

西安石油大学

本科毕业设计（论文）

题 目： 海量数据的异常项检测
 及分类方法研究

学生姓名： 刘阳

院 （系）： 计算机学院

专业班级： 软件 1302

指导教师： 张留美

完成时间： 2017 年 6 月

海量数据的异常项检测及分类方法研究

摘 要：随着互联网的发展，人们正处于一个信息爆炸的时代。相比于过去的信息匮乏，面对现阶段海量的信息数据，人们淹没在数据中难以快速的制定合适的决策，使用简单的数据处理办法已经不能适用于现如今的需求。

针对现状，本文采用不同数据处理方法来进行研究并对实现对给定的数据集进行数据检测分析，寻找能够在相同的硬件环境下提高海量数据处理效率的算法。提出了基于决策树的数据处理方法，即对数据进行特征提取、原始样本数据的预处理、构造决策树模型、进行数据挖掘预测等方面进行了深入讨论，并详细阐述了运用算子进行数据集处理的具体实现过程。利用 rapidminer 对数据处理过程进行可视化解析。通过对数据处理分析，所得数据具有较强的可读性、较好的可预测性。

设计并完成了数据在 Web 端的录入及储存在 mysql 数据库中，采用可视化工具 rapidminer 对 mysql 数据库中的数据进行读入，然后进行对读入数据进行预处理，即完成异常数据的检测及排除，在完成预处理后进一步进行所需属性的筛选，接着设置 Label 规则最后将处理后的数据传送给决策树模型，产生决策树可视化图像并最终根据决策树传出数据作出预测。

采用本文的数据处理方法可以大幅度提高数据使用效率，直观立体的反应海量数据对未来预测的指导性作用，明显优于传统数据处理方法。

关键词：海量数据；数据处理；推荐算法；属性聚类；决策树

Research on the Detection and Classification of Exception in Massive Data

Abstract: With the development of the Internet, people are in an era of information explosion. Compared to the lack of information in the past, in the face of massive information at this stage, it is difficult to develop appropriate decisions in the data, and the use of simple data processing can not be applied to today's needs.

In this paper, different data processing methods are used to study and realize the data detection and analysis of a given data set, and find an algorithm which can improve the efficiency of massive data processing in the same hardware environment. The data processing method based on decision tree is proposed, which is the feature extraction of the data, the preprocessing of the original sample data, the construction of the decision tree model, the data mining prediction and so on, and elaborates the use of the operator to carry out the data set Processing the specific implementation process. Using RapidMiner to visualize the data processing process. Through the data processing analysis, the obtained data has a strong readability, better predictability.

Design and complete the data in the Web side of the input and stored in the mysql database, the use of visual tools RapidMiner mysql database to read the data, and then read the data preprocessing, that is, the completion of abnormal data detection and exclusion, in the After the preprocessing, the filtering of the required attributes is carried out, and then the Label rule is set. Finally, the processed data is sent to the decision tree model, the decision tree visualization image is generated and finally the data is predicted based on the decision tree.

The data processing method of this paper can greatly improve the efficiency of data utilization, and the guidance function of intuitive stereo response to future forecasting is obviously superior to traditional data processing method.

Keywords: Massive Data; Data Processing; Recommendation Algorithm; Attribute Clustering; Decision tree

目 录

1	绪论	1
1.1	研究背景	1
1.2	研究需求与意义	1
1.3	常用数据挖掘软件简介	2
1.4	研究现状及其前景	2
1.4.1	海量数据处理的主要方法的研究现状	2
1.4.2	大数据的未来发展	3
2	异常检查	1
2.1	离群点检测	1
2.1.1	离群点检测方法	3
2.1.2	基于模型的离群点检测方法	3
2.1.3	基于聚类的离群点检测方法	5
3	决策树算法实现流程	1
3.1	什么是决策树算法	1
3.2	数据决策树处理	5
4	数据处理基础	6
4.1	原始数据集整理与统计	6
4.2	数据预处理基本方法整理	6
4.2.1	数据清洗	6
4.2.2	数据集成	8
4.2.3	数据变化	8
4.2.4	数据规约	9
5	数据预处理实现及结果分析	12
5.1	数据准备	12
5.2	数据清洗—数据集缺失值的处理	12
5.2.1	user 数据集中缺失值的处理:	12
5.2.2	年龄的处理	13
5.2.3	关于专业的预处理	14
5.3	数据变换	15
5.3.1	关于驻地的相关数据处理及决策树生成	16
5.3.2	关于工种的相关数据处理及决策树生成	18

6	实验结果及其分析	21
6.1	数据集来源	21
6.2	度量标准	21
6.3	用户分类	21
6.3.1	用户信息统计	22
6.4	数据预测	22
6.4.1	数据预测准备	22
6.4.2	数据预测操作	22
6.5	实验结果分析	24
7	结论	25
	致谢	26
	参考文献	26
	附录	27

1 绪论

1.1 研究背景

21 世纪是一个信息化时代，根据维基百科的定义，信息化时代是每个人都有能力去自由的传播信息，并且能够及时的获取信息的时代。信息化时代是以计算机技术及网络技术为核心，让信息的传播建立在网络及电子设备上。在信息化时代，为了满足时代的要求，获取大量且具有时效性的“信息”则变得非常重要。

随着互联网的发展，人们正处于一个信息爆炸的时代。相比于过去的信息匮乏，面对现阶段海量的信息数据，人们淹没在数据中难以快速的制定合适的决策。

面对上述的问题，人们开始思考如何从海量数据集中获取用户的信息和知识，然而面对高维、复杂、异构的海量数据，提取有用的潜在信息成为一大难题。面对这个难题，数据挖掘技术应运而生。

众所周知，对信息的筛选和过滤是衡量一个系统好坏的重要指标。一个具有良好用户体验的系统，会将海量信息进行筛选、过滤，将用户最关注最感兴趣的信息展现在用户面前。这大大增加了系统工作的效率，也节省了用户筛选信息的时间。

其中，使用数据处理工具进行数据处理其在用户对信息选择以及依据用户来刷选有用信息方面有很大的使用几率。降低过量数据给人们选择带来的困扰，减少用户选择成本，提高有用信息数据的获取几率。比如说在电子商务方面，如何让消费者在过多商品信息中选择出自己想要的商品以及自己潜在需求的商品，数据处理就能很好的解决这个问题。

1.2 研究需求与意义

搜索引擎的出现在一定程度上解决了信息筛选问题，能有效地实现从海量数据及中获取有用信息和知识，但还远远不够。搜索引擎需要用户主动提供关键词来对海量信息进行筛选。当用户无法准确描述自己的需求时，搜索引擎的筛选效果将大打折扣，而用户将自己的需求和意图转化成关键词的过程本身就是一个并不轻松的过程。

在这个时代，无论是信息消费者还是信息生产者都遇到了很大的挑战：对于信息消费者，从大量信息中找到自己感兴趣的信息是一件非常困难的事情；对于信息生产者，让自己生产的信息脱颖而出，受到广大用户的关注，也是一件非常困难的事情。推荐系统就是解决这一矛盾的重要工具。推荐系统的任务就是联系用户和信息，一方面帮助用户发现对自己有价值的信息，另一方面让信息能够展现在对它感兴趣的用户面前，从而实现信息消费者和信息生产者的双赢。

1.3 常用数据挖掘软件简介

商用数据挖掘软件比较著名的有 SPSS Clementine、IBM Intelligent Miner、SAS Enterprise Miner 等等，他们都能提供常规的挖掘过程和挖掘模式。Excel、Matlab 等提供了数据挖掘模块。开源数据挖掘工具主要有 WEKA、RapidMiner、ARMiner 和 AlphaMiner 等。

本文中采用 RapidMiner 软件对数据进行处理，接下来，我们主要介绍一下 RapidMiner 这款软件。

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 Rhadoop，可以和 Hadoop 集成起来，在 Hadoop 集群上运行任务。

RapidMiner Studio 结合技术性和适用性，为最新的及已建立的人性化数据挖掘技术提供服务。通过推拽算子，设置参数及组合算子，在 RapidMiner Studio 中定义分析流程。流程能从大量的随机的可嵌套的算子中产生，最终表示为所谓的流程图（流程设计）。流程结构由内部的 XML 来描述，通过图形用户界面来开发。在后台，RapidMiner Studio 不断地检查当前流程开发状态，确保语法一致，并在问题出现时，能自动推荐解决方案。以上功能是通过所谓的元数据转换实现的，即在流程设计阶段转换基础元数据，预知流程开发结果，并在出现不合适的算子组合时确定解决方案（快速修复）。此外，RapidMiner Studio 也能定义断点，因此能检查几乎所有的中间结果。成功组合的算子会被合并到构建模块中，因此在后期流程中它们还能被再次使用。通过 YALE，我们可以检测数据是否存在错误以及在建模过程中出现的问题，及时找出问题并加以解决。

1.4 研究现状及其前景

1.4.1 海量数据处理的主要方法的研究现状

主流的海量数据处理技术基本上可以归结为两步，即：数据质量分析和数据的特征分析方法。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。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。

(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，可做出频率分布表、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。

(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，选择得合适，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。

(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。

(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。

(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。比如对一个公司来讲，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。

(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。

1.4.2 大数据的未来发展

而大数据未来的发展趋势则从以下几个方面进行：

(1) 开放源代码

大数据获得动力，关键在于开放源代码，帮助分解和分析数据。Hadoop 和 NoSQL 数据库便是其中的赢家，他们让其他技术商望而却步、处境很被动。毕竟，我们需要清楚怎样创建一个平台，既能解开所有的数据，克服数据相互独立的障碍，又能将数据重新上锁。

(2) 市场细分

当今，许多通用的大数据分析平台已投入市场，人们同时期望更多平台的出现，可以运用在特殊领域，如药物创新、客户关系管理、应用性能的监控和使用。若市场逐步成熟，在通用分析平台之上，开发特定的垂直应用将会实现。但现在的技术有限，除非

考虑利用潜在的数据库技术作为通用平台（如 Hadoop、NoSQL）。人们期望更多特定的垂直应用出现，把目标定为特定领域的数据分析，这些特定领域包括航运业、销售业、网上购物、社交媒体用户的情绪分析等。同时，其他公司正在研发小规模分析引擎的软件套件。比如，社交媒体管理工具，这些工具以数据分析做为基础。

(3) 预测分析

建模、机器学习、统计分析和大数据经常被联系起来，用以预测即将发生的事情和行为。有些事情是很容易被预测的，比如坏天气可以影响选民的投票率，但是有些却很难被准确预测。例如，中间选民改变投票决定的决定性因素。但是，当数据累加时，我们基本上有能力可以大规模尝试一个连续的基础。网上零售商重新设计购物车，来探索何种设计方式能使销售利润最大化。根据病人的饮食、家族史和每天的运动量，医生有能力预测未来疾病的风险。当然，在人类历史的开端，我们就已经有各种预测。但是，在过去，许多预测都是基于直觉，没有依靠完整的数据集，或者单单靠的是常识。当然，即便有大量数据支撑你的预测，也不表明那些预测都是准确的。2007 年和 2008 年，许多对冲基金经理和华尔街买卖商分析市场数据，认为房地产泡沫将不会破灭。根据历史的数据，可以预测出房地产泡沫即将破裂，但是许多分析家坚持原有的观点。另一方面，预测分析在许多领域流行起来，例如欺诈发现（比如在外省使用信用卡时会接到的诈骗电话），保险公司和顾客维系的风险管理。

2 异常检查

支持向量机（Support Vector Machines）是由 Vapnik 等人于 1995 年提出来的。之后随着统计理论的发展，支持向量机也逐渐受到了各领域研究者的关注，在很短的时间就得到很广泛的应用。支持向量机是建立在统计学习理论的 VC 维理论和结构风险最小化原理基础上的，利用有限的样本所提供的信息对模型的复杂性和学习能力两者进行了寻求最佳的折衷，以获得最好的泛化能力。SVM 的基本思想是把训练数据非线性的映射到一个更高维的特征空间（Hilbert 空间）中，在这个高维的特征空间中寻找到一个超平面使得正例和反例两者间的隔离边缘被最大化。SVM 的出现有效的解决了传统的神经网络结果选择问题、局部极小值、过拟合等问题。并且在小样本、非线性、数据高维等机器学习问题中表现出很多令人瞩目的性质，被广泛地应用在模式识别，数据挖掘等领域（张学工 2000；崔伟东 2001）。支持向量机可以用于分类和回归问题，本章着重介绍分类相关的知识。

2.1 离群点检测

就餐饮企业而言，经常会碰到这样的问题：

- 如何根据客户的消费记录检测是否为异常刷卡消费？
- 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，如图 2-1 离群点远离其它数据点。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。

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、电子商务中、网络入侵、天气预报等领域，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。

- 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。
- 离群点的类型对离群点的大致分类见表 2-1

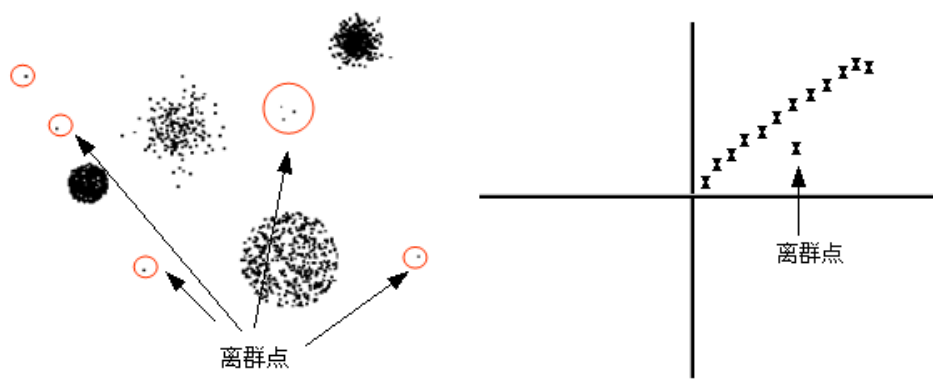


图 2-1: 离群点检测示意图

表 2-1: 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 2-2：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。
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。

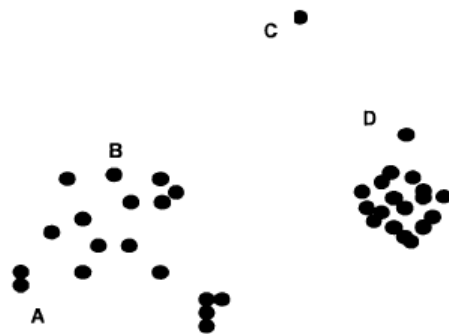


图 2-2: 全局离群点和局部离群点

表 2-2: 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的处理；大数据集不适用；参数选择是困难的。
基于聚类	一种是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。

2.1.1 离群点检测方法

常用离群点检测方法见表2-2

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。

本节重点介绍基于统计模型和聚类的离群点检测方法。

2.1.2 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跟该模型拟合，即如果它很可能不服从该分布，则它是一个离群点。

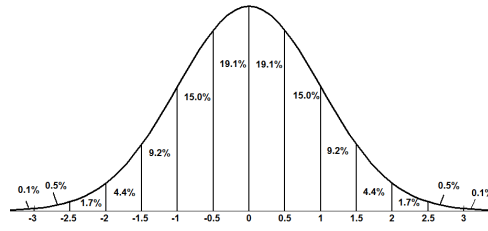
(1) 一元正态分布中的离群点检测

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 2-3 显示 $N(0, 1)$ 的密度函数：

$N(0, 1)$ 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。数据对象落在三倍标准差中心区域之外的概率仅有 0.0027。


 图 2-3: $N(0, 1)$ 的概率密度函数

(2) 混合模型的离群点检测

这里首先介绍下混合模型。混合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。

总的讲，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。重复该过程次，其中是对象的个数。

具体地讲，假定有 K 个分布和 m 个对象 $\chi = \{x_1, x_2, \dots, x_m\}$ 。设第 j 个分布的参数为 α_j ，并 A 设是所有参数的集合，即 $A = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ 。则 $p(x_i|\alpha_j)$ 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 $w_j \{1 \leq j \leq K\}$ 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：

$$p(x|A) = \sum_{j=1}^k w_j P_j(x|\theta_j) \quad (2.1)$$

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：

$$p(\chi|\alpha) = \prod_{i=1}^m P(x_i|\alpha) = \prod_{i=1}^m \sum_{j=1}^k w_j P_j(x_i|\alpha_j) \quad (2.2)$$

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，数据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。

这里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，只

要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 $N_0 = \emptyset$ 。

根据混合模型中公式 $p(x|A) = \sum_{j=1}^k w_j P_j(x|\theta_j)$ 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：

$$L_t(U) = \prod_{x_i \in U} P_U(x_i) = \left((1 - \lambda)^{|M_t|} \prod_{x_i \in M_i} P_{M_i}(x_i) \right) \left(\lambda^{|N_t|} \prod_{x_i \in N_i} P_{N_i}(x_i) \right) \quad (2.3)$$

$$\ln L_t(U) = |M_t| \ln(1 - \lambda) + \sum_{x_i \in M_i} \ln P_{M_i}(x_i) + |N_t| \ln \lambda + \sum_{x_i \in N_i} \ln P_{N_i}(x_i) \quad (2.4)$$

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的很多，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 $1 - \lambda$ 加权）减去该数据对象在正常数据点的分布下的概率（用 λ 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。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，可以考虑另外其他不需要建立模型的检测方法。

2.1.3 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。

图 2-4 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。

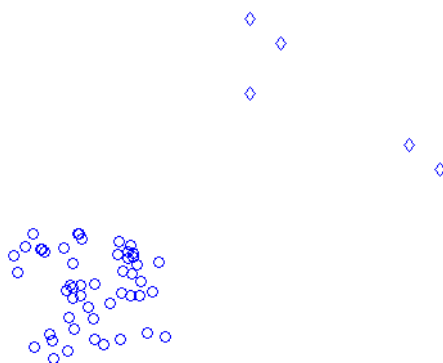


图 2-4: K-Means 算法的聚类图

(2) 基于原型的聚类

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，评估对象属于簇的程度（离群点得分）主要有两种方法：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。

如图 2-5，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，这三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。

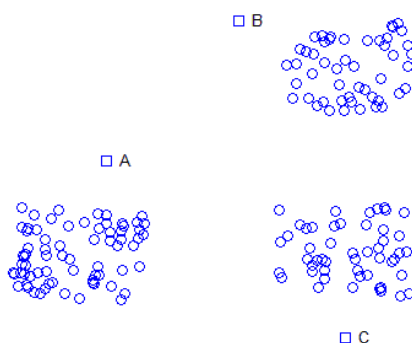


图 2-5: 基于距离的离群点检测

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：

(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，可以使用如下方法：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：取一组不能很好的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；而当前在该集合中的对象被测试，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，其想法是：

- 较小的簇倾向于更加凝聚；
- 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。

“Detect Outlier(Distances)”基于距离的离群点检测，参数设置中可设定要检测的离群点的个数，如图 2-6。

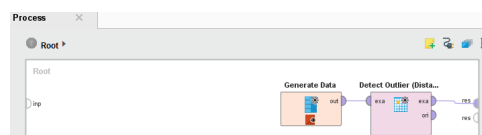


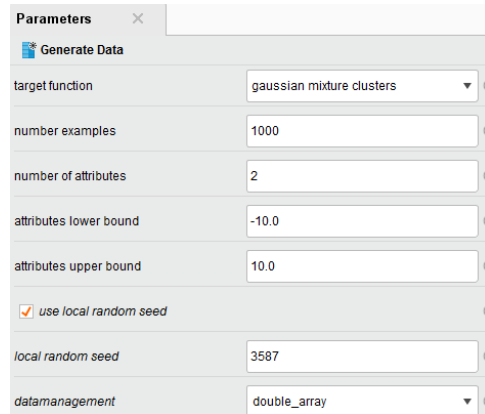
图 2-6: RapidMiner 自带的离群点检测流程

第三方离群点检测插件带有功能更强的离群点检测功能，例如“One-Class LIBSVM Anomaly Score”为半监督的离群点检测操作符。

离群点检测实例

下面，我们自己生成一个数据，来看看离群点检测的功能。

第一步：生成随机数据

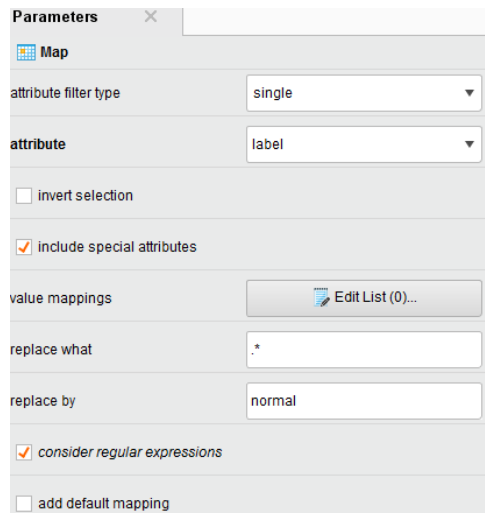


Parameters	
* Generate Data	
target function	gaussian mixture clusters
number examples	1000
number of attributes	2
attributes lower bound	-10.0
attributes upper bound	10.0
<input checked="" type="checkbox"/> use local random seed	
local random seed	3587
datamanagement	double_array

图 2-7：生成随机数据参数设置

调用“Generate Data”生成数据操作符，能帮助我们自动创建一些测试数据，创建参数设置图 2-7。

调用“Map”映射操作符，设置参数如图 2-8，将所有数据类型都转换为 normal 类型。



Parameters	
Map	
attribute filter type	single
attribute	label
<input type="checkbox"/> invert selection	
<input checked="" type="checkbox"/> include special attributes	
value mappings	Edit List (0)...
replace what	.*
replace by	normal
<input checked="" type="checkbox"/> consider regular expressions	
<input type="checkbox"/> add default mapping	

图 2-8：映射操作符参数设置

再次调用“Generate Data”生成数据操作符，参数设置如图 2-9，添加离群点

同样，添加 Map 操作符，参数设置如图 2-10

最后添加“Append”操作符将两个数据集合并，全部操作流程图如图 2-11，数据散点图如图 2-12.

第二步：离群点检测操作符应用

“k-NN Global Anomaly Score”k-NN 全局离群点检测操作符，检测结果如图 2-13

“Local Outlier Factor”基于本地的离群点检测操作符，操作流程如图 2-14，检测结果如图 2-15

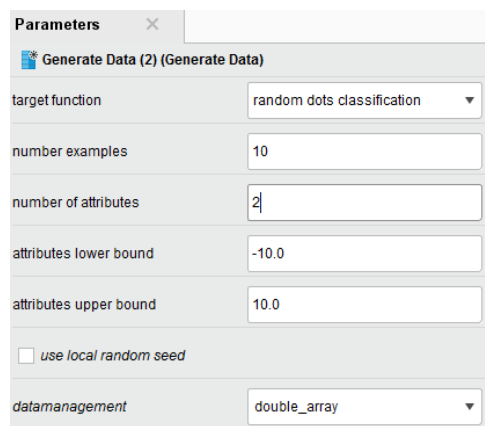


图 2-9: 添加离群点参数设置

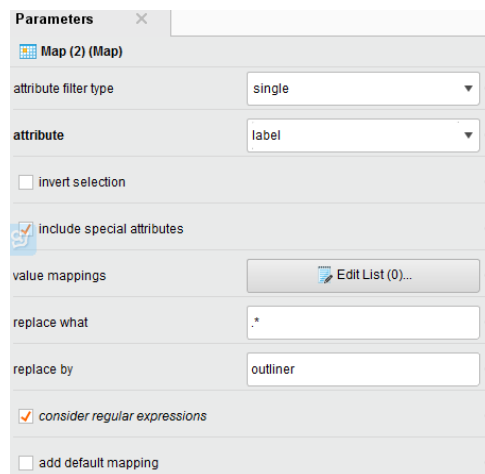


图 2-10: Map 映射离群点设置

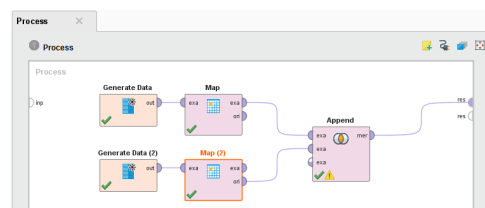


图 2-11: 1. 操作流程图

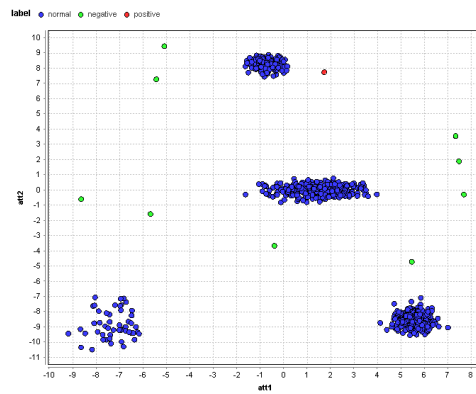


图 2-12: 2. 数据散点图

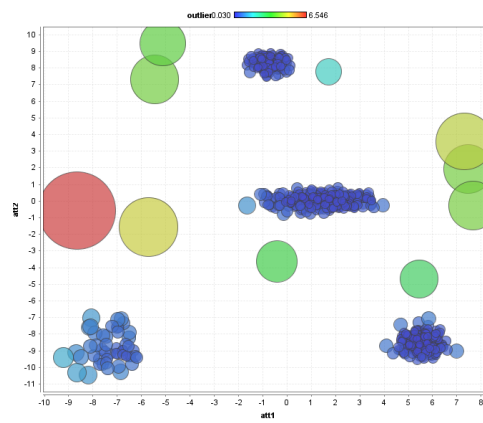


图 2-13: 全局离群点检测气泡图

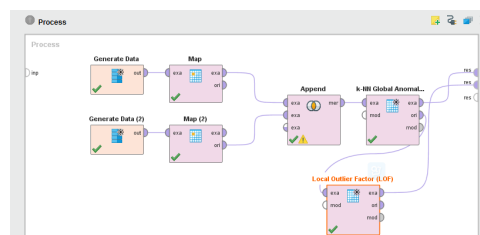


图 2-14: 离群点检测操作流程

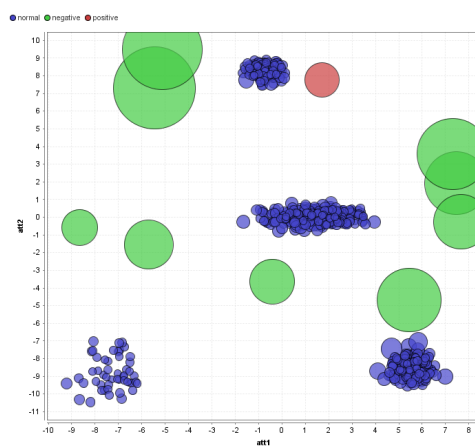


图 2-15: 离群点检测操作流程

3 决策树算法实现流程

3.1 什么是决策树算法

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。

本节下面的内容将开始讲解决策树的具体内容。首先将从一个最经典的决策树分类算法开始说明决策树。

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。假定类别属性有 m 个不同的值： $C_i (i = 1, 2, \dots, m)$ 。

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

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m P_i \log_2(P_i) \quad (3.1)$$

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， $\{a_1, a_2, \dots, a_k\}$ 利用属性 A 将集合 S 划分为 k 个子集， $\{S_1, S_2, \dots, S_k\}$ 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

$$E(A) = - \sum_{j=1}^k \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} I(s_{1j}, s_{2j}, \dots, s_{mj}) \quad (3.2)$$

其中 $I(s_{1j}, s_{2j}, \dots, s_{mj}) = - \sum_{i=1}^m P_{ij} \log_2(P_{ij})$, $P_{ij} = \frac{s_{ij}}{s_{1j} + s_{2j} + \dots + s_{mj}}$ 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

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A) \quad (3.3)$$

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 个不同的值对应的样本集 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来进行属性的选择，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，使分类后的数据集的熵最小。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。

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这些信息至关重要。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如表。

采用 ID3 算法构建决策树模型的具体步骤如下：

(1) 根据公式，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。

$$I(18, 16) = \frac{18}{34} \log_2 \frac{18}{34} - \frac{16}{34} \log_2 \frac{16}{34} = 0.997503 \quad (3.4)$$

(2) 根据公式，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，

表 3-1: 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
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

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：

$$I(11,6) = \frac{11}{17} \log_2 \frac{11}{17} - \frac{6}{17} \log_2 \frac{6}{17} = 0.936667 \quad (3.5)$$

$$I(7,10) = \frac{7}{17} \log_2 \frac{7}{17} - \frac{10}{17} \log_2 \frac{10}{17} = 0.977418 \quad (3.6)$$

$$E(\text{天气}) = \frac{17}{34} I(11,6) + \frac{17}{34} I(7,10) = 0.957043 \quad (3.7)$$

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：

$$I(11,3) = \frac{11}{14} \log_2 \frac{11}{14} - \frac{3}{14} \log_2 \frac{3}{14} = 0.749595 \quad (3.8)$$

$$I(7,13) = \frac{7}{20} \log_2 \frac{7}{20} - \frac{13}{20} \log_2 \frac{13}{20} = 0.934068 \quad (3.9)$$

$$E(\text{是否周末}) = \frac{14}{34} I(11,3) + \frac{20}{34} I(7,13) = 0.858109 \quad (3.10)$$

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：

$$I(15,7) = \frac{15}{22} \log_2 \frac{15}{22} - \frac{7}{22} \log_2 \frac{7}{22} = 0.902393 \quad (3.11)$$

$$I(3, 9) = \frac{3}{12} \log_2 \frac{3}{12} - \frac{9}{12} \log_2 \frac{9}{12} = 0.811278 \quad (3.12)$$

$$E(\text{是否有促销}) = \frac{22}{34} I(15, 7) + \frac{12}{34} I(3, 9) = 0.870235 \quad (3.13)$$

(3) 根据公式，计算天气、是否周末和是否有促销属性的信息增益值。

$$Gain(\text{天气}) = I(18, 16) - E(\text{天气}) = 0.997503 - 0.957043 = 0.04046$$

$$Gain(\text{是否周末}) = I(18, 16) - E(\text{是否周末}) = 0.997503 - 0.858109 = 0.139394$$

$$Gain(\text{是否有促销}) = I(18, 16) - E(\text{是否有促销}) = 0.997503 - 0.870235 = 0.127268$$

(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三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下图：

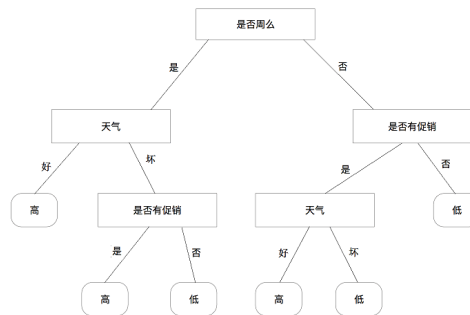


图 3-1: 全局离群点和局部离群点

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

3.2 数据决策树处理

本文主要使用 RapidMiner 中的决策树算子对所给数据进行处理后分别生产两个决策树：

(1) 驻地决策树：这个决策树是以文化程度为 label，以驻地为展示属性来反映不同文化程度数据类构造的驻地规则。

(2) 工种决策树：这个决策树是以工种为 label，反映出不同工种对应的文化程度，驻地的相关性规律。

4 数据处理基础

现实世界中数据大体上都是不完整，不一致的脏数据，无法直接进行数据挖掘，或挖掘结果差强人意。为了提高数据挖掘的质量产生了数据预处理技术。数据预处理有多种方法：数据清洗，数据聚合，数据变换，数据归约等。这些数据处理技术在数据挖掘之前使用，大大提高了数据挖掘模式的质量，降低实际挖掘所需要的时间。

4.1 原始数据集整理与统计

首先，简要的整理一下数据集所提供的数据：

本次所提供的数据集，总共包括 inf 表中：

‘user’‘age’‘sex’‘national’‘hometown’‘major’‘status’

‘residence’‘tel’‘health’‘email’

‘address’‘education’‘ability’‘experience’

15 类数据，其主 tel 为主键。

4.2 数据预处理基本方法整理

存在不完整的（有些感兴趣的属性缺属性值，或仅包含聚集数据）、含噪声的（包含错误的或存在偏离期望的孤立点值）和不一致的（用于分类的编码存在差异）数据是大型的、现实世界数据库或数据仓库的共同特点。数据预处理技术可以改进数据的质量，从而有助于提高其后的数据挖掘过程的精度和性能。由于高质量的决策必然依赖于高质量的数据，因此数据预处理是数据挖掘过程中的主要步骤。数据预处理技术主要包括：数据清理、数据集成、数据变换与数据归约。

4.2.1 数据清洗

其实通俗的来讲，数据清洗就是一个脏数据到”干净”数据的一个处理过程。众所周知，现实世界的数据库一般是不完整的、含噪声的和不一致的。数据清理主要从填充空缺值，识别孤立点，消除噪声，并纠正数据中的不一致这几个方面来对原始数据集进行处理。

(1) 空缺值的处理及其实现方式：

数据集中属性值的缺失并不少见，但是缺失的属性值并不是说其不重要，或者说其与最终的挖掘结果关联不大。缺失值也并不意味着数据就有错误。例如，一个班级在统计班级同学是否获得奖学金，因为奖学金只有少数人获得，因此，没获得者可以使这个字段为空。因此，统计数据信息的记录表格应该允许调查人使用“null”这样的无效值。此外，还有像父属性性别，子属性男或女，那么子属性其中之一避免不了会有空值。所以，

在分析数据时，应当考虑对不完整个的数据进行处理。在处理时可以采用以下办法来处理空缺值。

- 忽略元组；缺少类标号时通常这样处理。
- 忽略属性列；如果一个属性的缺失值占有属性值的 80% 以上，则可以从整个数据集中删除此属性列。
- 人工填写空缺值；通常情况下，此办法较为费时，只适合于数据集较小，空缺值较少的情况下。
- 自动填充空缺值；这个方法呢有三种策略。

策略一：使用一个全局常量填充空缺值，将空缺属性值用同一个常数替换

策略二：使用属性的均值或期望值或者众数进行默认填充。

策略三：可以通过线性回归、基于推理的工具或者决策树归纳确定空缺值的可能值来进行填充。由于使用现有数据的多数信息推测空缺值，所有有更大的机会保持属性之间的关联性。

(2) 噪声数据的清理方法：

噪声数据是一个测量变量中的随机错误或偏差，其包含错误或孤立点值。导致噪声产生的原因有多种，可能是采集设备出了故障，也可能是数据录入或搜集整理的过程出现人为的失误或疏忽，或者数据传输过程中的错误等等。目前，有以下几种处理噪声数据的方法：

①分箱；通过考察“邻居”（周围的值）来平滑存储数据的值。由于分箱方法考虑相邻的值，因此是一种局部平滑方法。按照取值不同可分为：按箱平均值平滑、按箱中值平滑、按箱边界值平滑。例如：有 4、6、17、32、3、9、17、65、23 等 9 个数，分为 3 箱。

箱 1: 4、6、17；

箱 2: 32、3、9；

箱 3: 17、65、23；

接下来，我们分别按照上述几种方法对以上三箱数据进行平滑。

箱 1: 9、9、9；

箱 2: 3、3、3；

箱 3: 17、17、17；

②聚类；孤立点可以被聚类检测，聚类就是将类似的值组织成群或分类，直观地看落在聚类集合之外的值被视为孤立点。我们通过删除离群点来平滑数据。

③人计算机和人工相结合；我们可以先通过已有经验对数据集中明显不符合逻辑的数据点进行处理之后，再通过回归或者数据处理算法对以初步处理后的数据集进行处理。

④回归分析；可以通过让数据适合一个回归函数来平滑数据。如：线性回归涉及找出两个变量的最佳直线，使得一个变量可以预测另一个。多线性回归涉及多个变量，数据适合一个维面，使用回归找出适合数据的数学方程式，能够帮助消除噪声。

4.2.2 数据集成

数据挖掘中经常需要对数据进行聚合，将两个或多个数据源中的数据，存放在一个一致的数据存储设备中，这些数据源可能包括多个数据库、数据立方体或一般文件。

在数据集成时，有许多问题需要考虑，数据一致性和冗余是两个重要的问题。

(1) 数据一致性；

(2) 数据属性值冗余；

(3) 元组重复问题。重复是指对于同一个数据，存在两个或多个相同的元组。

(4) 数据值表现形式冲突的检测与处理。就数据集中的某一具体实体而言，如果其来自不同数据源，那么它的属性值就有可能不同。这可能是因为数据的表示方式、缩减比例（通常用于数值属性）或数据格式编码不同。例如，重量属性可能在一个数据源中以 g 为单位存放，而在另一个数据源中可能 kg 为单位表示。

4.2.3 数据变化

数据变换是将数据转换成适合挖掘的形式，数据变化一般包括以下内容：

(1) 平滑。去掉数据中的噪声，这种技术包括分箱、聚类、回归。

(2) 聚类。对数据进行汇总和聚集，用来为多维度数据分析构造数据立方体。例如，可以以班级为单位来统计和分析班级的成绩情况。

(3) 数据概化。使用概念分层，用高层次概念替换低层次“原始”数据，例如对所调查 customer 的地理位置信息可以将经纬度映射到较高层次的概念，如：市、州甚至国家；对 ip 地址，可以通过对 ip 分段实现泛化。

数据概化，在数据的前期处理过程中很常见，其用来规约数据，尽管经过数据泛化，数据的具体情节被掩盖了，但泛化后的数据更有意义，更有利于人们去直观的理解。

在具体问题的处理过程中，我们常常会遇到数值属性、分类属性等类别的属性需要通过数据泛化来将数据由繁至简。接下来，我们就分别对不同类别的属性的泛化进行简要的分析。

①对于数值属性，我们可以根据数据的分布自动的进行构造；例如，可以用分箱、聚类分析、基于熵的离散化等技术，可以将数值属性泛化。

②对于分类属性，有时可能具有很多个值。如果分类属性是序数属性，则可以使用类似于处理连续性属性的办法，以减少分类值的数目。如果分类属性是标称或者无序的，就需要使用其他办法。比如，就这次我们要解决的餐馆可接受的付款方式，因为，全球有多种被人们采用的消费方式，比如信用卡、现金、兑奖卷等等，而信用卡又分很多公司的或者银行的。因此，我们就要对付款方式进行泛化处理，比如，把类似于信用

卡消费的归类与信用卡，将现金或者借记卡消费的归类于现金，将奖券等归类与其他方式消费等等。如果更深一层，我们可以根据餐馆可接受付款方式的多少将餐馆的消费方式设置成多样的和单一的两大类。

此外，通过说明属性值的偏序或全序，可以很容易的定义概念分层。

(4) 规范化。数据规范化是将原来的度量值转换为无量纲的值，即将属性数据按比例缩放，使之落入一个小的特定区间。对于基于距离的方法，规范化可以帮助平衡具有较大初始值域的属性与较小初始值域的属性可比性。常用的规范化方法有以下几种：

- 最小—最大规范化；
- z-score 规范化；
- 小数定标规范化；

(5) 属性构造；

利用已知属性，可以构造新的属性，以更好地刻画数据的特性，帮助整个数据挖掘的过程。

在模型的构建的过程中，越是经过数据集经预处理之后留下关联度高和与最终要解决问题 confidence 高的属性，得到结果的正确性越高。我们知道，数据集的特征维数太高容易导致维灾难，而唯独太低又不能有效地捕获数据集中重要的信息。在实际应用中，通常需要对数据集中的特征进行处理来创建新的特征。有原始特征创建新的特征，其目的指在帮助特高挖掘结果正确度的精度以及对高维度数据结构的理解。

(6) 数据离散化；

聚类、分类或关联分析中的某些算法要求数据是分类属性，因此需要对数值属性进行离散化。

4.2.4 数据规约

数据的不同视角反映出来的信息可能是不同的。

数据归约技术可以用来得到数据集的压缩表示，它比源数据集小得多，但仍然接近于保持原数据的完整性，这样在归约的数据集上挖掘将更有效，并能产生相同的分析结果。数据归约方法有以下几种：

(1) 维度规约和特征变换

维度规约是指通过使用数据编码或变换，得到原始数据数据的规约或“压缩”表示。唯独规约有多方面的好处，最大的好处是，如果维度较低，许多数据挖掘算法的效果会更好。一方面是因为维规约可删除不相关的特征并降低噪声，另一方面是因为维灾难。在本文中，我们将要对用户信息数据进行聚类分析，如果原始数据集就对用户进行聚类划分，因为用户信息数据集的维度较高，所划分的簇中样本点之间的密度和距离之间的

定义就变得没有多大意义了。此外，使用维规约，使模型涉及更少的特征，因而可以产生更容易理解的模型，可以降低数据挖掘算法的时间和空间复杂度。

接下来，简要介绍两种有效的有损规约方法：

①离散小波变换 (DWT)；

离散小波变换是一种线性信号处理技术，当用于数据向量 I 时，将它转换成数值上不同的小波系数的向量，两个向量具有相同的长度。小波变换也可以用于多维数据，如数据立方体。

②主成分分析；

设待压缩的数据由 N 个元组或数据向量组成，取 k 个维，主要成分分析搜索 C 个最能代表数据的 $K-1$ 维正交向量，这样，原来的数据投影到一个较小的空间，导致数据压缩。不像属性子集选择通过保留原属性集的一个子集来减少属性集的大小，主要成分分析通过创建一个替换的、较小的变量集来组合属性的精华，原数据可以投影到该较小的集合中。与数据压缩的小波变换相比，主要成分分析能较好地处理稀疏数据，而小波变换更适合高维数据。

(2) 抽样

选样作为一种数据归约技术，是用较小的随机样本子集表示大的数据集，选样种类：一是简单选择 n 个样本，不放回：由 N 个元组中抽取 n 个样本，其中任何元组被抽取的概率均为 $\frac{1}{N}$ 。

二是简单选择 n 个样本，回放：一个元组被抽取后，它又被放回，以便可以再次抽取。

三是聚类选样：先将所有元组聚类，在从每个聚类中随机选取一个样本。

四是分层选样：将元组划分成不相交的部分，称作层，通过对每一层的简单随机选样得到总体样本的分层选样。

(3) 数值压缩

数值归约技术可以通过选择替代的、“较小的”数据表示形式来减少数据量。这些技术可以是有参的，也可以是无参的。对于有参方法，使用一个模型来评估数据，是的只需要存放参数，而不是实际数据。无参方法包括直方图、聚类等等。

①回归和对数线性模型。

回归和对数线性模型可以用来近似给定数据，在线性回归中，我们可以分局样本点的聚集程度建模，并使其走向适合一条直线，可以用以下公式将随机变量 Y 表示为另一随机变量 X 的线性函数， $Y = \alpha + \beta X$ ，系数 α 和 β 称为回归系数，其值可以通过最小平方法求得，使得分离数据的实际直线与该直线的误差最小。对数线性模型近似离散的多维概率分布。基于较小的方体形成数据立方体的格，该方法可以用于估计具有离散属性集的基本方体中每个单元的概率。

其次，在对一个属性进行回归分析的时候，往往会涉及其他属性，尤其以线性回归作为处理方法。这时候涉及二维属性是否符合线性回归，就存在一个相关系数取值是否

为 1 的问题，即 $|\rho(X, Y)| = 1$ 。

相关系数 $\rho(X, Y)$ 反映了属性之间的线性关系。以离散属性值为例，当 $|\rho(X, Y)| = 1$ 时，其属性之间几乎成线性关系，即属性之间的样本点几乎全部落在 $Y = \alpha + \beta X$ 上。若 $|\rho(X, Y)| = 0$ 则属性之间无线性关联。现给出相关系数的求值公式：

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \quad (4.1)$$

其中， $\text{cov}(X, Y)$ 为属性 X, Y 之间的协方差， $D(X), D(Y)$ 为属性 X, Y 的方差。其中 $\text{cov}(X, Y) = E(XY) - E(X)E(Y)$ 。

②直方图。直方图使用分箱近似数据分布，是一种流行的数据归约形式，属性 (的直方图将 (的数据分布划分为不相交的子集或桶。桶安放在水平轴上，而桶的高度 G 和面积 H 是该桶所代表的值的平均频率。如每个桶只代表单个属性值 K 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。

③聚类。聚类技术将数据元组视为对象，将对象划分为群或聚类，使得在一个聚类中的对象“相似”。通常，类似性基于距离，用对象在空间中的接近程度定义，聚类的质量可以用直径表示，直径时一个聚类中两个任意对象的最大距离质心距离是聚类质量的另一种度量，是聚类质心到每个聚类对象的平均距离。在数据归约时，用数据的聚类表示替换实际数据。该技术的有效性依赖于数据的性质，如果数据能够组织成不同的聚类，该技术有效得多。

(4) 特征选择

特征选择是指从一组已知特征集合中选择最具有代表性的特征子集，使其保留原有数据的大部分信息，即所选择的特征子集可以像原来的全部特征一样用来正确区分数据集中的每个数据对象。

5 数据预处理实现及结果分析

在本文，将依次按照数据准备、数据清洗、数据变换、数据聚合以及特征值构造和维度规约的顺序对所挖掘数据集进行转换。

5.1 数据准备

首先针对每一张表进行冗余数据的校验处理，删掉重复项。比如在有些表中会有相同的元组数据，这就可以删除。

此过程借助于 Navicat 工具的自动排序功能，简化了原始数据集，有利于进一步对数据进行处理。

此外，依据 inf 这张表，设置 age 过滤范围，并以此确认 age 位于 18 到 70 之间。通过上述数据限定范围，对 inf 表进行范围限定处理。

5.2 数据清洗—数据集缺失值的处理

在本文中，本文中主要采用两类处理方法来对表中出现的缺失值进行处理。首先通过对数据录入时的限制函数对指定的录入数据类进行限制，然后在数据读取后对非必需类数据进行统一缺省处理。

5.2.1 user 数据集中缺失值的处理：

观察 inf 有关数据集，我们发现 user 相关数据属性出现缺失值现象：主要集中在读入数据的 No.10 No.25:

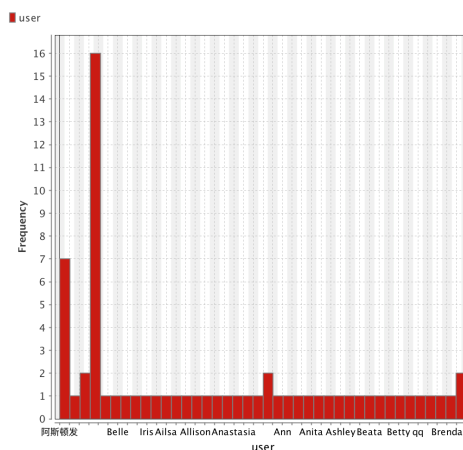


图 5-1: 缺失值现象

观察数据，可以发现图 5-1 中 Belle 这个 user 前的 16 个样本为空样本。现在，做一下几个工作：

- 首先，添加 Filter Examples 算子到 Process；
- 其次，添加 Entry；

设置 user 属性 is not in, 即过滤掉读入的 user 属性为缺省的数据项。
现在，给出此次过滤后的图 5-2:

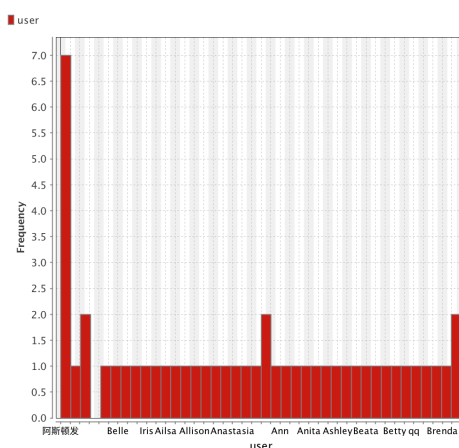


图 5-2: 过滤后的图片

观察上述分布图 5-2，会发现 user 缺失值的已经全部处理完成。

5.2.2 年龄的处理

观察用户信息有关数据集，会发现与用户年龄数据属性出现异常现象，主要在在 inf 表，其中有以下几个字段出现丢失值现象：

首先，分析以下分布图 5-3:

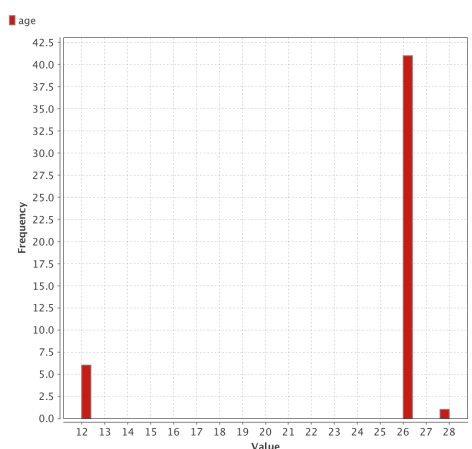


图 5-3: age 分布图

简要的对此属性值出现的情况做一简单统计：

<18 的，即 FALSE，有 6 人；

>=18 且 <=70，即 TRUE，有 42 人；

设置 Filter Examples 算子，添加 age 过滤条件过滤不符合的数据，如图 5-4:

age	<	70
age	≥	18

图 5-4: 添加 age 过滤条件

保存后运行算法，因此，得出 age 过滤后的图像，如图 5-5:

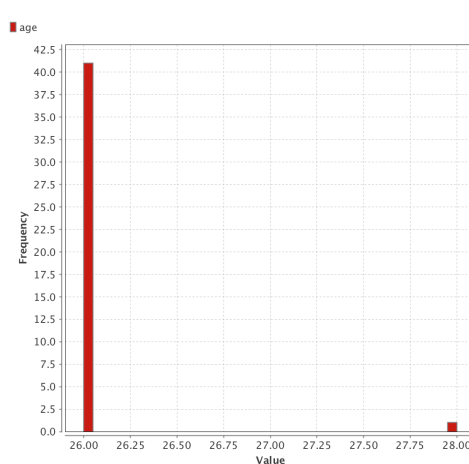


图 5-5: 过滤后的图

如图，age 不符合条件的数据已经全部处理完成

5.2.3 关于专业的预处理

经过前面对数据集的相关的处理，观察到有关专业的数据还存在有异常项，如图 5-6 所示:

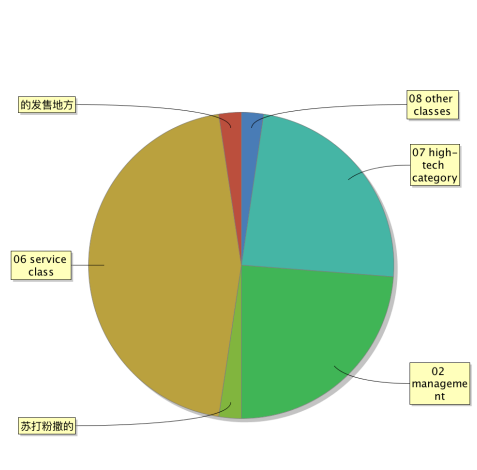


图 5-6: 专业预处理前

很显然可以从图 5-6 看出存在两个属性值的发售地方与苏打粉撒的为异常属性值，因此我们同样采取 Filter Examples 算子，设置过滤条件如图 5-7:

major	is not in		
major	starts with	0	

图 5-7: 设置过滤条件

保存后运行算子得到输出的 major 数据分布如图 5-8:

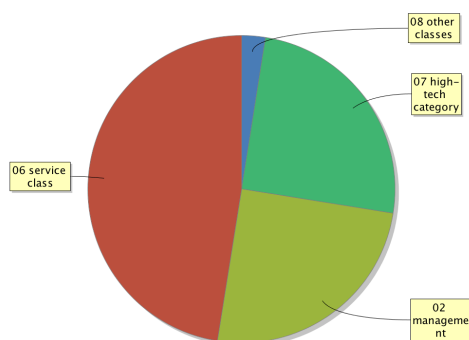


图 5-8: 工种预处理后的数据集合

由此，我们就得到了所有数据预处理后的数据集合，总计 40 examples。

5.3 数据变换

根据任务书要求，从工种和驻地以及文化程度这三个方面来对数据集进行数据变换处理。

5.3.1 关于驻地的相关数据处理及决策树生成

经过前面对数据集的相关的处理，得到了 40 条合理的数据集，接下来，依次对每个属性进行具体的相关分析。

(1) 设置关联属性；

观察表中数据，选取工种、驻地、文化程度三个属性设置相关，实现聚类。

具体做法：

首先，利用 RapidMiner 工具，使用 Read Database 读入数据库数据；

其次，利用 Filter Examples 完成数据预处理；

然后，利用 Select Attributes 选择所需三个属性，如图 5-9：

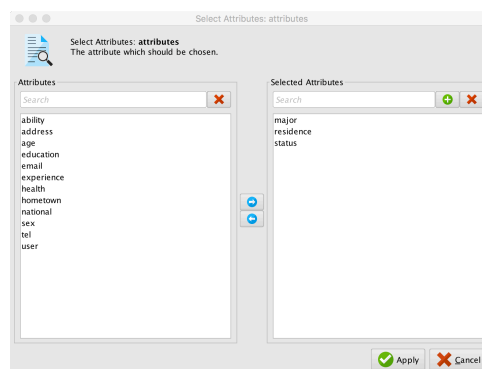


图 5-9: 选择所需三个属性

设置好算子后运行可见，数据 charts 如图 5-10：

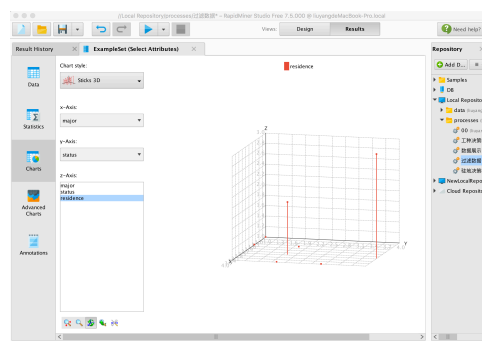


图 5-10: 数据 charts

x 轴代表工种属性，y 轴代表文化程度属性，z 轴代表驻地属性，通过 Sticks 3D 图像来表示出数据分布

(2) 对所选数据设置处理规则

选用 Set Role 算子，对 major, status, residence 三个属性分别设置处理规则如图 5-11：

目的为将文化程度设置为 label，工种跟驻地设置为 regular

处理结果如图 5-12：

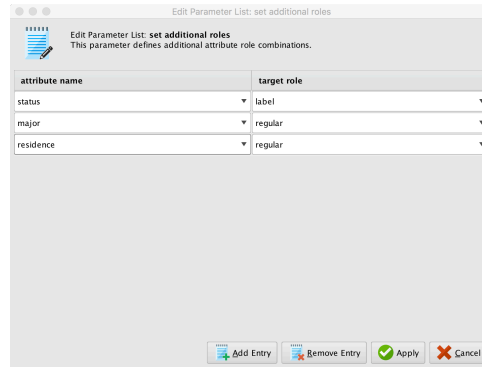


图 5-11: 设置处理规则

Row No.	status	major	residence
1	Bachelor de...	08 other Co...	State / Prov...
2	Bachelor de...	07 high-sec...	State / Prov...
3	Bachelor de...	07 high-sec...	State / Prov...
4	Bachelor de...	07 high-sec...	State / Prov...
5	Bachelor de...	07 high-sec...	State / Prov...
6	Bachelor de...	07 high-sec...	State / Prov...
7	Bachelor de...	07 high-sec...	State / Prov...
8	Bachelor de...	07 high-sec...	State / Prov...
9	Bachelor de...	07 high-sec...	State / Prov...
10	Bachelor de...	07 high-sec...	State / Prov...
11	Undergradu...	02 manage...	State / Prov...
12	Undergradu...	02 manage...	State / Prov...
13	Undergradu...	02 manage...	State / Prov...
14	Undergradu...	02 manage...	State / Prov...
15	Undergradu...	02 manage...	State / Prov...
16	Undergradu...	02 manage...	State / Prov...
17	Undergradu...	02 manage...	State / Prov...
18	Undergradu...	02 manage...	State / Prov...
19	Undergradu...	02 manage...	State / Prov...
20	high-school...	06 service C...	County, city...

图 5-12: 处理结果

如图 5-12 可见我们已经将文化程度设置为 label，工种跟驻地设置为 regular

(3) 生成驻地决策树

运用 Decision Tree 算子，通过对 label status 属性，regular major, residence 属性进行决策树生成如图 5-13:

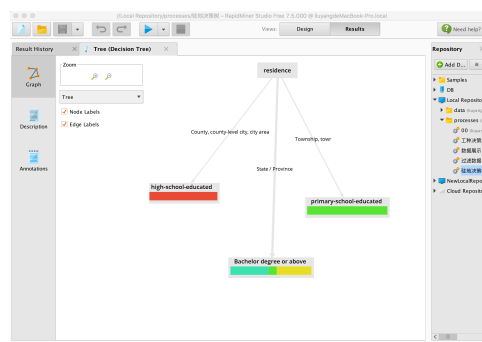


图 5-13: 驻地决策树

从图 5-13 可以看出经过数据分析后，不同的驻地导向了不同的文化程度，比如：驻地为乡镇就导向小学文化程度。

Tree

residence = County, county-level city, city area: high-school-educated

{Bachelor degree or above=0, primary-school-educated=0, Undergraduate-educated=0, high-school-educated=9} residence = State / Province: Bachelor degree or above

{Bachelor degree or above=10, primary-school-educated=2, Undergraduate-educated=9, high-school-educated=0} residence = Township, town: primary-school-educated

{Bachelor degree or above=0, primary-school-educated=10, Undergraduate-educated=0, high-school-educated=0}

5.3.2 关于工种的相关数据处理及决策树生成

经过前面对数据集的相关的处理，得到了 40 条合理的数据集，接下来，依次对每个属性进行具体的相关分析。

(1) 设置关联属性；

观察表中数据，选取工种、驻地、文化程度三个属性设置相关，实现聚类
具体做法：

首先，利用 RapidMiner 工具，使用 Read Database 读入数据库数据；

其次，利用 Filter Examples 完成数据预处理；

然后，利用 Select Attributes 选择所需三个属性，如图 5-14：

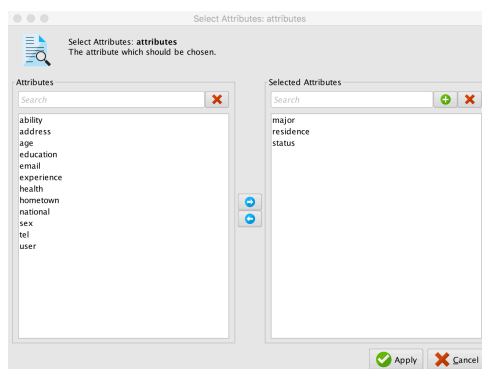


图 5-14: 设置关联属性

选择工种，驻地，文化程度为所需的三个属性

设置好算子后运行可见，数据 charts 如图 5-14：

Name	Type	Missing	Filter (3 / 3 attributes)	Search for Attribute
major	Nominal	0	08 other classes (1)	06 service class (19)
status	Nominal	0	high-school-educate...	primary- [...] ated (12)
residence	Nominal	0	County, [...] area (9)	State / Province (21)

图 5-15: 数据 charts

如图 5-15，我们所选择的工种，驻地，文化程度三个属性已经被挑选出来。

(2) 对所选数据设置处理规则



图 5-16: 设置处理规则

选用 Set Role 算子，对 major, status, residence 三个属性分别设置处理规则如图 5-16：目的为将工种设置为 label，文化程度跟驻地设置为 regular。处理结果如图 5-17：

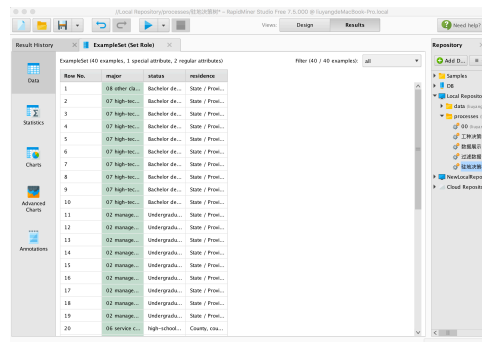


图 5-17: 处理结果

如图 5-17 可见，我们已经将工种设置为 label，文化程度跟驻地设置为 regular。

(3) 生成工种决策树

运用 Decision Tree 算子，通过对 label major 属性，regular status, residence 属性进行决策树生成如图 5-18：

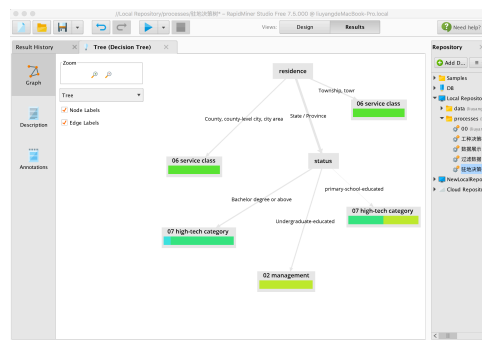


图 5-18: 工种决策树

如图 5-18 可见，不同的工种经过数据处理后对应到不同的学历及驻地，衍生出对应规则。如：02 管理工种就对应本科文化程度，省级驻地

Tree

residence = County, county-level city, city area: 06 service class 08 other classes=0, 07 high-tech category=0, 06 service class=9, 02 management=0 residence = State / Province

| status = Bachelor degree or above: 07 high-tech category 08 other classes=1, 07 high-tech category=9, 06 service class=0, 02 management=0

| status = Undergraduate-educated: 02 management 08 other classes=0, 07 high-tech category=0, 06 service class=0, 02 management=9

| status = primary-school-educated: 07 high-tech category 08 other classes=0, 07 high-tech category=1, 06 service class=0, 02 management=1

residence = Township, town: 06 service class 08 other classes=0, 07 high-tech category=0, 06 service class=26, 02 management=0

6 实验结果及其分析

6.1 数据集来源

本文采用自建数据库，使用 java web 页面进行数据录入，如图 6-1 所示：

图 6-1: 数据录入

提取 mysql 数据库中供给的数据集。

依据 inf 表，选用对应数据作为实验数据集，然后根据数据反映情况进行数据决策树生成及最终完成预测。

6.2 度量标准

本文的目的是针对海量数据的异常项检测及分类方法研究完成最终的预测行为，从而对未来的数据处理进行有针对性的指导。质量的度量标准主要包括统计精度度量方法和决策支持精度度量方法两类。统计精度度量方法中的平均绝对偏差 MAE(mean absolute error) 易于理解，可以直观地对推荐质量进行度量，是最常用的一种推荐质量度量方法，本文采用平均绝对偏差 MAE 作为度量标准。平均绝对偏差 MAE 通过计算预测的用户评分与实际的用户评分之间的偏差度量预测的准确性，MAE 越小，推荐质量越高。

设预测的用户评分集合表示为 $p_1, p_2, p_3, \dots, p_n$ ，对应的实际用户评分集合为 $q_1, q_2, q_3, \dots, q_n$ ，则平均绝对偏差 MAE 定义为：

$$MAE = \frac{\sum_{i=1}^n |p_i - q_i|}{n} \quad (6.1)$$

6.3 用户分类

对于用户信息，根据用户属性之间关联性，将用户通过数据规则自动分类。

6.3.1 用户信息统计

此次，涉及数据信息属性有三个，分别是 major,residence,status 依据预处理之后的信息数据，得知此次总共调查了 40 个数据集的信息。

6.4 数据预测

6.4.1 数据预测准备

经过之前的多级数据处理后我们可以都到关于驻地跟文化程度两个关于驻地，专业，文化程度的决策树，以此为前提，我们使用 RapidMiner 的相关算子进行对应的数据预测。

6.4.2 数据预测操作

使用之前处理过的数据，并在其后添加算子 Apply Model 如图 6-2:

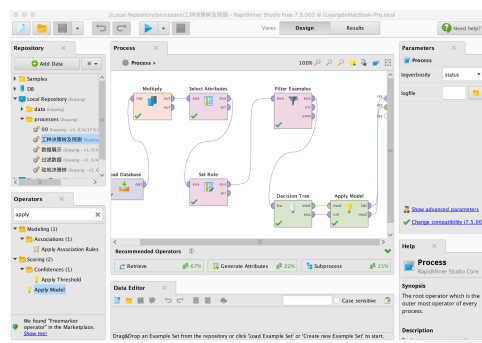


图 6-2: 添加算子 Apply Model

运行 Process，可以得到如下的预测图：

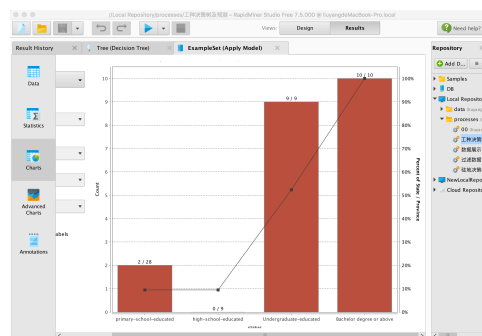


图 6-3: 预测图

图 6-3 反应了对不同文化程度对应不同驻地的预测，例如：小学文化程度对应驻地 为省级的概率为 20

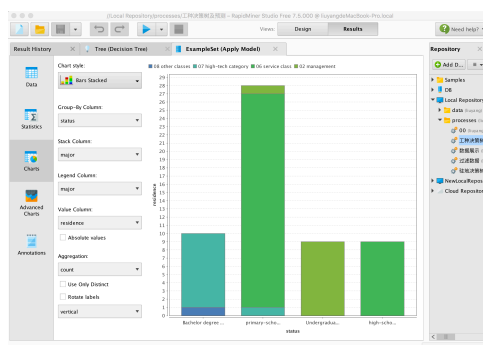


图 6-4: 预测图 2

图 6-4 以条形图的方式反应出不同的文化程度对应不同工种的预测比例，其中不同的工种代表不同的颜色，对应在不同文化程度的条形图中分布比例。

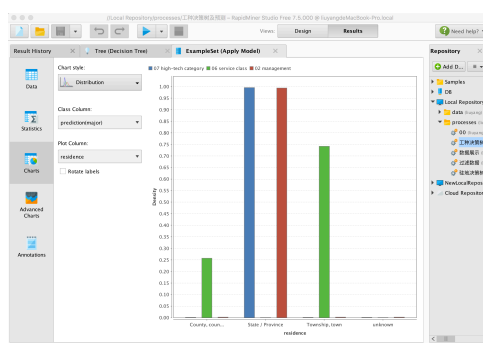


图 6-5: 预测图 3

图 6-5 反应出不同工种对应不同驻地的预测，例如：省级驻地对应 07 高科技类工种跟 02 管理类工种的概率均很高

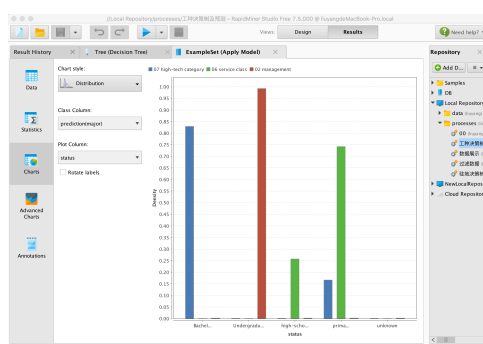


图 6-6: 预测图 4

图 6-6 反应出不同文化程度对应不同工种的预测，例如：本科文化程度对应从事 02 管理类工种的概率就很高

6.5 实验结果分析

依照上述数据预测结果，我们可以得到不同文化程度对应不同驻地的预测值（confidence），以及不同工种对应不同驻地的预测值，不同文化程度对应不同专业的预测值。即通过对样本数据的异常项检测及分类方法研究，实现了数据属性之间的相似性度量研究，并通过决策树算法产生基于样本数据的预测算法，实现数据的利用最大化。

7 结论

本文结合简历系统信息数据以产生对文化程度，工种，驻地的相关性预测。针对整个数据集，本文中所描述用户信息数据存在稀疏性问题，对于此问题，本文中采用对用户信息数据进行特征选择后以地理位置作为初始分类原则，将用户归类，然后对某一类用户进行聚类分析（这一过程本文中是通过对已处理后的数据属性来进行的）。之后，所产生的用户集具有统一分析的参考价值。

接下来，本文中对录入信息也做了一些预处理，进行了异常项检测，并对异常项进行了处理，以保障进入决策树的数据集是不包含脏数据的可以直接进行分类研究的可靠数据。

然后，本文中构造两个不同的决策树以对不同关联 lable 的相似度进行分别研究，探寻驻地，工种，文化程度之间的相关性。

最后，本文中通过 Apily Model 对不同文化程度对应工种，对应驻地进行了预测值分布预测，以及对不同驻地对应不同工种进行了预测值分布预测，揭示了海量数据的数据价值。

致谢

本次论文是在我的导师张留美张老师的亲切关心和悉心指导下完成的。他指导学生细心、认真，大到整个设计思路是否正确，小到一个数据属性字段的含义。从论文课题的选择到最终完成，张老师都始终给予我细心的指导和不懈的支持。在此谨向张老师致以诚挚的谢意和崇高的敬意。

此外，我还要感谢我的舍友，是他们开朗的性格，以及活跃的解决问题思路，让我找到算法设计的初模。当然，他们也在闲暇之余，和我一起讨论一些算法的思路，让我受益匪浅。在这，对你们说声谢谢，这份舍友情将一直会陪伴着我。

在我的十几年求学历程里，离不开父母的鼓励和支持，是他们辛勤的劳作，无私的付出，为我创造良好的学习条件，我才能顺利完成完成学业，感激他们一直以来对我的抚养与培育。

或许，毕业设计，将是我大学生涯交上的最后一个作业了。真的想借此机会对四年以来给我帮助的所有老师、同学、朋友说声谢谢，你们的友谊将是我人生的最大的财富，是我生命中不可或缺的一部分。

在整个系统的设计过程中，遇到过很多障碍，挫折，然而当整个系统完整的展现在自己的面前时，这种喜悦是只可意会的。几个月来忙碌紧张而又有条不紊的毕业设计，使我有机会对本专业的基本理论、专业知识和基本技术有了更深入的了解和体会，使我在大学中所学到的知识得到了系统和升华，真正达到了学以致用。

这次做论文的经历会使我终身受益，我感受到做一件事情，从开始到结束的那份收获。也让我明白认真做一件事情的重要性。希望这次的经历能让我在以后学习中激励我继续进步。

附录

在本次论文的写的过程，会用到以下算法，现我将简要将这些算法介绍。

(1) 决策树学习：根据数据的属性采用树状结构建立决策模型。决策树模型常常用来解决分类和回归问题。常见的算法包括 CART (Classification And Regression Tree)、ID3、C4.5、随机森林 (Random Forest) 等。

(2) 回归算法：试图采用对误差的衡量来探索变量之间的关系的一类算法。常见的回归算法包括最小二乘法 (Least Square)、逻辑回归 (Logistic Regression)、逐步式回归 (Stepwise Regression) 等。

(3) 聚类算法：通常按照中心点或者分层的方式对输入数据进行归并。所有的聚类算法都试图找到数据的内在结构，以便按照最大的共同点将数据进行归类。常见的聚类算法包括 K-Means 算法以及期望最大化算法 (Expectation Maximization) 等。

(4) 人工神经网络：模拟生物神经网络，是一类模式匹配算法。通常用于解决分类和回归问题。人工神经网络算法包括感知器神经网络 (Perceptron Neural Network)、反向传递 (Back Propagation) 和深度学习等。

建立了决策树模型后需要给出该模型的评估值，这样才可以来判断模型的优劣。学习算法模型使用训练集 (training set) 建立模型，使用校验集 (test set) 来评估模型。本文通过评估指标和评估方法来评估决策树模型。评估指标有分类准确度、召回率、虚警率和精确度等。而这些指标都是基于混淆矩阵 (confusion matrix) 进行计算的。

混淆矩阵是用来评价监督式学习模型的精确性，矩阵的每一列代表一个类的实例预测，而每一行表示一个实际的类的实例。以二类分类问题为例，如下表所示：

表 7-1: 混淆矩阵

实际的类	预测的类		
	类 =1	类 =0	
类 =1	TP	FN	P
类 =0	FP	TN	N

P (Positive Sample): 正例的样本数量。

N(Negative Sample): 负例的样本数量。

TP(True Positive): 正确预测到的正例的数量。

FP(False Positive): 把负例预测成正例的数量。

FN(False Negative): 把正例预测成负例的数量。

TN(True Negative): 正确预测到的负例的数量。

根据混淆矩阵可以得到评价分类模型的指标有以下几种。

分类准确度，就是正负样本分别被正确分类的概率，计算公式为：

$$Accuracy = \frac{TP + TN}{P + N} \quad (7.1)$$

召回率，就是正样本被识别出的概率，计算公式为：

$$Recall = \frac{TP}{P} \quad (7.2)$$

虚警率，就是负样本被错误分为正样本的概率，计算公式为：

$$FPrate = \frac{FP}{N} \quad (7.3)$$

精确度，就是分类结果为正样本的情况真实性程度，计算公式为：

$$Precision = \frac{TP}{TP + FP} \quad (7.4)$$

评估方法有保留法、随机二次抽样、交叉验证和自助法等。

保留法 (holdout) 是评估分类模型性能的最基本的一种方法。将被标记的原始数据集分成训练集和检验集两份，训练集用于训练分类模型，检验集用于评估分类模型性能。但此方法不适用样本较小的情况，模型可能高度依赖训练集和检验集的构成。

随机二次抽样 (random subsampling) 是指多次重复使用保留方法来改进分类器评估方法。同样此方法也不适用训练集数量不足的情况，而且也可能造成有些数据未被用于训练集。

交叉验证 (cross-validation) 是指把数据分成数量相同的 k 份，每次使用数据进行分类时，选择其中一份作为检验集，剩下的 $k-1$ 份为训练集，重复 k 次，正好使得每一份数据都被用于一次检验集 $k-1$ 次训练集。该方法的优点是尽可能多的数据作为训练集数据，每一次训练集数据和检验集数据都是相互独立的，并且完全覆盖了整个数据集。也存在一个缺点，就是分类模型运行了 K 次，计算开销较大。

自助法 (bootstrap) 是指在其方法中，训练集数据采用的是有放回的抽样，即已经选取为训练集的数据又被放回原来的数据集中，使得该数据有机会能被再一次抽取。用于样本数不多的情况下，效果很好。