别）

（续）

类　别	包括的主要算法
基于网格的方法	STING 算法（统计信息网络）、CLIQUE 算法（聚类高维空间）、WAVE-CLUSTER 算法（小波变换）
基于模型的方法	统计学方法、神经网络方法

常用聚类算法见表 5-11。

表5-11 常用聚类分析算法

算法名称	算法描述
K-Means	K- 均值聚类也称为快速聚类法，在最小化误差函数的基础上将数据划分为预定的类数 K。该算法原理简单并便于处理大量数据
K- 中心点	K- 均值算法对孤立点的敏感性，K- 中心点算法不采用簇中对象的平均值作为簇中心，而选用簇中离平均值最近的对象作为簇中心
系统聚类	系统聚类也称为多层次聚类，分类的单位由高到低呈树形结构，且所处的位置越低，其所包含的对象就越少，但这些对象间的共同特征越多。该聚类方法只适合在小数据量的时候使用，数据量大的时候速度会非常慢

5.2.2 K-Means 聚类算法

K-Means 算法[11]是典型的基于距离的非层次聚类算法，在最小化误差函数的基础上将数据划分为预定的类数 K，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。

1. 算法过程

1）从 N 个样本数据中随机选取 K 个对象作为初始的聚类中心。

2）分别计算每个样本到各个聚类中心的距离，将对象分配到距离最近的聚类中。

3）所有对象分配完成后，重新计算 K 个聚类的中心。

4）与前一次计算得到的 K 个聚类中心比较，如果聚类中心发生变化，转过程 2），否则转过程 5）。

5）当质心不发生变化时停止并输出聚类结果。

聚类的结果可能依赖于初始聚类中心的随机选择，可能使得结果严重偏离全局最优分类。实践中，为了得到较好的结果，通常选择不同的初始聚类中心，多次运行 K-Means 算法。在所有对象分配完成后，重新计算 K 个聚类的中心时，对于连续数据，聚类中心取该簇的均值，但是当样本的某些属性是分类变量时，均值可能无定义，可以使用 K- 众数方法。

2. 数据类型与相似性的度量

（1）连续属性

对于连续属性，要先对各属性值进行零 – 均值规范，再进行距离的计算。在 K-Means 聚类算法中，一般需要度量样本之间的距离、样本与簇之间的距离以及簇与簇之间的距离。

度量样本之间的相似性最常用的是欧几里得距离、曼哈顿距离和闵可夫斯基距离；样本与簇之间的距离可以用样本到簇中心的距离 $d(e_i, x)$；簇与簇之间的距离可以用簇中心的距离 $d(e_i, e_j)$。

用 p 个属性来表示 n 个样本的数据矩阵如下。

$$\begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

欧几里得距离：

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \tag{5-23}$$

曼哈顿距离：

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \tag{5-24}$$

闵可夫斯基距离：

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|)^q + (|x_{i2} - x_{j2}|)^q + \cdots + (|x_{ip} - x_{jp}|)^q} \tag{5-25}$$

q 为正整数，$q = 1$ 时即为曼哈顿距离；$q = 2$ 时即为欧几里得距离。

（2）文档数据

对于文档数据使用余弦相似性度量，先将文档数据整理成文档 - 词矩阵格式，见表 5-12。

表5-12 文档-词矩阵

词 / 文档	lost	win	team	score	music	happy	sad	…	coach
文档一	14	2	8	0	8	7	10	…	6
文档二	1	13	3	4	1	16	4	…	7
文档三	9	6	7	7	3	14	8	…	5

两个文档之间的相似度的计算公式为：

$$d(i,j) = \cos(i,j) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i}||\vec{j}|} \tag{5-26}$$

3. 目标函数

使用误差平方和 SSE 作为度量聚类质量的目标函数，对于两种不同的聚类结果，选择误差平方和较小的分类结果。

连续属性的 SSE 计算公式为：

$$SSE = \sum_{i=1}^{K} \sum_{x \in E_i} dist(e_i, x)^2 \tag{5-27}$$

文档数据的 SSE 计算公式为：

$$SSE = \sum_{i=1}^{K} \sum_{x \in E_i} \cos(e_i, x)^2 \tag{5-28}$$

簇 E_i 的聚类中心 e_i 计算公式为：

$$e_i = \frac{1}{n_i} \sum_{x \in E_i} x \tag{5-29}$$

表 5-13 为上述公式的符号说明。

表5-13 符号表

符 号	含 义	符 号	含 义
K	聚类簇的个数	e_i	簇 E_i 的聚类中心
E_i	第 i 个簇	n_i	第 i 个簇中样本的个数
x	对象（样本）		

下面结合具体案例来实现本节开始提出问题。

部分餐饮客户的消费行为特征数据见表 5-14。根据这些数据将客户分类成不同客户群，并评价这些客户群的价值。

表5-14 消费行为特征数据

ID	R（最近一次消费时间间隔）	F（消费频率）	M（消费总金额）
1	37	4	579
2	35	3	616
3	25	10	394
4	52	2	111
5	36	7	521
6	41	5	225
7	56	3	118
8	37	5	793
9	54	2	111
10	5	18	1086

采用 K-Means 聚类算法，设定聚类个数 K 为 3，最大迭代次数为 500 次，距离函数取欧氏距离。

K-Means 聚类算法的 Python 代码如代码清单 5-4 所示。

代码清单5-4 K-Means聚类算法代码

```
#-*- coding: utf-8 -*-
#使用K-Means算法聚类消费行为特征数据

import pandas as pd

#参数初始化
inputfile = '../data/consumption_data.xls' #销量及其他属性数据
outputfile = '../tmp/data_type.xls' #保存结果的文件名
k = 3 #聚类的类别
iteration = 500 #聚类最大循环次数
data = pd.read_excel(inputfile, index_col = 'Id') #读取数据
data_zs = 1.0*(data - data.mean())/data.std() #数据标准化
```

```
from sklearn.cluster import KMeans
model = KMeans(n_clusters = k, n_jobs = 4, max_iter = iteration) #分为k类，并发数4
model.fit(data_zs) #开始聚类

#简单打印结果
r1 = pd.Series(model.labels_).value_counts() #统计各个类别的数目
r2 = pd.DataFrame(model.cluster_centers_) #找出聚类中心
r = pd.concat([r2, r1], axis = 1) #横向连接（0是纵向），得到聚类中心对应的类别下的数目
r.columns = list(data.columns) + [u'类别数目'] #重命名表头
print(r)

#详细输出原始数据及其类别
r = pd.concat([data, pd.Series(model.labels_, index = data.index)], axis = 1)  #详细
   输出每个样本对应的类别
r.columns = list(data.columns) + [u'聚类类别'] #重命名表头
r.to_excel(outputfile) #保存结果
```

代码详见：示例程序 /code/k_means.py

注意 事实上，Scikit-Learn 中的 K-Means 算法仅仅支持欧氏距离，原因在于采用其他的距离并不一定能够保证算法的收敛性。

执行 K-Means 聚类算法输出的结果见表 5-15。

表5-15 聚类算法输出结果

分群类别		分群 1	分群 2	分群 3
样本个数		340	560	40
样本个数占比		12.77%	65.53%	21.70%
聚类中心	R	−0.162 950 92	−0.147 855 15	3.455 054 86
	F	1.116 721 77	−0.656 891 53	−0.295 653 57
	M	0.395 575 42	−0.272 251 03	0.449 123 42

接着用 Pandas 和 Matplotlib 绘制的不同客户分群的概率密度函数图，通过这些图能直观地比较不同客户群的价值，代码如下。

绘制聚类后的概率密度图（接代码清单5-4）

```
def density_plot(data, title): #自定义作图函数
  import matplotlib.pyplot as plt
  plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
  plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
  plt.figure()
  for i in range(len(data.iloc[0])): #逐列作图
    (data.iloc[:,i]).plot(kind='kde', label = data.columns[i], linewidth = 2)
  plt.ylabel(u'密度')
  plt.xlabel(u'人数')
  plt.title(u'聚类类别%s各属性的密度曲线' %title)
```

```
    plt.legend()
    return plt

def density_plot(data): #自定义作图函数
    import matplotlib.pyplot as plt
    plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
    plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
    p = data.plot(kind='kde', linewidth = 2, subplots = True, sharex = False)
    [p[i].set_ylabel(u'密度') for i in range(k)]
    plt.legend()
    return plt

pic_output = '../tmp/pd_' #概率密度图文件名前缀
for i in range(k):
    density_plot(data[r[u'聚类类别']==i]).savefig(u'%s%s.png' %(pic_output, i))
```

代码详见：示例程序 /code/k_means.py

分群的概率密度函数图如图 5-12 ～图 5-14 所示。

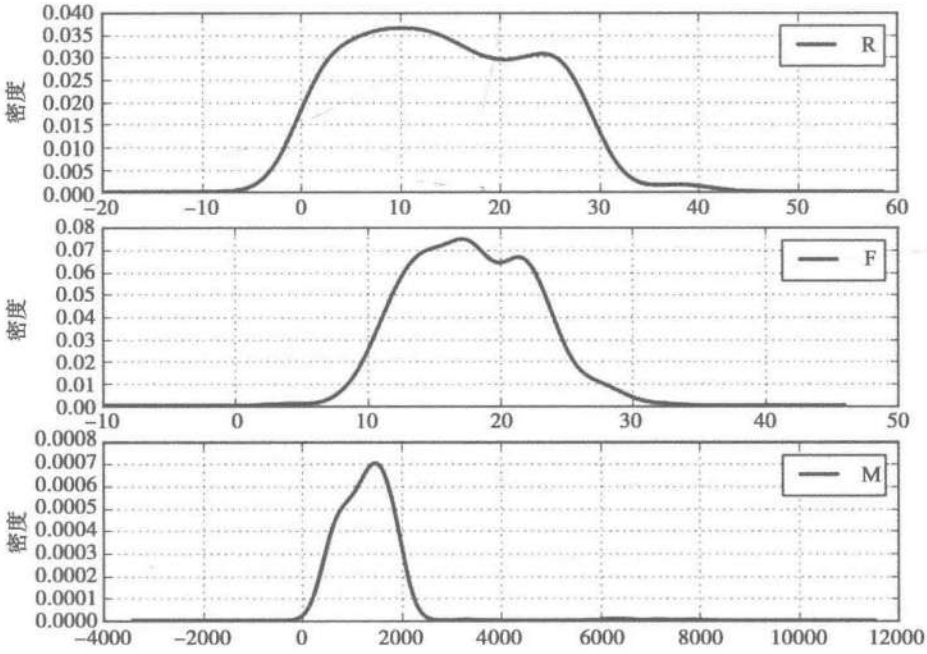


图 5-12 分群 1 的概率密度函数图

客户价值分析如下。

分群 1 特点：R 间隔相对较小，主要集中在 0 ～ 30 天；消费次数集中在 10 ～ 25 次；消费金额在 500 ～ 2000。

分群 2 特点：R 间隔分布在 0 ～ 30 天；消费次数集中在 0 ～ 12 次；消费金额在 0 ～ 1800。

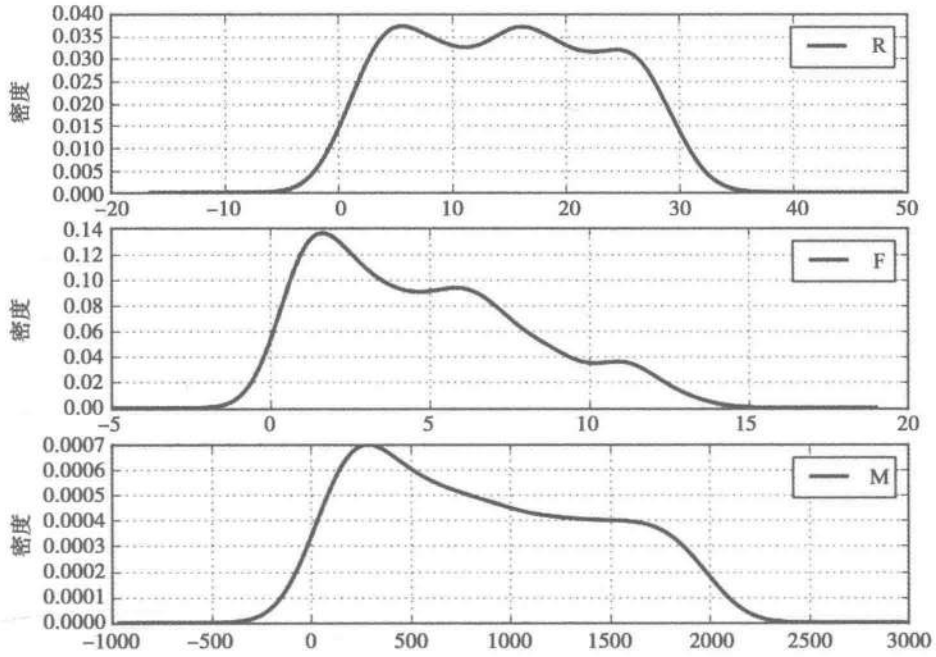


图 5-13　分群 2 的概率密度函数图

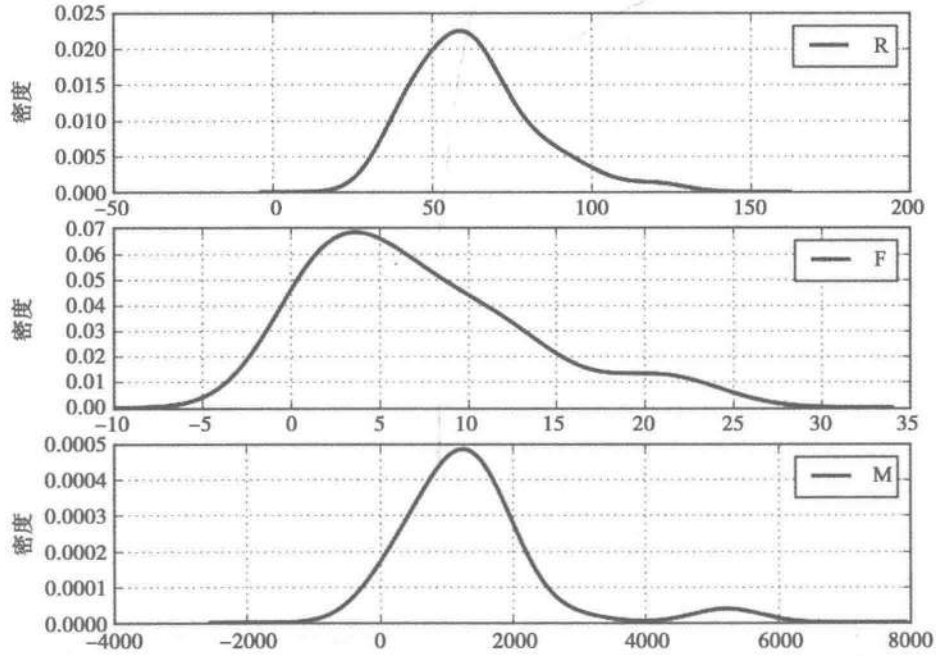


图 5-14　分群 3 的概率密度函数图

分群3特点：R间隔相对较大，间隔分布在30～80天；消费次数集中在0～15次；消费金额在0～2000。

对比分析：分群1时间间隔较短，消费次数多，而且消费金额较大，是高消费、高价值人群。分群2的时间间隔、消费次数和消费金额处于中等水平，代表着一般客户。分群3的时间间隔较长，消费次数较少，消费金额也不是特别高，是价值较低的客户群体。

5.2.3 聚类分析算法评价

聚类分析仅根据样本数据本身将样本分组。其目标是实现组内的对象相互之间是相似的（相关的），而不同组中的对象是不同的（不相关的）。组内的相似性越大，组间差别越大，聚类效果就越好。

（1）purity评价法

purity方法是极为简单的一种聚类评价方法，只需计算正确聚类数占总数的比例。

$$purity(X,Y) = \frac{1}{n}\sum_{k}\max_{i}|x_k \cap y_i| \tag{5-30}$$

其中，$x=(x_1, x_2,\cdots, x_k)$是聚类的集合。x_k表示第k个聚类的集合。$y=(y_1, y_2,\cdots, y_k)$表示需要被聚类的集合，y_i表示第i个聚类对象。n表示被聚类集合对象的总数。

（2）RI评价法

实际上，这是一种用排列组合原理来对聚类进行评价的手段，RI评价公式如下。

$$RI = \frac{R+W}{R+M+D+W} \tag{5-31}$$

其中，R是指被聚在一类的两个对象被正确分类了，W是指不应该被聚在一类的两个对象被正确分开了，M指不应该放在一类的对象被错误的放在了一类，D指不应该分开的对象被错误的分开了。

（3）F值评价法

这是基于上述RI方法衍生出的一个方法，F评价公式如下。

$$F_\alpha = \frac{(1+\alpha^2)pr}{\alpha^2 p + r} \tag{5-32}$$

其中，$p = \frac{R}{R+M}, r = \frac{R}{R+D}$。

实际上RI方法就是把准确率p和召回率r看得同等重要，事实上，有时候我们可能需要某一特性更多一点，这时候就适合使用F值方法。

5.2.4 Python主要聚类分析算法

Python的聚类相关的算法主要在Scikit-Learn中，Python里面实现的聚类主要包括K-Means聚类、层次聚类、FCM以及神经网络聚类，其主要相关函数如表5-16所示。

表5-16 聚类主要函数列表

对 象 名	函 数 功 能	所属工具箱
KMeans	K 均值聚类	sklearn.cluster
AffinityPropagation	吸引力传播聚类，2007 年提出，几乎优于所有其他方法，不需要指定聚类数，但运行效率较低	sklearn.cluster
MeanShift	均值漂移聚类算法	sklearn.cluster
SpectralClustering	谱聚类，具有效果比 K 均值好，速度比 K 均值快等特点	sklearn.cluster
AgglomerativeClustering	层次聚类，给出一棵聚类层次树	sklearn.cluster
DBSCAN	具有噪声的基于密度的聚类方法	sklearn.cluster
BIRCH	综合的层次聚类算法，可以处理大规模数据的聚类	sklearn.cluster

这些不同模型的使用方法是大同小异的，基本都是先用对应的函数建立模型，然后用 .fit() 方法来训练模型，训练好之后，就可以用 .label_ 方法给出样本数据的标签，或者用 .predict() 方法预测新的输入的标签。

此外，Scipy 库也提供了一个聚类子库 scipy.cluster，里边提供了一些聚类算法，如层次聚类等，但没有 Scikit-Learn 那么完善和丰富。scipy.cluster 的好处是它的函数名和功能基本跟 Python 是一一对应的（Scpiy 致力于让 Python 称为 Python 般强大），如层次聚类的 linkage、dendrogram 等，因此已经熟悉 Python 的朋友，可以尝试使用 Scipy 提供的聚类库，在此就不详细介绍了。

下面介绍一个聚类结果可视化的工具——TSNE。

TSNE 是 Laurens van der Maaten 和 Geoffrey Hintton 在 2008 年提出的，它的定位是高维数据的可视化。我们总喜欢能够直观地展示研究结果，聚类也不例外。然而，通常来说输入的特征数是高维的（大于 3 维），一般难以直接以原特征对聚类结果进行展示。而 TSNE 提供了一种有效的数据降维方式，让我们可以在 2 维或者 3 维的空间中展示聚类结果。

下面我们用 TSNE 对上述 KMeans 聚类的结果以二维的方式展示出来。

代码清单5-5 用TSNE进行数据降维并展示聚类结果

```
#-*- coding: utf-8 -*-
#接k_means.py
from sklearn.manifold import TSNE

tsne = TSNE()
tsne.fit_transform(data_zs) #进行数据降维
tsne = pd.DataFrame(tsne.embedding_, index = data_zs.index) #转换数据格式

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

#不同类别用不同颜色和样式绘图
```

```
d = tsne[r[u'聚类类别'] == 0]
plt.plot(d[0], d[1], 'r.')
d = tsne[r[u'聚类类别'] == 1]
plt.plot(d[0], d[1], 'go')
d = tsne[r[u'聚类类别'] == 2]
plt.plot(d[0], d[1], 'b*')
plt.show()
```

代码详见：示例程序 /code/tsne.py

结果如图 5-15 所示。

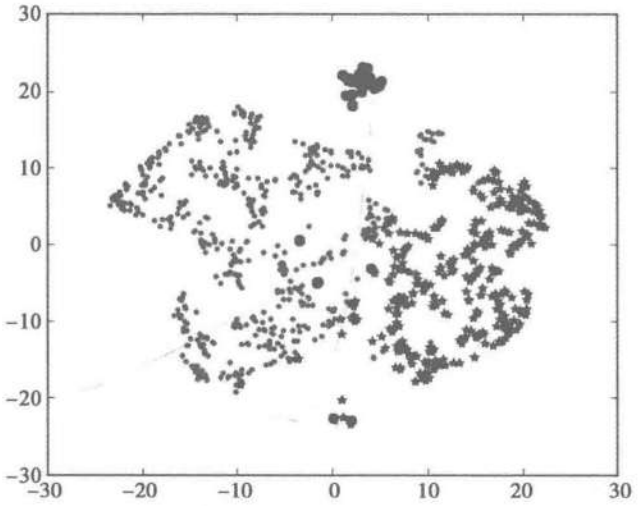


图 5-15 聚类效果图

5.3 关联规则

下面通过餐饮企业中的一个实际情景引出关联规则的概念。客户在餐厅点餐时，面对菜单中大量的菜品信息，往往无法迅速找到满意的菜品，既增加了点菜的时间，也降低了客户的就餐体验。实际上，菜品的合理搭配是有规律可循的：顾客的饮食习惯、菜品的荤素和口味，有些菜品之间是相互关联的，而有些菜品之间是对立或竞争关系（负关联），这些规律都隐藏在大量的历史菜单数据中，如果能够通过数据挖掘发现客户点餐的规则，就可以快速识别客户的口味，当他下了某个菜品的订单时推荐相关联的菜品，引导客户消费，提高顾客的就餐体验和餐饮企业的业绩水平。

关联规则分析也成为购物篮分析，最早是为了发现超市销售数据库中不同的商品之间的关联关系。例如，一个超市的经理想要更多地了解顾客的购物习惯，比如“哪组商品可能会在一次购物中同时购买？”或者“某顾客购买了个人电脑，那该顾客三个月后购买数码相机的概率有多大？”他可能会发现如果购买了面包的顾客同时非常有可能会购买牛奶，这就导

出了一条关联规则“面包 => 牛奶”，其中面包称为规则的前项，而牛奶称为后项。通过对面包降低售价进行促销，而适当提高牛奶的售价，关联销售出的牛奶就有可能增加超市整体的利润。

关联规则分析是数据挖掘中最活跃的研究方法之一，目的是在一个数据集中找出各项之间的关联关系，而这种关系并没有在数据中直接表示出来。

5.3.1 常用关联规则算法

常用关联算法如表 5-17 所示。

表5-17 常用关联规则算法

算法名称	算法描述
Apriori	关联规则最常用也是最经典的挖掘频繁项集的算法，其核心思想是通过连接产生候选项及其支持度然后通过剪枝生成频繁项集
FP-Tree	针对 Apriori 算法的固有的多次扫描事务数据集的缺陷，提出的不产生候选频繁项集的方法。Apriori 和 FP-Tree 都是寻找频繁项集的算法
Eclat 算法	Eclat 算法是一种深度优先算法，采用垂直数据表示形式，在概念格理论的基础上利用基于前缀的等价关系将搜索空间划分为较小的子空间
灰色关联法	分析和确定各因素之间的影响程度或是若干个子因素（子序列）对主因素（母序列）的贡献度而进行的一种分析方法

本节将详细介绍 Apriori 算法。

5.3.2 Apriori 算法

以超市销售数据为例，提取关联规则的最大困难在于当存在很多商品时，可能的商品的组合（规则的前项与后项）的数目会达到一种令人望而却步的程度。因而各种关联规则分析的算法从不同方面入手，以减小可能的搜索空间的大小以及减小扫描数据的次数。Apriori[12] 算法是最经典的挖掘频繁项集的算法，第一次实现了在大数据集上可行的关联规则提取，其核心思想是通过连接产生候选项与其支持度，然后通过剪枝生成频繁项集。

1. 关联规则和频繁项集

（1）关联规则的一般形式

项集 A、B 同时发生的概率称为关联规则的支持度（也称相对支持度）。

$$Support(A \Rightarrow B) = P(A \cup B) \tag{5-33}$$

项集 A 发生，则项集 B 发生的概率为关联规则的置信度。

$$Confidence(A \Rightarrow B) = P(B|A) \tag{5-34}$$

（2）最小支持度和最小置信度

最小支持度是用户或专家定义的衡量支持度的一个阈值，表示项目集在统计意义上的最低重要性；最小置信度是用户或专家定义的衡量置信度的一个阈值，表示关联规则的最低可

靠性。同时满足最小支持度阈值和最小置信度阈值的规则称作强规则。

（3）项集

项集是项的集合。包含 k 个项的项集称为 k 项集，如集合{牛奶，麦片，糖}是一个3项集。

项集的出现频率是所有包含项集的事务计数，又称作绝对支持度或支持度计数。如果项集I的相对支持度满足预定义的最小支持度阈值，则 I 是频繁项集。频繁 k 项集通常记作 k。

（4）支持度计数

项集A的支持度计数是事务数据集中包含项集A的事务个数，简称为项集的频率或计数。

已知项集的支持度计数，则规则 $A \Rightarrow B$ 的支持度和置信度很容易从所有事务计数、项集A和项集 $A \cup B$ 的支持度计数推出。

$$Support(A \Rightarrow B) = \frac{A,B\text{同时发生的事务个数}}{\text{所有事务个数}} = \frac{Support\_count(A \cap B)}{Total\_count(A)} \quad (5\text{-}35)$$

$$Confidence(A \Rightarrow B) = P(A \mid B) = \frac{Support(A \cap B)}{Support(A)} = \frac{Support\_count(A \cap B)}{Support\_count(A)} \quad (5\text{-}36)$$

也就是说，一旦得到所有事务个数，A，B和 $A \cap B$ 的支持度计数，就可以导出对应的关联规则 $A \Rightarrow B$ 和 $B \Rightarrow A$，并可以检查该规则是否是强规则。

在Python中实现上述Apriori算法的代码如代码清单5-6所示。其中，我们自行编写了Apriori算法的函数apriori.py，读者有需要的时候可以直接使用，此外可以参考代码读懂实现过程。

代码清单5-6 Apriori算法调用代码

```
#-*- coding: utf-8 -*-
#使用Apriori算法挖掘菜品订单关联规则
from __future__ import print_function
import pandas as pd
from apriori import * #导入自行编写的apriori函数

inputfile = '../data/menu_orders.xls'
outputfile = '../tmp/apriori_rules.xls' #结果文件
data = pd.read_excel(inputfile, header = None)

print(u'\n转换原始数据至0-1矩阵...')
ct = lambda x : pd.Series(1, index = x[pd.notnull(x)]) #转换0-1矩阵的过渡函数
b = map(ct, data.as_matrix()) #用map方式执行
data = pd.DataFrame(list(b)).fillna(0) #实现矩阵转换，空值用0填充
print(u'\n转换完毕。')
del b #删除中间变量b，节省内存

support = 0.2 #最小支持度
confidence = 0.5 #最小置信度
ms = '---' #连接符，默认'--'，用来区分不同元素，如A--B。需要保证原始表格中不含有该字符
```

```
find_rule(data, support, confidence, ms).to_excel(outputfile) #保存结果
```

代码详见：示例程序 /code/cal_apriori.py

运行结果如下。

```
           support  confidence
e---a          0.3    1.000000
e---c          0.3    1.000000
c---e---a      0.3    1.000000
a---e---c      0.3    1.000000
a---b          0.5    0.714286
c---a          0.5    0.714286
a---c          0.5    0.714286
c---b          0.5    0.714286
b---a          0.5    0.625000
b---c          0.5    0.625000
b---c---a      0.3    0.600000
a---c---b      0.3    0.600000
a---b---c      0.3    0.600000
a---c---e      0.3    0.600000
```

其中，e---a 表示 e 发生能够推出 a 发生，置信度为 100%，支持度为 30%；b---c---a 表示 b、c 同时发生时能够推出 a 发生，置信度为 60%，支持度为 30% 等。搜索出来的关联规则不一定具有实际意义，需要根据问题背景筛选适当的有意义的规则，并赋予合理的解释。

2. Ariori 算法：使用候选产生频繁项集

Apriori 算法的主要思想是找出存在于事务数据集中的最大的频繁项集，在利用得到的最大频繁项集与预先设定的最小置信度阈值生成强关联规则。

（1）Apriori 的性质

频繁项集的所有非空子集也必须是频繁项集。根据该性质可以得出：向不是频繁项集 I 的项集中添加事务 A，新的项集 $I \cup A$ 一定也不是频繁项集。

（2）Apriori 算法实现的两个过程如下。

1）找出所有的频繁项集（支持度必须大于等于给定的最小支持度阈值），在这个过程中连接步和剪枝步互相融合，最终得到最大频繁项集 L_k。

连接步：

连接步的目的是找到 K 项集。对给定的最小支持度阈值，分别对 1 项候选集 C_1，剔除小于该阈值的项集得到 1 项频繁集 L_1；下一步由 L_1 自身连接产生 2 项候选集 C_2，保留 C_2 中满足约束条件的项集得到 2 项频繁集，记为 L_2；再下一步由 L_2 与 L_3 连接产生 3 项候选集 C_3，保留 C_2 中满足约束条件的项集得到 3 项频繁集，记为 L_3……这样循环下去，得到最大频繁项集 L_k。

剪枝步：

剪枝步紧接着连接步，在产生候选项 C_k 的过程中起到减小搜索空间的目的。由于 C_k 是

L_{k-1} 与 L_1 连接产生的，根据 Apriori 的性质频繁项集的所有非空子集也必须是频繁项集，所以不满足该性质的项集不会存在于 C_k 中，该过程就是剪枝。

2）由频繁项集产生强关联规则：由过程 1）可知未超过预定的最小支持度阈值的项集已被剔除，如果剩下这些规则又满足了预定的最小置信度阈值，那么就挖掘出了强关联规则。

下面将结合餐饮行业的实例来讲解 Apriori 关联规则算法挖掘的实现过程。数据库中部分点餐数据见表 5-18。

表5-18 数据库中部分点餐数据

序 列	时 间	订 单 号	菜品 id	菜 品 名 称
1	2014/8/21	101	18491	健康麦香包
2	2014/8/21	101	8693	香煎葱油饼
3	2014/8/21	101	8705	翡翠蒸香茜饺
4	2014/8/21	102	8842	菜心粒咸骨粥
5	2014/8/21	102	7794	养颜红枣糕
6	2014/8/21	103	8842	金丝燕麦包
7	2014/8/21	103	8693	三丝炒河粉
…	…	…	…	…

将表 5-18 中的事务数据（一种特殊类型的记录数据）整理成关联规则模型所需的数据结构，从中抽取 10 个点餐订单作为事务数据集，设支持度为 0.2（支持度计数为 2），为方便起见将菜品 {18491，8842，8693，7794，8705} 分别简记为 {a，b，c，d，e}，见表 5-19。

表5-19 某餐厅事务数据集

订 单 号	菜品 id	菜品 id	订 单 号	菜品 id	菜品 id
1	18491, 8693，8705	a，c，e	6	8842，8693	b，c
2	8842,7794	b，d	7	18491，8842	a，b
3	8842，8693	b，c	8	18491，8842,8693,8705	a，b，c，e
4	18491，8842，8693，7794	a，b，c，d	9	18491，8842,8693	a，b，c
5	18491，8842	a，b	10	18491，8693，8705	a，c，e

算法过程如图 5-16 所示。

过程一：找最大 k 项频繁集

1）算法简单扫描所有的事务，事务中的每一项都是候选 1 项集的集合 C_1 的成员，计算每一项的支持度。例如，$P(\{a\}) = \dfrac{\text{项集}\{a\}\text{ 的支持度计数}}{\text{所有事务个数}} = \dfrac{7}{10} = 0.7$。

2）对 C_1 中各项集的支持度与预先设定的最小支持度阈值进行比较，保留大于或等于该阈值的项，得 1 项频繁集 L_1。

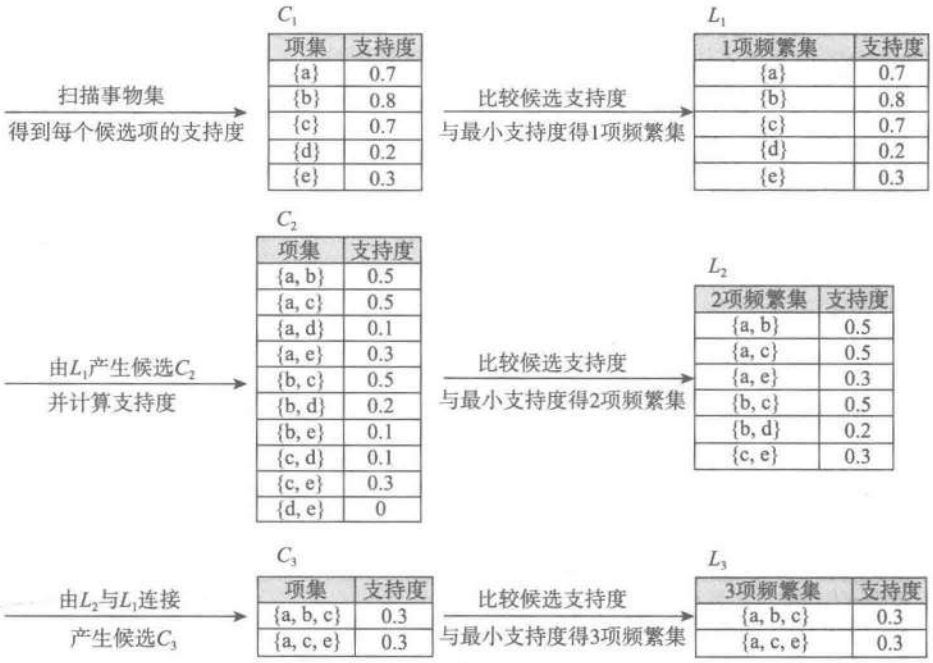


图 5-16 Apriori 算法实现过程

3）扫描所有事务，L_1 与 L_1 连接得候选 2 项集 C_2，并计算每一项的支持度。如 $P(\{a, b\}) = \frac{\text{项集}\{a,b\}\text{ 的支持度计数}}{\text{所有事务个数}} = \frac{5}{10} = 0.5$。接下来是剪枝步，由于 C_2 的每个子集（即 L_1）都是频繁集，所以没有项集从 C_2 中剔除。

4）对 C_2 中各项集的支持度与预先设定的最小支持度阈值进行比较，保留大于或等于该阈值的项，得 2 项频繁集 L_2。

5）扫描所有事务，L_2 与 L_1 连接得候选 3 项集 C_3，并计算每一项的支持度，如 $P(\{a, b, c\}) = \frac{\text{项集}\{a,b,c\}\text{ 的支持度计数}}{\text{所有事务个数}} = \frac{3}{10} = 0.3$。接下来是剪枝步，$L_2$ 与 L_1 连接的所有项集为：{a，b，c}，{a，b，d}，{a，b，e}，{a，c，d}，{a，c，e}，{b，c，d}，{b，c，e}，根据 Apriori 算法，频繁集的所有非空子集也必须是频繁集，因为 {b，d}，{b，e}，{c，d} 不包含在 b 项频繁集 L_2 中，即不是频繁集，应剔除，最后的 C_3 中的项集只有 {a，b，c} 和 {a，c，e}；

6）对 C_3 中各项集的支持度与预先设定的最小支持度阈值进行比较，保留大于或等于该阈值的项，得 3 项频繁集 L_3。

7）L_3 与 L_1 连接得候选 4 项集 C_4，易得剪枝后为空集。最后得到最大 3 项频繁集 {a，b，c} 和 {a，c，e}。

由以上过程可知 L_1，L_2，L_3 都是频繁项集，L_3 是最大频繁项集。

过程二：由频繁集产生关联规则

置信度的计算公式如下。

$$Confidence(A \Rightarrow B) = P(A \mid B) = \frac{Support(A \cap B)}{Support(A)} = \frac{Support\_count(A \cap B)}{Support\_count(A)}$$

其中，*Support_count*($A \cap B$) 是包含项集 $A \cap B$ 的事务数，*Support_count*(A) 是包含项集 A 的事务数，根据该公式，尝试基于该例产生关联规则。

Python 程序输出的关联规则如下。

Rule	(Support, Confidence)
$a \rightarrow b$	(50%, 71.428 6%)
$b \rightarrow a$	(50%, 62.5%)
$a \rightarrow c$	(50%, 71.428 6%)
$c \rightarrow a$	(30%, 71.428 6%)
$b \rightarrow c$	(50%, 62.5%)
$c \rightarrow b$	(50%, 71.428 6%)
$e \rightarrow a$	(30%, 100%)
$e \rightarrow c$	(30%, 100%)
$a, b \rightarrow c$	(30%, 60%)
$a, c \rightarrow b$	(30%, 60%)
$b, c \rightarrow a$	(30%, 60%)
$e \rightarrow a, c$	(30%, 100%)
$a, c \rightarrow e$	(30%, 60%)
$a, e \rightarrow c$	(30%, 100%)
$c, e \rightarrow a$	(30%, 100%)
$d \rightarrow b$	(20%, 100%)

就第一条输出结果进行解释：客户同时点菜品 a 和 b 的概率是 50%，点了菜品 a，再点菜品 b 的概率是 71.428 6%。知道了这些，就可以对顾客进行智能推荐，增加销量同时满足客户需求。

5.4 时序模式

就餐饮企业而言，经常会碰到如下问题。

由于餐饮行业是生产和销售同时进行的，因此销售预测对于餐饮企业十分必要。如何基于菜品历史销售数据，做好餐饮销售预测，以便减少菜品脱销现象和避免因备料不足而造成的生产延误，从而减少菜品生产等待时间，提供给客户更优质的服务，同时可以减少安全库存量，做到生产准时制，降低物流成本。

餐饮销售预测可以看作是基于时间序列的短期数据预测，预测对象为具体菜品销售量。

常用按时间顺序排列的一组随机变量 $X_1, X_2, \cdots, X_t$ 来表示一个随机事件的时间序列，简记

为 $\{X_t\}$；用 $x_1, x_2, \cdots, x_n$ 或 $\{x_t, t = 1, 2, \cdots, n\}$ 表示该随机序列的 n 个有序观察值，称之为序列长度为 n 的观察值序列。

本节应用时间序列分析[13]的目的就是给定一个已被观测了的时间序列，预测该序列的未来值。

5.4.1 时间序列算法

常用的时间序列模型见表 5-20。

表5-20 常用时间序列模型

模型名称	描 述
平滑法	平滑法常用于趋势分析和预测，利用修匀技术，削弱短期随机波动对序列的影响，使序列平滑化。根据所用平滑技术的不同，可具体分为移动平均法和指数平滑法
趋势拟合法	趋势拟合法把时间作为自变量，相应的序列观察值作为因变量，建立回归模型。根据序列的特征，可具体分为线性拟合和曲线拟合
组合模型	时间序列的变化主要受到长期趋势（T）、季节变动（S）、周期变动（C）和不规则变动（ε）这 4 个因素的影响。根据序列的特点，可以构建加法模型和乘法模型 加法模型：$x_t = T_t + S_t + C_t + \varepsilon_t$ 乘法模型：$x_t = T_t \times S_t \times C_t \times \varepsilon_t$
AR 模型	$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t$ 以前 p 期的序列值 $x_{t-1}, x_{t-2}, \cdots, x_{t-p}$ 为自变量、随机变量 X_t 的取值 x_t 为因变量建立线性回归模型
MA 模型	$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$ 随机变量 X_t 的取值 x_t 与以前各期的序列值无关，建立 x_t 与前 q 期的随机扰动 $\varepsilon_{t-1}, \varepsilon_{t-2}, \cdots, \varepsilon_{t-q}$ 的线性回归模型
ARMA 模型	$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$ 随机变量 X_t 的取值 x_t 不仅与以前 p 期的序列值有关，还与前 q 期的随机扰动有关
ARIMA 模型	许多非平稳序列差分后会显示出平稳序列的性质，称这个非平稳序列为差分平稳序列。对差分平稳序列可以使用 ARIMA 模型进行拟合
ARCH 模型	ARCH 模型能准确地模拟时间序列变量的波动性的变化，适用于序列具有异方差性并且异方差函数短期自相关
GARCH 模型及其衍生模型	GARCH 模型称为广义 ARCH 模型，是 ARCH 模型的拓展。相比于 ARCH 模型，GARCH 模型及其衍生模型更能反映实际序列中的长期记忆性、信息的非对称性等性质

本节将重点介绍 AR 模型、MA 模型、ARMA 模型和 ARIMA 模型。

5.4.2 时间序列的预处理

拿到一个观察值序列后，首先要对它的纯随机性和平稳性进行检验，这两个重要的检验称为序列的预处理。根据检验结果可以将序列分为不同的类型，对不同类型的序列会采取不同的分析方法。

对于纯随机序列，又称为白噪声序列，序列的各项之间没有任何相关关系，序列在进行完全无序的随机波动，可以终止对该序列的分析。白噪声序列是没有信息可提取的平稳序列。

对于平稳非白噪声序列，它的均值和方差是常数，现已有一套非常成熟的平稳序列的建模方法。通常是建立一个线性模型来拟合该序列的发展，借此提取该序列的有用信息。ARMA 模型是最常用的平稳序列拟合模型。

对于非平稳序列，由于它的均值和方差不稳定，处理方法一般是将其转变为平稳序列，这样就可以应用有关平稳时间序列的分析方法，如建立 ARMA 模型来进行相应的研究。如果一个时间序列经差分运算后具有平稳性，则该序列为差分平稳序列，可以使用 ARIMA 模型进行分析。

1. 平稳性检验

（1）平稳时间序列的定义

对于随机变量 X，可以计算其均值（数学期望）μ、方差 σ^2；对于两个随机变量量 X 和 Y，可以计算 X, Y 的协方差 $\mathrm{cov}(X, Y) = E[(X-\mu_X)(Y-\mu_Y)]$ 和相关系数 $\rho(X,Y) = \dfrac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}$，它们度量了两个不同事件之间的相互影响程度。

对于时间序列 $\{X_t, t \in T\}$，任意时刻的序列值 X_t 都是一个随机变量，每一个随机变量都会有均值和方差，记 X_t 的均值为 μ_t，方差为 σ_t；任取 $t, s \in T$，定义序列 $\{X_t\}$ 的自协方差函数 $\gamma(t,s) = E[(X_t-\mu_t)(X_s-\mu_s)]$ 和自相关系数 $\rho(t,s) = \dfrac{\mathrm{cov}(X_t,X_s)}{\sigma_t\sigma_s}$（特别地，$\gamma(t,t) = \gamma(0) = 1, \rho_0 = 1$），之所以称它们为自协方差函数和自相关系数，是因为它们衡量的是同一个事件在两个不同时期（时刻 t 和 s）之间的相关程度，形象地讲就是度量自己过去的行为对自己现在的影响。

如果时间序列 $\{X_t, t \in T\}$ 在某一常数附近波动且波动范围有限，即有常数均值和常数方差，并且延迟 k 期的序列变量的自协方差和自相关系数是相等的或者说延迟 k 期的序列变量之间的影响程度是一样的，则称 $\{X_t, t \in T\}$ 为平稳序列。

（2）平稳性的检验

对序列的平稳性的检验有两种检验方法，一种是根据时序图和自相关图的特征做出判断的图检验，该方法操作简单、应用广泛，缺点是带有主观性；另一种是构造检验统计量进行检验的方法，目前最常用的方法是单位根检验。

1）时序图检验。根据平稳时间序列的均值和方差都为常数的性质，平稳序列的时序图显示该序列值始终在一个常数附近随机波动，而且波动的范围有界；如果有明显的趋势性或者周期性，那它通常不是平稳序列。

2）自相关图检验。平稳序列具有短期相关性，这个性质表明对平稳序列而言通常只有近期的序列值对现时值的影响比较明显，间隔越远的过去值对现时值的影响越小。随着延迟期数 k 的增加，平稳序列的自相关系数 ρ_k（延迟 k 期）会比较快的衰减趋向于零，并在零附

近随机波动，而非平稳序列的自相关系数衰减的速度比较慢，这就是利用自相关图进行平稳性检验的标准。

3）单位根检验。单位根检验是指检验序列中是否存在单位根，如果存在单位根就是非平稳时间序列了。

2. 纯随机性检验

如果一个序列是纯随机序列，那么它的序列值之间应该没有任何关系，即满足 $\gamma(k)=0$，$k\neq 0$ 这是一种理论上才会出现的理想状态，实际上纯随机序列的样本自相关系数不会绝对为零，但是很接近零，并在零附近随机波动。

纯随机性检验也称白噪声检验，一般是构造检验统计量来检验序列的纯随机性，常用的检验统计量有 Q 统计量、LB 统计量，由样本各延迟期数的自相关系数可以计算得到检验统计量，然后计算出对应的 p 值，如果 p 值显著大于显著性水平 α，则表示该序列不能拒绝纯随机的原假设，可以停止对该序列的分析。

5.4.3 平稳时间序列分析

ARMA 模型的全称是自回归移动平均模型，它是目前最常用的拟合平稳序列的模型。它又可以细分为 AR 模型、MA 模型和 ARMA 三大类。都可以看作是多元线性回归模型。

1. AR 模型

具有如下结构的模型称为 p 阶自回归模型，简记为 $AR(p)$。

$$x_t=\phi_0+\phi_1x_{t-1}+\phi_2x_{t-2}+\cdots+\phi_px_{t-p}+\varepsilon_t \tag{5-37}$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 p 期 $x_{t-1}, x_{t-2},\cdots, x_{t-p}$ 的多元线性回归，认为 x_t 主要是受过去 p 期的序列值的影响。误差项是当期的随机干扰 ε_t，为零均值白噪声序列。

平稳 AR 模型的性质见表 5-21。

表5-21 平稳AR模型的性质

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数（ACF）	拖尾
方差	常数方差	偏自相关系数（PACF）	p 阶截尾

（1）均值

对满足平稳性条件的 $AR(p)$ 模型的方程，两边取期望，得：

$$E(x_t)=E(\phi_0+\phi_1x_{t-1}+\phi_2x_{t-2}+\cdots+\phi_px_{t-p}+\varepsilon_t) \tag{5-38}$$

已知 $E(x_t)=\mu$，$E(\varepsilon_t)=0$，所以有 $\mu=\phi_0+\phi_1\mu+\phi_2\mu+\cdots+\phi_p\mu$，

解得：

$$\mu=\frac{\phi_0}{1-\phi_1-\phi_2-\cdots-\phi_p} \tag{5-39}$$

（2）方差

平稳 $AR(p)$ 模型的方差有界，等于常数。

（3）自相关系数（ACF）

平稳 $AR(p)$ 模型的自相关系数 $\rho_k = \rho(t, t-k) = \dfrac{\mathrm{cov}(X_t, X_{t-k})}{\sigma_t \sigma_{t-k}}$ 呈指数的速度衰减，始终有非零取值，不会在 k 大于某个常数之后就恒等于零，这个性质就是平稳 $AR(p)$ 模型的自相关系数 ρ_k 具有拖尾性。

（4）偏自相关系数（PACF）

对于一个平稳 $AR(p)$ 模型，求出延迟 k 期自相关系数 ρ_k 时，实际上的得到的并不是 X_t 与 X_{t-k} 之间单纯的相关关系，因为 X_t 同时还会受到中间 $k-1$ 个随机变量 $X_{t-1}, X_{t-2}, \cdots, X_{t-k}$ 的影响，所以自相关系数 ρ_k 里实际上掺杂了其他变量对 X_t 与 X_{t-k} 的相关影响，为了单纯地测度 X_{t-k} 对 X_t 的影响，引进偏自相关系数的概念。

可以证明平稳 $AR(p)$ 模型的偏自相关系数具有 p 阶截尾性。这个性质连同前面的自相关系数的拖尾性是 $AR(p)$ 模型重要的识别依据。

2. MA 模型

具有如下结构的模型称为 q 阶自回归模型，简记为 $MA(q)$。

$$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{5-40}$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 q 期的随机扰动 $\varepsilon_{t-1}, \varepsilon_{t-2}, \cdots, \varepsilon_{t-q}$ 的多元线性函数，误差项是当期的随机干扰 ε_t，为零均值白噪声序列，μ 是序列 $\{X_t\}$ 的均值。认为 x_t 主要是受过去 q 期的误差项的影响。

平稳 $MA(q)$ 模型的性质见表 5-22。

表5-22 平稳MA模型的性质

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数（ACF）	q 阶截尾
方差	常数方差	偏自相关系数（PACF）	拖尾

3. ARMA 模型

具有如下结构的模型称为自回归移动平均模型，简记为 $ARMA(p, q)$。

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{5-41}$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 p 期 $x_{t-1}, x_{t-2}, \cdots, x_{t-p}$ 和前 q 期 $\varepsilon_{t-1}, \varepsilon_{t-2}, \cdots, \varepsilon_{t-q}$ 的多元线性函数，误差项是当期的随机干扰 ε_t，为零均值白噪声序列。认为 x_t 主要是受过去 p 期的序列值和过去 q 期的误差项的共同影响。

特别的，当 $q = 0$ 时，是 $AR(p)$ 模型；当 $p = 0$ 时，是 $MA(q)$ 模型。

平稳 $ARMA(p, q)$ 的性质见表 5-23。

表5-23 平稳ARMA模型的性质

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数（ACF）	拖尾
方差	常数方差	偏自相关系数（PACF）	拖尾

4. 平稳时间序列建模

某个时间序列经过预处理，被判定为平稳非白噪声序列，就可以利用 ARMA 模型进行建模。计算出平稳非白噪声序列 $\{X_t\}$ 的自相关系数和偏自相关系数，再由 $AP(p)$ 模型、$MA(q)$ 和 $ARMA(p, q)$ 的自相关系数和偏自相关系数的性质，选择合适的模型。平稳时间序列建模步骤如图 5-17 所示。

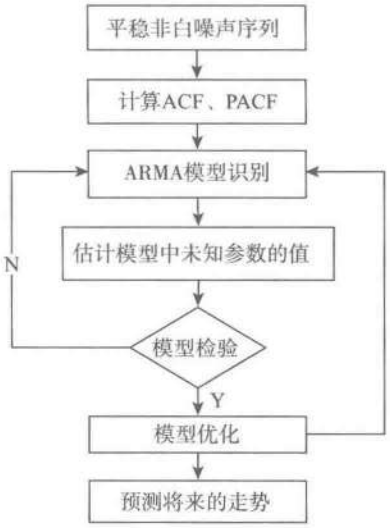


图 5-17 平稳时间序列 ARMA 模型建模步骤

1）计算 ACF 和 PACF。先计算非平稳白噪声序列的自相关系数（ACF）和偏自相关系数（PACF）。

2）ARMA 模型识别。也称为模型定阶，由 $AR(p)$ 模型、$MA(q)$ 和 $ARMA(p, q)$ 的自相关系数和偏自相关系数的性质，选择合适的模型。识别的原则见表 5-24。

表5-24 ARMA模型识别原则

模　型	自相关系数（ACF）	偏自相关系数（PACF）
$AR(p)$	拖尾	p 阶截尾
$MA(q)$	q 阶截尾	拖尾
$ARMA(p, q)$	p 阶拖尾	q 阶拖尾

3）估计模型中未知参数的值并进行参数进行检验。

4）模型检验。

5）模型优化。

6）模型应用：进行短期预测。

5.4.4 非平稳时间序列分析

前面介绍了对平稳时间序列进行分析的方法。实际上，在自然界中绝大部分序列都是非平稳的。因而对非平稳序列的分析更普遍、更重要，创造出来的分析方法也更多。

对非平稳时间序列的分析方法可以分为确定性因素分解的时序分析和随机时序分析两大类。

确定性因素分解的方法把所有序列的变化都归结为 4 个因素（长期趋势、季节变动、循环变动和随机波动）的综合影响，其中长期趋势和季节变动的规律性信息通常比较容易提取，而由随机因素导致的波动则非常难确定和分析，对随机信息浪费严重，会导致模型拟合精度不够理想。

随机时序分析法的发展就是为了弥补确定性因素分解方法的不足。根据时间序列的不同

特点，随机时序分析可以建立的模型有 ARIMA 模型、残差自回归模型、季节模型、异方差模型等。本节重点介绍使用 ARIMA 模型对非平稳时间序列进行建模的方法。

1. 差分运算

（1）p 阶差分

相距一期的两个序列值之间的减法运算称为 1 阶差分运算。

（2）k 步差分

相距 k 期的两个序列值之间的减法运算称为 k 步差分运算。

2. ARIMA 模型

差分运算具有强大的确定性信息提取能力，许多非平稳序列差分后会显示出平稳序列的性质，这时称这个非平稳序列为差分平稳序列。对差分平稳序列可以使用 ARMA 模型进行拟合。ARIMA 模型的实质就是差分运算与 ARMA 模型的组合，掌握了 ARMA 模型的建模方法和步骤以后，对序列建立 ARIMA 模型是比较简单的。

差分平稳时间序列建模步骤如图 5-18 所示。

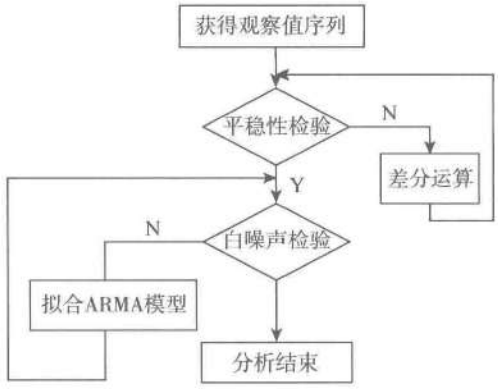


图 5-18 差分平稳时间序列建模步骤

下面应用以上的理论知识，对表 5-25 中 2015/1/1 ～ 2015/2/6 某餐厅的销售数据进行建模。

表5-25 某餐厅的销量数据

日　期	销量（元）	日　期	销量（元）
2015/1/1	3023	2015/1/13	3142
2015/1/2	3039	2015/1/14	3252
2015/1/3	3056	2015/1/15	3342
2015/1/4	3138	2015/1/16	3365
2015/1/5	3188	2015/1/17	3339
2015/1/6	3224	2015/1/18	3345
2015/1/7	3226	2015/1/19	3421
2015/1/8	3029	2015/1/20	3443
2015/1/9	2859	2015/1/21	3428
2015/1/10	2870	2015/1/22	3554
2015/1/11	2910	2015/1/23	3615
2015/1/12	3012	2015/1/24	3646

（续）

日　期	销量（元）	日　期	销量（元）
2015/1/25	3614	2015/2/1	4210
2015/1/26	3574	2015/2/2	4493
2015/1/27	3635	2015/2/3	4560
2015/1/28	3738	2015/2/4	4637
2015/1/29	3707	2015/2/5	4755
2015/1/30	3827	2015/2/6	4817
2015/1/31	4039		

数据详见：示例程序 /data/arima_data.xls

（1）检验序列的平稳性

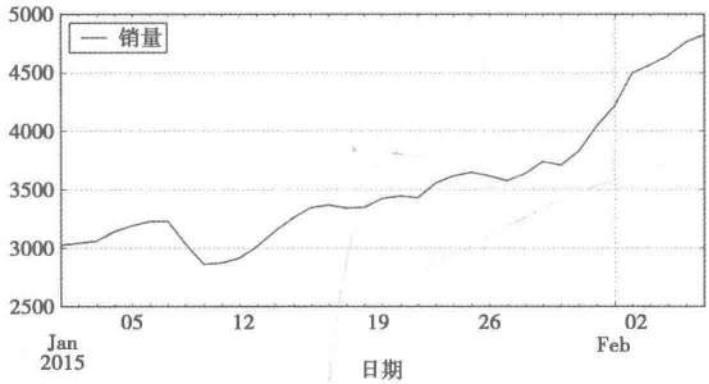


图 5-19　原始序列的时序图

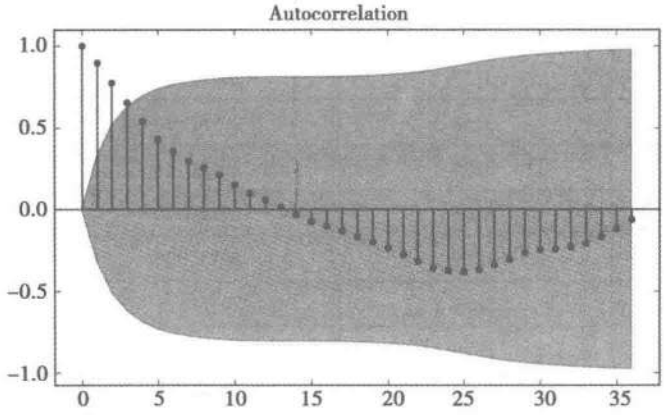


图 5-20　原始序列的自相关图

表5-26 原始序列的单位根检验

adf	cValue			p值
	1%	5%	10%	
1.8138	-3.7112	-2.9812	-2.6301	0.9984

图 5-19 时序图显示该序列具有明显的单调递增趋势，可以判断为是非平稳序列；图 5-20 的自相关图显示自相关系数长期大于零，说明序列间具有很强的长期相关性；表 5-26 单位根检验统计量对应的 p 值显著大于 0.05，最终将该序列判断为非平稳序列（非平稳序列一定不是白噪声序列）。

（2）对原始序列进行一阶差分，并进行平稳性和白噪声检验

1）对一阶差分后的序列再次做平稳性判断。

过程同上。

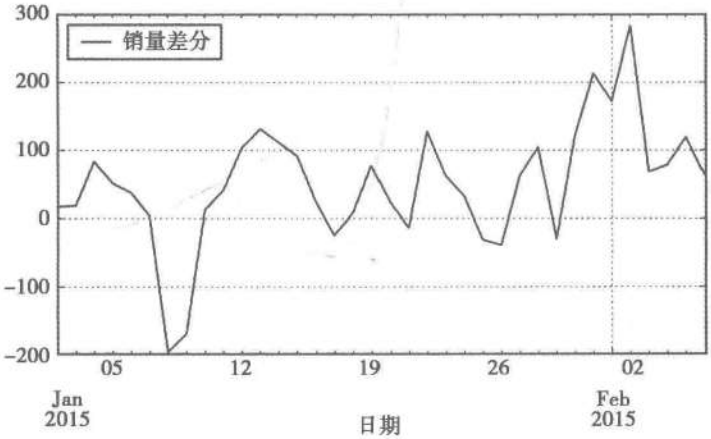


图 5-21 一阶差分之后序列的时序图

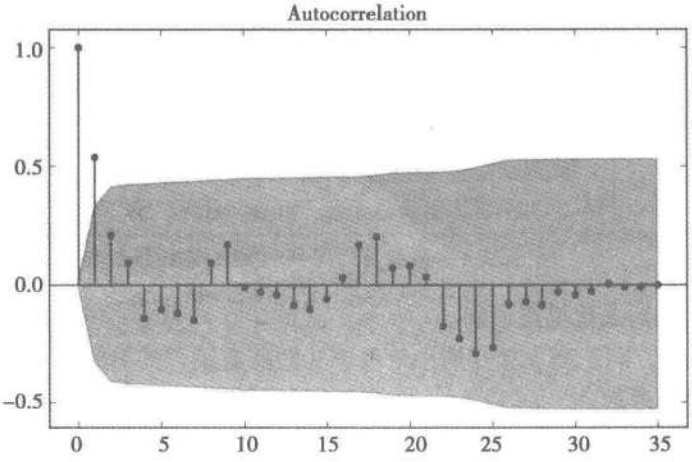


图 5-22 一阶差分之后序列的自相关图

一阶差分之后序列的单位根检验如表 5-27 所示。

表5-27 一阶差分之后序列的单位根检验

adf	cValue			p 值
	1%	5%	10%	
-3.156 1	-3.6327	-2.9485	-2.6130	0.0227

结果显示，一阶差分之后的序列的时序图在均值附近比较平稳的波动、自相关图有很强的短期相关性、单位根检验 p 值小于 0.05，所以一阶差分之后的序列是平稳序列。

2）对一阶差分后的序列做白噪声检验（结果见表 5-28）。

表5-28 一阶差分之后序列的白噪声检验

stat	p 值
11.304	0.007 734

输出的 p 值远小于 0.05，所以一阶差分之后的序列是平稳非白噪声序列。

（3）对一阶差分之后的平稳非白噪声序列拟合 ARMA 模型

下面进行模型定阶。模型定阶就是确定 p 和 q。

第一种方法：人为识别的方法，根据表 5-24 进行模型定阶。

一阶差分后自相关图（见图 5-23）显示出 1 阶截尾，偏自相关图显示出拖尾性，所以可以考虑用 MA（1）模型拟合 1 阶差分后的序列，即对原始序列建立 ARIMA（0，1，1）模型。

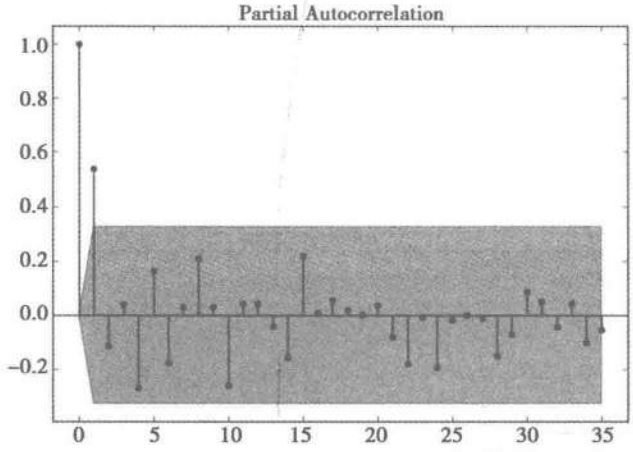


图 5-23 一阶差分后序列的偏自相关图

第二种方法：相对最优模型识别。

计算 ARMA（p，q）。当 p 和 q 均小于等于 3 的所有组合的 BIC 信息量，取其中 BIC 信息量达到最小的模型阶数。

计算完成 BIC 矩阵如下。

432.068472	422.510082	426.088911	426.595507
423.628276	426.073601	NaN	NaN
426.774824	427.395787	430.709154	NaN
430.317524	NaN	NaN	436.478109

p 值为 0、q 值为 1 时最小 BIC 值为：430.137 4。p、q 定阶完成!

用 AR（1）模型拟合一阶差分后的序列，即对原始序列建立 ARIMA（0，1，1）模型。虽然两种方法建立的模型是一样的，但模型是非唯一的，可以检验 ARIMA（1，1，0）和 ARIMA（1，1，1），这两个模型也能通过检验。

下面对一阶差分后的序列拟合 AR（1）模型进行分析。

1）模型检验。残差为白噪声序列，p 值为：0.627 016。

2）参数检验和参数估计见表 5-29。

表5-29 模型参数

Parameter	Coef.	Std. Err.	t
const	49.956	20.139	2.4806
ma.L1.D. 销量	0.671	0.1648	4.0712

（4）ARIMA 模型预测

应用 ARIMA（0，1，1）对表 530 中 2015/1/1 ～ 2015/2/6 某餐厅的销售数据做为期 5 天的预测，结果如下。

2015/2/7	2015/2/8	2015/2/9	2015/2/10	2015/2/11
4874.0	4923.9	4973.9	5023.8	5073.8

需要说明的是，利用模型向前预测的时期越长，预测误差将会越大，这是时间预测的典型特点。

在 Python 中实现 ARIMA 模型建模过程的代码如代码清单 5-7 所示。可以看到，我们使用了 StatsModels，读者或许记得，我们在第 2 章介绍了它，在这里才真正用上它。这表明对于通常的数据探索任务来说，Numpy 与 Pandas 的结合已经相当强大了，只有到较为深入的统计模型之时，才用到 StatsModels。

代码清单5-7 ARIMA模型实现代码

```
#-*- coding: utf-8 -*-
#arima时序模型

import pandas as pd

#参数初始化
discfile = '../data/arima_data.xls'
forecastnum = 5
```

```
#读取数据，指定日期列为指标，Pandas自动将“日期”列识别为Datetime格式
data = pd.read_excel(discfile, index_col = u'日期')

#时序图
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
data.plot()
plt.show()

#自相关图
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(data).show()

#平稳性检测
from statsmodels.tsa.stattools import adfuller as ADF
print(u'原始序列的ADF检验结果为：', ADF(data[u'销量']))
#返回值依次为adf、pvalue、usedlag、nobs、critical values、icbest、regresults、resstore

#差分后的结果
D_data = data.diff().dropna()
D_data.columns = [u'销量差分']
D_data.plot() #时序图
plt.show()
plot_acf(D_data).show() #自相关图
from statsmodels.graphics.tsaplots import plot_pacf
plot_pacf(D_data).show() #偏自相关图
print(u'差分序列的ADF检验结果为：', ADF(D_data[u'销量差分'])) #平稳性检测

#白噪声检验
from statsmodels.stats.diagnostic import acorr_ljungbox
print(u'差分序列的白噪声检验结果为：', acorr_ljungbox(D_data, lags=1)) #返回统计量和p值

from statsmodels.tsa.arima_model import ARIMA

#定阶
pmax = int(len(D_data)/10) #一般阶数不超过length/10
qmax = int(len(D_data)/10) #一般阶数不超过length/10
bic_matrix = [] #bic矩阵
for p in range(pmax+1):
  tmp = []
  for q in range(qmax+1):
    try: #存在部分报错，所以用try来跳过报错。
      tmp.append(ARIMA(data, (p,1,q)).fit().bic)
    except:
      tmp.append(None)
  bic_matrix.append(tmp)

bic_matrix = pd.DataFrame(bic_matrix) #从中可以找出最小值
```

```
p,q = bic_matrix.stack().idxmin() #先用stack展平，然后用idxmin找出最小值位置。
print(u'BIC最小的p值和q值为：%s、%s' %(p,q))
model = ARIMA(data, (p,1,q)).fit() #建立ARIMA(0, 1, 1)模型
model.summary2() #给出一份模型报告
model.forecast(5) #作为期5天的预测，返回预测结果、标准误差、置信区间。
```

代码详见：示例程序 /code/arima_test.py

运行代码清单 5-7 可以得到输出结果如下。

```
原始序列的ADF检验结果为： (1.8137710150945285, 0.99837594215142644, 10, 26, {'5%':
   -2.9812468047
337282, '10%': -2.6300945562130176, '1%': -3.7112123008648155}, 299.46989866024177)

差分序列的ADF检验结果为： (-3.1560562366723537, 0.022673435440048798, 0, 35, {'5%':
   -2.948510204
0816327, '10%': -2.6130173469387756, '1%': -3.6327426647230316}, 287.59090907803341)

差分序列的白噪声检验结果为： (array([ 11.30402222]), array([ 0.00077339]))

BIC最小的p值和q值为：0、1

                          Results: ARIMA
==========================================================================
Model:              ARIMA            Log-Likelihood:       -205.88
Dependent Variable: D.销量            Scale:                 1.0000
Date:               2015-08-06 10:24 Method:               css-mle
No. Observations:   36               Sample:               01-02-2015
Df Model:           2                                      02-06-2015
Df Residuals:       34               S.D. of innovations: 73.086
AIC:                417.7595         HQIC:                 419.418
BIC:                422.5101
--------------------------------------------------------------------------
               Coef.    Std.Err.     t      P>|t|     [0.025     0.975]
--------------------------------------------------------------------------
const         49.9560    20.1390   2.4806   0.0182   10.4843   89.4277
ma.L1.D.销量     0.6710     0.1648   4.0712   0.0003    0.3480    0.9941
------------------------------------------------------------------------------
                 Real           Imaginary           Modulus          Frequency
------------------------------------------------------------------------------
MA.1           -1.4902             0.0000           1.4902             0.5000

(array([ 4873.96649322,  4923.9224731 ,  4973.87845298,  5023.83443286,
        5073.79041274]), array([   73.08574266,   142.32680564,   187.54283091,
           223.80283119,
        254.95705731]), array([[ 4730.72106983,  5017.21191661],
       [ 4644.96706002,  5202.87788618],
       [ 4606.30125884,  5341.45564712],
       [ 4585.18894409,  5462.47992162],
       [ 4574.08376281,  5573.49706266]]))
```

5.4.5 Python 主要时序模式算法

Python 实现时序模式的主要库是 StatsModels（当然，如果 Pandas 能做的，就可以利用 Pandas 先做），算法主要是 ARIMA 模型，在使用该模型进行建模时，需要进行一系列判别操作，主要包含平稳性检验、白噪声检验、是否差分、AIC 和 BIC 指标值、模型定阶，最后再做预测。与其相关的函数见表 5-30。

表5-30 时序模式算法函数列表

函 数 名	函 数 功 能	所属工具箱
acf()	计算自相关系数	statsmodels.tsa.stattools
plot_acf()	画自相关系数图	statsmodels.graphics.tsaplots
pacf()	计算偏相关系数	statsmodels.tsa.stattools
plot_pacf()	画偏相关系数图	statsmodels.graphics.tsaplots
adfuller()	对观测值序列进行单位根检验	statsmodels.tsa.stattools
diff()	对观测值序列进行差分计算	Pandas 对象自带的方法
ARIMA()	创建一个 ARIMA 时序模型	statsmodels.tsa.arima_model
summary() 或 summaty2	给出一份 ARIMA 模型的报告	ARIMA 模型对象自带的方法
aic/bic/hqic	计算 ARIMA 模型的 AIC/BIC/HQIC 指标值	ARIMA 模型对象自带的变量
forecast()	应用构建的时序模型进行预测	ARIMA 模型对象自带的方法
acorr_ljungbox()	Ljung-Box 检验，检验是否为白噪声	statsmodels.stats.diagnostic

（1）acf()

- 功能：计算自相关系数
- 使用格式：

```
autocorr = acf(data, unbiased=False, nlags=40, qstat=False, fft=False, alpha=None)
```

输入参数 data 为观测值序列（即为时间序列，可以是 DataFrame 或 Series），返回参数 autocorr 为观测值序列自相关函数。其余为可选参数，如 qstat=True 时同时返回 Q 统计量和对应 p 值。

（2）plot_acf()

- 功能：画自相关系数图
- 使用格式：

```
p = plot_acf(data)
```

返回一个 Matplotlib 对象，可以用 .show() 方法显示图像。

（3）acf() / plot_acf()

- 功能：计算偏相关系数 / 画偏相关系数图
- 使用格式：使用跟 acf() / plot_acf() 类似，不再赘述。

（4）adfuller()

❑ 功能：对观测值序列进行单位根检验（ADF test）

❑ 使用格式：

h = adffuller(Series, maxlag=None, regression='c', autolag='AIC', store=False, regresults=False)

输入参数 Series 为一维观测值序列，返回值依次为 adf、pvalue、usedlag、nobs、critical values、icbest、regresults、resstore。

（5）diff()

❑ 功能：对观测值序列进行差分计算

❑ 使用格式：

D.diff() D 为 Pandas 的 DataFrame 或 Series。

（6）arima

❑ 功能：设置时序模式的建模参数，创建 ARIMA 时序模型

❑ 使用格式：

arima = ARIMA(data, (p,1,q)).fit()

data 参数为输入的时间序列，p、q 为对应的阶，d 为差分次数。

（7）summary() /summary2()

❑ 功能：生成已有模型的报告

❑ 使用格式：

arima.summary() / arima.summary2()

其中，arima 为已经建立好的 ARIMA 模型，返回一份格式化的模型报告，包含模型的系数、标准误差、p 值、AIC 和 BIC 等详细指标。

（8）aic/bic/hqic

❑ 功能：计算 ARIMA 模型的 AIC、BIC、HQIC 指标值

❑ 使用格式：

arima.aic/arima.bic/arima.hqic

其中，arima 为已经建立好的 ARIMA 模型，返回值是 Model 时序模型得到的 AIC、BIC 和 HQIC 指标值。

（9）forecast()

❑ 功能：用得到的时序模型进行预测

❑ 使用格式：

a,b,c = arima.forecast(num)

输入参数 num 为要预测的天数，arima 为已经建立好的 ARIMA 模型。a 为返回的预测值，b 为预测的误差，c 为预测置信区间。

（10）acorr_ljungbox()

❑ 功能：检测是否为白噪声序列

❑ 使用格式：

acorr_ljungbox(data, lags=1),

输入参数 data 为时间序列数据，lags 为滞后数，返回统计量和 p 值。

5.5 离群点检测

就餐饮企业而言，经常会碰到如下问题。

1）如何根据客户的消费记录检测是否为异常刷卡消费？

2）如何检测是否有异常订单？

这一类异常问题可以通过离群点检测来解决。

离群点检测是数据挖掘中重要的一部分，它的任务是发现与大部分其他对象显著不同的对象。大部分数据挖掘方法都将这种差异信息视为噪声而丢弃，然而在一些应用中，罕见的数据可能蕴含着更大的研究价值。

在数据的散布图中，图 5-24 所示离群点远离其他数据点。因为离群点的属性值明显偏离期望的或常见的属性值，所以离群点检测也称偏差检测。

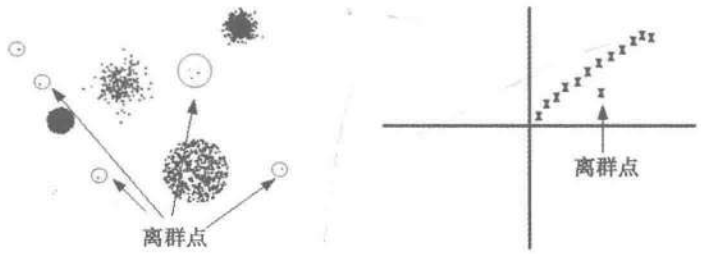


图 5-24 离群点检测示意图

离群点检测已经被广泛应用于电信和信用卡的诈骗检测、贷款审批、电子商务、网络入侵和天气预报等领域。例如，可以利用离群点检测分析运动员的统计数据，以发现异常的运动员。

（1）离群点的成因

离群点的主要成因有：数据来源于不同的类、自然变异、数据测量和收集误差。

（2）离群点的类型

对离群点的大致分类见表 5-31。

表5-31 离群点的大致分类

分类标准	分类名称	分类描述
从数据范围	全局离群点和局部离群点	从整体来看，某些对象没有离群特征，但是从局部来看，却显示了一定的离群性。如图 5-25 所示，C 是全局离群点，D 是局部离群点
从数据类型	数值型离群点和分类型离群点	这是以数据集的属性类型进行划分的

（续）

分类标准	分类名称	分类描述
从属性的个数	一维离群点和多维离群点	一个对象可能有一个或多个属性

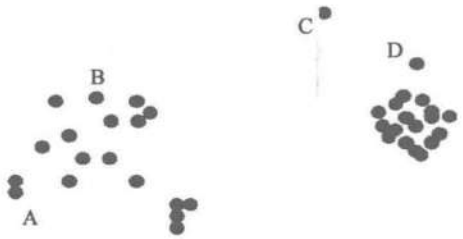


图 5-25 全局离群点和局部离群点

5.5.1 离群点检测方法

常用离群点检测方法[14]见表 5-32。

表5-32 常用离群点检测方法

离群点检测方法	方法描述	方法评估
基于统计	大部分的基于统计的离群点检测方法是构建一个概率分布模型，并计算对象符合该模型的概率，把具有低概率的对象视为离群点	基于统计模型的离群点检测方法的前提是必须知道数据集服从什么分布；对于高维数据，检验效果可能很差
基于邻近度	通常可以在数据对象之间定义邻近性度量，把远离大部分点的对象视为离群点	简单，二维或三维的数据可以做散点图观察；大数据集不适用；对参数选择敏感；具有全局阈值，不能处理具有不同密度区域的数据集
基于密度	考虑数据集可能存在不同密度区域这一事实，从基于密度的观点分析，离群点是在低密度区域中的对象。一个对象的离群点得分是该对象周围密度的逆	给出了对象是离群点的定量度量，并且即使数据具有不同的区域也能够很好处理；大数据集不适用；参数选择是困难的
基于聚类	一种利用聚类检测离群点的方法是丢弃远离其他簇的小簇；另一种更系统的方法，首先聚类所有对象，然后评估对象属于簇的程度（离群点得分）	基于聚类技术来发现离群点可能是高度有效的；聚类算法产生的簇的质量对该算法产生的离群点的质量影响非常大

基于统计模型的离群点检测方法需要满足统计学原理，如果分布已知，则检验可能非常有效。基于邻近度的离群点检测方法比统计学方法更一般、更容易使用，因为确定数据集有意义的邻近度量比确定它的统计分布更容易。基于密度的离群点检测与基于邻近度的离群点检测密切相关，因为密度常用邻近度定义：一种是定义密度为到 K 个最邻近的平均距离的倒数，如果该距离小，则密度高；另一种是使用 DBSCAN 聚类算法，一个对象周围的密度等于该对象指定距离 d 内对象的个数。

下面重点介绍基于统计模型和聚类的离群点检测方法。

5.5.2 基于模型的离群点检测方法

通过估计概率分布的参数来建立一个数据模型。如果一个数据对象不能很好地同该模型拟合，即如果它很可能不服从该分布，则它是一个离群点。

（1）一元正态分布中的的离群点检测

正态分布是统计学中最常用的分布之一。

若随机变量 x 的密度函数 $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} (x \in R)$，则称 x 服从正态分布，简称 x 服从正态分布 $N(\mu, \sigma)$，其中参数 μ 和 σ 分别为均值和标准差。

图 5-26 显示 $N(0, 1)$ 的密度函数。

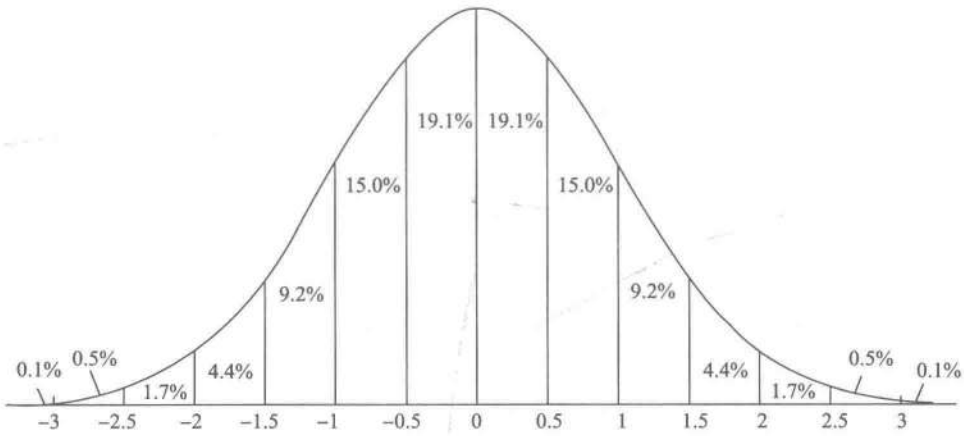


图 5-26 $N(0, 1)$ 的概率密度函数

$N(0, 1)$ 的数据对象出现在该分布的两边尾部的机会很小，因此可以用它作为检测数据对象是否是离群点的基础。数据对象落在 3 倍标准差中心区域之外的概率仅有 0.002 7。

（2）混合模型的离群点检测 [14]

这里先介绍一下混合模型。混合模型是一种特殊的统计模型，它使用若干统计分布对数据建模。每一个分布对应一个簇，而每个分布的参数提供对应簇的描述，通常用中心和发散描述。

混合模型将数据看作从不同的概率分布得到的观测值的集合。概率分布可以是任何分布，但是通常是多元正态的，因为这种类型的分布不难理解，容易从数学上进行处理，并且已经证明在许多情况下都能产生好的结果。这种类型的分布可以对椭圆簇建模。

总的来说，混合模型数据产生过程为：给定几个类型相同但参数不同的分布，随机地选取一个分布并由它产生一个对象。重复该过程 m 次，其中 m 是对象的个数。

具体地讲，假定有 K 个分布和 m 个对象 $\chi = \{x_1, x_2, \cdots, x_m\}$。设第 j 个分布的参数为 α_j，并设 A 是所有参数的集合，即 $A = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$。则 $P(x_i \mid \alpha_j)$ 是第 i 个对象来自第 j 个分布

的概率。选取第 j 个分布产生一个对象的概率由权值 $w_j(1 \leqslant j \leqslant K)$ 给定，其中权值（概率）受限于其和为 1 的约束，即 $\sum_{j=1}^{K} w_j = 1$。于是，对象 x 的概率由以下公式给出：

$$P(x \mid A) = \sum_{j=1}^{K} w_j P_j(x \mid \theta_j) \tag{5-42}$$

如果对象以独立的方式产生，则整个对象集的概率是每个个体对象 x_i 的概率的乘积，公式如下。

$$P(\chi \mid \alpha) = \prod_{i=1}^{m} P(x_i \mid \alpha) = \prod_{i=1}^{m} \sum_{j=1}^{K} w_j P_j(x \mid \alpha_j) \tag{5-43}$$

对于混合模型，每个分布描述一个不同的组，即一个不同的簇。通过使用统计方法，可以由数据估计这些分布的参数，从而描述这些分布（簇）。也可以识别哪个对象属于哪个簇。然而，混合模型只是给出具体对象属于特定簇的概率。

聚类时，混合模型方法假定数据来自混合概率分布，并且每个簇可以用这些分布之一识别。同样，对于离群点检测，用两个分布的混合模型建模，一个分布为正常数据，而另一个为离群点。

聚类和离群点检测的目标都是估计分布的参数，以最大化数据的总似然。

我们提供一种离群点检测常用的简单的方法：先将所有数据对象放入正常数据集，这时离群点集为空集；再用一个迭代过程将数据对象从正常数据集转移到离群点集，该转移能提高数据的总似然。

具体操作如下。

假设数据集 U 包含来自两个概率分布的数据对象：M 是大多数（正常）数据对象的分布，而 N 是离群点对象的分布。数据的总概率分布可以记作：

$U(x) = (1-\lambda)M(x) + \lambda N(x)$ 其中，x 是一个数据对象；$\lambda \in [0, 1]$，给出离群点的期望比例。分布 M 由数据估计得到，而分布 N 通常取均匀分布。设 M_t 和 N_t 分别为时刻 t 正常数据和离群点对象的集合。初始 $t=0$，$M_0=D$，而 $N_0 \neq \varnothing$。

根据公式混合模型中公式 $P(x \mid A) = \sum_{j=1}^{K} w_j P_j(x \mid \alpha_j)$ 推导，在整个数据集的似然和对数似然可分别由下面两式给出。

$$L_t(U) = \prod_{x_i \in U} P_U(x_i) = \left((1-\lambda)^{|M_t|} \prod_{x_i \in M_i} P_{M_i}(x_i)\right)\left(\lambda^{|N_t|} \prod_{x_i \in N_i} P_{N_i}(x_i)\right) \tag{5-44}$$

$$\ln L_t(U) = |M_t| \ln(1-\lambda) + \sum_{x_i \in M_i} \ln P_{M_i}(x_i) + |N_t| \ln\lambda + \sum_{x_i \in N_i} \ln P_{N_i}(x_i) \tag{5-45}$$

其中 P_D、P_{M_t}、P_{N_t} 分别是 D、M_t、N_t 的概率分布函数。

因为正常数据对象的数量比离群点对象的数量大很多，因此当一个数据对象移动到离群点集后，正常数据对象的分布变化不大。在这种情况下，每个正常数据对象的总似然的贡献保持不变。此外，如果假定离群点服从均匀分布，则移动到离群点集的每一个数据对象对离群点的似然贡献一个固定的量。这样，当一个数据对象移动到离群点集时，数据总似然的改

变粗略地等于该数据对象在均匀分布下的概率（用 λ 加权）减去该数据对象在正常数据点的分布下的概率（用 $1-\lambda$ 加权）。从而，离群点由这样一些数据对象组成。这样，数据对象在均匀分布下的概率比正常数据对象分布下的概率高。

在某些情况下是很难建立模型的。例如，因为数据的统计分布未知或没有训练数据可用。在这种情况下，可以考虑其他不需要建立模型的检测方法。

5.5.3 基于聚类的离群点检测方法

聚类分析用于发现局部强相关的对象组，而异常检测用来发现不与其他对象强相关的对象。因此，聚类分析非常自然地可以用于离群点检测。本节主要介绍两种基于聚类的离群点检测方法。

（1）丢弃远离其他簇的小簇

一种利用聚类检测离群点的方法是丢弃远离其他簇的小簇。通常，该过程可以简化为丢弃小于某个最小阈值的所有簇。

这个方法可以和其他任何聚类技术一起使用，但是需要最小簇大小和小簇与其他簇之间距离的阈值。而且这种方案对簇个数的选择高度敏感，使用这个方案很难将离群点得分附加到对象上。

在图 5-27 中，聚类簇数 K = 2，可以直观地看出其中一个包含 5 个对象的小簇远离大部分对象，可以视为离群点。

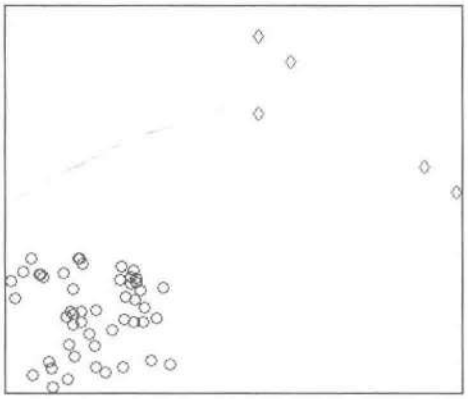


图 5-27 K-Means 算法的聚类图

（2）基于原型的聚类

基于原型的聚类是另一种更系统的方法。首先聚类所有对象，然后评估对象属于簇的程度（离群点得分）。在这种方法中，可以用对象到它的簇中心的距离来度量属于簇的程度。特别地，如果删除一个对象导致该目标的显著改进，则可将该对象视为离群点。例如，在 K 均值算法中，删除远离其相关簇中心的对象能够显著地改进该簇的误差平方和（SSE）。

对于基于原型的聚类，主要有两种方法评估对象属于簇的程度（离群点得分）：一是度量对象到簇原型的距离，并用它作为该对象的离群点得分；二是考虑到簇具有不同的密度，可以度量簇到原型的相对距离，相对距离是点到质心的距离与簇中所有点到质心的距离的中位数之比。

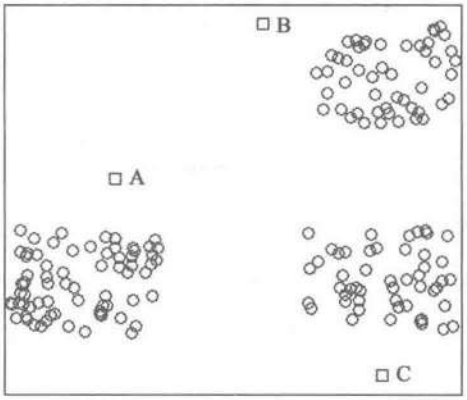


图 5-28 基于距离的离群点检测

如图 5-28 所示，如果选择聚类簇数 K = 3，则

对象 A、B、C 应分别属于距离它们最近的簇，但相对于簇内的其他对象，这 3 个点又分别远离各自的簇，所以有理由怀疑对象 A、B、C 是离群点。

诊断步骤如下。

1）进行聚类。选择聚类算法（如 K-Means 算法），将样本集聚为 K 簇，并找到各簇的质心。

2）计算各对象到它的最近质心的距离。

3）计算各对象到它的最近质心的相对距离。

4）与给定的阈值作比较。

如果某对象距离大于该阈值，就认为该对象是离群点。

基于聚类的离群点检测的改进如下。

1）离群点对初始聚类的影响：通过聚类检测离群点时，离群点会影响聚类结果。为了处理该问题，可以使用方法：对象聚类，删除离群点，对象再次聚类（这个不能保证产生最优结果）。

2）还有一种更复杂的方法：取一组不能很好地拟合任何簇的特殊对象，这组对象代表潜在的离群点。随着聚类过程的进展，簇在变化。不再强属于任何簇的对象被添加到潜在的离群点集合；测试当前在该集合中的对象，如果它现在强属于一个簇，就可以将它从潜在的离群点集合中移除。聚类过程结束时还留在该集合中的点被分类为离群点（这种方法也不能保证产生最优解，甚至不比前面的简单算法好，在使用相对距离计算离群点得分时，这个问题特别严重）。

对象是否被认为是离群点可能依赖于簇的个数（如 k 很大时的噪声簇）。该问题也没有简单的答案。一种策略是对于不同的簇个数重复该分析。另一种方法是找出大量小簇，其想法如下。

1）较小的簇倾向于更加凝聚；

2）如果存在大量小簇时，一个对象是离群点，则它多半是一个真正的离群点。

不利的一面是一组离群点可能形成小簇从而逃避检测。

利用表 5-14 的数据进行聚类分析，并计算各个样本到各自聚类中心的距离，分析离群样本，得到如图 5-29 所示的距离误差图。

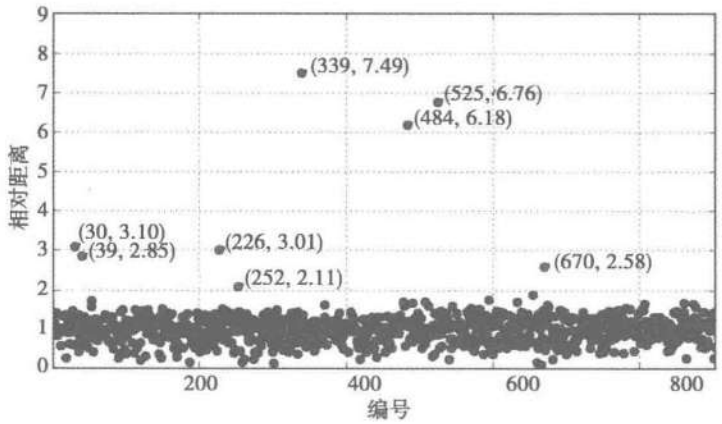


图 5-29 离散点检测距离误差图

分析图 5-29 可以得到，如果距离阈值设置为 2，那么所给的数据中有 8 个离散点，在聚类的时候这些数据应该剔除。

其 Python 代码如代码清单 5-8 所示。

代码清单5-8 离散点检测

```
#-*- coding: utf-8 -*-
#使用K-Means算法聚类消费行为特征数据

import numpy as np
import pandas as pd

#参数初始化
inputfile = '../data/consumption_data.xls' #销量及其他属性数据
k = 3 #聚类的类别
threshold = 2 #离散点阈值
iteration = 500 #聚类最大循环次数
data = pd.read_excel(inputfile, index_col = 'Id') #读取数据
data_zs = 1.0*(data - data.mean())/data.std() #数据标准化

from sklearn.cluster import KMeans
model = KMeans(n_clusters = k, n_jobs = 4, max_iter = iteration) #分为k类，并发数4
model.fit(data_zs) #开始聚类

#标准化数据及其类别
r = pd.concat([data_zs, pd.Series(model.labels_, index = data.index)], axis = 1)
#每个样本对应的类别
    r.columns = list(data.columns) + [u'聚类类别'] #重命名表头

norm = []
for i in range(k): #逐一处理
  norm_tmp = r[['R', 'F', 'M']][r[u'聚类类别'] == i]-model.cluster_centers_[i]
  norm_tmp = norm_tmp.apply(np.linalg.norm, axis = 1) #求出绝对距离
  norm.append(norm_tmp/norm_tmp.median()) #求相对距离并添加

norm = pd.concat(norm) #合并

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
norm[norm <= threshold].plot(style = 'go') #正常点

discrete_points = norm[norm > threshold] #离群点
discrete_points.plot(style = 'ro')

for i in range(len(discrete_points)): #离群点做标记
  id = discrete_points.index[i]
  n = discrete_points.iloc[i]
  plt.annotate('(%s, %0.2f)'%(id, n), xy = (id, n), xytext = (id, n))
```

```
plt.xlabel(u'编号')
plt.ylabel(u'相对距离')
plt.show()
```

代码详见：示例程序 /code/discrete_point_test.py

5.6 小结

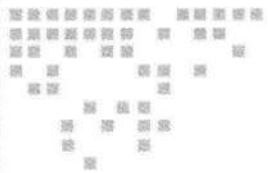
本章主要根据数据挖掘的应用分类，重点介绍了对应的数据挖掘建模方法及实现过程。通过对本章的学习，可在以后的数据挖掘过程中采用适当的算法，并按所陈述的步骤实现综合应用，希望本章能给读者一些启发，思考如何改进或创造更好的挖掘算法。

归纳起来，数据挖掘技术的基本任务主要体现在分类与预测、聚类、关联规则、时序模式、离群点检测 5 个方面。5.1 节主要介绍了决策树和人工神经网络两个分类模型、回归分析预测模型及其实现过程；5.2 节主要介绍了 K-Means 聚类算法，建立分类方法按照接近程度对观测对象给出合理的分类并解释类与类之间的区别；5.3 节主要介绍了 Apriori 算法，以在一个数据集中找出各项之间的关系；5.4 节从序列的平稳性和非平稳型出发，在平稳时间序列中主要介绍了 ARMA 模型，在差分平稳序列中介绍了 ARIMA 模型，应用这两个模型对相应的时间序列进行研究，找寻变化发展的规律，预测将来的走势；5.5 节主要介绍了基于模型和离群点的检测方法，是发现与大部分对象显著不同的对象。

前 5 章是数据挖掘必备的原理知识，为本书后面章节的案例理解和实验操作奠定了理论基础。

实 战 篇

- 第 6 章　电力窃漏电用户自动识别
- 第 7 章　航空公司客户价值分析
- 第 8 章　中医证型关联规则挖掘
- 第 9 章　基于水色图像的水质评价
- 第 10 章　家用电器用户行为分析与事件识别
- 第 11 章　应用系统负载分析与磁盘容量预测
- 第 12 章　电子商务网站用户行为分析及服务推荐
- 第 13 章　财政收入影响因素分析及预测模型
- 第 14 章　基于基站定位数据的商圈分析
- 第 15 章　电商产品评论数据情感分析



Chapter 6 第 6 章

电力窃漏电用户自动识别

6.1 背景与挖掘目标

传统的防窃漏电方法主要通过定期巡检、定期校验电表、用户举报窃电等方法来发现窃电或计量装置故障。但这种方法对人的依赖性太强，抓窃查漏的目标不明确。目前，很多供电局主要通过营销稽查人员、用电检查人员和计量工作人员利用计量异常报警功能和电能量数据查询功能开展用户用电情况的在线监控工作，通过采集电量异常、负荷异常、终端报警、主站报警、线损异常等信息，建立数据分析模型，来实时监测窃漏电情况和发现计量装置的故障。根据报警事件发生前后客户计量点有关的电流、电压、负荷数据情况等，构建基于指标加权的用电异常分析模型，实现检查客户是否存在窃电、违章用电及计量装置故障等。

以上防窃漏电的诊断方法，虽然能获得用电异常的某些信息，但由于终端误报或漏报过多，无法达到真正快速精确定位窃漏电嫌疑用户的目的，往往令稽查工作人员无所适从。而且在采用这种方法建模时，模型各输入指标权重的确定需要用专家的知识和经验来判断，具有很大的主观性，存在明显的缺陷，所以实施效果往往不尽如人意。

现有的电力计量自动化系统能够采集到各相电流、电压、功率因数等用电负荷数据以及用电异常等终端报警信息。异常告警信息和用电负荷数据能够反映用户的用电情况，同时稽查工作人员也会通过在线稽查系统和现场稽查来找出窃漏电用户，并录入系统。若能通过这些数据信息提取出窃漏电用户的关键特征，构建窃漏电用户的识别模型，就能自动检查、判断用户是否存在窃漏电行为。

表 6-1 给出了某企业大用户的用电负荷数据，采集时间间隔为 15 分钟，即 0.25 小时，可进一步计算该大用户的用电量。

表6-1　某企业大用户用电负荷数据

用户编号	时间	有功总	B 相	C 相	电流 A 相	电流 B 相	电流 C 相	电压 A 相	电压 B 相	电压 C 相	功率因数	功率因数 A	功率因数 B	功率因数 C
0319001000019011001	2011/11/10	202	0	349.2	33.6	0	33.4	10500	0	10500	0.784	0.573	−10000	0.996
0319001000019011001	2011/11/10 0:15	194.8	0	355.4	32.4	0	34	10500	0	10500	0.789	0.573	−10000	0.996
0319001000019011001	2011/11/10 0:30	210.4	0	366	35	0	35	10500	0	10500	0.784	0.573	−10000	0.996
0319001000019011001	2011/11/10 0:45	199.6	0	376.4	33.2	0	36	10500	0	10500	0.793	0.573	−10000	0.996
0319001000019011001	2011/11/10 1:00	191.2	0	334.6	31.8	0	32	10500	0	10500	0.785	0.573	−10000	0.996
0319001000019011001	2011/11/10 1:15	192.4	0	340.8	32	0	32.6	10500	0	10500	0.786	0.573	−10000	0.996
0319001000019011001	2011/11/10 1:30	192.4	0	353.4	32	0	33.8	10500	0	10500	0.79	0.573	−10000	0.996
0319001000019011001	2011/11/10 1:45	197.2	0	357.6	32.8	0	34.2	10500	0	10500	0.789	0.573	−10000	0.996
0319001000019011001	2011/11/10 2:00	178	0	320.8	29.6	0	30.4	10500	0	10600	0.788	0.573	−10000	0.996
0319001000019011001	2011/11/10 2:15	173.2	0	311.6	28.8	0	29.8	10500	0	10500	0.788	0.573	−10000	0.996
0319001000019011001	2011/11/10 2:30	185.2	0	332.4	30.8	0	31.8	10500	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 2:45	175.6	0	326.2	29.2	0	31.2	10500	0	10500	0.791	0.573	−10000	0.996
0319001000019011001	2011/11/10 3:00	164.8	0	311.6	27.4	0	29.8	10500	0	10500	0.793	0.573	−10000	0.996
0319001000019011001	2011/11/10 3:15	185.8	0	317.8	31.2	0	30.4	10400	0	10500	0.782	0.573	−10000	0.996
0319001000019011001	2011/11/10 3:30	169.6	0	303.2	28.2	0	29	10500	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 3:45	179.2	0	320	29.8	0	30.6	10500	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 4:00	175.6	0	305.2	29.2	0	29.2	10500	0	10500	0.784	0.573	−10000	0.995
0319001000019011001	2011/11/10 4:15	178.6	0	324	30	0	31	10400	0	10500	0.788	0.572	−10000	0.995
0319001000019011001	2011/11/10 4:30	173.2	0	313.6	28.8	0	30	10500	0	10500	0.788	0.573	−10000	0.996
0319001000019011001	2011/11/10 4:45	166	0	297	27.6	0	28.4	10500	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 5:00	170.8	0	303.2	28.4	0	29	10500	0	10500	0.786	0.573	−10000	0.996
0319001000019011001	2011/11/10 5:15	176.8	0	322	29.4	0	30.8	10500	0	10500	0.789	0.573	−10000	0.996
0319001000019011001	2011/11/10 5:30	175.6	0	301	29.2	0	28.8	10500	0	10500	0.783	0.573	−10000	0.995
0319001000019011001	2011/11/10 5:45	164.4	0	299	27.6	0	28.6	10400	0	10500	0.789	0.573	−10000	0.996
0319001000019011001	2011/11/10 6:00	168.4	0	315.8	28	0	30.2	10500	0	10500	0.792	0.573	−10000	0.996
0319001000019011001	2011/11/10 6:15	165.6	0	284.4	27.8	0	27.2	10400	0	10500	0.783	0.573	−10000	0.996
0319001000019011001	2011/11/10 6:30	164.4	0	297	27.6	0	28.4	10400	0	10500	0.788	0.573	−10000	0.996
0319001000019011001	2011/11/10 6:45	188.2	0	334.6	31.6	0	32	10400	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 7:00	179.8	0	315.8	30.2	0	30.2	10400	0	10500	0.785	0.572	−10000	0.996
0319001000019011001	2011/11/10 7:15	165.6	0	290	27.8	0	28	10400	0	10400	0.785	0.573	−10000	0.996
0319001000019011001	2011/11/10 7:30	219	0	391	36.4	0	37.4	10500	0	10500	0.787	0.573	−10000	0.996
0319001000019011001	2011/11/10 7:45	227.6	0	403.6	38.2	0	38.6	10400	0	10500	0.786	0.573	−10000	0.996

表 6-2 给出了该企业大用户的终端报警数据，其中与窃漏电相关的报警能较好的识别用户的窃漏电行为。表 6-3 给出了某企业大用户违约、窃电处理通知书，表中记录了用户的用电类别和窃电时间。

表6-2 某企业大用户终端报警信息

用户名称	时 间	计量点ID	报警编号	报警名称
某企业大用户	2010/4/1 0:01	0319001000045110001	135	最大需量复零
某企业大用户	2010/4/2 18:44	0319001000045110001	152	电流不平衡
某企业大用户	2010/4/2 18:47	0319001000045110001	143	A 相电流过负荷
某企业大用户	2010/4/2 18:47	0319001000045110001	145	C 相电流过负荷
某企业大用户	2010/4/2 21:07	0319001000045110001	152	电流不平衡
某企业大用户	2010/4/2 21:22	0319001000045110001	145	C 相电流过负荷
某企业大用户	2010/4/2 21:25	0319001000045110001	143	A 相电流过负荷
某企业大用户	2010/4/3 5:45	0319001000045110001	145	C 相电流过负荷

注：由于各方面原因，终端报警存在一定误报和漏报情况。

表6-3 某企业大用户违约、窃电处理通知书

<table>
<tr><td rowspan="3">用户基本信息</td><td>用户名称</td><td colspan="2">某企业大用户</td><td>用户编号</td><td colspan="3">7210100429</td></tr>
<tr><td>用电地址</td><td colspan="2">******</td><td>用电类别</td><td>大工业</td><td>报装容量</td><td>1515kVA</td></tr>
<tr><td>计量方式</td><td>高供高计</td><td>电流互感器变比</td><td>100/5</td><td>电压互感器变比</td><td colspan="2">10 000V/100V</td></tr>
<tr><td>现场情况</td><td colspan="7">我局用电检查人员根据群众举报，于 2014 年 11 月 17 日到你户进行用电检查，发现你户（客户编号：7210100429）配电变压器（3 台容量为 400kVA 和 1 台容量为 315kVA）的高压计量柜的前门封印（SJL00014930）被人为破坏，计费电能表（NO：01026660；条形码 NO：SFF5104000864）的计量接线盒 C 相电压连接片被人为断开，计费电能表显示 C 相电流为 0，现场检测计费电能表 C 相同时失压失流，导致少计电量。即时报当地公安机关并拍照取证，现场对你户作停电处理。当时计费电能表抄见有功止码为 16 448.77</td></tr>
<tr><td>违约、窃电行为</td><td colspan="7">故意使供电企业用电计量装置不准或失效</td></tr>
<tr><td>计算方法及依据</td><td colspan="7">确定依据：计量自动化系统记录（2014 年 11 月 12 日计费电能表存在失压失流记录，直至 2014 年 11 月 17 日 C 相电压和电流数值均为 0）。
结论：现确定你户窃电时间由 2014 年 11 月 12 日至 2014 年 11 月 17 日，共 6 天。
根据现场计量装置检查情况，计费电能表 C 相失压失流，依据计量自动化系统召测数据分析，你户计费电能表（NO：01026660；条形码 NO：SFF5104000864）的 2014-11-12 功率因数：$COS(30°+\phi)=0.572$，即 $\phi=25.11°$，$COS\phi=0.905$。更正系数＝P 正确 /P 错误＝$UICOS\phi/[UICOS(\phi+30°)]=1.732\times0.905/0.572=2.74$，更正率＝更正系数 −1 ＝ 2.74−1 ＝ 1.74。2014 年 11 月 12 日计费电能表记录有功止码为 16 431.45，查处现场计费电能表抄见有功止码为 16 448.77，电流互感器变比为 100/5，电压互感器变比为 10 000/100。根据《供电营业规则》第一百零二条规定，窃电者应按所窃电量补交电费，并承担补交电费三倍的违约使用电费。具体计算如下：
1. 计费电能表已计收电量＝（16 448.77−16 431.45）×100/5×10 000/100 ＝ 34 640（kW.h）
2. 窃电电量＝已计收电量 × 更正率＝ 34 640×1.74 ＝ 60 274（kW.h）</td></tr>
</table>

（续）

计算方法及依据	3. 窃电电费＝ 60 274×0.6709 ＝ 40 437.83（元） 4. 城市建设附加费＝ 60 274×0.014 ＝ 843.84（元） 5. 违约使用电费＝ 40 437.83×3 ＝ 121 313.46（元） 6. 合计金额＝ 40 437.83 ＋ 843.84 ＋ 121 313.46 ＝ 162 595.12（元）	
	合计电费：162 595.13 元	大写金额：拾陆万贰仟伍佰玖拾伍圆壹角叁分

本次数据挖掘建模目标如下。

1）归纳出窃漏电用户的关键特征，构建窃漏电用户的识别模型。

2）利用实时监测数据，调用窃漏电用户识别模型实现实时诊断。

6.2　分析方法与过程

窃漏电用户在电力计量自动化系统的监控大用户中只占一小部分，同时某些大用户也不可能存在窃漏电行为，如银行、税务、学校和工商等非居民类别，故在数据预处理时有必要将这些类别用户剔除。系统中的用电负荷不能直接体现出用户的窃漏电行为，终端报警存在很多误报和漏报的情况，故需要进行数据探索和预处理，总结窃漏电用户的行为规律，再从数据中提炼出描述窃漏电用户的特征指标。最后结合历史窃漏电用户信息，整理出识别模型的专家样本数据集，再进一步构建分类模型，实现窃漏电用户的自动识别。

窃漏电用户识别流程如图 6-1 所示，主要包括以下步骤。

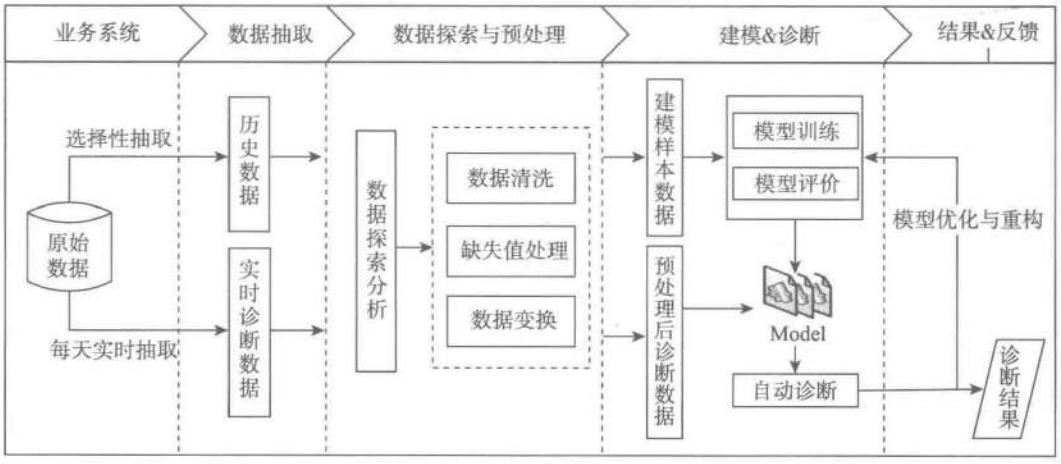


图 6-1　窃漏电用户识别流程

1）从电力计量自动化系统、营销系统有选择性地抽取部分大用户用电负荷、终端报警及违约窃电处罚信息等原始数据。

2）对样本数据探索分析，剔除不可能存在窃漏电行为行业的用户，即白名单用户，初步审视正常用户和窃漏电用户的用电特征。

3）对样本数据进行预处理，包括数据清洗、缺失值处理和数据变换。

4）构建专家样本集。

5）构建窃漏电用户识别模型。

6）在线监测用户用电负荷及终端报警，调用模型实现实时诊断。

6.2.1 数据抽取

与窃漏电相关的原始数据主要有用电负荷数据、终端报警数据、违约窃电处罚信息以及用户档案资料等，故进行窃漏电诊断建模时需从营销系统和计量自动化系统中抽取如下数据。

（1）从营销系统抽取的数据

1）用户基本信息：用户名称、用户编号、用电地址、用电类别、报装容量、计量方式、电流互感器变比、电压互感器变比。

2）违约、窃电处理记录。

3）计量方法及依据。

（2）从计量自动化系统采集的数据属性

1）实时负荷：时间点、计量点、总有功功率、A/B/C 相有功功率、A/B/C 相电流、A/B/C 相电压、A/B/C 相功率因数。

2）终端报警。

为了尽可能全面覆盖各种窃漏电方式，建模样本要包含不同用电类别的所有窃漏电用户及部分正常用户。窃漏电用户的窃漏电开始时间和结束时间是表征其窃漏电的关键时间节点，在这些时间节点上，用电负荷和终端报警等数据也会有一定的特征变化，故样本数据抽取时务必要包含关键时间节点前后一定范围的数据，并通过用户的负荷数据计算出当天的用电量，公式如下。

$$f_l = 0.25 \sum_{m_i \in l\text{天}} m_i \tag{6-1}$$

其中，f_l 为第 l 天的用电量，m_i 为第 l 天每隔 15 分钟的总有功功率，对其累加求和得到当天用电量。

基于此，本案例抽取某市近 5 年来所有的窃漏电用户有关数据和不同用电类别正常用电用户共 208 个用户的有关数据，时间为 2009 年 1 月 1 日至 2014 年 12 月 31 日，同时包含每天是否有窃漏电情况的标识。

6.2.2 数据探索分析

数据探索分析是对数据进行初步研究，发现数据的内在规律特征，有助于选择合适的数据预处理和数据分析技术。本案例主要采用分布分析和周期性分析等方法对电量数据进行数据探索分析。

1. 分布分析

对 2009 年 1 月 1 日至 2014 年 12 月 31 日共 5 年所有的窃漏用户进行分布分析，统计出各个用电类别的窃漏电用户分布情况，从图 6-2 可以发现非居民类别不存在窃漏电情况，故在接下来的分析中不考虑非居民类别的用电数据。

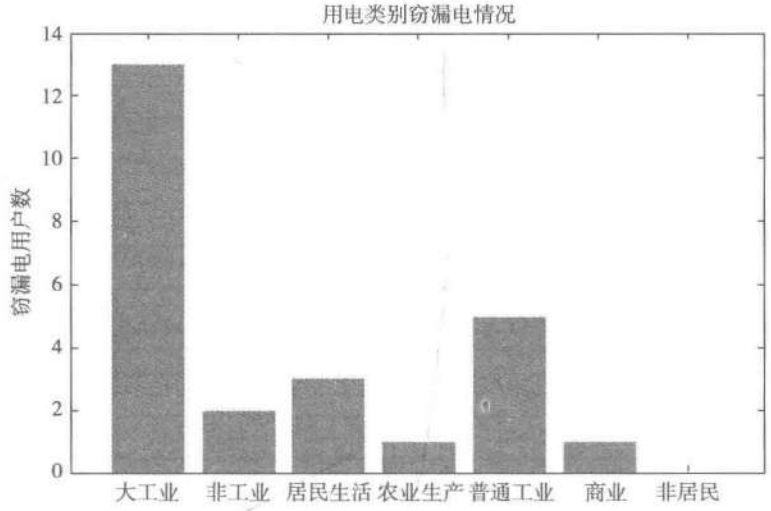


图 6-2　用电类别窃漏电情况图

2. 周期性分析

随机抽取一个正常用电用户和一个窃漏电用户，采用周期性分析对用电量进行探索。

（1）正常用电电量探索分析

正常用电量特征表现见图 6-3 和表 6-4。总体来看该用户用量比较平稳，没有太大的波动，这就是用户正常用电的电量指标特征。

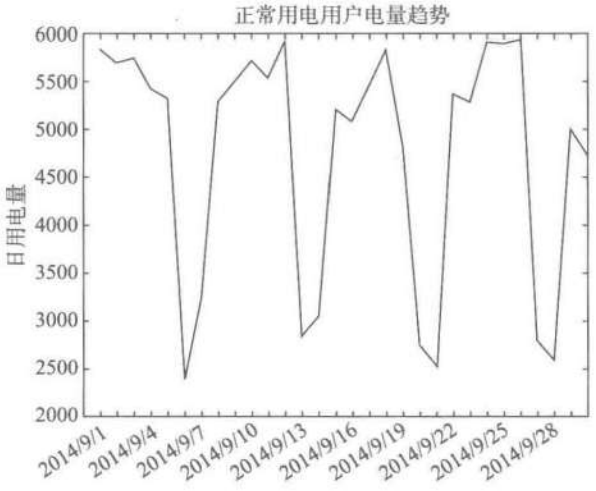


图 6-3　正常用电用户电量趋势图

表6-4 正常用电电量数据

日 期	日电量（kW）	日 期	日电量（kW）
2014/9/1	5840	2014/9/16	5072
2014/9/2	5704	2014/9/17	5480
2014/9/3	5754	2014/9/18	5832
2014/9/4	5431	2014/9/19	4816
2014/9/5	5322	2014/9/20	2748
2014/9/6	2392	2014/9/21	2536
2014/9/7	3225	2014/9/22	5384
2014/9/8	5296	2014/9/23	5288
2014/9/9	5488	2014/9/24	5928
2014/9/10	5713	2014/9/25	5896
2014/9/11	5542	2014/9/26	5952
2014/9/12	5928	2014/9/27	2792
2014/9/13	2848	2014/9/28	2600
2014/9/14	3048	2014/9/29	5000
2014/9/15	5216	2014/9/30	4704

（2）窃漏电用电电量探索分析

窃漏电用电量特征表现见图 6-4 和表 6-5。这里可以明显看出该用户用电量出现明显下降的趋势，这就是用户异常用电的电量指标特征。

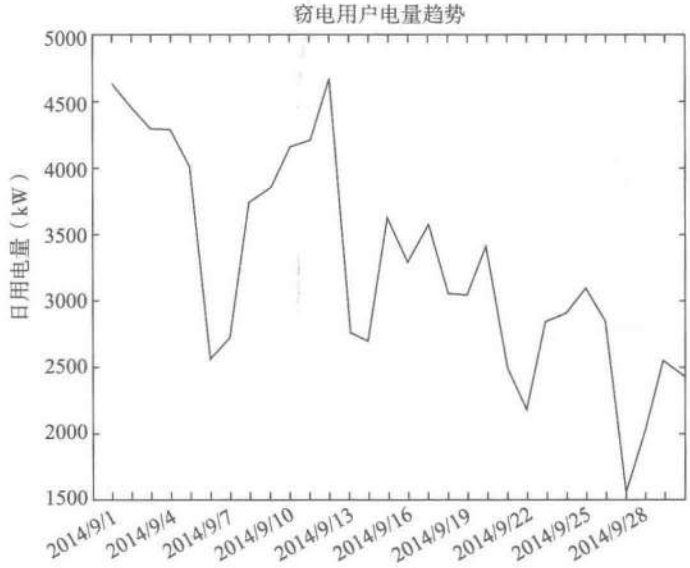


图 6-4 窃漏电用户电量趋势图

表6-5　窃漏电用电电量数据

日　　期	日电量（kW）	日　　期	日电量（kW）
2014/9/1	4640	2014/9/16	3260
2014/9/2	4450	2014/9/17	3590
2014/9/3	4300	2014/9/18	3040
2014/9/4	4290	2014/9/19	3030
2014/9/5	4010	2014/9/20	3410
2014/9/6	2560	2014/9/21	2490
2014/9/7	2720	2014/9/22	2160
2014/9/8	3740	2014/9/23	2850
2014/9/9	3850	2014/9/24	2900
2014/9/10	4150	2014/9/25	3090
2014/9/11	4210	2014/9/26	2840
2014/9/12	4680	2014/9/27	1530
2014/9/13	2760	2014/9/28	2020
2014/9/14	2680	2014/9/29	2540
2014/9/15	3630	2014/9/30	2440

分析结论：从图 6-4 看出，正常用电到窃漏电过程是用电量持续下降的过程，该用户从 2014 年 9 月 1 开始用电量下降，并且持续下降，这就是用户开始窃漏电时所表现出来的重要特征。

6.2.3　数据预处理

本案例主要从数据清洗、缺失值处理和数据变换等方面对数据进行预处理。

1. 数据清洗

数据清洗的主要目的是从业务以及建模的相关需要方面考虑，筛选出需要的数据。由于原始数据中并不是所有的数据都需要进行分析，因此需要在数据处理时，将赘余的数据进行过滤。本案例主要进行如下操作。

1）通过数据的探索分析，发现在用电类别中，非居民用电类别不可能存在漏电窃电的现象，需要将非居民用电类别的用电数据过滤掉。

2）结合本案例的业务，节假日用电量与工作日相比，会明显偏低。为了尽可能达到较好数据效果，过滤节假日的用电数据。

2. 缺失值处理

在原始计量数据，特别是用户电量抽取过程中，发现存在缺失的现象。若将这些值抛弃

掉，会严重影响供出电量的计算结果，最终导致日线损率数据误差很大。为了达到较好的建模效果，需要对缺失值处理。本案例采用拉格朗日插值法对缺失值进行插补。

选取数据中部分数据做为实例，如表 6-6 是三个用户一个月工作日的电量数据，对缺失值采用拉格朗日插值法进行插补。

表6-6 三个用户一个月工作日用电量数据

用户用电量（kW） 日 期	用 户 A	用 户 B	用 户 C
2014/9/1	235.8333	324.0343	478.3231
2014/9/2	236.2708	325.6379	515.4564
2014/9/3	238.0521	328.0897	517.0909
2014/9/4	235.9063		514.89
2014/9/5	236.7604	268.8324	
2014/9/8		404.048	486.0912
2014/9/9	237.4167	391.2652	516.233
2014/9/10	238.6563	380.8241	
2014/9/11	237.6042	388.023	435.3508
2014/9/12	238.0313	206.4349	487.675
2014/9/15	235.0729		
2014/9/16	235.5313	400.0787	660.2347
2014/9/17		411.2069	621.2346
2014/9/18	234.4688	395.2343	611.3408
2014/9/19	235.5	344.8221	643.0863
2014/9/22	235.6354	385.6432	642.3482
2014/9/23	234.5521	401.6234	
2014/9/24	236	409.6489	602.9347
2014/9/25	235.2396	416.8795	589.3457
2014/9/26	235.4896		556.3452
2014/9/29	236.9688		538.347

数据详见：demo/data/missing_data.xls

拉格朗日插值法补值，具体方法如下。

首先从原始数据集中确定因变量和自变量，取出缺失值前后 5 个数据（前后数据中遇到数据不存在或者为空的，直接将数据舍去，将仅有的数据组成一组），根据取出来的 10 个数据组成一组。然后采用拉格朗日多项式插值公式

$$L_n(x) = \sum_{i=0}^{n} l_i(x) y_i \tag{6-2}$$

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{6-3}$$

其中，x 为缺失值对应的下标序号，$L_n(x)$ 为缺失值的插值结果，x_i 为非缺失值 y_i 的下标序号。对全部缺失数据依次进行插补，直到不存在缺失值为止。数据插补代码如代码清单 6-1 所示。

代码清单6-1　拉格朗日插值代码

```
#-*- coding: utf-8 -*-
#拉格朗日插值代码
import pandas as pd #导入数据分析库Pandas
from scipy.interpolate import lagrange #导入拉格朗日插值函数

inputfile = '../data/missing_data.xls' #输入数据路径,需要使用Excel格式;
outputfile = '../tmp/missing_data_processed.xls' #输出数据路径,需要使用Excel格式

data = pd.read_excel(inputfile, header=None) #读入数据

#自定义列向量插值函数
#s为列向量, n为被插值的位置, k为取前后的数据个数, 默认为5
def ployinterp_column(s, n, k=5):
    y = s[list(range(n-k, n)) + list(range(n+1, n+1+k))] #取数
    y = y[y.notnull()] #剔除空值
    return lagrange(y.index, list(y))(n) #插值并返回插值结果

#逐个元素判断是否需要插值
for i in data.columns:
    for j in range(len(data)):
        if (data[i].isnull())[j]: #如果为空即插值。
            data[i][j] = ployinterp_column(data[i], j)

data.to_excel(outputfile, header=None, index=False) #输出结果
```

代码详见：demo/code/Lagrange_interpolation.py

根据代码清单 6-1 补全的数据见表 6-7，斜体加粗表示补全的数据。

表6-7　用户电量补全数据

用户用电量（kW） 日　期	用　户　A	用　户　B	用　户　C
2014/9/4	235.9063	***203.4621***	514.89
2014/9/5	236.7604	268.8324	***493.3526***
2014/9/8	***237.1512***	404.048	486.0912
2014/9/10	238.6563	380.8241	***516.233***
2014/9/15	235.0729	***237.3481***	***609.1936***
2014/9/17	***235.315***	411.2069	621.2346

（续）

用户用电量（kW） / 日　期	用　户　A	用　户　B	用　户　C
2014/9/23	234.5521	401.6234	***618.1972***
2014/9/26	235.4896	***420.7486***	556.3452
2014/9/29	236.9688	***408.9632***	538.347

3. 数据变换

通过电力计量系统采集的电量、负荷，虽然在一定程度上能反映用户窃漏电行为的某些规律，但要作为构建模型的专家样本，特征不明显，需要进行重新构造。基于数据变换，得到新的评价指标来表征窃漏电行为所具有的规律，其评价指标体系如图 6-5 所示。

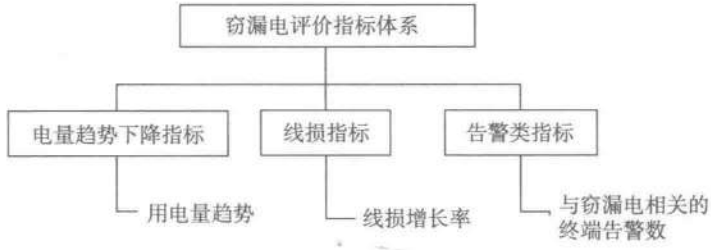


图 6-5　窃漏电评价指标体系

窃漏电评价指标如下。

（1）电量趋势下降指标

由 6.2.2 节的周期性分析可以发现，正常用户的用电量较为平稳，窃漏电用户的用电量呈现下降的趋势，然后趋于平缓，因此可考虑前后几天作为统计窗口期，考虑期间的下降趋势，利用电量做直线拟合得到的斜率作为衡量，如果斜率随时间不断下降，那该用户的窃漏电可能性就很大，如图 6-6 所示。第一幅图展示了每天的用电量，其他图表示了随着时间推移在各自统计窗口期以用电量做直线拟合的斜率，可以看出斜率随着时间逐步下降。

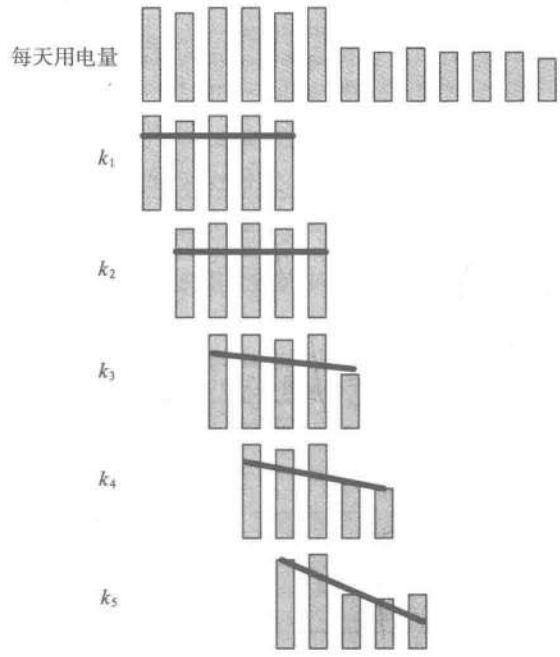


图 6-6　电量趋势下降示意图

对统计当天设定前后 5 天为统计窗口期，计算这 11 天内的电量趋势下降情况。首先计算这 11 天中每天的电量趋势，其中第 i 天的用电量趋势是考虑前后 5 天期间的用电量斜率，即

$$k_i = \frac{\sum_{l=i-5}^{i+5} (f_l - \bar{f})(l - \bar{l})}{\sum_{l=i-5}^{i+5} (l - \bar{l})^2} \tag{6-4}$$

其中，$\bar{f} = \frac{1}{11}\sum_{l=i-5}^{i+5} f_l$，$\bar{l} = \frac{1}{11}\sum_{l=i-5}^{i+5} l$，$k_i$为第 i 天的电量趋势，f_l为第 l 天的用电量。

若电量趋势为不断下降的，则认为具有一定的窃电嫌疑，故计算这 11 天内，当天比前一天用电量趋势为递减的天数，即设有

$$D(i) = \begin{cases} 1, & k_i < k_{i-1} \\ 0, & k_i \geqslant k_{i-1} \end{cases} \tag{6-5}$$

则这 11 天内的电量趋势下降指标为

$$T = \sum_{n=i-4}^{i+5} D(n) \tag{6-6}$$

（2）线损指标

线损率是用于衡量供电线路的损失比例，同时可结合线户拓扑关系（如图 6-7）计算出用户所属线路在当天的线损率，一条线路上同时供给多个用户，若第 l 天的线路供电量为 s_l，线路上各个用户的总用电量为$\sum_m f_l^{(m)}$，线路的线损率公式为

$$t_l = \frac{s_l - \sum_m f_l^{(m)}}{s_l} \times 100\% \tag{6-7}$$

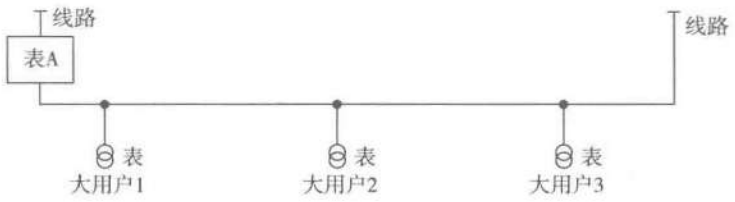


图 6-7　线路与大用户的拓扑关系示意图

线路的线损率可作为用户线损率的参考值，若用户发生窃漏电，则当天的线损率会上升，但由于用户每天的用电量存在波动，单纯以当天线损率上升了作为窃漏电特征则误差过大，所以考虑前后几天的线损率平均值，判断其增长率是否大于 1%，若线损率的增长率大于 1% 则具有窃漏电的可能性。

对统计当天设定前后 5 天为统计窗口期，首先分别统计当天与前 5 天之间的线损率平均值 V_i^1和统计当天与后 5 天之间的线损率平均值 V_i^2，若 V_i^1比 V_i^2 的增长率大于 1%，则认为具有一定的窃电嫌疑，故定义线损指标

$$E(i) = \begin{cases} 1, \dfrac{V_i^1 - V_i^2}{V_i^2} > 1\% \\ 0, \dfrac{V_i^1 - V_i^2}{V_i^2} \leqslant 1\% \end{cases} \tag{6-8}$$

（3）告警类指标

与窃漏电相关的终端报警主要有电压缺相、电压断相、电流反极性等告警，计算发生与窃漏电相关的终端报警的次数总和，作为告警类指标。

6.2.4 构建专家样本

对 2009 年 1 月 1 日至 2014 年 12 月 31 日所有窃漏电用户及正常用户的电量、告警及线损数据和该用户在当天是否窃漏电的标识，按窃漏电评价指标进行处理并选取其中 291 个样本数据，得到专家样本库，部分数据如表 6-8 所示。

表6-8 专家样本数据

时 间	用户编号	电量趋势下降指标	线损指标	告警类指标	是否窃漏电
2014 年 9 月 6 日	9900667154	4	1	1	1
2014 年 9 月 20 日	9900639431	4	0	4	1
2014 年 9 月 17 日	9900585516	2	1	1	1
2014 年 9 月 14 日	9900531154	9	0	0	0
2014 年 9 月 17 日	9900491050	3	1	0	0
2014 年 9 月 13 日	9900461501	2	0	0	0
2014 年 9 月 22 日	9900412593	5	0	2	1
2014 年 9 月 20 日	9900366180	3	1	3	1
2014 年 9 月 19 日	9900322960	3	0	0	0
2014 年 9 月 9 日	9900254673	4	1	0	0
2014 年 9 月 18 日	9900196505	10	1	2	1
2014 年 9 月 16 日	9900145248	10	1	3	1
2014 年 9 月 6 日	9900137535	2	0	3	0
2014 年 9 月 7 日	9900064537	4	0	2	0
2014 年 9 月 9 日	9110103867	3	0	0	0
2014 年 9 月 23 日	9010100689	0	0	3	0
2014 年 9 月 21 日	8910101840	9	0	3	1
2014 年 9 月 11 日	8910101209	0	0	2	0
2014 年 9 月 19 日	8910101132	8	1	4	1

（续）

时　间	用户编号	电量趋势下降指标	线损指标	告警类指标	是否窃漏电
2014 年 9 月 19 日	8910100309	2	0	4	0
2014 年 9 月 9 日	8810101463	3	0	1	0
2014 年 9 月 9 日	8710100857	7	0	0	0

数据详见：demo/data/model.xls

6.2.5　模型构建

1. 构建窃漏电用户识别模型

在专家样本准备完成后，需要划分测试样本和训练样本，随机选取 20% 作为测试样本，剩下的作为训练样本。窃漏电用户识别可通过构建分类预测模型来实现，比较常用的分类预测模型有 LM 神经网络和 CART 决策树，各个模型都有各自的优点，故采用这两种方法构建窃漏电用户识别，并从中选择最优的分类模型。构建 LM 神经网络和 CART 决策树模型时输入项包括电量趋势下降指标、线损类指标和告警类指标，输出项为窃漏电标识。

（1）数据划分

对专家样本随机选取 20% 作为测试样本，剩下的 80% 作为训练样本。其代码如代码清单 6-2 所示。

代码清单6-2　原始数据分为训练数据测试数据

```
#-*- coding: utf-8 -*-
import pandas as pd #导入数据分析库
from random import shuffle #导入随机函数shuffle，用来打算数据

datafile = '../data/model.xls' #数据名
data = pd.read_excel(datafile) #读取数据，数据的前三列是特征，第四列是标签
data = data.as_matrix() #将表格转换为矩阵
shuffle(data) #随机打乱数据

p = 0.8 #设置训练数据比例
train = data[:int(len(data)*p),:] #前80%为训练集
test = data[int(len(data)*p):,:] #后20%为测试集
```

（2）LM 神经网络

使用 Keras 库为我们建立神经网络模型。设定 LM 神经网络的输入节点数为 3，输出节点数为 1，隐层节点数为 10，使用 Adam 方法求解。对于激活函数，在隐藏层使用 Relu（x）= max（x, 0）作为激活函数，实验表明该激活函数能够大幅提高模型的准确率。训练样本建模的混淆矩阵见图 6-8，可以算得分类准确率为（161+58）/（161+58+6+7）= 94.4%，正常用户被误判为窃漏电用户占正常用户的 7/（161+7）= 4.2%，窃漏电用户被误判为正常用户占正常窃漏电用户的 6/（6+58）= 9.4%。构建 LM 神经网络模型的代码如代码清单 6-3 所示。

代码清单6-3 构建LM神经网络模型代码（接6-2）

```
#构建LM神经网络模型
from keras.models import Sequential #导入神经网络初始化函数
from keras.layers.core import Dense, Activation #导入神经网络层函数、激活函数

netfile = '../tmp/net.model' #构建的神经网络模型存储路径

net = Sequential() #建立神经网络
net.add(Dense(3, 10)) #添加输入层（3节点）到隐藏层（10节点）的连接
net.add(Activation('relu')) #隐藏层使用relu激活函数
net.add(Dense(10, 1)) #添加隐藏层（10节点）到输出层（1节点）的连接
net.add(Activation('sigmoid')) #输出层使用sigmoid激活函数
net.compile(loss = 'binary_crossentropy', optimizer = 'adam', class_mode = "binary")
    #编译模型，使用adam方法求解

net.fit(train[:,:3], train[:,3], nb_epoch=1000, batch_size=1) #训练模型，循环1000次
net.save_weights(netfile) #保存模型

predict_result = net.predict_classes(train[:,:3]).reshape(len(train)) #预测结果变形
'''这里要提醒的是，keras用predict给出预测概率，predict_classes才是给出预测类别，而且两者的
   预测结果都是n x 1维数组，而不是通常的 1 x n'''

from cm_plot import * #导入自行编写的混淆矩阵可视化函数
cm_plot(train[:,3], predict_result).show() #显示混淆矩阵可视化结果
```

代码详见：demo/code/lm_model.py

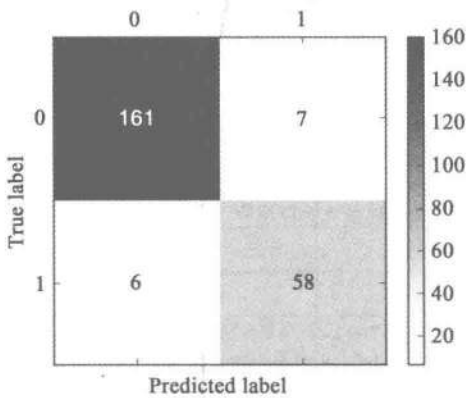


图 6-8 利用训练样本构建 LM 神经网络的混淆矩阵

（3）CART 决策树

通过 Scikit-Lrean 利用训练样本构建 CART 决策树模型，得到混淆矩阵如图 6-9 所示，分类准确率为 (160+56)/(160+56+3+13) = 93.1%，正常用户被误判为窃漏电用户占正常用户的 13/(13+160) = 7.5%，窃漏电用户被误判为正常用户占正常窃漏电用户的 3/(3+56) = 5.1%。构建决策树的代码如代码清单 6-4 所示。

代码清单6-4　构建CART决策树模型代码（接6-2）

```
#构建CART决策树模型
from sklearn.tree import DecisionTreeClassifier #导入决策树模型

treefile = '../tmp/tree.pkl' #模型输出名字
tree = DecisionTreeClassifier() #建立决策树模型
tree.fit(train[:,:3],train[:,3]) #训练

#保存模型
from sklearn.externals import joblib
joblib.dump(tree, treefile)

from cm_plot import * #导入自行编写的混淆矩阵可视化函数
cm_plot(train[:,3], tree.predict(train[:,:3])).show() #显示混淆矩阵可视化结果
#注意到Scikit-Learn使用predict方法直接给出预测结果。
```

完整的代码详见：demo/code/dt_model.py

2. 模型评价

对于训练样本，LM 神经网络和 CART 决策树的分类准确率相差不大，分别为 94% 和 93%。为了进一步评估模型分类的性能，故利用测试样本对两个模型进行评价，采用 ROC 曲线评价方法进行评估，一个优秀分类器所对应的 ROC 曲线应该是尽量靠近左上角的。分别画出 LM 神经网络和 CART 决策树在测试样本下的 ROC 曲线，如图 6-10 和图 6-11 所示。LM 神经网络和 CART 决策树对测试数据集的测试代码如代码清单 6-5、代码清单 6-6 所示。

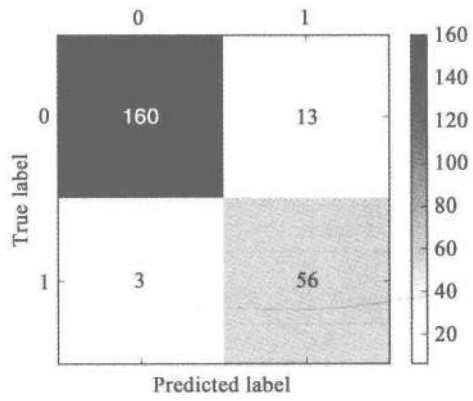


图 6-9　利用训练样本构建 CART 决策树的混淆矩阵

代码清单6-5　绘制决策树模型的ROC曲线（接6-3）

```
from sklearn.metrics import roc_curve #导入ROC曲线函数

predict_result = net.predict(test[:,:3]).reshape(len(test)) #预测结果变形
fpr, tpr, thresholds = roc_curve(test[:,3], predict_result, pos_label=1)
plt.plot(fpr, tpr, linewidth=2, label = 'ROC of LM') #作出ROC曲线
plt.xlabel('False Positive Rate') #坐标轴标签
plt.ylabel('True Positive Rate') #坐标轴标签
plt.ylim(0,1.05) #边界范围
plt.xlim(0,1.05) #边界范围
plt.legend(loc=4) #图例
plt.show() #显示作图结果
```

代码详见：demo/code/lm_model.py

代码清单6-6 绘制决策树模型的ROC曲线（接6-4）

```
from sklearn.metrics import roc_curve #导入ROC曲线函数

fpr, tpr, thresholds = roc_curve(test[:,3], tree.predict_proba(test[:,:3])[:,1],
    pos_label=1)
plt.plot(fpr, tpr, linewidth=2, label = 'ROC of CART') #作出ROC曲线
plt.xlabel('False Positive Rate') #坐标轴标签
plt.ylabel('True Positive Rate') #坐标轴标签
plt.ylim(0,1.05) #边界范围
plt.xlim(0,1.05) #边界范围
plt.legend(loc=4) #图例
plt.show() #显示作图结果
```

代码详见：demo/code/dt_model.py

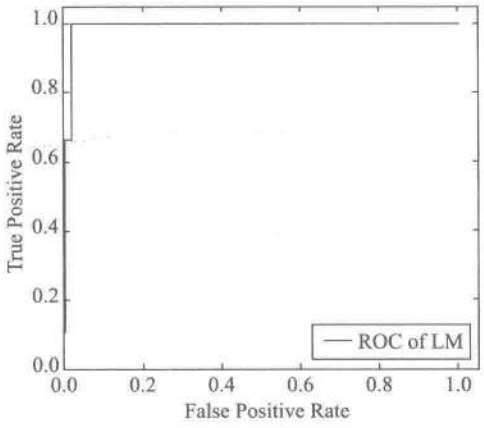


图 6-10 LM 神经网络在测试样本下的 ROC 曲线

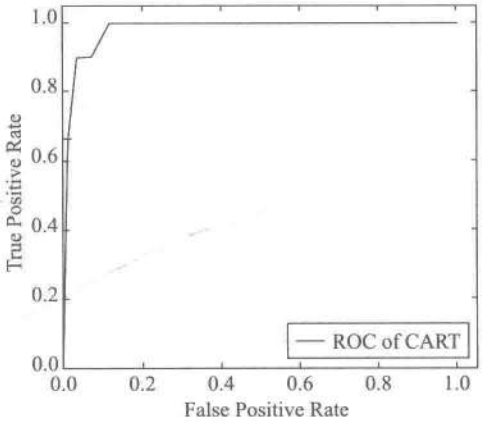


图 6-11 CART 决策树在测试样本下的 ROC 曲线

经过对比发现 LM 神经网络的 ROC 曲线比 CART 决策树的 ROC 曲线更加靠近单位方形的左上角，LM 神经网络 ROC 曲线下的面积更大，说明 LM 神经网络模型的分类性能较好，能应用于窃漏电用户识别。

3. 进行窃漏电诊断

在线监测用户用电负荷及终端报警数据，并经过 6.2.3 节的处理，得到模型输入数据，利用构建好的窃漏电用户识别模型计算用户的窃漏电诊断结果，实现窃漏电用户实时诊断，并与实际稽查结果作对比，见表 6-9。可以发现，正确识别出窃漏电用户有 10 个，错误地判断用户为窃漏电用户有 1 个，诊断结果没发现窃漏电用户有 4 个，整体来看窃漏电诊断的准确率是比较高的。下一步的工作是针对漏判的用户，研究其在窃漏电期间的用电行为，优化模型的特征，提高识别的准确率。

表6-9　窃漏电诊断结果与实际稽查结果作对比

客户编号	客户名称	窃电开始日期	结　果
7110100608	某塑胶制品厂	2014.6.2	正确诊断
9900508537	某经济合作社	2014.8.20	正确诊断
9900531988	某模具有限公司	2014.8.21	正确诊断
8210101409	某科技有限公司	2014.8.10	正确诊断
8910100571	某股份经济合作社	2014.2.23	漏判
8210100795	某表壳加工厂	2014.6.1	正确诊断
9900287332	某电子有限公司	2014.5.15	漏判
6710100757	某镇某经济联合社	2014.2.21	漏判
9900378363	某装饰材料有限公司	2014.7.6	误判
9900145275	某实业投资有限公司	2014.11.3	正确诊断
8410101508	某玩具厂有限公司	2014.9.1	正确诊断
9900150075	某镇某经济联合社	2014.4.14	漏判
8010106555	某电子有限公司	2014.5.19	正确诊断
7410101282	某投资有限公司	2014.2.8	正确诊断
8410101060	某电子有限公司	2014.5.4	正确诊断

6.3　上机实验

1. 实验目的

- 掌握拉格朗日插值法进行缺失值处理的方法。
- 掌握 LM 神经网络和 CART 决策树构建分类模型的方法。

2. 实验内容

- 用户的用电数据存在缺失值，数据见“ test/data/missing_data.xls”，利用拉格朗日插值算法补全数据。
- 对所有窃漏电用户及正常用户的电量、告警及线损数据和该用户在当天是否窃漏电的标识，按窃漏电评价指标进行处理并选取其中 291 个样本数据，得到专家样本，数据见“ test/data/model.xls”，分别使用 LM 神经网络和 CART 决策树实现分类预测模型，利用混淆矩阵和 ROC 曲线对模型进行评价。

数据 80% 作为训练样本，剩下的 20% 作为测试样本。

3. 实验方法与步骤

实验一

1）打开 Python 软件，把“test/data/missing_data.xls”数据放入当前工作目录。

2）使用 Pandas 把数据读入当前工作目录。

3）针对读入的数据的每一列，进行编程。编程主要参考第 4 章的拉格朗日插值算法，具体步骤如下。

- 针对每列数据的每一个缺失值，逐个进行补数（这样可以在连续两个缺失值的情况下，使用前面一个已经补数的值来再次补数后面的一个值）。
- 针对一个缺失值，构造参考组。选取前面 5 个作为前参考组，后面 5 个为后参考组。如果前参考组或后参考组不足 5 个，则按实际个数构造参考组。
- 确认缺失值在参考组中的相对位置，然后使用拉格朗日插值进行缺失值插值。
- 根据插值后的值更新原始数据中相应位置的值。

4）编写并运行程序后，查看插值补数的值是否和给定的参考值一致。

实验二

1）把经过预处理的专家样本数据“test/data/model.xls”数据放入当前工作目录，并使用 Pandas 读入当前工作空间。

2）把工作空间的建模数据随机分为两部分，一部分用于训练，一部分用于测试。

3）使用 Scikit-Lrean 库的 sklearn.tree 的 DecisionTreeClassifier 函数以及训练数据构建 CART 决策树模型，使用 predict 函数和构建的 CART 决策树模型分别对训练和测试数据进行分类，并与真实值进行对比，得到模型正确率，同时使用 sklearn.metrics 的 confusion_matrix 和 roc_curve 函数画混淆矩阵和 ROC 曲线图（参考本章代码）。

4）使用 Keras 库以及训练数据构建 LM 神经网络模型，使用 predict 函数和构建的神经网络模型分别对训练和测试数据进行分类，参考第 3）步得到模型正确率、混淆矩阵和 ROC 曲线图。

5）对比分析 CART 决策树模型和 LM 神经网络模型针对专家样本数据处理结果的好坏。

4. 思考与实验总结

1）在进行插值补数选取参考值时，为什么选择 10 个为一组？

2）在 Pandas 中，Series 对象自带了的缺失值补数方法 .interpolate()，请尝试使用它，并将它和拉格朗日插值补数方法进行对比。

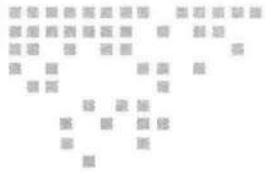
6.4 拓展思考

目前企业偷漏税现象泛滥，严重影响国家的经济基础。为了维护国家的权力与利益，应该加大对企业偷漏税行为的防范。如何用数据挖掘的思想，智能的识别企业偷漏税行为，有力地打击企业偷漏税的违法行为，维护国家的经济损失和社会秩序，是大家要研究的课题。

汽车销售行业，通常是指销售汽车整车的行业。汽车销售行业在税收上存在少开发票金额、少计收入，上牌、按揭、保险等一条龙服务未入账，不及时确认保修索赔款等多种情况，导致政府损失大量税收。汽车销售企业的部分经营指标能在一定程度上评估企业的偷漏税倾向，附件数据（见：拓展思考 / 拓展思考样本数据 .xls）提供了汽车销售行业纳税人的各个属性和是否偷漏税标识，请结合汽车销售行业纳税人的各个属性，总结衡量纳税人的经营特征，建立偷漏税行为识别模型，识别偷漏税纳税人。

6.5　小结

本章结合窃漏电用户识别的案例，重点介绍了数据挖掘算法中 LM 神经网络和 CART 决策树算法在实际案例中的应用。研究窃漏电用户的行为特征，总结出窃漏电用户的特征指标，对比 LM 神经网络和 CART 决策树算法在窃漏电用户的识别效果，从中选取最优模型进行窃漏电诊断，并详细地描述了数据挖掘的整个过程，并对其相应的算法给出了 Python 上机实验步骤。



Chapter 7 第 7 章

航空公司客户价值分析

7.1 背景与挖掘目标

信息时代的来临使得企业营销焦点从产品中心转变为客户中心，客户关系管理成为企业的核心问题。客户关系管理的关键问题是客户分类，通过客户分类，区分无价值客户、高价值客户，企业针对不同价值的客户制定优化的个性化服务方案，采取不同营销策略，将有限营销资源集中于高价值客户，实现企业利润最大化目标。准确的客户分类结果是企业优化营销资源分配的重要依据，客户分类越来越成为客户关系管理中亟待解决的关键问题之一。

面对激烈的市场竞争，各个航空公司都推出了更优惠的营销方式来吸引更多的客户，国内某航空公司面临着旅客流失、竞争力下降和航空资源未充分利用等经营危机。通过建立合理的客户价值评估模型，对客户进行分群，分析比较不同客户群的客户价值，并制定相应的营销策略，对不同的客户群提供个性化的客户服务是必须和有效的。目前该航空公司已积累了大量的会员档案信息和其乘坐航班记录，经加工后得到表 7-1 所示的部分数据信息。

请根据这些数据（见表 7-2）实现以下目标。

1）借助航空公司客户数据，对客户进行分类。

2）对不同的客户类别进行特征分析，比较不同类客户的客户价值。

3）对不同价值的客户类别提供个性化服务，制定相应的营销策略。

表7-1 航空信息属性表

	属性名称	属性说明
客户基本信息	MEMBER_NO	会员卡号
	FFP_DATE	入会时间

（续）

	属性名称	属性说明
客户基本信息	FIRST_FLIGHT_DATE	第一次飞行日期
	GENDER	性别
	FFP_TIER	会员卡级别
	WORK_CITY	工作地城市
	WORK_PROVINCE	工作地所在省份
	WORK_COUNTRY	工作地所在国家
	AGE	年龄
乘机信息	FLIGHT_COUNT	观测窗口内的飞行次数
	LOAD_TIME	观测窗口的结束时间
	LAST_TO_END	最后一次乘机时间至观测窗口结束时长
	AVG_DISCOUNT	平均折扣率
	SUM_YR	观测窗口的票价收入
	SEG_KM_SUM	观测窗口的总飞行公里数
	LAST_FLIGHT_DATE	末次飞行日期
	AVG_INTERVAL	平均乘机时间间隔
	MAX_INTERVAL	最大乘机间隔
积分信息	EXCHANGE_COUNT	积分兑换次数
	EP_SUM	总精英积分
	PROMOPTIVE_SUM	促销积分
	PARTNER_SUM	合作伙伴积分
	POINTS_SUM	总累计积分
	POINT_NOTFLIGHT	非乘机的积分变动次数
	BP_SUM	总基本积分

观测窗口：以过去某个时间点为结束时间，某一时间长度作为宽度，得到历史时间范围内的一个时间段。

表7-2　航空信息数据表

MEMBER_NO	FFP_DATE	FIRST_FLIGI	GENDE	FFP_TIER	WORK_CITY	WORK_PROVIN	WORK	AGE	LOAD_TIME	FLIGHT_COUNT	BP_SUM
289047040	2013/03/16	2013/04/28	男	6			US	56	2014/03/31	14	147 158
289053451	2012/06/26	2013/05/16	男	6	乌鲁木齐	新疆	CN	50	2014/03/31	65	112 582
289022508	2009/12/08	2010/02/05	男	5		北京	CN	34	2014/03/31	33	77 475
289004181	2009/12/10	2010/10/19	男	4	S.P.S	CORTES	HN	45	2014/03/31	6	76 027
289026513	2011/08/25	2011/08/25	男	6	乌鲁木齐	新疆	CN	47	2014/03/31	22	70 142
289027500	2012/09/26	2013/06/01	男	5	北京	北京	CN	36	2014/03/31	26	63 498
289058898	2010/12/27	2010/12/27	男	4	ARCADIA	CA	US	35	2014/03/31	5	62 810
289037374	2009/10/21	2009/10/21	男	4	广州	广东	CN	34	2014/03/31	4	60 484
289036013	2010/04/15	2013/06/02	女	6	广州	广东	CN	54	2014/03/31	25	59 357

（续）

MEMB-ER_NO	FFP_ DATE	FIRST_FLIGI	GENDE	FFP_ TIER	WORK_ CIT Y	WORK_ PROVIN	WORK	AGE	LOAD_ TIME	FLIGHT _COUNT	BP_ SUM
289046087	2007/01/26	2013/04/24	男	6	.	天津	CN	47	2014/03/31	36	55 562
289062045	2006/12/26	2013/04/17	女	5	长春市	吉林省	CN	55	2014/03/31	49	54 255
289061968	2011/08/15	2011/08/20	男	6	沈阳	辽宁	CN	41	2014/03/31	51	53 926
289022276	2009/08/27	2013/04/18	男	5	深圳	广东	CN	41	2014/03/31	62	49 224
289056049	2013/03/18	2013/07/28	男	4	Simi Valley		US	54	2014/03/31	12	49 121
289000500	2013/03/12	2013/04/01	男	5	北京	北京	CN	41	2014/03/31	65	46 618
289037025	2007/02/01	2011/08/22	男	6	昆明	云南	CN	57	2014/03/31	28	45 531
289029053	2004/12/18	2005/05/06	男	4			CN	46	2014/03/31	6	41 872
289048589	2008/08/15	2008/08/15	男	5	NUMAZU		CN	60	2014/03/31	15	41 610
289005632	2011/08/09	2011/08/09	男	5	南阳县	河南	CN	47	2014/03/31	6	40 726
289041886	2011/11/23	2013/09/17	女	5	温州	浙江	CN	42	2014/03/31	7	40 589
289049670	2010/04/18	2010/04/18	男	5	广州	广东	CN	39	2014/03/31	35	39 973
289020872	2008/06/22	2013/06/30	男	6	.	北京	CN	47	2014/03/31	33	39 737
289021001	2008/03/09	2013/07/10	男	6			CN	47	2014/03/31	40	39 584
289041371	2011/10/15	2013/09/04	男	6	武汉	湖北	CN	56	2014/03/31	30	38 089
289062046	2007/10/19	2007/10/19	男	5	上海	上海	CN	39	2014/03/31	48	37 188
289037246	2007/08/30	2013/04/18	男	6	贵阳	贵州	CN	47	2014/03/31	40	36 471
289045852	2006/08/16	2006/11/08	男	4	ARCADIA	CA	US	69	2014/03/31	8	35 707

数据详见：示例程序 /data/air_data.csv

7.2 分析方法与过程

本案例的目标是客户价值识别，即通过航空公司客户数据识别不同价值的客户。识别客户价值应用最广泛的模型是通过 3 个指标（最近消费时间间隔（Recency）、消费频率（Frequency）和消费金额（Monetary））来进行客户细分，识别出高价值的客户，简称 RFM 模型[15]。

在 RFM 模型中，消费金额表示在一段时间内，客户购买该企业产品金额的总和。由于航空票价受到运输距离、舱位等级等多种因素影响，同样消费金额的不同旅客对航空公司的价值是不同的。例如，一位购买长航线、低等级舱位票的旅客与一位购买短航线、高等级舱位票的旅客相比，后者对于航空公司而言价值可能更高。因此，这个指标并不适用于航空公司的客户价值分析[15]。我们选择客户在一定时间内累积的飞行里程 M 和客户在一定时间内乘坐舱位所对应的折扣系数的平均值 C 两个指标代替消费金额。此外，考虑航空公司会员入会时间的长短在一定程度上能够影响客户价值，所以在模型中增加客户关系长度 L，作为区分客户的另一指标。

本案例将客户关系长度 L、消费时间间隔 R、消费频率 F、飞行里程 M 和折扣系数的平均值 C 五个指标作为航空公司识别客户价值指标（见表 7-3），记为 LRFMC 模型。

表7-3　指标含义

模　型	L	R	F	M	C
航空公司 LRFMC 模型	会员入会时间距观测窗口结束的月数	客户最近一次乘坐公司飞机距观测窗口结束的月数	客户在观测窗口内乘坐公司飞机的次数	客户在观测窗口内累计的飞行里程	客户在观测窗口内乘坐舱位所对应的折扣系数的平均值

针对航空公司 LRFMC 模型，如果采用传统 RFM 模型分析的属性分箱方法，如图 7-1 所示[16]（它是依据属性的平均值进行划分，其中大于平均值的表示为↑，小于平均值的表示为↓），虽然也能够识别出最有价值的客户，但是细分的客户群太多，提高了针对性营销的成本。因此，本案例采用聚类的方法识别客户价值。通过对航空公司客户价值的 LRFMC 模型的五个指标进行 K-Means 聚类，识别出最有价值客户。

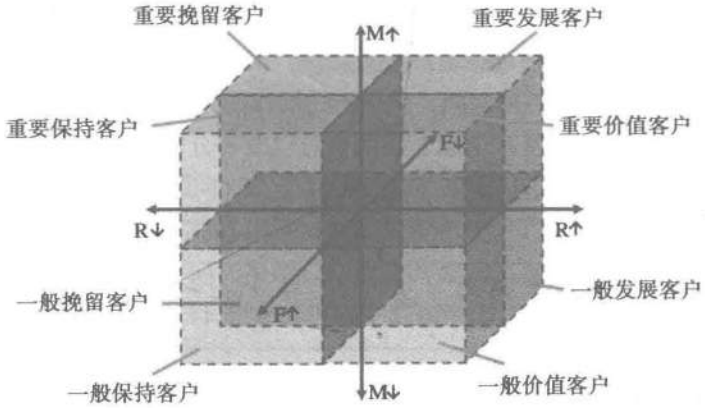


图 7-1　RFM 模型分析

本案例航空客户价值分析的总体流程如图 7-2 所示。

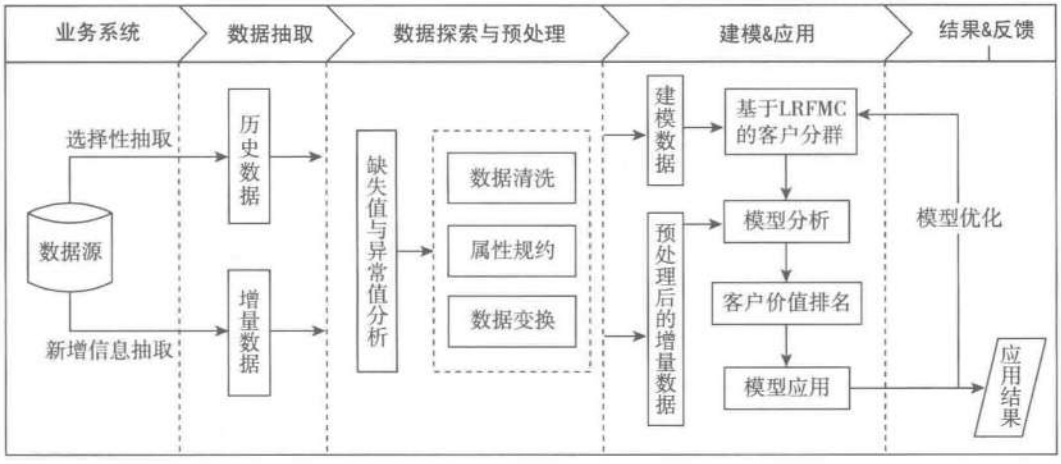


图 7-2　航空客运数据挖掘建模总体流程

航空客运信息挖掘主要包括以下步骤。

1）从航空公司的数据源中进行选择性抽取与新增数据抽取分别形成历史数据和增量数据。

2）对步骤 1）中形成的两个数据集进行数据探索分析与预处理，包括数据缺失值与异常值的探索分析，数据的属性规约、清洗和变换。

3）利用步骤 2）中形成的已完成数据预处理的建模数据，基于旅客价值 LRFMC 模型进行客户分群，对各个客户群进行特征分析，识别出有价值的客户。

4）针对模型结果得到不同价值的客户，采用不同的营销手段，提供定制化的服务。

7.2.1 数据抽取

以 2014-03-31 为结束时间，选取宽度为两年的时间段作为分析观测窗口，抽取观测窗口内有乘机记录的所有客户的详细数据形成历史数据。对于后续新增的客户详细信息，以后续新增数据中最新的时间点作为结束时间，采用上述同样的方法进行抽取，形成增量数据。

从航空公司系统内的客户基本信息、乘机信息以及积分信息等详细数据中，根据末次飞行日期（LAST_FLIGHT_DATE），抽取 2012-04-01 至 2014-03-31 内所有乘客的详细数据，总共有 62 988 条记录。其中包含了会员卡号、入会时间、性别、年龄、会员卡级别、工作地城市、工作地所在省份、工作地所在国家、观测窗口结束时间、观测窗口乘机积分、飞行公里数、飞行次数、飞行时间、乘机时间间隔和平均折扣率等 44 个属性。

7.2.2 数据探索分析

本案例的探索分析是对数据进行缺失值分析与异常值分析，分析出数据的规律以及异常值。通过对数据观察发现原始数据中存在票价为空值，票价最小值为 0、折扣率最小值为 0、总飞行公里数大于 0 的记录。票价为空值的数据可能是客户不存在乘机记录造成，其他的数据可能是客户乘坐 0 折机票或者积分兑换产生的。

查找每列属性观测值中空值个数、最大值、最小值的 Python 代码如代码清单 7-1 所示。

代码清单7-1 数据探索分析代码

```
#-*- coding: utf-8 -*-
#对数据进行基本的探索
#返回缺失值个数以及最大最小值

import pandas as pd

datafile= '../data/air_data.csv' #航空原始数据,第一行为属性标签
resultfile = '../tmp/explore.xls' #数据探索结果表

data = pd.read_csv(datafile, encoding = 'utf-8') #读取原始数据，指定UTF-8编码（需要用
    文本编辑器将数据转换为UTF-8编码）
```

```
explore = data.describe(percentiles = [], include = 'all').T #包括对数据的基本描述，
    percentiles参数是指定计算多少的分位数表（如1/4分位数、中位数等）；T是转置，转置后更方便查阅
explore['null'] = len(data)-explore['count'] #describe()函数自动计算非空值数，需要手
    动计算空值数

explore = explore[['null', 'max', 'min']]
explore.columns = [u'空值数', u'最大值', u'最小值'] #表头重命名
'''这里只选取部分探索结果。
describe()函数自动计算的字段有count（非空值数）、unique（唯一值数）、top（频数最高者）、freq
    （最高频数）、mean（平均值）、std（方差）、min（最小值）、50%（中位数）、max（最大值）'''

explore.to_excel(resultfile) #导出结果
```

代码详见：示例程序 /code/data_explore.py

根据上面的代码得到的探索结果见表 7-4。

表7-4　数据探索分析结果表

属性名称	空值记录数	最大值	最小值
SUM_YR_1	551	239 560	0
SUM_YR_2	138	234 188	0
…	…	…	…
SEG_KM_SUM	0	580 717	368
AVG_DISCOUNT	0	1.5	0

7.2.3　数据预处理

本案例主要采用数据清洗、属性规约与数据变换的预处理方法。

1. 数据清洗

通过数据探索分析，发现数据中存在缺失值，票价最小值为 0、折扣率最小值为 0、总飞行公里数大于 0 的记录。由于原始数据量大，这类数据所占比例较小，对于问题影响不大，因此对其进行丢弃处理。具体处理方法如下。

- 丢弃票价为空的记录。
- 丢弃票价为 0、平均折扣率不为 0、总飞行公里数大于 0 的记录。

使用 Pandas 对满足清洗条件的数据进行丢弃，处理方法：满足清洗条件的一行数据全部丢弃，其代码如代码清单 7-2 所示。

代码清单7-2　数据清洗代码

```
#-*- coding: utf-8 -*-
#数据清洗，过滤掉不符合规则的数据

import pandas as pd
```

```
datafile= '../data/air_data.csv' #航空原始数据,第一行为属性标签
cleanedfile = '../tmp/data_cleaned.csv' #数据清洗后保存的文件

data = pd.read_csv(datafile,encoding='utf-8') #读取原始数据，指定UTF-8编码（需要用文本
    编辑器将数据装换为UTF-8编码）

data = data[data['SUM_YR_1'].notnull()*data['SUM_YR_2'].notnull()] #票价非空值才保留

#只保留票价非零的，或者平均折扣率与总飞行公里数同时为0的记录。
index1 = data['SUM_YR_1'] != 0
index2 = data['SUM_YR_2'] != 0
index3 = (data['SEG_KM_SUM'] == 0) & (data['avg_discount'] == 0) #该规则是“与”
data = data[index1 | index2 | index3] #该规则是“或”

data.to_excel(cleanedfile) #导出结果
```

代码详见：示例程序 /code/data_clean.py

2. 属性规约

原始数据中属性太多，根据航空公司客户价值 LRFMC 模型，选择与 LRFMC 指标相关的 6 个属性：FFP_DATE、LOAD_TIME、FLIGHT_COUNT、AVG_DISCOUNT、SEG_KM_SUM、LAST_TO_END。删除与其不相关、弱相关或冗余的属性，例如，会员卡号、性别、工作地城市、工作地所在省份、工作地所在国家和年龄等属性。经过属性选择后的数据集，见表 7-5。

表7-5 属性选择后的数据集

LOAD_TIME	FFP_DATE	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	AVG_DISCOUNT
2014/3/31	2013/3/16	23	14	126 850	1.02
2014/3/31	2012/6/26	6	65	184 730	0.76
2014/3/31	2009/12/8	2	33	60 387	1.27
2014/3/31	2009/12/10	123	6	62 259	1.02
2014/3/31	2011/8/25	14	22	54 730	1.36
2014/3/31	2012/9/26	23	26	50 024	1.29
2014/3/31	2010/12/27	77	5	61 160	0.94
2014/3/31	2009/10/21	67	4	48 928	1.05
2014/3/31	2010/4/15	11	25	43 499	1.33
2014/3/31	2007/1/26	22	36	68 760	0.88
2014/3/31	2006/12/26	4	49	64 070	0.91
2014/3/31	2011/8/15	22	51	79 538	0.74
2014/3/31	2009/8/27	2	62	91 011	0.67
2014/3/31	2013/3/18	9	12	69 857	0.79

（续）

LOAD_TIME	FFP_DATE	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	AVG_DISCOUNT
2014/3/31	2013/3/12	2	65	75 026	0.69
2014/3/31	2007/2/1	13	28	50 884	0.86
2014/3/31	2004/12/18	56	6	73 392	0.66
2014/3/31	2008/8/15	23	15	36 132	1.07
2014/3/31	2011/8/9	48	6	55 242	0.79
2014/3/31	2011/11/23	36	7	44 175	0.89

3. 数据变换

数据变换是将数据转换成“适当的”格式，以适应挖掘任务及算法的需要。本案例中主要采用的数据变换方式为属性构造和数据标准化。

由于原始数据中并没有直接给出 LRFMC 五个指标，需要通过原始数据提取这五个指标，具体的计算方式如下。

（1）L = LOAD_TIME − FFP_DATE

会员入会时间距观测窗口结束的月数=观测窗口的结束时间−入会时间 [单位：月]

（2）R = LAST_TO_END

客户最近一次乘坐公司飞机距观测窗口结束的月数=最后一次乘机时间至观察窗口末端时长 [单位：月]

（3）F = FLIGHT_COUNT

客户在观测窗口内乘坐公司飞机的次数=观测窗口的飞行次数 [单位：次]

（4）M = SEG_KM_SUM

客户在观测时间内在公司累计的飞行里程=观测窗口的总飞行公里数 [单位：公里]

（5）C = AVG_DISCOUNT

客户在观测时间内乘坐舱位所对应的折扣系数的平均值=平均折扣率 [单位：无]

5 个指标的数据提取后，对每个指标数据分布情况进行分析，其数据的取值范围见表 7-6。从表中数据可以发现，5 个指标的取值范围数据差异较大，为了消除数量级数据带来的影响，需要对数据进行标准化处理。

表7-6　LRFMC指标取值范围

属 性 名 称	L	R	F	M	C
最小值	12.23	0.03	2	368	0.14
最大值	114.63	24.37	213	580 717	1.5

标准差标准化处理的 Python 代码如代码清单 7-3 所示，datafile 为输入数据文件，zscoredata 为标准差标准化后数据集。

代码清单7-3 标准差标准化

```
#-*- coding: utf-8 -*-
#标准差标准化

import pandas as pd

datafile = '../data/zscoredata.xls' #需要进行标准化的数据文件;
zscoredfile = '../tmp/zscoreddata.xls' #标准差化后的数据存储路径文件;

#标准化处理
data = pd.read_excel(datafile)
data = (data - data.mean(axis = 0))/(data.std(axis = 0)) #简洁的语句实现了标准化变
    换，类似地可以实现任何想要的变换。
data.columns=['Z'+i for i in data.columns] #表头重命名。

data.to_excel(zscoredfile, index = False) #数据写入
```

代码详见：示例程序 /code/zscore_data.py

标准差标准化处理后，形成 ZL、ZR、ZF、ZM、ZC 5 个属性的数据，如表 7-7 所示。

表7-7 标准化处理后的数据集

ZL	ZR	ZF	ZM	ZC
1.690	0.140	−0.636	0.069	−0.337
1.690	−0.322	0.852	0.844	−0.554
1.682	−0.488	−0.211	0.159	−1.095
1.534	−0.785	0.002	0.273	−1.149
0.890	−0.427	−0.636	−0.685	1.232
−0.233	−0.691	-0.636	−0.604	−0.391
−0.497	1.996	−0.707	−0.662	−1.311
−0.869	−0.268	−0.281	−0.262	3.396
−1.075	0.025	−0.423	−0.521	0.150
1.907	−0.884	2.979	2.130	0.366
0.478	−0.565	0.852	−0.068	−0.662
0.469	−0.939	0.073	0.104	−0.013
0.469	−0.185	−0.140	−0.220	−0.932
0.453	1.517	0.073	−0.301	3.288
0.369	0.747	−0.636	−0.626	−0.283
0.312	−0.896	0.498	0.954	−0.500
−0.026	−0.681	0.073	0.325	0.366
−0.051	2.723	−0.636	−0.749	0.799
−0.092	2.879	−0.707	−0.734	−0.662
−0.150	−0.521	1.278	1.392	1.124

数据详见：示例程序 /data/zscoreddata.xls

7.2.4　模型构建

客户价值分析模型构建主要由两个部分构成，第一个部分根据航空公司客户 5 个指标的数据，对客户进行聚类分群。第二部分结合业务对每个客户群进行特征分析，分析其客户价值，并对每个客户群进行排名。

1. 客户聚类

采用 K-Means 聚类算法对客户数据进行客户分群，聚成 5 类（需要结合业务的理解与分析来确定客户的类别数量）。

K-Means 聚类算法位于 Scikit-Learn 库下的聚类子库（sklearn.cluster），代码如代码清单 7-4 所示，输入数据集为 inputfile，聚类类别数为 k = 5。

代码清单7-4　K-Means聚类算法

```
#-*- coding: utf-8 -*-
#K-Means聚类算法

import pandas as pd
from sklearn.cluster import KMeans #导入K均值聚类算法

inputfile = '../tmp/zscoreddata.xls' #待聚类的数据文件
k = 5                          #需要进行的聚类类别数

#读取数据并进行聚类分析
data = pd.read_excel(inputfile) #读取数据

#调用k-means算法，进行聚类分析
kmodel = KMeans(n_clusters = k, n_jobs = 4) #n_jobs是并行数，一般等于CPU数较好
kmodel.fit(data) #训练模型

kmodel.cluster_centers_ #查看聚类中心
kmodel.labels_ #查看各样本对应的类别
```

代码详见：示例程序 /code/KMeans_cluster.py

对数据进行聚类分群的结果如表 7-8 所示。

表7-8　客户聚类结果

聚类类别	聚类个数	聚类中心				
		ZL	ZR	ZF	ZM	ZC
客户群 1	5 337	0.483	−0.799	2.483	2.424	0.308
客户群 2	15 735	1.160	−0.377	−0.087	−0.095	−0.158
客户群 3	12 130	−0.314	1.686	−0.574	−0.537	−0.171
客户群 4	24 644	−0.701	−0.415	−0.161	−0.165	−0.255
客户群 5	4 198	0.057	−0.006	−0.227	−0.230	2.191

注：由于 K-Means 聚类是随机选择类标号，因此重复此实验得到结果中的类标号可能与此不同；另外，由于算法的精度问题，重复实验得到的聚类中心也可能略有不同。

2. 客户价值分析

针对聚类结果进行特征分析，如图 7-3 所示。其中，客户群 1 在 F、M 属性上最大，在 R 属性上最小；客户群 2 在 L 属性上最大；客户群 3 在 R 属性上最大，在 F、M 属性上最小；客户群 4 在 L、C 属性上最小；客户群 5 在 C 属性上最大。结合业务分析，通过比较各个指标在群间的大小对某一个群的特征进行评价分析。例如客户群 1 在 F、M 属性最大，在 R 指标最小，因此可以说 F、M、R 在客户群 1 是优势特征。以此类推，F、M、R 在客户群 3 上是劣势特征。从而总结出每个群的优势和弱势特征，具体结果如表 7-9 所示。

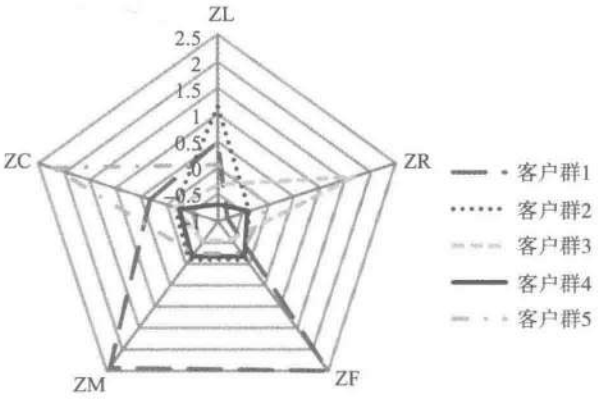


图 7-3 客户群特征分析图

表7-9 客户群特征描述表

群 类 别	优势特征			弱势特征		
客户群 1	F	M	*R*			
客户群 2	L	*F*	*M*			
客户群 3				*F*	*M*	R
客户群 4				*L*		*C*
客户群 5	C			*R*	F	M

注：正常字体表示最大值、加粗字体表示次大值、斜体字体表示最小值、带下划线的字体表示次小值。

由上述的特征分析的图表说明每个客户群的都有显著不同的表现特征，基于该特征描述，本案例定义五个等级的客户类别：重要保持客户、重要发展客户、重要挽留客户、一般客户、低价值客户。他们之间的区别如图 7-4 所示，其中每种客户类别的特征如下：

- **重要保持客户**：这类客户的平均折扣率（C）较高（一般所乘航班的舱位等级较高），最近乘坐过本公司航班（R）低，乘坐的次数（F）或里程（M）较高。他们是航空公司的高价值客户，是最为理想的客户类型，对航空公司的贡献最大，所占比例却较小。航空公司应该优先将资源投放到他们身上，对他们进行差异化管理和一对一营销，提高这类客户的忠诚度与满意度，尽可能延长这类客户的高水平消费。
- **重要发展客户**：这类客户的平均折扣率（C）较高，最近乘坐过本公司航班（R）低，但乘坐次数（F）或乘坐里程（M）较低。这类客户入会时长（L）短，他们是航空公司的潜在价值客户。虽然这类客户的当前价值并不是很高，但却有很大的发展潜力。航空公司要努力促使这类客户增加在本公司的乘机消费和合作伙伴处的消费，也就是增加客户的钱包份额。通过客户价值的提升，加强这类客户的满意度，提高他们

转向竞争对手的转移成本，使他们逐渐成为公司的忠诚客户。

- **重要挽留客户**：这类客户过去所乘航班的平均折扣率（C）、乘坐次数（F）或者里程（M）较高，但是较长时间已经没有乘坐本公司的航班（R）高或是乘坐频率变小。他们客户价值变化的不确定性很高。由于这些客户衰退的原因各不相同，所以掌握客户的最新信息、维持与客户的互动就显得尤为重要。航空公司应该根据这些客户的最近消费时间、消费次数的变化情况，推测客户消费的异动状况，并列出客户名单，对其重点联系，采取一定的营销手段，延长客户的生命周期。
- **一般与低价值客户**：这类客户所乘航班的平均折扣率（C）很低，较长时间没有乘坐过本公司航班（R）高，乘坐的次数（F）或里程（M）较低，入会时长（L）短。他们是航空公司的一般用户与低价值客户，可能是在航空公司机票打折促销时，才会乘坐本公司航班。

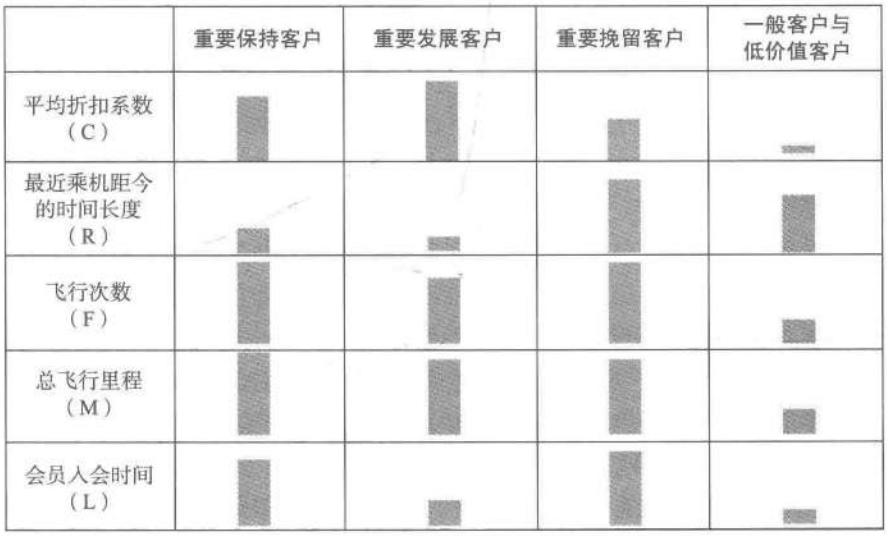


图 7-4　客户类别的特征分析

其中，重要发展客户、重要保持客户、重要挽留客户这三类重要客户分别可以归入客户生命周期管理的发展期、稳定期、衰退期三个阶段。

根据每种客户类型的特征，对各类客户群进行客户价值排名，其结果如表 7-10 所示。针对不同类型的客户群提供不同的产品和服务，提升重要发展客户的价值、稳定和延长重要保持客户的高水平消费、防范重要挽留客户的流失并积极进行关系恢复。

表7-10　客户群价值排名

客　户　群	排　　名	排 名 含 义
客户群 1	1	重要保持客户
客户群 5	2	重要发展客户

（续）

客 户 群	排 名	排名含义
客户群 2	3	重要挽留用户
客户群 4	4	一般客户
客户群 3	5	低价值客户

本模型采用历史数据进行建模，随着时间的变化，分析数据的观测窗口也在变换。因此，对于新增客户详细信息，考虑业务的实际情况，该模型建议每一个月运行一次，对其新增客户信息通过聚类中心进行判断，同时对本次新增客户的特征进行分析。如果增量数据的实际情况与判断结果差异大，需要业务部门重点关注，查看变化大的原因以及确认模型的稳定性。如果模型稳定性变化大，需要重新训练模型进行调整。目前模型进行重新训练的时间没有统一标准，大部分情况都是根据经验来决定。根据经验建议：每隔半年训练一次模型比较合适。

3. 模型应用

根据对各个客户群进行特征分析，采取下面的一些营销手段和策略，为航空公司的价值客户群管理提供参考。

（1）会员的升级与保级

航空公司的会员可以分为白金卡会员、金卡会员、银卡会员、普通卡会员，其中非普通卡会员可以统称为航空公司的精英会员。虽然各个航空公司都有自己的特点和规定，但会员制的管理方法是大同小异的。成为精英会员一般都是要求在一定时间内（如一年）积累一定的飞行里程或航段，达到这种要求后就会在有效期内（通常为两年）成为精英会员，并享受相应的高级别服务。有效期快结束时，根据相关评价方法确定客户是否有资格继续作为精英会员，然后对该客户进行相应地升级或降级。

然而，由于许多客户并没有意识到或根本不了解会员升级或保级的时间与要求（相关的文件说明往往复杂且不易理解），经常在评价期过后才发现自己其实只差一点就可以实现升级或保级，却错过了机会，使之前的里程积累白白损失。同时，这种认知还可能导致客户的不满，干脆放弃在本公司的消费。

因此，航空公司可以在对会员升级或保级进行评价的时间点之前，对那些接近但尚未达到要求的较高消费客户进行适当提醒甚至采取一些促销活动，刺激他们通过消费达到相应标准。这样既可以获得收益，同时也提高了客户的满意度，增加了公司的精英会员。

（2）首次兑换

航空公司常旅客计划中最能够吸引客户的内容就是客户可以通过消费积累的里程来兑换免票或免费升舱等。各个航空公司都有一个首次兑换标准，也就是当客户的里程或航段积累到一定程度时才可以实现第一次兑换，这个标准会高于正常的里程兑换标准。但是很多公司的里程积累随着时间会进行一定地削减，例如有的公司会在年末对该年积累的里程进行折半

处理。这样会导致许多不了解情况的会员白白损失自己好不容易积累的里程，甚至总是难以实现首次兑换。同样，这也会引起客户的不满或流失。可以采取的措施是从数据库中提取出接近但尚未达到首次兑换标准的会员，对他们进行提醒或促销，使他们通过消费达到标准。一旦实现了首次兑换，客户在本公司进行再次消费兑换就比在其他公司进行兑换要容易许多，在一定程度上等于提高了转移的成本。另外，在一些特殊的时间点（如里程折半的时间点）之前可以给客户一些提醒，这样可以增加客户的满意度。

（3）交叉销售

通过发行联名卡等与非航空类企业的合作，使客户在其他企业的消费过程中获得本公司的积分，增强与公司的联系，提高他们的忠诚度。例如，可以查看重要客户在非航空类合作伙伴处的里程积累情况，找出他们习惯的里程积累方式（是否经常在合作伙伴处消费、更喜欢消费哪些类型合作伙伴的产品），对他们进行相应促销。

客户识别期和发展期为客户关系打下基石，但是这两个时期带来的客户关系是短暂的、不稳定的。企业要获取长期的利润，必须具有稳定的、高质量的客户。保持客户对于企业是至关重要的，不仅因为争取一个新客户的成本远远高于维持老客户的成本，更重要的是客户流失会造成公司收益的直接损失。因此，在这一时期，航空公司应该努力维系客户关系，使之处于较高的水准，最大化生命周期内公司与客户的互动价值，并使这样的高水平尽可能延长。对于这一阶段的客户，主要应该通过提供优质的服务产品和提高服务水平来提高客户的满意度。通过对旅客数据库的数据挖掘、进行客户细分，可以获得重要保持客户的名单。这类客户一般所乘航班的平均折扣率（C）较高，最近乘坐过本公司航班（R 低）、乘坐的频率（F）或里程（M）也较高。他们是航空公司的价值客户，是最理想的客户类型，对航空公司的贡献最大，所占比例却比较小。航空公司应该优先将资源投放到他们身上，对他们进行差异化管理和一对一营销，提高这类客户的忠诚度与满意度，尽可能延长这类客户的高水平消费。

7.3　上机实验

1. 实验目的

- 了解 K-Means 聚类算法在客户价值分析实例中的应用。
- 利用 Pandas 快速实现数据 z-score（标准差）标准化以及用 Scikit-Learn 的聚类库实现 K-Means 聚类。

2. 实验内容

依据航空公司客户价值分析的 LRFMC 模型提取客户信息的 LRFMC 指标。对其进行标准差标准化并保存后，采用 K-Means 算法完成客户的聚类，分析每类的客户特征，从而获得每类的客户价值。

- 利用 Pandas 程序，读入 LRFMC 指标文件，分别计算各个指标的均值和标准差，使用标准差标准化公式完成 LRFMC 指标的标准化，并将标准化后的数据进行保存。
- 编写 Python 程序，完成客户的 K-Means 聚类，获得聚类中心与类标号。输出聚类中心的特征图，并统计每个类别的客户数。

3. 实验方法与步骤

实验一

对 L、R、F、M、C 五个指标进行 z-score（标准差）标准化。

1）启动 Python 并导入 Pandas，使用 read_excel() 函数将待标准差标准化的数据“上机实验 /data/zscoredata.xls”读入到 Python 中。

2）使用 mean() 与 std() 函数，获得 L、R、F、M、C 五个指标的平均值与标准差。

3）根据 z-score（标准差）标准化公式 $z_{ij} = (x_{ij} - x_i)/s_i$，其中 z_{ij} 是标准化后的变量值；x_{ij} 是实际变量值，x_i 为变量的算术平均值，s_i 是变量的标准差，进行标准差标准化。

实验二

1）使用 read_excel 函数将航空数据预处理后的数据读入 Python 工作空间，截取最后 5 列数据作为 K-Means 算法的输入数据。

2）调用 KMeans 函数对 1）中的数据进行聚类，得到聚类标号和聚类中心点。

3）根据聚类标号统计每个类别的客户数，同时根据聚类中心点向量画出客户聚类中心向量图并保存。

4. 思考与实验总结

1）Scikit-Learn 中 KMeans 函数中的初始聚类中心可以使用什么算法得到？默认是什么算法？

2）使用不同的预处理对原始数据进行变换，再使用 K-Means 算法进行聚类，对比聚类结果，分析不同数据预处理对 K-Means 算法的影响。

7.4 拓展思考

本章主要针对客户价值进行分析，对客户流失并没有提出具体的分析。由于在航空客户关系管理中客户流失的问题未被重视，故对航空公司造成了巨大的损害。客户流失对利润增长造成的负面影响非常大，仅次于公司规模、市场占有率和单位成本等因素的影响。客户与航空公司之间的关系越长久，给航空公司带来的利润就会越高。所以流失一个客户，比获得一个新客户对公司的损失更大。因为要获得新客户，需要在销售、市场、广告和人员工资上花费很多，并且大多数新客户产生的利润不如那些流失的老客户多。

因此，在国内航空市场竞争日益激烈的背景下，航空公司在客户流失方面应该引起足够的重视。如何改善流失问题，继而提高客户满意度、忠诚度是航空公司维护自身市场并面对

激烈竞争的一件大事，客户流失分析将成为帮助航空公司开展持续改进活动的指南。

客户流失分析可以针对目前老客户进行分类预测。针对航空公司客户信息数据（见表 7-2），可以进行老客户以及客户类型的定义（其中将飞行次数大于 6 次的客户定义为老客户，已流失客户定义为：第二年飞行次数与第一年飞行次数比例小于 50% 的客户；准流失客户定义为：第二年飞行次数与第一年飞行次数比例在［50%，90%）内的客户；未流失客户定义为：第二年飞行次数与第一年飞行次数比例大于 90% 的客户）。同时，需要选取客户信息中的关键属性，如会员卡级别、客户类型（流失、准流失、未流失）、平均乘机时间间隔、平均折扣率、积分兑换次数、非乘机积分总和、单位里程票价和单位里程积分等。随机选取数据的 80% 作为分类的训练样本，剩余的 20% 作为测试样本。构建客户的流失模型，运用模型预测未来客户的类别归属（未流失、准流失或已流失）。

7.5　小结

本章结合航空公司客户价值分析的案例，重点介绍了数据挖掘算法中 K-Means 聚类算法在实际案例中的应用。针对客户价值识别传统的 RFM 模型的不足，采用 K-Means 算法进行分析，并详细地描述了数据挖掘的整个过程，对其相应的算法给出了 Python 上机实验步骤。

Chapter 8 第 8 章

中医证型关联规则挖掘

8.1 背景与挖掘目标

恶性肿瘤俗称癌症，当前已成为危害我国居民生命健康的主要杀手。应用中医药治疗恶性肿瘤已成为公认的综合治疗方法之一，且中医药治疗乳腺癌有着广泛的适应证和独特的优势。从整体出发，调整机体气血、阴阳、脏腑功能的平衡，根据不同的临床证候进行辨证论治。确定“先证而治”的方向：即后续证侯尚未出现之前，需要截断恶化病情的哪些后续证侯。发现中医症状间的关联关系和诸多症状间的规律性，并且依据规则分析病因、预测病情发展以及为未来临床诊治提供有效借鉴。这样，在治疗患者的过程中，医生可以有效地减少西医治疗的毒副作用，为后续治疗打下基础。并且还能够帮助乳腺癌患者在手术后恢复体质，改善生存质量，有利于提高患者的生存机率。

三阴乳腺癌患者的临床患病信息见表 8-1，由信息整理而成的原始数据见表 8-2，请根据这些数据实现以下目标。

1）借助三阴乳腺癌患者的病理信息，挖掘患者的症状与中医证型之间的关联关系。

2）对截断治疗提供依据，挖掘潜性证素。

表8-1 原始属性表

序号	属性名称	属性描述
1	实际年龄	A1：≤ 30 岁；A2：31–40 岁；A3：41–50 岁；A4：51–60 岁；A5：61–70 岁；A6：≥ 71 岁
2	发病年龄	a1：≤ 30 岁；a2：31–40 岁；a3：41–50 岁；a4：51–60 岁；a5：61–70 岁；a6：≥ 71 岁
3	初潮年龄	C1：≤ 12 岁；C2：13–15 岁；C3：≥ 16 岁

（续）

序号	属性名称	属性描述
4	既往月经是否规律	D1：月经规律；D2：月经先期；D3：月经后期；D4：月经先后不定期
5	是否痛经	Y：是；N：否
6	是否绝经	Y：是；N：否
	……	
64	肝气郁结证得分	总分 40 分
65	热毒蕴结证得分	总分 44 分
66	冲任失调证得分	总分 41 分
67	气血两虚证得分	总分 43 分
68	脾胃虚弱证得分	总分 43 分
69	肝肾阴虚证得分	总分 38 分
70	TNM 分期	H1：Ⅰ；H2：Ⅱ；H3：Ⅲ；H4：Ⅳ
71	确诊后几年发现转移	1. 无转移：BU0；2. 小于等于三年：BU1；3. 大于三年小于等于五年：BU2；4. 大于五年：BU3
72	转移部位	R1：骨；R2：肺；R3：脑；R4：肝；R5：其他；R0：无转移
73	病程阶段	S1：围手术期；S2：围化疗期；S3：围放疗期；S4：巩固期

8.2　分析方法与过程

由于患者在围手术期、围化疗期、围放疗期和内分泌治疗期等各个病程阶段，基本都会出现特定的临床症状，故而可以运用中医截断疗法进行治疗，在辨病的基础上围绕各个病程的特殊证候先证而治。截断扭转的主要观点是强调早期治疗，力图快速控制病情，截断病情邪变深入，扭转阻止疾病恶化[17]。

目前，患者的临床病理信息大部分都记录在纸张上，包含了患者的基本信息、具体患病信息等，很少会将患者的患病信息存放于系统中，因此进行数据分析时会面临数据缺乏的情况。针对这种状况，本章采用问卷调查的方式收集数据；运用数据挖掘技术对收集的数据进行数据探索与预处理，形成建模数据；采用关联规则算法，挖掘各中医证素与乳腺癌 TNM 分期之间的关系，其中乳腺癌 TNM 分期是乳腺癌分期基本原则，Ⅰ期较轻，Ⅳ期较严重。探索不同分期阶段的三阴乳腺癌患者的中医证素分布规律，以及采用截断病变发展、先期干预的治疗思路，指导三阴乳腺癌的中医临床治疗。

本次数据挖掘建模的总体流程如图 8-1 所示。

中医证型关联规则挖掘主要包括以下步骤。

1）以问卷调查的方式对数据进行收集，并将问卷信息整理成原始数据。

2）对原始数据集进行数据预处理，包括数据清洗、属性规约、数据变换。

表8-2 原始数据表

患者编号	实际年龄	发病年龄	初潮年龄	既往月经是否规律	是否痛经	是否绝经	是否有更年期	婚否	育几胎	产几胎	流几胎	生育年龄	是否哺乳	哺乳时间	乳汁量	肿块部位	肿块是否疼痛
20140002	A2	a2	B2	C1	N	Y	Y	Y	3	2	1	D3	Y	E3	F1	G1	N
20140003	A5	a5	B2	C1	Y	Y	Y	Y	2	1	1	D3	Y	E3	F1	G1	N
20140007	A4	a4	B2	C1	Y	Y	Y	Y	1	1	0	D2	Y	E3	F1	G1	N
20140010	A5	a5	B2	C1	Y	Y	Y	Y	2	1	1	D2	Y	E3	F1	G1	N
20140020	A1	a1	B2	C1	N	N	N	Y	2	1	1	D1	Y	E2	F1	G3	Y
20140027	A2	a2	B3	C1	N	N	N	Y	2	1	1	D1	Y	E1	F1	G2	N
20140028	A3	a3	B3	C2	Y	Y	Y	Y	5	2	3	D1	Y	E3	F1	G2	Y
20140004	A3	a3	B2	C1	N	Y	N	Y	1	1	0	D3	N	NULL	F2	G1	N
20140009	A3	a3	B2	C1	Y	Y	Y	Y	1	1	0	D2	N	NULL	F2	G3	N
20140012	A2	a2	B1	C4	N	Y	Y	Y	1	1	0	D2	Y	E1	F2	G4	N
20140016	A5	a4	B2	C3	Y	Y	Y	Y	3	2	1	D2	Y	E3	F2	G5	N
20140017	A3	a3	B2	C1	Y	Y	N	Y	1	1	0	D2	N	NULL	F2	G1	Y
20140019	A1	a1	B2	C4	Y	N	N	Y	2	1	1	D1	Y	E1	F2	G3	N
20140023	A2	a2	B2	C1	Y	N	N	Y	3	2	1	D1	Y	E3	F2	G1	Y
20140025	A2	a2	B3	C1	N	Y	Y	Y	3	1	2	D1	Y	E2	F2	G5	N
20140026	A3	a3	B2	C1	N	Y	Y	Y	2	1	1	D1	n	NULL	F2	G3	N
20140005	A4	a4	B2	C1	N	Y	Y	Y	1	1	0	D3	Y	E3	F3	G1	N
20140006	A4	a4	B3	C1	N	Y	N	Y	2	1	1	D2	Y	E3	F3	G5	N
20140008	A5	a5	B2	C1	N	Y	Y	Y	3	1	2	D2	Y	E2	F3	G3	Y
20140011	A5	a4	B2	C1	N	Y	Y	Y	2	2	0	D2	Y	E3	F3	G1	N

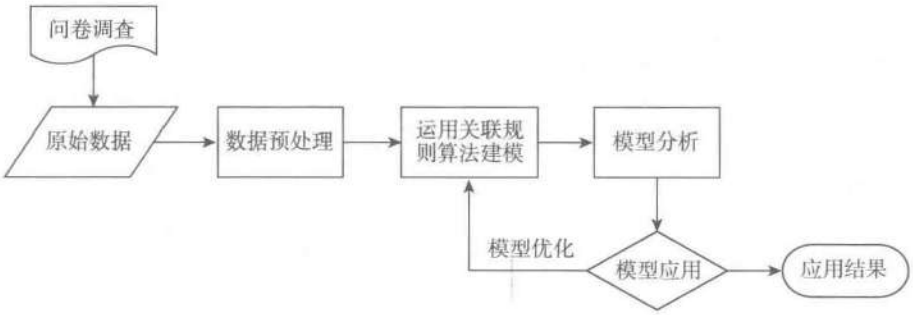


图 8-1　中医证型关联规则挖掘模型总体流程图

3）利用步骤 2）形成的建模数据，采用关联规则算法，调整模型输入参数，获取各中医证素与乳腺癌 TNM 分期之间的关系。

4）结合实际业务，对模型结果进行分析，且将模型结果应用到实际业务中，最后输出关联规则结果。

8.2.1　数据获取

本案例采用调查问卷的形式对数据进行搜集，数据获取的具体过程如下。

1）拟定调查问卷表并形成原始指标表。

2）定义纳入标准与排除标准。

3）将收集回来的问卷表整理成原始数据。

首先根据中华中医药学会制定的相关指南与标准，从乳腺癌 6 种分型的症状（见表 8-3）中提取相应证素拟定调查问卷表，见表 8-4，并制定三阴乳腺癌中医证素诊断量表（见表 8-5），从调查问卷中提炼信息形成原始属性表。然后依据标准定义表（见表 8-6），将有效的问卷表整理成原始数据（见表 8-2）。问卷调查需要满足两个条件：①问卷信息采集者均要求有中医诊断学基础，能准确识别病人的舌苔脉象，用通俗的语言解释医学术语，并确保患者信息填写准确。②问卷调查对象必须是三阴乳腺癌患者。本章的调查对象是某省中医院以及肿瘤医院等处于各病程阶段的 1 253 位三阴乳腺癌患者。

表8-3　乳腺癌辨证分型

证　型	主要症状
肝气郁结证	乳房肿块，时觉胀痛，情绪忧郁或急躁，心烦易怒，苔薄白或薄黄，脉弦滑
热毒蕴结证	乳房肿块，增大迅速，疼痛，间或红肿，甚则溃烂、恶臭，或发热，心烦口干，便秘，小便短赤，舌暗红，有瘀斑，苔黄腻，脉弦数
冲任失调证	乳房肿块，月经前胀痛明显，或月经不调，腰腿酸软，烦劳体倦，五心烦热，口干咽燥，舌淡，苔少，脉细无力
气血两虚证	乳房肿块，与胸壁粘连，推之不动，头晕目眩，气短乏力，面色苍白，消瘦纳呆，舌淡，脉沉细无力

（续）

证　型	主要症状
脾胃虚弱证	纳呆或腹胀，便溏或便秘，舌淡，苔白腻，脉细弱
肝肾阴虚证	头晕目眩，腰膝酸软，目涩梦多，咽干口燥，大便干结，月经紊乱或停经，舌红，苔少脉细数

表8-4　三阴乳腺癌中医证素调查问卷

我们很希望了解一些有关您及您的健康状况的信息。请独立回答以下所有问题，并圈出对您最合适的答案。答案无“正确”与“错误”之分。您提供的信息，我们将绝对保密。

[基本信息]

编号			填表日期		年　月　日	
姓名		性别		年龄	确诊为乳腺癌的年龄	
婚姻状况		□已婚　□未婚　□离异　□丧偶				
文化程度		□小学　□初中　□高中　□中专　□大学及以上　□其他				
职业		□工人　□农民　□知识分子　□干部　□个体经商户　□无职业				
工作单位 / 家庭住址						
联系方式				病人种类	□门诊　□住院	
月经史	初潮　岁；月经（□规律　□不规律）；持续　天；间隔　天；					
	痛经	□有　□无		末次月经时间		
	闭经	□是　□否【若是，则：闭经于 岁；闭经症状（□有　□无）】				
婚育史	婚否	□未婚　□已婚【若已婚，则：结婚年龄为　岁】				
	生育状况	□未生育　□已生育【若已生育，则育　胎，生产　胎，流产　胎；首胎生于　岁，末胎生于　岁】				
哺乳史	是否哺乳	□是　□否【若是，则哺乳　个孩子；最长哺乳　年　月，最短哺乳　年　月】				
	乳汁量	□少　□一般　□多		哺乳部位	□双侧　□左侧　□右侧	
乳腺肿块	部位	□外上　□内上　□外下　□内下　□乳头后				
	发生时间及经过					
乳腺疼痛	有无疼痛	□有　□无	性质	□刺痛　□胀痛　□隐痛　□灼痛		
	与月经来潮关系		□有　□无			
乳头溢液		□有　□无	性质	□水样　□乳汁样　□血样　□脓性　□浆液性		
皮肤水肿		□有　□无	腋下肿块	□有　□无		
乳头乳晕糜烂		□有　□无	其他症状			
曾经治疗	新辅助治疗	化疗：方案（剂量）　　已进行　周期 内分泌治疗：方案（剂量）　　使用时间				
	术前放疗	部位：□乳房　□内乳区　□锁骨区　　剂量：次数：				
	辅助治疗	化疗：方案（剂量）　　已进行　周期 内分泌治疗：方案（剂量）　　使用时间				
	中医药治疗	治疗时间： 效果：				

（续）

[术后病理及免疫组化资料]

原发肿瘤直径	区域淋巴结状态	TNM 分期	组织学类型	组织学分级

P-Gp	GST π	TOPO Ⅱ	Ki-67	VEGF 表达	P53 表达

[病程阶段分期]

围手术期	围化疗期	围放疗期	巩　固　期

表8-5　三阴乳腺癌中医证素诊断量表

Ⅰ. 肝气郁结证					
定　义	肝失疏泄，气机郁滞，所表现的情志抑郁，胁胀，胁痛等证候				
必备证素	肝，气滞		或兼证素	心神 [脑]，(胆)，胞宫	
常见证候及计量值					
3 分		2 分		1 分	
抑郁或忧虑 // 喜叹气	□	情志有关	□	排便不爽	□
胁胀	□	烦躁 // 急躁易怒	□	嗳气	□
乳房胀	□	胸闷	□	咽部异物感	□
乳房痛	□	腹胀 // 脘痞胀	□	口苦	□
胁痛 // 右上腹痛	□	胀痛或窜痛	□		
		大便时溏时结	□		
		痛经	□		
		月经错乱	□		
		乳房结块	□		
		肝大 // 胆囊肿大 // 脾大	□		
		脉弦	□		
小计（A）	×3 分 = 分	小计（B）	×2 分 = 分	小计（C）	×1 分 =　分
总分 41 分		总得分（A+B+C）		分	

表8-6　标准定义表

标　准	详细信息
纳入标准	❑ 病理诊断为乳腺癌 ❑ 病历完整，能提供既往接受检查、治疗等相关信息，包括发病年龄、月经状态、原发肿瘤大小、区域淋巴结状态、组织学类型、组织学分级、P53 表达、VEGF 表达等，作为临床病理及肿瘤生物学的特征指标。 ❑ 没有精神类疾病，能自主回答问卷调查者

（续）

标　准	详 细 信 息
排除标准	❑ 本研究中临床、病理、肿瘤生物学指标不齐全者 ❑ 存在第二肿瘤（非乳腺癌转移） ❑ 精神病患者或不能自主回答问卷调查者 ❑ 不愿意参加本次调查者或中途退出本次调查者 ❑ 填写的资料无法根据诊疗标准进行分析者

8.2.2 数据预处理

本案例中数据预处理过程包括数据清洗、属性规约和数据变换。数据来源于问卷调查，因此在数据预处理开始阶段，需要把纸质的问卷整理成原始数据集。针对原始数据集，经过数据预处理，形成建模数据集。

1. 数据清洗

在收回的问卷中，存在无效的问卷，为了便于模型分析，需要对其进行处理。在经过问卷有效性条件（见表 8-6）筛选后，数据量变化情况如图 8-2 所示。将有效问卷整理成原始数据，共 930 条记录。

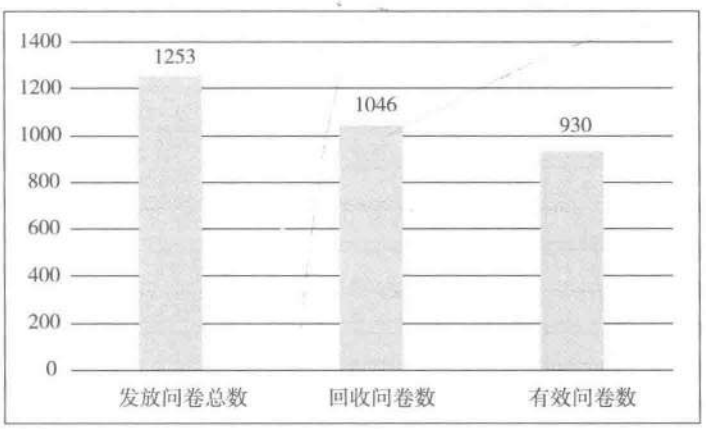


图 8-2　样本数据采集情况

2. 属性规约

本案例收集到的数据共有 73 个属性，为了更有效地对其进行挖掘，将其中冗余属性与挖掘任务不相关属性剔除。因此选取其中 6 种证型得分、TNM 分期的属性值构成数据集，见表 8-7。

表8-7　属性选择后的数据集

患者编号	肝气郁结证得分	热毒蕴结证得分	冲任失调证得分	气血两虚证得分	脾胃虚弱证得分	肝肾阴虚证得分	TNM 分期
20140001	7	30	7	23	18	17	H4

（续）

患者编号	肝气郁结证得分	热毒蕴结证得分	冲任失调证得分	气血两虚证得分	脾胃虚弱证得分	肝肾阴虚证得分	TNM 分期
20140179	12	34	12	16	19	5	H4
……	……	……	……	……	……	……	……
20140930	4	4	12	12	7	15	H4

3. 数据变换

本章数据变换主要采用属性构造和数据离散化两种方法对数据进行处理。首先通过属性构造，获得证型系数，然后通过聚类算法对数据进行离散化处理，形成建模数据。

（1）属性构造

为了更好地反映出中医证素分布的特征，采用证型系数代替具体单证型的证素得分，证型相关系数计算公式为：证型系数＝该证型得分 / 该证型总分。

针对各种证型得分进行属性构造后的数据集见表 8-8。

表8-8　属性构造后的数据集

肝气郁结证型系数	热毒蕴结证型系数	冲任失调证型系数	气血两虚证型系数	脾胃虚弱证型系数	肝肾阴虚证型系数
0.175	0.682	0.171	0.535	0.419	0.447
0.3	0.773	0.293	0.372	0.442	0.132
……	……	……	……	……	……
0.1	0.091	0.293	0.279	0.163	0.395

数据详见：demo/data/data.xls

（2）数据离散化

由于 Apriori 关联规则算法无法处理连续型数值变量，为了将原始数据格式转换为适合建模的格式，需要对数据进行离散化。本章采用聚类算法对各个证型系数进行离散化处理，将每个属性聚成 4 类，其离散化后的数据格式见表 8-9 ～表 8-14。

表8-9　肝气郁结证型系数离散表

范围标识	肝气郁结证型系数范围	范围内元素的个数
A1	(0, 0.179]	244
A2	(0.179, 0.258]	355
A3	(0.258, 0.35]	278
A4	(0.35, 0.504]	53

表8-10　热毒蕴结证型系数离散表

范围标识	热毒蕴结证型系数范围	范围内元素的个数
B1	[0, 0.15]	325

（续）

范围标识	热毒蕴结证型系数范围	范围内元素的个数
B2	(0.15, 0.296]	396
B3	(0.296, 0.485]	180
B4	(0.485, 0.78]	29

表8-11 冲任失调证型系数离散表

范围标识	冲任失调证型系数范围	范围内元素的个数
C1	(0, 0.201]	296
C2	(0.201, 0.288]	393
C3	(0.288, 0.415]	206
C4	(0.415, 0.61]	35

表8-12 气血两虚证型系数离散表

范围标识	气血两虚证型系数范围	范围内元素的个数
D1	(0, 0.172]	283
D2	(0.172, 0.251]	375
D3	(0.251, 0.357]	228
D4	(0.357, 0.552]	44

表8-13 脾胃虚弱证型系数离散表

范围标识	脾胃虚弱证型系数范围	范围内元素的个数
E1	(0, 0.154]	285
E2	(0.154, 0.256]	307
E3	(0.256, 0.375]	244
E4	(0.375, 0.526]	94

表8-14 肝肾阴虚证型系数离散表

范围标识	肝肾阴虚证型系数范围	范围内元素的个数
F1	(0, 0.178]	200
F2	(0.178, 0.261]	237
F3	(0.261, 0.353]	265
F4	(0.353, 0.607]	228

数据离散化的代码如代码清单 8-1 所示。

代码清单8-1　数据聚类离散化代码

```
#-*- coding: utf-8 -*-
'''
聚类离散化，最后的result的格式为：
      1           2           3           4
A     0    0.178698    0.257724    0.351843
An  240  356.000000  281.000000   53.000000
即(0, 0.178698]有240个，(0.178698, 0.257724]有356个，依此类推。
'''
from __future__ import print_function
import pandas as pd
from sklearn.cluster import KMeans #导入K均值聚类算法

datafile = '../data/data.xls' #待聚类的数据文件
processedfile = '../tmp/data_processed.xls' #数据处理后文件
typelabel ={u'肝气郁结证型系数':'A', u'热毒蕴结证型系数':'B', u'冲任失调证型系数':'C', u'
    气血两虚证型系数':'D', u'脾胃虚弱证型系数':'E', u'肝肾阴虚证型系数':'F'}
k = 4 #需要进行的聚类类别数

#读取数据并进行聚类分析
data = pd.read_excel(datafile) #读取数据
keys = list(typelabel.keys())
result = pd.DataFrame()

if __name__ == '__main__': #判断是否主窗口运行，这句代码的作用比较神奇，有兴趣了解的读取请
                           #自行搜索相关材料
  for i in range(len(keys)):
    #调用k-means算法，进行聚类离散化
    print(u'正在进行“%s”的聚类...' % keys[i])
    kmodel = KMeans(n_clusters = k, n_jobs = 4) #n_jobs是并行数，一般等于CPU数较好
    kmodel.fit(data[[keys[i]]].as_matrix()) #训练模型

    r1 = pd.DataFrame(kmodel.cluster_centers_, columns = [typelabel[keys[i]]])
        #聚类中心
    r2 = pd.Series(kmodel.labels_).value_counts() #分类统计
    r2 = pd.DataFrame(r2, columns = [typelabel[keys[i]]+'n']) #转为DataFrame，记
        录各个类别的数目
    r = pd.concat([r1, r2], axis = 1).sort(typelabel[keys[i]]) #匹配聚类中心和类别
       数目
    r.index = [1, 2, 3, 4]

    r[typelabel[keys[i]]] = pd.rolling_mean(r[typelabel[keys[i]]], 2) #rolling_
        mean()用来计算相邻2列的均值，以此作为边界点
    r[typelabel[keys[i]]][1] = 0.0 #这两句代码将原来的聚类中心改为边界点
    result = result.append(r.T)

  result = result.sort() #以Index排序，即以A,B,C,D,E,F顺序排
  result.to_excel(processedfile)
```

代码详见：demo/code/discretization.py

原始数据集经过数据预处理后，形成建模数据，见表 8-15。

表8-15 建模数据集

肝气郁结证型系数	热毒蕴结证型系数	冲任失调证型系数	气血两虚证型系数	脾胃虚弱证型系数	肝肾阴虚证型系数	TNM 分期
A1	B4	C1	D4	E4	F4	H4
A3	B4	C3	D4	E4	F1	H4
……	……	……	……	……	……	……
A1	B1	C3	D3	E2	F4	H4

8.2.3 模型构建

本案例的目标是探索乳腺癌患者 TNM 分期与中医证型系数之间的关系，因此采用关联规则算法，挖掘它们之间的关联关系。

关联规则算法主要用于寻找数据集中项之间的关联关系。它揭示了数据项间的未知关系，基于样本的统计规律，进行关联规则挖掘。根据所挖掘的关联关系，可以从一个属性的信息来推断另一个属性的信息。当置信度达到某一阈值时，就可以认为规则成立。

1. 中医证型关联规则模型

本次中医证型关联规则建模的流程如图 8-3 所示。

由图 8-3 可知，模型主要由输入、算法处理、输出部分组成。输入部分包括：①建模样本数据的输入；②建模参数的输入。算法处理部分是 Apriori 关联规则算法。输出部分为关联规则的结果。

模型具体实现步骤为：首先设置建模参数最小支持度、最小置信度，输入建模样本数据。然后采用 Apriori 关联规则算法对建模的样本数据进行分析，以模型参数设置的最小支持度、最小置信度以及分析目标作为条件，如果所有的规则都不满足条件，则需要重新调整模型参数，否则输出关联规则结果。

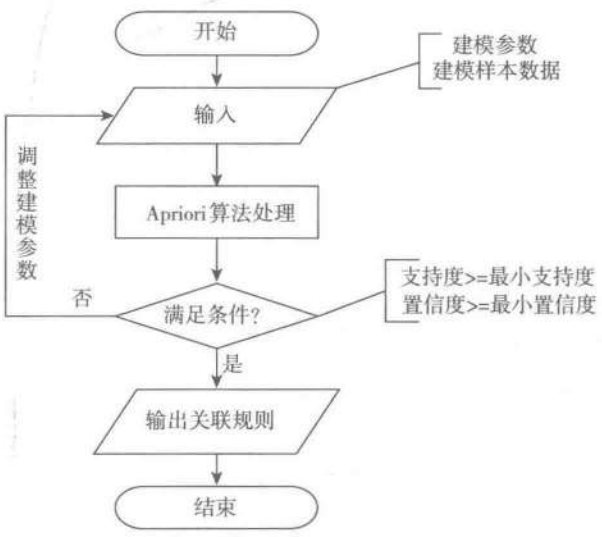


图 8-3 中医证型关联规则模型流程图

目前，如何设置最小支持度与最小置信度，并没有统一的标准。大部分都是根据业务经验设置初始值，然后经过多次调整，获取与业务相符的关联规则结果。本章经过多次调整并结合实际业务分析，选取模型的输入参数为：最小支持度 6%、最小置信度 75%。其关联规则代码如代码清单 8-2 所示。

代码清单8-2　Apriori关联规则代码

```
%% Apriori关联规则算法
#-*- coding: utf-8 -*-
from __future__ import print_function
import pandas as pd
from apriori import * #导入自行编写的高效的Apriori函数
import time #导入时间库用来计算用时

inputfile = '../data/apriori.txt' #输入事务集文件
data = pd.read_csv(inputfile, header=None, dtype = object)

start = time.clock() #计时开始
print(u'\n转换原始数据至0-1矩阵...')
ct = lambda x : pd.Series(1, index = x[pd.notnull(x)]) #转换0-1矩阵的过渡函数，即将标
    签数据转换为1
b = map(ct, data.as_matrix()) #用map方式执行
data = pd.DataFrame(b).fillna(0) #实现矩阵转换，除了1外，其余为空，空值用0填充
end = time.clock() #计时结束
print(u'\n转换完毕，用时：%0.2f秒' %(end-start))
del b #删除中间变量b，节省内存

support = 0.06 #最小支持度
confidence = 0.75 #最小置信度
ms = '---' #连接符，默认'--'，用来区分不同元素，如A--B。需要保证原始表格中不含有该字符

start = time.clock() #计时开始
print(u'\n开始搜索关联规则...')
find_rule(data, support, confidence, ms)
end = time.clock() #计时结束
print(u'\n搜索完成，用时：%0.2f秒' %(end-start))
```

代码详见：demo/code/apriori_rules.py

运行界面输出类似如下内容。

```
转换原始数据至0-1矩阵...

转换完毕，用时：0.15秒

开始搜索关联规则...

正在进行第1次搜索...
数目：276...

正在进行第2次搜索...
数目：947...

正在进行第3次搜索...
数目：41...

结果为：
                    support    confidence
```

```
A3---F4---H4          0.078495    0.879518
C3---F4---H4          0.075269    0.875000
B2---F4---H4          0.062366    0.794521
C2---E3---D2          0.092473    0.754386
D2---F3---H4---A2     0.062366    0.753247

搜索完成，用时：0.97秒
```

2. 模型分析

根据上述运行结果，我们得出了5个关联规则，如A3---F4---H4，它的意思是A3，F4=>H4，类似的，D2---F3---H4---A2的意思是D2，F3，H4=>A2。但是，并非所有关联规则都有意义的，我们只在乎那些以H为规则结果的规则，也就是表8-16所示的规则。

每个关联规则都可以表示成X=>Y，其中X表示各个证型系数范围标识组合而成的规则，Y表示TNM分期为H4期。A3表示肝气郁结证型系数处于（0.258，0.35］范围内的数值，B2表示热毒蕴结证型系数处于（0.15，0.296］范围内的数值，C3表示冲任失调证型系数处于（0.288，0.415］范围内的数值，F4表示肝肾阴虚证型系数处于（0.353，0.607］范围内的数值。

表8-16 中医证型关联规则模型结果

X			X=>Y	
规则编号	范围标识1	范围标识2	支持度（%）	置信度（%）
1	A3	F4	7.85	87.96
2	C3	F4	7.53	87.5
3	B2	F4	6.24	79.45

分析表8-16可以得到如下结论。

1）A3、F4=>H4支持度最大，达到7.85%，置信度最大，达到87.96%，说明肝气郁结证型系数处于（0.258，0.35］，肝肾阴虚证型系数处于（0.353，0.607］范围内，TNM分期诊断为H4期的可能性为87.96%，而这种情况发生的可能性为7.85%。

2）C3、F4=>H4支持度7.53%，置信度87.5%，说明冲任失调证型系数处于（0.201，0.288］，肝肾阴虚证型系数处于（0.353，0.607］范围内，TNM分期诊断为H4期的可能性为87.5%，而这种情况发生的可能性为7.53%。

3）B2、F4=>H4支持度6.24%，置信度79.45%，说明热毒蕴结证型系数处于（0.15，0.296］，肝肾阴虚证型系数处于（0.353，0.607］范围内，TNM分期诊断为H4期的可能性为79.45%，而这种情况发生的可能性为6.24%。

综合以上分析，TNM分期为H4期的三阴乳腺癌患者证型主要为肝肾阴虚证、热毒蕴结证、肝气郁结证和冲任失调，H4期患者肝肾阴虚证和肝气郁结证的临床表现较为突出，其置信度最大达到87.96%。

对于模型结果，从医学角度进行分析：生理上，肝藏血，肾藏精，精血同源，肝肾同源，如《张氏医通》所言："气不耗，归精于肾而为精；精不泄，归精于肝而化清血。"在病理上，

肝肾病变常相互影响，肾阴不足无以养肝阴，肝阳化火则燔灼肾阴。Ⅳ期三阴乳腺癌患者多病程迁延，癌毒久蕴，不论是化疗还是放疗，均会耗伤气血津液，故见肝肾阴虚之证。由于肝肾阴液是冲任二脉的物质基础，肝肾阴虚则精血不足，故冲任失调。且古今医家皆认为乳腺癌的形成与“肝气不舒郁积而成”有关系，心理学中抑郁内向的 C 型人格特征也被认为是肿瘤发生的高危因素之一，所以Ⅳ期三阴乳腺癌患者多有肝气郁结证的表现。

3. 模型应用

模型结果表明 TNM 分期为Ⅳ期的三阴乳腺癌患者证型主要为肝肾阴虚证、热毒蕴结证、肝气郁结证和冲任失调证。其中，Ⅳ期患者肝肾阴虚证和肝气郁结证的临床表现较为突出，其置信度最大达到 87.96%，且肝肾阴虚证临床表现都存在。故当Ⅳ期患者出现肝肾阴虚证之表现时，应当选取滋补肝肾、清热解毒类抗癌中药，以滋养肝肾为补，清热解毒为攻，攻补兼施，截断热毒蕴结证的出现，为患者接受进一步治疗争取机会。由于患者多有肝气郁结证的表现，在进行治疗时须本着身心一体、综合治疗的精神，重视心理调适。一方面要在药方中注重疏肝解郁，另一方需要及时疏导患者抑郁、焦虑的不良情绪，帮助患者建立合理的认知，树立继续治疗延长生存期的勇气。

8.3　上机实验

1. 实验目的

- ❑ 掌握 Python 结合 Pandas 实现 Apriori 关联算法的过程。
- ❑ 了解 Apriori 关联算法的输入与输出的数据形式，且需要注意对输出数据进行相应的筛选。

2. 实验内容

- ❑ 用 Pandas 读取案例的事务集，每一行为一个事务集。调用下载资源的“上机实验 / code/”目录中的关联算法函数，输入算法的最小支持度与最小置信度，获得中医症型系数与患者 TNM 分期的关联关系规则，并将规则进行保存。
- ❑ 依据分析的目标，编写过滤函数代码，从输出结果中筛选与分析目标相关的规则，并按照特定的格式进行保存。

3. 实验方法与步骤

1）打开 Python，使用 Pandas 库中的 read_csv() 函数将关联分析的数据“demo/data/apriori.txt”读入到工作环境中，其中每个事务集为一行，每行事务集的分隔符默认为字符‘,’。如“A2, B1, C3, D3, E1, F1, H1”这样的一行数据为一个事务集。

2）将读入的“demo/data/apriori.txt”文档中的事务集转换为 0，1 矩阵，每一行事务集为 0，1 矩阵的一行，以方便规则的寻找与记录。

3）根据支持度找出频繁集，直至找到最大频繁集后停止。

4）根据置信度得到大于等于置信度的规则，即为 Apriori 算法所求的关联规则。

5）对 Apriori 算法输出的规则，编写过滤函数。因为该实验探究的是表 8-15 中 6 个症型系数与患者 TNM 分期的规则，所以只留下关联规则中后项有 H 的规则，得到的相应结果展示见表 8-16。

4. 思考与实验总结

1）Python 的流行库中都没有自带的关联规则函数，因此本书编写了相应的关联规则函数，该函数依赖于 Pandas 库。该函数是很高效的（就实现 Apriori 算法而言），可作为工具函数在需要时使用。

2）Apriori 算法的关键两步为找频繁集与根据置信度筛选规则，明白这两步才能清晰地编写相应程序，读者可按照自己的思路编写与优化关联规则程序。

3）本案例采用聚类的方法进行数据离散化，读者可以自己上机实验其他的离散化方法，如等距、等频、决策树、基于卡方检验等，试比较各个方法的优缺点。

8.4 拓展思考

利用本章案例中的原始数据（各属性说明见表 8-17），采用 Apriori 关联规则算法，分析中医证型系数与病程阶段、转移部位和确诊后几年发现转移 3 个指标的关联分析。

表8-17 关联规则模型输入变量

序 号	变量名称	变量描述 / 取值范围
1	肝气郁结证型系数	0 ～ 1
2	热毒蕴结证型系数	0 ～ 1
3	冲任失调证型系数	0 ～ 1
4	气血两虚证型系数	0 ～ 1
5	脾胃虚弱证型系数	0 ～ 1
6	肝肾阴虚证型系数	0 ～ 1
7	病程阶段	S1：围手术期；S2：围化疗期； S3：围放疗期；S4：巩固期
8	转移部位	R1：骨；R2：肺；R3：脑 R4：肝；R5：其他；R0：无转移
9	确诊后几年发现转移	J0：未转移；J1：小于等于 3 年 J2：3 年以上，小于等于 5 年；J3：5 年以上

8.5 小结

本章结合中医证型关联规则的案例，重点介绍了数据挖掘算法中 Apriori 关联算法在实际案例中的应用，并详细地描述了数据获取、数据离散化以及模型构建的过程，最后对其相应的算法及过程给出了 Python 上机实验。

第 9 章 Chapter 9

基于水色图像的水质评价

9.1 背景与挖掘目标

有经验的从事渔业生产的从业者可通过观察水色变化调控水质，以维持养殖水体生态系统中浮游植物、微生物类、浮游动物等合理的动态平衡。由于这些多是通过经验和肉眼观察进行判断的，存在主观性引起的观察性偏倚，使观察结果的可比性、可重复性降低，不易推广应用。当前，数字图像处理技术为计算机监控技术在水产养殖业的应用提供更大的空间。在水质在线监测方面，数字图像处理技术是基于计算机视觉的，以专家经验为基础，对池塘水色进行优劣分级，实现对池塘水色的准确快速判别。

在“上机实验 /data/images”目录下给出了某地区的多个罗非鱼池塘水样的数据，包含水产专家按水色判断水质分类的数据以及用数码相机按照标准进行水色采集的数据（见表 9-1，图 9-1），每个水质图片命名规则为“类别 _ 编号 .jpg”，如“1_1.jpg”说明当前图片属于第 1 类的样本。请根据这些数据，利用图像处理技术，通过水色图像实现水质的自动评价。

表9-1 水色分类

水色	浅绿色（清水或浊水）	灰蓝色	黄褐色	茶褐色（姜黄、茶褐、红褐、褐中带绿等）	绿色（黄绿、油绿、蓝绿、墨绿、绿中带褐等）
水质类别	1	2	3	4	5

9.2 分析方法与过程

通过拍摄水样，采集得到水样图像，而图像数据的维度过大，不容易分析，需要从中提

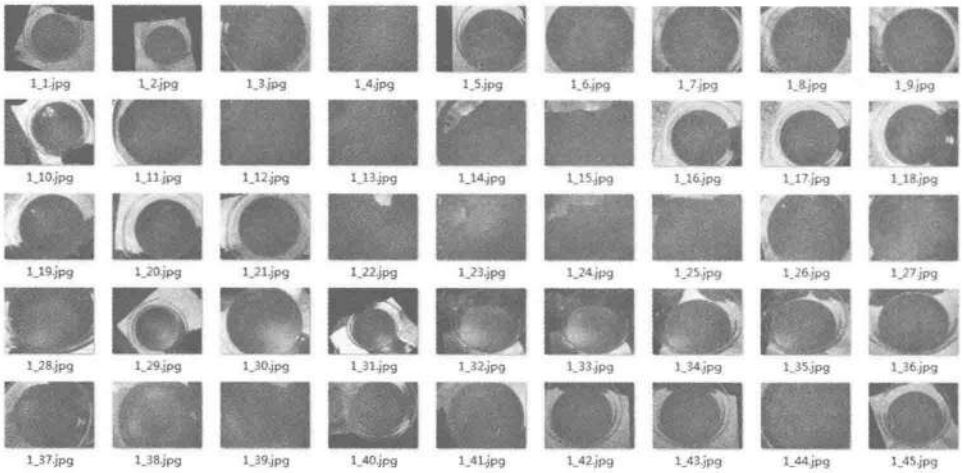


图 9-1 标准条件下拍摄的水样图像

数据详见：test/data/images/

取水样图像的特征，提取反映图像本质的一些关键指标，以达到自动进行图像识别或分类的目的。显然，图像特征提取是图像识别或分类的关键步骤，图像特征提取的效果直接影响到图像识别和分类的好坏。

图像特征主要包括颜色特征、纹理特征、形状特征和空间关系特征等。与几何特征相比，颜色特征更为稳健，对于物体的大小和方向均不敏感，表现出较强的鲁棒性。由于本案例中水色图像是均匀的，故主要关注颜色特征。颜色特征是一种全局特征，描述了图像或图像区域所对应的景物的表面性质。一般颜色特征是基于像素点的特征，所有属于图像或图像区域的像素都有各自的贡献。在利用图像的颜色信息进行图像处理、识别、分类的研究中，在实现方法上已有大量的研究成果，主要采用颜色处理常用的直方图法和颜色矩方法等。

颜色直方图是最基本的颜色特征表示方法，它反映的是图像中颜色的组成分布，即出现了哪些颜色以及各种颜色出现的概率。其优点在于它能简单描述一幅图像中颜色的全局分布，即不同色彩在整幅图像中所占的比例，特别适用于描述那些难以自动分割的图像和不需要考虑物体空间位置的图像。其缺点在于它无法描述图像中颜色的局部分布及每种色彩所处的空间位置，即无法描述图像中的某一具体的对象或物体。

基于颜色矩[18]提取图像特征的数学基础为图像中任何的颜色分布均可以用它的矩来表示。根据概率论的理论，随机变量的概率分布可以由其各阶矩唯一的表示和描述。一幅图像的色彩分布也可认为是一种概率分布，那么图像可以由其各阶矩来描述。颜色矩包含各个颜色通道的一阶距、二阶矩和三阶矩，对于一幅 RGB 颜色空间的图像，具有 R、G 和 B 三个颜色通道，则有 9 个分量。

颜色直方图产生的特征维数一般大于颜色矩的特征维数，为了避免过多变量影响后续的分类效果，在本案例中选择采用颜色矩来提取水样图像的特征，即建立水样图像与反映该图像特征的数据信息关系，同时由有经验的专家对水样图像根据经验进行分类，建立水样数据

信息与水质类别的专家样本库，进而构建分类模型，得到水样图像与水质类别的映射关系，并经过不断调整系数优化模型，最后利用训练好的分类模型，用户就能方便地通过水样图像，自动判别出该水样的水质类别。图 9-2 为基于水色图像特征提取的水质评价流程，主要包括以下步骤。

1）从采集到的原始水样图像中进行选择性抽取与实时抽取，形成建模数据和增量数据。

2）对 1）形成的两个数据集进行数据预处理，包括图像切割和颜色矩特征提取。

3）利用 2）形成的已完成数据预处理的建模数据，由有经验的专家对水样图像根据经验进行分类，构建专家样本。

4）利用 3）的专家样本构建分类模型。

5）利用 4）的构建好的分类模型进行水质评价。

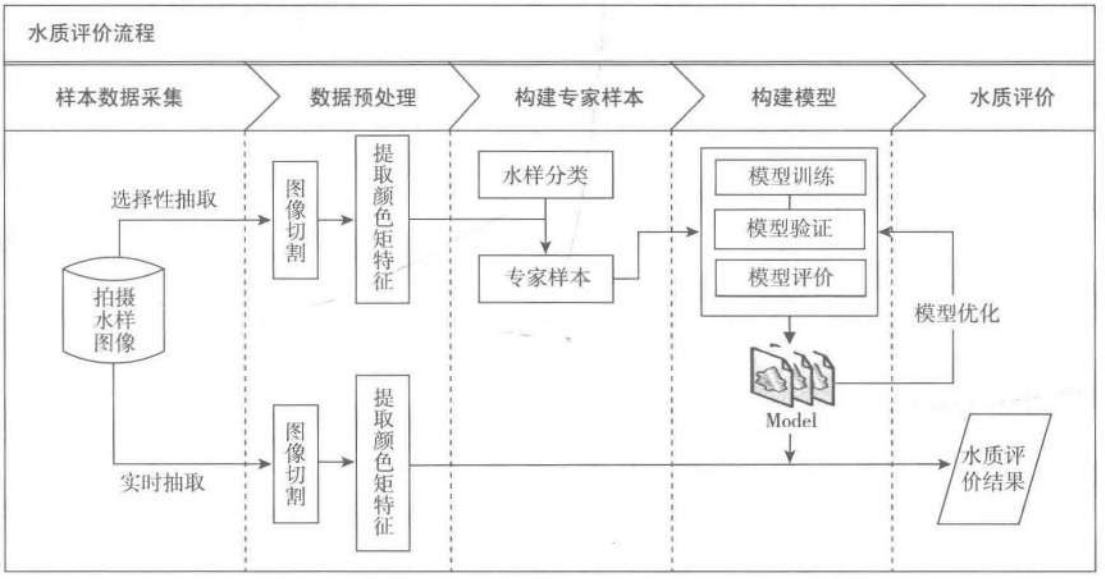


图 9-2　基于水色图像特征提取的水质评价流程

9.2.1　数据预处理

1. 图像切割

采集到的水样图像包含盛水容器，容器的颜色与水体颜色差异较大，同时水体位于图像中央，为了提取水色的特征，需要提取水样图像中央部分具有代表意义的图像，具体实施方式是提取水样图像中央 101×101 像素的图像。设原始图像 I 的大小是 $M \times N$，则截取宽从第 $fix\left(\frac{M}{2}\right)-50$ 个像素点到第 $fix\left(\frac{M}{2}\right)+50$ 个像素点，长从第 $fix\left(\frac{N}{2}\right)-50$ 个像素点到第 $fix\left(\frac{M}{2}\right)+50$ 个像素点的子图像。

使用其他编程软件进行编程，即可把图 9-3 中左边的切割前的水样图像切割并保存到右

边的切割后的水样图像。

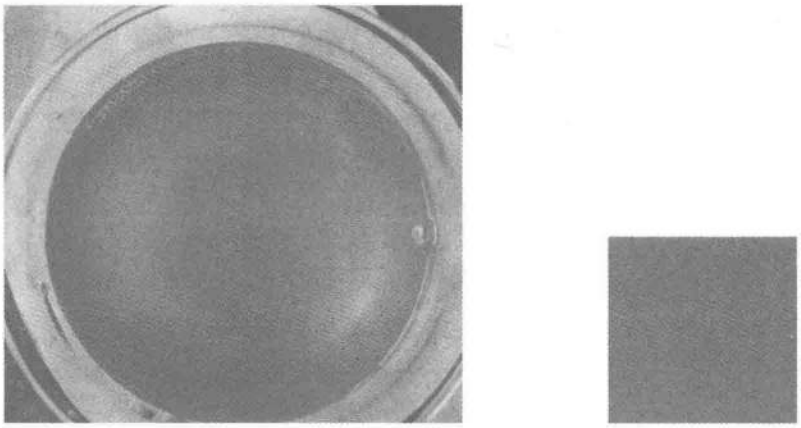


图 9-3 切割前水样图像（左）和切割后水样图像（右）

2. 特征提取

在本案例中选择采用颜色矩来提取水样图像的特征，下面给出各阶颜色矩的计算公式。

（1）一阶颜色矩

一阶颜色矩采用一阶原点矩，反映图像的整体明暗程度。

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \tag{9-1}$$

其中，E_i 是在第 i 个颜色通道的一阶颜色矩，对于 RGB 颜色空间的图像，$i = 1, 2, 3$，p_{ij} 是第 j 个像素的第 i 个颜色通道的颜色值。

（2）二阶颜色矩

二阶颜色矩采用的是二阶中心距的平方根，反映图像颜色的分布范围。

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N} (p_{ij} - E_i)^2} \tag{9-2}$$

其中，δ_i 是在第 i 个颜色通道的二阶颜色矩，E_i 是在第 i 个颜色通道的一阶颜色矩。

（3）三阶颜色矩

三阶颜色矩采用的是三阶中心距的立方根，反映图像颜色分布的对称性。

$$s_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N} (p_{ij} - E_i)^3} \tag{9-3}$$

其中，s_i 是在第 i 个颜色通道的三阶颜色矩，E_i 是在第 i 个颜色通道的一阶颜色矩。

提取切割后的图像颜色矩，作为图像的颜色特征。在颜色矩的提取中，提取每个文件名中的类别和序号，同时针对所有的图片都进行同样的操作，即可得到表 9-2 的数据。

表9-2　水色图像特征与相应的水色类别的部分数据

水质类别	序号	R 通道一阶矩	G 通道一阶矩	B 通道一阶矩	R 通道二阶矩	G 通道二阶矩	B 通道二阶矩	R 通道三阶矩	G 通道三阶矩	B 通道三阶矩
1	1	0.582 823	0.543 774	0.252 829	0.014 192	0.016 144	0.041 075	–0.012 64	–0.016 09	–0.041 54
2	1	0.495 169	0.539 358	0.416 124	0.011 314	0.009 811	0.014 751	0.015 367	0.016 01	0.019 748
3	1	0.510 911	0.489 695	0.186 255	0.012 417	0.010 816	0.011 644	–0.007 47	–0.007 68	–0.005 09
4	1	0.420 351	0.436 173	0.167 221	0.011 22	0.007 195	0.010 565	–0.006 28	0.003 173	–0.007 29
5	1	0.211 567	0.335 537	0.111 969	0.012 056	0.013 296	0.008 38	0.007 305	0.007 503	0.003 65
1	2	0.563 773	0.534 851	0.271 672	0.009 723	0.007 856	0.011 873	–0.005 13	0.003 032	–0.005 47
2	2	0.465 186	0.508 643	0.361 016	0.013 753	0.012 709	0.019 557	0.022 785	0.022 329	0.031 616
3	2	0.533 052	0.506 734	0.185 972	0.011 104	0.007 902	0.012 65	0.004 797	–0.002 9	0.004 214
4	2	0.398 801	0.425 56	0.191 341	0.014 424	0.010 462	0.015 47	0.009 207	0.006 471	0.006 764
5	2	0.298 194	0.427 725	0.097 936	0.014 778	0.012 456	0.008 322	0.008 51	0.006 117	0.003 47
1	3	0.630 328	0.594 269	0.298 577	0.007 731	0.005 877	0.010 148	0.003 447	–0.003 45	–0.006 53
2	3	0.491 916	0.546 367	0.425 871	0.010 344	0.008 293	0.012 26	0.009 285	0.009 663	0.011 549
3	3	0.559 437	0.522 702	0.194 201	0.012 478	0.007 927	0.012 183	0.004 477	–0.003 41	–0.005 29
4	3	0.402 068	0.431 443	0.177 364	0.010 554	0.007 287	0.010 748	0.006 261	–0.003 41	0.006 419
5	3	0.408 963	0.486 953	0.178 113	0.012 662	0.009 752	0.014 497	–0.006 72	0.002 168	0.009 992
1	4	0.638 606	0.619 26	0.319 711	0.008 125	0.006 045	0.009 746	–0.004 87	0.003 083	–0.004 5

数据详见：demo/data/moment.csv

9.2.2　模型构建

1. 模型输入

对特征提取后的样本进行抽样，抽取 80% 作为训练样本，剩下的 20% 作为测试样本，用于水质评价检验。其数据抽样代码如代码清单 9-1 所示。

代码清单9-1　数据抽样代码

```
#-*- coding: utf-8 -*-
import pandas as pd

inputfile = '../data/moment.csv' #数据文件
data = pd.read_csv(inputfile, encoding = 'gbk') #读取数据，指定编码为gbk
data = data.as_matrix()

from random import shuffle #引入随机函数
shuffle(data) #随机打乱数据
data_train = data[:int(0.8*len(data)), :] #选取前80%为训练数据
data_test = data[int(0.8*len(data)):, :] #选取前20%为测试数据
```

代码详见：demo/code/svm.py

本案例采用支持向量机作为水质评价分类模型，模型的输入包括两部分，一部分是训练样本的输入，另一部分是建模参数的输入。各参数说明如表 9-3 所示。

表9-3 预测模型输入变量

序 号	变量名称	变量描述	取值范围
1	R 通道一阶矩	水样图像在 R 颜色通道的一阶矩	0 ～ 1
2	G 通道一阶矩	水样图像在 G 颜色通道的一阶矩	0 ～ 1
3	B 通道一阶矩	水样图像在 B 颜色通道的一阶矩	0 ～ 1
4	R 通道二阶矩	水样图像在 R 颜色通道的二阶矩	0 ～ 1
5	G 通道二阶矩	水样图像在 G 颜色通道的二阶矩	0 ～ 1
6	B 通道二阶矩	水样图像在 B 颜色通道的二阶矩	0 ～ 1
7	R 通道三阶矩	水样图像在 R 颜色通道的三阶矩	−1 ～ 1
8	G 通道三阶矩	水样图像在 G 颜色通道的三阶矩	−1 ～ 1
9	B 通道三阶矩	水样图像在 B 颜色通道的三阶矩	−1 ～ 1
10	水质类别	不同类别能表征水中浮游植物的种类和多少	1，2，3，4，5

其中，1 ～ 9 为输入的特征，我们发现特征的取值范围都在 0 ～ 1 之间，也就是说，如果直接输入 SVM 模型的话，彼此之间区分度会比较小，因此，我们不妨将所有特征都统一乘以一个适当的常数 k，从而提高区分度和准确率。常数 k 的选取不能过大也不能过小，过小导致区分度较低，模型精确度差，较大则容易导致模型在训练样本中过拟合（如 k ＝ 1 000 时，在训练样本准确率为 100%，在测试样本准确率不到 20%），总的来说，k 过大过小都会导致测试样本的准确率下降，所以可以根据测试准确率来选择 k 的最优值。

经过反复测试，本次建模中 k 的最优值大约为 30，因此，我们将输入的特征统一乘以 30，然后建立支持向量机模型，代码如代码清单 9-2 所示。

代码清单9-2 构建支持向量机模型代码（接9-1）

```
#构造特征和标签
x_train = data_train[:, 2:]*30 #放大特征
y_train = data_train[:, 0].astype(int)
x_test = data_test[:, 2:]*30 #放大特征
y_test = data_test[:, 0].astype(int)

#导入模型相关的函数，建立并且训练模型
from sklearn import svm
model = svm.SVC()
model.fit(x_train, y_train)
import pickle
pickle.dump(model, open('../tmp/svm.model', 'wb'))
#最后一句保存模型，以后可以通过下面语句重新加载模型：
#model = pickle.load(open('../tmp/svm.model', 'rb'))

#导入输出相关的库，生成混淆矩阵
```

```
from sklearn import metrics
cm_train = metrics.confusion_matrix(y_train, model.predict(x_train)) #训练样本的混
    淆矩阵
cm_test = metrics.confusion_matrix(y_test, model.predict(x_test)) #测试样本的混淆矩阵

#保存结果
pd.DataFrame(cm_train, index = range(1, 6), columns = range(1, 6)).to_excel(outputfile1)
pd.DataFrame(cm_test, index = range(1, 6), columns = range(1, 6)).to_excel(output-
    file2)
```

代码详见：demo/code/svm.py

2. 结果及分析

建立模型后，利用训练样本进行回判，得到的混淆矩阵见表 9-4，分类准确率为 96.91%，分类效果较好，可应用模型进行水质评价。

表9-4　模型混淆矩阵

预测值 / 实际值	1	2	3	4	5
1	41	1	1	0	0
2	0	34	0	0	0
3	0	0	59	0	0
4	0	0	1	20	0
5	0	1	0	1	3

9.2.3　水质评价

取所有测试样本为输入样本，代入已构建好的支持向量机模型，得到输出结果，即预测的水质类型。水质评价的混淆矩阵见表 9-5，分类准确率为 95.12%，说明水质评价模型对于新增的水色图像的分类效果较好，可将模型应用到水质自动评价系统，实现水质评价。（注意，由于用随机函数来打乱数据，因此重复试验所得到的结果可能有所不同。）

表9-5　水质评价的混淆矩阵

预测值 / 实际值	1	2	3	4	5
1	7	0	1	0	0
2	0	10	0	0	0
3	0	0	19	0	0
4	0	0	0	3	0
5	0	0	0	1	0

9.3 上机实验

1. 实验目的

加深对支持向量机原理的理解及使用。

2. 实验内容

实验数据是截取后的图像的颜色矩特征，包括一阶矩、二阶矩、三阶矩，同时由于图像具有 R、G 和 B 三个颜色通道，所以颜色矩特征具有 9 个分量。结合水质类别和颜色矩特征构成专家样本数据，以水质类别作为目标输出，构建支持向量机模型，并利用混淆矩阵评价模型优劣。

注意 数据 80% 作为训练样本，剩下的 20% 作为测试样本。

3. 实验方法与步骤

1）使用 read_csv() 函数把经过预处理的专家样本数据“ test/data/moment.csv”读入当前工作空间。

2）把工作空间的建模数据随机分为两部分，一部分用于训练，一部分用于测试。

3）使用 Scikit-Learn 里的 svm() 函数以及训练数据构建支持向量机模型，使用 predict() 函数和构建的支持向量机模型分别对训练数据进行分类，使用 Scikit-Learn 库的子库 metrics 中的 confusion_matrix() 函数求出混淆矩阵，如果仅仅是想知道准确率，可以用 model.score(x_test, y_test) 的方式返回。

4）使用 predict() 函数和步骤 3）构建好的支持向量机模型分别对测试数据进行分类，参考步骤 3）得到模型分类正确率和混淆矩阵。

4. 思考与实验总结

1）如何在 Python 环境下处理图像数据？

2）支持向量机模型的参数有哪些可以设置，如何针对数据特征进行参数择优选择？

9.4 拓展思考

我国环境质量评价工作是 20 世纪 70 年代后才逐步发展起来的。发展至今，在评价指标体系及评价理论探索等方面均有较大进展。但目前在我国环境评价实际工作中，所采用的通常是一些比较传统的评价方法，往往是从单个污染因子的角度对其进行简单评价。然而，对某区域的环境质量（如水质、大气质量等）的综合评价一般涉及较多的评价因素，且各因素与区域环境整体质量关系复杂，因而采用单项污染指数评价法无法客观准确地反映各污染因子之间相互作用对环境质量的影响。

基于上述原因，要客观评价一个区域的环境质量状况，需要综合考虑各种因素之间以及影响因素与环境质量之间错综复杂的关系。采用传统的方法存在着一定的局限性和不合理性。因此，从学术研究的角度对环境评价的技术方法及其理论进行探讨，寻求能更全面、客观、准确反映环境质量的新的理论方法具有重要的现实意义。

有人根据空气中 SO_2、NO、NO_2、NO_x、PM10 和 PM2.5 的含量，建立分类预测模型，实现对空气质量的评价。在某地实际监测的部分原始样本数据经预处理后如表 9-6 所示（完整数据见：/ 拓展思考 / 拓展思考样本数据 .xls）。请采用 C4.5 决策树进行模型构建，并评价模型效果。

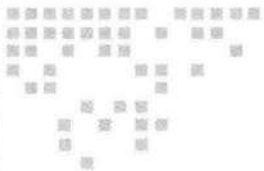
表9-6　建模样本数据

SO_2	NO	NO_2	NO_x	PM_{10}	$PM_{2.5}$	空气等级
0.031	0	0.046	0.047	0.085	0.058	Ⅰ
0.022	0	0.053	0.053	0.07	0.048	Ⅱ
0.017	0	0.029	0.029	0.057	0.04	Ⅰ
0.026	0	0.026	0.026	0.049	0.034	Ⅰ
0.018	0	0.027	0.027	0.051	0.035	Ⅰ
0.019	0	0.052	0.053	0.06	0.04	Ⅱ
0.022	0	0.059	0.06	0.064	0.042	Ⅱ
0.023	0.01	0.085	0.099	0.07	0.044	Ⅱ
0.022	0.012	0.066	0.084	0.073	0.042	Ⅱ
0.017	0.007	0.037	0.048	0.069	0.04	Ⅰ

数据详见：拓展思考 / 拓展思考样本数据 .xls

9.5　小结

本章结合基于水色图像进行水质评价的案例，重点介绍了图像处理算法中的颜色矩提取和数据挖掘算法中支持向量机算法在实际案例中的应用。利用水色图像颜色矩的特征，采用支持向量机算法进行水质评价，并详细地描述了数据挖掘的整个过程，也对其相应的算法给出了 Python 语言上机实验。



Chapter 10 第 10 章

家用电器用户行为分析与事件识别

10.1 背景与挖掘目标

居民在使用家用电器过程中，会因地区气候、不同区域、用户年龄性别差异，形成不同的使用习惯。家电企业若能深入了解不同用户群的使用习惯，开发新功能，就能开拓新市场。

要了解用户使用家用电器的习惯，必须采集用户使用电器的相关数据下面以热水器为例，分析用户的使用行为。在热水器用户行为分析过程中，用水事件识别是最关键的环节。比如，国内某热水器生产厂商新研发的一种高端智能热水器，在状态发生改变或者有水流状态时，会采集各监控指标数据。该厂商根据其采集的用户的用水数据，分析用户的用水行为特征。热水器采集到用户用水数据见表 10-1。由于用户不仅仅使用热水器来洗浴，还可能包括洗手、洗脸、刷牙、洗菜、做饭等用水行为，所以热水器采集到的数据来自各种不同的用水事件。本案例基于热水器采集的时间序列数据，将顺序排列的离散的用水时间节点根据水流量和停顿时间间隔划分为不同大小的时间区间，每个区间是一个可理解的一次完整用水事件，并以热水器一次完整用水事件为一个基本事件，将时间序列数据划分为独立的用水事件并识别出其中属于洗浴的事件。基于以上工作，该厂商可从热水器智能操作和节能运行等多方面对产品进行优化。

热水器厂商根据洗浴事件识别模型，对不同地区的用户的用水进行识别，根据识别结果比较不同客户群的客户使用习惯、加深对客户的理解等。从而，厂商可以给不同的客户群提供最适合的个性化产品、改进新产品的智能化的研发和制定相应的营销策略。

请根据提供的数据实现以下目标。

1）根据热水器采集到的数据，划分一次完整用水事件。

2）在划分好的一次完整用水事件中，识别出洗浴事件。

表10-1　热水器用户用水数据

热水器编号	发生时间	开关机状态	加热中	保温中	有无水流	实际温度	热水量	水流量	节能模式	加热剩余时间	当前设置温度
R_00001	20141019160855	开	开	关	无	47℃	25%	0	关	4 分钟	50℃
R_00001	20141019160954	开	开	关	无	47℃	25%	0	关	2 分钟	50℃
R_00001	20141019161040	开	开	关	无	48℃	25%	0	关	2 分钟	50℃
R_00001	20141019161042	开	开	关	无	48℃	25%	0	关	1 分钟	50℃
R_00001	20141019161106	开	开	关	无	49℃	25%	0	关	1 分钟	50℃
R_00001	20141019161147	开	开	关	无	49℃	25%	0	关	0 分钟	50℃
R_00001	20141019161149	开	关	开	无	50℃	100%	0	关	0 分钟	50℃
R_00001	20141019172319	开	关	开	无	50℃	50%	0	关	0 分钟	50℃
R_00001	20141019172321	关	关	关	有	50℃	50%	62	关	0 分钟	50℃
R_00001	20141019172323	关	关	关	有	50℃	50%	63	关	0 分钟	50℃
R_00001	20141019172325	关	关	关	有	50℃	50%	61	关	0 分钟	50℃
R_00001	20141019172331	关	关	关	有	50℃	50%	62	关	0 分钟	50℃
R_00001	20141019172333	关	关	关	有	50℃	50%	63	关	0 分钟	50℃
R_00001	20141019172337	关	关	关	有	50℃	50%	62	关	0 分钟	50℃
R_00001	20141019172341	关	关	关	有	50℃	50%	63	关	0 分钟	50℃
R_00001	20141019172456	关	关	关	无	50℃	50%	0	关	0 分钟	50℃
R_00001	20141019172458	关	关	关	有	50℃	50%	46	关	0 分钟	50℃
R_00001	20141019172500	关	关	关	有	50℃	50%	50	关	0 分钟	50℃
R_00001	20141019172505	关	关	关	有	50℃	50%	51	关	0 分钟	50℃
R_00001	20141019172506	关	关	关	有	50℃	50%	50	关	0 分钟	50℃
R_00001	20141019172512	关	关	关	有	50℃	50%	51	关	0 分钟	50℃

数据详见：demo/data/original_data.xls

10.2　分析方法与过程

本次数据挖掘建模的总体流程如图 10-1 所示。

热水器用户用水事件划分与识别主要包括以下步骤。

1）对热水用户的历史用水数据进行选择性抽取，构建专家样本。

2）对步骤 1）形成的数据集进行数据探索分析与预处理，包括探索用水事件时间间隔的分布、规约冗余属性、识别用水数据的缺失值，并对缺失值进行处理，根据建模的需要进行属性构造等。根据以上处理，对用水样本数据建立用水事件时间间隔识别模型和划分一次完整的用水事件模型，再在一次完整用水事件划分结果的基础上，剔除短暂用水事件，缩小识别范围。

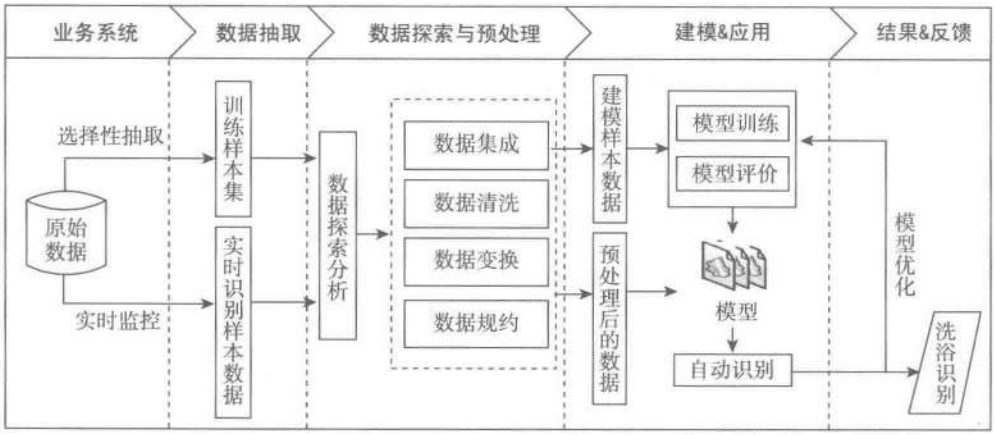


图 10-1 热水器用户用水识别建模总体流程

3）在步骤 2）得到的建模样本数据基础上，建立洗浴事件识别模型，对洗浴事件识别模型进行模型分析评价。

4）对步骤 3）形成的模型结果应用并对洗浴事件划分进行优化。

5）调用洗浴事件识别模型，对实时监控的热水器流水数据进行洗浴事件自动识别。

10.2.1 数据抽取

在使用热水器的过程中，热水器的状态会经常发生改变，比如开机和关机、由加热转到保温、由无水流到有水流、水温由 50℃变为 49℃等。而智能热水器在状态发生改变或者水流量非零时，每两秒会采集一条状态数据。由于数据的采集频率较高，并且数据来自大量用户，数据总量非常大。本案例对原始数据采用无放回随机抽样法抽取 200 家热水器用户从 2014 年 1 月 1 日至 2014 年 12 月 31 日的用水记录作为原始建模数据。

热水器采集的用水数据包含以下 12 个属性：热水器编码、发生时间、开关机状态、加热中、保温中、有无水流、实际温度、热水量、水流量、节能模式、加热剩余时间、当前设置温度。12 个属性的说明见表 10-2，具体的数据见表 10-1。

表10-2 热水器属性说明

属性名称	属性说明
热水器编码	热水器出厂编号
发生时间	记录热水器处于某状态的时刻
开关机状态	热水器是否开机
加热中	热水器处于对水进行加热的状态
保温中	热水器处于对水进行保温的状态
有无水流	热水水流量大于等于 10L/min 为有水，否则为无

（续）

属性名称	属性说明
实际温度	热水器中热水的实际温度
热水量	热水器热水的含量
水流量	热水器热水的水流速度 单位：L/min
节能模式	热水器的一种节能工作模式
加热剩余时间	加热到设定温度还需多长时间
当前设置温度	热水器加热时热水能够到达的最大温度

10.2.2　数据探索分析

用水停顿时间间隔为一条水流量不为 0 的流水记录同下一条水流量不为 0 的流水记录之间的时间间隔。根据现场实验统计，两次用水过程的用水停顿的间隔时长一般不大于 4 分钟。为了探究用户真实用水停顿时间间隔的分布情况，统计用水停顿的时间间隔并作频率分布直方图。通过频率分布直方图分析用户用水停顿时间间隔的规律性，从而探究划分一次完整用水事件的时间间隔阈值。具体的数据见表 10-3。

表10-3　用水停顿时间间隔频数分布表　（单位：分钟）

间隔时长	0 ～ 0.1	0.1 ～ 0.2	0.2 ～ 0.3	0.3 ～ 0.5	0.5 ～ 1	1 ～ 2	2 ～ 3	3 ～ 4	4 ～ 5
停顿频率	78.71%	9.55%	2.52%	1.49%	1.46%	1.29%	0.74%	0.48%	0.26%
间隔时长	5 ～ 6	6 ～ 7	7 ～ 8	8 ～ 9	9 ～ 10	10 ～ 11	11 ～ 12	12~13	13 以上
停顿频率	0.27%	0.19%	0.17%	0.12%	0.09%	0.09%	0.10%	0.11%	2.36%

分析表 10-3 可知，停顿时间间隔为 0 ～ 0.3 分钟的频率很高，根据日常用水经验可以判断其为一次用水时间中的停顿；停顿时间间隔为 6 ～ 13 分钟的频率较低，分析其为两次用水事件之间的停顿间隔。两次用水事件的停顿时间间隔分布在 3 ～ 7 分钟。根据现场实验统计用水停顿的时间间隔近似。

10.2.3　数据预处理

本案例的数据集的特点是数据量涉及上万个用户而且每个用户每天的用水数据多达数万条、存在缺失值、与分析主题无关的属性或未直接反应用水事件的属性等。在数据预处理阶段，针对这些情况相应地应用了缺失值处理、数据规约和属性构造等来解决这些问题。

1. 数据规约

由于热水器采集的用水数据属性较多，本案例对建模数据做以下数据规约。

- 属性规约：因为要对热水器用户的洗浴行为的一般规律进行挖掘分析，所以“热水器编号”可以去除；因热水器采集的数据中，“有无水流”可以通过“水流量”反映

出来，“节能模式”数据都只为“关”，对建模无作用，可以去除。最终用来建模的属性指标如表 10-4 所示。

- 数值规约：当热水器“开关机状态”为“关”且水流量为 0 时，说明热水器不处于工作状态，数据记录可以规约掉。

表10-4 属性规约后部分数据列表

发生时间	开关机状态	加热中	保温中	实际温度	热水量	水流量	加热剩余时间	当前设置温度
20141019161042	开	开	关	48℃	25%	0	1 分钟	50℃
20141019161106	开	开	关	49℃	25%	0	1 分钟	50℃
20141019161147	开	开	关	49℃	25%	0	0 分钟	50℃
20141019161149	开	关	开	50℃	100%	0	0 分钟	50℃
20141019172319	开	关	开	50℃	50%	0	0 分钟	50℃
20141019172321	关	关	关	50℃	50%	62	0 分钟	50℃
20141019172323	关	关	关	50℃	50%	63	0 分钟	50℃

数据详见：demo/data/water_heater.xls

2. 数据变换

由于本案例的挖掘目标是对热水器用户的洗浴事件进行识别，这就需要从原始数据中识别出哪些状态记录是一个完整的用水事件（包括洗脸、洗手、刷牙、洗头、洗菜和洗浴等），从而再识别出用水事件中的洗浴事件；一次完整的用水事件是根据水流量和停顿时间间隔的阈值去划分的，所以本案例还建立了阈值寻优模型；为了提高在大量的一次完整用水事件中寻找洗浴事件的效率，本案例建立了筛选规则剔除可以明显判定不是洗浴的事件，得到建模数据样本集。数据变换流程如图 10-2 所示。

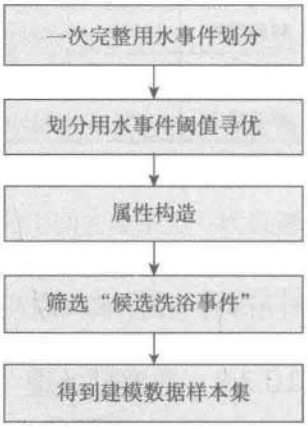


图 10-2 数据变换流程图

（1）一次完整用水事件的划分模型

用户的用水数据存储在数据库中，记录了各种各样的用水事件，包括洗浴、洗手、刷牙、洗脸、洗衣和洗菜等，而且一次用水事件由数条甚至数千条的状态记录组成。所以，本案例首先需要在大量的状态记录中划分出哪些连续的数据是一次完整的用水事件。

在用水状态记录中，水流量不为 0 表明用户正在使用热水；而水流量为 0 时用户用热水发生停顿或者用热水结束。如果水流量为 0 的状态记录之间的时间间隔超过一个阈值 T，则从该段水流量为 0 的状态记录向前找到最后一条水流量不为 0 的用水记录作为上一次用水事件的结束；向后找到水流量不为 0 的状态记录作为下一个用水事件的开始。划分模型的符号说明见表 10-5。

表10-5　一次完整用水事件模型构建符号说明表

名　称	符　号
状态记录 i	R_i　$i \in \{1, 2 \cdots n\}$
时间间隔阈值	T
R_{i+1} 与 R_i 之间的时间间隔	gap_i　$i \in \{1, 2 \cdots n\}$

一次完整用水事件的划分步骤如下。

1）读取数据记录，识别到第一条水流量不为 0 的数据记录记为 R_1，按顺序识别接下来的一条水流量不为 0 数据记录为 R_2。

2）若 $gap_i > T$，则 R_{i+1} 与 R_i 及之间的数据记录不能划分到同一次用水事件。同时将 R_{i+1} 记录作为新的读取数据记录的开始，返回步骤 1）；若 $gap_i < T$，则将 R_{i+1} 与 R_i 之间数据记录的划分到同一次用水事件，并记录接下来的水流量不为 0 数据记录为 R_{i+2}。

3）循环执行步骤 2），直到数据记录读取完毕，结束事件划分。

使用 Pandas 对用户的用水数据进行一次完整用水事件的划分，阈值 T 暂时假设为 4 分钟，详细代码如代码清单 10-1 所示。

代码清单10-1　划分一次用水事件代码

```
#-*- coding: utf-8 -*-
#用水事件划分
import pandas as pd

threshold = pd.Timedelta(minutes = 4) #阈值为4分钟
inputfile = '../data/water_heater.xls' #输入数据路径，需要使用Excel格式
outputfile = '../tmp/dividsequence.xls' #输出数据路径，需要使用Excel格式

data = pd.read_excel(inputfile)
data[u'发生时间'] = pd.to_datetime(data[u'发生时间'], format = '%Y%m%d%H%M%S')
data = data[data[u'水流量'] > 0] #只要流量大于0的记录
d = data[u'发生时间'].diff() > threshold #相邻时间作差分，比较是否大于阈值
data[u'事件编号'] = d.cumsum() + 1 #通过累积求和的方式为事件编号

data.to_excel(outputfile)
```

代码详见：demo/code/divide_event.py

对用户的用水数据进行划分，划分结果如表 10-6 所示。

表10-6　用水数据划分结果

	发 生 时 间	...	事 件 编 号
2	2014-10-19 07:01:56	...	1
56	2014-10-19 07:38:16	...	2
381	2014-10-19 09:46:38	...	3

（续）

	发生时间	...	事件编号
382	2014-10-19 09:46:40	...	3
384	2014-10-19 09:47:15	...	3
404	2014-10-19 11:50:17	...	4
...	...	...	...

（2）用水事件阈值寻优模型

考虑到不同地区的人们用热水器的习惯不同，以及不同季节使用热水器时停顿的时长也可能不同，固定的停顿时长阈值对于某些特殊的情况的处理是不理想的，存在把一个事件划分为两个事件或者把两个事件合为一个事件的情况。所以，考虑到在不同的时间段内要更新阈值，本案例建立了阈值寻优模型来更新寻找最优的阈值，这样可以解决因时间变化和地域不同导致阈值存在差异的问题。

对某热水器用户的数据进行了不同阈值划分，得到了相应的事件个数，阈值变化与划分得到的事件个数如表 10-7 所示，阈值与划分事件个数关系如图 10-3 所示。

表10-7 某热水器用户家庭某时间段不同用水时间间隔阈值事件划分个数

阈值（分钟）	2.25	2.5	2.75	3	3.25	3.5	3.75	4	4.25	4.5	4.75	5
事件个数	650	644	626	602	588	565	533	530	530	530	522	520
阈值（分钟）	5.25	5.5	5.75	6	6.25	6.5	6.75	7	7.25	7.5	7.75	8
事件个数	510	506	503	500	480	472	466	462	460	460	460	460

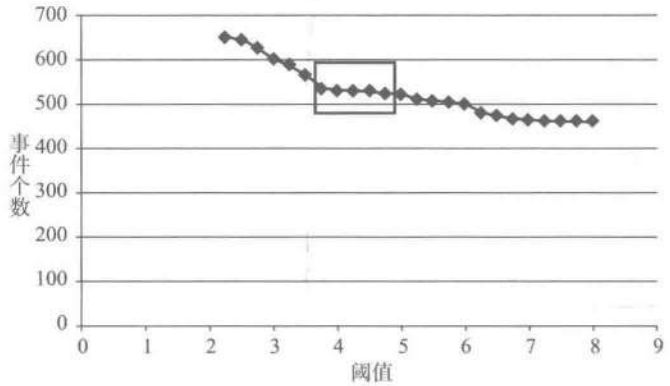


图 10-3 阈值与划分事件个数关系

图 10-3 中某段阈值范围内，下降趋势明显，说明在该段阈值范围内，用户的停顿习惯比较集中。如果趋势比较平缓，则说明用户的停顿习惯趋于稳定，所以取该段时间开始作为阈值，既不会将短的用水事件合并，又不会将长的用水事件拆开。在图 10-3 中，用户停顿热水

的习惯在方框的位置趋于稳定，说明热水器用户的用水停顿习惯用方框开始的时间点作为划分阈值会有一个好的效果。

在图 10-3 中，曲线趋于稳定时，其方框开始的点的斜率趋于一个较小的值。为了用程序来识别这一特征，将这一特征提取为规则。可以从图 10-4 中相邻 2 个点之间斜率值的大小说明程序是如何识别图 10-3 方框中的起始时间的。

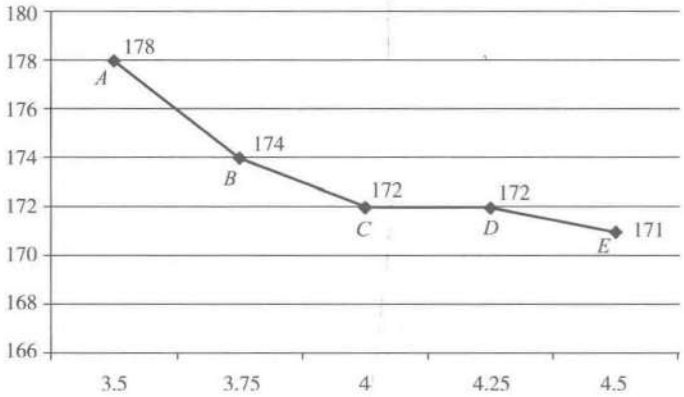


图 10-4　斜率计算图

每个阈值对应一个点，计算每个阈值得到一个斜率指标。如图 10-4 所示，A 点是要计算斜率指标的点，为了直观的展示，用下面的符号来进行说明，见表 10-8。

表10-8　阈值寻优模型符号说明

k_s	相邻两点的斜率的绝对值 $s \in \{AB, BC, CD, DE\}$	K	5 个点的斜率之和的平均值
k	任意两点 (x_1, y_1)，(x_2, y_2) 的斜率的绝对值	(x_i, y_i)	i 点的坐标 $i \in \{A, B, C, D, E\}$

$$k = \left| \frac{y_1 - y_2}{x_1 - x_2} \right| \tag{10-1}$$

根据（10-1）式，计算出 k_{AB}、k_{BC}、k_{CD}、k_{DE} 四个斜率。再计算出四个斜率之和的平均值 K：

$$K = (k_{AB} + k_{BC} + k_{CD} + k_{DE})/4 \tag{10-2}$$

将 K 作为 A 点的斜率指标，特别指出横坐标上的最后 4 个点没有斜率指标，因为找不出在它以后的 4 个更长的阈值。但这不影响对最优阈值的寻找，因为可以提高阈值的上限，以使最后的 4 个阈值不是考虑范围内的阈值。

于是，阈值优化的结果如下：

当存在一个阈值的斜率指标 $K < 1$ 时，则取阈值最小的点 A（可能存在多个阈值的斜率指标小于 1）的横坐标 x_A 作为用水事件划分的阈值，其中 $K < 1$ 中的 1 是经过实际数据验证的一个专家阈值。

当不存在 $K < 1$ 时，则找所有阈值中斜率指标最小的阈值；如果该阈值的斜率指标小于

5，则取该阈值作为用水事件划分的阈值；如果该阈值的斜率指标不小于 5，则阈值取默认值的阈值为 4 分钟。其中，斜率指标小于 5 中的 5 是经过实际数据验证的一个专家阈值。

使用 Python 对用户的用水数据划分阈值进行寻优，寻优区间在 1 分钟～ 9 分钟，详细代码见代码清单 10-2。

代码清单10-2 阈值寻优代码

```
#-*- coding: utf-8 -*-
#阈值寻优
import numpy as np
import pandas as pd

inputfile = '../data/water_heater.xls' #输入数据路径，需要使用Excel格式
n = 4 #使用以后四个点的平均斜率

threshold = pd.Timedelta(minutes = 5) #专家阈值
data = pd.read_excel(inputfile)
data[u'发生时间'] = pd.to_datetime(data[u'发生时间'], format = '%Y%m%d%H%M%S')
data = data[data[u'水流量'] > 0] #只要流量大于0的记录

def event_num(ts):
  d = data[u'发生时间'].diff() > ts #相邻时间作差分，比较是否大于阈值
  return d.sum() + 1 #这样直接返回事件数

dt = [pd.Timedelta(minutes = i) for i in np.arange(1, 9, 0.25)]
h = pd.DataFrame(dt, columns = [u'阈值']) #定义阈值列
h[u'事件数'] = h[u'阈值'].apply(event_num) #计算每个阈值对应的事件数
h[u'斜率'] = h[u'事件数'].diff()/0.25 #计算每两个相邻点对应的斜率
h[u'斜率指标'] = pd.rolling_mean(h[u'斜率'].abs(), n) #采用后n个的斜率绝对值平均作为斜率
    指标
ts = h[u'阈值'][h[u'斜率指标'].idxmin() - n]
#注：用idxmin返回最小值的Index，由于rolling_mean()自动计算的是前n个斜率的绝对值平均
#所以结果要进行平移（-n）

if ts > threshold:
  ts = pd.Timedelta(minutes = 4)

print(ts)
```

代码详见：demo/code/threshold_optimization.py

根据读入的数据文件，进行阈值寻优，得到该段时间用水事件划分的最优阈值为 4 分钟。

（3）属性构造

本案例研究的是用水行为，可构造 4 类指标：时长指标、频率指标、用水的量化指标以及用水的波动指标。具体请参见表 10-9。

表10-9 4类属性指标的构建表

时长指标	用水开始时间、用水结束时间、总用水时长、停顿时长、总停顿时长、用水时长、平均停顿时长、用水时长 / 总用水时长

（续）

频率指标	停顿次数
用水量化指标	总用水量、平均水流量
用水波动指标	水流量波动、停顿时长波动

对一次用水事件抽取主要的用水数据，具体见表 10-10。

表10-10　一次用水事件的用水数据表

发生时间	开关机状态	加热中	保温中	实际温度	热水量	水流量	加热剩余时间	当前设置温度
20141021200010	开	关	开	50℃	100%	0	0 分钟	50℃
20141021200012	开	关	开	50℃	50%	80	0 分钟	50℃
20141021200120	开	关	开	49℃	50%	70	0 分钟	50℃
20141021200330	开	开	关	46℃	50%	78	5 分钟	50℃
20141021200350	开	开	关	46℃	50%	70	4 分钟	50℃
20141021200352	开	开	关	46℃	50%	0	4 分钟	50℃
20141021200720	开	关	开	50℃	100%	0	0 分钟	50℃
20141021200820	开	关	开	50℃	100%	0	0 分钟	50℃
20141021200822	开	关	开	50℃	100%	78	0 分钟	50℃
20141021201010	开	开	关	45℃	25%	90	5 分钟	50℃
20141021201116	开	开	关	46℃	25%	80	4 分钟	50℃
20141021201118	开	开	关	46℃	25%	0	4 分钟	50℃
20141021201200	开	关	开	50℃	100%	80	0 分钟	50℃

根据用水数据，得到用水事件的属性构造说明图，如图 10-5 所示。

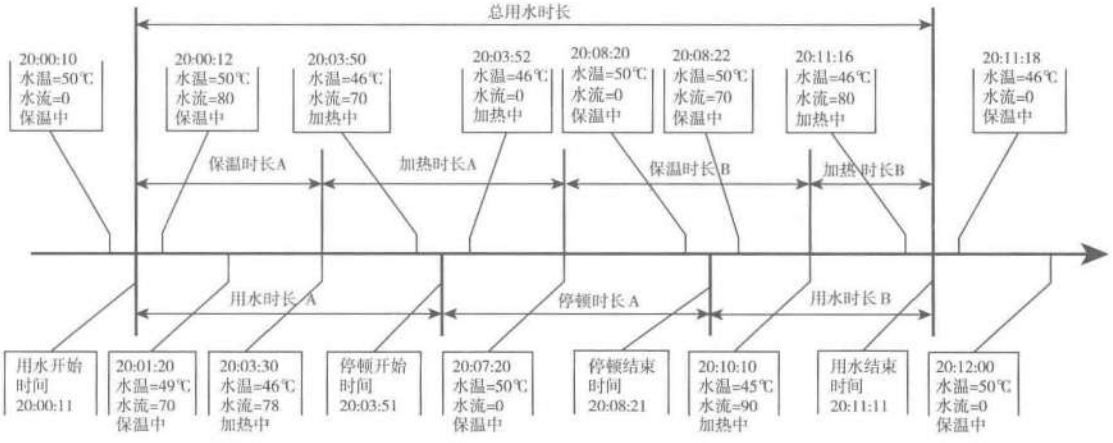


图 10-5　一次用水事件及相关属性说明

下面对 4 类指标的构建方法做详细说明。

- **时长类指标**。由图 10-5 及表 10-10 可知，在 20:00:10 热水器记录到的数据还没有用水，而在 20:00:12 热水器记录有用水行为，所以，用水开始时间在 20:00:10~20:00:12 之间。考虑到网络不稳定导致网络数据传输延时数分钟或数小时等因素，取平均值会导致很大的偏差，综合分析构建“用水开始时间”为起始数据的时间减去“发送阈值”的一半，发送阈值是指热水器传输数据的频率的大小。同理构造用水结束时间、停顿开始时间和停顿结束时间等。图 10-5 中的“用水时长 A”是“用水开始时间”到“停顿开始时间”的间隔时长，构建一次用水事件中“用水时长”为各段用水时长之和。同理构造总用水时长、停顿时长等。详细信息见表 10-11。
- **频率类指标**。统计一次用水事件中各种用水操作的频率。详细信息见表 10-12。
- **用水量化指标**。总用水量定义为：在水流量不为 0 时，一次用水事件（见表 10-10）中每条状态记录的水流量与下一条状态记录的时间间隔的乘积。平均水流量定义为：总用水量与用水时长的商。详细信息见表 10-13。
- **用水波动指标**。“水流量波动”指标定义为：当前水流的值与平均水流量差的平方乘以持续时间的总和除以总的有水流量的时间。同理构造温度波动、热水量波动和停顿时长波动等指标。详细信息见表 10-14。

表10-11 主要时长类指标构建说明

指 标	构建方法	说 明
用水开始时间	用水开始时间＝起始数据的时间－发送阈值 /2	热水事件开始发生的时间
用水结束时间	用水结束时间＝结束数据的时间＋发送阈值 /2	热水事件结束发生的时间
用水时长	一次完整用水事件中，对水流量不为 0 的数据做计算 用水时长＝每条用水数据时长的和＝（和下条数据的间隔时间 /2 ＋和上条数据的间隔时间 /2）的和	一次用水过程中有热水流出的时长
总用水时长	从划分出的用水事件，起始数据的时间到终止数据的时间间隔＋发送阈值	记录整个用水阶段的时长
用水时长 / 总用水时长	用水时长与总用水时长的比值	判断用水时长占总用水时长的比重
停顿时长	一次完整用水事件中，对水流量为 0 的数据做计算 停顿时长＝每条用水停顿数据时长的和＝（和下条数据的间隔时间 /2 ＋和上条数据的间隔时间 /2）的和	标记一次完整用水事件中的每次用水停顿的时长
总停顿时长	一次完整用水事件中的所有停顿时长之和	标记一次完整用水事件中的总停顿时长
平均停顿时长	一次完整用水事件中的所有停顿时长的平均值	标记一次完整用水事件中的停顿的平均时长

表10-12 频数类指标构建说明

指 标	构建方法	说 明
停顿次数	一次完整用水事件中关掉热水的次数之和	帮助识别洗浴及连续洗浴事件

表10-13　用水量化指标构建说明

指　标	构建方法	说　明
总用水量	总用水量＝每条有水流数据的用水量＝持续时间 × 水流大小	一次用水过程中使用的总的水量，单位为 L
平均水流量	平均水流量＝总用水量 / 有水流时间	一次用水过程中，开花洒时平均水流量大小（为热水），单位为 L/min

表10-14　用水波动指标构建说明

指　标	构建方法	说　明
水流量波动	水流量波动＝∑（（单次水流的值 – 平均水流量）^2× 持续时间）/ 总的有水流量的时间	一次用水过程中，开花洒时水流量的波动大小
停顿时长波动	停顿时长波动＝∑（（单次停顿时长 – 平均停顿时长）^2）× 持续时间）/ 总停顿时长	一次用水过程中，用水停顿时长的波动情况

（4）筛选得“候选洗浴事件”

洗浴事件的识别是建立在一次用水事件识别的基础上，也就是从已经划分好的一次用水事件中识别出哪些一次用水事件是洗浴事件。

首先，用 3 个比较宽松的条件筛选掉那些非常短暂的用水事件，将剩余的洗浴事件称为“候选洗浴事件”。这 3 个条件是“或”的关系。也就是说，只要一次完整的用水事件满足任意一个条件，就被判定为短暂用水事件，即会被筛选掉。3 个筛选条件如下。

1）一次用水事件中总用水量（纯热水）小于 y 升。

2）用水时长小于 100 秒。

3）总用水时长小于 120 秒。

下面对 y 的合理取值进行探究。洗澡的水温一般为 37 ～ 41℃。因为花洒喷头出水的温度变化也在 37 ～ 41℃，所以热水器设定温度越高，热水器水的实际温度就越高，热水器热水的使用量就越少。

经过实验分析，热水器设定温度为 50℃时，一次普通的洗浴时长为 15 分钟，总用水时长 10 分钟左右，热水的使用量为 10 ～ 15 升。

为了不影响特殊的短暂的洗浴事件，以及考虑到夏天用的热水较少，放宽范围假定热水器在设定温度为 50℃时，一次洗浴的总热水使用量为 5 升，同时取洗浴温度的均值为 39℃来计算热水器不同设定温度下的热水使用量阈值。

热水使用量模型变量符号说明，见表 10-15。

表10-15　标准热水量换算模型符号说明

洗浴用水温度	T （39℃）	设定温度	X （℃）
自来水水温	C （℃）	设定温度为 X 时的用水量	Y （升）
自来水注入量	M （升）	50 摄氏度时的用水量	V （5 升）

假定每次洗浴习惯变化不大且热水器水温恒定，则每次洗浴使用的热水的热量应该趋近于一个定值。如果热水器设定温度 X 调高使热水器水温变高，则一次洗浴使用的热水量就减少；相反，使用的热水量就增多。

假设两次洗浴事件热水和冷水混合后的花洒出水水温度恒为 T 摄氏度，总用水量不变且为 $M+V$ 升，根据热量守恒建立方程组（10-2）。

$$\begin{cases}(50-T)\times V+(C-T)\times M=0 & (1)\\ (X-T)\times Y+(C-T)\times(M+V-Y)=0 & (2)\end{cases} \tag{10-3}$$

其中（1）式是 50℃的热水 V 升与 M 升 C℃自来水混合得到 $M+V$ 升 T 摄氏度的洗浴用水的热守恒公式。（2）式是 X℃的热水 Y 升与 $M+V-Y$ 升 C℃自来水混合得到 $M+V$ 升 T℃的洗浴用水的热守恒公式。从而得出 Y 与 X、C、V 之间的关系。

$$Y=\frac{(50-C)\times \mathrm{V}}{X-C} \tag{10-4}$$

其中，V 是热水器的水恒为 50℃时洗浴时的最低用水量。根据公式（10-4）可以计算用水事件在不同实际用水温度下的标准热水使用量。其中，自来水每月平均温度取平均室温。

3. 数据清洗

本案例中存在用水数据状态记录缺失的情况，需要对缺失的数据状态记录进行添加。在热水器工作态改变或处于用水阶段时，热水器每 2 秒（发送阈值）传输一条状态记录，而划分一次完整用水事件时，需要一个开始用水的状态记录和结束用水的状态记录。但是，在划分一次完整用水事件时，发现数据中存在没有结束用水的状态记录情况，该类缺失值问题如表 10-16 所示。热水器状态发生改变，第 5 条状态记录和第 7 条状态记录的时间间隔应该为 2 秒，而表中两条记录间隔为 1 小时 27 分 28 秒。

表10-16 状态记录中的缺失值

序号	发生时间	开关机状态	加热中	保温中	实际温度	热水量	水流量	加热剩余时间	当前设置温度
1	20141019094636	关	关	关	29℃	0%	0	0 分钟	50℃
2	20141019094638	关	关	关	29℃	0%	16	0 分钟	50℃
3	20141019094640	关	关	关	29℃	0%	13	0 分钟	50℃
4	20141019094658	关	关	关	29℃	0%	0	0 分钟	50℃
5	20141019094715	关	关	关	29℃	0%	20	0 分钟	50℃
6	20141019111443	关	关	关	29℃	0%	0	0 分钟	50℃

这可能是由于网络故障等原因导致状态记录时间间隔为几十分钟甚至几小时的情况，该类问题若用均值去填充会造成用水时间为几十分钟甚至几小时的误差。对于上述特殊情况，本案例数据进行如下处理：在存在用水状态记录缺失的情况下，填充一条状态记录使水流量为 0，发生时间加 2 秒，其余属性状态不变。即在表 10-16 的第 5 条状态记录和第 7 条状态

记录之间加一条记录，即第 6 条状态记录，如表 10-17 所示。

表10-17　状态记录中缺失值的处理

序号	发生时间	开关机状态	加热中	保温中	实际温度	热水量	水流量	加热剩余时间	当前设置温度
1	20141019094636	关	关	关	29℃	0%	0	0 分钟	50℃
2	20141019094638	关	关	关	29℃	0%	16	0 分钟	50℃
3	20141019094640	关	关	关	29℃	0%	13	0 分钟	50℃
4	20141019094658	关	关	关	29℃	0%	0	0 分钟	50℃
5	20141019094715	关	关	关	29℃	0%	20	0 分钟	50℃
6	*20141019094717*	*关*	*关*	*关*	*29℃*	*0%*	*0*	*0 分钟*	*50℃*
7	20141019111443	关	关	关	29℃	0%	0	0 分钟	50℃

10.2.4　模型构建

经过数据预处理后，得到的建模样本数据如表 10-18 所示。

表10-18　部分建模样本数据示例列表

热水事件	起始数据编号	终止数据编号	开始时间	是否为洗浴（1 表示是，0 表示否）	总用水时长	总停顿时长	平均停顿时长	停顿次数	用水时长	用水 / 总时长	总用水量	平均水流量	水流量波动	停顿时长波动
1	218	344	2014-10-19 08:51:30'	0	592	304	51	6	288	0.5	13.0	2.7	0.9	650.1
2	569	965	2014-10-19 15:55:23'	1	1008	46	46	1	962	1.0	50.6	3.2	0.2	0
3	1077	1128	2014-10-19 18:21:40'	0	468	269	54	5	199	0.4	7.1	2.1	0.4	531.4
4	1973	2236	2014-10-20 16:42:41'	1	661	23	23	1	638	1.0	32.2	3.0	0.3	0
5	2320	2435	2014-10-20 18:05:28'	1	550	165	33	5	385	0.7	13.5	2.1	0.4	180.4
6	2438	2606	2014-10-20 18:25:24'	1	649	201	201	1	448	0.7	22.6	3.0	0.6	0
7	2693	2810	2014-10-20 20:00:42'	1	298	8	2	4	290	1.0	15.1	3.1	1.1	0
8	2835	3033	2014-10-20 20:15:13'	0	624	5	5	1	619	1.0	41.0	4.0	0.2	0

根据建模样本数据和用户记录的包含用水的用途、用水开始时间、用水结束时间等属性的用水日志，建立多层神经网络模型识别洗浴事件。由于洗浴事件与普通用水事件在特征上

存在不同，而且这些不同的特征在属性上被体现出来。于是，根据用户提供的用水日志，将其中洗浴事件的数据状态记录作为训练样本训练多层神经网络。然后根据训练好的网络来检验新采集到的数据，具体过程如图 10-6 所示。

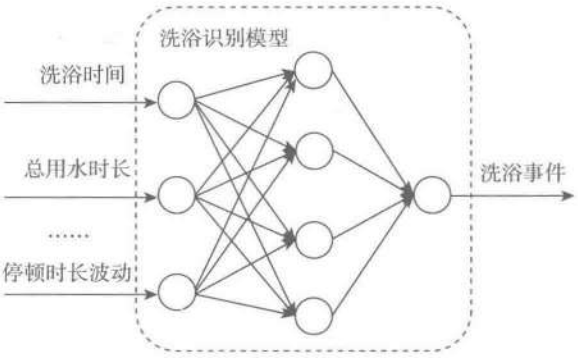


图 10-6 BP 神经模型识别洗浴事件

在训练神经网络的时候，选取了“候选洗浴事件”的 11 个属性作为网络的输入，分别为：洗浴时间点、总用水时长、总停顿时长、平均停顿时长、停顿次数、用水时长、用水时长 / 总用水时长、总用水量、平均水流量、水流量波动和停顿时长波动。训练 BP 网络时给定的输出（教师信号）为 1 与 0，其中 1 代表该次事件为洗浴事件，0 表示该次事件不是洗浴事件。其中，是否为洗浴事件根据用户提供的用水记录日志得到。

在训练神经网络时，对神经网络的参数进行了寻优，发现含二个隐层的神经网络训练效果较好，其中二个隐层的隐节点数分别为 17、10 时训练的效果较好。

使用 Python 的 Keras 库来训练神经网络，训练样本为根据用户记录的日志标记好的用水事件，详细代码见代码清单 10-3。

代码清单10-3 训练多层神经网络代码

```
#-*- coding: utf-8 -*-
#建立、训练多层神经网络，并完成模型的检验
from __future__ import print_function
import pandas as pd

inputfile1='../data/train_neural_network_data.xls' #训练数据
inputfile2='../data/test_neural_network_data.xls' #测试数据
testoutputfile = '../tmp/test_output_data.xls' #测试数据模型输出文件
data_train = pd.read_excel(inputfile1) #读入训练数据(由日志标记事件是否为洗浴)
data_test = pd.read_excel(inputfile2) #读入测试数据(由日志标记事件是否为洗浴)
y_train = data_train.iloc[:,4].as_matrix() #训练样本标签列
x_train = data_train.iloc[:,5:17].as_matrix() #训练样本特征
y_test = data_test.iloc[:,4].as_matrix() #测试样本标签列
x_test = data_test.iloc[:,5:17].as_matrix() #测试样本特征

from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation

model = Sequential() #建立模型
model.add(Dense(11, 17)) #添加输入层、隐藏层的连接
model.add(Activation('relu')) #以Relu函数为激活函数
model.add(Dense(17, 10)) #添加隐藏层、隐藏层的连接
```

```
model.add(Activation('relu')) #以Relu函数为激活函数
model.add(Dense(10, 1)) #添加隐藏层、输出层的连接
model.add(Activation('sigmoid')) #以sigmoid函数为激活函数
#编译模型，损失函数为binary_crossentropy，用adam法求解
model.compile(loss='binary_crossentropy', optimizer='adam', class_mode="binary")

model.fit(x_train, y_train, nb_epoch = 100, batch_size = 1) #训练模型
model.save_weights('../tmp/net.model') #保存模型参数

r = pd.DataFrame(model.predict_classes(x_test), columns = [u'预测结果'])
pd.concat([data_test.iloc[:,:5], r], axis = 1).to_excel(testoutputfile)
model.predict(x_test)
```

代码详见：demo/code/ neural_network.py

根据样本，得到训练好的神经网络后，就可以用来识别对应用户的洗浴事件，待检测的样本的 11 个属性作为输入，输出层输出一个值在 [-1，1] 范围内，如果该值小于 0，则该事件不是洗浴事件，如果该值大于 0，则该事件是洗浴事件。某热水器用户记录了两周的热水器用水日志，将前一周的数据作为训练数据，后一周的数据作为测试数据，代入上述模型进行测试。

10.2.5　模型检验

根据该热水器用户提供的用水日志来判断事件是否为洗浴与多层神经网络模型识别结果的比较，如表 10-19 所示，总共 21 条检测数据，准确识别了 18 条数据，模型对洗浴事件的识别准确率为 85.5%。

表10-19　用户日志判断结果与模型输出判断结果比较

热水事件	起始数据编号	终止数据编号	开始时间	根据日志判断是否为洗浴（1 表示是，-1 表示否）	神经网络判断是否为洗浴
1	73	336	'2015-01-05 9:42:41'	1	1
2	420	535	'2015-01-05 18:05:28'	1	1
3	538	706	'2015-01-05 18:25:24'	1	1
4	793	910	'2015-01-05 20:00:42'	1	1
5	935	1133	'2015-01-05 20:15:13'	1	1
6	1172	1274	'2015-01-05 20:42:41'	1	1
7	1641	1770	'2015-01-06 08:08:26'	0	0
8	2105	2280	'2015-01-06 11:31:13'	1	1
9	2290	2506	'2015-01-06 17:08:35'	1	1
10	2562	2708	'2015-01-06 17:43:48'	1	1
11	3141	3284	'2015-01-07 10:01:57'	0	1

（续）

热水事件	起始数据编号	终止数据编号	开始时间	根据日志判断是否为洗浴（1 表示是，–1 表示否）	神经网络判断是否为洗浴
12	3524	3655	'2015-01-07 13:32:43'	0	1
13	3659	3863	'2015-01-07 17:48:22'	1	1
14	3937	4125	'2015-01-07 18:26:49'	1	1
15	4145	4373	'2015-01-07 18:46:07'	1	1
16	4411	4538	'2015-01-07 19:18:08'	1	1
17	5700	5894	'2015-01-08 7:08:43'	0	1
18	5913	6178	'2015-01-08 13:23:42'	1	1
19	6238	6443	'2015-01-08 18:06:47'	1	1
20	6629	6696	'2015-01-08 20:18:58'	1	1
21	6713	6879	'2015-01-08 20:32:16'	1	1

由于训练数据为一周数据，训练样本过少，可能会造成模型训练不准确，但长期让用户记录用水日志存在一定的操作困难，在这里模型检验用了两周的用户用水日志。

10.3 上机实验

1. 实验目的

- 使用 Python 对数据进行预处理，掌握使用 Python 进行数据预处理的方法。
- 掌握数据转换，属性提取过程。

2. 实验内容

- 对采集到的热水器用户数据以 4 分钟为阈值进行用水事件划分。
- 对划分得到的用水事件提取用水事件时长、一次用水事件中开关机切换次数、一次用水事件的总用水量和平均水流量等 4 个属性。

3. 实验方法与步骤

实验一

1）打开 Python 载入 Pandas 库，使用 read_excel() 函数将“test/data/water_heater.xls”数据读入到 Python 中。water_heater.xls 文件中的数据形式如表 10-4 所示，数据为热水器用户一个月左右的用水数据，数据量为 2 万行左右。

2）利用 Pandas 的函数和方法，得到用水事件的序号、事件起始数据编号、事件终止数据编号，其中用水事件的序号为一个连续编号（1，2，3……）。根据水流量的值是否为 0，明确地确定用户是否在用热水。再根据各条数据的发生时间，如果停顿时间超过阈值 4 分

钟，则认为是 2 次用水事件。算法具体步骤可参考 10.2.3 节的数据变换中一次完整用水事件的划分模型，也可根据自己理解编写。

3）使用 to_excel() 函数将得到用水事件序号、事件起始数据编号和事件终止数据编号等，划分结果保存到 Excel 文件中。

实验二

1）打开 Python 载入 Pandas，使用 read_excel() 函数将“ test/data/water_heater.xls”数据读入到 Python 中，并将“实验一”中得到的划分结果读入到 Python 中。

2）数据转换，属性提取。用水事件时长，根据事件终止数据时间点减去事件起始数据时间点得到。再得到一次用水事件中开关机切换次数、一次用水事件的总用水量、平均水流量等属性。这些属性的提取方法见表 10-11 ～表 10-20。

3）用 to_excel() 函数将每个用水事件的基本信息与提取得到的属性保存到 Excel 文件中。

4. 思考与实验总结

1）在划分用水事件中采用的阈值为 4 分钟，而案例中有阈值寻优的模型，可用阈值寻优模型对每家热水器用户每个时间段寻找最优的阈值。

2）试着自行用循环语句（for 或者 while）实现相同的功能，对比案例提供的代码（即用内置的广播式的函数），运行效率会下降多少？

10.4　拓展思考

根据模型划分的结果，发现有时候会将两次（或多次）洗浴划分为一次洗浴，因为在实际情况中，存在着一个人洗完澡后，另一个人马上洗的情况，过渡期间的停顿间隔小于阈值。针对两次（或多次）洗浴事件被合并为一次洗浴事件的情况，需要进行优化，对连续洗浴事件进行识别，提高模型识别精确度。

本案例给出的连续洗浴识别法如下：

对每次用水事件，建立一个连续洗浴判别指标。连续洗浴判别指标初始值为 0，每当有一个属性超过设定的阈值，就给该指标加上相应的值，最后判别连续洗浴指标是否超过给定的阈值，如果超过给定的阈值，认为该次用水事件为连续洗浴事件。

选取 5 个前面章节提取得到的属性，作为判别连续洗浴事件的特征属性，这 5 个属性分别为：总用水时长、停顿次数、用水时长 / 总用水时长、总用水量和停顿时长波动。详细的说明如下。

1）总用水时长的阈值为 900 秒，如果超过 900 秒，就认为可能是连续洗浴，对于每超出的一秒，在该事件的连续洗浴判别指标上加上 0.005，详情见表 10-20。

2）停顿次数的阈值为 10 次，如果超过 10 次，就认为可能是连续洗浴，对于每超出的一次，在该事件的连续洗浴判别指标上加上 0.5，详情见表 10-20。

3）用水时长 / 总用水时长的阈值为 0.5，如果小于 0.5，就认为可能是连续洗浴，对于每

少一个单位，在该事件的连续洗浴判别指标上加上 0.2，详情见表 10-20。

4）总用水量的阈值为 30 升次，如果超过 30 升，就认为可能是连续洗浴，对于每超出的 1 升，在该事件的连续洗浴判别指标上加上 0.2，详情见表 10-20。

5）停顿时长波动的阈值为 1 000，如果超过 1 000，就认为是连续洗浴，对于每超出一个单位，在该事件的连续洗浴判别指标上加上 0.002，详情见表 10-20。

表10-20 连续洗浴事件划分模型符号说明

属性名称	符 号	阈 值	单 位	权 重
停顿次数	P	7	每超 1 次	0.5
总用水量	A	30	每超 1 升	0.2
用水时长 / 总用水时长	D	0.5	每少 1	2
总用水时长	T	900	每超 1 秒	0.005
停顿时长波动	W	1000	每超 1	0.002

根据以上信息建立优化模型，其中 S 是连续洗浴判别指标。

$$P = \begin{cases} 0.5 \times (p - 10) & p > 10 \\ 0 & p \in [0,10] \end{cases} \tag{10-5}$$

$$A = \begin{cases} 0.2 \times (a - 30) & a > 30 \\ 0 & a \in [0,30] \end{cases} \tag{10-6}$$

$$D = \begin{cases} 0.2 \times (0.5 - d) & d < 0.5 \\ 0 & d \in [0.5,1] \end{cases} \tag{10-7}$$

$$T = \begin{cases} 0.005 \times (t - 900) & t > 900 \\ 0 & t \in [0,900] \end{cases} \tag{10-8}$$

$$W = \begin{cases} 0.002 \times (t - 1000) & w > 1000 \\ 0 & w \in [0,1000] \end{cases} \tag{10-9}$$

$$S = P + A + D + T + W \tag{10-10}$$

所以，连续洗浴事件的划分模型如下。

- 当用水事件的连续洗浴判别指标 S 大于 5 时，确定为连续洗浴事件或一次洗浴事件加一次短暂用水事件，取中间停顿时间最长的停顿，划分为两次事件。
- 如果 S 不大于 5，确定为一次洗浴事件。

10.5 小结

本案例基于实时监控的智能热水器的用户使用数据，重点介绍了数据挖掘中的数据预处理的数据清洗、数据规约、数据变换等方法，以及数据预处理在实际案例中的应用，建立了热水器的洗浴事件识别的神经网络模型，并针对数据变换部分给出了 Python 上机实验。

第 11 章 Chapter 11

应用系统负载分析与磁盘容量预测

11.1 背景与挖掘目标

某大型企业为了信息化发展的需要，建设了办公自动化系统、人力资源管理系统、财务管理系统、企业信息门户系统等几大企业级应用系统。因应用系统在日常运行时，会对底层软、硬件造成负荷，显著影响应用系统性能。如图 11-1 所示，影响应用系统性能的因素包括：服务器、数据库、中间件和存储设备。任何一种资源负载过大，都可能会引起应用系统性能下降甚至瘫痪。因此需要关注服务器、数据库、中间件和存储设备的运行状态，及时了解当前应用系统的负载情况，以便提前预防，确保系统安全稳定运行。

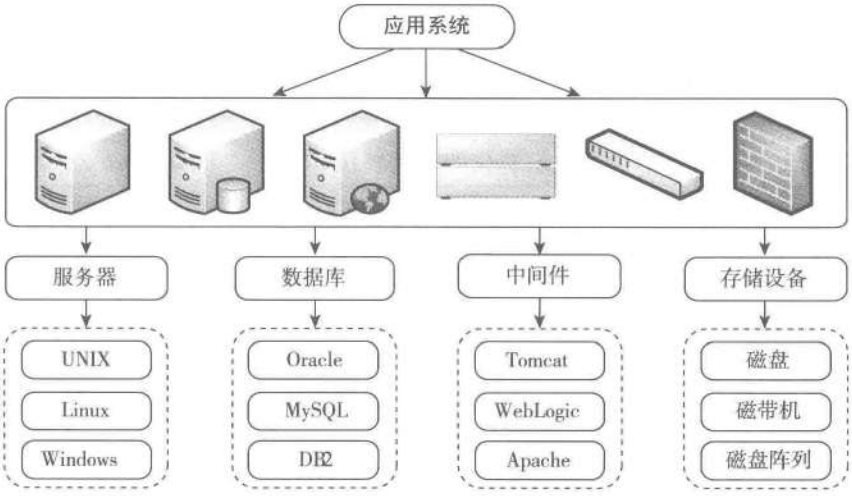


图 11-1 应用系统拓扑关系图

应用系统的负载率可以通过对一段时间内软、硬件性能的运行状况进行综合评分而获得。通过对系统的当前负载率与历史平均负载率进行比较，获得负载率的当前趋势。通过负载率以及负载趋势可对系统进行负载分析，如图 11-2 所示。当出现应用系统的负载高或者负载趋势大的情况，代表系统目前处于高危工作环境中。如果系统管理员不及时进行相应的处理，系统很容易出现故障，从而导致用户无法访问系统，严重影响企业的利益。本章重点分析存储设备中磁盘容量预测，通过对磁盘容量进行预测，可预测磁盘未来的负载情况，避免应用系统因出现存储容量耗尽的情况而导致应用系统负载率过高，最终引发系统故障。

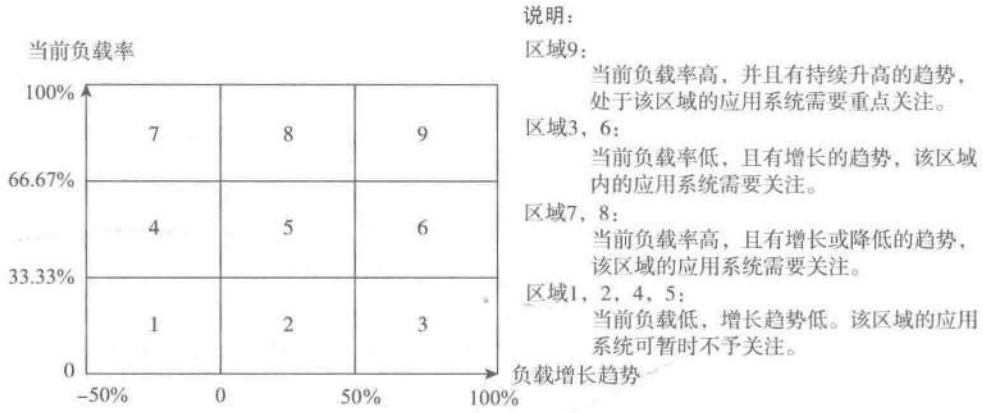


图 11-2 应用系统负载分析

目前监控采集的性能数据主要包含 CPU 使用信息、内存使用信息和磁盘使用信息等，性能说明表见表 11-1。通过分析磁盘容量相关数据（见表 11-2），预测应用系统服务器磁盘空间是否满足系统健康运行的要求。根据这些数据可实现以下目标。

- 针对历史磁盘数据，采用时间序列分析方法，预测应用系统服务器磁盘已使用空间大小。
- 根据用户需求设置不同的预警等级，将预测值与容量值进行比较，对其结果进行预警判断，为系统管理员提供定制化的预警提示。

表11-1 性能说明表

属性名称	属性说明	属性名称	属性说明
SYS_NAME	资产所在的系统名称	ENTITY	具体的属性
NAME	资产名称	VALUE	采集到的值
TARGET_ID	属性的标识号 183 表示磁盘容量大小 184 表示磁盘已使用大小	COLLECTTIME	采集的时间
DESCRIPTION	针对属性标识的说明		

表11-2　磁盘原始数据集

SYS_NAME	NAME	TARGET_ID	DESCRIPTION	ENTITY	VALUE	COLLECTTIME
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 270 787. 33	2014/10/1
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	80 262 592.65	2014/10/1
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324	2014/10/1
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/1
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 328 899. 02	2014/10/2
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	83200151. 65	2014/10/2
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324	2014/10/2
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/2
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 327 553.5	2014/10/3
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	83 208 320	2014/10/3
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324	2014/10/3
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/3
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 288 672.21	2014/10/4
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	83 099 271.65	2014/10/4
财务管理系统	CWXT_DB	183	磁盘容量	D:\	52 323 324	2014/10/4
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/4
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 190 978.41	2014/10/5
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	82 765 171.65	2014/10/5
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324	2014/10/5
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/5
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 187 614.43	2014/10/6
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	82 522 895	2014/10/6
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324	2014/10/6
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328	2014/10/6

数据详见：demo/data/discdata.xls

11.2　分析方法与过程

应用系统出现故障通常不是突然瘫痪造成的（除非对服务器直接断电），而是一个渐变的过程[19]。例如，系统长时间运行，数据会持续写入存储，存储空间逐渐变少，最终磁盘被写满而导致系统故障。由此可知，在不考虑人为因素的影响时，存储空间随时间变化存在很强的关联性，且历史数据对未来的发展存在一定的影响，故本案例可采用时间序列分析法对磁

盘已使用空间进行预测分析。

采用时间序列分析法分析磁盘性能数据，预测未来的磁盘使用空间的情况，其挖掘建模的总体流程如图 11-3 所示。

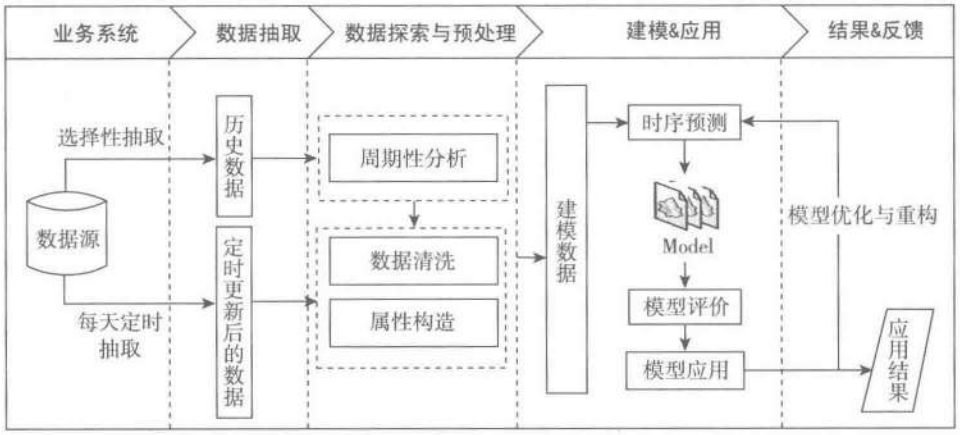


图 11-3 建模流程图

应用系统容量预测建模过程主要包含以下步骤。

1）从数据源中选择性抽取历史数据与每天定时抽取数据。

2）对抽取的数据进行周期性分析以及数据清洗、数据变换等操作后，形成建模数据。

3）采用时间序列分析法对建模数据进行模型的构建，利用模型预测服务器磁盘已使用情况。

4）应用模型预测服务器磁盘将要使用情况，通过预测到的磁盘使用大小与磁盘容量大小按照定制化标准进行判断，将结果反馈给系统管理员，提示管理员需要注意磁盘的使用情况。

11.2.1 数据抽取

磁盘使用情况的数据都存放在性能数据中，而监控采集的性能数据中存在大量的其他属性数据。为了抽取出磁盘数据，以属性的标识号（TARGET_ID）与采集指标的时间（COLLECTTIME）为条件，对性能数据进行抽取。本案例抽取 2014-10-01 至 2014-11-16 财务管理系统中某一台数据库服务器的磁盘的相关数据。

11.2.2 数据探索分析

由于本例是采用时序分析法进行建模的，为了建模的需要，需要探索数据的平稳性。通过时序图可以初步发现数据的平稳性。针对服务器磁盘已使用大小，以天为单位，进行周期性分析，其时序图如图 11-4 和图 11-5 所示。

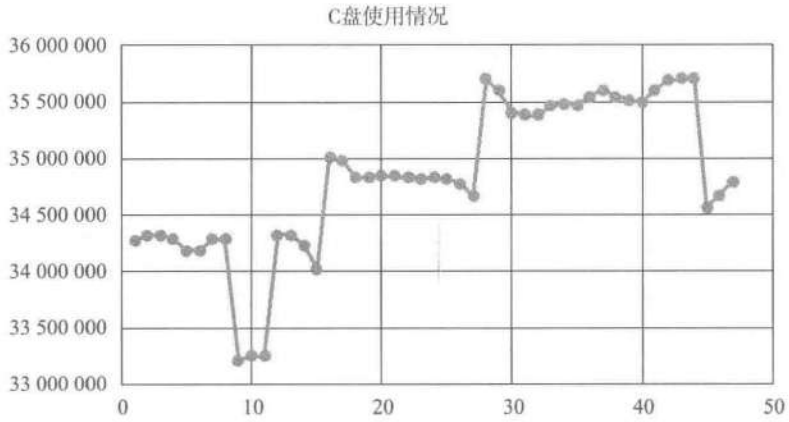


图 11-4　C 盘已使用空间的时序图

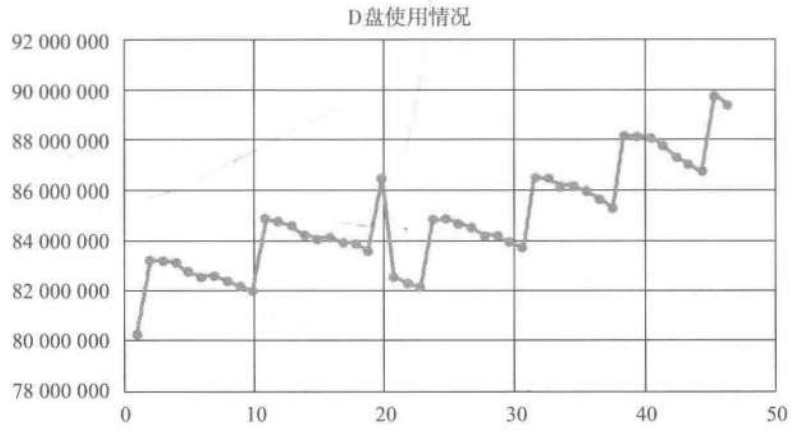


图 11-5　D 盘已使用空间的时序图

由图 11-4 和图 11-5 可以得知，磁盘的使用情况都不具备有周期性，它们表现出缓慢性增长，呈现趋势性。因此，可以初步确认数据是非平稳的。

11.2.3　数据预处理

1. 数据清洗

在实际的业务中，监控系统会每天定时对磁盘的信息进行收集，但是一般情况下，磁盘容量属性都是一个定值（不考虑中途扩容的情况），因此磁盘原始数据中会存在磁盘容量的重复数据。在数据清洗过程中，剔除磁盘容量的重复数据，并且将所有服务器的磁盘容量作为一个固定值，方便模型预警，磁盘容量表见表 11-3。

表11-3 磁盘容量表

SYS_NAME	NAME	TARGET_ID	DESCRIPTION	ENTITY	VALUE
财务管理系统	CWXT_DB	183	磁盘容量	C:\	52 323 324
财务管理系统	CWXT_DB	183	磁盘容量	D:\	157 283 328

2. 属性构造

经过数据清洗后的磁盘数据见表 11-4，其中磁盘相关属性以记录的形式存在数据中，其单位为 KB。因为每台服务器的磁盘信息可以通过表中 NAME、TARGET_ID、ENTITY 三个属性进行区分，且每台服务器的上述三个属性值是不变的，所以可以将三个属性的值合并，构造新的属性，如表 11-5 所示。(本质上是进行行列互换操作)

表11-4 原始性能表

SYS_NAME	NAME	TARGET_ID	DESCRIPTION	ENTITY	VALUE	COLLECTTIME
财务管理系统	CWXT_DB	184	磁盘已使用大小	C:\	34 270 787.33	2014/10/1
财务管理系统	CWXT_DB	184	磁盘已使用大小	D:\	80 262 592.65	2014/10/1

表11-5 属性变换后的性能表

SYS_NAME	CWXT_DB:184:C:\	CWXT_DB:184:D:\	COLLECTTIME
财务管理系统	34 270 787.33	80 262 592,65	2014/10/1

属性变换的 Python 代码如代码清单 11-1 所示。

代码清单11-1 属性变换代码

```
#-*- coding: utf-8 -*-
#属性变换
import pandas as pd

#参数初始化
discfile = '../data/discdata.xls' #磁盘原始数据
transformeddata = '../tmp/discdata_processed.xls' #变换后的数据

data = pd.read_excel(discfile)
data = data[data['TARGET_ID'] == 184].copy() #只保留TARGET_ID为184的数据

data_group = data.groupby('COLLECTTIME') #以时间分组

def attr_trans(x): #定义属性变换函数
  result = pd.Series(index = ['SYS_NAME', 'CWXT_DB:184:C:\\', 'CWXT_DB:184:D:\\',
      'COLLECTTIME'])
  result['SYS_NAME'] = x['SYS_NAME'].iloc[0]
  result['COLLECTTIME'] = x['COLLECTTIME'].iloc[0]
  result['CWXT_DB:184:C:\\'] = x['VALUE'].iloc[0]
  result['CWXT_DB:184:D:\\'] = x['VALUE'].iloc[1]
```

```
    return result

data_processed = data_group.apply(attr_trans) #逐组处理
data_processed.to_excel(transformeddata, index = False)
```

代码详见：demo/code/attribute_transform.py

11.2.4　模型构建

为了方便对模型进行评价，将经过数据预处理后的建模数据划分两部分。1）建模样本数据；2）模型验证数据。选取建模数据的最后 5 条记录作为验证数据，其他数据作为建模样本数据。

1. 容量预测模型

本章容量预测模型的建模流程如图 11-6 所示。

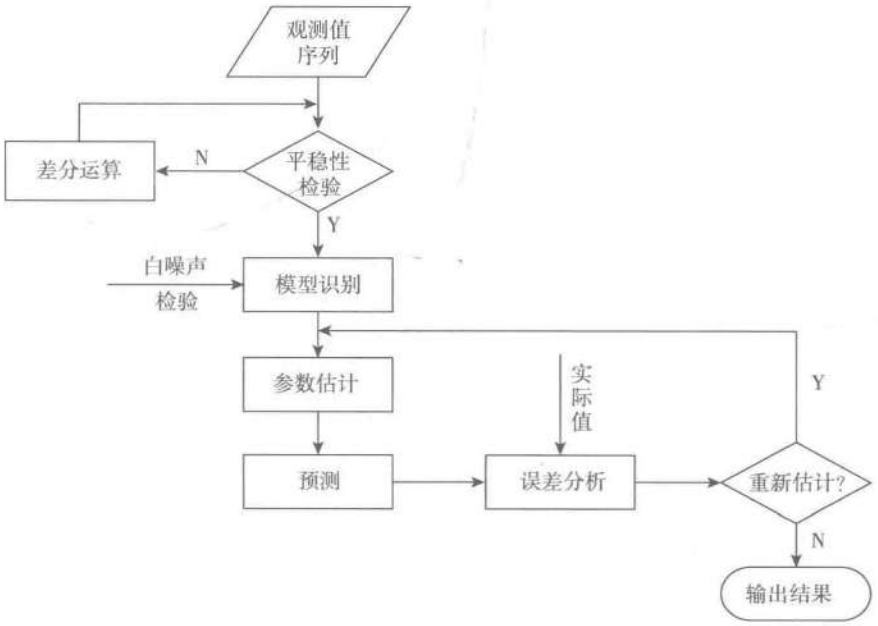


图 11-6　容量预测建模图

首先需要对观测值序列进行平稳性检验，如果不平稳，则对其进行差分处理直到差分后的数据平稳。在数据平稳后，对其进行白噪声检验。如果没有通过白噪声检验，就进行模型识别，识别其模型属于 AR、MA 和 ARMA 中的哪一种模型。并且通过 BIC 信息准则对模型进行定阶，确定 ARIMA 模型的 p，q 参数。在模型识别后需进行模型检验，检测模型残差序列是否为白噪声序列。如果模型没有通过检测，需要对其进行重新识别。对已通过检验的模型采用极大似然估计方法进行模型参数估计。最后，应用模型进行预测，将实际值与预测值进行误差分析。如果误差比较小（误差阈值需通过业务分析进行设定），表明模型拟合效果较

好，则模型可以结束。反之需要重新估计参数。

模型构建的过程中需要用到以下方法。

1）**平稳性检验：**为了确定原始数据序列中没有随机趋势或确定趋势，需要对数据进行平稳性检验，否则将会产生“伪回归”的现象。本章采用单位根检验（ADF）的方法或者时序图的方法进行平稳性检验，其检验的结果见表 11-6，时序图的方法见 11.2.2 小节数据探索分析。

表11-6 平稳性检验结果

数据序列名称	平 稳 性	对应的 p 值	n 阶差分后平稳
D 盘使用大小	非平稳	0.8921	1

平稳性检验的 Python 代码如代码清单 11-2 所示。

代码清单11-2 平稳性检验代码

```
#-*- coding: utf-8 -*-
#平稳性检验
import pandas as pd

#参数初始化
discfile = '../data/discdata_processed.xls'

data = pd.read_excel(discfile)
data = data.iloc[: len(data)-5] #不使用最后5个数据

#平稳性检测
from statsmodels.tsa.stattools import adfuller as ADF
diff = 0
adf = ADF(data['CWXT_DB:184:D:\\'])
while adf[1] >= 0.05: # adf[1]为p值，p值小于0.05认为是平稳的。
  diff = diff + 1
  adf = ADF(data['CWXT_DB:184:D:\\'].diff(diff).dropna())

print(u'原始序列经过%s阶差分后归于平稳，p值为%s' %(diff, adf[1]))
```

代码详见：demo/code/stationarity_test.py

2）**白噪声检验：**为了验证序列中有用的信息是否已被提取完毕，需要对序列进行白噪声检验。如果序列检验为白噪声序列，就说明序列中有用的信息已经被提取完毕了，剩下的全是随机扰动，无法进行预测和使用。本章采用 LB 统计量的方法进行白噪声检验，其结果见表 11-7。

表11-7 白噪声检验结果

数据序列名称	是否白噪声	对应的 p 值
D 盘	非白噪声	$9.9585*10^{-6}$
D 盘一阶差分后	白噪声	0.1143

白噪声检验的 Python 代码如代码清单 11-3 所示。

代码清单11-3　白噪声检验代码

```
#-*- coding: utf-8 -*-
#白噪声检验
import pandas as pd

#参数初始化
discfile = '../data/discdata_processed.xls'

data = pd.read_excel(discfile)
data = data.iloc[: len(data)-5] #不使用最后5个数据

#白噪声检测
from statsmodels.stats.diagnostic import acorr_ljungbox

[[lb], [p]] = acorr_ljungbox(data['CWXT_DB:184:D:\\'], lags = 1)
if p < 0.05:
  print(u'原始序列为非白噪声序列，对应的p值为：%s' %p)
else:
  print(u'原始该序列为白噪声序列，对应的p值为：%s' %p)

[[lb], [p]] = acorr_ljungbox(data['CWXT_DB:184:D:\\'].diff().dropna(), lags = 1)
if p < 0.05:
  print(u'一阶差分序列为非白噪声序列，对应的p值为：%s' %p)
else:
  print(u'一阶差分该序列为白噪声序列，对应的p值为：%s' %p)
```

代码详见：demo/code/whitenoise_test.py

3）**模型识别**：采用极大似然比方法进行模型的参数估计，估计各个参数的值。然后针对各个不同模型，采用 BIC 信息准则对模型进行定阶，确定 p，q 参数，从而选择最优模型。根据此方法选择的模型结果见表 11-8。

表11-8　模型结果

数据序列	模型类型	最小 BIC 值
D 盘使用大小	ARIMA(0,1,1)	1300.46

模型识别代码见代码清单 11-4。

代码清单11-4　模型识别代码

```
#-*- coding: utf-8 -*-
#确定最佳p、d、q值
import pandas as pd

#参数初始化
discfile = '../data/discdata_processed.xls'
```

```
data = pd.read_excel(discfile, index_col = 'COLLECTTIME')
data = data.iloc[: len(data)-5] #不使用最后5个数据
xdata = data['CWXT_DB:184:D:\\']

from statsmodels.tsa.arima_model import ARIMA

#定阶
pmax = int(len(xdata)/10) #一般阶数不超过length/10
qmax = int(len(xdata)/10) #一般阶数不超过length/10
bic_matrix = [] #bic矩阵
for p in range(pmax+1):
  tmp = []
  for q in range(qmax+1):
    try: #存在部分报错，所以用try来跳过报错
      tmp.append(ARIMA(xdata, (p,1,q)).fit().bic)
    except:
      tmp.append(None)
  bic_matrix.append(tmp)

bic_matrix = pd.DataFrame(bic_matrix) #从中可以找出最小值

p,q = bic_matrix.stack().idxmin() #先用stack展平，然后用idxmin找出最小值位置
print(u'BIC最小的p值和q值为：%s、%s' %(p,q))
```

代码详见：demo/code/find_optimal_pq.py

4）**模型检验**：模型确定后，检验其残差序列是否为白噪声。如果不是白噪声，说明残差中还存在有用的信息，需要修改模型或者进一步提取。本案例所确定的 ARIMA(0, 1, 1) 模型成功地通过了检验。模型检验代码如代码清单 11-5 所示。

代码清单11-5　模型检验代码

```
#-*- coding: utf-8 -*-
#模型检验
import pandas as pd

#参数初始化
discfile = '../data/discdata_processed.xls'
lagnum = 12 #残差延迟个数

data = pd.read_excel(discfile, index_col = 'COLLECTTIME')
data = data.iloc[: len(data)-5] #不使用最后5个数据
xdata = data['CWXT_DB:184:D:\\']

from statsmodels.tsa.arima_model import ARIMA #建立ARIMA(0,1,1)模型

arima = ARIMA(xdata, (0, 1, 1)).fit() #建立并训练模型
xdata_pred = arima.predict(typ = 'levels') #预测
pred_error = (xdata_pred - xdata).dropna() #计算残差

from statsmodels.stats.diagnostic import acorr_ljungbox #白噪声检验
```

```
lb, p= acorr_ljungbox(pred_error, lags = lagnum)
h = (p < 0.05).sum() #p值小于0.05，认为是非白噪声。
if h > 0:
  print(u'模型ARIMA(0,1,1)不符合白噪声检验')
else:
  print(u'模型ARIMA(0,1,1)符合白噪声检验')
```

代码详见：demo/code/arima_model_check.py

5）**模型预测**：应用通过检验的模型进行预测，获取未来 5 天的预测值，并且与实际值作比较，也就是我们在建模型的时候所忽略的最后 5 个数据。为了方便比较，将单位换算成 GB，其结果见表 11-9。

表11-9　预测结果

日　期	预 测 值	实 际 值
2014-11-12	88.034 302 6	87.249 335 55
2014-11-13	88.217 008 31	86.986 142 2
2014-11-14	88.399 714 02	86.678 24
2014-11-15	88.582 419 72	89.766 6
2014-11-16	88.765 125 43	89.377 527 25

2. 模型评价

为了评价时序预测模型效果的好坏，本章采用 3 个衡量模型预测精度的统计量指标：平均绝对误差、均方根误差和平均绝对百分误差。这 3 个指标从不同侧面反映了算法的预测精度[20]。

选择建模数据的后 5 条记录作为实际值，将预测值与实际值进行误差分析，模型的各个评价指标值见表 11-10。

表11-10　模型评价表

平均绝对误差	均方根误差	平均绝对百分误差
1.106 8	1.1723	1.2610

模型评价的 Python 代码如代码清单 11-6 所示。

代码清单11-6　模型评价代码

```
#-*- coding: utf-8 -*-
#计算预测误差
import pandas as pd

#参数初始化
file = '../data/predictdata.xls'
data = pd.read_excel(file)
```

```
#计算误差
abs_ = (data[u'预测值'] - data[u'实际值']).abs()
mae_ = abs_.mean() # mae
rmse_ = ((abs_**2).mean())**0.5 # rmse
mape_ = (abs_/data[u'实际值']).mean() # mape

print(u'平均绝对误差为：%0.4f，\n均方根误差为：%0.4f，\n平均绝对百分误差为：%0.6f。' %(mae_,
    rmse_, mape_))
```

代码详见：demo/code/cal_errors.py

结合实际业务分析，将误差阈值设定为 1.5。表 11-11 中实际值与预测值之间的误差全都小于误差阈值。因此，模型的预测效果在实际业务可接受的范围内，可以采用此模型进行预测。

3. 模型应用

在上述模型构建完成后，就可以对模型进行应用，实现对应用系统容量的预测，其模型应用如下过程。

1）从系统中每日定时抽取服务器磁盘数据。

2）对定时抽取的数据进行数据清洗、数据变换预处理操作。

3）将预处理后的定时数据存放到模型的初始数据中，获得模型的输入数据，调用模型对服务器磁盘已使用空间进行预测，预测后 5 天的磁盘已使用空间大小。

4）将预测值与磁盘的总容量进行比较，获得预测的磁盘使用率。如果某一天预测的使用率达到业务设置的预警级别，就会以预警的方式提醒系统管理员。

模型应用的预警流程图如图 11-7 所示。

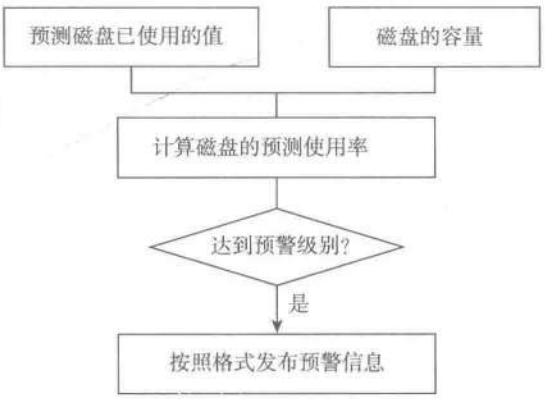


图 11-7 预警流程图

其中，预警等级的设定需要结合实际应用，根据业务的应用一般设置的阈值见表 11-11，也可以根据管理员要求进行相应的调整，调整使用率的阈值即可。如果预测值达到预警等级以上，可以发布预警信息，其示例见表 11-12，提示管理员注意，需要清理磁盘或者准备扩容，保证应用系统的健康运行。

表11-11 阈值设置表

预测已使用空间率	预 警 等 级
85%	Ⅰ
90%	Ⅱ
95%	Ⅲ

表11-12　预警信息格式

属性名称	预警时间	信　息	预警等级
D:	2014-11-12	该服务器磁盘 D 盘使用率预计 2014-11-12 将达到 85% 以上	I

因为模型采用历史数据进行建模，随着时间的变化，每天会定时地将新增数据加入初始建模数据中。在正常的情况下，模型需要重新调整。但考虑到建模的复杂性高，且磁盘的已使用大小每天的变化量相对很小，对于整个模型的预测影响较小。因此，结合实际业务情况，每半个月对模型进行一次调整。

11.3　上机实验

1. 实验目的

- 了解时间序列算法的用法以及利用时间序列算法构建预测模型的流程。
- 掌握 Python 实现时间序列算法的检验及预测的过程，以及模型的误差分析。

2. 实验内容

通过服务器的历史磁盘数据，根据时间序列算法模型的流程，预测未来磁盘的使用情况。为了方便对模型进行误差分析，将服务器的磁盘数据划分为模型输入数据与模型验证数据。采用时间序列算法对模型输入数据进行模型拟合、检验与预测。依据误差公式，计算预测值与验证数据之间的误差，分析其是否属于业务接受的范围内。

- 采用 Pandas 读取数据文件，按照划分规则将数据划分为两个部分，并将其保存。
- 调用 StatsModels 内置函数，编写代码实现本例模型构建的流程。对模型输入数据进行平稳性检验和差分，记录差分阶数。采用 BIC 准则确定模型的参数，依据各个参数构建时序模型，并对模型进行相关的检验。
- 采用通过检验的模型进行预测，比较预测值与验证数据的大小，计算其误差。利用误差公式，编写代码并分析误差是否处于业务接受的范围内。

3. 实验方法与步骤

1）打开 Python 并引入 Pandas 库，使用 read_excel() 函数将数据文件读入 Python 工作空间中，选择要进行时序预测的磁盘数据，截取最后 5 条数据为验证数据，其他数据为模型输入数据。

2）确定 ARIMA 模型的 D 参数，即差分阶数。使用 adfuller() 函数确定输入数据是否平稳化，如果不平稳，则进行使用 diff() 函数进行差分，记录差分的阶数；否则 D 值为 0，并直接进行下一步。

3）确定 ARIMA 模型的 p、q 参数。p、q 参数的取值范围为 [0，N/10]，选择不同的 p、q 值，计算输入数据的 BIC 值。当 BIC 值取最小值时，p、q 值即是所求。

4）使用 ARIMA 函数以及前面得到的 p、D、q 构建 ARIMA 模型，使用 summary() 函数确定模型的其他参数，使用 acorr_ljungbox() 函数计算模型残差白噪声。检验其是否通过白噪声检验，如果不通过则返回步骤 3）去掉上一步的 p、q 组合重新进行计算；如果通过则进行下一步。

5）使用 forecast() 函数进行时序预测，并把实际值和预测值进行对比，计算其误差。

4. 思考与实验总结

1）用其他的方法进行平稳性检验，如游程检验、自相关系数分析等。

2）采用其他的方法进行模型定阶，确定 p 与 q 的参数值。

11.4 拓展思考

监控不仅能够获取软、硬件的性能数据，同时也能检测到软硬件的日志事件，并通过告警的方式提示用户。在监控的告警表中存在很多类别的告警，其中服务器类的告警包含：CPU 告警、内存告警、磁盘告警；数据库类的告警包含：日志告警、表空间告警；网络类型的告警包含：Ping 告警、Telnet 告警，以及应用系统类别的告警。一旦应用系统发生故障，会影响整个公司的利润。因此，管理员在维护系统的过程中，必须特别关注应用系统类别的告警。但是在监控收集性能以及事件的过程中，有时会存在信息收集有误的情况，因此各类型告警会出现误告。（注：应用系统发生误告时系统实际处于正常阶段）

根据每天的各种类型的告警数，通过相关性判断哪些类型告警与应用系统真正故障有关，其原始数据见表 11-13。通过相关类型的告警，预测明后两天的告警数。针对历史的告警数与应用系统的关系，判断系统未来是否发生故障。首先通过时序算法预测未来相关类型的告警数，然后采用分类预测算法对预测值进行判断，判断系统未来是否发生故障。（针对原始数据可以选择一部分数据进行时序预测。）

表11-13 系统告警原始数据

日　　期	CPU 告警	内存告警	磁盘告警	日志类告警	表空间告警	Ping 告警	Telnet 告警	故障类别
2013/01/01	4	1	0	0	0	2	4	0
2013/01/02	0	2	0	0	0	0	0	0
2013/01/03	1	0	0	0	0	0	2	0
2013/01/04	0	0	0	0	0	1	2	1
2013/01/05	1	0	0	2	0	4	0	1
2013/01/06	1	1	0	0	0	3	4	0
2013/01/07	1	0	0	0	0	0	0	0
2013/01/08	1	2	0	0	0	0	2	0

（续）

日　期	CPU 告警	内存告警	磁盘告警	日志类告警	表空间告警	Ping 告警	Telnet 告警	故障类别
2013/01/09	0	0	0	0	0	1	0	0
2013/01/10	3	0	0	2	0	4	0	1
2013/01/11	0	1	0	0	0	3	4	0
2013/01/12	3	1	0	0	0	0	0	0
2013/01/13	0	0	0	0	0	0	2	0
2013/01/14	5	1	0	0	0	1	0	0
2013/01/15	0	0	0	0	0	4	0	0
2013/01/16	1	0	0	1	0	2	4	1
2013/01/17	0	2	0	0	0	0	0	0
2013/01/18	2	2	0	0	0	0	2	0
2013/01/19	0	0	0	0	0	0	0	0
2013/01/20	0	1	0	0	0	4	0	0
2013/01/21	0	0	0	0	0	3	3	0
2013/01/22	1	0	0	0	0	0	0	0
2013/01/23	0	0	0	0	0	0	2	0
2013/01/24	0	3	0	0	0	0	0	0

数据详见：拓展思考 / 拓展思考样本数据 .xls

11.5　小结

本章结合应用系统磁盘容量预测的案例，重点介绍了数据挖掘算法中时间序列分析法在实际案例中的应用，并详细地描述了系统磁盘容量预测数据挖掘以及时间序列分析建模的整个过程。同时对相应的算法以及整个数据挖掘流程给出了 Python 上机实验。

Chapter 12 第 12 章

电子商务网站用户行为分析及服务推荐

12.1 背景与挖掘目标

随着互联网和信息技术的快速发展，电子商务、网上服务与交易等网络业务越来越普及，大量的信息聚集起来，形成海量信息。用户想要从海量信息中快速准确地寻找到自己感兴趣的信息已经变得越来越困难，在电子商务领域这点显得更加突出。因此，信息过载的问题已经成为互联网技术中的一个重要难题。为了解决这个问题，搜索引擎诞生了，例如Google、百度等。搜索引擎在一定程度上缓解了信息过载问题，用户通过输入关键词，搜索引擎就会返回给用户与输入的关键词相关的信息。但是搜索引擎无法解决用户的很多其他需求，例如用户想找到准确描述自己需求的关键词时，搜索引擎就无能为力了。

与搜索引擎不同，推荐系统并不需要用户提供明确的需求，而是通过分析用户的历史行为，从而主动向用户推荐能够满足他们兴趣和需求的信息。因此，对于用户而言，推荐系统和搜索引擎是两个互补的工具。搜索引擎满足有明确目标用户的需求，而推荐系统能够帮助用户发现其感兴趣的内容。因此，在电子商务领域中推荐技术可以起到以下作用：1）帮助用户发现其感兴趣的物品，节省用户时间、提升用户体验。2）提高用户对电子商务网站的忠诚度，如果推荐系统能够准确地发现用户的兴趣点，并将合适的资源推荐给用户，用户就会对该电子商务网站产生依赖，从而建立稳定的企业忠实顾客群。

本例主要的研究对象是北京某家法律网站，它是一家电子商务类的大型法律资讯网站，致力于为用户提供丰富的法律信息与专业咨询服务，并为律师与律师事务所提供卓有成效的互联网整合营销解决方案。随着其网站访问量增大，数据信息量也在大幅增长。用户在面对大量信息时无法及时从中获得自己需要的信息，对信息的使用效率越来越低。这种浏览大量无关信息的过程，使用户需要花费大量的时间才能找到自己需要的信息，从而使得用户不断流失，给企

业造成巨大的损失。为了能够更好地满足用户需求，依据其网站海量的数据，研究用户的兴趣偏好，分析用户的需求和行为，发现用户的兴趣点，从而引导用户发现自己的信息需求，将长尾网页准确地推荐给所需用户，帮助用户发现他们感兴趣但很难发现的网页信息。为用户提供个性化的服务，并且建立网站与用户之间的密切关系，让用户对推荐系统产生依赖，从而建立稳定的企业忠实顾客群，实现客户链式反应增值，提高消费者满意度。通过提高服务效率帮助消费者节约交易成本等，制定有针对性的营销战略方针，促进企业长期稳定高速发展。

目前网站上已经存在部分推荐，例如，当访问主页时，可以在婚姻栏目发现如下热点推荐，如图 12-1 所示。当访问具体的知识页面时，可以在页面的右边以及下面发现一些热点推荐和基于内容的关键字推荐，如图 12-2 所示。

图 12-1　主页热点推荐

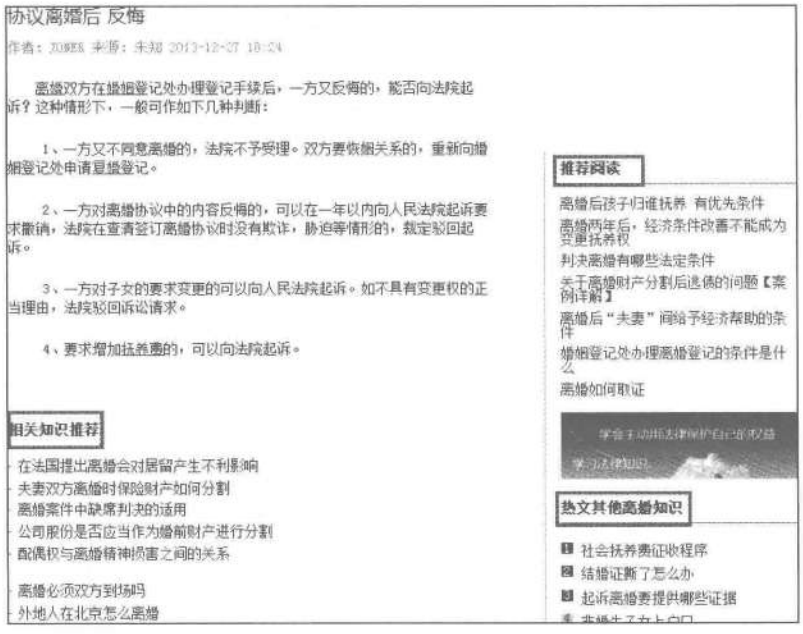


图 12-2　婚姻知识目前的推荐

当用户访问网站页面时，系统会记录用户访问网站的日志，其访问的数据记录见表 12-2，其中记录了用户 IP（已做数据脱敏处理）、用户访问的时间、访问内容等多项属性的记录，并针对其中的各个属性进行说明，见表 12-1。

表12-1 访问记录属性表

属性名称	属性说明	属性名称	属性说明
realIP	真实 ip	fullURLId	网址类型
realAreacode	地区编号	hostname	源地址名
userAgent	浏览器代理	pageTitle	网页标题
userOS	用户浏览器类型	pageTitleCategoryId	标题类型 ID
userID	用户 ID	pageTitleCategoryName	标题类型名称
clientID	客户端 ID	pageTitleKw	标题类型关键字
timestamp	时间戳	fullReferrer	入口源
timestamp_format	标准化时间	fullReferrerURL	入口网址
pagePath	路径	organicKeyword	搜索关键字
ymd	年月日	source	搜索源
fullURL	网址		

依据所提供的原始数据，试着分析如下目标。

- 按地域研究用户访问时间、访问内容和访问次数等分析主题，深入了解用户对访问网站的行为和目的及关心的内容。
- 借助大量的用户的访问记录，发现用户的访问行为习惯，对不同需求的用户进行相关的服务页面的推荐。

12.2 分析方法与过程

本案例的目标是对用户进行推荐，即以一定的方式将用户与物品（本书指网页）之间建立联系[21]。为了更好地帮助用户从海量的数据中快速发现感兴趣的网页，在目前相对单一的推荐系统上进行补充，采用协同过滤算法进行推荐，其推荐原理如图 12-3 所示。

由于用户访问网站的数据记录很多，如果对数据不进行分类处理，对所有记录直接采用推荐系统进行推荐，这样会存在以下问题。①数据量太大意味着物品数与用户数很多，在模型构建用户与物品的稀疏矩阵时，出现设备内存空间不够的情况，并且模型计算需要消耗大量的时间。②用户区别很大，不同的用户关注的信息不一样，因此，即使能够得到推荐结果，其推荐效果也会不好。为了避免出现上述问题，需要进行分类处理与分析，如图 12-4 所示。正常的情况下，需要对用户的兴趣爱好以及需求进行分类。因为在用户访问记录中，没有记录用户访问网页时间的长短，因此不容易判断用户兴趣爱好。因此，本文根据用户浏览

表12-2　用户访问记录表

realIP	realAreacode	userAgent	userOS	userID	clientID	timestamp	timestamp format	pagePath	ymd	fullURL	fullURLId	hostname	pageTitle	pageTitle	pageTitle
1531222030	140100	UCWEB/2.0	Other	499670012.1	499670012.1	1428041479371	2015/4/3 14:11	/ask/ques	20150403	http://www.lawtime.cn/ask/question_8399551.html	101003	www.lawt	做作住房公	69	房产买卖纠
1531222030	140100	UCWEB/2.0	Other	499670012.1	499670012.1	1428041479536	2015/4/3 14:11	/ask/ques	20150403	http://www.lawtime.cn/ask/question_8399551.html	101003	www.lawt	做作住房公	69	房产买卖纠
1706656375	140100	Mozilla/5.	Windows 7	1259341818	1259341818	1429353422107	2015/4/18 18:37	/ask/ques	20150418	http://www.lawtime.cn/ask/question_10937991.html	101003	www.lawt	做资金监管	37	立案侦查
4223238775	140100	Mozilla/5.	Windows 7	908370090.1	908370090.1	1426579834667	2015/3/17 16:10	/ask/ques	20150317	http://www.lawtime.cn/ask/question_421092.html	101003	www.lawt	做庄赌博公	26	定罪量刑
1110054106	140100	Mozilla/5.	Windows XP	2068832749	2068832749	1423635850415	2015/2/11 14:24	/ask/ques	20150211	http://www.lawtime.cn/ask/question_6925984.html	101003	www.lawt	做重伤的仍	68	工伤赔偿纠
1046706190	140100	Mozilla/5.	Windows 7	847612256.1	847612256.1	1427852615867	2015/4/1 9:43	/ask/exp/	20150401	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawt	做钟点工要	62	劳动合同纠
465868558	140100	Mozilla/5.	Windows XP	1610868312	1610868312	1428394886464	2015/4/7 16:21	/ask/exp/	20150407	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawt	做钟点工要	62	劳动合同纠
1571227319	140100	Mozilla/5.	Windows XP	371696939.1	371696939.1	1429849254956	2015/4/24 12:20	/ask/exp/	20150424	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawt	做钟点工要	62	劳动合同纠
2092450417	140100	Mozilla/5.	Windows 7	1632334036	1632334036	1427170907797	2015/3/24 12:21	/ask/ques	20150324	http://www.lawtime.cn/ask/question_5598369.html	101003	www.lawt	做直销产品	51	违约赔偿
1121454201	140100	Mozilla/4.	Windows XP	2136917696	2136917696	1428039745607	2015/4/3 13:42	/ask/ques	20150403	http://www.lawtime.cn/ask/question_3432948.html	101003	www.lawt	做侄子的法	17	儿童监护
2561650547	140100	Mozilla/5.	Mac OS X	1316725305	1316725305	1425978540658	2015/3/10 17:09	/info/yil	20150310	http://www.lawtime.cn/info/yiliao/zrsb/2010072143	107001	www.lawt	做整形手术	64	医疗事故赔
409120636	140100	Mozilla/5.	Windows XP	362380012.1	362380012.1	1429463755830	2015/4/20 1:15	/ask/ques	20150420	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawt	做杂志,使	26	定罪量刑
1699823729	140100	Mozilla/5.	Windows 7	38549427.14	38549427.14	1429871000544	2015/4/24 18:23	/ask/ques	20150424	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawt	做杂志,使	26	定罪量刑
1257332336	140100	Mozilla/5.	Windows 7	1242996761	1242996761	1430292457522	2015/4/29 15:27	/ask/ques	20150429	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawt	做杂志,使	26	定罪量刑
2586839161	140100	Mozilla/5.	Windows 7	1283840943	1283840943	1423489880092	2015/2/9 21:51	/ask/ques	20150209	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawt	做有限责任	41	股权纠纷
2828223345	140100	Mozilla/5.	Mac OS X	980744139.1	980744139.1	1425095196930	2015/2/28 11:46	/ask/ques	20150228	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawt	做有限责任	41	股权纠纷
1018329870	140100	Mozilla/5.	Windows 7	967084001.1	967084001.1	1428378107184	2015/4/7 11:41	/ask/ques	20150407	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawt	做有限责任	41	股权纠纷
2119376756	140100	Mozilla/5.	Windows 7	1599121760	1599121760	1428486243962	2015/4/8 17:44	/ask/ques	20150408	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawt	做有限责任	41	股权纠纷
2249626126	140100	Mozilla/5.	Windows 7	1796985316	1796985316	1423798297363	2015/2/13 11:31	/ask/ques	20150213	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawt	做有限公司	41	股权纠纷
3020128887	140100	Mozilla/5.	Windows 7	1188934890	1188934890	1429697248802	2015/4/22 18:07	/ask/ques	20150422	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawt	做有限公司	41	股权纠纷
3423224433	140100	Mozilla/5.	Windows 7	1150849692	1150849692	1430291567285	2015/4/29 15:12	/ask/ques	20150429	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawt	做有限公司	41	股权纠纷
3423224433	140100	Mozilla/5.	Windows 7	1150849692	1150849692	1430291685261	2015/4/29 15:14	/ask/ques	20150429	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawt	做有限公司	41	股权纠纷
1859491598	140100	Mozilla/5.	Windows 7	1102973681	1102973681	1422725918555	2015/2/1 1:38	/ask/ques	20150201	http://www.lawtime.cn/ask/question_914636.html	101003	www.lawt	做游戏外挂	26	定罪量刑
1242119863	140100	Mozilla/5.	Windows 7	984584705.1	984584705.1	1423536816616	2015/2/10 10:53	/ask/ques	20150210	http://www.lawtime.cn/ask/question_3402035.html	101003	www.lawt	做引产犯法	26	定罪量刑
3609131066	140100	Mozilla/4.	Windows XP	1221287319	1221287319	1429685946200	2015/4/22 14:59	/ask/ques	20150422	http://www.lawtime.cn/ask/question_6548781.html	101003	www.lawt	做银行黑户	78	信用卡恶意
2731637774	140100	Mozilla/5.	Windows XP	1204438324	1204438324	1427360738669	2015/3/26 17:05	/ask/ques	20150326	http://www.lawtime.cn/ask/question_7658765.html	101003	www.lawt	做银行黑户	78	信用卡恶意
2731637774	140100	Mozilla/5.	Windows XP	1204438324	1204438324	1427369024680	2015/3/26 19:23	/ask/ques	20150326	http://www.lawtime.cn/ask/question_7658765.html	101003	www.lawt	做银行黑户	78	信用卡恶意
2601561724	140100	Mozilla/5.	Other	118385063.1	118385063.1	1423150087176	2015/2/5 23:28	/ask/ques	20150205	http://www.lawtime.cn/ask/question_1152117.html	101003	www.lawt	做银行贷款	52	金融债务
3787742832	140100	Mozilla/5.	Windows 7	1342347607	1342347607	1425099001584	2015/2/28 12:50	/ask/ques	20150228	http://www.lawtime.cn/ask/question_10382308.html	101003	www.lawt	做银保工作	26	定罪量刑
1296711793	140100	Mozilla/5.	Windows 7	1546761287	1546761287	1422889805462	2015/2/2 23:10	/ask/ques	20150202	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
2059693169	140100	Mozilla/5.	Windows XP	202229259.1	202229259.1	1423645371761	2015/2/11 17:02	/ask/ques	20150211	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
364932977	140100	Mozilla/5.	Windows 8.	1911420797	1911420797	1424879011165	2015/2/25 23:43	/ask/ques	20150225	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
698003068	140100	Mozilla/5.	Windows 7	601074264.1	601074264.1	1425913507611	2015/3/9 23:05	/ask/ques	20150309	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
460808305	140100	Mozilla/5.	Windows 7	812125454.1	812125454.1	1426514986691	2015/3/16 22:09	/ask/ques	20150316	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
3080340593	140100	Mozilla/4.	Windows XP	919277708.1	919277708.1	1427280973102	2015/3/25 18:56	/ask/ques	20150325	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
221596127	140100	Mozilla/5.	Windows 7	372903090.1	372903090.1	1428398837055	2015/4/7 17:27	/ask/ques	20150407	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
705385592	140100	Mozilla/5.	Windows 7	690991433.1	690991433.1	1428475106146	2015/4/8 14:38	/ask/ques	20150408	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
3827394762	140100	Mozilla/5.	Windows XP	730261919.1	730261919.1	1428567244656	2015/4/9 16:14	/ask/ques	20150409	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawt	做遗产继承	21	医患纠纷
1011128334	140100	Mozilla/5.	Android	1963560652	1963560652	1423496417446	2015/2/9 23:40	/ask/ques	20150209	http://www.lawtime.cn/ask/question_100745.html	101003	www.lawt	做医疗事故	31	故意伤害
2228186993	140100	Mozilla/5.	Windows 7	1328923830	1328923830	1426219851133	2015/3/13 12:10	/ask/ques	20150313	http://www.lawtime.cn/ask/question_100745.html	101003	www.lawt	做医疗事故	31	故意伤害

数据详见：demo/data/7law.sql

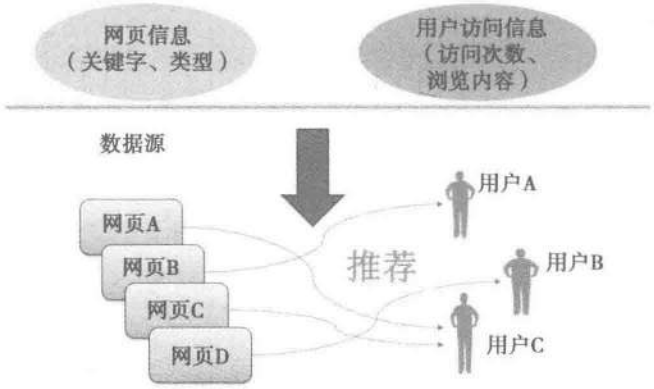


图 12-3 推荐系统原理图

的网页信息进行分类处理，主要采用以下方法处理：以用户浏览网页的类型进行分类，然后对每个类型中的内容进行推荐。

采用上述的分析方法与思路，结合本例的原始数据以及分析目标，可获得整个分析的流程图，如图 12-5 所示。其分析过程主要包含以下内容。

- 从系统中获取用户访问网站的原始记录。
- 对数据进行多维度分析，包括用户访问内容，流失用户分析以及用户分类等分析。
- 对数据进行预处理，包含数据去重、数据变换和数据分类等处理过程。

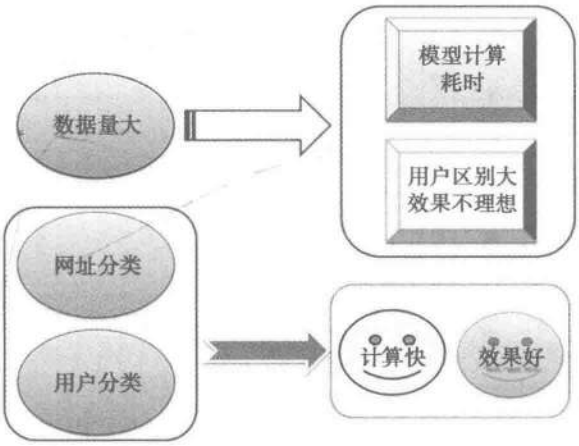


图 12-4 数据处理分析图

- 以用户访问 html 后缀的网页为关键条件，对数据进行处理。
- 对比多种推荐算法进行推荐，通过模型评价，得到比较好的智能推荐模型。通过模型对样本数据进行预测，获得推荐结果。

12.2.1 数据抽取

因为本例是以协同过滤算法为主导，其他的推荐算法为辅，而协同过滤算法的特性就是通过历史数据找出相似的用户或者网页。因此，在数据抽取的过程中，尽可能选择大量的数据，这样就能降低推荐结果的随机性，提高推荐结果的准确性，能更好地发掘长尾网页中用户感兴趣的网页。

以用户的访问时间为条件，选取 3 个月内（2015-02-01 ～ 2015-04-29）用户的访问数据作

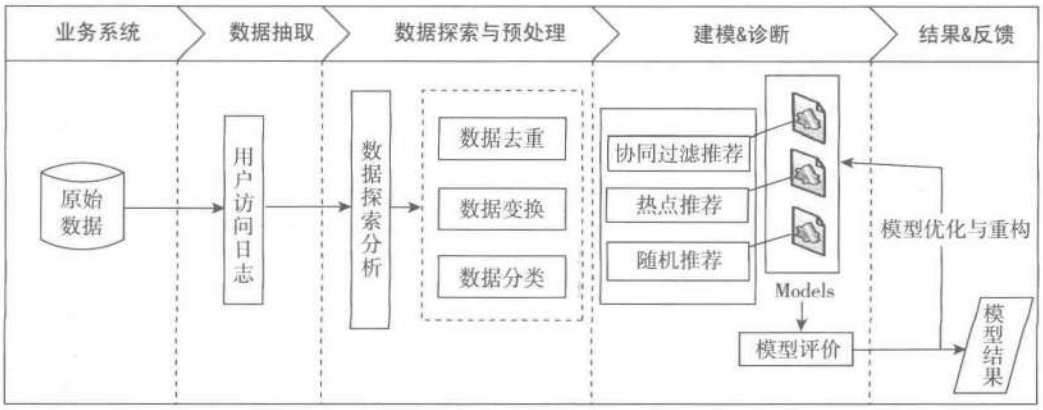


图 12-5　智能推荐系统整理流程图

为原始数据集。每个地区的用户访问习惯以及兴趣爱好存在差异性，本例抽取广州地区的用户访问数据进行分析，其数据量总计有 837 450 条记录，其中包括用户号、访问时间、来源网站、访问页面、页面标题、来源网页、标签、网页类别和关键词等属性。

虽然 837 450 条记录对于当今科学的“大数据”的概念而言，并不是特别大的数据量，但是这个数据量对于配置比较低的电脑（尤其是笔记本电脑）还是颇有压力的。因此，本章的处理过程，真正地、初步地体现了用 Python 处理大数据的味道。

本章的处理过程是：建立数据库→导入数据→搭建 Python 的数据库操作环境→对数据进行分析→建立模型。其中，用到的开源数据库为 MariaDB 10.0.17（网站 https://mariadb.org/en/ 可下载并自行安装，是 MySQL 的一个分支）。安装数据库后导入本章的数据原始文件 7law.sql，就成功地配置好了数据库平台。

而在 Python 中，Pandas 库本身可以利用 read_sql() 函数来读取数据库，但是它依赖于 SQLAlchemy 库，而 SQLAlchemy 又依赖于 PyMySQL，所以需要先安装 SQLAlchemy 再安装 SQLAlchemy，这样就可以用 Pandas 对数据库中的数据进行快速而便捷的分析工作了。安装的方法参考本书的第 2 章，这里不再赘述。

安装完成后，可以通过 Python 连接到数据库。为了方便处理数据，我们利用了 Pandas。但要注意，Pandas 在读取数据（不管是之前的 csv、Excel 或者现在的 sql），都是将全部数据读入内存中，因此在数据量较大时是难以实现的。幸运的是，Pandas 也提供了 chunksize 参数，可以让我们分块读取大数据文件。代码如代码清单 12-1 所示。

代码清单12-1　Python访问数据库

```
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine('mysql+pymysql://root:123456@127.0.0.1:3306/test?charset=
    utf8')
sql = pd.read_sql('all_gzdata', engine, chunksize = 10000)
```

```
'''
用create_engine建立连接，连接地址的意思依次为“数据库格式（mysql）+程序名（pymysql）+账号密
    码@地址端口/数据库名（test）”，最后指定编码为utf8；
all_gzdata是表名，engine是连接数据的引擎，chunksize指定每次读取1万条记录。这时候sql是一个容
    器，未真正读取数据。
'''
```

代码详见：demo/code/sql_value_counts.py

12.2.2 数据探索分析

对原始数据中的网页类型、点击次数和网页排名等各个维度进行分布分析，获得其内在的规律。并通过验证数据，解释其出现的结果可能的原因。

1. 网页类型分析

作为第一步，我们针对原始数据中用户点击的网页类型进行统计，网页类型是指“网址类型”中的前 3 位数字（它本身有 6/7 位数字）。前面已经提到过，此处处理的要义在于“分块进行”，必要时可以使用多线程甚至分布式计算。所以，代码清单 12-2 所给出的例子，已经展示了处理大数据的要义所在。后面的各项统计均按照类似的方法进行，不再赘述。

代码清单12-2 Python访问数据库并进行分块统计（接代码清单12-1）

```
counts = [ i['fullURLId'].value_counts() for i in sql] #逐块统计
counts = pd.concat(counts).groupby(level=0).sum() #合并统计结果，把相同的统计项合并（即
    按index分组并求和）
counts = counts.reset_index() #重新设置index，将原来的index作为counts的一列。
counts.columns = ['index', 'num'] #重新设置列名，主要是第二列，默认为0
counts['type'] = counts['index'].str.extract('(\d{3})') #提取前三个数字作为类别id
counts_ = counts[['type', 'num']].groupby('type').sum() #按类别合并
counts_.sort('num', ascending = False) #降序排列
```

代码详见：demo/code/ sql_value_counts.py

结果见表 12-3，从中发现点击与咨询相关（网页类型为 101，http://www.****.com/ask/）的记录占了 49.16%，其次是其他的类型（网页类型为 199）占比 24% 左右，然后是知识相关（网页类型为 107，http://www.****.com/info/）占比 22% 左右。

表12-3 网页类型统计

记 录 数	百 分 比	网 页 类 型
411 665	49.157	101
201 426	24.0523	199
182 900	21.8401	107
18 430	2.2007	301
17 357	2.0726	102
3957	0.4725	106
1715	0.2048	103

因此，可以得到用户点击的页面类型的排行榜为：咨询相关、知识相关、其他方面的网页、法规（类型为 301）、律师相关（类型为 102）。可以初步得出相对于长篇的知识，用户更加偏向于查看咨询或者进行咨询。进一步对咨询类别内部进行统计分析，其结果见表 12-4。其中浏览咨询内容页（101003）记录是最多，其次是咨询列表页（101002）和咨询首页（101001）。结合上述初步结论，可以得出用户都喜欢通过浏览问题的方式找到自己需要的信息，而不是以提问的方式或者查看长篇的知识的方式得到所需信息。

表12-4　咨询类别内部统计

记　录　数	百　分　比	101 开头类型
396 612	96.3434	101003
7 776	1.8889	101002
5 603	1.3611	101001
1 674	0.4067	其他

统计分析知识类型内部的点击情况，因知识类型中只有一种类型（107001），所以利用网址对其进行分类，获得知识内容页（http://www.****.com/info/*/ 数字 .html，其中数字部分可能带有下划线 _）以及知识首页（http://www.****.com/info/*/）和知识列表页（http://www.****.com/info/*.html，是除了知识内容页外的 html 页面）的分布情况，其结果见表 12-5。

表12-5　知识类型内部统计

记　录　数	百　分　比	107 类型
164 243	89.80	知识内容页
17 843	9.75	知识首页
814	0.45	知识列表页

所用到的技巧主要是正则表达式，基本的统计代码如代码清单 12-3 所示。

代码清单12-3　Python访问数据库并进行分块统计（接代码清单12-2）

```
#统计107类别的情况
def count107(i): #自定义统计函数
  j = i[['fullURL']][i['fullURLId'].str.contains('107')].copy() #找出类别包含107的网址
  j['type'] = None #添加空列
  j['type'][j['fullURL'].str.contains('info/.+?/')] = u'知识首页'
  j['type'][j['fullURL'].str.contains('info/.+?/.+?')] = u'知识列表页'
  j['type'][j['fullURL'].str.contains('/\d+?_*\d+?\.html')] = u'知识内容页'
  return j['type'].value_counts()

counts2 = [count107(i) for i in sql] #逐块统计
counts2 = pd.concat(counts2).groupby(level=0).sum() #合并统计结果
```

分析其他（199）页面的情况，其中网址中带有“？”的占了32%左右，其他咨询相关与法规专题占比达到43%，地区和律师占比26%左右。在网页的分类中，有律师、地区、咨询相关的网页分类，为何这些还会存在其他类别中？进行数据查看后，发现大部分是以下网址的形式存在。

- http://www.****.com/guangzhou/p2lawfirm 地区律师事务所。
- http://www.****.com/guangzhou 地区网址。
- http://www.****.com/ask/ask.php。
- http://www.****.com/ask/midques_10549897.html 中间类型网页。
- http://www.****.com/ask/exp/4317.html 咨询经验。
- http://www.****.com/ask/online/138.html 在线咨询页。

带有标记的3类网址本应该有相应的分类，但是由于分类规则的匹配问题，没有相应的匹配。带有lawfirm关键字对应的是律师事务所，带有ask/exp、ask/online关键字对应的是咨询经验和在线咨询页。所以，在处理数据过程中将其进行清楚分类，便于后续数据分析。

综上分析的3种情况，可以发现大部分用户浏览的网页的情况为：咨询内容页、知识内容页、法规专题页、咨询经验（在线咨询页）。因此，在后续的分析中，选取其中占比最多的两类（咨询内容页和知识内容页）进行模型分析。

上述在其他类别中，发现网址中存在带“?”的情况，对其进行统计，一共有65 492条记录，占所有记录7.8%，统计分析此情况，其结果见表12-6。可以从表中得出网址中带有“?”的情况不仅仅出现在其他类别中，同时也会出现在咨询内容页和知识内容页中。但其他类型中（1999001）占了98.8%，因此需要进一步分析其类型内部的规律。

表12-6 带问号字符网址类型统计表

总 数	网页ID	百 分 比
64 718	1999001	98.8182
356	301001	0.5436
346	107001	0.5283
47	101003	0.0718
25	102002	0.0382

表12-7 其他类型统计表

1999001总数	网页标题	百 分 比
49 894	快车－律师助手	77.0945
6166	免费发布咨询	9.5275
5220	咨询发布成功	8.0658
1943	快搜	3.0023
1495	其他类型	2.3102

通过统计分析结果见表 12-7，在 1999001 类型中，标题为快车 - 律师助手的这类信息占比 77%，通过对业务了解，这是律师的一个登录页面。标题为咨询发布成功页面是自动跳转的页面。其他剩下的带有“？”的页面记录，占其记录的 15% 左右，占所有记录的 1% 左右。其他类型中的大部分为“ http://www.****.com/ask/question_9152354.html?&from=androidqq”，这种类型的网页是被分享过的，可以对其进行处理，截取“？”前面的网址，还原其类型。因为快搜和免费发布咨询网址中，类型很混杂，不能直接采用“？”进行截取，无法还原其原来类型，且整个数据集中占比很小，因此在处理数据环节可以对这部分数据进行删除。网址中不包含主网址、不包含关键字的网址有 101 条记录，类似的网址为：“ http://www.baidu.com/link?url=O7iBD2KmoJdkHWTZHagDXrxfBFM0AwLmpid12j2d_aejNfq6bwSBeqT-1Ov2jWOFMpIt5XUpXGmNiLDlGg0rMCwstskhB5ftAYtO2_voEnu”。

在查看数据的过程中，发现存在这样一部分的用户，他们没有单击具体的网页（以 .html 后缀结尾），他们单击的大部分是目录网页，这样的用户可以称为“瞎逛”，总计有 7 668 条记录。分析其中的网页类型，统计结果见表 12-8。可以从中看出，小部分是与知识、咨询相关，大部分是与地区、律师和事务所相关的。这部分用户有可能是找律师服务的，或者是瞎逛的。

表12-8　“瞎逛”用户点击行为分析

总　数	网页 ID	总　数	网页 ID
3689	199	846	107
1764	102	241	101
1079	106	49	301

从上述网址类型分布分析中，可以发现一些与分析目标无关数据的规则。①咨询发布成功页面。②中间类型网页（带有 midques_ 关键字）。③网址中带有“？”类型，无法还原其本身类型的快搜页面与发布咨询网页。④重复数据（同一时间同一用户，访问相同网页）。⑤其他类别的数据（主网址不包含关键字）。⑥无点击 .html 页面行为的用户记录。⑦律师的行为记录（通过快车 - 律师助手判断）。记录这些规则，有利于在数据清洗阶段对数据进行清洗操作。

上述过程就是对网址类型进行统计得到的分析结果，针对网页的点击次数也进行下述分析。

2. 点击次数分析

统计分析原始数据用户浏览网页次数（以“真实 IP”区分）的情况，其结果见表 12-9，可以从表中发现浏览一次的用户占所有用户总量的 58% 左右，大部分用户浏览的次数在 2 ～ 7 次，用户浏览的平均次数是 3 次。

表12-9 用户点击次数统计表

点击次数	用 户 数	用户百分比	记录百分比
1	132 119	57.41	15.78
2	44 175	19.19	10.55
3	17 573	7.64	6.30
4	10 156	4.41	4.85
5	5952	2.59	3.55
6	4132	1.80	2.96
7	2632	1.14	2.20
7次以上	13 410	5.82	53.81

详细代码见代码清单 12-4。

代码清单12-4 Python访问数据库并进行分块统计（接代码清单12-3）

```
#统计点击次数
c = [i['realIP'].value_counts() for i in sql] #分块统计各个IP的出现次数
count3 = pd.concat(c).groupby(level = 0).sum() #合并统计结果，level=0表示按index分组
count3 = pd.DataFrame(count3) #Series转为DataFrame
count3[1] = 1 #添加一列，全为1
count3.groupby(0).sum() #统计各个“不同的点击次数”分别出现的次数
```

从上表中可以看出大约 80% 的用户（不超过 3 次）只提供了大约 30% 的浏览量（几乎满足二八定律）。在数据中，点击次数最大值为 42 790 次，对其进行分析，发现是律师的浏览信息（通过律师助手进行判断）。表 12-10 是对浏览次数达到 7 次以上的情况进行的分析，可以从中看出大部分用户浏览 8 到 100 次。

表12-10 浏览7以上的用户分析表

点击次数	用 户 数
8~100	12 952
101~1000	439
1000 以上	19

针对浏览次数为一次的用户进行分析，其结果如表 12-11 所示。其中，问题咨询页占比 78%，知识页占比 15%，而且这些记录基本上全是通过搜索引擎进入的。由此可以猜测两种可能：1）用户为流失用户，在问题咨询与知识页面上没有找到相关的需要。2）用户找到其需要的信息，因此直接退出。综合这些情况，可以将这些点击一次的用户行为定义为网页的跳出率。为了降低网页的跳出率，需要对这些网页进行针对用户的个性化推荐，帮助用户发现其感兴趣或者需要的网页。

表12-11　浏览一次的用户行为分析

网页类型 ID	个　数	百 分 比
101003	102 560	77.63
107001	19 443	14.72
1999001	9381	7.10
301001	515	0.39
其他	202	0.15

针对点击一次的用户浏览的网页进行统计分析，其结果见表 12-12。可以看出排名靠前的都是知识与咨询页面，因此可以猜测大量用户的关注都在知识或咨询方面上。

表12-12　点击一次用户浏览网页统计

网　址	点 击 数
http://www.****.com/info/shuifa/slb/2012111978933.html	1013
http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html	501
http://www.****.com/ask/question_925675.html	423
http://www.****.com/ask/exp/13655.html	301
http://www.****.com/ask/exp/8495.html	241
http://www.****.com/ask/exp/13445.html	199
http://www.****.com/ask/exp/17357.html	171

3. 网页排名

由分析目标可知，个性化推荐主要针对以 html 为后缀的网页（与物品的概念类似）。从原始数据中统计以 html 为后缀的网页的点击率，其点击率排名的结果见表 12-13。从表中可以看出，点击次数排名前 20 名中，“法规专题”占了大部分，其次是“知识”，然后是“咨询”。但是，从前面分析的结果中可知，原始数据中与咨询主题相关的记录占了大部分。在其 html 后缀的网页排名中，“专题与知识”的占了大部分。通过对业务了解，专题是属于知识大类里的一个小类。在统计以 html 为后缀的网页点击排名，出现这种现象的原因见表 12-14。其中，知识页面相对咨询的页面要少很多，当大量的用户在浏览咨询页面时，呈现一种比较分散的浏览次数，即其各个页面点击率不高，但是其总的浏览量高于知识。所以造成网页排名中咨询方面的排名比较低。

表12-13　点击率排名表

网　址	点 击 数
http://www.****.com/faguizt/23.html	6503
http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html	4938

（续）

网　址	点 击 数
http://www.****.com/faguizt/9.html	4562
http://www.****.com/info/shuifa/slb/2012111978933.html	4495
http://www.****.com/faguizt/11.html	3976
http://www.****.com/info/hunyin/lhlawlhxy/20110707137693_2.html	3305
http://www.****.com/faguizt/43.html	3251
http://www.****.com/faguizt/15.html	2718
http://www.****.com/faguizt/117.html	2670
http://www.****.com/faguizt/41.html	2455
http://www.****.com/info/shuifa/slb/2012111978933_2.html	2161
http://www.****.com/faguizt/131.html	1561
http://www.****.com/ask/browse_a1401.html	1305
http://www.****.com/faguizt/21.html	1210
http://www.****.com/ask/exp/13655.html	1060
http://www.****.com/faguizt/39.html	1059
http://www.****.com/faguizt/79.html	916
http://www.****.com/ask/question_925675.html	879
http://www.****.com/faguizt/7.html	845
http://www.****.com/ask/exp/8495.html	726

表12-14　类型点击数

html 网页类型	总点击次数	用　户　数	平均点击率
知识类（包含专题和知识）	231 702	65 483	3.54
咨询类	437 132	185 478	2.37

从原始 html 的点击率排行榜中可以发现如下情况，排行榜中存在这样两种类似的网址“http://www.****.com/info/hunyin/lhlawlhxy/20110707137693_2.html”和“http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html”。通过访问其网址，发现两者属于同一网页，但由于系统在记录用户的访问网址的信息时会将其记录在数据中。因此，在用户访问网址的数据中存在这些翻页的情况，针对这些翻页的网页进行统计，结果见表 12-15。

表12-15　翻页网页统计表

网　页	次　数	比　例
http://www.****.com/info/gongsi/slbgzcdj/201312312876742.html	243	
http://www.****.com/info/gongsi/slbgzcdj/201312312876742_2.html	190	0.782

（续）

网　页	次　数	比　例
http://www.****.com/info/hetong/ldht/201311152872128.html	197	0.468
http://www.****.com/info/hetong/ldht/201311152872128_2.html	421	
http://www.****.com/info/hetong/ldht/201311152872128_3.html	293	0.696
http://www.****.com/info/hetong/ldht/201311152872128_4.html	180	0.614
http://www.****.com/info/hunyin/hunyinfagui/20110813143541.html	299	
http://www.****.com/info/hunyin/hunyinfagui/20110813143541_2.html	234	0.783
http://www.****.com/info/hunyin/hunyinfagui/20110813143541_3.html	175	0.748

通过了解业务，同一网页中登录次数最多都是从外部搜索引擎直接搜索到的网页。对其中的浏览翻页的情况进行分析，平均大概 60% ～ 80% 的人会选择看下一页，基本每一页都会丢失 20% ～ 40% 的点击率。同时，对知识类型网页进行检查，发现页面上并无全页显示功能，但是知识页面中大部分都存在翻页的情况。这样就造成了大量的用户基本选择浏览 2 ～ 5 页后，很少会选择浏览完全部的内容。因此，用户就会直接就放弃此次的搜索，从而增加网站的跳出率，降低了客户的满意度，不利于企业的长期稳定发展。

12.2.3　数据预处理

本案例在原始数据的探索分析的基础上，发现与分析目标无关或模型需要处理的数据，针对此类数据进行处理。其中涉及的数据处理方式有：数据清洗、数据集成和数据变换。通过这几类的处理方式，将原始数据处理成模型需要的输入数据，其数据处理流程图如图 12-6 所示。

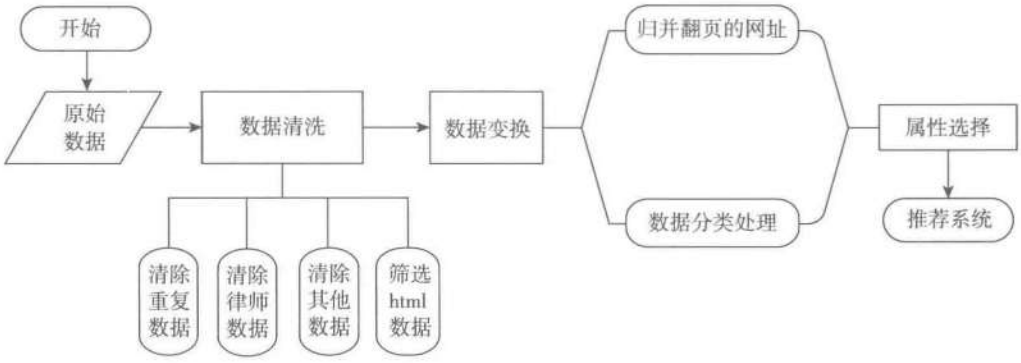


图 12-6　数据处理流程图

1. 数据清洗

从探索分析的过程中发现与分析目标无关的数据，归纳总结其数据满足如下规则：中间

页面的网址、咨询发布成功页面、律师登录助手的页面等。将其整理成删除数据的规则，其清洗的结果见表 12-16。从表中可以发现，律师用户信息占了所有记录中的 22% 左右。其他类型的数据，占比很小，大概 5% 左右。

表12-16 规则清洗表

删除数据规则	删除数据记录	原始数据记录	百 分 比
中间类型网页（带 midques_ 关键字）	2036	837 450	0.24
（快车 - 律师助手）律师的浏览信息	185 437	837 450	22.14
咨询发布成功	4819	837 450	0.58
主网址不包含关键字	92	837 450	0.01
快搜与免费发布咨询的记录	9982	837 450	1.19
其他类别带有？的记录	571	837 450	0.07
无 .html 点击行为的用户记录	7668	837 450	0.92
重复记录	25 598	837 450	3.06

经过上述数据清洗后的记录中仍然存在大量的目录网页（可理解为用户浏览信息的路径），在进入推荐系统时，这些信息的作用不大，反而会影响推荐的结果，因此需要进一步筛选以 html 为后缀的网页。根据分析目标以及探索结果可知，咨询与知识是其主要业务来源，故需筛选咨询与知识相关的记录，将此部分数据作为模型分析需要的数据。

针对数据进行清洗操作，Python 实现的代码例子（部分）如代码清单 12-5 所示。

代码清单12-5 Python访问MariaDB(MySQL)数据库进行清洗操作

```
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine('mysql+pymysql://root:123456@127.0.0.1:3306/test?charset=utf8')
sql = pd.read_sql('all_gzdata', engine, chunksize = 10000)

for i in sql:
  d = i[['realIP', 'fullURL']] #只要网址列
  d = d[d['fullURL'].str.contains('\.html')].copy() #只要含有.html的网址
  #保存到数据库的cleaned_gzdata表中（如果表不存在则自动创建）
  d.to_sql('cleaned_gzdata', engine, index = False, if_exists = 'append')
```

代码详见：demo/code/ sql_clean_save.py

2. 数据变换

由于在用户访问知识的过程中，存在翻页的情况，不同的网址属于同一类型的网页，见表 12-17。数据处理过程中需要对这类网址进行处理，最简单的处理方法是将翻页的网址删掉。但是，用户访问页面是通过搜索引擎进入网站的，所以其入口网页不一定是其原始类别的首页，采用删除的方法会损失大量的有用数据，在进入推荐系统时，会影响推荐结果。因

此，针对这些网页需要还原其原始类别，处理方式为首先识别翻页的网址，然后对翻页的网址进行还原，最后针对每个用户访问的页面进行去重操作，其操作结果见表 12-18。

表12-17　用户翻页网址表

用户 ID	时　间	访 问 网 页
978851598	2015-02-11 15:24:25	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246.html
978851598	2015-02-11 15:25:46	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246_2.html
978851598	2015-02-11 15:25:52	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246_4.html
978851598	2015-02-11 15:26:00	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246_5.html
978851598	2015-02-11 15:26:10	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246_6.html

表12-18　数据变换后的用户翻页表

用户 ID	时　间	访 问 网 页
978851598	2015-02-11 15:24:25	http://www.****.com/info/jiaotong/jtlawdljtaqf/201410103308246.html

有关于用户翻页的数据处理代码如代码清单 12-6 所示。

代码清单12-6　Python访问MariaDB(MySQL)数据库进行数据变换

```
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine('mysql+pymysql://root:123456@127.0.0.1:3306/test?charset=utf8')
sql = pd.read_sql('cleaned_gzdata', engine, chunksize = 10000)

for i in sql: #逐块变换并去重
  d = i.copy()
  d['fullURL'] = d['fullURL'].str.replace('_\d{0,2}.html', '.html') #将下划线后面部
      分去掉，规范为标准网址
  d = d.drop_duplicates() #删除重复记录
  d.to_sql('changed_gzdata', engine, index = False, if_exists = 'append') #保存
```

代码详见：demo/code/ sql_data_change.py

由于在探索阶段发现有部分网页的所属类别是错误的，需对其数据进行网址分类，且分析目标是分析咨询类别与知识类别，因此需对这些网址进行手动分类，其分类的规则和结果见表 12-19，其中对网址中包含“ask”、“askzt”关键字的记录人为归类至咨询类别，对网址中包含“zhishi”、“faguizt”关键字的网址归类为知识类别。

表12-19　网页类别规则

类　型	总 记 录 数	百　分　比	说　明
咨询类	384 092	66.6%	网址中包含“ask”、“askzt”关键字
知识类	188 421	32.7%	网址中包含“zhishi”、“faguizt”关键字

因为目标是需要为用户提供个性化的推荐，在处理数据的过程中需要进一步对数据进行分类，其分类方法如图 12-7 所示，图中知识部分是由很多小的类别组成。由于所提供的原始数据中知识类别无法进行内部分类，故从业务上进行分析，可以采用其网址的构成对其进行分类。对表 12-20 中的用户访问记录进行分类，其分类的结果见表 12-21。

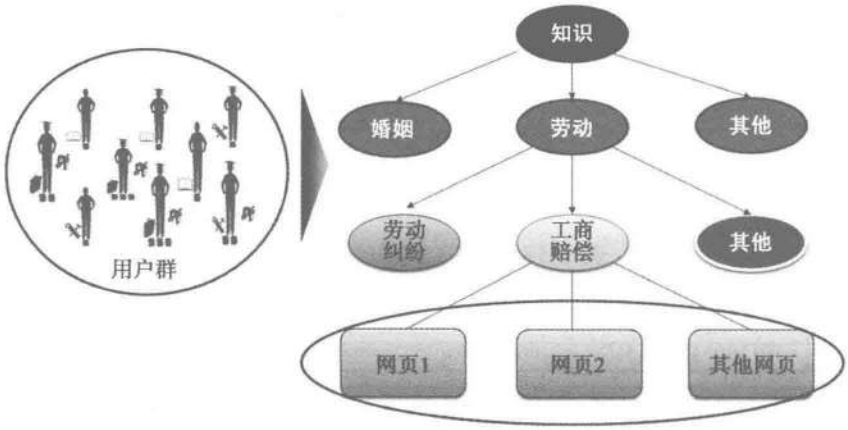


图 12-7 网页分类图

表12-20 网页分类表

用 户	网 址
863142519	http://www.****.com/info/minshi/fagui/2012111982349.html
863142519	http://www.****.com/info/shuifa/yys/201403042882164_2.html
863142519	http://www.****.com/info/jiaotong/jtnews/20130123121426.html

表12-21 网页分类结果表

用 户	类 别 1	类 别 2	类 别 3
863142519	zhishi	minshi	fagui
863142519	zhishi	shuifa	yys
863142519	zhishi	jiaotong	jtnews

网址分类的数据处理见代码清单 12-7。

代码清单12-7 Python访问MariaDB(MySQL)数据库进行网址分类

```
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine('mysql+pymysql://root:123456@127.0.0.1:3306/test?charset=utf8')
sql = pd.read_sql('changed_gzdata', engine, chunksize = 10000)
```

```
for i in sql: #逐块变换并去重
  d = i.copy()
  d['type_1'] = d['fullURL'] #复制一列
  d['type_1'][d['fullURL'].str.contains('(ask)|(askzt)')] = 'zixun' #将含有ask、
      askzt关键字的网址的类别一归为咨询（后面的规则就不详细列出来了，实际问题自己添加即可）
  d.to_sql('splited_gzdata', engine, index = False, if_exists = 'append') #保存
```

代码详见：demo/code/ sql_data_split.py

统计分析每一类中的记录，以知识类别中的婚姻法为例进行统计分析见表 12-22。可见其网页的点击率基本满足二八定律，即 80% 的网页只占了浏览量的 20% 左右，通过这个规则，按点击行为进行分类，20% 的网页是热点网页，其他 80% 的页面属于点击次数少的。因此在进行推荐过程中，需要将其分开进行推荐，已达到最优的推荐效果。

表12-22　婚姻知识点击次数统计表

点击次数	网页个数（3314）	网页百分比	记录数（16849）	记录百分比
1	1884	56.85	1884	11.18
2	618	18.65	1236	7.34
3	247	7.45	741	4.4
4	151	4.56	604	3.58
5 ～ 4679	414	12.49	12 384	73.5

3. 属性规约

由于推荐系统模型的输入数据需要，需对处理后的数据进行属性规约，提取模型需要的属性。本案例中模型需要的数据属性为用户和用户访问的网页。因此删除其他的属性，只选择用户与用户访问的网页，其输入数据集见表 12-23。

表12-23　模型输入数据集

用　户	网　页
2018622772	http://www.****.com/info/hunyin/hunyinfagui/201312112874686.html
1032300855	http://www.****.com/info/hunyin/lihuntiaojian/201408273306990.html
1032300856	http://www.****.com/info/gongsi/gzczgqgz/2010090150526.html
3029700497	http://www.****.com/info/xingshisusongfa/xingshipanjueshu/20110427115148.html
1971856960	http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html
1875780750	http://www.****.com/info/xingshisusongfa/xingshipanjueshu/20110706119307.html
1032299799	http://www.****.com/info/xingshisusongfa/xingshipanjueshu/20110503115363.html
1033227430	http://www.****.com/info/hunyin/yizhu/20120924165440.html
1928928104	http://www.****.com/info/hunyin/hunyinfagui/20111012157587.html
2937714434	http://www.****.com/info/jiaotong/jtaqchangshi/20121218120961.html

（续）

用　户	网　页
3029700498	http://www.****.com/info/fangdichan/tudizt/zhaijidi/20111019165581.html
1033227430	http://www.****.com/info/hunyin/yizhudingli/2010102668080.html
1032299831	http://www.****.com/info/yimin/England/yymtj/20100119259.html
3029700501	http://www.****.com/info/hunyin/lihuntiaojian/2011010894137.html
3029700365	http://www.****.com/info/fangdichan/tudizt/zhaijidi/201405152978392.html
1033227430	http://www.****.com/info/hunyin/yizhu/20120924165440.html
3029700372	http://www.****.com/info/fangdichan/tudizt/zhaijidi/201405152978392_2.html
1033227430	http://www.****.com/info/hunyin/yizhu/20120924165439.html
1875780622	http://www.****.com/info/hunyin/wuxiaohunyin/201412193311538.html

12.2.4 模型构建

在实际应用中，构造推荐系统时，并不是采用单一的推荐方法进行推荐。为了实现较好的推荐效果，大部分都结合多种推荐方法将推荐结果进行组合，最后得出推荐结果，在组合推荐结果时，可以采用串行或者并行的方法。本例所展示的是并行的组合方法，如图 12-8 ⊖ 所示。

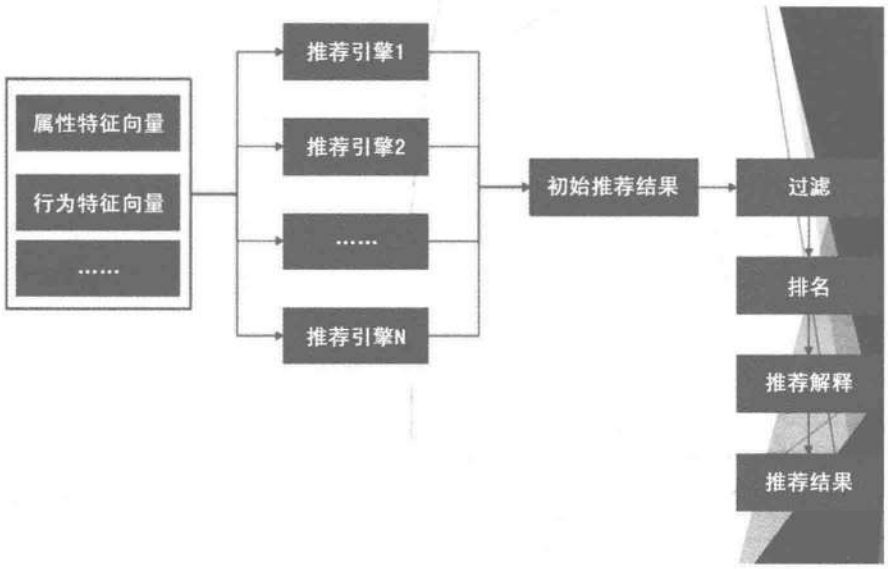


图 12-8　推荐系统流程图

针对此项目的实际情况，其分析目标的特点为：长尾网页丰富、用户个性化需求强烈

⊖ 图片来源于 http://www.docin.com/p-613240540.html。

以及推荐结果的实时变化，以及结合原始数据的特点：网页数明显小于用户数。本例采用基于物品的协同过滤推荐系统对用户进行个性化推荐，以其推荐结果作为推荐系统结果的重要部分。因其利用用户的历史行为为用户进行推荐，可以令用户容易信服其推荐结果。

基于用户和基于物品的协同过滤算法的区别在于：基于用户的协同过滤回答的是“将物品 A 推荐给哪个用户？”（假设答案是用户 B），基于物品的协同过滤回答的是“将哪个物品推荐给用户 B？”（在前面的假设下，答案是 A）。也就是说，两者的问法并不一样，但是最终的推荐结果是相同的。基于用户的协同过滤是用在用户少、物品多的场景，反之，基于物品的协同过滤就是用在用户多、物品少的场景。总的来说，都是为了减少计算量。而在数学上，两者的区别是在输入的用户 - 物品评分矩阵中，要不要进行转置，换句话说，只要对用户 - 物品评分矩阵进行转置，就可以将基于用户和基于物品的协同过滤相互转换。（通俗地说，计算机可没法识别究竟是“将物品推荐给人”还是“将人推荐给物品”。）

基于物品的协同过滤系统的一般处理过程：分析用户与物品的数据集，通过用户对项目的浏览与否（喜好）找到相似的物品，然后根据用户的历史喜好，推荐相似的项目给目标用户。图 12-9 是基于物品的协同过滤推荐系统图[⊖]，从图中可知用户 A 喜欢物品 A 和物品 C，用户 B 喜欢物品 A、物品 B 和物品 C，用户 C 喜欢物品 A。从这些用户的历史喜好可以分析出物品 A 和物品 C 是比较类似的，喜欢物品 A 的人都喜欢物品 C，基于这个数据可以推断用户 C 很有可能也喜欢物品 C，所以系统会将物品 C 推荐给用户 C。

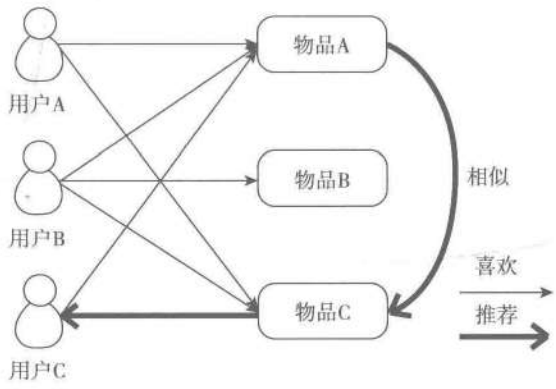


图 12-9　基于物品的推荐系统原理图

根据上述处理过程可知，基于物品的协同过滤算法主要分为两步。

- 计算物品之间的相似度。
- 根据物品的相似度和用户的历史行为给用户生成推荐列表。

其中，关于物品相似度计算的方法有：1）夹角余弦；2）杰卡德（Jaccard）相似系数；3）相关系数等。将用户对某一个物品的喜好或者评分作为一个向量，例如所有用户对物品 1 的评分或者喜好程度表示为 $A_1 = (x_{11}, x_{21}, x_{31}, \cdots x_{n1})$，所有用户对物品 M 的评分或者喜好程度表示为 $A_M = (x_{1m}, x_{2m}, x_{3m}, \cdots x_{nm})$，其中 m 为物品，n 为用户数。可以采用上述几种方法计算两个物品之间的相似度，其计算公式见表 12-24。由于用户的行为是二元选择（0-1 型），因此本例在计算物品的相似度过程中采用杰卡德相似系数法。

⊖ 图片引用网站 http://www.haodaima.net/art/2167399 中有关于物品的协同过滤推荐的原理图。

表12-24 相似度计算公式

方 法	公 式	说 明
夹角余弦	$sim_{1m}=\dfrac{\sum_{k=1}^{n}x_{k1}x_{km}}{\sqrt{\sum_{k=1}^{n}x_{k1}^2}\sqrt{\sum_{k=1}^{n}x_{km}^2}}$ $\left(sim_{1m}=\dfrac{A_1\cdot A_M}{\lvert A_1\rvert\times\lvert A_M\rvert}\right)$	取值范围为[-1，1]，当余弦值接近±1，表明两个向量有较强的相似性。当余弦值为0时，表示不相关
杰卡德相似系数	$J(A_1,A_M)=\dfrac{\lvert A_1\cap A_M\rvert}{\lvert A_1\cup A_M\rvert}$	分母 $A_1\cup A_M$ 表示喜欢物品1与喜欢物品M的用户总数，分子 $A_1\cap A_M$ 表示同时喜欢物品1和物品M的用户数
相关系数	$sim_{1m}=\dfrac{\sum_{k=1}^{n}(x_{k1}-\overline{A_1})(x_{km}-\overline{A_M})}{\sqrt{\sum_{k=1}^{n}(x_{k1}-\overline{A_1})^2}\sqrt{\sum_{k=1}^{n}(x_{km}-\overline{A_M})^2}}$	相关系数的取值范围是[-1，1]。相关系数的绝对值越大，则表明两者相关度越高

在协同过滤系统分析的过程中，用户行为存在很多种，例如浏览网页与否、是否购买、评论、评分、点赞等行为。如果要采用统一的方式表示所有行为是很困难的，因此，只能针对具体的分析目标进行具体的表示。在本例中，原始数据只记录了用户访问网站的浏览行为，因此用户的行为是浏览网页与否，并没有进行类似电子商务网站上的购买、评分和评论等用户行为。

完成各个物品之间的相似度的计算后，即可构成一个物品之间的相似度矩阵，类似于表12-25。通过采用相似度矩阵，推荐算法会给用户推荐与其物品最相似的 K 个物品。采用公式 $P=SIM\times R$，度量了推荐算法中用户对所有物品的感兴趣程度。其中，R 代表用户对物品的兴趣，SIM 代表所有物品之间的相似度，P 为用户对物品感兴趣的程度。因为用户的行为是二元选择（是与否），所以在用户对物品的兴趣 R 矩阵中只存在0和1。

表12-25 相似度矩阵

物 品	A	B	C	D
A	1	0.763	0.251	0
B	0.763	1	0.134	0.529
C	0.251	0.134	1	0.033
D	0	0.529	0.033	1

由于推荐系统是根据物品的相似度以及用户的历史行为对用户的兴趣度进行预测并推荐，因此在评价模型的时候需要用到一些评测指标。为了得到评测指标，一般是将数据集分成两部分：大部分作为模型训练集，小部分数据作为测试集。通过训练集得到的模型，在测试集上进行预测，然后统计出相应的评测指标，通过各个评测指标的值可以知道预测效果的好与坏。

其中，用 Python 借助 Numpy 来实现协同过滤算法并不困难，其代码如代码清单 12-8 所示。

代码清单12-8　Python实现协同过滤算法

```
import numpy as np

def Jaccard(a, b): #自定义杰卡德相似系数函数，仅对0-1矩阵有效
  return 1.0*(a*b).sum()/(a+b-a*b).sum()

class Recommender():

  sim = None #相似度矩阵

  def similarity(self, x, distance): #计算相似度矩阵的函数
    y = np.ones((len(x), len(x)))
    for i in range(len(x)):
      for j in range(len(x)):
        y[i,j] = distance(x[i], x[j])
    return y

  def fit(self, x, distance = Jaccard): #训练函数
    self.sim = self.similarity(x, distance)

  def recommend(self, a): #推荐函数
    return np.dot(self.sim, a)*(1-a)
```

代码详见：demo/code/ Recommender.py

本例采用随机打乱数据的方法完成模型的评测，具体方法为：首先用随机函数打乱原始数据的顺序（用 random 库的 shuffle() 函数可以轻松做到），然后将用户行为数据集按照均匀分布随机分成 M 份（本例取 $M = 10$），挑选一份作为测试集，将剩下的 M–1 份作为训练集。然后在训练集上建立模型，并在测试集上对用户行为进行预测，统计出相应的评测指标。为了保证评测指标并不是过拟合的结果，需要进行多次重复，由于开始时的随机函数打乱顺序保证了测试的随机性，因此，仅需要少数几次试验，就可以得出比较稳定可靠的评测结果，最后将实验测出的评测指标的平均值作为最终的评测指标。

1. 基于物品的协同过滤

基于协同过滤推荐算法包括两部分：基于用户的协同过滤推荐和基于物品的协同过滤推荐。本文结合实际的情况，选择基于物品的协同过滤算法进行推荐，其模型构建的流程如图 12-10 所示。

其中，训练集与测试集是通过交叉验证的方法划分后的数据集。通过协同过滤算法的原理可知，在建立推荐系统时，建模的数据量越大，越能消除数据中的随机性，得到的推荐结果对比数据量小要好。但是数据量越大，模型建立以及模型计算耗时就越久。因此本文选择数据处理后的婚姻与咨询的数据，其数据分布情况见表 12-26。在实际应用中，应当以大量的数据进行模型构建，得到的推荐结果相对会好些。

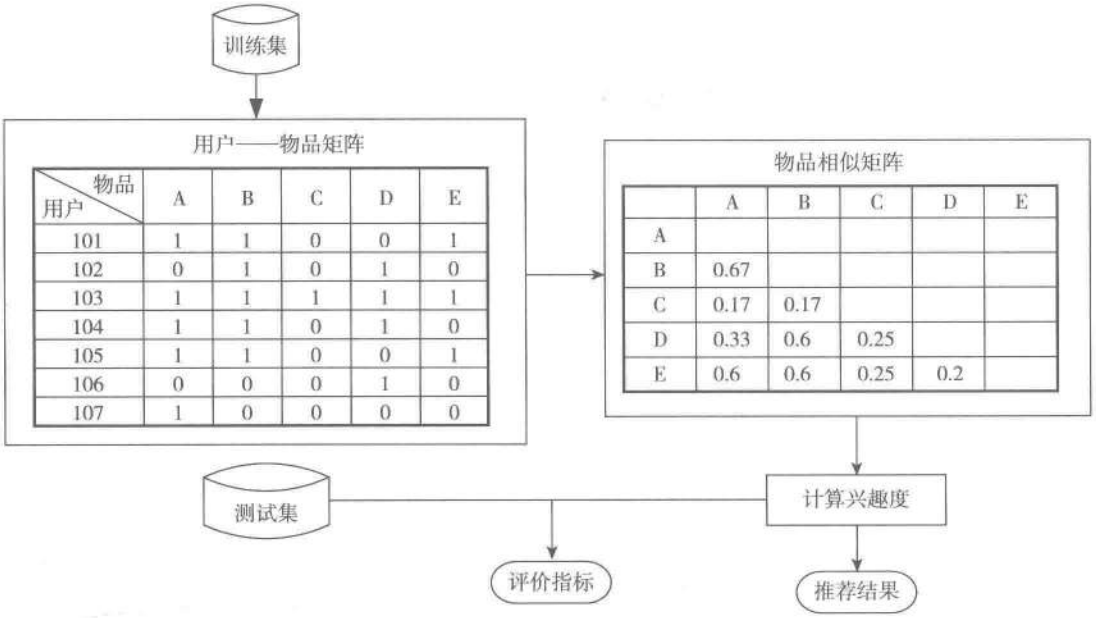


图 12-10 基于物品协同过滤建模流程图

表12-26 模型数据统计表

数 据 类 型	训练数据总数	物 品 个 数	访问平均次数	测试数据总数
婚姻类型	16 499	4428	4	1800
咨询类型	8000	4017	2	893

由于在实际数据中，物品数目过多，建立的用户物品矩阵与物品相似度矩阵是一个很庞大的矩阵。因此，在用户物品矩阵的基础上采用杰卡德相似系数的方法，计算出物品相似度矩阵。通过物品相似矩阵与测试集的用户行为，计算用户的兴趣度，获得推荐结果，进而计算出各种评价指标。

为了对比个性化推荐算法与非个性化推荐算法的好坏，本文选择了两种非个性化算法和一种个性化算法进行建模并对其进行模型评价与分析。两种非个性化算法为：Random 算法和 Popular 算法。其中，Random 算法是每次都随机挑选用户没有产生过行为的物品并推荐给他。Popular 算法是按照物品的流行度，为用户推荐他没有产生过行为的物品中最热门的物品。个性化算法为基于物品的协同过滤算法。利用 3 种算法，采用相同的交叉验证的方法，对数据进行建模分析，获得各个算法的评价指标。

2. 模型评价

如何去评价一个推荐系统的优劣？一般可以从如下几个方面整体进行考虑：用户、物品提供者、提供推荐系统网站[7]。好的推荐系统能够满足用户的需求，推荐其感兴趣的物品。同时在推荐的物品中，不能全部是热门的物品，也需要用户反馈意见帮助完善其推荐系统。

因此，好的推荐系统不仅能预测用户的行为，而且能帮助用户发现可能会感兴趣，但却不易被发现的物品。同时，推荐系统还应该帮助商家将长尾中的好商品发掘出来，推荐给可能会对它们感兴趣的用户。在实际应用中，评测推荐系统对三方影响是必不可少的。评测指标主要来源于 3 种评测推荐效果的实验方法，即离线测试、用户调查和在线实验。

离线测试是通过从实际系统中提取数据集，然后采用各种推荐算法对其进行测试，获得各个算法的评测指标。这种实验方法的好处是不需要真实用户参与。

注意　离线测试的指标和实际商业指标存在差距，比如预测准确率和用户满意度之间就存在很大差别，高预测准确率不等于高用户满意度。所以，当推荐系统投入实际应用之前，需要利用测试的推荐系统进行用户调查。

用户调查利用测试的推荐系统调查真实用户，观察并记录他们的行为，并让他们回答一些相关的问题。通过分析用户的行为和他们反馈的意见，判断测试推荐系统的好坏。

顾名思义，在线测试就是直接将系统投入实际应用中，通过不同的评测指标比较不同的推荐算法的结果，比如点击率、跳出率等。

由于本例中的模型是采用离线的数据集构建的，因此在模型评价阶段采用离线测试的方法获取评价指标。因为不同表现方式的数据集，其评测指标也不同，针对不同的数据方式，其评测指标的公式见表 12-27。

表12-27　评测指标表

数据表现方式	指　标　1	指　标　2	指　标　3
预测准确度	$RMSE = \sqrt{\frac{1}{N}\sum (r_{ui} - r_{ui})^2}$	$MAE = \frac{1}{N}\sum \lvert r_{ui} - r_{ui} \rvert$	
分类准确度	$precesion = \frac{TP}{TP + FP}$	$recall = \frac{TP}{TP + FN}$	$F1 = \frac{2PR}{P + R}$

在某些电子商务的网站中，存在对物品进行打分的功能。在此种数据的情况下，如果要预测用户对某个物品的评分，就需要用预测准确度的数据表现方式，其中评测的指标有均方根误差（*RMSE*），平均绝对误差（*MAE*）。其中，r_{ui} 代表用户 u 对物品 i 的实际评分，r_{ui} 代表推荐算法预测的评分，N 代表实际参与评分的物品总数。

在电子商务网站中，用户只有二元选择，例如，喜欢与不喜欢、浏览与否等。针对这种类型的数据预测，就要用到分类准确度。其评测指标有准确率（P、*precesion*），它表示用户对一个被推荐产品感兴趣的可能性。召回率（R、*recall*）表示一个用户喜欢的产品被推荐的概率。F1 指标表示综合考虑准确率与召回率因素，更好地评价算法的优劣。相关的指标说明见表 12-28。

除了上述指标外，还有一些评价指标如下。

- 真正率 TPR = TP /（TP + FN）意思为：正样本预测结果数 / 正样本实际数，即召回率。
- 假正率 FPR = FP /（FP + TN）意思为：被预测为正的负样本结果数 / 负样本实际数。

表12-28 分类准确度指标说明表

		预 测		合 计
		合计推荐物品数（正）	未被推荐物品数（负）	
实际	用户喜欢物品数（正）	TP	FN	TP+FN
	用户不喜欢物品数（负）	FP	TN	FP+TN
合计		TP+FP	TN+FN	

由于本例用户的行为是二元选择，因此在对模型进行评价的指标为分类准确度指标。针对婚姻知识类与咨询类的数据进行模型构造，通过 3 种推荐算法，以及不同 K 值（推荐 K 取值为 3、5、10、15、20、30）的情况下所得出的准确率与召回率的评价指标。婚姻知识类的评价指标图如图 12-11 所示，从图中可看出，Popular 算法是随着推荐个数 K 的增加，其召回率 R 将变大，准确率 P 将变小。基于物品的协同过滤算法的不同，随着推荐个数 K 的增加，其召回率 R 变大，准确率 P 也会上升。当达到某一临界点时，其准确率 P 随着 K 的增大而变小。3 种算法的其他评价指标，见表 12-29。从表中可以看出，在此数据下，随机推荐的结果最差，但是随着 K 值的增加，其 F1 值也在增加，而 Popular 算法的推荐效果随着 K 值的增加会越来越差，其 F1 值一直在下降，相对协同过滤算法，在 K = 5 的时候，其 F1 值最大，然后会随着 K 值增加而下降。比较不同算法之间的差异，从表中可以看出，随机推荐的效果最差。当 K 取值 3 和 5 时，Popular 算法优于协同过滤算法。但是当 K 值增加时，其推荐效果就不如协同过滤算法。从表中可以看出协同过滤算法相对较“稳定”。

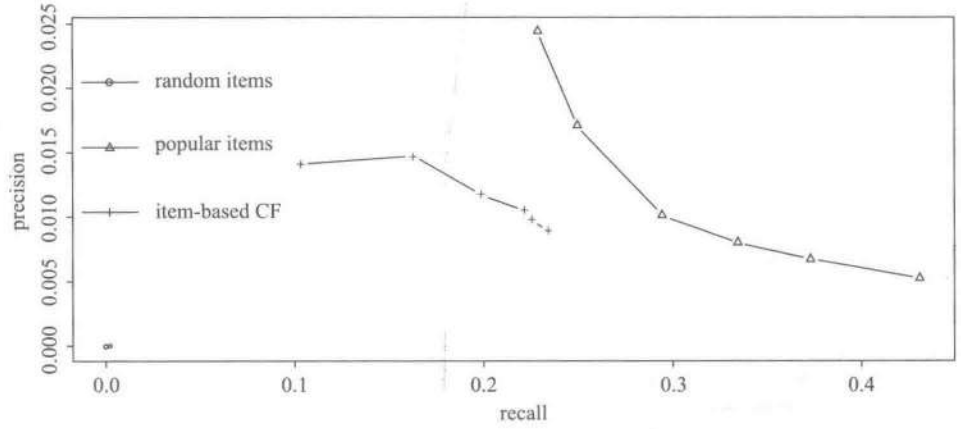


图 12-11 婚姻知识类准确率 – 召回率图

对于咨询类的数据，3 种算法得出的准确率与召回率的结果如图 12-12 所示。可以看出 Popular 算法、随机算法的准确率和召回率都很低。但是协同过滤算法推荐的结果比其他算法推荐要好得多。产生这样的结果主要是因为数据问题：1）咨询类的数据量不够；2）业务上分析咨询的页面会很多，很少存在大量访问的页面。算法的其他评价指标，见表 12-30。从

表中可以看出，在此数据下，Popular 算法与随机推荐算法的结果差，其 F1 值基本上是 0。在协同过滤算法中，当 K = 5 的时候，其 F1 值最大，然后会随着 K 值增加而下降。针对这种情况，协同过滤算法优于其他两种算法。

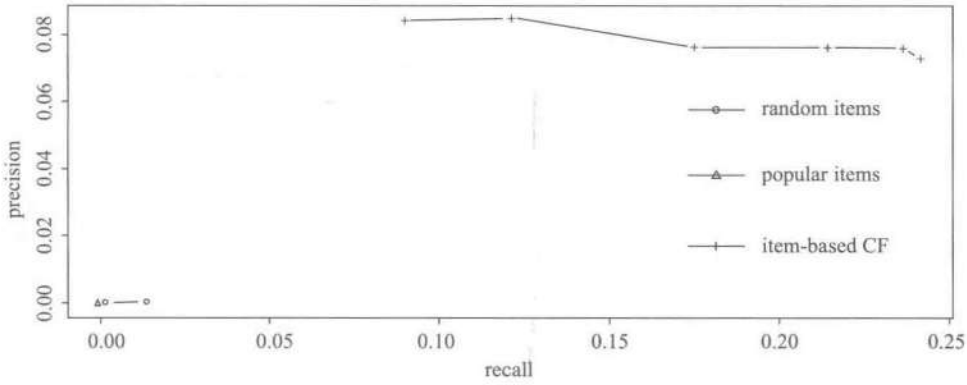


图 12-12　咨询类准确率 – 召回率图

表12-29　婚姻知识类模型评价指标

算　法	TP	FP	FN	TN	precision	recall	TPR	FPR	fvalue
random items 3	0.00	3.00	1.31	4222.69	0.00%	0.00%	0.00%	0.07%	NA
random items 5	0.00	5.00	1.31	4220.69	0.00%	0.00%	0.00%	0.12%	NA
random items 10	0.00	10.00	1.31	4215.69	0.01%	0.00%	0.00%	0.24%	0.00%
random items 15	0.00	15.00	1.31	4210.69	0.01%	0.00%	0.00%	0.35%	0.00%
random items 20	0.00	20.00	1.31	4205.69	0.01%	0.00%	0.00%	0.47%	0.00%
random items 30	0.00	30.00	1.31	4195.69	0.01%	0.20%	0.20%	0.71%	0.02%
popular items 3	0.07	2.93	1.24	4222.76	2.45%	22.86%	22.86%	0.07%	4.42%
popular items 5	0.09	4.91	1.22	4220.78	1.72%	24.98%	24.98%	0.12%	3.21%
popular items 10	0.10	9.90	1.21	4215.79	1.02%	29.48%	29.48%	0.23%	1.97%
popular items 15	0.12	14.88	1.19	4210.81	0.81%	33.48%	33.48%	0.35%	1.58%
popular items 20	0.14	19.86	1.17	4205.83	0.68%	37.29%	37.29%	0.47%	1.34%
popular items 30	0.16	29.84	1.15	4195.85	0.53%	43.19%	43.19%	0.71%	1.05%
item-based CF 3	0.03	2.26	1.28	4223.43	1.42%	10.33%	10.33%	0.05%	2.49%
item-based CF 5	0.05	3.63	1.26	4222.05	1.48%	16.29%	16.29%	0.09%	2.71%
item-based CF 10	0.06	6.93	1.25	4218.76	1.17%	19.90%	19.90%	0.16%	2.21%
item-based CF 15	0.07	10.06	1.24	4215.63	1.05%	22.17%	22.17%	0.24%	2.01%
item-based CF 20	0.07	13.02	1.24	4212.67	0.98%	22.61%	22.61%	0.31%	1.87%
item-based CF 30	0.08	18.61	1.24	4207.08	0.90%	23.48%	23.48%	0.44%	1.73%

表12-30 咨询类模型评价指标

算 法	TP	FP	FN	TN	precision	recall	TPR	FPR	fvalue
random items 3	0.00	3.00	1.19	3877.81	0.00%	0.00%	0.00%	0.08%	0.00%
random items 5	0.00	5.00	1.19	3875.81	0.00%	0.00%	0.00%	0.13%	0.00%
random items 10	0.00	10.00	1.19	3870.81	0.00%	0.00%	0.00%	0.26%	0.00%
random items 15	0.00	15.00	1.19	3865.81	0.02%	0.18%	0.18%	0.39%	0.04%
random items 20	0.01	19.99	1.18	3860.82	0.05%	1.13%	1.13%	0.52%	0.10%
random items 30	0.01	29.99	1.18	3850.82	0.03%	1.13%	1.13%	0.77%	0.07%
popular items 3	0.00	3.00	1.19	3877.81	0.00%	0.00%	0.00%	0.08%	0.00%
popular items 5	0.00	5.00	1.19	3875.81	0.00%	0.00%	0.00%	0.13%	0.00%
popular items 10	0.00	10.00	1.19	3870.81	0.00%	0.00%	0.00%	0.26%	0.00%
popular items 15	0.00	15.00	1.19	3865.81	0.00%	0.00%	0.00%	0.39%	0.00%
popular items 20	0.00	20.00	1.19	3860.81	0.00%	0.00%	0.00%	0.52%	0.00%
popular items 30	0.00	30.00	1.19	3850.81	0.00%	0.00%	0.00%	0.77%	0.00%
item-based CF 3	0.08	0.85	1.11	3879.96	8.41%	8.98%	8.98%	0.02%	16.83%
item-based CF 5	0.13	1.32	1.06	3879.49	8.48%	12.10%	12.10%	0.03%	16.95%
item-based CF 10	0.22	2.40	0.97	3878.41	7.62%	17.51%	17.51%	0.06%	15.24%
item-based CF 15	0.31	3.23	0.88	3877.58	7.61%	21.41%	21.41%	0.08%	15.21%
item-based CF 20	0.36	3.88	0.83	3876.92	7.58%	23.63%	23.63%	0.10%	15.16%
item-based CF 30	0.37	4.81	0.83	3876.00	7.29%	24.14%	24.14%	0.12%	14.57%

3. 结果分析

通过基于项目的协同过滤算法，针对每个用户进行推荐，推荐相似度排名前 5 的项目，其婚姻知识类推荐结果见表 12-31，其咨询类的推荐结果见表 12-32。

表12-31 婚姻知识类推荐结果

用 户	访问网址	推荐网址
116010	http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html	[1] http://www.****.com/info/hunyin/lihunshouxu/201312042874014.html [2] http://www.****.com/info/hunyin/lhlawlhxy/201403182883138.html [3] http://www.****.com/info/hunyin/hunyinfagui/201411053308986.html [4] http://www.****.com/info/hunyin/jihuashengyu/20120215163891.html [5] http://www.****.com/info/hunyin/hynews/201407073018800.html
11175899	http://www.****.com/info/hunyin/lhlawlhss/2010120781273.html http://www.****.com/info/hunyin/lhlawlhzx/20120821165124.html	[1]http://www.****.com/info/hunyin/fuyangyiwu/201404222884700.html [2] http://www.****.com/info/hunyin/hunyinfagui/201410153308460.html

（续）

用　户	访问网址	推荐网址
11175899	http://www.****.com/info/hunyin/lhlawlhzx/201311292873596.html http://www.****.com/info/hunyin/lhlawlhzx/201408253306854.html	[3] http://www.****.com/info/hunyin/hunyinjiufen/pohuaijunhunzui/20130719167114.html [4] http://www.****.com/info/hunyin/jiehuncaili/2011011297291.html [5] http://www.****.com/info/hunyin/lhlawlhxy/2011010492149.html
418673	http://www.****.com/info/hunyin/lihunfangchan/20110310125984.html	null

表12-32　咨询类推荐结果

用　户	访问网址	推荐网址
3951071	http://www.****.com/ask/question_10244513.html http://www.****.com/ask/question_10244238.html	[1] http://www.****.com/ask/question_10243783.html [2] http://www.****.com/ask/question_10244541.html [3] http://www.****.com/ask/question_10223080.html [4] http://www.****.com/ask/question_10223488.html [5] http://www.****.com/ask/question_10246475.html
21777264	http://www.****.com/ask/question_10383635.html http://www.****.com/ask/question_10383635.html	[1] http://www.****.com/ask/question_10162051.html
22027534	http://www.****.com/ask/question_10290587.html	null

从上述推荐结果的表格中可知，根据用户访问的相关网址，对用户进行推荐。但是其推荐结果存在 null 的情况，产生这种情况是由于在目前的数据集中，出现访问此网址的只有单独一个用户，因此在协同过滤算法中计算它与其他物品的相似度为 0，所以出现无法推荐的情况。一般出现这样的情况，在实际中可以考虑其他的非个性化的推荐方法进行推荐，例如基于关键字、基于相似行为的用户等。

由于本例采用的是最基本的协同过滤算法进行建模，因此得出的模型结果也是一个初步的效果，实际应用过程中要结合业务进行分析，对模型进一步改造。一般情况下，最热门物品往往具有较高的“相似性”。例如，热门的网址，访问各类网页的大部分人都会进行访问，在计算物品相似度的过程中，就可以知道各类网页都和某些热门的网址有关。因此处理热门网址的方法有：1）在计算相似度的过程中，可以加强对热门网址的惩罚，降低其权重，比如对相似度平均化或者对数化等方法。2）将推荐结果中的热门网址过滤掉，推荐其他的网址，将热门网址以热门排行榜的形式进行推荐，见表 12-33。

表12-33　婚姻知识类热门排行榜

网　址	内　容	点击次数
http://www.****.com/info/hunyin/lhlawlhxy/20110707137693.html	离婚协议书范本（2015 年版）	4697
http://www.****.com/info/hunyin/jihuashengyu/20120215163891.html	2015 最新产假规定	574
http://www.****.com/info/hunyin/hunyinfagui/201411053308986.html	新婚姻法 2015 全文	531

（续）

网　址	内　容	点击次数
http://www.****.com/info/hunyin/jiehun/hunjia/20110920152787.html	广州法定婚假多少天	222
http://www.****.com/info/hunyin/jihuashengyu/201411053308990.html	男人陪产假国家规定 2015	211

在协同过滤推荐过程中，两个物品相似是因为它们共同出现在很多用户的兴趣列表中，也可以说是每个用户的兴趣列表都对物品的相似度产生贡献。但是，并不是每个用户的贡献度都相同。通常不活跃的用户要么是新用户，要么是只来过网站一两次的老用户。在实际分析中，一般认为新用户倾向于浏览热门物品，首先他们对网站还不熟悉，只能点击首页的热门物品，而老用户会逐渐开始浏览冷门的物品。因此可以说，活跃用户对物品相似度的贡献应该小于不活跃的用户。所以，在改进相似度的过程中，取用户活跃度对数的倒数作为分子，即本例中相似度的公式为：

$$J(A_1, A_M) = \frac{\sum_{N \in |A_1 \cap A_M|} \frac{1}{\log(1 + A(N))}}{|A_1 \cup A_M|}$$

然而，在实际应用中，为了尽量提高推荐的准确率，还会将基于物品的相似度矩阵按最大值归一化，其好处不仅仅在于增加推荐的准确度，还可以提高推荐的覆盖率和多样性。由于本例的推荐是针对某一类数据进行，因此不存在类间的多样性，所以本节就不进行讨论。

当然，除了个性化推荐列表，还有另一个重要的推荐应用就是相关推荐列表。有过网购经历的用户都知道，当你在电子商务平台上购买一个商品时，它会在商品信息下面展示相关的商品。一种是包含购买了这个商品的用户也经常购买的其他商品，另一种是包含浏览过这个商品的用户经常购买的其他商品。这两种相关推荐列表的区别为：使用了不同用户行为计算物品的相似性。

12.3 上机实验

1. 实验目的

- 了解协同过滤算法在互联网电子商务中的应用以及实现过程。
- 了解 Python 连接数据库的方法，并对其进行操作的过程。主要步骤有 MariaDB 的安装搭建，PyMySQL、SQLAlchemy 的安装，以及 Pandas 读取数据库等。

2. 实验内容

依据本例的数据抽取以及数据处理方法，得到用户与物品（访问网页）的记录，通过使用用户与婚姻知识类型和婚姻咨询类型的数据，采用 Python 构建其推荐系统模型。

- 因数据量大，采用 Python 连接数据库的方式抽取数据，并且通过 Python 对数据库进行日常的数据操作。

- 用户点击网页体现了用户对某些网页的关注程度，利用协同过滤算法能计算出与某些网页相似的网页的相似程度，根据相似程度的高低，将用户未点击过的并且有可能有兴趣的网页推荐给用户，实现智能推荐。

3. 实验方法与步骤

实验一

利用 Python 连接 MariaDB (MySQL) 数据库，实现对数据的查询、删除、增加等日常操作。

1）打开 Python，安装 PyMySQL、SQLAlchemy，然后参考本章代码连接本地安装的数据库。当然，可以不用 Python，直接用 PyMySQL 或 SQLAlchemy 进行数据库操作，它们本身是一个完善的数据库操作工具（用 Pandas 是为了更好地进行数据分析，就数据库操作而言，PyMySQL、SQLAlchemy 之一就很不错了）。

2）由于数据库中含有中文内容，需要正确设置连接的编码格式。

3）通过 Pandas 连接数据库后，所进行的数据操作和之前通过 read_csv()、read_excel() 函数读取的数据操作并无不同。

4）通过 to_sql() 方法，试着将处理后的数据保存到数据库中。

5）基于 Pandas 的 sql 操作简单直接，读者熟悉后，请查阅相关教程，尝试直接通过 PyMySQL 或 SQLAlchemy 进行数据库操作，以增加对这两个工具的了解。

实验二

利用 Python 完成推荐系统的模型构建，以及预测的推荐结果，并完成模型的评价工作。

1）由于协同过滤算法并不复杂，因此，读者应该可以读懂该算法，并且能够参考本章提供的代码，自行编写出协同过滤算法的代码。

2）通过自行编写的协同过滤算法的代码，给出预测的推荐结果。

3）采用 3 种模型对输入数据进行建模，用随机打乱数据验证的方法，获取各个模型在不同的推荐值的情况下的评价指标值，并计算出各个模型下的 F1 指标。

4）画出 3 种模型的准确率与召回率的指标图，并将各个指标保存到文本。

4. 思考与实验总结

1）如何通过 Python 操作数据库中存在的中文编码？

2）如何设置计算相似度的方法，例如采用余弦方法计算其物品间的相似度？

12.4　拓展思考

本例中主要分析的内容为婚姻知识类别与婚姻咨询类别的有关记录，其结果比目前网页上基于关键词的推荐发散性要强，取到一个互补的效果。但由于目前公司主营业务侧重于咨询方面，且在探索分析的环节可以看出咨询记录占整个记录里的 50% 左右，因此对于咨询类别的页面的推荐需要进一步改造，其数据可以从用户访问的原始数据中提取，见表 12-34。

表12-34 原始数据

realIP	realAreacode	userAgent	userOS	userID	clientID	timestamp	timestamp_format	pagePath	ymd	fullURL	fullURLId	hostname	pageTitle	pageTitle	pageTitle
1531222030	140100	UCWEB/2.0	Other	499670012.1	499670012.1	1428041479371	2015/4/3 14:11	/ask/ques	20150403	http://www.lawtime.cn/ask/question_8399551.html	101003	www.lawti	做作住房公	69	房产买卖纠
1531222030	140100	UCWEB/2.0	Other	499670012.1	499670012.1	1428041479536	2015/4/3 14:11	/ask/ques	20150403	http://www.lawtime.cn/ask/question_8399551.html	101003	www.lawti	做作住房公	69	房产买卖纠
1706656375	140100	Mozilla/5.	Windows 7	1259341818	1259341818	1429353422107	2015/4/18 18:37	/ask/ques	20150418	http://www.lawtime.cn/ask/question_10937991.html	101003	www.lawti	做资金监管	37	立案侦查
4223238775	140100	Mozilla/5.	Windows 7	908370090.1	908370090.1	1426579834667	2015/3/17 16:10	/ask/ques	20150317	http://www.lawtime.cn/ask/question_421092.html	101003	www.lawti	做庄赌博公	26	定罪量刑
1110054106	140100	Mozilla/5.	Windows XP	2068832749	2068832749	1423635850415	2015/2/11 14:24	/ask/ques	20150211	http://www.lawtime.cn/ask/question_6925984.html	101003	www.lawti	做重伤的你	68	工伤赔偿纠
1046706190	140100	Mozilla/5.	Windows 7	847612256.1	847612256.1	1427852615867	2015/4/1 9:43	/ask/exp/	20150401	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawti	做钟点工要	62	劳动合同纠
465868558	140100	Mozilla/5.	Windows XP	1610868312	1610868312	1428394886464	2015/4/7 16:21	/ask/exp/	20150407	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawti	做钟点工要	62	劳动合同纠
1571227319	140100	Mozilla/5.	Windows XP	371696939.1	371696939.1	1429849254956	2015/4/24 12:20	/ask/exp/	20150424	http://www.lawtime.cn/ask/exp/8587.html	1999001	www.lawti	做钟点工要	62	劳动合同纠
2092450417	140100	Mozilla/5.	Windows 7	1632334036	1632334036	1427170907797	2015/3/24 12:21	/ask/ques	20150324	http://www.lawtime.cn/ask/question_5598369.html	101003	www.lawti	做直销产品	51	违约赔偿
1121454201	140100	Mozilla/4.	Windows XP	2136917696	2136917696	1428039745607	2015/4/3 13:42	/ask/ques	20150403	http://www.lawtime.cn/ask/question_3432948.html	101003	www.lawti	做侄子的法	17	儿童监护
2561650547	140100	Mozilla/5.	Mac OS X	1316725305	1316725305	1425978540658	2015/3/10 17:09	/info/yil	20150310	http://www.lawtime.cn/info/yiliao/zrsb/20100721431	107001	www.lawti	做整形手术	64	医疗事故赔
409120636	140100	Mozilla/5.	Windows XP	362380012.1	362380012.1	1429463755830	2015/4/20 1:15	/ask/ques	20150420	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawti	做杂志,使	26	定罪量刑
1699823729	140100	Mozilla/5.	Windows 7	38549427.14	38549427.14	1429871000544	2015/4/24 18:23	/ask/ques	20150424	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawti	做杂志,使	26	定罪量刑
1257332336	140100	Mozilla/5.	Windows 7	1242996761	1242996761	1430292457522	2015/4/29 15:27	/ask/ques	20150429	http://www.lawtime.cn/ask/question_3764976.html	101003	www.lawti	做杂志,使	26	定罪量刑
2586839161	140100	Mozilla/5.	Windows 7	1283840943	1283840943	1423489880092	2015/2/9 21:51	/ask/ques	20150209	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawti	做有限责任	41	股权纠纷
2828223345	140100	Mozilla/5.	Mac OS X	980744139.1	980744139.1	1425095196930	2015/2/28 11:46	/ask/ques	20150228	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawti	做有限责任	41	股权纠纷
1018329870	140100	Mozilla/5.	Windows 7	967084001.1	967084001.1	1428378107184	2015/4/7 11:41	/ask/ques	20150407	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawti	做有限责任	41	股权纠纷
2119376756	140100	Mozilla/5.	Windows 7	1599121760	1599121760	1428486243962	2015/4/8 17:44	/ask/ques	20150408	http://www.lawtime.cn/ask/question_3173773.html	101003	www.lawti	做有限责任	41	股权纠纷
2249626126	140100	Mozilla/5.	Windows 7	1796985316	1796985316	1423798297363	2015/2/13 11:31	/ask/ques	20150213	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawti	做有限公司	41	股权纠纷
3020128887	140100	Mozilla/5.	Windows 7	1188934890	1188934890	1429697248802	2015/4/22 18:07	/ask/ques	20150422	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawti	做有限公司	41	股权纠纷
3423224433	140100	Mozilla/5.	Windows 7	1150849692	1150849692	1430291567285	2015/4/29 15:12	/ask/ques	20150429	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawti	做有限公司	41	股权纠纷
3423224433	140100	Mozilla/5.	Windows 7	1150849692	1150849692	1430291685261	2015/4/29 15:14	/ask/ques	20150429	http://www.lawtime.cn/ask/question_6617278.html	101003	www.lawti	做有限公司	41	股权纠纷
1859491598	140100	Mozilla/5.	Windows 7	1102973681	1102973681	1422725918555	2015/2/1 1:38	/ask/ques	20150201	http://www.lawtime.cn/ask/question_914636.html	101003	www.lawti	做游戏外挂	26	定罪量刑
1242119863	140100	Mozilla/5.	Windows 7	984584705.1	984584705.1	1423536816616	2015/2/10 10:53	/ask/ques	20150210	http://www.lawtime.cn/ask/question_3402035.html	101003	www.lawti	做引产犯法	26	定罪量刑
3609131066	140100	Mozilla/4.	Windows XP	1221287319	1221287319	1429685946200	2015/4/22 14:59	/ask/ques	20150422	http://www.lawtime.cn/ask/question_6548781.html	101003	www.lawti	做银行黑户	78	信用卡恶意
2731637774	140100	Mozilla/5.	Windows XP	1204438324	1204438324	1427360738669	2015/3/26 17:05	/ask/ques	20150326	http://www.lawtime.cn/ask/question_7658765.html	101003	www.lawti	做银行黑户	78	信用卡恶意
2731637774	140100	Mozilla/5.	Windows XP	1204438324	1204438324	1427369024680	2015/3/26 19:23	/ask/ques	20150326	http://www.lawtime.cn/ask/question_7658765.html	101003	www.lawti	做银行黑户	78	信用卡恶意
2601561724	140100	Mozilla/5.	Other	118385063.1	118385063.1	1423150087176	2015/2/5 23:28	/ask/ques	20150205	http://www.lawtime.cn/ask/question_1152117.html	101003	www.lawti	做银行贷款	52	金融债务
3787742832	140100	Mozilla/5.	Windows 7	1342347607	1342347607	1425099001584	2015/2/28 12:50	/ask/ques	20150228	http://www.lawtime.cn/ask/question_10382308.html	101003	www.lawti	做银保工作	26	定罪量刑
1296711793	140100	Mozilla/5.	Windows 7	1546761287	1546761287	1422889805462	2015/2/2 23:10	/ask/ques	20150202	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
2059693169	140100	Mozilla/5.	Windows XP	202229259.1	202229259.1	1423645371761	2015/2/11 17:02	/ask/ques	20150211	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
364932977	140100	Mozilla/5.	Windows 8.	1911420797	1911420797	1424879011165	2015/2/25 23:43	/ask/ques	20150225	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
698003068	140100	Mozilla/5.	Windows 7	601074264.1	601074264.1	1425913507611	2015/3/9 23:05	/ask/ques	20150309	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
460808305	140100	Mozilla/5.	Windows 7	812125454.1	812125454.1	1426514986691	2015/3/16 22:09	/ask/ques	20150316	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
3080340593	140100	Mozilla/4.	Windows XP	919277708.1	919277708.1	1427280973102	2015/3/25 18:56	/ask/ques	20150325	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
221596127	140100	Mozilla/5.	Windows 7	372903090.1	372903090.1	1428398837055	2015/4/7 17:27	/ask/ques	20150407	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
705385592	140100	Mozilla/5.	Windows 7	690991433.1	690991433.1	1428475106146	2015/4/8 14:38	/ask/ques	20150408	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
3827394762	140100	Mozilla/5.	Windows XP	730261919.1	730261919.1	1428567244656	2015/4/9 16:14	/ask/ques	20150409	http://www.lawtime.cn/ask/question_3653924.html	101003	www.lawti	做遗产继承	21	医患纠纷
1011128334	140100	Mozilla/5.	Android	1963560652	1963560652	1423496417446	2015/2/9 23:40	/ask/ques	20150209	http://www.lawtime.cn/ask/question_100745.html	101003	www.lawti	做医疗事故	31	故意伤害
2228186993	140100	Mozilla/5.	Windows 7	1328923830	1328923830	1426219851133	2015/3/13 12:10	/ask/ques	20150313	http://www.lawtime.cn/ask/question_100745.html	101003	www.lawti	做医疗事故	31	故意伤害

数据详见：demo/data/7law.sql

首先需要解决冷启动问题，当新的用户产生，如何对其进行推荐？然后在进行相似度设计的过程中未考虑到对热门网址的处理以及那些无法得到推荐结果的网页。由于在原始数据中，每个网页都存在一个标题，可以通过采用文本挖掘的分析方法。通过文本挖掘，找出每个网页文本中的隐含语义，然后通过分析文本中隐含特征，将用户与物品联系在一起，相关的名称有 LSI、pLSA、LDA 和 Topic Model。当然，也可以通过这种方法提取出关键字，通过 tf-idf 的方法对其关键字进行定义权重，然后采用最近邻的方法求出那些无法得到推荐列表的结果。因此，针对本例的数据，可以采用隐语义模型实现推荐，同样采用离线的方法对其进行测试，然后对比各种推荐方法的评价指标，最后将各种推荐结果进行结合。

12.5　小结

本章主要介绍了协同过滤算法在电子商务领域中的应用，实现了对用户的个性化推荐。通过对用户访问日志的数据进行分析与处理，采用基于物品的协同过滤算法对处理好的数据进行建模分析，最后通过模型评价与结果分析，发现基于物品的协同过滤算法的优缺点，同时对其缺点提出改进的方法。结合上机实验，有助于更好地理解协同过滤推荐算法的原理以及处理过程。

Chapter 13 第 13 章

财政收入影响因素分析及预测模型

13.1 背景与挖掘目标

在我国现行的分税制财政管理体制下，地方财政收入不仅是国家财政收入的重要组成部分，还具有其相对独立的构成内容。如何有效地利用地方财政收入，合理地分配来促进地方的发展，提高市民的收入和生活质量是每个地方政府需要考虑的首要问题。因此，对地方财政收入进行预测，不但是必要的，而且是可能的。科学、合理地预测地方财政收入，对于克服年度地方预算收支规模的随意性和盲目性，正确处理地方财政与经济的相互关系具有十分重要的意义。

某市作为改革开放的前沿城市，其经济发展在全国经济中的地位举足轻重。目前，该市在财政收入规模、结构等方面与北京、上海和深圳等城市仍有一定差距，存在不断完善的空间。本案例旨在通过研究，发现影响该市目前以及未来地方财源建设的因素，并对其进行深入分析，提出对该市地方财源优化的具体建议，供政府决策参考，同时为其他经济发展较快的城市提供借鉴。

考虑到数据的可得性，本案例所用的财政收入分为地方一般预算收入和政府性基金收入。地方一般预算收入包括：①税收收入，主要包括企业所得税和地方所得税中中央和地方共享的 40%，地方享有的 25% 的增值税、营业税和印花税等；②非税收入，包括专项收入、行政事业性收费、罚没收入、国有资本经营收入和其他收入等。政府性基金收入是国家通过向社会征收以及出让土地、发行彩票等方式取得的收入，并专项用于支持特定基础设施建设和社会事业发展的收入。

由于 1994 年我国对财政体制进行了重大改革，开始实行分税制财政体制，影响了财政收入相关数据的连续性，在 1994 年前后不具有可比性。由于没有合适的数学手段来调整这种数据的跃变，仅对 1994 年及其后的数据进行分析，本案例所用数据均来自《某市统计年鉴》(1995–2014)。

表 13-1 给出了某市 1994 ～ 2013 年财政收入以及相关因素的数据，为进一步寻找某市财

表13-1　某市财政收入及其相关数据

日期	社会从业人数	在岗职工工资总额	社会消费品零售总额	城镇居民人均可支配收入	城镇居民人均消费性支出	年末总人口	全社会固定资产投资额	地区生产总值	第一产业产值	税收	居民消费价格指数	第三产业与第二产业产值比	居民消费水平	财政收入
1994	3 831 732	181.54	448.19	7571	6212.7	6 370 241	525.71	985.31	60.62	65.66	120	1.029	5321	64.87
1995	3 913 824	214.63	549.97	9 038.16	7 601.73	6 467 115	618.25	1 259.2	73.46	95.46	113.5	1.051	6529	99.75
1996	3 928 907	239.56	686.44	9905.31	8092.82	6 560 508	638.94	1468.06	81.16	81.16	108.2	1.064	7008	88.11
1997	4 282 130	261.58	802.59	10 444.6	8 767.98	6 664 862	656.58	1678.12	85.72	91.7	102.2	1.092	7694	106.07
1998	4 453 911	283.14	904.57	11 255.7	9422.33	6 741 400	758.83	1893.52	88.88	114.61	97.7	1.2	8027	137.32
1999	4 548 852	308.58	1000.69	12 018.52	9751.44	6 850 024	878.26	2139.18	92.85	152.78	98.5	1.198	8549	188.14
2000	4 962 579	348.09	1121.13	13 966.53	11 349.47	7 006 896	923.67	2 492.74	94.37	170.62	102.8	1.348	9566	219.91
2001	5 029 338	387.81	1248.29	14 694	11 467.35	7 125 979	978.21	2841.65	97.28	214.53	98.9	1.467	10 473	271.91
2002	5 070 216	453.49	1370.68	13 380.47	10 671.78	7 206 229	1 009.24	3 203.96	103.07	202.18	97.6	1.56	11 469	269.1
2003	5 210 706	533.55	1 494.7	15 002.59	11 570.58	7 251 888	1 175.17	3 758.62	109.91	222.51	100.1	1.456	12 360	300.55
2004	5 407 087	598.33	1 677.77	16 884.16	13 120.83	7 376 720	1 348.93	4 450.55	117.15	249.01	101.7	1.424	14 174	338.45
2005	5 744 550	665.32	1 905.84	18 287.24	14 468.24	7 505 322	1519.16	5154.23	130.22	303.41	101.5	1.456	16 394	408.86
2006	5 994 973	738.97	2 199.14	19 850.66	15 444.93	7 607 220	1 696.38	6081.86	128.51	356.99	102.3	1.438	17 881	476.72
2007	6 236 312	877.07	2624.24	22 469.22	18 951.32	7 734 787	1863.34	7140.32	149.87	429.36	103.4	1.474	20 058	838.99
2008	6 529 045	1005.37	3187.39	25 316.72	20 835.95	7 841 695	2 105.54	8 287.38	169.19	508.84	105.9	1.515	22 114	843.14
2009	6 791 495	1118.03	3615.77	27 609.59	22 820.89	7 946 154	2 659.85	9 138.21	172.28	557.74	97.5	1.633	24 190	1107.67
2010	7 110 695	1304.48	4476.38	30 658.49	25 011.61	8 061 370	3263.57	10 748.28	188.57	664.06	103.2	1.638	29 549	1399.16
2011	7 431 755	1700.87	5243.03	34 438.08	28 209.74	8 145 797	3412.21	12 423.44	204.54	710.66	105.5	1.67	34 214	1535.14
2012	7 512 997	1969.51	5977.27	38 053.52	30 490.44	8 222 969	3758.39	13 551.21	213.76	760.49	103	1.825	37 934	1579.68
2013	7 599 295	2110.78	6882.85	42 049.14	33 156.83	8 323 096	4 454.55	15 420.14	228.46	852.56	102.6	1.906	41 972	2088.14

政收入的关键影响因素做准备。

本次数据挖掘建模目标如下。

1）梳理影响地方财政收入的关键特征，分析、识别影响地方财政收入的关键特征的选择模型。

2）结合目标 1）的因素分析，对某市 2015 年的财政总收入及各个类别收入进行预测。

13.2 分析方法与过程

我国很多学者已经对财政收入的影响因素进行了很多研究，但是他们大多先建立财政收入与各待定的影响因素之间的多元线性回归模型，运用最小二乘估计方法来估计回归模型的系数，通过系数能否通过检验来检验它们之间的关系，这样的结果对数据的依赖程度很大，并且普通最小二乘估计求得的解往往是局部最优解，后续的检验可能就会失去应有的意义。

近几十年来，现代统计技术不断完善和发展，对新的数据运用新的方法来考察地方财政收入的影响因素是有必要的。本案例在已有研究的基础上运用 Adaptive-Lasso 变量选择方法来研究影响地方财政收入的因素。

在以往的文献中，对影响财政收入的因素的分析大多使用普通最小二乘法来对回归模型的系数进行估计，预测变量的选取采用的则是逐步回归。然而，无论是最小二乘法还是逐步回归，都有其不足之处。它们一般都局限于局部最优解而不是全局最优解。如果预测变量过多，子集选择的计算过程具有不可实行性，且子集选择具有内在的不连续性，从而导致子集选择极度多变。Lasso 是近年来被广泛应用于参数估计和变量选择的方法之一，并且在确定的条件下，使用 Lasso 方法进行变量选择已经被证明是一致的。案例选用了 Adaptive-Lasso 方法来探究地方财政收入与各因素之间的关系。

Lasso 是由 Tibshirani[22]（1996）提出的将参数估计与变量选择同时进行的一种正则化方法。Lasso 参数估计被定义如下。

$$\hat{\beta}(\text{lasso}) = \underset{\beta}{\operatorname{argmin}}^{2} \left\| y - \sum_{j=1}^{p} x_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{13-1}$$

其中，λ 为非负正则参数，$\lambda \sum_{j=1}^{p} |\beta_j|$称为惩罚项。

Lasso 方法虽然可以解决最小二乘法和逐步回归局部最优估计的不足，但是其自身需要满足一定的苛刻条件。Hui ZOU[23]（2006）提出了一种改进的 Lasso 方法，其改进之处为给不同的系数加上了不同的权重，被称为 Adaptive-Lasso 方法，定义如下。

$$\hat{\beta}^{*(n)} = \underset{\beta}{\operatorname{argmin}} \left\| y - \sum_{j=1}^{p} x_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{\omega} |\beta_j| \tag{13-2}$$

其中，权重 $\hat{\omega}_j = \dfrac{1}{|\hat{\beta}_j|^{\gamma}} (\gamma > 0)$，$j = 1, 2, \cdots, p$，$\hat{\beta}_j$为由普通最小二乘法得出的系数。

设变量 $X^{(0)} = \{X^{(0)}(i),\ i = 1,\ 2\cdots n\}$ 为一非负单调原始数据序列，建立灰色预测模型：首先对 $X^{(0)}$ 进行一次累加得到一次累加序列 $X^{(1)} = \{X^{(1)}(k),\ k = 1,\ 2,\ \cdots,\ n\}$。

对 $X^{(1)}$ 可建立下述一阶线性微分方程。

$$\frac{\mathrm{d}X^{(1)}}{\mathrm{d}t} + aX^{(1)} = u \tag{13-3}$$

即 GM(1, 1) 模型。

求解微分方程，得到预测模型如下。

$$\hat{X}^{(1)}(k+1) = \left[\hat{X}^{(1)}(0) - \frac{\hat{u}}{\hat{a}}\right]e^{-\hat{a}k} + \frac{\hat{u}}{\hat{a}} \tag{13-4}$$

由于 GM(1, 1) 模型得到的是一次累加量，将 GM(1, 1) 模型所得数据 $\hat{X}^{(1)}(k+1)$ 经过累减还原为 $\hat{X}^{(0)}(k+1)$，即 $X^{(0)}$ 的灰色预测模型为：

$$\hat{X}^{(0)}(k+1) = (e^{-\hat{a}} - 1)\left[X^{(0)}(n) - \frac{\hat{u}}{\hat{a}}\right]e^{-\hat{a}k} \tag{13-5}$$

后验差检验模型精度表见表 13-2。

表13-2　后验差检验判别参照表

P	C	模型精度
>0.95	<0.35	好
>0.80	<0.5	合格
>0.70	<0.65	勉强合格
<0.70	>0.65	不合格

13.2.1　灰色预测与神经网络的组合模型

在 Adaptive-Lasso 变量选择的基础上，鉴于灰色预测对小数据量数据预测的优良性能，对单个选定的影响因素建立灰色预测模型，得到它们在 2014 年及 2015 年的预测值。由于神经网络较强的适用性和容错能力，对历史数据建立训练模型，把灰色预测的数据结果代入训练好的模型中，就得到了充分考虑历史信息的预测结果，即 2015 年某市财政收入及各个类别的收入。

图 13-1 为基于数据挖掘技术的财政收入分析预测模型流程，主要包括以下步骤[⊖]。

1）从某市统计局网站以及各统计年鉴搜集到该市财政收入以及各类别收入相关数据。

2）利用步骤 1）形成的已完成数据预处理的建模数据，建立 Adaptive-Lasso 变量选择模型。

3）在步骤 2）的基础上建立单变量的灰色预测模型以及人工神经网络预测模型。

4）利用步骤 3）的预测值代入构建好的人工神经网络模型中，从而得到 2014/2015 年某市财政收入以及各类别收入的预测值。

⊖ 陈庚，卢丹丹，万浩文．基于数据挖掘技术的市财政收入分析预测模型．第三届泰迪杯全国大学生数据挖掘竞赛（http://www.tipdm.org）优秀作品。

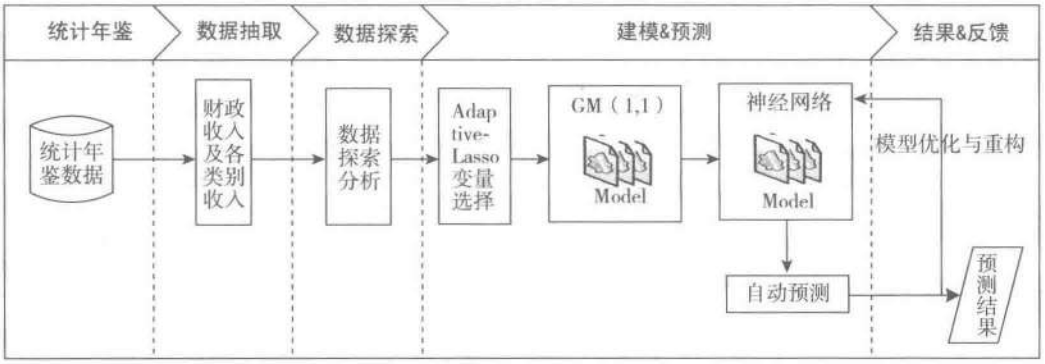


图 13-1 基于数据挖掘技术的财政收入分析预测模型流程

13.2.2 数据探索分析

影响财政收入（y）的因素有很多，在查阅大量文献的基础上，通过经济理论对财政收入的解释以及对实践的观察，考虑一些与能源消耗关系密切并且直观上有线性关系的因素，初步选取以下因素为自变量，分析它们之间的关系。

社会从业人数（x1）：就业人数的上升伴随着居民消费水平的提高，从而间接增加财政收入。

在岗职工工资总额（x2）：在岗职工工资总额反映的是社会分配情况，主要影响财政收入中的个人所得税、房产税以及潜在的消费能力。

社会消费品零售总额（x3）：代表社会整体消费情况，是可支配收入在经济生活中的体现。当社会消费品零售总额增长时，表明社会消费意愿强烈，某种程度上会导致财政收入中增值税的增长；同时，当消费增长时，也会引起经济系统中其他方面发生变动，最终导致财政收入的增长。

城镇居民人均可支配收入（x4）：居民收入越高，消费能力越强，同时意味着其工作积极性越高，创造出的财富越多，从而能带来财政收入的更快和持续增长。

城镇居民人均消费性支出（x5）：居民在消费商品的过程中会产生各种税费，税费又是调节生产规模的手段之一。在商品经济发达的今天，居民消费得越多，对财政收入的贡献就越大。

年末总人口（x6）：在地方经济发展水平既定的条件下，人均地方财政收入与地方人口数呈反比例变化。

全社会固定资产投资额（x7）：全社会固定资产投资额是建造和购置固定资产的经济活动，即固定资产再生产活动。主要通过投资来促进经济增长，扩大税源，进而拉动财政税收收入整体增长。

地区生产总值（x8）：表示地方经济发展水平。一般来讲，政府财政收入来源于即期的地区生产总值。在国家经济政策不变、社会秩序稳定的情况下，地方经济发展水平与地方财政

收入之间存在着密切的相关性，越是经济发达的地区，其财政收入的规模就越大。

第一产业产值（x9）：取消农业税、实施三农政策，第一产业对财政收入的影响更小。

税收（x10）：由于其具有征收的强制性、无偿性和固定性特点，可以为政府履行其职能提供充足的资金来源。因此，各国都将其作为政府财政收入的最重要的收入形式和来源。

居民消费价格指数（x11）：反映居民家庭购买的消费品及服务价格水平的变动情况，影响城乡居民的生活支出和国家的财政收入。

第三产业与第二产业产值比（x12）：表示产业结构。第三产业生产总值代表国民经济水平，是财政收入的主要影响因素，当产业结构逐步优化时，财政收入也会随之增加。

居民消费水平（x13）：在很大程度上受整体经济状况 GDP 的影响，从而间接影响地方财政收入。

（1）描述分析

首先对已有数据进行描述性统计分析，获得对数据的整体性认识，表 13-3 显示了主要变量的描述性统计结果。由表可见财政收入（y）的均值和标准差分别为 618.08 和 609.25，这说明：第一，某市各年份财政收入存在较大差异。第二，2008 年后，某市各年份财政收入大幅上升。

表13-3　主要变量的描述性统计

	Min	Max	Mean	STD
x1	3 831 732.00	7 599 295.00	5 579 519.95	126 219.50
x2	181.54	2110.78	765.04	595.70
x3	448.19	6882.85	2370.83	1919.17
x4	7571.00	42 049.14	19 644.69	10 203.02
x5	6212.70	33 156.83	15 870.95	8199.77
x6	6 370 241.00	8 323 096.00	7 350 513.60	621 341.90
x7	525.71	4454.55	1712.24	1184.71
x8	985.31	15 420.14	5705.80	4478.40
x9	60.62	228.46	129.50	5.05
x10	65.66	852.56	340.22	251.58
x11	97.50	120.00	103.31	5.51
x12	1.03	1.91	1.42	2.53
x13	5321.00	41 972.00	17 273.80	11 109.19
y	64.87	2088.14	618.08	609.25

代码清单 13-1 是原始数据的概括性度量。

代码清单13-1　原始数据概括性度量

```
#-*- coding: utf-8 -*-
import numpy as np
```

```
import pandas as pd
inputfile = '../data/data1.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据
r = [data.min(), data.max(), data.mean(), data.std()] #依次计算最小值、最大值、均值、标准差
r = pd.DataFrame(r, index = ['Min', 'Max', 'Mean', 'STD']).T  #计算相关系数矩阵
np.round(r, 2) #保留两位小数
```

代码详见：demo/code/ gaikuo.py

（2）相关分析

相关系数可以用来描述定量和变量之间的关系，初步判断因变量与解释变量之间是否具有线性相关性。原始数据求解 Pearson 相关系数如代码清单 13-2 所示。

表13-4 变量Pearson相关系数矩阵

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	y
x1	1.00	0.95	0.95	0.97	0.97	0.99	0.95	0.97	0.98	0.98	−0.29	0.94	0.96	0.94
x2	0.95	1.00	1.00	0.99	0.99	0.92	0.99	0.99	0.98	0.98	−0.13	0.89	1.00	0.98
x3	0.95	1.00	1.00	0.99	0.99	0.92	1.00	0.99	0.98	0.99	−0.15	0.89	1.00	0.99
x4	0.97	0.99	0.99	1.00	1.00	0.95	0.99	1.00	0.99	1.00	−0.19	0.91	1.00	0.99
x5	0.97	0.99	0.99	1.00	1.00	0.95	0.99	1.00	0.99	1.00	−0.18	0.90	0.99	0.99
x6	0.99	0.92	0.92	0.95	0.95	1.00	0.93	0.95	0.97	0.96	−0.34	0.95	0.94	0.91
x7	0.95	0.99	1.00	0.99	0.99	0.93	1.00	0.99	0.98	0.99	−0.15	0.89	1.00	0.99
x8	0.97	0.99	0.99	1.00	1.00	0.95	0.99	1.00	0.99	1.00	−0.15	0.90	1.00	0.99
x9	0.98	0.98	0.98	0.99	0.99	0.97	0.98	0.99	1.00	0.99	−0.23	0.91	0.99	0.98
x10	0.98	0.98	0.99	1.00	1.00	0.96	0.99	1.00	0.99	1.00	−0.17	0.90	0.99	0.99
x11	−0.29	−0.13	−0.15	−0.19	−0.18	−0.34	−0.15	−0.15	−0.23	−0.17	1.00	−0.43	−0.16	−0.12
x12	0.94	0.89	0.89	0.91	0.90	0.95	0.89	0.90	0.91	0.90	−0.43	1.00	0.90	0.87
x13	0.96	1.00	1.00	1.00	0.99	0.94	1.00	1.00	0.99	0.99	−0.16	0.90	1.00	0.99
y	0.94	0.98	0.99	0.99	0.99	0.91	0.99	0.99	0.98	0.99	−0.12	0.87	0.99	1.00

由表 13-4 可知，居民消费价格指数（x11）与财政收入的线性关系不显著，而且呈现负相关。其余变量均与财政收入呈现高度的正相关关系。

代码清单13-2 原始数据求解Pearson相关系数

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
inputfile = '../data/data1.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据
np.round(data.corr(method = 'pearson'), 2) #计算相关系数矩阵，保留两位小数
```

代码详见：demo/code/ correlation.py

13.2.3　模型构建

1. Adaptive-Lasso 变量选择模型

运用 LARS 算法来解决公式（2）的 Adaptive-Lasso 估计，对于每给一个 γ，该算法会寻找一个最优的 λ_n。此处取 $\gamma = 1$，用 Python 编制相应的程序后运行得到如下结果，见表 13-5。

表13-5　系数表

x1	x2	x3	x4	x5	x6	x7
−0.0001	−0.2309	0.1375	−0.0401	0.0760	0.0000	0.3069
x8	x9	x10	x11	x12	x13	
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

代码清单13-3　Adaptive-Lasso变量选择

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../data/data1.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据

#导入AdaptiveLasso算法，要在较新的Scikit-Learn才有。
from sklearn.linear_model import AdaptiveLasso
model = AdaptiveLasso(gamma=1)
model.fit(data.iloc[:,0:13],data['y'])
model.coef_ #各个特征的系数
```

代码详见：demo/code/ 1-adaptive-lasso.py

由表 13-5 可以看出，年末总人口、地区生产总值、第一产业产值、税收、居民消费价格指数、第三产业与第二产业产值比以及居民消费水平等因素的系数为 0，即在模型建立的过程中这几个变量被剔除了，这是因为居民消费水平与城镇居民人均消费性支出存在明显的共线性，在构建模型的过程中，Adaptive-Lasso 方法剔除了这个变量；由于某市存在流动人口与外来打工人口多的特性，年末总人口并不显著影响某市财政收入；居民消费价格指数与财政收入的相关性太小以致可以忽略；由于农牧业各税在各项税收总额中所占比重过小，而且该市于 2005 年取消了农业税，因而第一产业对地方财政收入的贡献率极低；其他变量被剔除均有类似于上述的原因。这说明使用 Adaptive-Lasso 方法构建模型时，能够剔除存在共线性关系的变量，同时体现了 Adaptive-Lasso 方法对多指标进行建模的优势。

综上所述，利用 Adaptive-Lasso 方法识别影响财政收入的关键影响因素是社会从业人数、在岗职工工资总额、社会消费品零售总额、城镇居民人均可支配收入、城镇居民人均消费性支出以及全社会固定资产投资额。

2. 财政收入及各类别收入预测模型

（1）某市财政收入预测模型

对 Adaptive-Lasso 变量选择方法识别的影响财政收入的因素建立灰色预测与神经网络的

组合预测模型，其参数设置为误差精度 10^{-7}，学习次数 10 000 次，神经元个数为 Lasso 变量选择方法选择的变量个数 6。社会从业人数（x1）、在岗职工工资总额（x2）、社会消费品零售总额（x3）、城镇居民人均可支配收入（x4）、城镇居民人均消费性支出（x5）、固定资产投资额（x7）指标的 2014 年及 2015 年数值均通过 Python 建立灰色预测模型得出。Python 及流行的扩展库并没有提供灰色预测功能，因此我们自行编写了灰色预测函数（GM11.py）。预测结果的精度等级见表 13-6。

表13-6 灰色预测模型地方财政收入相关因素精度表

	x1	x2	x3	x4	x5	x7
2014 预测值	8 142 148	2239.295	7042.313	43 611.84	35 046.63	4600.405
预测精度等级	好	好	好	好	好	好

地方财政收入灰色预测的数据处理见代码清单 13-4。

代码清单13-4 地方财政收入灰色预测

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数

inputfile = '../data/data1.csv' #输入的数据文件
outputfile = '../tmp/data1_GM11.xls' #灰色预测后保存的路径
modelfile = '../tmp/net.model' #模型保存路径
data = pd.read_csv(inputfile) #读取数据
data.index = range(1994, 2014)

data.loc[2014] = None
data.loc[2015] = None
l = ['x1', 'x2', 'x3', 'x4', 'x5', 'x7']
for i in l:
  f = GM11(data[i][range(1994, 2014)].as_matrix())[0]
  data[i][2014] = f(len(data)-1) #2014年预测结果
  data[i][2015] = f(len(data)) #2015年预测结果
  data[i] = data[i].round(2) #保留两位小数

data[l+['y']].to_excel(outputfile) #结果输出
```

代码详见：demo/code/ 1-huise.py

将数据零均值标准化后，代入地方财政收入所建立的 3 层神经网络预测模型（输入层 6 节点，隐藏层 12 节点，输出层 1 节点），得到某市财政收入 2015 年的预测值为 2 366.42 亿元，相关数据见表 13-7，其中红色字体的数据为预测数据。图 13-2 为神经网络地方财政收入真实值与预测值对比图。

表13-7 地方财政收入及其相关因素历史数据和预测表

	x1	x2	x3	x4	x5	x7	y	y_pred
1994	3 831 732	181.54	448.19	7571	6212.7	525.71	64.87	65.42

（续）

	x1	x2	x3	x4	x5	x7	y	y_pred
1995	3 913 824	214.63	549.97	9038.16	7601.73	618.25	99.75	100.76
1996	3 928 907	239.56	686.44	9905.31	8092.82	638.94	88.11	94.51
1997	4 282 130	261.58	802.59	10444.6	8767.98	656.58	106.07	126.59
1998	4 453 911	283.14	904.57	11255.7	9422.33	758.83	137.32	156.15
1999	4 548 852	308.58	1000.69	12 018.52	9751.44	878.26	188.14	188.99
2000	4 962 579	348.09	1121.13	13 966.53	11 349.47	923.67	219.91	229.06
2001	5 029 338	387.81	1248.29	14 694	11 467.35	978.21	271.91	251.40
2002	5 070 216	453.49	1370.68	13 380.47	10 671.78	1009.24	269.1	264.33
2003	5 210 706	533.55	1494.27	15 002.59	11 570.58	1175.17	300.55	304.14
2004	5 407 087	598.33	1677.77	16 884.16	13 120.83	1348.93	338.45	347.75
2005	5 744 550	665.32	1905.84	18 287.24	14 468.24	1519.16	408.86	412.09
2006	5 994 973	738.97	2199.14	19 850.66	15 444.93	1696.38	476.72	478.98
2007	6 236 312	877.07	2624.24	22 469.22	18 951.32	1863.34	838.99	829.60
2008	6 529 045	1005.37	3187.39	25 316.72	20 835.95	2105.54	843.14	850.68
2009	6 791 495	1118.03	3615.77	27 609.59	22 820.89	2659.85	1107.67	1105.85
2010	7 110 695	1304.48	4476.38	30 658.49	25 011.61	3263.57	1399.16	1400.83
2011	7 431 755	1700.87	5243.03	34 438.08	28 209.74	3412.21	1535.14	1533.20
2012	7 512 997	1969.51	5977.27	38 053.52	30 490.44	3758.39	1579.68	1577.76
2013	7 599 295	2110.78	6882.85	42 049.14	33 156.83	4454.55	2088.14	2086.17
2014	8 142 148	2239.30	7042.31	43 611.84	35 046.63	4600.41		2114.62
2015	8 460 489	2581.14	8166.92	47 792.22	38 384.22	5214.78		2366.42

数据详见：demo/data/revenue.xls

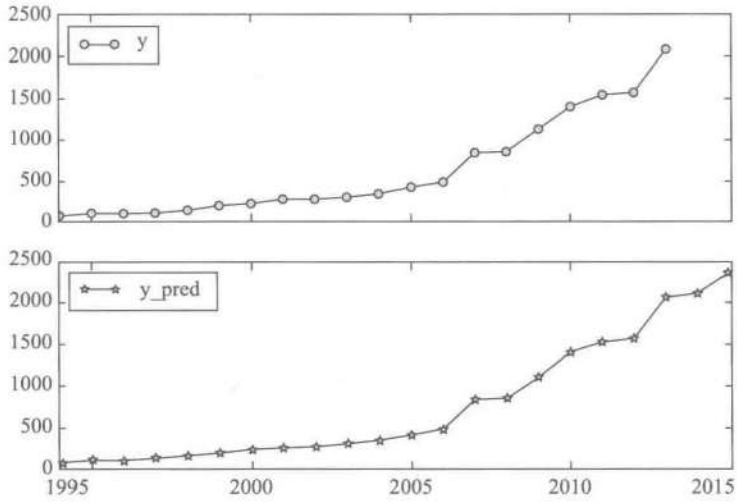


图 13-2　地方财政收入真实值与预测值对比图

代码清单 13-5 为地方财政收入神经网络预测模型。

代码清单13-5 地方财政收入神经网络预测模型

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../tmp/data1_GM11.xls' #灰色预测后保存的路径
outputfile = '../data/revenue.xls' #神经网络预测后保存的结果
modelfile = '../tmp/1-net.model' #模型保存路径
data = pd.read_excel(inputfile) #读取数据
feature = ['x1', 'x2', 'x3', 'x4', 'x5', 'x7'] #特征所在列

data_train = data.loc[range(1994,2014)].copy() #取2014年前的数据建模
data_mean = data_train.mean()
data_std = data_train.std()
data_train = (data_train - data_mean)/data_std #数据标准化
x_train = data_train[feature].as_matrix() #特征数据
y_train = data_train['y'].as_matrix() #标签数据

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(6, 12))
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(12, 1))
model.compile(loss='mean_squared_error', optimizer='adam') #编译模型
model.fit(x_train, y_train, nb_epoch = 10000, batch_size = 16) #训练模型，学习一万次
model.save_weights(modelfile) #保存模型参数

#预测，并还原结果。
x = ((data[feature] - data_mean[feature])/data_std[feature]).as_matrix()
data[u'y_pred'] = model.predict(x) * data_std['y'] + data_mean['y']
data.to_excel(outputfile)

import matplotlib.pyplot as plt #画出预测结果图
p = data[['y','y_pred']].plot(subplots = True, style=['b-o','r-*'])
plt.show()
```

代码详见：demo/code/ 1-yuce.py

（2）增值税预测模型

利用 Adaptive-Lasso 方法进行增值税影响因素的变量选择，通过表 13-8 可以看出，商品进口总值（x1）、工业增加值（x3）和工业增加值占 GDP（x5）比重这 3 个因素进入选择，其他因素的系数为 0。因为可以根据工业增加值及其占 GDP 比重可以算出地区生产总值，所以 Adaptive-Lasso 方法在构建模型的过程中剔除了地区生产总值这个变量；由于批发零售业对增值税的贡献率较低，所以该因素也被剔除。

表13-8　系数表

x1	x2	x3	x4	x5	x6
−1365.173 46	0.000 00	0.060 98	0.000 00	−447 747.889 28	0.000 00

Adaptive-Lasso 变量选择见代码清单 13-6。

代码清单13-6　Adaptive-Lasso变量选择

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../data/data2.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据

#导入AdaptiveLasso算法，要在较新的Scikit-Learn才有。
from sklearn.linear_model import AdaptiveLasso
model = AdaptiveLasso(gamma=1)
model.fit(data.iloc[:,0:6],data['y'])
model.coef_ #各个特征的系数
```

代码详见：demo/code/ 2-adaptive-lasso.py

对 Adaptive-Lasso 变量选择方法识别的影响增值税的因素建立神经网络预测模型，其参数设置为误差精度 10^{-7}，学习次数 10 000 次，神经元个数为 Lasso 变量选择方法选择的变量个数 3。商品进口总值（x1）、工业增加值（x3）和工业增加值占 GDP 比重（x5）指标的 2014 年及 2015 年数值可以通过 Python 建立灰色预测模型得出，后验差比值、预测精度等级见表 13-9。

表13-9　灰色预测模型增值税相关因素精度表

	x1	x3	x5
后验差比值	0.1853（<0.35）	0.0807	0.5067（[0.5，0.65]）
预测精度等级	好	好	勉强合格

增值税灰色预测如代码清单 13-7 所示。

代码清单13-7　增值税灰色预测

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数

inputfile = '../data/data2.csv' #输入的数据文件
outputfile = '../tmp/data2_GM11.xls' #灰色预测后保存的路径
data = pd.read_csv(inputfile) #读取数据
data.index = range(1999, 2014)

data.loc[2014] = None
data.loc[2015] = None
l = ['x1', 'x3', 'x5']
for i in l:
```

```
    f = GM11(data[i][range(1999, 2014)].as_matrix())[0]
    data[i][2014] = f(len(data)-1) #2014年预测结果
    data[i][2015] = f(len(data)) #2015年预测结果
    data[i] = data[i].round(6) #保留六位小数

data[l+['y']].to_excel(outputfile) #结果输出
```

代码详见：demo/code/ 2-huise.py

将数据零均值标准化后，代入增值税所建立的 3 层神经网络预测模型（输入层 3 节点，隐藏层 6 节点，输出层 1 节点），得到增值税的 2015 年预测值为 2 696 434 万元，相关数据见表 13-10，其中红色字体的数据为预测数据。图 13-3 为神经网络增值税真实值与预测值对比图。

表13-10 增值税及其相关因素历史数据和预测表

	x1	x3	x5	增 值 税	预 测 值
1999	93.18	7 980 207	0.373 051	288 972	275 731.96
2000	115.6	8 779 835	0.352 216	350 495	392 159.44
2001	114.13	9 554 676	0.336 237	443 213	436 172.48
2002	141.49	10 509 450	0.328 014	526 377	526 495.64
2003	180.52	13 141 254	0.349 63	581 898	536 624.45
2004	233.14	15 941 538	0.358 193	528 365	608 085.48
2005	268.07	18 439 550	0.357 756	816 119	745 395.53
2006	313.85	22 270 093	0.366 172	967 265	962 517.29
2007	355.91	26 029 310	0.364 54	1 115 007	1 127 192.77
2008	389.47	29 724 781	0.358 675	1 287 226	1 295 267.13
2009	392.82	31 173 422	0.341 133	1 375 085	1 374 782.89
2010	553.89	36 449 611	0.339 12	1 594 182	1 594 241.91
2011	596.94	41 405 926	0.333 289	1 573 830	1 573 826.78
2012	582.52	42 641 557	0.314 67	1 758 311	1 758 471.66
2013	560.89	47 548 175	0.308 351	2 216 017	2 216 008.18
2014	767.59	58 163 231	0.327 438		2 379 707.76
2015	862.30	65 803 730	0.325 358		2 696 434.08

数据详见：demo/data/VAT.xls

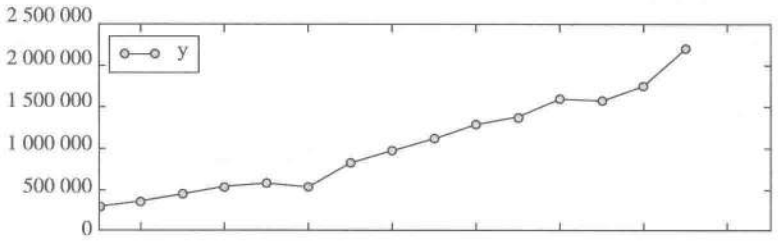


图 13-3 增值税真实值与预测值对比图

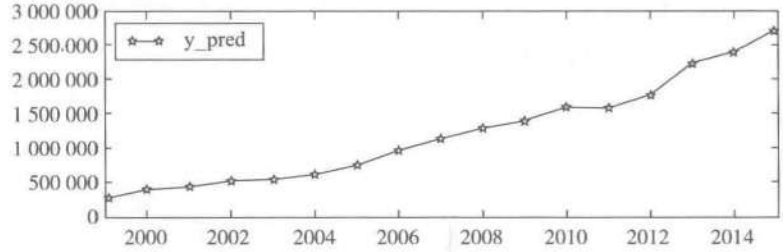


图 13-3 （续）

增值税神经网络预测模型如代码清单 13-8 所示。

代码清单13-8　增值税神经网络预测模型

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../tmp/data2_GM11.xls' #灰色预测后保存的路径
outputfile = '../data/VAT.xls' #神经网络预测后保存的结果
modelfile = '../tmp/2-net.model' #模型保存路径
data = pd.read_excel(inputfile) #读取数据
feature = ['x1', 'x3', 'x5'] #特征所在列

data_train = data.loc[range(1999,2014)].copy() #取2014年前的数据建模
data_mean = data_train.mean()
data_std = data_train.std()
data_train = (data_train - data_mean)/data_std #数据标准化
x_train = data_train[feature].as_matrix() #特征数据
y_train = data_train['y'].as_matrix() #标签数据

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(3, 6))
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(6, 1))
model.compile(loss='mean_squared_error', optimizer='adam') #编译模型
model.fit(x_train, y_train, nb_epoch = 10000, batch_size = 16) #训练模型，学习一万次
model.save_weights(modelfile) #保存模型参数

#预测，并还原结果。
x = ((data[feature] - data_mean[feature])/data_std[feature]).as_matrix()
data[u'y_pred'] = model.predict(x) * data_std['y'] + data_mean['y']
data[u'y_pred'] = data[u'y_pred'].round()
data.to_excel(outputfile)

import matplotlib.pyplot as plt #画出预测结果图
p = data[['y','y_pred']].plot(subplots = True, style=['b-o','r-*'])
plt.show()
```

代码详见：demo/code/ 2-yuce.py

（3）营业税预测模型

利用 Adaptive-Lasso 方法进行营业税影响因素的变量选择，通过表 13-11 可以看出，全社会固定资产投资额（x3）、城市商品零售价格指数（1978 = 100）（x4）、规模以上国有及国有控股工业企业亏损面（x6）和建筑业企业利润总额（x8）这 4 个因素进入选择，其他因素的系数为 0。

表13-11 系数表

x1	x2	x3	x4	x5
0.000 00	−0.000 00	0.141 97	0.053 83	0.000 00
x6	x7	x8	x9	x10
−0.185 23	0.000 00	−0.141 07	0.000 00	0.000 00

代码清单13-9 adaptive-lasso变量选择

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../data/data3.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据

#导入AdaptiveLasso算法，要在较新的Scikit-Learn才有。
from sklearn.linear_model import AdaptiveLasso
model = AdaptiveLasso(gamma=1)
model.fit(data.iloc[:,0:10],data['y'])
model.coef_ #各个特征的系数
```

代码详见：demo/code/ 3-adaptive-lasso.py

对 Adaptive-Lasso 变量选择方法识别的影响营业税的因素建立神经网络预测模型，其参数设置为误差精度 10^{-7}，学习次数 10 000 次，神经元个数为 Lasso 变量选择方法选择的变量个数 3。变量选择的指标的 2014 年及 2015 年数值可以通过 Python 建立灰色预测模型得出，后验差比值、预测精度等级见表 13-12。

表13-12 灰色预测模型营业税相关因素精度表

	x3	x4	x6	x8
后验差比值	0.1153	0.0179	0.1566	0.0719
预测精度等级	好	好	好	好

代码清单 13-10 是营业税灰色预测。

代码清单13-10 营业税灰色预测

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数
```

```
inputfile = '../data/data3.csv' #输入的数据文件
outputfile = '../tmp/data3_GM11.xls' #灰色预测后保存的路径
data = pd.read_csv(inputfile) #读取数据
data.index = range(1999, 2014)

data.loc[2014] = None
data.loc[2015] = None
l = ['x3', 'x4', 'x6', 'x8']
for i in l:
  f = GM11(data[i][range(1999, 2014)].as_matrix())[0]
  data[i][2014] = f(len(data)-1) #2014年预测结果
  data[i][2015] = f(len(data)) #2015年预测结果
  data[i] = data[i].round() #取整

data[l+['y']].to_excel(outputfile) #结果输出
```

代码详见：demo/code/ 3-huise.py

将数据零均值标准化后，代入营业税所建立的 3 层神经网络预测模型（输入层 4 节点，隐藏层 8 节点，输出层 1 节点），得到营业税的 2015 年预测值为 2 371 484 万元，相关数据见表 13-13，其中红色字体的数据为预测数据。图 13-4 为神经网络营业税真实值与预测值对比图。

表13-13　营业税及其相关因素历史数据和预测表

	x3	x4	x6	x8	营　业　税	预　测　值
1999	1 330 484	11 152 545	2 878 473	2 470 523	433 360	428 159
2000	1 436 406	13 767 475	3 250 326	2 561 326	479 698	480 787
2001	1 568 267	16 320 762	3 316 894	3 403 870	540 075	548 755
2002	1 603 966	18 895 479	3 457 617	3 733 922	613 161	603 208
2003	1 718 007	21 627 825	3 522 168	4 785 787	650 119	655 439
2004	1 939 100	25 453 413	3 712 961	5 459 314	793 520	786 952
2005	2 012 633	29 787 941	3 777 003	6 331 382	892 678	903 491
2006	2 145 067	35 118 425	3 783 416	6 870 406	1 027 971	1 027 857
2007	2 228 495	41 646 681	5 041 090	7 507 109	1 235 374	1 202 495
2008	2 553 936	48 903 250	5 398 216	8 754 491	1 279 793	1 315 502
2009	2 878 166	55 607 710	5 246 903	10 134 050	1 516 049	1 518 091
2010	3 573 047	65 574 525	5 727 122	12 805 288	1 777 343	1 768 435
2011	4 363 837	76 419 207	8 116 313	15 613 171	1 625 593	1 631 062
2012	4 564 947	86 167 948	8 626 775	17 417 072	1 747 616	1 742 505
2013	4 725 256	99 643 373	9 969 708	21 828 895	1 623 520	1 624 275

（续）

	x3	x4	x6	x8	营 业 税	预 测 值
2014	5 319 885	118 049 333	10 017 413	23 746 699		2 151 944
2015	5 919 520	137 165 274	11 096 340	27 870 540		2 371 484

数据详见：demo/data/sales_tax.xls

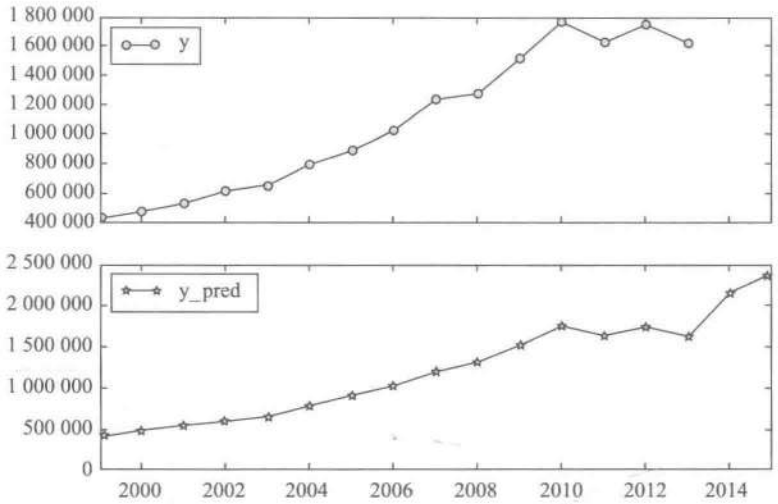


图 13-4 营业税真实值与预测值对比图

营业税神经网络预测模型如代码清单 13-11 所示。

代码清单13-11 营业税神经网络预测模型

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../tmp/data3_GM11.xls' #灰色预测后保存的路径
outputfile = '../data/sales_tax.xls' #神经网络预测后保存的结果
modelfile = '../tmp/3-net.model' #模型保存路径
data = pd.read_excel(inputfile) #读取数据
feature = ['x3', 'x4', 'x6', 'x8'] #特征所在列

data_train = data.loc[range(1999,2014)].copy() #取2014年前的数据建模
data_mean = data_train.mean()
data_std = data_train.std()
data_train = (data_train - data_mean)/data_std #数据标准化
x_train = data_train[feature].as_matrix() #特征数据
y_train = data_train['y'].as_matrix() #标签数据

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(4, 8))
```

```
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(8, 1))
model.compile(loss='mean_squared_error', optimizer='adam') #编译模型
model.fit(x_train, y_train, nb_epoch = 10000, batch_size = 16) #训练模型，学习一万次
model.save_weights(modelfile) #保存模型参数

#预测，并还原结果。
x = ((data[feature] - data_mean[feature])/data_std[feature]).as_matrix()
data[u'y_pred'] = model.predict(x) * data_std['y'] + data_mean['y']
data[u'y_pred'] = data[u'y_pred'].round(2)
data.to_excel(outputfile)

import matplotlib.pyplot as plt #画出预测结果图
p = data[['y','y_pred']].plot(subplots = True, style=['b-o','r-*'])
plt.show()
```

代码详见：demo/code/ 3-yuce.py

（4）企业所得税预测模型

利用 Adaptive-Lasso 方法进行企业所得税影响因素的变量选择，通过表 13-14 可以看出，规模以上工业企业盈亏相抵后的利润总额和建筑业企业利润总额这两个因素的系数为 0。这是因为建筑业企业利润总额与建筑业总产值存在线性关系，所以该变量应该被剔除。第二产业增加值（x1）、第三产业增加值（x2）、全社会固定资产投资额（x3）、城市商品零售价格指数（1978 = 100）（x4）、规模以上国有及国有控股工业企业亏损面（x6）、建筑业总产值（x7）、限额以上连锁店（公司）零售额（x9）、地方财政总收入（x10）8 个变量被选入影响企业所得税（y）的因素中。

表13-14　系数表

x1	x2	x3	x4	x5
0.014 74	−0.007 53	−0.006 07	3507.633 91	0.000 00
x6	x7	x8	x9	x10
−8893.786 00	0.020 10	0.000 00	0.007 12	0.005 26

Adaptive-Lasso 变量选择如代码清单 13-12 所示。

代码清单13-12　Adaptive-Lasso变量选择

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../data/data4.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据

#导入AdaptiveLasso算法，要在较新的Scikit-Learn才有。
from sklearn.linear_model import AdaptiveLasso
model = AdaptiveLasso(gamma=1)
```

```
model.fit(data.iloc[:,0:10],data['y'])
model.coef_ #各个特征的系数
```

代码详见：demo/code/ 4-adaptive-lasso.py

对 Adaptive-Lasso 变量选择方法识别的影响企业所得税的因素建立神经网络预测模型，其参数设置为误差精度 10^{-7}，学习次数 5000 次（数据量较小，过多容易过拟合），神经元个数为 Lasso 变量选择方法选择的变量个数 8。变量选择的指标的 2014 年及 2015 年数值可以通过 Python 建立灰色预测模型得出，后验差比值、预测精度等级见表 13-15。

表13-15 灰色预测模型企业所得税相关因素精度表

	x1	x2	x3	x4	x6	x7	x9	x10
后验差比值	0.0696	0.0179	0.0743	0.2294	0.3182	0.0719	0.2904	0.1038
预测精度等级	好	好	好	好	好	好	好	好

企业所得税灰色预测如代码清单 13-13 所示。

代码清单13-13 企业所得税灰色预测

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数

inputfile = '../data/data4.csv' #输入的数据文件
outputfile = '../tmp/data4_GM11.xls' #灰色预测后保存的路径
data = pd.read_csv(inputfile) #读取数据
data.index = range(2002, 2014)

data.loc[2014] = None
data.loc[2015] = None
l = ['x1', 'x2', 'x3', 'x4', 'x6', 'x7', 'x9', 'x10']
for i in l:
  f = GM11(data[i][range(2002, 2014)].as_matrix())[0]
  data[i][2014] = f(len(data)-1) #2014年预测结果
  data[i][2015] = f(len(data)) #2015年预测结果
  data[i] = data[i].round(2) #保留两位小数

data[l+['y']].to_excel(outputfile) #结果输出
```

代码详见：demo/code/ 4-huise.py

将数据零均值标准化后，代入企业所得税所建立的 3 层神经网络预测模型（输入层 8 节点，隐藏层 6 节点，输出层 1 节点），得到企业所得税的 2015 年预测值为 1 608 361 万元，相关数据见表 13-16，其中红色字体的数据为预测数据。图 13-5 为神经网络企业所得税真实值与预测值对比图。

表13-16　企业所得税及其相关因素历史数据和预测表

	x1	x2	x3	x4	x6	x7	x9	x10	y	预测值
2002	12 113 416	18 895 479	10 092 421	559.6	31.99	3 733 922	1 053 156	2 690 984	236 416	232 359
2003	14 859 261	21 627 825	11 751 668	554.5	29.87	4 785 787	1 154 425	3 005 475	268 360	276 392
2004	17 880 638	25 453 413	13 489 283	566.1	30.69	5 459 314	1 434 440	3 384 477	326 556	328 473
2005	20 452 183	29 787 941	15 191 582	575.2	31.63	6 331 382	3 621 757	4 088 545	373 397	366 297
2006	24 415 160	35 118 425	16 963 824	582.1	28.95	6 870 406	4 196 301	4 767 231	455 820	457 024
2007	28 257 805	41 646 681	18 633 437	599	24.88	7 507 109	7 068 265	8 389 925	596 693	593 250
2008	32 278 717	48 903 250	21 055 373	633.1	30.85	8 754 491	17 829 885	8 431 405	756 412	760 217
2009	34 051 588	55 607 710	26 598 516	612.8	23.16	10 134 050	17 019 222	11 076 649	732 282	747 795
2010	40 022 658	65 574 525	32 635 731	632.4	20.42	12 805 288	26 192 835	13 991 612	935 248	927 175
2011	45 769 763	76 419 207	34 122 005	664.7	22.55	15 613 171	21 639 131	15 351 387	1 061 594	1 040 909
2012	47 206 504	86 167 948	37 583 868	677.3	20.9	17 417 072	21 396 742	15 796 804	1 075 045	1 075 636
2013	52 273 431	99 643 373	44 545 508	680.7	19.7	21 828 895	22 659 148	20 881 374	1 155 923	1 169 546
2014	61 323 698	117 154 722	50 851 081	699.8	19.29	24 171 631	38 486 973	25 968 500		1 446 997
2015	68 835 983	135 969 753	58 208 842	715.1	18.4	28 436 657	45 686 095	31 069 406		1 608 361

数据详见：demo/data/enterprise_income.xls

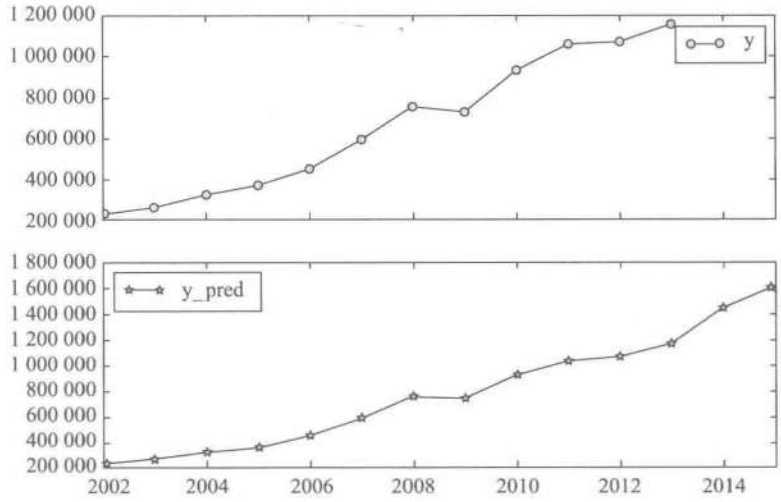


图 13-5　企业所得税真实值与预测值对比图

企业所得税神经网络预测模型如代码清单 13-14 所示。

代码清单13-14　企业所得税神经网络预测模型

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../tmp/data4_GM11.xls' #灰色预测后保存的路径
```

```
outputfile = '../data/enterprise_income.xls' #神经网络预测后保存的结果
modelfile = '../tmp/4-net.model' #模型保存路径
data = pd.read_excel(inputfile) #读取数据
feature = ['x1', 'x2', 'x3', 'x4', 'x6', 'x7', 'x9', 'x10'] #特征所在列

data_train = data.loc[range(2002,2014)].copy() #取2014年前的数据建模
data_mean = data_train.mean()
data_std = data_train.std()
data_train = (data_train - data_mean)/data_std #数据标准化
x_train = data_train[feature].as_matrix() #特征数据
y_train = data_train['y'].as_matrix() #标签数据

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(8, 6))
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(6, 1))
model.compile(loss='mean_squared_error', optimizer='adam') #编译模型
model.fit(x_train, y_train, nb_epoch = 5000, batch_size = 16) #训练模型，学习五千次
model.save_weights(modelfile) #保存模型参数

#预测，并还原结果。
x = ((data[feature] - data_mean[feature])/data_std[feature]).as_matrix()
data[u'y_pred'] = model.predict(x) * data_std['y'] + data_mean['y']
data[u'y_pred'] = data[u'y_pred'].round()
data.to_excel(outputfile)

import matplotlib.pyplot as plt #画出预测结果图
p = data[['y','y_pred']].plot(subplots = True, style=['b-o','r-*'])
plt.show()
```

代码详见：demo/code/ 4-yuce.py

（5）个人所得税预测模型

利用 Adaptive-Lasso 方法进行个人所得税（y）影响因素的变量选择，通过表 13-17 可以看出，城市居民年人均可支配收入（x1）、地区生产总值（x4）、第二产业增加值（x5）和地方财政收入（x7）这 4 个因素进入选择，其他因素的系数为 0。

表13-17 系数表

x1	x2	x3	x4	x5	x6	x7
16.983 45	0.000 00	0.000 00	-0.010 35	0.017 59	0.000 00	0.024 17

Adaptive-Lasso 变量选择如代码清单 13-15 所示。

代码清单13-15 Adaptive-Lasso变量选择

```
#-*- coding: utf-8 -*-
```

```
import pandas as pd
inputfile = '../data/data5.csv' #输入的数据文件
data = pd.read_csv(inputfile) #读取数据

#导入AdaptiveLasso算法，要在较新的Scikit-Learn才有。
from sklearn.linear_model import AdaptiveLasso
model = AdaptiveLasso(gamma=1)
model.fit(data.iloc[:,0:7],data['y'])
model.coef_ #各个特征的系数
```

代码详见：demo/code/ 5-adaptive-lasso.py

对 Adaptive-Lasso 变量选择方法识别的影响个人所得税的因素建立神经网络预测模型，其参数设置为误差精度 10^{-7}，学习次数 15 000 次，神经元个数为 Lasso 变量选择方法选择的变量个数 4。变量选择的指标的 2014 年及 2015 年数值可以通过 Python 建立灰色预测模型得出，后验差比值、预测精度等级见表 13-18。

表13-18　灰色预测模型个人所得税相关因素精度表

	x1	x4	x5	x7
后验差比值	0.0747	0.0349	0.0696	0.1038
预测精度等级	好	好	好	好

代码清单13-16　个人所得税灰色预测

```
#-*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数

inputfile = '../data/data5.csv' #输入的数据文件
outputfile = '../tmp/data5_GM11.xls' #灰色预测后保存的路径
data = pd.read_csv(inputfile) #读取数据
data.index = range(2000, 2014)

data.loc[2014] = None
data.loc[2015] = None
l = ['x1', 'x4', 'x5', 'x7']
for i in l:
  f = GM11(data[i][range(2000, 2014)].as_matrix())[0]
  data[i][2014] = f(len(data)-1) #2014年预测结果
  data[i][2015] = f(len(data)) #2015年预测结果
  data[i] = data[i].round() #取整

data[l+['y']].to_excel(outputfile) #结果输出
```

代码详见：demo/code/ 5-huise.py

将数据零均值标准化后，代入个人所得税所建立的 3 层神经网络预测模型（输入层 4 节点，隐藏层 8 节点，输出层 1 节点），得到个人所得税的 2015 年预测值为 400 352.4 万元，相

关数据见表 13-19，其中红色字体的数据为预测数据。图 13-2 为神经网络个人所得税真实值与预测值对比图。

表13-19 个人所得税及其相关因素历史数据和预测表

	x1	x4	x5	x7	y	预 测 值
2000	13 967	24 927 434	10 216 241	2 199 077	185 625	185 492
2001	14 694	28 416 511	11 122 943	2 719 058	254 892	254 890
2002	13 380	32 039 616	12 113 416	2 690 984	159 684	159 605
2003	15 003	37 586 166	14 859 261	3 005 475	153 080	153 223
2004	16 884	44 505 503	17 880 638	3 384 477	167 379	167 250
2005	18 287	51 542 283	20 452 183	4 088 545	198 017	196 685
2006	19 851	60 818 614	24 415 160	4 767 231	231 794	232 000
2007	22 469	71 403 223	28 257 805	8 389 925	295 316	296 298
2008	25 317	82 873 816	32 278 717	8 431 405	353 372	353 621
2009	27 610	91 382 135	34 051 588	11 076 649	389 824	391 391
2010	30 658	107 482 828	40 022 658	13 991 612	472 154	470 520
2011	34 438	124 234 390	45 769 763	15 351 387	462 098	462 154
2012	38 054	135 512 072	47 206 504	15 796 804	439 592	439 533
2013	42 049	154 201 434	52 273 431	20 881 374	489 777	489 827
2014	46 004	182 800 609	62 913 076	26 017 778		413 707
2015	50 884	209 686 804	70 996 676	31 143 677		447 977

数据详见：demo/data/personal_Income.xls

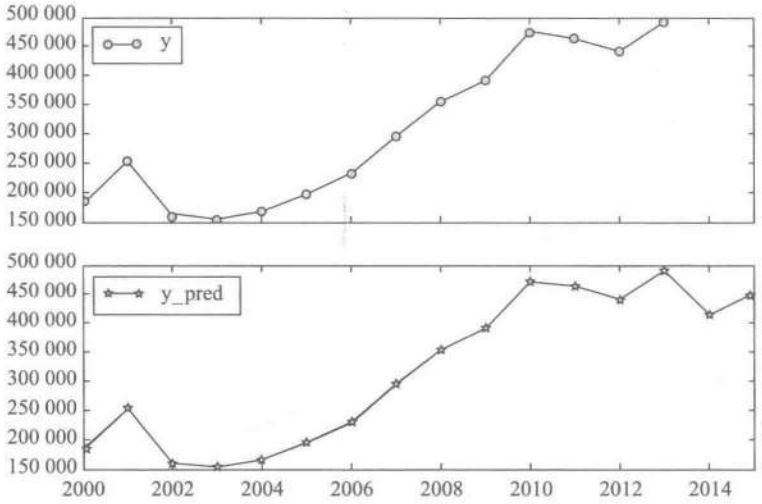


图 13-6 个人所得税真实值与预测值对比图

个人所得税神经网络预测模型如代码清单 13-17 所示。

代码清单13-17　个人所得税神经网络预测模型

```
#-*- coding: utf-8 -*-
import pandas as pd
inputfile = '../tmp/data5_GM11.xls' #灰色预测后保存的路径
outputfile = '../data/personal_Income.xls' #神经网络预测后保存的结果
modelfile = '../tmp/5-net.model' #模型保存路径
data = pd.read_excel(inputfile) #读取数据
feature = ['x1', 'x4', 'x5', 'x7'] #特征所在列

data_train = data.loc[range(2000,2014)].copy() #取2014年前的数据建模
data_mean = data_train.mean()
data_std = data_train.std()
data_train = (data_train - data_mean)/data_std #数据标准化
x_train = data_train[feature].as_matrix() #特征数据
y_train = data_train['y'].as_matrix() #标签数据

from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential() #建立模型
model.add(Dense(4, 8))
model.add(Activation('relu')) #用relu函数作为激活函数，能够大幅提供准确度
model.add(Dense(8, 1))
model.compile(loss='mean_squared_error', optimizer='adam') #编译模型
model.fit(x_train, y_train, nb_epoch = 15000, batch_size = 16) #训练模型，学习
    一万五千次
model.save_weights(modelfile) #保存模型参数

#预测，并还原结果。
x = ((data[feature] - data_mean[feature])/data_std[feature]).as_matrix()
data[u'y_pred'] = model.predict(x) * data_std['y'] + data_mean['y']
data[u'y_pred'] = data[u'y_pred'].round()
data.to_excel(outputfile)

import matplotlib.pyplot as plt #画出预测结果图
p = data[['y','y_pred']].plot(subplots = True, style=['b-o','r-*'])
plt.show()
```

代码详见：demo/code/ 5-yuce.py

（6）政府性基金收入预测模型

相比于 2006 年及以往年份，2007 年的该市土地出让金大幅上涨，而土地出让金收入的大幅上涨直接影响了政府性基金收入。所以，为了数据的连续性，本例利用灰色预测法对 2007 年至 2013 年的政府性基金收入进行预测，灰色预测的后验差比值为 0.2390，小于 0.35，预测精度为好。

将数值代入计算，即可得到 2014 年政府性基金收入为 10 387 002.56 万元，2015 年政府性基金收入为 12 929 795.07 万元，预测对比图如图 13-7 所示。

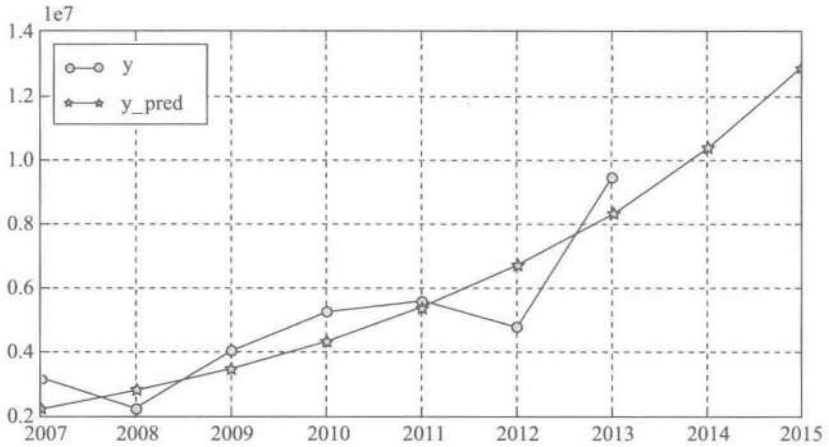


图 13-7 灰色预测政府性基金收入真实值与预测值对比图

政府性基金收入灰色预测如代码清单 13-18 所示。

代码清单13-18 政府性基金收入灰色预测

```
#-*- coding: utf-8 -*-
from __future__ import print_function
import numpy as np
import pandas as pd
from GM11 import GM11 #引入自己编写的灰色预测函数

x0 = np.array([3152063, 2213050, 4050122, 5265142, 5556619, 4772843, 9463330])
f, a, b, x00, C, P = GM11(x0)
print(u'2014年、2015年的预测结果分别为: \n%0.2f万元和%0.2f万元' %(f(8), f(9)))
print(u'后验差比值为: %0.4f' %C)
p = pd.DataFrame(x0, columns = ['y'], index = range(2007, 2014))
p.loc[2014] = None
p.loc[2015] = None
p['y_pred'] = [f(i) for i in range(1,10)]
p['y_pred'] = p['y_pred'].round(2)
p.index = pd.to_datetime(p.index, format='%Y')

import matplotlib.pylab as plt
p.plot(style=['b-o','r-*'], xticks = p.index)
plt.show()
```

代码详见：demo/code/ 6-huise.py

13.3 上机实验

1. 实验目的

掌握 Adaptive-lasso 变量选择和神经网络预测模型。

2. 实验内容

- 对搜集的某市地方财政收入以及各类别收入数据进行分析，识别影响地方财政收入的关键特征，数据见“test/data/data1—data5.csv”。使用 Adaptive-lasso 变量选择方法筛选出地方财政收入以及各类别收入的关键影响因素。
- 用 GM(1, 1) 灰色预测方法得到筛选出的关键影响因素的 2014 年、2015 年的预测值。
- 代入用历史数据训练的神经网络模型，从而得到某市地方财政收入以及各类别收入 2014 年、2015 年的预测值。

3. 实验方法与步骤

1）使用 read_csv() 函数把经过预处理的“test/data/data(1—5).csv”数据读入当前工作空间。

2）使用 adaptive-lasso 函数“test/code/ (1—5)-adaptive-lasso.py”对预处理的数据“data(1—5).csv”进行变量选择。

3）使用 GM(1,1) 灰色预测方法“test/code/ (1—5)-huise.py”得到筛选出的关键影响因素的 2014 年、2015 年的预测值。

4）使用神经网络对某市地方财政收入以及各类别收入进行预测。

4. 思考与实验总结

1）尝试其他的变量选择方法。

2）尝试采用岭回归方法进行变量选择与 lasso 进行比较，尝试使用更复杂的神经网络模型进行预测。

13.4　拓展思考

由于电力工业与其他的产业不同，其产品（即电能）无法大量储存，电力的生产和消费必须在同一瞬间进行，因此电力负荷预测成为电力系统运行调度、生产规划、电力市场竞价决策的重要组成部分。做好电力负荷预测管理工作可以有效降低电网公司运行成本并提高电力设备运行效率，其预测精度不但可以保证电网安全可靠供电，而且直接影响到电网经营企业的生产经营决策及经营效益。随着电力改革的深化、电力市场的开放，进一步提高短期负荷预测管理水平愈加显得重要和迫切。电网结构示意图如图 13-8 所示。

目前，国内外的预测应用软件大多基于特定的少数几种模型，而选择模型单一造成的后果是：预测结果往往只对某种发展规律有效，当事物发展规律改变时，如果仍然采用原有的单一模型，就会造成预测结果偏差过大，从而失去了预测的实际意义。尤其是对于使用系统的各个供电公司，由于发展水平不同，用电结构不同，负荷特性差异很大，特定的某种预测方法很难在各地都发挥出良好的效果。另外，在电力市场环境下，影响电力发展的因素非常多，包括经济发展、能源消费、气象条件等众多影响因素，加之不同系统间数据共享性

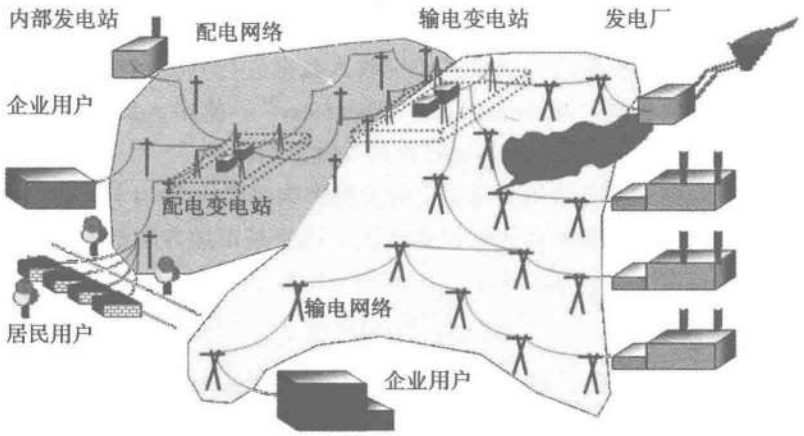


图 13-8 电网结构示意图

差，很难考虑相关因素的影响等，导致目前基于用电侧的电力负荷预测效果不甚理想。基于用电侧数据如何进行有效的电力负荷预测，为供电部门进行安全监视、预防性控制和紧急状态处理提供依据？

13.5 小结

本章结合某市地方财政收入以及各类别收入分析和预测的案例，重点介绍了数据挖掘算法中 Adaptive-Lasso 方法和神经网络算法在实际案例中的应用。重点研究影响某市地方财政收入的关键因素，并在这些关键影响因素的基础上采用神经网络算法对 2015 年各类别的收入进行预测，并详细地描述了数据挖掘的整个过程，也对相应的算法给出了 Python 上机实验。

第 14 章 Chapter 14

基于基站定位数据的商圈分析

14.1 背景与挖掘目标

随着个人手机终端的普及，出行群体中手机拥有率和使用率已达到相当高的比例，手机移动网络也基本实现了城乡空间区域的全覆盖。根据手机信号在真实地理空间上的覆盖情况，将手机用户时间序列的手机定位数据，映射至现实的地理空间位置，即可完整、客观地还原出手机用户的现实活动轨迹，从而挖掘得到人口空间分布与活动联系的特征信息。移动通信网络的信号覆盖逻辑上被设计成由若干六边形的基站小区相互邻接而构成的蜂窝网络面状服务区，如图 14-1 所示，手机终端总是与其中某一个基站小区保持联系，移动通信网络的控制中心会定期或不定期地主动或被动地记录每个手机终端时间序列的基站小区编号信息。

商圈是现代市场中企业市场活动的空间，最初是站在商品和服务提供者的产地角度提出来的，后来逐渐扩展到商圈，同时也是商品和服务享用者的区域。商圈划分的目的之一是为了研究潜在的顾客的分布以制定适宜的商业对策。

从某通信运营商提供的特定接口解析得到用户的定位数据，见表 14-1，定位数据各属性如表 14-2 所示。定位数据是以基站小区进行标识，利用基站小区的覆盖范围作为商圈区域的划分，那如何对用户的历史定位数据进行科学的分析，归纳出商圈的人流特征和规律，识别出不同类别的商圈，选择合适的区域进行运营商的促销活动？

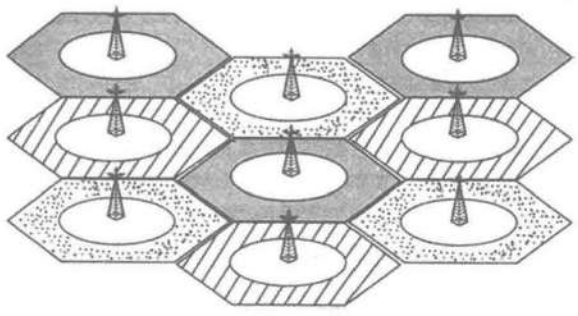


图 14-1 移动基站示意图

表14-1 某市某区域的定位数据示例

年	月	日	时	分	秒	毫秒	网络类型	LOC 编号	基站编号	EMASI 号	信令类型
2014	1	1	0	53	46	96	2	962947809921085	36902	55555	333789CA
2014	1	1	0	31	48	38	2	281335167708768	36908	55555	333333CA
2014	1	1	0	17	25	46	3	187655709192839	36911	55558	333477CA
2014	1	1	0	5	40	83	3	232648776184248	36908	55561	333381CA
2014	1	1	0	50	29	4	2	611763545227777	36906	55563	333405CA
2014	1	1	0	1	40	31	2	44710067012246	36909	55563	333717CA
2014	1	1	0	27	32	17	2	975579082112825	36912	55563	333981CA
2014	1	1	0	52	35	83	2	820798260690697	36906	55564	333861CA
2014	1	1	0	11	2	21	3	380420663155326	36910	55564	334149CA
2014	1	1	0	43	38	95	2	897743952380637	36903	55565	334053CA
2014	1	1	0	40	30	87	3	7775693027472	36910	55565	333453CA
2014	1	1	0	1	30	68	3	113404095624425	36911	55565	334125CA
2014	1	1	0	39	20	24	3	393808837659011	36905	55566	334077CA

表14-2 定位数据属性列表

序 号	属性编码	属性名称	数据类型	备 注
1	year	年	int	
2	month	月	int	
3	day	日	int	
4	hour	时	int	
5	minute	分	int	
6	second	秒	int	
7	millisecond	毫秒	int	
8	generation	网络类型	int	2 代表 2G，3 代表 3G，4 代表 4G
9	loc	LOC 编号	string	15 位字符串
10	cell_id	基站编号	string	基站 ID，15 位字符串
11	emasi	EMASI 号	string	需要关联用户表取用户号码（用户号码需要关联用户表得到用户 ID）
12	type	信令类型	string	小于 15 个字符

本次数据挖掘建模目标如下。

1）对用户的历史定位数据，采用数据挖掘技术，对基站进行分群。

2）对不同的商圈分群进行特征分析，比较不同商圈类别的价值，选择合适的区域进行运营商的促销活动。

14.2　分析方法与过程

手机用户在使用短信业务、通话业务、开关机、正常位置更新、周期位置更新和切入呼叫的时候均产生定位数据，定位数据记录手机用户所处基站的编号、时间和唯一标识用户的EMASI 号等。历史定位数据描绘了用户的活动模式，一个基站覆盖的区域可等价于商圈，通过归纳经过基站覆盖范围的人口特征，识别出不同类别的基站范围，即可等同地识别出不同类别的商圈。衡量区域的人口特征可从人流量和人均停留时间的角度进行分析，所以在归纳基站特征时可针对这两个特点进行提取。

由图 14-2 可知，基于移动基站定位数据的商圈分析主要包括以下步骤。

1）从移动通信运营商提供的特定接口上解析、处理、并滤除用户属性后得到用户定位数据。

2）以单个用户为例，进行数据探索分析，研究在不同基站的停留时间，并进一步地进行预处理，包括数据规约和数据变换。

3）利用步骤 2）形成的已完成数据预处理的建模数据，基于基站覆盖范围区域的人流特征进行商圈聚类，对各个商圈分群进行特征分析，选择合适的区域进行运营商的促销活动。

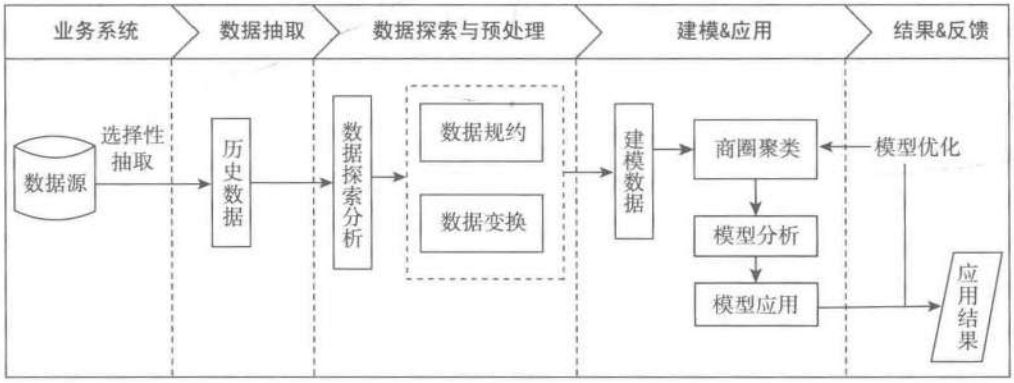


图 14-2　基于基站定位数据的商圈分析流程

14.2.1　数据抽取

从移动通信运营商提供的特定接口上解析、处理、并滤除用户属性后得到位置数据，以2014-1-1 为开始时间，2014-6-30 为结束时间作为分析的观测窗口，抽取观测窗口内某市某区域的定位数据形成建模数据，部分数据见表 14-1。

14.2.2　数据探索分析

为了便于观察数据，先提取 EMASI 号为“55555”的用户在 2014 年 1 月 1 日的定位数据，如表 14-3 所示，可以发现用户在 2014 年 1 月 1 日 00:31:48 处于 36908 基站的范围，

下一个记录是用户在2014年1月1日00:53:46处于36902基站的范围，这表明了用户从00:31:48到00:53:46都是处于36908基站，共停留了21分58秒，并且在00:53:46进入了36902基站的范围。再下一条记录是用户在2014年1月1日01:26:11处于36902基站的范围，这可能是由于用户在进行通话或者其他产生定位数据记录的业务，此时的基站编号未发生改变，用户依旧处于36902基站的范围，若要计算用户在36902基站范围停留的时间，则需要继续判断下一条记录，可以发现用户在2014年1月1日02:13:46处于36907基站的范围，故用户从00:53:46到02:13:46都是处于36902基站，共停留了80分。停留示意图如图14-3所示。

表14-3 EMASI号为“55555”的用户在2014年1月1日的位置数据

年	月	日	时	分	秒	毫秒	网络类型	LOC 编号	基站编号	EMASI 号	信令类型
2014	1	1	0	31	48	38	2	281335167708768	36908	55555	333333CA
2014	1	1	0	53	46	96	2	962947809921085	36902	55555	333789CA
2014	1	1	1	26	11	23	2	262095068434776	36902	55555	333334CA
2014	1	1	2	13	46	28	2	712890120478723	36907	55555	333551CA
2014	1	1	7	57	18	92	2	85044254500058	36902	55555	333796CA
2014	1	1	8	20	32	93	2	995208321887481	36903	55555	334109CA
2014	1	1	9	43	31	45	2	555114267094822	36908	55555	333798CA
2014	1	1	12	20	47	35	2	482996504023472	36907	55555	333393CA
2014	1	1	14	40	4	26	2	329606106134793	36903	55555	333587CA
2014	1	1	14	50	32	82	2	645164951070747	36908	55555	333731CA
2014	1	1	15	19	2	17	2	830855298094409	36902	55555	334068CA
2014	1	1	18	26	43	88	2	323108074844193	36912	55555	334023CA
2014	1	1	19	0	21	82	2	553245971859183	36909	55555	333952CA
2014	1	1	19	50	7	90	2	987606797101505	36906	55555	334096CA
2014	1	1	22	35	0	4	2	756416566337609	36908	55555	333427CA
2014	1	1	23	28	7	98	2	919108833174494	36904	55555	333500CA

图 14-3 停留示意图

14.2.3　数据预处理

1. 数据规约

原始数据的属性较多，但网络类型、LOC 编号和信令类型这 3 个属性对于挖掘目标没有用处，故剔除这 3 个冗余的属性。而衡量用户的停留时间，并不需要精确到毫秒级，故可把毫秒这一属性删除。

在计算用户的停留时间时，只计算两条记录的时间差，为了减少数据维度，把年、月和日合并记为日期，时、分和秒合并记为时间，则表 14-3 可处理得到表 14-4。

表14-4　数据规约后数据

日　期	时　间	基站编号	EMASI 号
2014 年 1 月 1 日	00:31:48	36908	55555
2014 年 1 月 1 日	00:53:46	36902	55555
2014 年 1 月 1 日	01:26:11	36902	55555
2014 年 1 月 1 日	02:13:46	36907	55555
2014 年 1 月 1 日	07:57:18	36902	55555
2014 年 1 月 1 日	08:20:32	36903	55555
2014 年 1 月 1 日	09:43:31	36908	55555
2014 年 1 月 1 日	12:20:47	36907	55555
2014 年 1 月 1 日	14:40:04	36903	55555
2014 年 1 月 1 日	14:50:32	36908	55555
2014 年 1 月 1 日	15:19:02	36902	55555
2014 年 1 月 1 日	18:26:43	36912	55555
2014 年 1 月 1 日	19:00:21	36909	55555
2014 年 1 月 1 日	19:50:07	36906	55555
2014 年 1 月 1 日	22:35:00	36908	55555
2014 年 1 月 1 日	23:28:07	36904	55555

2. 数据变换

挖掘的目标是寻找出高价值的商圈，需要根据用户的定位数据提取出衡量基站覆盖范围区域的人流特征，如人均停留时间和人流量等，高价值的商圈具有人流量大，人均停留时间长的特点，但是在写字楼工作的上班族在白天所处的基站范围基本固定，停留时间也相对较长，晚上的住宅区的居民所处的基站范围基本固定，停留时间也相对较长，仅通过停留时间作为人流特征难以区分高价值商圈和写字楼与住宅区，所以提取出来的人流特征必须能较为明显地区别这些基站范围。下面设计工作日上班时间人均停留时间、凌晨人均停留时间、周末人均停留时间和日均人流量作为基站覆盖范围区域的人流特征。

工作日上班时间人均停留时间是所有用户在工作日上班时间处在该基站范围内的平均时间，居民一般的上班工作时间是在 9:00 ～ 18:00，所以工作日上班时间人均停留时间是计算所有用户在工作日 9:00 ～ 18:00 处在该基站范围内的平均时间。

凌晨人均停留时间是指所有用户在 00:00 ～ 07:00 处在该基站范围内的平均时间，一般居民在 00:00 ～ 07:00 都是在住处休息，利用这个指标则可以表征出住宅区基站的人流特征。

周末人均停留时间是指所有用户周末处在该基站范围内的平均时间，高价值商圈在周末的逛街人数和时间都会大幅增加，利用这个指标则可以表征出高价值商圈的人流特征。

日均人流量指平均每天曾经在该基站范围内的人数，日均人流量大说明经过该基站区域的人数多，利用这个指标则可以表征出高价值商圈的人流特征。

这 4 个指标的计算直接从原始数据计算比较复杂，需先处理成中间过程数据，再从中计算出这 4 个指标。

中间过程数据的计算以单个用户在一天里的定位数据为基础，计算在各个基站范围下的工作日上班时间停留时间、凌晨停留时间、周末停留时间和是否处于基站范围。假设原始数据所有用户在观测窗口期间（L 天）曾经经过的基站有 N 个，用户有 M 个，用户 i 在 j 天经过的基站有 $num1$，$num2$ 和 $num3$，则用户 i 在 j 天在 $num1$ 基站的工作日上班时间停留时间为 $weekday\_num1_{ij}$，在 $num2$ 基站的工作日上班时间停留时间为 $weekday\_num2_{ij}$，在 $num3$ 基站的工作日上班时间停留时间为 $weekday\_num3_{ij}$；在 $num1$ 基站的凌晨停留时间为 $night\_num1_{ij}$，在 $num2$ 基站的凌晨停留时间为 $night\_num2_{ij}$，在 $num3$ 基站的凌晨停留时间为 $night\_num3_{ij}$；在 $num1$ 基站的周末停留时间为 $weekend\_num1_{ij}$，在 $num2$ 基站的周末停留时间为 $weekend\_num2_{ij}$，在 $num3$ 基站的周末停留时间为 $weekend\_num3_{ij}$；在 $num1$ 基站是否停留为 $stay\_num1_{ij}$，在 $num2$ 基站是否停留为 $stay\_num2_{ij}$，在 $num3$ 基站是否停留为 $stay\_num3_{ij}$，其中 $stay\_num1_{ij}$、$stay\_num2_{ij}$ 和 $stay\_num3_{ij}$ 的值均为 1；对于未停留的其他基站，工作日上班时间停留时间、凌晨停留时间、周末停留时间和是否处于基站范围的值均为 0。

对于 $num1$ 基站，4 个基站覆盖范围区域的人流特征的计算公式如下。

- 工作日上班时间人均停留时间：$weekday_{num1} = \frac{1}{LM}\sum_{j=1}^{L}\sum_{i=1}^{M} weekday\_num1_{ij}$。
- 凌晨人均停留时间：$night_{num1} = \frac{1}{LM}\sum_{j=1}^{L}\sum_{i=1}^{M} night\_num1_{ij}$。
- 周末人均停留时间：$weekend_{num1} = \frac{1}{LM}\sum_{j=1}^{L}\sum_{i=1}^{M} weekend\_num1_{ij}$。
- 日均人流量：$stay_{num1} = \frac{1}{L}\sum_{j=1}^{L}\sum_{i=1}^{M} stay\_num1_{ij}$。

对于其他基站，计算公式一致。

对采集到的数据，按基站覆盖范围区域的人流特征进行计算，得到各个基站的样本数据，见表 14-5。

表14-5　样本数据

基站编号	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
36902	78	521	602	2863
36903	144	600	521	2245
36904	95	457	468	1283
36905	69	596	695	1054
36906	190	527	691	2051
36907	101	403	470	2487
36908	146	413	435	2571
36909	123	572	633	1897
36910	115	575	667	933
36911	94	476	658	2352
36912	175	438	477	861
35138	176	477	491	2346
37337	106	478	688	1338
36181	160	493	533	2086
38231	164	567	539	2455
38015	96	538	636	960
38953	40	469	497	1059
35390	97	429	435	2741
36453	95	482	479	1913
36855	159	554	480	2515

数据详见：示例程序 /data/business_circle.xls

由于各个属性之间的差异较大，为了消除数量级数据带来的影响，在进行聚类前，需要进行离差标准化处理，离差标准化处理的 Python 代码如下代码清单 14-1，离差后的数据文件存储在当前目录的 standardized.xls 文件中。

代码清单14-1　离差标准化

```
#-*- coding: utf-8 -*-
#数据标准化到[0,1]
import pandas as pd

#参数初始化
filename = '../data/business_circle.xls' #原始数据文件
standardizedfile = '../tmp/standardized.xls' #标准化后数据保存路径
```

```
data = pd.read_excel(filename, index_col = u'基站编号') #读取数据

data = (data - data.min())/(data.max() - data.min()) #离差标准化
data = data.reset_index()

data.to_excel(standardizedfile, index = False) #保存结果
```

代码详见：示例程序 /code/standardization.py

标准化后的样本数据见表 14-6。

表14-6 标准化后样本数据

基站编号	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
36902	0.103 865	0.856 364	0.850 539	0.169 153
36903	0.263 285	1	0.725 732	0.118 21
36904	0.144 928	0.74	0.644 068	0.038 909
36905	0.082 126	0.992 727	0.993 837	0.020 031
36906	0.374 396	0.867 273	0.987 673	0.102 217
36907	0.159 42	0.641 818	0.647 149	0.138 158
36908	0.268 116	0.66	0.593 22	0.145 083
36909	0.212 56	0.949 091	0.898 305	0.089 523
36910	0.193 237	0.954 545	0.950 693	0.010 057
36911	0.142 512	0.774545	0.936 826	0.127 03
36912	0.338 164	0.705 455	0.657 935	0.004 122
35138	0.340 58	0.776 364	0.679 507	0.126 535
37337	0.171 498	0.778 182	0.983 051	0.043 442
36181	0.301 932	0.805 455	0.744 222	0.105 103
38231	0.311 594	0.94	0.753 467	0.135 521
38015	0.147 343	0.887 273	0.902 928	0.012 283
38953	0.012 077	0.761 818	0.688 752	0.020 443
35390	0.149 758	0.689 091	0.593 22	0.159 097
36453	0.144 928	0.785 455	0.661 017	0.090 842
36855	0.299 517	0.916 364	0.662 558	0.140 467

数据详见：示例程序 /data/standardized.xls

14.2.4 模型构建

1. 构建商圈聚类模型

数据经过预处理过后，形成建模数据。采用层次聚类算法对建模数据进行基于基站数据

的商圈聚类，画出谱系聚类图，Python 代码如代码清单 14-2 所示，输入数据集为离差标准化后的数据。

代码清单14-2　谱系聚类图

```
#-*- coding: utf-8 -*-
#谱系聚类图
import pandas as pd

#参数初始化
standardizedfile = '../data/standardized.xls' #标准化后的数据文件
data = pd.read_excel(standardizedfile, index_col = u'基站编号') #读取数据

import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage,dendrogram
#这里使用scipy的层次聚类函数

Z = linkage(data, method = 'ward', metric = 'euclidean') #谱系聚类图
P = dendrogram(Z, 0) #画谱系聚类图
plt.show()
```

代码详见：示例程序 /code/hierarchical_clustering_pic.m

根据代码清单 14-2，可以得到的谱系聚类图，如图 14-4 所示。

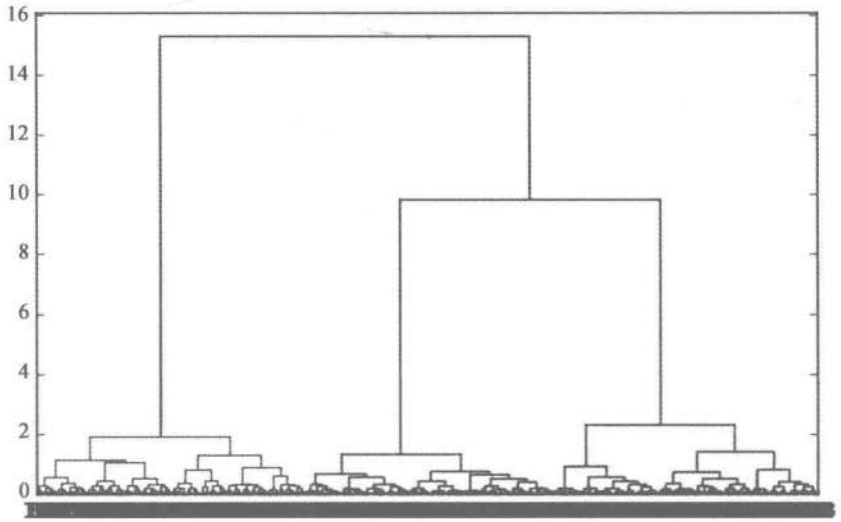


图 14-4　谱系聚类图

从图 14-5 可以看出，可把聚类类别数取 3 类，Python 代码中取聚类类别数为 k = 3，输出结果 typeindex 为每个样本对应的类别号。层次聚类算法详见代码清单 14-3。

代码清单14-3　层次聚类算法

```
#-*- coding: utf-8 -*-
```

```
#层次聚类算法
import pandas as pd

#参数初始化
standardizedfile = '../data/standardized.xls' #标准化后的数据文件
k = 3 #聚类数
data = pd.read_excel(standardizedfile, index_col = u'基站编号') #读取数据

from sklearn.cluster import AgglomerativeClustering #导入sklearn的层次聚类函数
model = AgglomerativeClustering(n_clusters = k, linkage = 'ward')
model.fit(data) #训练模型

#详细输出原始数据及其类别
r = pd.concat([data, pd.Series(model.labels_, index = data.index)], axis = 1)  #
详细输出每个样本对应的类别
r.columns = list(data.columns) + [u'聚类类别'] #重命名表头

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号

style = ['ro-', 'go-', 'bo-']
xlabels = [u'工作日人均停留时间', u'凌晨人均停留时间', u'周末人均停留时间', u'日均人流量']
pic_output = '../tmp/type_' #聚类图文件名前缀

for i in range(k): #逐一作图，作出不同样式
  plt.figure()
  tmp = r[r[u'聚类类别'] == i].iloc[:,:4] #提取每一类
  for j in range(len(tmp)):
    plt.plot(range(1, 5), tmp.iloc[j], style[i])

  plt.xticks(range(1, 5), xlabels, rotation = 20) #坐标标签
  plt.subplots_adjust(bottom=0.15) #调整底部
  plt.savefig(u'%s%s.png' %(pic_output, i)) #保存图片
```

代码详见：示例程序 /code/hierarchical_clustering.py

2. 模型分析

针对聚类结果按不同类别画出 4 个特征的折线图，如图 14-5、图 14-6 和图 14-7 所示。对于商圈类别 3，这部分基站覆盖范围的工作日上班时间人均停留时间较长，同时凌晨人均停留时间、周末人均停留时间相对较短，该类别基站覆盖的区域类似于白领上班族的工作区域。对于商圈类别 1，日均人流量较大，同时工作日上班时间人均停留时间、凌晨人均停留时间和周末人均停留时间相对较短，该类别基站覆盖的区域类似于商业区。对于商圈类别 2，凌晨人均停留时间和周末人均停留时间相对较长，而工作日上班时间人均停留时间较短，日均人流量较少，该类别基站覆盖的区域类似于住宅区。

商圈类别 2 的人流量较少，商圈类别 3 的人流量一般，而且白领上班族的工作区域一般的人员流动集中在上、下班时间和午间吃饭时间，这两类商圈均不利于运营商的促销活动的

开展，商圈类别 1 的人流量大，在这样的商业区有利于进行运营商的促销活动。

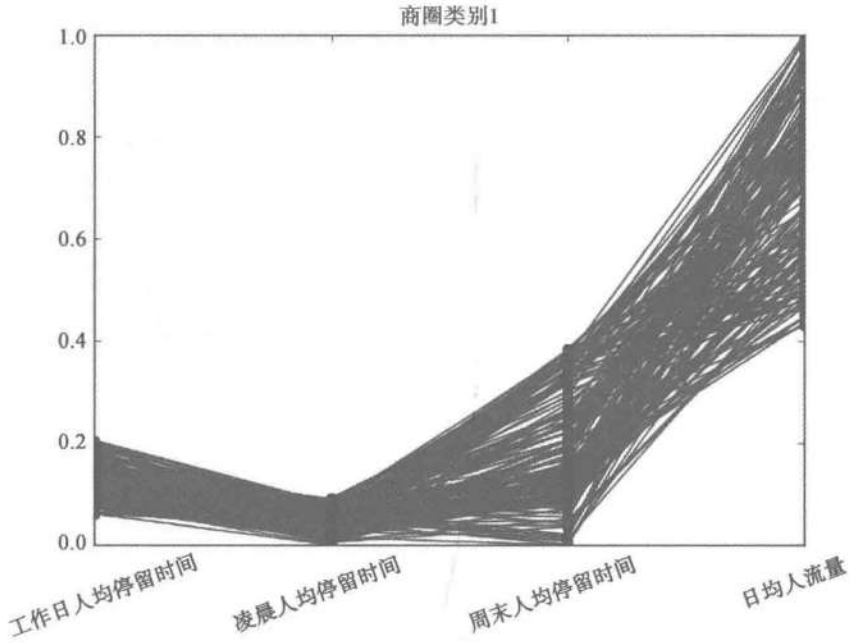


图 14-5　商圈类别 1 折线图

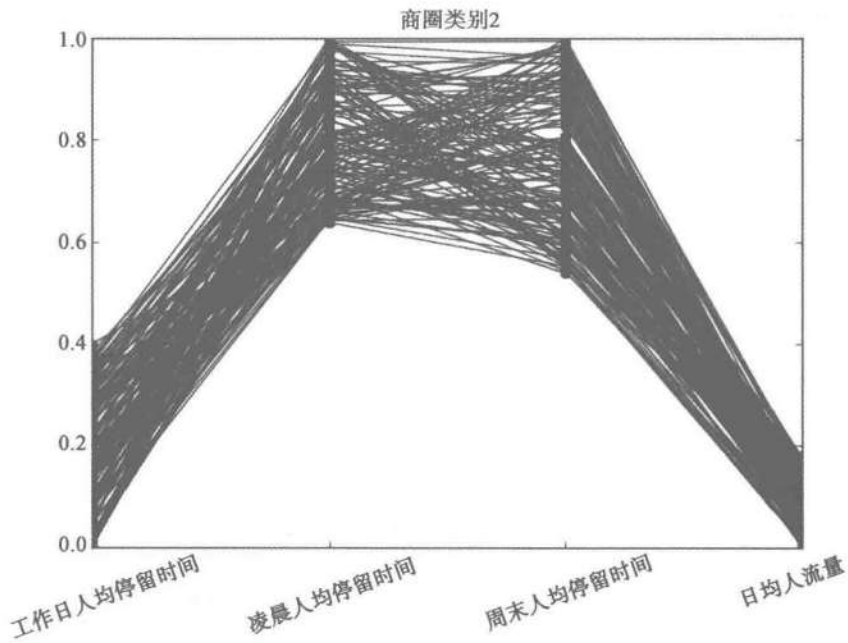


图 14-6　商圈类别 2 折线图

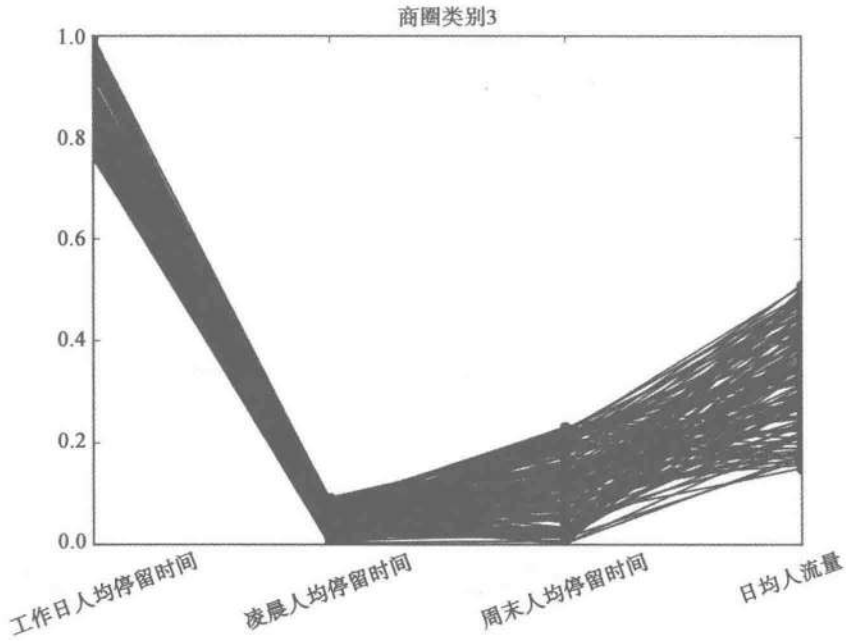


图 14-7 商圈类别 3 折线图

14.3 上机实验

1. 实验目的

掌握离差标准化进行数据预处理和层次聚类算法。

2. 实验内容

对采集到的数据，按基站覆盖范围区域的人流特征进行计算，得到各个基站的样本数据，处理好的数据见“test/data/business_circle.xls”，需要对各个基站进行商圈聚类。但为了避免单个特征的值过大影响聚类效果，需要对数据先进行离差标准化，再采用层次聚类实现商圈聚类，并分析聚类结果。

3. 实验方法与步骤

1）把原始数据，即表 14-5 的数据读取到 Python 工作空间。根据业务需求只需截取后面 4 列的数据进行标准化即可。

2）对原始数据进行离差标准化，需要设置离散化区间为 [0，1]，同时考虑是否可以使用其他标准化方式。

3）构建层次聚类模型。比较 scipy.cluster 和 sklearn.cluster 两个子库的联系，在 Scipy 中使用 linkage 函数构建谱系聚类图，method 参数设置为“ward”，metric 参数设置为“euclidean”。

4）使用 Scikit-Learn 中的 AgglomerativeClustering() 函数对构建好的谱系聚类图进行分

类，通过 n_clusters 参数指定需要分类的类别数为 3。

5）使用 scipy.cluster 的 dendrogram() 函数对构建的谱系聚类图可视化，即画出其谱系聚类图并保存；针对每个群组使用 Matplotlib 画其趋势图并保存。

4. 思考与实验总结

1）数据标准化的方法有哪些？这里为什么使用离差标准化？

2）构建层次聚类模型时，可以调节哪些参数，对模型有何影响？

14.4　拓展思考

轨迹挖掘可以定义为从移动定位数据中提取隐含的、人们预先不知道的、但又潜在有用的移动轨迹模式的过程。轨迹挖掘可应用到多个重要领域，如社交网络、公共安全、智能交通管理、城市规划与发展等。面向拼车推荐应用是轨迹挖掘的新兴研究主题。拼车是指相同路线的人乘坐同一辆车上下班、上学及放学回家、节假日出游等，车费由乘客平均分摊。拼车不仅能节省出行费用，而且有利于缓解城市交通。现在大部分的拼车网站的普遍做法仍然是通过拼车司机在拼车服务网站上发布出发地、目的地、出发时间等信息，再由拼车客户在网站上输入出发地和目的地来搜索符合自己情况的拼车对象。这在很大程度上浪费了拼车用户在网上搜索拼车伙伴的时间，使用户的拼车体验变差。而面向拼车推荐应用是需要先对用户的定位数据进行轨迹挖掘，发现用户的轨迹模式集合，再根据两个用户之间移动轨迹模式的相似性，推荐合适的拼车路线。

14.5　小结

本章结合基于基站定位数据的商圈分析的案例，重点介绍了数据挖掘算法中层次聚类算法在实际案例中的应用。研究用户的定位数据，总结出人流特征，并采用层次聚类算法进行商圈聚类，识别出不同类别的商圈，最后选择合适的区域进行运营商的促销活动。案例详细地描述了数据挖掘的整个过程，也对其相应的算法给出了 Python 上机实验。

Chapter 15 第 15 章

电商产品评论数据情感分析

15.1 背景与挖掘目标

随着网上购物越来越流行，人们对于网上购物的需求变得越来越高，这让京东、淘宝等电商平台得到了很大的发展机遇。但是，这种需求也推动了更多的店商平台的崛起，引发了激烈的竞争。在这种电商平台激烈竞争的大背景下，除了提高商品质量、压低商品价格外，了解更多消费者的心声对于店商平台来说也变得越来越有必要，其中非常重要的方式就是对消费者的文本评论数据进行内在信息的数据挖掘分析。而得到的这些信息，也有利于对应商品的生产厂家自身竞争力的提升。

本文对京东平台上的热水器评论进行文本挖掘分析，本次数据挖掘建模目标如下。

1）分析某一品牌热水器的用户情感倾向。

2）从评论文本中挖掘出该品牌热水器的优点与不足。

3）提炼不同品牌热水器的卖点。

15.2 分析方法与过程

本次建模针对京东商城上“美的”品牌的热水器的消费者的文本评论数据，在对文本进行基本的机器预处理、中文分词、停用词过滤后，通过建立包括栈式自编码深度学习、语义网络与 LDA 主题模型等多种数据挖掘模型，实现对文本评论数据的倾向性判断以及所隐藏的信息的挖掘并分析，以期望得到有价值的内在内容。

图 15-1 为电商产品评论数据情感分析流程，主要包括以下步骤[⊖]。

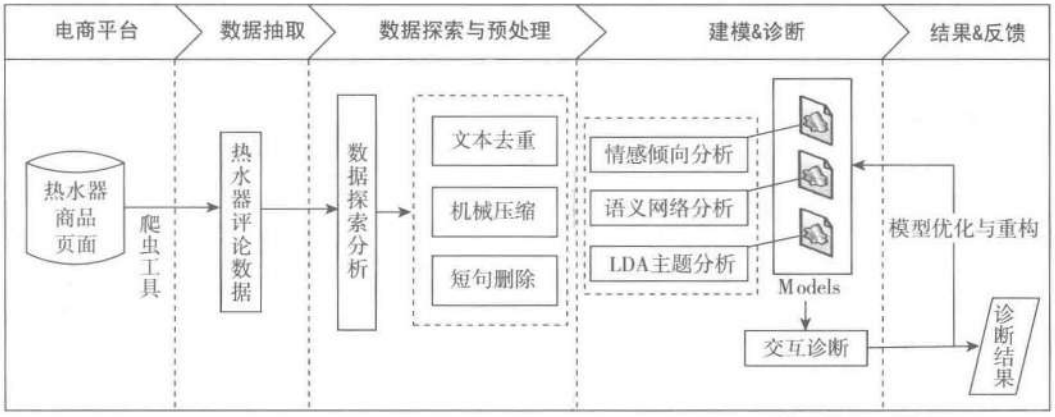


图 15-1　电商产品评论数据情感分析流程

1）利用爬虫工具——八爪鱼采集器，对京东商城进行热水器评论的数据采集。

2）对获取的数据进行基本的处理操作，包括数据预处理、中文分词、停用词过滤等操作。

3）文本评论数据经过处理后，运用多种手段对评论数据进行多方面的分析。

4）从对应结果的分析中获取文本评论数据中有价值的内容。

15.2.1　评论数据采集

要分析电商平台的热水器评论数据，需要先对评论数据进行采集，对比多种网络爬虫工具后，发现八爪鱼采集器属于“易用型”，它主要通过模仿用户的网页操作进行数据采集，只需指定数据采集逻辑和可视化选择采集的数据，即可完成采集规则的制定。因此，在案例的网页数据抓取工具选择的是八爪鱼采集器。

首先在八爪鱼采集器中新建任务，设置打开页面为“http://list.jd.com/list.html?cat=737%2C794%2C1706&ev=998_28702%40&page=1&JL=3_产品类型_电热水器”，页面如图 15-2 所示。

由于热水器下有多种产品，而且呈分页显示，所以抓取数据时需要制定翻页循环列表，再单击每个产品，进入产品的详细页面，如图 15-3 所示。

在本页面下需要抓取产品的名称，价格和评论信息。评论信息可见产品详细页面的下方，如图 15-4 所示，这里需要采集的有用户评论、评论时间、购买信息和用户名。同时，由于评论是多页显示，也需要制定翻页循环列表，循环抓取每页评论信息。

⊖ 周涛，吴家舜，邵悦涵 . 基于情感分析、语义网络和主题模型的评论文本分析 . 第三届泰迪杯全国大学生数据挖掘竞赛（http://www.tipdm.org）优秀作品。

图 15-2 热水器列表页面

图 15-3 产品的详细页面

图 15-4　产品评论

经过以上分析，可在八爪鱼采集器中设计出流程，如图 15-5 所示，进行单机采集后得到结果截图如图 15-6 所示。

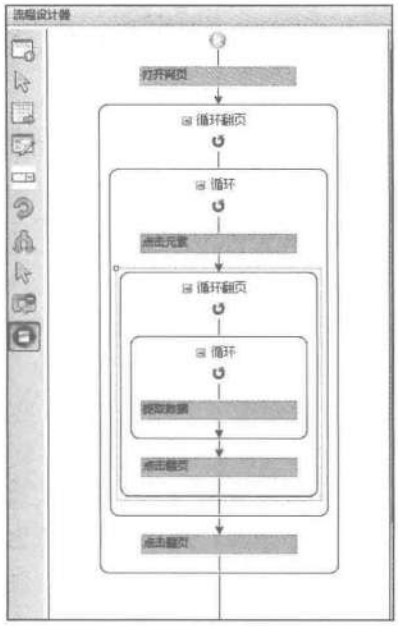


图 15-5　流程设计

商品名称	价格	累计评价数	好评度	中评度	差评度	买家印象	评价	评价时间	购买信息	用户名	购买时间
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器刚刚安装，你懂人很	2015-02-07 11:33	颜色：强排 B系列	s***5	2015-02-04 22:13 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器新屋用，东西不错...	2014-03-18 16:04	颜色：88-20	有***期	2014-03-15 14:49 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器热水器的保温效果不	2014-02-17 16:42	颜色：88-20	刘青CON	2013-12-28 14:54 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器外观大气，很上档次	2015-05-19 13:18	颜色：强排 B系列	g***1	2015-05-11 17:08 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器很好，挺喜欢的。万	2015-05-19 09:53	颜色：强排 B系列	j***8	2014-11-11 09:14 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器送货及时,快递员服务	2015-05-19 09:25	颜色：强排 B系列	j***k	2015-05-13 15:47 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器质量不错，值得推荐	2015-05-19 07:57	颜色：强排 B系列	我***主	2015-02-17 10:11 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器用着不错，性价比很	2015-05-19 07:04	颜色：强排 B系列	j***4	2015-01-24 13:26 购买
万和(Vanward) JSQ16-88-2	¥698.00	5589	(88%)	(7%)	(5%)		很实用(812)热水器有三个档的，还有个	2015-05-18 23:07	颜色：强排 B系列	j***a	2015-04-26 09:16 购买

图 15-6 评论采集结果

对采集到的评论数据进行处理，得到原始文本的评论数据，如表 15-1 所示。

表15-1 原始评论文本

	A	B	C	D	E	F	G	H	I	J	K	L
1	Id	已采	已发	电商平台	品牌	评论	时间	型号	PageUrl			
5900	1	TRUE	FALSE	京东	美的	京东商城信得过，买的放心，用的	2014-11-2	美的(Mide	http://s.club.jd.com/productpage			
5901	2	TRUE	FALSE	京东	美的	给公司宿舍买的，上门安装很快，	2014-11-1	美的(Mide	http://s.club.jd.com/productpage			
5902	3	TRUE	FALSE	京东	美的	美的值得信赖，质量不错	2014-09-0	美的(Mide	http://s.club.jd.com/productpage			
5903	4	TRUE	FALSE	京东	美的	不错不错的哦，第一次在京东买这	2014-11-1	美的(Mide	http://s.club.jd.com/productpage			
5904	5	TRUE	FALSE	京东	美的	很满意，水方一晚上都还是热的早	2014-11-1	美的(Mide	http://s.club.jd.com/productpage			
5905	6	TRUE	FALSE	京东	美的	自己动手安装的，买材料发了不到	2014-11-2	美的(Mide	http://s.club.jd.com/productpage			
5906	7	TRUE	FALSE	京东	美的	几套出租房一直用这款。	2014-09-2	美的(Mide	http://s.club.jd.com/productpage			
5907	8	TRUE	FALSE	京东	美的	还不错，就是快递有点慢，不打电	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5908	9	TRUE	FALSE	京东	美的	东西很不错 双十一抢的 物美价廉	2014-11-1	美的(Mide	http://s.club.jd.com/productpage			
5909	10	TRUE	FALSE	京东	美的	性价比高！下次还会光顾的！	2014-11-1	美的(Mide	http://s.club.jd.com/productpage			
5910	11	TRUE	FALSE	京东	美的	前天晚上定货，第二天早上就送货	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5911	12	TRUE	FALSE	京东	美的	还好吧	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5912	13	TRUE	FALSE	京东	美的	应该值得信任的品牌。。。。。。	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5913	14	TRUE	FALSE	京东	美的	价格便宜，购物方便快捷	2014-11-2	美的(Mide	http://s.club.jd.com/productpage			
5914	15	TRUE	FALSE	京东	美的	很好很好很好很好很好很好很好很	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5915	16	TRUE	FALSE	京东	美的	的(Midea) F40-15A1 40升 电热水	2014-11-2	美的(Mide	http://s.club.jd.com/productpage			
5916	17	TRUE	FALSE	京东	美的	帮同事买的他说不错，送货到家！	2014-11-2	美的(Mide	http://s.club.jd.com/productpage			
5917	18	TRUE	FALSE	京东	美的	用了一段时间了，好用，没什么问	2014-09-2	美的(Mide	http://s.club.jd.com/productpage			
5918	19	TRUE	FALSE	京东	美的	怎么这样，前天买的，今天到货，	2014-11-3	美的(Mide	http://s.club.jd.com/productpage			
5919	20	TRUE	FALSE	京东	美的	很好用，很方便！第二次购买了&h	2014-12-0	美的(Mide	http://s.club.jd.com/productpage			
5920	21	TRUE	FALSE	京东	美的	给公司买的，就是方便而已	2014-11-0	美的(Mide	http://s.club.jd.com/productpage			
5921	22	TRUE	FALSE	京东	美的	2个人洗澡的水还可以，再多就最好	2014-09-2	美的(Mide	http://s.club.jd.com/productpage			

数据详见：01- 示例数据 / 汇总 - 京东 .xlsx

再将品牌为“美的”的“评论”一列抽取，另存为“\data\meidi_jd.txt”，编码为 UTF-8。评论抽取的代码如代码清单 15-1 所示。

代码清单15-1 评论抽取代码

```
#-*- coding: utf-8 -*-
import pandas as pd

inputfile = '../data/huizong.csv' #评论汇总文件
outputfile = '../data/meidi_jd.txt' #评论提取后保存路径
data = pd.read_csv(inputfile, encoding = 'utf-8')
data = data[[u'评论']][data[u'品牌'] == u'美的']
data.to_csv(outputfile, index = False, header = False)
```

代码详见：demo/code/excel2txt.py

15.2.2 评论预处理

取到文本后，首先要进行文本评论数据的预处理。文本评论数据里存在大量价值含量很

低甚至没有价值含量的条目，如果将这些评论数据也引入进行分词、词频统计乃至情感分析等，必然会对分析造成很大的影响，得到的结果的质量也必然是存在问题的。那么，在利用到这些文本评论数据之前就必须先进行文本预处理，把大量的此类无价值含量的评论去除。

文本评论数据的预处理主要由 3 个部分组成：文本去重、机械压缩去词以及短句删除。

1. 文本去重

（1）文本去重的基本解释及原因

文本去重，顾名思义，就是去除文本评论数据中重复的部分。无论获取到什么样的文本评论数据，首先要进行的预处理都应当是文本去重。文本去重的主要原因如下。

1）一些电商平台为了避免一些客户长时间不进行评论，往往会设置一道程序，如果用户超过规定的时间仍然没有做出评论，系统会自动替客户做出评论，当然这种评论的结果大多都会是好评（比如国美）。但是，这类数据显然没有任何分析价值，而且这种评论是大量重复出现的，必须去除。

2）同一个人可能会出现重复的评论，因为同一个人可能会购买多种热水器，然后在进行评论的过程中可能为了省事，就在多个热水器中采用同样或相近的评论，这里当然不乏有价值的评论，但是即使有价值也只有第一条有作用。

3）由语言的特点可知，在大多数情况下，不同人之间的有价值的评论不会出现完全重复，如果出现了不同人评论之间的完全重复，这些评论一般都是毫无意义的，诸如“好好好好好”“XX 牌热水器　XX 升”等或者是直接复制、粘贴上一人的评论，这种评论显然就只有最早评论出的才有意义（即只有第一条有作用）。

（2）常见文本去重算法概述及缺陷

前人的研究成果中，有许多的文本去重算法，大多都是先通过计算文本之间的相似度，再以此为基础进行去重，包括编辑距离去重、Simhash 算法去重等，大多都存在一些缺陷。以编辑距离算法去重为例，编辑距离算法去重实际上就是先计算两条语料的编辑距离，然后进行阈值判断，如果编辑距离小于某个阈值则进行去除重复处理，这种方法针对类如：“XX 牌热水器　XX 升　大品牌高质量”以及“XX 牌热水器　XX 升　大品牌　高质量　用起来真的不错”的接近重复而又无任何意义的评论文本，去除的效果是很好的，主要为了去除接近重复或完全重复的评论数据，而并不要求完全重复，但是当这种方法测到都有意义，但是有相近的表达的时候就可能也会采取删除操作，这样就会造成错删，例如下面的例子，“还没正式使用，不知道怎样，但安装的材料费确实有点高，380”以及“还没使用，不知道质量如何，但安装的材料费确实贵，380”。这组语句的编辑距离只是比上一组大 2 而已，但是很明显这两句都是有意义的，如果阈值设为 10（该组为 9），就会带来错删问题。可惜的是，这一类的评论数据组还是不少的，特别是差评的语料，许多顾客不会用太多的言语表达，直至中心，这时问题就来了。

（3）文本去重选用的方法及原因

既然这一类相对复杂的文本去重的算法容易去除有用的数据，那么就需要考虑一些相对简单的文本去重思路。由于相近的语料存在不少是有用的评论，去除这类语料显然不合适，那么为了存留更多的有用语料，就只能对完全重复的语料下手。处理完全重复的语料直接采用最简单的比较删除法就好了，也就是两两对比，完全相同就去除的方法。

从上述的总结我们可以知道，存在文本重复问题的条目归结到底只有 1 条语料甚至 0 条语料是有用的，但是透过观察评论知道存在重复但是起码有 1 条评论有用的语料，而运用比较删除法显然只能定为留 1 条或者是全去除，因此只能设为留 1 条，以确保尽可能存留有用的文本评论信息。

观察比较删除法实现后的结果，总体效果还是很不错的。其代码如代码清单 15-2 所示。

代码清单15-2 原始数据去重

```
#-*- coding: utf-8 -*-
import pandas as pd

inputfile = '../data/meidi_jd.txt' #评论文件
outputfile = '../data/meidi_jd_process_1.txt' #评论处理后保存路径
data = pd.read_csv(inputfile, encoding = 'utf-8', header = None)
l1 = len(data)
data = pd.DataFrame(data[0].unique())
l2 = len(data)
data.to_csv(outputfile, index = False, header = False, encoding = 'utf-8')
print(u'删除了%s条评论。' %(l1 - l2))
```

代码详见：demo/code/clean_same.py

2. 机械压缩去词

（1）机械压缩去词的思想

由于电商品牌的文本评论数据质量参差不齐，没有意义的文本数据很多，因此通过文本去重就已经可以删除掉非常多的没有意义的评论文本。但是文本去重还远远不够，经过文本去重后的评论仍然有很多评论需要处理掉，例如，

“非常好非常好非常好非常好非常好非常好”

以及

“好呀好呀好呀好呀好呀好呀好呀好呀好呀好呀”。

这一类是存在连续重复的语料，也是最常见的较长的无意义语料。因为大多数给出无意义评论的人都只是为了获得一些额外奖励，并不对评论真正抱有兴趣，而他们为了省事就很可能进行这样的评论。显然这一类语料并不会重复，但是也是毫无意义的评论，是需要删除的。

可惜的是，计算机不可能自动识别出所有的这种类型的语料，比如“非常好”可以有从 1 到无上限的有穷个的叠加，即使运用词典透过某些方式识别了这一类的文本评论数据，比

如算出“非常好”比较多意味着可能是无意义评论，一位制造无意义评论的顾客还可以以任何一个词进行重复，还可以重复某词，但次数不一定多，而这种显然只需要保留第一个即可，若不处理，可能会影响情感倾向的判断，例如：

“15 分钟就出热水了，感觉还不错，但是安装费实在是太贵太贵太贵太贵”

与

“15 分钟就出热水了，感觉还不错，但是安装费实在是太贵太贵太贵”

是没有差别的，但是若不处理，就会出现差别。

因此，就需要对语料进行机械压缩去词处理，也就是说要去掉一些连续重复累赘的表达，例如把：

“哈哈哈哈哈哈哈哈哈哈哈”

缩成

“哈”

不过这样仍然会保留无意义的评论（比如上述的评论），但是这些评论在经过这步处理后，在最后一个预处理环节：短句删除环节就会被去除掉。当然，机械压缩去词法不能像分词那样去识别词语。

（2）机械压缩去词处理的语料结构

机械压缩去词实际上要处理的语料就是语料中有连续累赘重复的部分，从一般的评论偏好角度来讲，一般人制造无意义的连续重复只会在开头或者结尾进行，例如：

“为什么为什么为什么安装费这么贵，毫无道理！”

以及

“真的很好好好好好好好好”

等，而中间的连续重复虽然也有，但是非常少见（中间重复在输入上显得麻烦，无意义评论本就为了随意了事），而且中间容易有成语的问题，例如：

“安装师傅滔滔不绝的向我阐述这款热水器有多好”

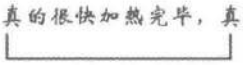
这种语料显然在去掉一个“滔”字后肯定就会出现问题，因此只对开头以及结尾的连续重复进行机械压缩去词的处理。

（3）机械压缩去词处理过程的连续累赘重复的判断及压缩规则的阐述

连续累赘重复的判断可通过建立两个存放国际字符的列表来完成，先放第一个列表，再放第二个列表，一个个读取国际字符，并按照不同情况，将其放入带第一或第二个列表或触发压缩判断，若得出重复（及列表 1 与列表 2 有意义的部分完全一对一相同）则压缩去除，这样当然就要有相关的放置判断及压缩规则。在进行机械压缩去词处理的连续累赘重复的判断及压缩规则设定的时候，必然要考虑到词法结构的问题。综合文字表达特点，设定如下 7 条规则（说明：1）这里为了初始化列表而放入的空格不算输入了国际字符；2）由于批量的评论中可能会存在某些评论无法识别，因此在进行这一步时需要结合运行进程人工删除一些无法识别语句）。

规则 1：如果读入的字符与第一个列表的第一个字符相同，而第二个列表没有任何放入的国际字符，则将这个字符放入第二个列表中。

解释：因为一般情况下同一个字再次出现时大多数都是意味着上一个词或是一个语段的结束以及下一个词或下一个语段的开始，举例如下。

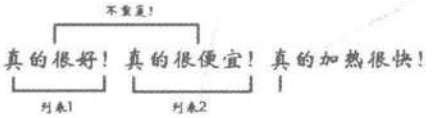


规则 2：如果读入的字符与第一个列表的第一个字符相同，而第二个列表也有国际字符，则触发压缩判断，若得出重复，则进行压缩去除，清空第二个列表。

解释：判断连续重复最直接的方法，举例如下。

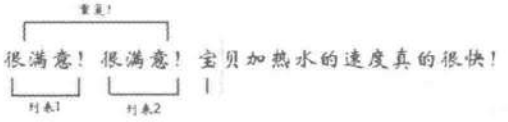
规则 3：如果读入的字符与第一个列表的第一个字符相同，而第二个列表也有国际字符，则触发压缩判断，若得出不重复，则清空两个列表，把读入的这个字符放入第一个列表第一个位置。

解释：即判断得出两个词是不相同的，都应保留，举例如下。

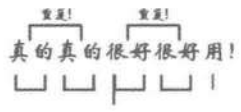


规则 4：如果读入的字符与第一个列表的第一个字符不相同，触发压缩判断，如果得出重复且列表所含国际字符数目大于等于 2，则进行压缩去除，清空两个列表，把读入的这个放入第一个列表第一个位置。

解释：用以去除下图情况的重复，并避免如“滔滔不绝”这种情况的“滔”被删除，并可顺带压缩去除另一类连续重复，见下图示例。



顺带可以处理的语料：



规则 5：如果读入的字符与第一个列表的第一个字符不相同，触发压缩判断，若得出不重复且第二个列表没有放入国际字符，则继续在第一个列表放入国际字符。

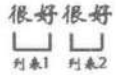
解释：没出现重复字就不会有连续重复语料，第二个列表未启用则继续填入第一个列表，直至出现重复情况为止。

规则 6：如果读入的字符与第一个列表的第一个字符不相同，触发压缩判断，若得出不重复且第二个列表已放入国际字符，则继续在第二个列表放入国际字符。

解释：类似规则 5，此处省略叙述。

规则 7：读完所有国际字符后，触发压缩判断，对第一个列表以及第二个列表有意义部分进行比较，若得出重复，则进行压缩去除。

解释：按照上述规则，在读完所有国际字符后不会再触发压缩判断条件，故为了避免下图实例连续重复情况，补充这一规则。



（4）机械压缩去词处理操作流程

根据上述规则，便可以完成对开头连续重复的处理。类似的规则，也可以对处理过的文本再进行一次结尾连续重复的机械压缩去词，算法思想是相近的，只是从尾部开始读词罢了。从结尾开始的处理结束后就得到了已压缩去词完成的精简语料。

输出被压缩的语句和原语句的对比，下图截取了一部分前向机械压缩的对比例子，如图 15-7 所示。

```
可以，可以可以可以可以可以
可以，可以
好用好用好用好用！！！
好用！
不错，不错，价格便宜
不错，价格便宜
不错不错，帮人买的！！！！
不错，帮人买的！！！！
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
a
好好好好好
好
很贵电很贵电很贵电很贵电很贵电很贵电
很贵电
```

图 15-7　被压缩的语句和原语句对比

3. 短句删除

（1）短句删除的原因及思想

完成机械压缩去词处理后，则进行最后的预处理步骤：短句删除。虽然精简的辞藻在很多时候是一种比较良好的习惯，但是由语言的特点知道，从根本上说，字数越少所能够表达的意思就越少，要想表达一些相关的意思就一定要有相应量的字数，过少的字数的评论必然是没有任何有意义的评论，比如3个字，就只能表达诸如“很不错”“质量差”等。为此，就要删除掉过短的评论文本数据，以去除掉没有意义的评论，例如，

1）原本就过短的评论文本，如“很不错”。

2）经机械压缩去词处理后过短的评论文本，即原本为存在连续重复的且无意义的长文本，如“好好好好好好好好好好好好好好好好”。

（2）保留的评论的字数下限的确定

显然，短句删除最重要的环节就是保留的评论的字数下限的确定，这个没有精确的标准，可以结合特定语料来确定，一般4～8个国际字符都是较为合理的下限，在此处设定下限为7个国际字符，即经过前两步预处理后得到的语料若小于等于4个国际字符，则将该语料删去。

经过前两步的处理后，第三步（短句删除）的效果是比较明显的，可以看出该程序能过滤掉众多的垃圾信息。

15.2.3 文本评论分词

在中文中，只有字、句和段落能够通过明显的分界符进行简单的划界，而对于“词”和“词组”来说，它们的边界模糊，没有一个形式上的分界符。因此，进行中文文本挖掘时，首先应对文本分词，即将连续的字序列按照一定的规范重新组合成词序列的过程。

分词结果的准确性对后续文本挖掘算法有着不可忽视的影响，如果分词效果不佳，即使后续算法优秀也无法实现理想的效果。例如，在特征选择的过程中，不同的分词效果，将直接影响词语在文本中的重要性，从而影响特征的选择。

本文采用Python的中文分词包“jieba”（结巴分词），对TXT文档中的商品评论数据进行中文分词。“结巴分词”提供分词、词性标注、未登录词识别，支持用户词典等功能。经过相关测试，此系统的分词精度高达97%以上。为进一步进行词频统计，分词过程将词性标注作用去掉。

15.2.4 模型构建

1. 情感倾向性模型

（1）训练生成词向量

首先训练以得到词向量，为了将文本情感分析（情感分类）转化为机器学习问题，首先就是需要将符号数学化。在NLP中，最常见的词表示方法就是One-hot Representation：将一

个词映射成一个很长的单位向量，向量的长度就是词表的大小，如“学习”表示成 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 …]，“复习”表示成 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 …]；这样就完成了词语的数学化表示。

但是，这样就存在“词汇鸿沟”的问题：即使两个词之间存在明显的联系但是在向量表示法中却体现不出来，无法反映语义关联。然而，Distributed Representation 却是能反映出词语与词语之间的距离远近关系，而用 Distributed Representation 表示的向量专门称为词向量，如“学习”可能被表示成 [0.1,0.1,0.1,0.15,0.2……]，“复习”可能被表示成 [0.11,0.12,0.1,0.15,0.22……]，这样，两个词义相近的词语被表示成词向量后，它们的距离也是较近的，词义关联不大的两个词的距离会较远。一般而言，不同的训练方法或语料库训练得到的词向量是不一样的，它们的维度常见为 50 维和 100 维。

word2vec 采用神经网络语言模型 NNLM 和 N-gram 语言模型，每个词都可以表示成一个实数向量。模型如图 15-8 所示。

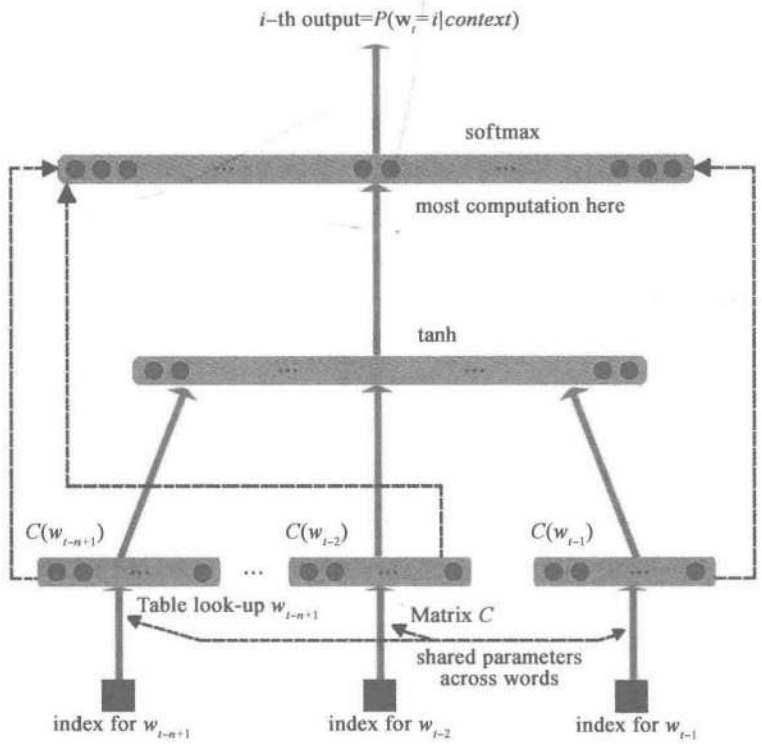


图 15-8　word2vec 模型展示图

图 15-8 最下方的 w_{t-n+1}, …, w_{t-2}, w_{t-1} 就是前 $n-1$ 个词。现在需要根据这已知的 $n-1$ 个词预测下一个词 w_t。$C(w)$ 表示词 w 所对应的词向量，存在矩阵 C（一个 $|V|\times m$ 的矩阵）中。其中 $|V|$ 表示词表的大小（语料中的总词数），m 表示词向量的维度。w 到 $C(w)$ 的转化就是从矩阵中取出一行。

网络的第一层（输入层）是将 $C(w_{t-n+1}),\cdots, C(w_{t-2}), C(w_{t-1})$ 这 $n-1$ 个向量首尾相接拼起来，形成一个 $(n-1)m$ 维的向量，记为 x。

网络的第二层（隐藏层）就如同普通的神经网络，直接使用 $d + Hx$ 计算得到。d 是一个偏置项。在此之后，使用 tanh() 作为激活函数。

网络的第三层（输出层）一共有 $|V|$ 个节点，每个节点 y_i 表示下一个词为 i 的未归一化 log 概率。最后使用 softmax() 激活函数将输出值 y 归一化成概率。最终，y 的计算公式为：

$$y = b + Wx + U\tanh(d + Hx)$$

其中，U 是隐藏层到输出层的参数，整个模型的多数计算集中在 U 和隐藏层的矩阵乘法中。矩阵 W（一个 $|V|\times(n-1)m$ 的矩阵），这个矩阵包含了从输入层到输出层的直连边。

（2）评论集子集的人工标注与映射

利用词向量构建的结果，再进行评论集子集的人工标注，正面评论标为 1，负面评论标记为 2。（或者采用 Python 的 NLP 包 snownlp 的 sentiment 功能进行简单的机器标注，减少人为工作量），然后将每条评论映射为一个向量，将分词后评论中的所有词语对应的词向量相加做平均，使得一条评论对应一个向量。

（3）训练栈式自编码网络

自编码网络是由原始的 BP 神经网络演化而来。在原始的 BP 神经网络中从特征空间输入到神经网络中，并用类别标签与输出空间来衡量误差，用最优化理论不断求得极小值，从而得到一个与类别标签相近的输出。但是，在编码网络并不是如此，并不用类别标签来衡量与输出空间的误差，而是用从特征空间的输入来衡量与输出空间的误差。其结构如图 15-9 所示。

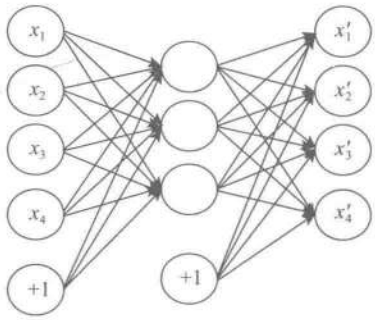


图 15-9 自编码网络结构示意图

把特征空间的向量 (x_1, x_2, x_3, x_4) 作为输入，把经过神经网络训练后的向量 (x'_1, x'_2, x'_3, x'_4) 与输入向量 (x_1, x_2, x_3, x_4) 来衡量误差，最终得到一个能从原始数据中自主学习特征的一个特征提取的神经网络。从代数角度而言，即从一个线性相关的向量中，寻找出了一组低维的基，而这组基线性组合之后又能还原成原始数据。自编码网络正是寻找了一组这样的基。

神经网络的出现，时来已久，但是因为局部极值、梯度弥散、数据获取等问题而构建不出深层的神经网络，直到 2007 年深度学习的提出，才让神经网络的相关算法得到质的改变。而栈式自编码就属于深度学习理论中一种能够得到优秀深层神经网络的方法。

栈式自编码神经网络是一个由多层稀疏自编码器组成的网络。它的思想是利用逐层贪婪训练的方法，把原来多层的神经网络剖分成一个个小的自编码网络，每次只训练一个自编码器，然后将前一层自编码的输出作为其后一层自编码器的输入，最后连接一个分类器，可以是 SVM、SoftMax 等。上述步骤是为了得到一个好的初始化深度神经网络的权重，当连接好一个分类器后，还可以用 BP 神经网络的思想，反向传播微调神经元的权重，以期得到一个

分类准确率更好的栈式自编码神经网络。

完成评论映射后，将标注的评论划分为训练集和测试集，在 Python 下利用标注好的训练集（标注值和向量）训练栈式自编码网络（SAE），对原始向量进行深度学习提取特征，后接 Softmax 分类器做分类，并用测试集测试训练好的模型的正确率。

2. 基于语义网络的评论分析

本节使用语义网络分析对评论进行进一步的分析，包括各产品独有优势、各产品抱怨点以及顾客购买原因等，并结合以上分析对品牌产品的改进提出建议。

这一部分主要通过对 3 种品牌型号的好、差评文本数据生成的语义网络图，结合共词矩阵以及评论定向筛选回查来完成对评论的分析。

（1）语义网络的概念、结构与构建本质

语义网络是由 R.F.Simon 提出的用于理解自然语言并获取认知的概念，是一种语言的概念及关系的表达。语义网络实际上就是一幅有向网络图，举例如图 15-10 所示。

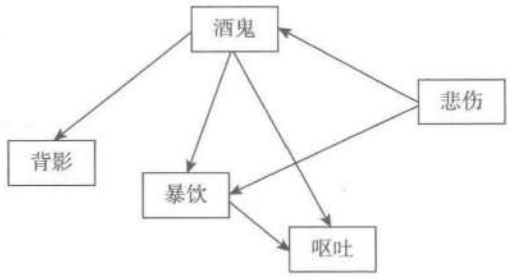


图 15-10　语义网络举例示意图

节点中的物体可以是各种用文字所表达的事物，而节点之间的有向弧则被用以表达节点之间的语言意义上的关系，其中的弧的方向是语言关系的因果指向。例如，A 指向 B 就意味着 A 与 B 有语言关系牵连且 A 与 B 分别是语义复杂关系的主动方与从动方。当然，这种用语言意义上的关系往往是复杂的。以上图为例，由于是一名酒鬼，那么他或她就经常会在特定情况之下（诸如朋友聚会、婚宴等）暴饮；一个人因受到各种挫折而显得的悲伤，长期的悲伤无法释怀，只能通过借酒浇愁，就可能会成为酒鬼。这些都是些复杂的关系。

虽然每一个语义网络结构中事物（节点）之间的关系是复杂的，但是从本质上看，语义网络的每一道弧的形成就是由于这种语义关系的存在。不同的用词表达的特定事物之间就是因为存在千丝万缕的联系，才会形成一个个的语义网络。

（2）基于语义网络进行评论分析的优势

从前面的论述中我们可知道，要想对中文的热水器评论进行合理的分析，必须要采取的一项措施就是分词，因为计算机不可能像人一样去识别每一个整句的语义，不能直接识别语句的整体结构思想。但是，分词又会使得语句的整体结构变得凌乱，因此对分词后的语句直接进行诸如产品差异等复杂的分析就不合实际，所以必须采取方法尽可能将这种原已凌乱关系重新整合起来，使得复杂的分析重新变为可能。建立起事物之间（这里分出的每一个词料代表一项事物）的语义网络关系就能够使得原已凌乱的关系得以整合，特别是那些可以连成通顺语料的词语的关系（即连接“因果”关系）的重新整合，而这种关系的成功重建能够清晰地还原语料中所反映出来的许多内容，特别是单独的词语无法清晰表达相应的情况的时候，例如，“安装”与“方便”分开的时候，任何一方都不能清晰表达相关的情况，单独一

个“安装”可以表达很多的东西，可以是“安装很容易”，也可以是“有师傅上门帮忙安装”，还可以是“安装要收手续费”等；而单独一个“方便”也可以表达很多的东西，可以是“使用十分方便”，也可以是“商品签收方便快捷”，还可以是“交款方式方便简易”等，但是如果“安装”和“方便”通过语义网络方式连接起来，如图 15-11 所示，就可以清晰地反映出是相关热水器产品在安装的时候比较便利。再如“热水”与“不足”也是这样的情况，此处就不再赘述。

图 15-11 “安装”和“方便”的语义网络连接示意图

当这种语义网络建立起来后，就可以借助它进行各种各样的特定的分析，特别是在判断特定产品优点、抽取各品牌的顾客关注点等方面具有一定的优势。以判断特定产品优点为例，如果某种产品相对于其他产品具有某种特定的优势，那么由该种商品的正面评论形成的语义网络上就会生成与其他产品正面评论形成的语义网络不一样的且蕴含着这种优势的关系连接，通过可视化，就能够从中抽取出来。

（3）基于语义网络进行评论分析的前期步骤与解释

进行语义网络分析，实际上所需要的前期步骤就是在二分类文本情感分析的基础上进行增添，语义网络的分析之所以要以二分类文本情感分析的结果为基础，在于正面的以及负面的评论大多都会具有不同的语意结构，且对于同一商品而言，正面以及负面的评论关注的点是不完全一样的，信息也是不完全一样的，正面以及负面评论之间是存在逻辑冲突的。而这种正面、负面评论的分割需要用到情感分析的技术。具体前期步骤如下。

1）数据预处理、分词以及对停用词的过滤。

2）进行情感倾向性分析，并将评论数据分割成正面（好评）、负面（差评）、中性（中评）3 大组。

3）抽取正面（好评）、负面（差评）两组，以进行语义网络的构建与分析。

第一步可以直接按照原有的流程来进行，第三步的抽取只需要在第二步分成的三组结果中抽取即可，不对中性评论进行分析是因为中性评论往往携带着比较复杂的信息，难以对细节进行倾向性提取。

而第二步的情感倾向性分析并将评论数据分类可以在原有的情感分析工作基础上做出修改来完成，但是在此处使用 ROSTCM6 来完成该项操作。ROST 系统是由武汉大学开发的一款免费反剽窃系统（ROSTCM6 全称为 ROST Content Mining System（Version 6.0）），可用以检测论文是否抄袭；同时 ROST 系统又是一款大型的免费用于社会计算的软件，可以用来实现多种类型的分析，包括情感倾向性分析以及后面将要进行语义网络的构建等。之所以使用 ROSTCM6 来完成情感分析，是因为 ROSTCM6 软件的情感倾向性分析使用的是基于优化的情感词典的方法，目前来讲，其准确率会比基于词向量以及基于神经网络的情感分析方法的正确率高，而前述用于情感倾向性分析的方法是基于词向量以及基于神经网络的情感倾向性分析方法。另外，受限于现今中文分词技术的缺陷以及评论本身的特性，能够通过中文评论所挖掘出来的内容还是偏少的，因此对情感倾向性分析的正确率要求就更高。当需要以此为

基础进一步分析的时候，就需要利用基于情感词典的方法。第二步的具体流程如下。

单击“功能性分析”项，再单击“情感分析”菜单，然后将待分析的文件地址输入“待分析文件路径”对应框内，单击“分析”选项就得到了情感倾向性分析的结果，三种情感倾向被放入 3 个不同的 TXT 文件内。操作步骤如图 15-12 所示。

图 15-12　ROSTCM6 实现情感倾向性分析的步骤示意图

这 3 步完成后，便可以开始进行语义网络分析。

（4）基于语义网络进行评论分析的实现过程

要进行语义网络分析，首先要分别对两大组重新进行分词处理，并提取出高频词（为了实现更好的分词效果，在分词词典中引入更多的词汇）。因为只有高频词之间的语义联系才是真正有意义的，个性化词语间关系不具代表性。然后在此基础上过滤掉无意义的成分，减少分析干扰。最后再抽取行特征，处理完后便可进行两组的语义网络的构建。

利用软件 ROSTCM6 来完成这一部分及语义网络构建的操作。打开 ROSTCM6 软件，单击“功能性分析”选项，再单击“社会网络与语义网络分析”菜单，便得到社会网络与语义网络分析的界面，如图 15-13 所示。

图 15-13 ROSTCM6 实现语义网络构建的步骤示意图

将分好的好、差评两个文本文档中的好评文档的地址输入“待处理文件”对应框内，并单击“提取高频词”“过滤无意义词”以及“提取行特征”按钮，这样便完成了对应的操作，系统还会自动生成对应的处理后的文件。在此之后，依次单击“构建网络”与“启动 NetDraw”按钮，就可得到好评文档的语义网络图（其生成的语义网络图可能不便观察，可以移动 NetDraw 生成的语义网络结果中的节点以增强该网络的可读性），为了方便分析，再单击“构建矩阵”按钮，形成被挑选出的节点词的矩阵词表，该操作会生成一个 xls 文件。完成好评文档的语义网络图的构建后再对差评文档进行同样的操作，将得到相应的语义网络图。3 种牌子 3 种型号对应就会有 6 个好评文档及差评文档，对应就会生成 6 个语义网络图，并以此为基础，结合共词矩阵（可在语义网络生成后再单击“构建矩阵”形成）与评论定向筛选回查，便可进行相关评论分析。

3. 基于 LDA 模型的主题分析

基于语义网络的评论分析进行初步数据感知后，从统计学习的角度，对主题的特征词出现频率进行量化表示。本文运用 LDA 主题模型，用以挖掘 3 种品牌评论中更多的信息。

主题模型在机器学习和自然语言处理等领域是用来在一系列文档中发现抽象主题的一种

统计模型。从直观上来说，传统判断两个文档相似性的方法是通过查看两个文档共同出现的单词的多少，如 TF、TF-IDF 等，这种方法没有考虑到文字背后的语义关联，可能在两个文档共同出现的单词很少甚至没有，但两个文档是相似的，因此在判断文档相似性时，应进行语义挖掘，而语义挖掘的有效工具即为主题模型。

如果一篇文档有多个主题，则一些特定的可代表不同主题的词语会反复的出现，此时，运用主题模型，能够发现文本中使用词语的规律，并且把规律相似的文本联系到一起，以寻求非结构化的文本集中的有用信息。例如，对于热水器的商品评论，代表热水器特征的词语如“安装”“出水量”“服务”等会频繁地出现在评论中，运用主题模型，将与热水器代表性特征相关的情感描述性词语，同相应的特征词语联系起来，从而深入了解热水器评价的聚焦点及用户对于某一特征的情感倾向。LDA 模型作为其中一种主题模型，属于无监督的生成式主题概率模型。

（1）LDA 主题模型介绍

潜在狄利克雷分配（Latent Dirichlet Allocation，LDA）是由 Blei 等人在 2003 年提出的生成式主题模型[24]。生成模型，即认为每一篇文档的每一个词都是通过“一定的概率选择了某个主题，并从这个主题中以一定的概率选择了某个词语”。LDA 模型也被称为 3 层贝叶斯概率模型，包含文档（d）、主题（z）和词（w）3 层结构，能够有效地对文本进行建模，和传统的空间向量模型（VSM）相比，增加了概率的信息。通过 LDA 主题模型，能够挖掘数据集中的潜在主题，进而分析数据集的集中关注点及其相关特征词。

LDA 模型采用词袋模型（Bag Of Words，BOW）将每一篇文档视为一个词频向量，从而将文本信息转化为易于建模的数字信息。

定义词表大小为 L，一个 L 维向量 (1, 0, 0,⋯, 0, 0) 表示一个词。由 N 个词构成的评论记为 $d = (w_1, w_2, \cdots, w_N)$。假设某一商品的评论集 D 由 M 篇评论构成，记为 $D = (d_1, d_2, \cdots, d_M)$。$M$ 篇评论分布着 K 个主题，记为 $z_i(i = 1, 2, \cdots, K)$。记 α 和 β 为狄利克雷函数的先验参数，θ 为主题在文档中的多项分布的参数，其服从超参数为 α 的 Dirichlet 先验分布，ϕ 为词在主题中的多项分布的参数，其服从超参数 β 的 Dirichlet 先验分布。LDA 模型图示如图 15-14 所示。

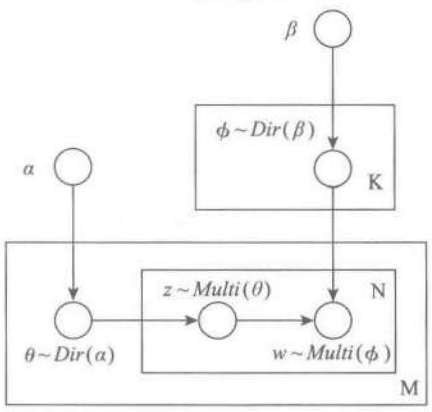


图 15-14　LDA 模型结构示意图

LDA 模型假定每篇评论由各个主题按一定比例随机混合而成，混合比例服从多项分布，记为：

$$Z|\theta = Multionomial(\theta)$$

而每个主题由词汇表中的各个词语按一定比例混合而成，混合比例也服从多项分布，记为：

$$W|Z,\phi = Multinomial(\phi)$$

在评论 d_j 条件下生成词 w_i 的概率表示为：

$$P(w_j|d_j) = \sum_{s=1}^{K} P(w_i|z=s) \times P(z=s|d_j)$$

其中，$P(w_i|z=s)$ 表示词 w_i 属于第 s 个主题的概率，$P(z=s|d_j)$ 表示第 s 个主题在评论 d_j 中的概率。

（2）LDA 主题模型估计

LDA 模型对参数 θ、ϕ 的近似估计通常使用马尔科夫链蒙特卡洛（Markov Chain Monte Carlo，MCMC）[25] 算法中的一个特例 Gibbs 抽样。利用 Gibbs 抽样对 LDA 模型进行参数估计，依据下式：

$$P(z_i = s \mid Z_{-i}, W) \propto (n_{s,-i} + \beta_i)/(\sum_{i=1}^{V} n_{s,-i} + \beta_i) \times (n_{s,-j} + \alpha_s)$$

其中，$z_i = s|$ 表示词 w_i 属于第 $s|$ 个主题的概率，Z_{-i} 表示其他所有词的概率，$n_{s,-i}$ 表示不包含当前词 w_i 的被分配到当前主题 z_s 下的个数，$n_{s,-j}$ 表示不包含当前文档 d_j 的被分配到当前主题 z_s 下的个数。

通过对上式的推导，可以推导得到词 w_i 在主题 z_s 中的分布的参数估计 $\phi_{s,i}$，主题 z_s 在评论 d_j 中的多项分布的参数估计 $\theta_{j,s}$，如下：

$$\phi_{s,i} = (n_{s,i} + \beta_i)/(\sum_{i=1}^{V} n_{s,i} + \beta_i)$$

$$\theta_{j,s} = (n_{j,s} + \alpha_s)/(\sum_{s=1}^{K} n_{j,s} + \alpha_s)$$

其中，$n_{s,i}$ 表示词 w_i 在主题 z_s 中出现的次数，$n_{j,s}$ 表示文档 d_j 中包含主题 z_s 的个数。

LDA 主题模型在文本聚类、主题挖掘和相似度计算等方面都有广泛的应用，相对于其他主题模型，其引入了狄利克雷先验知识，因此，模型的泛化能力较强，不易出现过拟合现象。其次，它是一种无监督的模式，只需要提供训练文档，它就可以自动训练出各种概率，无需任何人工标注过程，节省大量人力及时间。再者，LDA 主题模型可以解决多种指代问题。例如，在热水器的评论中，根据分词的一般规则，经过分词的语句会将“费用”一词单独分割出来，而“费用”是指安装费用，还是热水器费用等其他情况，如果简单地进行词频统计及情感分析，是无法识别的，从而无法准确了解用户反映的情况。运用 LDA 主题模型，可以求得词汇在主题中的概率分布，进而判断“费用”一词属于哪个主题，并求得属于这一主题的概率和同一主题下的其他特征词，从而解决多种指代问题。

（3）运用 LDA 模型进行主题分析的实现过程

本文在商品评论关注点的研究中，即对评论中的潜在主题进行挖掘，评论中的特征词是模型中的可观测变量。一般来说，每则评论中都存在一个中心思想，即主题。如果某个潜在主题同时是多则评论中的主题，则这一潜在主题很可能是整个评论语料集的热门关注点。在这个潜在主题上越高频的特征词越可能成为热门关注点中的评论词。

为提高主题分析在不同情感倾向下热门关注点反映情况的精确度，本文在语义网络的情感分类结果的基础上，对不同情感倾向下的潜在主题分别进行挖掘分析，从而得到不同情感倾向下用户对热水器不同方面的反映情况。例如，选取差评中的一则评论“售后服务差极了，不买他们的材料不给安装，还谎称免费安装，其实要收挺贵的安装费，十分不合理。这也算了，安装费之前说二百，安好之后要四百，更贵了，更加不合理，不管是安装师傅自己还是美的规定，都是很差很差的体验，我看其他人的了，一样的安装，比别人贵的安装费。而且安装师傅做事粗糙，态度粗鲁。”在这则评论中，“安装费”和“安装师傅”在这则评论中出现频率较高，可作为潜在主题。同时，可以得到潜在主题上特征词的概率分布情况，反映潜在主题“安装费”的特征词包括“贵”“不合理”，反映“安装师傅”的特征词包括“粗糙”“粗鲁”。

分别统计整个评论语料库中正负情感倾向的主题分布情况，对两种情感倾向下各个主题出现的次数从高到低进行排序，根据分析需要，选择排在前若干位的主题作为评论集中的热门关注点，然后根据潜在主题上的特征词的概率分布情况，得到所对应的热门关注点的评论词。

本文运用 LDA 主题模型的算法，并采用 Gibbs 抽样方法对 LDA 模型的参数进行近似估计，由上文的模型介绍可知，模型中存在 3 个可变量需要确定最佳取值，分别是狄利克雷函数的先验参数 α、β 和主题个数 K。本文中将狄利克雷函数的先验参数 α 和 β 设置为经验值，分别是 $\alpha = 50/K$，$\beta = 0.1$。而主题个数 K 采用统计语言模型中常用的评价标准困惑度 [26] 来选取，即令 $K = 50$。

（4）LDA 模型的实现

虽然 LDA 可以直接对文本做主题分析，但是文本的正面评价和负面评价混淆在一起，并且由于分词粒度的影响（否定词或程度词等），可能在一个主题下生成一些令人迷惑的词语。因此，将文本分为正面评价和负面评价两个文本，再分别进行 LDA 主题分析是一个比较好的主意。

为将文本一分为二，可以进行手工分类，但是极耗精力和时间。为此，可以进行机器标注。这里采用 COSTCM6 中的情感分析做机器分类，生成“正面情感结果”“负面情感结果”和“中性情感结果”，这里抛弃“中性情感结果”文本，分别对“正面情感结果”和“负面情感结果”文本进行 LDA 分析，挖掘出商品的优点与不足。

图 15-15 是 meidi_jd_process_end.txt 得到的负面评价文本，由于 COSTCM6 得到的结果还有评分前缀，还需要对前缀进行评分删除，并且分类文本是 unicode 编码，则统一另存为 UTF-8 编码再删除评分。删除前缀评分的代码如代码清单 15-3 所示。

```
-1	还行吧 安装有点小贵
-6	热水器很好用 但安装说明不清楚
-9	安装费超贵无比 一副挂件150元 角阀36元一个 我说我自己有 安装师傅说&ldquo;你的不好 质量不好 我去 我买的可是九牧啊 &rdquo;最可气的是 角阀在墙上有个装饰盖 竟然另收费 12元一个 结果按照师傅的&ldquo;满意度&rdquo;装完后 光安装就花了327元&mdash;&mdash;半个热水器 安装挺坑人的
```

图 15-15　负面评价文本

代码清单15-3 删除前缀评分代码

```
#-*- coding: utf-8 -*-
import pandas as pd

#参数初始化
inputfile1 = '../data/meidi_jd_process_end_负面情感结果.txt'
inputfile2 = '../data/meidi_jd_process_end_正面情感结果.txt'
outputfile1 = '../data/meidi_jd_neg.txt'
outputfile2 = '../data/meidi_jd_pos.txt'

data1 = pd.read_csv(inputfile1, encoding = 'utf-8', header = None) #读入数据
data2 = pd.read_csv(inputfile2, encoding = 'utf-8', header = None)

data1 = pd.DataFrame(data1[0].str.replace('.*?\d+?\\t ', '')) #用正则表达式修改数据
data2 = pd.DataFrame(data2[0].str.replace('.*?\d+?\\t ', ''))

data1.to_csv(outputfile1, index = False, header = False, encoding = 'utf-8') #保存结果
data2.to_csv(outputfile2, index = False, header = False, encoding = 'utf-8')
```

代码详见：demo/code/clean_prefix.py

接下来，需要对两文本进行分词，保存成两个文本文档，并和停用词文档一起作为 LDA 程序的输入。分词代码如代码清单 15-4 所示。

代码清单15-4 分词代码

```
#-*- coding: utf-8 -*-
import pandas as pd
import jieba #导入结巴分词，需要自行下载安装

#参数初始化
inputfile1 = '../data/meidi_jd_neg.txt'
inputfile2 = '../data/meidi_jd_pos.txt'
outputfile1 = '../data/meidi_jd_neg_cut.txt'
outputfile2 = '../data/meidi_jd_pos_cut.txt'

data1 = pd.read_csv(inputfile1, encoding = 'utf-8', header = None) #读入数据
data2 = pd.read_csv(inputfile2, encoding = 'utf-8', header = None)

mycut = lambda s: ' '.join(jieba.cut(s)) #自定义简单分词函数
data1 = data1[0].apply(mycut) #通过“广播”形式分词，加快速度。
data2 = data2[0].apply(mycut)

data1.to_csv(outputfile1, index = False, header = False, encoding = 'utf-8') #保存结果
data2.to_csv(outputfile2, index = False, header = False, encoding = 'utf-8')
```

代码详见：demo/code/cut.py

在分好词的正面评价、负面评价文件以及过滤用的停用词表的基础上，使用 Python 的 Gensim 库完成 LDA 分析的代码如代码清单 15-5 所示。

代码清单15-5　LDA代码

```
#-*- coding: utf-8 -*-
import pandas as pd

#参数初始化
negfile = '../data/meidi_jd_neg_cut.txt'
posfile = '../data/meidi_jd_pos_cut.txt'
stoplist = '../data/stoplist.txt'

neg = pd.read_csv(negfile, encoding = 'utf-8', header = None) #读入数据
pos = pd.read_csv(posfile, encoding = 'utf-8', header = None)
stop = pd.read_csv(stoplist, encoding = 'utf-8', header = None, sep = 'tipdm')
#sep设置分割词，由于csv默认以半角逗号为分割词，而该词恰好在停用词表中，因此会导致读取出错
#所以解决办法是手动设置一个不存在的分割词，如tipdm。
stop = [' ', ''] + list(stop[0]) #Pandas自动过滤了空格符，这里手动添加

neg[1] = neg[0].apply(lambda s: s.split(' ')) #定义一个分割函数，然后用apply广播
neg[2] = neg[1].apply(lambda x: [i for i in x if i not in stop]) #逐词判断是否停用词，思路同上
pos[1] = pos[0].apply(lambda s: s.split(' '))
pos[2] = pos[1].apply(lambda x: [i for i in x if i not in stop])

from gensim import corpora, models

#负面主题分析
neg_dict = corpora.Dictionary(neg[2]) #建立词典
neg_corpus = [neg_dict.doc2bow(i) for i in neg[2]] #建立语料库
neg_lda = models.LdaModel(neg_corpus, num_topics = 3, id2word = neg_dict) #LDA模型训练
for i in range(3):
  neg_lda.print_topic(i) #输出每个主题

#正面主题分析
pos_dict = corpora.Dictionary(pos[2])
pos_corpus = [pos_dict.doc2bow(i) for i in pos[2]]
pos_lda = models.LdaModel(pos_corpus, num_topics = 3, id2word = pos_dict)
for i in range(3):
  neg_lda.print_topic(i) #输出每个主题
```

代码详见：demo/code/LDA.py

经过 LDA 主题分析后，评论文本被聚成 3 个主题，每个主题下生成 10 个最有可能出现的词语以及相应的概率，表 15-2 显示了美的正面评价文本中的潜在主题，表 15-3 展示了美的负面评价文本中的潜在主题。

表15-2　美的正面评价潜在主题

主题 1	主题 2	主题 3	主题 1	主题 2	主题 3
很好	不错	安装	就是	还不错	自己
送货	的	了	好	京东	美的
快	东西	师傅	加热	美的	的

（续）

主题 1	主题 2	主题 3	主题 1	主题 2	主题 3
速度	价格	元	服务	很不错	售后
很快	感觉	没有	非常	值得	上门

表15-3 美的负面评价潜在主题

主题 1	主题 2	主题 3	主题 1	主题 2	主题 3
安装	就是	了	售后	有点	自己
师傅	不错	的	服务	还可以	还是
美的	加热	东西	不好	使用	但是
元	不知道	没有	上门	速度	这个
送货	不过	京东	好	吧	可以

根据对美的热水器好评的 3 个潜在主题的特征词提取，主题 1 中的高频特征词，即很好、送货快、加热、速度、很快、服务和非常等，主要反映京东送货快、服务非常好；美的热水器加热速度快；主题 2 中的高频特征词，即热门关注点主要是价格、东西和值得等，主要反映美的热水器不错，价格合适值得购买等；主题 3 中的高频特征词，即热门关注点主要是售后、师傅、上门和安装等，主要反映京东的售后服务以及师傅上门安装等。

从美的热水器差评的 3 个潜在主题中，我们可以看出，主题 1 中的高频特征词主要是安装、服务、元等，即主题 1 主要反映的是美的热水器安装收费高、热水器售后服务不好等；主题 2 中的高频特征词主要是不过、有点、还可以等情感词汇；主题 3 主要反映的是美的热水器可能不满足其需求等；主题 3 中的高频特征词主要是没有、但是、自己等，主题 3 可能主要反映美的热水器自己安装等。

综合以上对主题及其中的高频特征词可以看出，美的热水器的优势有以下几个方面：价格实惠、性价比高、外观好看、热水器实用、使用起来方便、加热速度快、服务好。

相对而言，用户对美的热水器的抱怨点主要体现以下几个方面：美的热水器安装的费用贵及售后服务等。

因此，用户的购买原因可以总结为以下几个方面：美的大品牌值得信赖，美的热水器价格实惠，性价比高。

根据对京东平台上美的热水器的用户评价情况进行 LDA 主题模型分析，我们对美的品牌提出以下建议。

1）在保持热水器使用方便、价格实惠等优点的基础上，对热水器进行改进，从整体上提升热水器的质量。

2）提升安装人员及客服人员的整体素质，提高服务质量。安装费用收取明文细则，并进行公开透明，减少安装过程的乱收费问题。适度降低安装费用和材料费用，以此在大品牌的竞争中凸显优势。

15.3　上机实验

1. 实验目的

- 学习运用 Python 对文本数据做数据清洗（预处理）的方法。
- 学习运用 ROSTCM6 对评论文本分类的方法。
- 学习运用 Python 对文本做分词处理的方法。
- 加深对 LDA 主题分析算法原理的理解及使用。
- 掌握使用 LDA 主题分析算法解决实际问题的方法。

2. 实验内容

- 京东评论数据中有众多品牌的评论，数据见“ test/data/huizong.csv ，利用 Pandas 从 csv 文件中提取出某个品牌的评论数据（如海尔），并对数据做预处理。
- 将原始数据做完预处理后，再利用 ROSTCM6 对划分评论数据分类，并对两个文本做分词处理和 LDA 主题分析。

3. 实验方法与步骤

实验一

1）打开 Python 软件，仿照 excel2txt.py 脚本输入命令。

2）使用 readLines() 函数读取数据，并筛选出某个品牌的评论，保存成文本文件（编码统一用 UTF-8，下同）。

- 依次编写 clean_same.py 函数和 clean_prefix.py 函数。
- clean_ prefix.py 函数对于各个类别的评分进行删除。

3）编写并运行程序后，与之前的对比、观察预处理后的效果。

实验二

1）利用 ROSTCM6 将预处理后的文本一分为二（只保留正面评价和负面评价）。

- 打开 ROSTCM6 软件，选择“功能性分析”——“情感分析”。
- 在“待分析文件路径”中选择预处理后的文件路径，单击“分析”。
- 将得到的正面评价和负面评价文本另存为到“ \test\data\” 目录下，并将编码改回 UTF-8（而非 unicode）。
- 编写 clean_prefix.py 代码，运用正则表达式将上述两个文本的前缀评分和空格去除，并保存为 meidi_jd_pos.txt 和 meidi_jd_pos.txt 文本。

2）利用 jieba 模块分别对上述所得的两个文本做分词，为达到更好的分词效果，添加自定义词典 myDict.txt（在 \data 中），可以尝试往 myDict.txt 中自定义编辑添加词组。

3）编写 LDA.py 代码，分别对 meidi_jd_pos_cut.txt 和 meidi_jd_pos_cut.txt 文本运行，分析产品的优点和不足。

4. 思考与实验总结

如何在 Python 中实现情感分析，将评论内容分为正面和负面评价？

15.4 拓展思考

应用层次分析法（Analytical Hierachy Process，AHP）是匹兹堡大学 T. L. Saaty 教授在 20 世纪 70 年代初期提出对定性问题进行定量分析的一种渐变灵活的多准则决策方案，其特点是把复杂问题中的各种因素通过划分为相互联系的有序层次，使之条理化，根据对有一定客观现实的主观两两比较，把专家意见和分析者的客观判断结果直接有效地结合起来，然后利用数学方法计算每一层元素相对重要性次序的权值，最终通过所有层次间的总排序计算所有元素的相对权重并进行排序，从而分析消费者决策。

模糊综合评判（Fuzzy Comprehensive Evaluation，FCE）是 20 世纪 80 年代初，我国模糊数学领域的汪培庄教授提出的综合评判模型，并通过广大实际工作者的不断地补充发展，衍生出的适用于各种领域的评判方法。模糊综合评判的过程可简述为：决策者将目标看成是由多重因素组成的因素集 U，再设定这些因素所能选取的评审等级，组成评语的评判集合 V，分别求出各单一因素对各个评审等级的模糊矩阵，然后根据各个因素在评价中的权重分配，通过模糊矩阵合成，求出评价的定量值。

但是，这两种方法各有利弊：AHP 能够准确地对决策定性，但其决策过程需要经过大量数据比对来最终通过概率确定权重；而在 FCE 中虽然有很好的定量评价，但是无法很好地对决策定性。请利用本案例的数据，尝试通过对二者的结合来实现对电商平台上热水器的购买决策分析。

AHP-FCE 模型需要经历以下 3 个步骤，具体流程如图 15-16 所示。

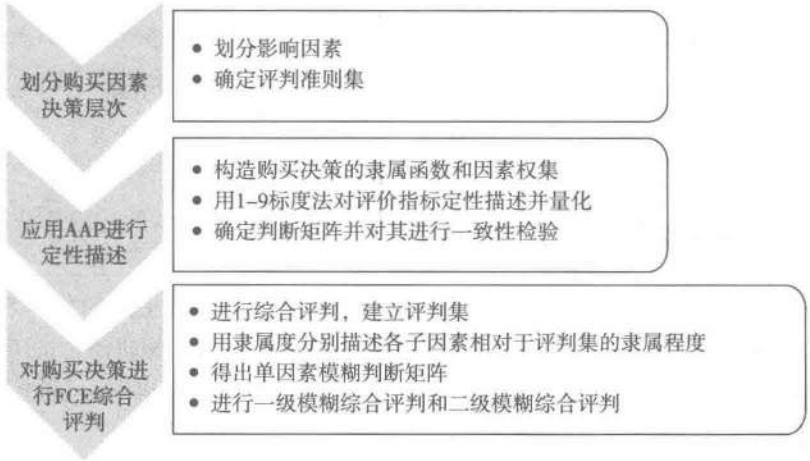


图 15-16 AHP-FCE 模型

- 划分因素层。
- 应用 AHP 构造消费者心理的隶属函数和因素权集合。
- 对所求结果进行综合评判。

15.5　小结

本章结合京东商城美的热水器评论的文本分析的案例，重点介绍了数据挖掘算法中文本挖掘分词算法以及 LDA 主题模型在实际案例中的应用。本章研究京东平台上的热水器评论问题，从分析某一热水器的用户情感倾向出发挖掘出该热水器的优点与不足，从而提升对应商品的生产厂家自身的竞争力。

参考文献

[1] Programming Community Index：http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

[2] 天文科研中的 Python：http://qosmology.org/python-in-astronomy-research/.

[3] 方积乾 . 生物医学研究的统计方法 [M]. 北京：高等教育出版社 .2007:16-17.

[4] 张静远，张冰，蒋方舟 . 基于小波变换的特征提取方法分析 [J].2000:1-8.

[5] 张良均 , 王靖涛 , 李国成 . 小波变换在桩基完整性检测中的应用 [J].2002：1-2.

[6] 廖芹 . 数据挖掘与数学建模 [M]. 北京：国防工业出版社 .2010:49-50.

[7] 何晓群 . 应用回归分析 [M]. 北京：中国人民大学出版社 .2011.

[8] Quinlna J R，Induction of decision tree，Machine Learning[M]. 198，(1):81-106.

[9] 张良均 . 神经网络从入门到精通 [M]. 北京：机械工业出版社 .2012.11-12.

[10] 周春光 . 计算智能 [M]. 吉林：吉林大学出版社 .2009.43-44.

[11] 张良均 . 数据挖掘：实用案例分析 [M]. 北京：机械工业出版社 .2013.

[12] Jiawei Han，Micheline Kamber. Data Mining Concepts and Techniques[M]. 机械工业出版社 .2012:247-254.

[13] 王燕 . 应时间序列分析 [M]. 北京：中国人民大学出版社 .2012.

[14] Pang-Ning Tan,Michael Steinbach，Vipin Kumar.Introdution to Data Mining[M]. 北京：人民邮电出版社 .2010:404-415.

[15] 罗亮生，张文欣 . 基于常旅客数据库的航空公司客户细分方法研究 [J]. 现代商业，2008(23).

[16] 电子商务网站 RFM 分析 . [EB/OL] .http://www.skynuo.com/Seo_detail131.Html/.

[17] 徐力，鹿竞文 . 三阴乳腺癌证素变化规律及截断疗法研究 [D]. 人民卫生出版社，2012.

[18] Stricker M A, Orengo M. Similarity of color images[C]//IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology. International Society for Optics and Photonics, 1995: 381-392.

[19] 袁守正，丁富强，裴国才 . 云计算环境下业务系统健康度模型研究 [J]. 电信技术，2014(03).

[20] 张利 . 基于时间序列 ARIMA 模型的分析预测算法研究及系统实现 [D]. 江苏大学，2008.

[21] 项亮 . 推荐系统实战 [M]. 北京：人民邮电出版社，2012.6.

[22] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso[J].J. Royal. Statist. Soc B., Vol. 58, No. 1, pages 267-288.

[23] Zou H. The Adaptive Lasso and its oracle properties [J]. Journal of the American Statistical Association，2006，101(476): 1418-1429.

[24] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research，2003，3:2003.

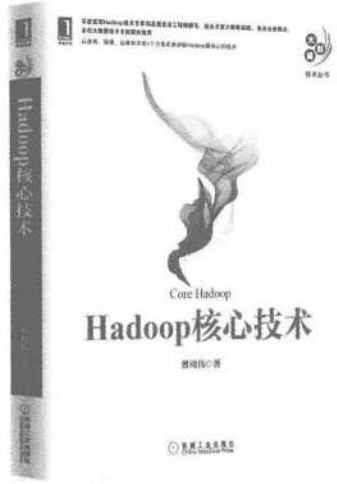
[25] BERG B A .Markov Chain Monte Carlo Simulations and Their Statistical Analysis [M]. Singapore: World Scientific. 2004.

[26] Cao Juan, Xia Tian，Li Jin Tao，A density method for adaptive LDA model selection[J]. Neurocomputing 2009(72):1775-1781.

推荐阅读

Spark大数据处理：技术、应用与性能优化

作者：高彦杰 ISBN：978-7-111-48386-1 定价：59.00元



Hadoop核心技术

作者：翟周伟 ISBN：978-7-111-49468-3 定价：69.00元

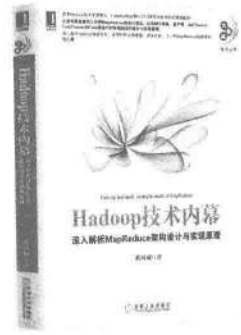
大规模分布式系统架构与设计实战

作者：彭渊 ISBN：978-7-111-45503-5 定价：59.00元

大规模分布式存储系统：原理解析与架构实战

作者：杨传辉 ISBN：978-7-111-43052-0 定价：59.00元

推荐阅读



Hadoop实战（第2版）
作者：陆嘉恒
ISBN：978-7-111-39583-6
定价：79.00元

Hadoop技术内幕：
深入解析MapReduce架构设计与实现原理
作者：董西成
ISBN：978-7-111-42226-6
定价：69.00元

数据挖掘与数据化运营实战：
思路、方法、技巧与应用
作者：卢辉
ISBN：978-7-111-42650-9
定价：59.00元

数据挖掘：实用案例分析
作者：张良均
ISBN：978-7-111-42591-5
定价：79.00元

Hadoop技术内幕：
深入解析Hadoop Common和HDFS架构设计与实现原理
作者：蔡斌 等
ISBN：978-7-111-41766-8
定价：89.00元

网站数据分析：
数据驱动的网站管理、优化和运营
作者：张洪举
ISBN：978-7-111-43514-3
定价：69.00元

随着云时代的来临，大数据技术将具有越来越重要的战略意义。大数据已经渗透到每一个行业和业务职能领域，逐渐成为重要的生产要素，人们对于海量数据的运用将预示着新一轮生产率增长和消费者盈余浪潮的到来。大数据分析技术将帮助企业用户在合理时间内攫取、管理、处理、整理海量数据，为企业经营决策提供积极的帮助。大数据分析作为数据存储和挖掘分析的前沿技术，广泛应用于物联网、云计算、移动互联网等战略性新兴产业。

为了满足目前的大数据分析人才需求，本书以大家熟知的数据挖掘建模工具Python语言来展开，以解决某个应用的挖掘目标为前提，先介绍案例背景提出挖掘目标，再阐述分析方法与过程，最后完成模型构建，在介绍建模过程中穿插操作训练，把相关的知识点嵌入相应的操作过程中，使读者轻松理解并掌握相关的理论和知识点。

上架指导：计算机/数据挖掘
ISBN 978-7-111-52123-5
9 787111 521235 >
定价：69.00元

投稿热线：(010) 88379604
客服热线：(010) 88379426 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

[General Information]
书名=Python数据分析与挖掘实战
作者=张良均著
页数=336
SS号=13932384
DX号=
出版日期=2016.01
出版社=北京机械工业出版社